



SBRC 2019

Gramado | RS

XXXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos

6 a 10 de maio

Gramado | RS

ANAIS DO
CoUrb 2019

sbrc2019.sbc.org.br



SBRC **2019** **Gramado | RS**

XXXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos

6 a 10 de maio

Gramado | RS

**ANAIS DO
CoUrb 2019**

sbrc2019.sbc.org.br

EDITORA

Sociedade Brasileira de Computação (SBC)

ORGANIZAÇÃO

Alberto Egon Schaeffer Filho (UFRGS)

Weverton Luis da Costa Cordeiro (UFRGS)

Stênio Fernandes (UFPE)

Leandro Villas (UNICAMP)

Daniel Guidoni (UFSJ)

Thiago Henrique Silva (UTFPR)

Bruno Kimura (UNIFESP)

Roberto Sadao Yokoyama (UTFPR)

REALIZAÇÃO

Sociedade Brasileira de Computação (SBC)

Instituto de Informática - Universidade Federal
do Rio Grande do Sul (UFRGS)

Laboratório Nacional de Redes de
Computadores (LARC)

Copyright ©2019 da Sociedade Brasileira de Computação
Todos os direitos reservados

Capa: Dórian Fogliatto

Produção Editorial: Carlos Raniery P. dos Santos (UFSM)

Cópias Adicionais:

Sociedade Brasileira de Computação (SBC)

Av. Bento Gonçalves, 9500- Setor 4 - Prédio 43.412 - Sala 219

Bairro Agronomia - CEP 91.509-900 - Porto Alegre - RS

Fone: (51) 3308-6835

E-mail: sbc@sbc.org.br

III Workshop de Computação Urbana (3: 2019: Gramado, RS).

Anais / III Workshop de Computação Urbana - CoUrb; organizado por Alberto Egon Schaeffer Filho, Weverton Luis da Costa Cordeiro, Stênio Fernandes, Bruno Kimura, Daniel Guidoni, Leandro Villas, Roberto Sadao Yokoyama, Thiago Henrique Silva - Porto Alegre: SBC, 2019

224 p. il. 21 cm.

Vários autores

Inclui bibliografias

1. Redes de Computadores. 2. Sistemas Distribuídos. I. Schaeffer-Filho, Alberto Egon II. Cordeiro, Weverton Luis da Costa III. Fernandes, Stênio IV. Kimura, Bruno V. Guidoni, Daniel VI. Villas, Leandro V. Yokoyama, Roberto Sadao VI. Silva, Thiago Henrique VII Título.

Sociedade Brasileira de Computação

Presidência

Lisandro Zambenedetti Granville (UFRGS), Presidente

Thais Vasconcelos Batista (UFRN), Vice-Presidente

Diretorias

Renata de Matos Galante (UFGRS), Diretora Administrativa

Carlos André Guimarães Ferraz (UFPE), Diretor de Finanças

Antônio Jorge Gomes Abelém (UFPA), Diretor de Eventos e Comissões Especiais

Renata Mendes de Araujo (UPM), Diretora de Educação

José Viterbo Filho (UFF), Diretor de Publicações

Claudia Lage Rebello da Motta (UFRJ), Diretora de Planejamento e Programas Especiais

Marcelo Duduchi Feitosa (CEETEPS), Diretor de Secretarias Regionais

Eliana Almeida (UFAL), Diretora de Divulgação e Marketing

Ricardo de Oliveira Anido (UNICAMP), Diretor de Relações Profissionais

Esther Colombini (UNICAMP), Diretora de Competições Científicas

Raimundo José de Araújo Macêdo (UFBA), Diretor de Cooperação com Sociedades Científicas

Cláudia Cappeli (UNIRIO), Diretora de Articulação com Empresas

Contato

Av. Bento Gonçalves, 9500

Setor 4 - Prédio 43.412 - Sala 219

Bairro Agronomia

91.509-900 – Porto Alegre RS

CNPJ: 29.532.264/0001-78

<http://www.sbc.org.br>

Laboratório Nacional de Redes de Computadores (LARC)

Diretor do Conselho Técnico-Científico

Paulo André da Silva Gonçalves (UFPE)

Vice-Diretora do Conselho Técnico-Científico

Rossana Maria de Castro Andrade (UFC)

Diretor Executivo

Ronaldo Alves Ferreira (UFMS)

Vice-Diretor Executivo

Danielo Gonçalves Gomes (UFC)

Membros Institucionais

SESU/MEC, INPE/MCT, UFRGS, UFMG, UFPE, UFCG (ex-UFPB Campus Campina Grande), UFRJ, USP, PUC-Rio, UNICAMP, LNCC, IME, UFSC, UTFPR, UFC, UFF, UFSCar, IFCE (CEFET-CE), UFRN, UFES, UFBA, UNIFACS, UECE, UFPR, UFPA, UFAM, UFABC, PUCPR, UFMS, UnB, PUC-RS, PUCMG, UNIRIO, UFS e UFU.

Contato

Universidade Federal de Mato Grosso do Sul (UFMS)

Faculdade de Computação

Caixa Postal 549

CEP 79.070-900 Campo Grande - MS - Brasil

<http://www.larc.org.br>

Organização do SBRC 2019

Coordenadores Gerais

Alberto Egon Schaeffer Filho (UFRGS)
Weverton Luis da Costa Cordeiro (UFRGS)

Coordenadores do Comitê de Programa

Antônio Jorge Gomes Abelém (UFPA)
Fabiola Gonçalves Pereira Greve (UFBA)

Coordenador de Palestras e Tutoriais

Ítalo Cunha (UFMG)

Coordenador de Painéis e Debates

Artur Ziviani (LNCC)

Coordenador de Minicursos

Miguel Elias Mitre Campista (UFRJ)

Coordenador de Workshops

Stênio Fernandes (UFPE)

Coordenador do Salão de Ferramentas

Leandro Villas (UNICAMP)

Coordenador do Concurso de Teses e Dissertações

Daniel Fernandes Macedo (UFMG)

Coordenadores do Hackaton

Raquel Lopes (UFMG)
Luis Carlos de Bona (UFPR)

Comitê de Organização Local

Avelino Zorzo (PUCRS)
Carlos Raniery Paula dos Santos (UFSM)
Cristiano Bonato Both (Unisinos)
Guilherme Rodrigues (IFSUL Charqueadas)
Jéferson Campos Nobre (UFRGS)
Juliano Wickboldt (UFRGS)
Marcelo Caggiani Luizelli (Unipampa)
Marcelo da Silva Conterato (PUCRS)
Rafael Pereira Esteves (IFRS)
Rodrigo Mansilha (Unipampa)
Tiago Ferreto (PUCRS)
Vinícius Guimarães (IFSUL Charquadas)

Comite Consultivo

Fabiola Gonçalves Pereira Greve (UFBA)

Paulo André da Silva Gonçalves (UFPE)
Fábio Luciano Verdi (UFSCAR)
Jó Ueyama (USP)
Antônio Jorge Gomes Abelém (UFPA)
Eduardo Cerqueira (UFPA)
Luiz Fernando Bittencourt (Unicamp)
Rossana Andrade (UFC)
Michele Nogueira (UFPR)
Edmundo Madeira (UNICAMP)

Mensagem dos Coordenadores Gerais

É com grande alegria e orgulho que, após 9 anos, estamos trazendo o Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC) novamente para a cidade de Gramado. O SBRC 2019 acontece em um ano que é especial por várias razões: primeiro, o aniversário de 50 anos da ARPANET, a primeira rede a implementar o conjunto de protocolos TCP/IP, que acabaram se estabelecendo mais tarde como os fundamentos da Internet; os 30 anos da invenção da World Wide Web pelo cientista Inglês Tim Berners-Lee; a comemoração dos 30 anos do .br (country code top-level domain), o qual tem atualmente mais de 4 milhões de domínios registrados; por fim, nesse ano comemora-se os 30 anos da Rede Nacional de Ensino e Pesquisa (RNP), que iniciou em 1989 como um projeto de pesquisa. É, portanto, um imenso privilégio termos a oportunidade de sediar o SBRC em um ano que carrega tanto significado para a área de redes de computadores e sistemas distribuídos.

A 37ª edição do evento se apoia no histórico de sucesso do SBRC, que tradicionalmente inclui sessões técnicas, minicursos, painéis e debates, workshops, salão de ferramentas e palestras. Além de contar novamente com um Hackathon e com um Concurso de Teses e Dissertações, teremos muitas novidades no SBRC 2019, incluindo as reuniões de mentoria, o evento MUSAS e o workshop de estudantes latino-americanos. As reuniões de mentoria permitem oportunizar conversas 1-a-1 entre estudantes de pós-graduação e pesquisadores de excelência do Brasil e do exterior. O MUSAS (MULheres em redeS de computadores e sistemAs diStribuídos) visa fomentar conexões entre mulheres atuando nas áreas de redes de computadores e de sistemas distribuídos, no Brasil e no exterior. Já o workshop de estudantes latino-americanos (LANCOMM) serve como um ponto de encontro para estudantes da região apresentarem e discutirem o andamento de suas pesquisas, e obterem feedback construtivo de pesquisadores estrangeiros de grande prestígio.

Neste ano, a trilha principal do SBRC recebeu 228 submissões completas e que entraram no processo de avaliação. Todos os artigos receberam pelo menos 3 revisões, e após um rigoroso processo seletivo, 80 artigos foram aceitos para publicação e organizados em 23 sessões técnicas. O Salão de Ferramentas selecionou 12 trabalhos que a serem demonstrados ao longo do SBRC 2019. Por sua vez, o Concurso de Teses e Dissertações selecionou 8 dissertações de mestrado e 8 teses de doutorado para apresentação durante o evento. Além disso, esse ano o evento conta com 4 palestrantes internacionais, 3 painéis, 1 tutorial, 5 minicursos e 11 workshops.

A organização de um evento com o porte do SBRC é um processo longo, que demanda muita energia e empenho. O evento desse ano só foi possível graças ao apoio e suporte incondicional de muitos grupos de pessoas e instituições. Agradecemos em particular o apoio da SBC, do LARC, do Comitê Consultivo do SBRC e da Comissão Especial de Redes de Computadores e Sistemas Distribuídos da SBC. O evento contou com o apoio do Comitê Gestor da Internet no Brasil (CGI.br) e do Núcleo de Informação e Coordenação do Ponto BR (NIC.br), da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), da Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul (FAPERGS) e da ACM SIGCOMM. Contamos novamente também com

o apoio institucional da Rede Nacional de Ensino e Pesquisa (RNP). Além disso, o SBRC 2019 recebeu o apoio de importantes patrocinadores, incluindo SAP, HUAWEI, DATACOM, Google, Bedu.tech, BRDigital e AdylNet. Também agradecemos à Universidade Federal do Rio Grande do Sul (UFRGS), e em especial ao Instituto de Informática (INF), pelo suporte incondicional.

Agradecemos a todas as pessoas que trabalharam diretamente na organização do evento. Obrigado pelo excelente trabalho de todos os membros do comitê de organização: Antônio Jorge Gomes Abelém/UFPA e Fabíola Gonçalves Pereira Greve/UFBA (Coordenadores do Comitê de Programa), Stênio Fernandes/UFPE (Coordenador de Workshops), Ítalo Cunha/UFMG (Coordenador de Palestras, Tutoriais e Mentoria), Artur Ziviani/LNCC (Coordenador de Painéis), Miguel Elias Mitre Campista/UFRJ (Coordenador de Minicursos), Leandro Villas/UNICAMP (Coordenador do Salão de Ferramentas), Daniel Fernandes Macedo/UFMG (Coordenador do Concurso de Teses e Dissertações), e Raquel Lopes/UFMG e Luis Carlos De Bona/UFPR (Coordenadores do Hackathon). Agradecemos também aos membros do nosso comitê local, que nos ajudaram na operacionalização de diversas tarefas relacionadas à organização: Avelino Zorzo/PUCRS, Carlos Raniery Paula dos Santos/UFMS, Cristiano Bonato Both/Unisinos, Guilherme Rodrigues/IFSUL Charqueadas, Jéferson Campos Nobre/UFRGS, Juliano Wickboldt/UFRGS, Marcelo Caggiani Luizelli/Unipampa, Marcelo da Silva Conterato/PUCRS, Rafael Esteves/IFRS, Rodrigo Mansilha/Unipampa, Tiago Ferreto/PUCRS, e Vinícius Guimarães/IFSUL Charqueadas. Agradecemos ao Luis Otávio Luz Soares, técnico administrativo do INF/UFRGS, pela imensa ajuda nos mais variados aspectos relacionados à organização do evento, assim como pelo apoio de Leandro Disconzi Vieira e Carlos Alberto da Silveira Junior, também técnicos administrativos da UFRGS. Por fim, agradecemos à nossa equipe local, formada por alunos de graduação, pós-graduação e pós-doutorando da UFRGS, pela dedicação e pronta ajuda sempre que precisamos: Arthur Selle Jacobs, Augusto Zanella Bardini, Bruno Dalmaz, Fernanda da Silva Bonetti, Guilherme Bueno de Oliveira, Guilherme Rotth Zibetti, Isadora Pedrini Possebon, Leonardo Lauryel, Libardo Andrey Quintero González, Lucas Bondan, Lucas Castanheira, Luciano Zembruzki, Mateus Saquetti, Jonatas Marques, Rafael Hengen Ribeiro, e Ricardo Parizotto.

Desejamos a todos um ótimo evento e uma semana produtiva e com muitas trocas de ideias em Gramado.

Alberto Egon Schaeffer Filho e Weverton Luis da Costa Cordeiro
Coordenadores Gerais do SBRC 2019

Mensagem do Coordenador de Workshops

Os Workshops do SBRC são uma importante oportunidade para aprofundamento no conhecimento de temas especializados ou emergentes na nossa comunidade científica. Como em anos anteriores, a coordenação manteve a *Chamada de Propostas de Workshops do SBRC*, que tem estimulado a comunidade brasileira de Redes e Sistemas Distribuídos a discutir a viabilidade de workshops com temas de pesquisa mais tradicionais, bem como estimular o debate para adoção de eventos que tratam de temas mais emergentes.

Tivemos uma diversidade salutar no Comitê de Avaliação em termos de localização geográfica e experiência, gerando, portanto, opiniões complementares sobre as propostas submetidas. Consideramos todas as propostas de altíssima qualidade, com temas, focos e escopo diversos. Nesta edição de 2019, buscamos equilibrar os benefícios trazidos aos participantes com a capacidade de alocação de espaços no evento. Desta forma, foram selecionados 10 Workshops de alta qualidade, além do tradicional Workshop da RNP (*WRNP*). Dentre as propostas aceitas, oito são reedições de workshops tradicionais do SBRC, a saber: **Gerência e Operação de Redes e Serviços** (*WGRS*), **Testes e Tolerância a Falhas** (*WTF*), **Pesquisa Experimental da Internet do Futuro** (*WPEIF*), **Segurança Cibernética em Dispositivos Conectados** (*WSCDC*), **Trabalhos de Iniciação Científica e Graduação** (*WTICG*), **Blockchain: Teoria, Tecnologias e Aplicações** (*WBlockchain*), **Clouds e Aplicações** (*WCGA*), e **Computação Urbana** (*CoUrb*). Como novidade, teremos dois novos workshops, a saber: o Workshop de **Teoria, Tecnologias e Aplicações de Slicing para Infraestruturas Softwarizadas** (*WSlice*) e o **Latin American Student Workshop on Data Communication Networks** (*LANCOMM Student Workshop*), que tem como público-alvo principal os alunos de pós-graduação e pesquisadores da América Latina, além de estreitar os relacionamentos da nossa comunidade com a Association for Computing Machinery (*ACM*).

Como coordenador dos Workshops do SBRC 2019, gostaria de agradecer a todos os envolvidos na seleção das propostas. Primeiramente aos coordenadores gerais do SBRC 2019, Weverton Cordeiro (UFRGS) e Alberto Egon Schaeffer Filho (UFRGS), pelo convite para a coordenação desta chamada de trabalhos, além de todo o apoio necessário para sua condução apropriada. Agradeço também a todos os membros do comitê de avaliação pelo imenso esforço nas revisões e discussões de alta qualidade de todas as propostas submetidas. Por fim, agradeço aos coordenadores dos workshops aceitos, pela dedicação no cumprimento dos prazos e na condução dos trabalhos internos de seleção dos artigos para seus respectivos eventos, mantendo a alta qualidade geral do SBRC. Os inscritos nos diversos workshops esperam ansiosamente pelas apresentações dos trabalhos e pelas frutíferas discussões que serão geradas.

Stênio Fernandes
Coordenador dos Workshops do SBRC 2019

Comitê de Programa

- Alex Vieira (Universidade Federal de Juiz de Fora)
- Alexandre Ferreira (Universidade Estadual de Campinas)
- Alexandre Graeml (Universidade Tecnológica Federal do Paraná)
- Alfredo Goldman (IME - USP)
- Aline Viana (Inria Saclay - Ile de France)
- Anelise Munaretto (UTFPR)
- Antonio Alfredo Ferreira Loureiro (UFMG)
- Artur Ziviani (LNCC)
- Carlos Mauricio Figueiredo (Universidade do Estado do Amazonas)
- Cristiano Silva (Universidade Federal de São João del Rei)
- Daniel Batista (IME - USP)
- Daniel Guidoni (Universidade Federal de São João del-Rei)
- Denis Rosário (Federal University of Pará)
- Edmundo Madeira (UNICAMP)
- Eduardo Cerqueira (Universidade Federal do Pará - UFPA)
- Fabio Verdi (Universidade Federal de São Carlos)
- Felipe Domingos da Cunha (PUC Minas)
- Fernanda Sumika Souza (UFSJ)
- Genaina Rodrigues (Universidade de Brasília)
- Guilherme Maia (Universidade Federal de Minas Gerais)
- Heitor Ramos (Universidade Federal de Minas Gerais)
- Islene Garcia (UNICAMP)
- Jo Ueyama (USP)
- Jonice Oliveira (Universidade Federal do Rio de Janeiro (UFRJ))
- Jussara Almeida (DCC-UFMG)
- Keiko Fonseca (UTFPR)
- Leandro Balico (Universidade Federal de Roraima)
- Leandro Villas (UNICAMP)
- Leonardo Botega (State University of São Paulo (UNESP))
- Leyza Dorini (Universidade Tecnológica Federal do Paraná)
- Linnyer Ruiz (UEM)
- Luanna Lopes Lobato (Universidade Federal de Goiás)
- Luis Carlos De Bona (University Federal of Parana)
- Luiz Fernando Bittencourt (UNICAMP)
- Markus Endler (PUC-Rio)
- Maycon Peixoto (Universidade Federal da Bahia)
- Michele Nogueira (Universidade Federal do Paraná)
- Pedro Olmo Vaz de Melo (Universidade Federal de Minas Gerais)
- Rafael Lopes Gomes (Universidade Estadual do Ceará (UECE))
- Robson De Grande (uOttawa)
- Rodolfo Meneguette (Instituto Federal de Educação, Ciência e Tecnologia de São Paulo)
- Ronaldo Ferreira (Universidade Federal de Mato Grosso do Sul)
- Sidney Lucena (Universidade Federal do Estado do Rio de Janeiro - UNIRIO)

- Silvana Rossetto (UFRJ)
- Thiago Genez (University of Cambridge)
- Thiago Henrique Silva (Universidade Tecnológica Federal do Paraná)
- Valério Rosset (Universidade Federal de São Paulo)
- Vinicius Mota (Universidade Federal do Espírito Santo)

Sumário

Sessão Técnica 1 - Mobilidade Urbana e Redes Veiculares	1
Uma Abordagem V2X para Disseminação de Dados em Redes Veiculares	2
Luiz Henrique Silva (Pontifícia Universidade Católica de Minas Gerais), Raquel Mini (Pontifícia Universidade Católica de Minas Gerais), Felipe Domingos da Cunha (Pontifícia Universidade Católica de Minas Gerais)	
Um Novo Serviço de Gerenciamento de Tráfego para ITS baseado em Computação em Névoa	16
Celso Brennand (Universidade Estadual de Campinas), Geraldo Pereira (Universidade de Brasília)	
Realocação em Sistemas de Compartilhamento de Veículos em Uma Via 	30
Felipe Rooke da Silva (Universidade Federal de Juiz de Fora), Victor Aquiles Soares de Barros Alencar (Universidade Federal de Juiz de Fora), Alex Vieira (Universidade Federal de Juiz de Fora), Marcelo Machado (Universidade Federal do Estado do Rio de Janeiro)	
Speed: Uma solução inter-veicular para detectar e controlar vias congestionadas no TMS	44
Geraldo Pereira (Universidade de Brasília), Rodolfo Meneguette (Instituto Federal de Educação, Ciência e Tecnologia de São Paulo), Gustavo Pessin (Universidade de São Paulo), Jó Ueyama (Universidade de São Paulo), Leandro Villas (Universidade Estadual de Campinas)	
Sessão Técnica 2 - Modelagem e Mineração de Dados Urbanos	58
Analyzing Patterns of a Bicycle Sharing System for Generating Rental Flow Predictive Models	59
Johnattan Douglas Ferreira Viana (Universidade Federal Rural do Semi- Árido), Oton Braga (Universidade Federal Rural do Semi-Árido), Lenardo Silva (Universidade Federal Rural do Semi-Árido), Francisco Mendes Neto (Universidade Federal Rural do Semi-Árido)	
Avaliação da Qualidade do Transporte Público no Acesso a Unidades de Saúde de Curitiba	73
Clauciane Lima (Universidade Tecnológica Federal do Paraná), Ricardo Lüders (Universidade Tecnológica Federal do Paraná), Luiz Celso Gomes Jr (Universidade Tecnológica Federal do Paraná), Keiko Fonseca (Universidade Tecnológica Federal do Paraná), Altiéris Peixoto (Universidade Tecnológica Federal do Paraná)	

A Framework for Scalable Data Analysis and Model Aggregation for Public Bus Systems	85
Mayuri Morais (Universidade Federal do ABC), Raphael Camargo (Universidade Federal do ABC)	
Integração de Dados Públicos de Saúde e Transporte: Caracterização para Modelagem Multicamadas	99
Priscila Santin (Universidade Tecnológica Federal do Paraná), Anelise Munaretto (Universidade Tecnológica Federal do Paraná), Mauro Fonseca (Universidade Tecnológica Federal do Paraná), Thiago Henrique Silva (Universidade Tecnológica Federal do Paraná)	
Uma Infraestrutura para o Monitoramento e Predição de Rotas e Paradas de Ônibus no Transporte Universitário	113
Paulo Miranda e Silva Sousa (Universidade Federal do Ceará), Robertty Freitas (Universidade Federal do Ceará), Abilio e Silva (Universidade Federal do Ceará), Wilton Neto (Universidade Federal do Ceará), Carla Ilane Moreira Bezerra (Universidade Federal do Ceará), Emanuel Coutinho (Universidade Federal do Ceará)	
Development of a Semantic Representation Model of Criminal Information to Support the Assessment of Risk Situations	127
Jordan Saran (Paulista State University), Leonardo Botega (Centro Universitário Eurípides de Marília)	
Sessão Técnica 3 - Estratégias e Aplicações para o Desenvolvimento Urbano	141
Busca de Serviços Baseada em Perfis Sociais dos Objetos em uma Rede SIoT Urbana	142
Iury Rogerio Sales de Araujo (Universidade Federal da Paraíba), Mikaelly Pedrosa (Universidade Federal da Paraíba), Eudisley Gomes dos Anjos (Universidade Federal da Paraíba), Fernando Matos (Universidade Federal da Paraíba)	
Uma Análise dos Protocolos de Comunicação para Internet das Coisas	156
Arthur Cosmi (Universidade Federal do Espírito Santo), Vinicius Mota (Universidade Federal do Espírito Santo)	
Caracterização de Desempenho de uma Rede LoRa em Ambientes Urbanos: Simulação vs. Prática	170
Fernando Molano Ortiz (Universidade Federal do Rio de Janeiro), Thales Almeida (Universidade Federal do Rio de Janeiro), Ana Elisa Ferreira (Universidade Federal do Rio de Janeiro), Luis Henrique Costa (Universidade Federal do Rio de Janeiro)	

Detecção de eventos no Twitter através de Grafos de visibilidade natural	184
Fernanda Tenorio (Universidade Federal de Alagoas), Eduarda Chagas (Universidade Federal de Alagoas), Pedro Barros (Universidade Federal de Alagoas), Heitor Ramos (Universidade Federal de Alagoas)	
Um Sistema e-Health Inteligente para Análise e Detecção de Doenças Cardíacas	197
Diego Moreira (Universidade Estadual do Ceará), Levy Chaves (Universidade Estadual do Ceará), Rafael Lopes Gomes (Universidade Estadual do Ceará), Joaquim Celestino Júnior (Universidade Estadual do Ceará)	
Energesis: Fog Smart Meter para Hospedagem Compartilhada	211
Rafael Pereira (Universidade Federal da Bahia), Maycon Peixoto (Universidade Federal da Bahia), José Jesus (Universidade Federal de Pernambuco), Matheus Thiago Barbosa (Universidade Federal da Bahia), Eric Bernardes Chagas Barros (Universidade Federal da Bahia)	

**III Workshop de Computação Urbana
SBRC 2019
Sessão Técnica 1
Mobilidade Urbana e Redes Veiculares**

Uma Abordagem V2X para Disseminação de Dados em Redes Veiculares

Luiz Henrique C. Silva¹, Raquel A. F. Mini¹, Felipe D. Cunha¹

¹Departamento de Ciência da Computação
Pontifícia Universidade Católica de Minas Gerais (PUC-Minas)

lhcsilva@sga.pucminas.br, {raquelmini, felipe} @pucminas.br

Abstract. *Vehicular ad-hoc networks are a promising type of networks that allow communication between vehicles with the aim to promote safe and efficient traffic. During the trips, vehicles can communicate with each other and with other networks by interacting with them and with the cell phone network. The literature does not present many solutions that involve this existing infrastructure in the cities, and in order to disseminate data to a group of vehicles, it is interesting to know better the global behavior of the scenario. So, in this work, we proposed a protocol to disseminate data that select the vehicle to relay the information according to its position in relation to the destination of the message. Simulation results evaluated in scenarios of various densities have shown that it is possible to reduce the number of messages transmitted with a good delivery ratio.*

Resumo. *Redes Veiculares são um tipo promissor de rede ad-hoc que permite a comunicação entre veículos, com o objetivo de promover um tráfego seguro e mais eficiente. Durante suas viagens, os veículos podem se comunicar uns com os outros e com outras redes, por meio das interações entre eles e a rede celular. A literatura não apresenta muitas soluções que envolvam esta infraestrutura existente nas cidades, e para efetuar a disseminação de dados para um grupo de veículos é interessante conhecer melhor o comportamento global do cenário. Assim, neste trabalho é proposto um protocolo para disseminar os dados, que seleciona veículos para retransmitir a informação de acordo com suas posições em relação ao destino da mensagem. Resultados de simulação avaliados em cenários de diferentes densidades mostraram que é possível reduzir o número de mensagens transmitidas mantendo uma boa taxa de entrega.*

1. Introdução

A crescente demanda por comunicação no século XXI tem impulsionado os estudos de redes sem fio para aplicações que melhoram a qualidade de vida. Na última década, a comunicação em redes veiculares tem despertado interesse da academia, indústria e órgãos governamentais [Peng et al. 2018]. Este interesse ocorre devido ao grande potencial que as redes veiculares têm de aumentar a segurança e desempenho no trânsito. Inseridas no campo das redes móveis, as redes veiculares são formadas por veículos equipados com dispositivos de comunicação sem fio, processamento e sensoriamento.

Embora as redes veiculares apresentem novas oportunidades para o provimento de serviços inovadores, também apresentam novos desafios. A alta mobilidade dos nós

em redes veiculares caracteriza a dinamicidade da topologia dessas redes, trazendo consigo a necessidade do fornecimento de informação em tempo real para os envolvidos na comunicação. Duas áreas de aplicações que se destacam no contexto das redes veiculares são as de segurança no trânsito e de melhoria no desempenho do trânsito.

Um exemplo de aplicação para segurança no trânsito por meio das redes veiculares é a notificação de eventos de segurança. Estes eventos podem ser naturais (oriundos de mudanças climáticas em virtude das estações do ano e das características do ambiente) ou ainda não-naturais (provenientes, por exemplo, da negligência dos condutores de veículos). Aplicações de notificação como esta exigem da rede a atividade de disseminação de dados que consiste no envio da mensagem por um veículo origem e a entrega desta mensagem a todos os veículos na região de interesse (*Region of Interest - RoI*).

A atividade de disseminação de dados também pode ser utilizada para o aumento do desempenho no trânsito. Uma das principais causas dos congestionamentos é a restrição espacial das vias, assim sendo, uma boa gestão do recurso espaço pode ser feita por meio da comunicação entre os veículos informando os condutores para impedir que situações mais críticas como acidentes ocorram ou, se ocorrerem, pode reduzir os prejuízos para o desempenho do trânsito. Neste ponto, a disseminação de dados tem um papel fundamental, o de levar a informação ao maior número de condutores no menor espaço de tempo, permitindo que os condutores tomem decisões inteligentes evitando acidentes e congestionamentos. Independente da aplicação, a atividade de disseminação de dados é fortemente influenciada pela densidade de veículos no cenário, e dois cenários opostos são bastante comuns nas grandes cidades.

Num primeiro cenário, de baixas densidades de veículos, os nós apresentam-se provavelmente mais distantes e, além de se encontrarem menos, o que proporciona um número menor de conexões, têm liberdade para alcançarem velocidades mais altas, provocando conexões por tempos mais curtos que a média (desconexões mais constantes). A ausência de conexões ou ainda as conexões por curtos espaços de tempo são características que identificam o problema de partições na rede.

Num segundo cenário, de altas densidades de veículos, os nós aglomeram-se mais facilmente, além disso, devido às restrições espaciais, movem-se mais lentamente, permitindo conexões mais longas entre si, apesar de naturalmente curtas. Este cenário é facilmente percebido em horários de pico, durante o início da manhã e o fim da tarde de dias úteis, caracterizado pelos constantes congestionamentos.

Este cenário de altas densidades apresenta ainda complicações relacionadas aos recursos de rede. Devido à comunicação em redes veiculares ser via *broadcast*, o número de transmissões feitas numa disseminação de dados pode aumentar conforme a densidade de veículos. Por exemplo, a decisão sobre retransmitir deve filtrar bem os veículos responsáveis pela tarefa para impedir o crescimento do número de mensagens duplicadas, reduzindo o consumo do recurso de rede e impedindo que ocorram colisões e perdas de dados. Este problema é conhecido na literatura como *broadcast storm*.

Nestes dois cenários comuns encontrados nas grandes cidades, as redes veiculares tem um papel muito promissor na melhoria da segurança no trânsito. O trabalho de [Toor et al. 2008] mostrou que aplicações de segurança em redes veiculares podem reduzir acidentes em cerca de 60% se capazes de informar os condutores acerca de eventos

relacionados com cerca de 0,5 segundo de antecedência. Em 2013, a Organização Mundial de Saúde publicou um relatório [W.H.O. 2013] que apresenta o atual estado global da segurança no trânsito. De acordo com o relatório, cerca de 1.24 milhão de pessoas morrem em acidentes de trânsito todos os anos. Garantir a entrega de notificações de segurança em tempo hábil apresenta potencial para uma grande redução nestes números. A atividade que dá suporte a esta entrega é a disseminação de dados.

Muitos dos estudos em redes veiculares voltados para disseminação de mensagens de segurança têm concentrado esforços na comunicação V2V primariamente. Embora o cenário ideal, do ponto de vista financeiro, para implantação de redes veiculares não envolva aquisição de infraestrutura, a utilização da infraestrutura existente pode apresentar grandes avanços sem custo adicional, integrando redes celulares e de computadores convencionais, passos necessários para o futuro da Internet das Coisas.

Diante da necessidade de segurança no trânsito e das emergentes redes celulares de 5ª Geração, este trabalho propõe o *TBD – Trajectory Based Dissemination*, um novo protocolo de disseminação de dados em cenários de densidades variadas. O objetivo do TBD é utilizar a rede celular como auxílio para o conhecimento da rede, permitindo que se defina uma trajetória para guiar a disseminação da mensagem da origem até a região de interesse. Através de resultados de simulação em diferentes configurações de densidades foi possível verificar uma redução no número de mensagens transmitidas, mantendo uma boa taxa de entrega.

O restante deste artigo está organizado da seguinte forma: a Seção 2 apresenta trabalhos que tratam os problemas da disseminação de dados encontrados na literatura. A solução proposta neste trabalho é detalhada na Seção 3. E na Seção 4 é descrito o processo de avaliação de desempenho, bem como a discussão dos resultados. Por fim, a Seção 5 apresenta as conclusões deste estudo e dá direções para trabalhos futuros.

2. Trabalhos Relacionados

A forma mais simples conhecida para disseminação de dados em redes veiculares é através do algoritmo de *flooding*, como mostrado por [Tseng et al. 2002]. Neste algoritmo, uma fonte S inicia a transmissão de uma mensagem m a todos os veículos no seu raio de alcance. Cada veículo alcançado por S efetua a retransmissão de m também a todos que estão em seu raio de alcance. Estas retransmissões ocorrem sucessivamente até que a rede seja inundada com os pacotes. Apesar de simples, essa abordagem é genérica, desconsidera o tipo de mensagem e sofre com o problema de *broadcast storm*.

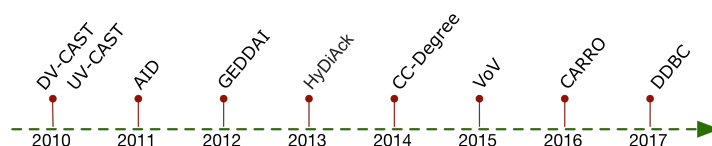


Figura 1. Protocolos de disseminação de dados com propósito semelhante apresentado nos últimos anos.

Outra solução proposta para VANETs é o protocolo AID (*Adaptive approach for Information Dissemination*) [Bakhouya et al. 2011]. Este é um protocolo descentralizado e adaptativo para a disseminação da informação. Nele, a decisão do veículo de encaminhar

ou não o dado é feita baseada em probabilidade. Esta probabilidade é configurada baseada no número de vezes que o veículo recebe a mesma mensagem em um dado período de tempo. Desta forma, em cenários com alta densidade de tráfego, muitos veículos podem decidir a não enviar a informação baseado no fato de outros já terem a enviado. Entretanto, esta não é uma solução que lida com o problema das conexões intermitentes na rede.

Em [Akabane et al. 2016], os autores propuseram um protocolo que atua tanto em cenários urbanos como em rodovias. Além disso, o protocolo proposto CARRO (*Context-Aware Routing pROtocol*) lida com os dois problemas clássicos de disseminação de dados, *broadcast storm* e partições na rede. A forma como abordaram a disseminação de dados envolve um mecanismo de difusão de *beacons*. *Beacons* são pacotes de dados de tamanho reduzidos que armazenam informações de contexto e são transmitidos a cada intervalo de tempo definido. As informações de contexto podem ser, por exemplo, a localização estimada por meio de sistemas de posicionamento global (*Global Positioning System - GPS*), número de vizinhos, velocidade do veículo, etc. Estas informações permitem que os veículos façam estimativas acerca da topologia da vizinhança a um determinado número de saltos.

O processo de disseminação de dados ocorre a partir da observação da densidade de veículos no cenário, estimada por meio das informações contidas nos *beacons*. Em cenários densos de veículos o protocolo CARRO trata o problema de *broadcast storm* utilizando um mecanismo de supressão de *broadcast* chamado Zona de Preferência. Zonas de Preferência são setores na borda do raio de comunicação em que veículos localizados em seu interior são considerados melhores potenciais retransmissores de tal forma que bastando a retransmissão desses veículos a disseminação será eficiente. Por outro lado, em cenários pouco densos de veículos, o protocolo proposto pelos autores utiliza o mecanismo de armazena-e-encaminha para dar continuidade ao processo de disseminação.

Considerando as soluções que consideram ambos os cenários, o protocolo UV-CAST [Viriyasitavat et al. 2010] é proposto considerando as diferentes condições de tráfego. Nele o veículo trabalha em dois estados: supressão de *broadcast* e armazena-e-encaminha. Quando o veículo recebe uma mensagem de dados pela primeira vez, ele inicialmente verifica se é ou não um veículo de borda. Veículos de borda são aqueles que conectam dois ou mais componentes na rede. Assim, o UV-CAST assume que esses veículos têm grande probabilidade de encontrar novos veículos vizinhos. Neste caso, o veículo armazenará a mensagem e a encaminhará a cada novo encontro. Caso contrário, não sendo veículo de borda, ele executa os algoritmos de supressão de *broadcast* para encaminhamento da mensagem.

A Figura 1 apresenta alguns trabalhos desenvolvidos ao longo dos últimos anos acerca da disseminação de dados encontrados na literatura e detalhes acerca das propostas são resumidos na Tabela 1. A grande maioria dos protocolos da literatura propõe soluções com a utilização da comunicação V2V apenas, poucos utilizando recursos infraestruturados para a comunicação. Além disso, ao efetuarem avaliações os protocolos consideram que todo o mapa é a região de interesse ou que o veículo origem da mensagem está no interior da região de interesse.

De forma diferente, neste trabalho a disseminação de dados é efetuada de forma a transmitir a mensagem até a região de interesse, e a partir daí disseminá-la para todos os

Protocolo	Abordagem	Beacons	Problemas		Cenário	
			Broadcast storm	Partições na rede	Urbano	Rodovia
DV-CAST [Tonguz et al. 2010]	estatística e contexto	1-salto	✓	✓	-	✓
UV-CAST [Viriyasitavat et al. 2010]	estatística e contexto	1-salto	✓	✓	✓	-
AID [Bakhouya et al. 2011]	estatística	-	✓	-	✓	-
GEDDAI [Villas et al. 2012]	geográfica	-	✓	-	✓	-
HYDIACK [Maia et al. 2013]	geográfica	1-salto	✓	✓	✓	-
CC-DEGREE [Cunha et al. 2014]	métricas sociais	1-salto	✓	✓	✓	-
VOV [Maia et al. 2015]	geográfica	1-salto	✓	✓	✓	-
CARRO [Akabane et al. 2016]	geográfica	1 salto	✓	✓	✓	✓
DDBC [Costa et al. 2017]	grafos	2-saltos	✓	-	✓	-

Tabela 1. Protocolos propostos na literatura nos últimos anos e suas características em termos abordagem, problemas tratados e cenários atendidos.

veículos no interior da região. Isso é feito a partir do conhecimento do comportamento global da cidade, obtido pela rede móvel.

3. TBD – Trajectory Based Dissemination

Nesta seção será descrito o protocolo *TBD – Trajectory Based Dissemination*, o qual considera a definição de uma trajetória para direcionar a disseminação de dados da origem até a região a ser alcançada. Para isso, o algoritmo tira proveito da comunicação com a rede celular para obter informações acerca da densidade da rede e assim, definir a trajetória da mensagem para a disseminação. Assim, vale ressaltar que:

- (i) Os veículos estão equipados com dispositivo de comunicação sem fio com mesmo alcance de comunicação
- (ii) Os veículos estão conectados à rede celular
- (iii) Os veículos estão equipados com dispositivo GPS
- (iv) A mensagem a ser transmitida leva a localização da região de interesse
- (v) O veículo origem da mensagem está fora da região de interesse

Como requisito fundamental da disseminação, a mensagem deve alcançar os veículos no interior da região de interesse. Neste contexto, o TBD trabalha pra efetuar esta tarefa minimizando o consumo de recurso da rede. Assim, através de uma interação com a rede celular, o protocolo obtém informação da densidade de veículos por região, de forma a definir a trajetória que a mensagem irá percorrer até alcançar a região de destino. Esse conhecimento só é possível pois os veículos encaminham periodicamente sua localização para a rede celular, com o intuito de que se tenha ciência de todo o comportamento global dos veículos. Essa abordagem evita a troca constante de *beacons* entre os veículos, o que contribui para reduzir o consumo de recursos da rede.

Como pode ser visto na Figura 2(a), a solução proposta divide a rede em células. Cada célula trata-se de um quadrado de lado equivalente ao tamanho do raio de comunicação. A partir desta divisão e dos envios de mensagens de localização de cada veículo, têm-se uma matriz de densidades. A periodicidade do envio destas mensagens pode ser definida de acordo com o tempo e também com a frequência de movimentação de cada veículo.

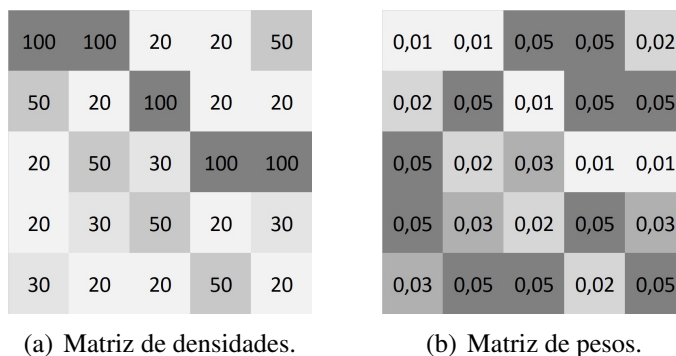


Figura 2. (a) Exemplo de matriz de densidades, conforme submissões de veículos nas células. (b) Densidades convertidas em pesos, entrada para o algoritmo de Dijkstra.

Conhecendo a matriz de densidades, os veículos podem encontrar o caminho mais denso de veículos da origem até o destino da mensagem. Na Figura 2(b), pode-se verificar o valor das densidades invertidos, de forma que os mesmos possam ser usados como entrada para o cálculo do problema de encontrar o caminho mais curto entre dois pontos. Assim, densidades maiores serão convertidas a pesos menores e a saída do algoritmo favorecerá essas regiões. Neste trabalho, o caminho mais curto é encontrado através do algoritmo de Dijkstra.

Definida a rota, a mensagem então será disseminada por meio de comunicação V2V da origem ao destino. Assim, todos os veículos pertencentes a essa rota encaminharão a mensagem. Entretanto, a fim de evitar transmissões desnecessárias e até mesmo colisões, optou-se por utilizar um mecanismo de transmissão orientado, definido por um cone (conforme Figura 3).

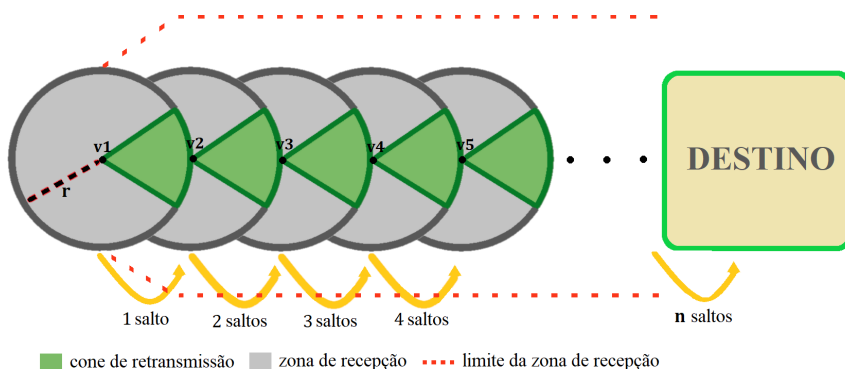


Figura 3. A retransmissão só é feita na área do cone, a cada salto até o destino da mensagem.

Na Figura 3, vI é o veículo fonte da mensagem, *Destino* é a região de interesse, destino da mensagem, e r é o raio de alcance do dispositivo de comunicação. Ao efetuar a transmissão da mensagem, vI alcança todos os veículos dentro do raio do dispositivo de comunicação, no entanto, define-se que apenas os veículos no interior do cone efetuarão a retransmissão das mensagens. De igual forma, os veículos que receberem a mensagem de vI , conhecendo o destino da mensagem podem estimar a posição do cone e efetuarem retransmissões direcionadas. O ângulo de abertura do cone pode ser ajustado conforme a densidade de veículos, permitindo uma maior abrangência para densidades menores e uma menor abrangência para densidades maiores, o que contribui para mitigar o problema de *broadcast storm*. As retransmissões ocorrem sucessivamente, até que a mensagem alcance o destino. Conhecendo a matriz de densidades e utilizando o mecanismo de transmissão orientado ao cone de direção, o protocolo TBD define o mecanismo de retransmissão através do Algoritmo 1.

Algoritmo 1: Algoritmo de supressão de *broadcast*.

```

1 Evento: mensagem de dados  $m$  recebida do vizinho  $v$  sinalizada para retransmissão
2 adicionar  $m$  na lista de mensagens recebidas;
3  $t \leftarrow obterTempoDeRetransmissão()$ ;
4 se o envio de  $m$  está agendado então
5   | se agendada 1 vez então
6   |   | cancela e efetua reagendamento para tempo  $t$ ;
7   |   | fim
8   |   | senão
9   |   |   | cancela o envio da mensagem  $m$ ;
10  |   |   | fim
11  |   | fim
12  |   | senão
13  |   |   | efetua agendamento para tempo  $t$ ;
14  |   |   | fim
15 Evento: agendamento da mensagem  $m$  expira
16 Retransmite  $m$ ;
```

No Algoritmo 1, quando um veículo recebe uma mensagem de dados sinalizada para retransmissão (linha 1), ele deve adicionar a mensagem m na lista de mensagens recebidas (linha 2). Em seguida, calcula um intervalo de tempo para retransmissão (linha 3) de acordo com a distância em relação ao vizinho v . Enquanto maior a distância, menor o tempo definido para a retransmissão. Vale destacar que se a distância for igual a r , a retransmissão é feita imediatamente. Com o tempo calculado, verifica-se a existência de agendamento para retransmissão futura (linha 4). Se existir e for o primeiro, cancela e efetua um reagendamento para o tempo t calculado (linha 6). Caso não seja o primeiro agendamento, cancela o envio da mensagem m (linha 9). Ao verificar que não existe agendamento de retransmissão da mensagem m , efetua o primeiro agendamento para o tempo t (linha 13). Assim que o tempo t expira, a mensagem é então retransmitida (linhas 15 e 16). Através do mecanismo de cone de direção e deste mecanismo de supressão de *broadcast*, os veículos mais próximos do objetivo de levarem a mensagem ao destino na região de interesse terão prioridade na retransmissão da mensagem.

A partir dos mecanismos apresentados, o algoritmo implementado pelo protocolo TBD toma decisão sobre a retransmissão da mensagem conforme o Algoritmo 2.

Algoritmo 2: Algoritmo TBD – *Trajectory Based Dissemination*.

```

1 Evento: Decisão sobre retransmitir  $m$  recebida do vizinho  $v$ 
2 se veículo está na região de interesse então
3   | retransmite  $m$  com algoritmo de supressão;
4 fim
5 senão
6   | se veículo está no cone de  $v$  então
7     | retransmite  $m$  com algoritmo de supressão;
8     fim
9     senão
10    | descarta  $m$ ;
11    fim
12 fim

```

Conforme o Algoritmo 2, ao receber uma mensagem, o veículo verifica se encontra-se na região de interesse (linha 2), neste caso, retransmite a mensagem por meio do Algoritmo 1 (linha 3). Caso o veículo não se encontre na região de interesse ainda, existe a necessidade de seguir uma rota para que a mensagem chegue até a região de interesse. Isso é feito verificando se o veículo está no cone de retransmissão (linha 6) do veículo que o envio. Caso esteja, o veículo efetua a retransmissão também utilizando o Algoritmo 1 (linha 7). Caso o veículo não esteja na região de interesse e também não esteja no cone, a mensagem é descartada (linha 10).

4. Avaliação de Desempenho

A avaliação de desempenho da solução proposta foi efetuada por meio de simulações usando como simulador de rede a ferramenta OMNeT++ 4.6 [Varga 2003] e como simulador de mobilidade a ferramenta SUMO 0.25.0 [Behrisch et al. 2011], ambos por meio do *framework* de redes veiculares Veins [Sommer et al. 2011]. Os resultados foram comparados com protocolos da literatura AID, CARRO e UV-CAST, apresentados na Seção 2. Estes protocolos implementam a difusão de *beacons* para ciência de contexto local, de maneira semelhante ao protocolo TBD que efetua submissões à rede móvel para ciência de contexto global. A seguir serão descritos os detalhes das simulações e os resultados. Na Seção 4.1 são apresentados os detalhes dos cenários de simulação bem como os parâmetros utilizados. Na Seção 4.2 são descritas as métricas usadas na avaliação, e finalmente a Seção 4.3 apresenta as discussões e os resultados de simulação.

4.1. Cenário e Parâmetros de Simulação

Com o objetivo de avaliar o desempenho do protocolo proposto foi implementado um cenário que favorece a ocorrência do problema de *broadcast storm*. Conforme a Figura 4, este é um *grid* de Manhattan com 20 vias duplas uniformemente espaçadas em uma área de $2000\text{m} \times 1000\text{m}$.

Assim, foi assumido que cada quarteirão tem 100×100 metros. Para este cenário de simulação, assumiu-se que o dispositivo de comunicação tem alcance equivalente a 250m,

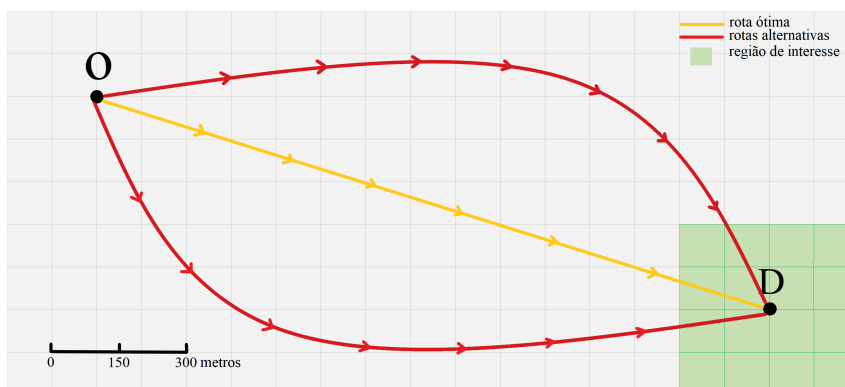


Figura 4. Cenário de Manhattan implementado, com veículo origem da mensagem numa extremidade e região de interesse na outra, o que leva aos protocolos a fazerem vários saltos até encontrarem o destino.

valor constante em 360° , para todos os veículos. O ângulo do cone de direção foi definido de maneira dinâmica, variando entre 110° e 130° , eliminando transmissões de veículos que se encontram no espaço dos 250° e 230° fora do cone, respectivamente. Com o objetivo de quantificar a evolução do tráfego neste cenário, a densidade de veículos foi variada de 100 veículos/ km^2 a 300 veículos/ km^2 . O tráfego neste cenário foi gerado pelo simulador de mobilidade SUMO, e na posição (200,200) do mapa foi posicionado um veículo fonte da mensagem, gerando 1 mensagem de 2048 bytes aos 15s de simulação. Na outra extremidade do mapa, a região de interesse foi definida como os últimos 400×400 metros. A Tabela 2 resume os parâmetros de simulação.

Parâmetro	Valor
Tamanho do área total	2000 \times 1000 metros
Tamanho da área da ROI	400 \times 400 metros
Alcance do raio de comunicação	250 metros
Ângulo do Cone dinâmico	110° até 130°
Origem da mensagem	(200,200)
Tamanho da mensagem	2048 bytes
Freq. mensagens de localização	1, 5, 9 s.
Densidades avaliadas	100 – 300 veículos/ km^2

Tabela 2. Resumo dos parâmetros de simulação.

4.2. Métricas

Abaixo as métricas que foram consideradas para avaliação das soluções são apresentadas. O objetivo principal é avaliar a cobertura do protocolo, o *overhead* gerado pelas mensagens de dados e o atraso em diferentes condições de tráfego.

- **Taxa de Entrega:** O percentual de mensagens de dados gerados pelo veículo fonte e entregues a todos os veículos no cenário. O quão mais próximo de 100% for este valor, maior também será a cobertura do mapa.
- **Taxa de Entrega na ROI:** Representa o percentual de veículos alcançados dentro da região de interesse. O desafio nesta avaliação é garantir uma cobertura total desta região.

- **Mensagens Duplicadas:** Esta métrica representa o total de mensagens repetidas recebidas pelos veículos durante o processo de disseminação.
- **Percentual de Veículos Transmissores:** Com esta métrica é possível avaliar o percentual de veículos envolvidos durante a disseminação. É importante destacar que uma boa solução deve garantir uma boa cobertura da ROI, evitando o *overhead*, i.e., mensagens transmitidas desnecessariamente. Assim, quanto menor for essa taxa, melhor será o desempenho da solução.
- **Atraso médio na entrega:** Esta métrica representa a média do tempo que uma mensagem de dados leva para ser transmitida da origem ao destino. Muitas aplicações em VANETs possuem restrições de atraso, por exemplo, serviços de alerta. Para estes tipos de aplicações, as mensagens devem ser disseminadas o mais rápido possível.
- **Número de saltos:** Esta é uma métrica que reflete o número de veículos envolvidos na entrega da mensagem, ou seja, o número de saltos entre o veículo origem e o veículo receptor da mensagem.

4.3. Resultados

Nesta Seção são apresentados os resultados das simulações feitas conforme parâmetros definidos na Seção 4.1. De forma a avaliar o impacto da atualização de posicionamento do protocolo TBD, os intervalos de envio de posição feitas junto à rede móvel foram definidos entre 1, 5 e 9 segundos (TBD_1, TBD_5 e TBD_9 respectivamente). Para cada uma das configurações, foram efetuados um total de 50 replicações. Os resultados apresentados a seguir representam a média dos valores obtidos com essas replicações.

Os gráficos da Figura 5 apresentam os resultados de simulações do ponto de vista do número de mensagens disseminadas. No gráfico da Figura 5-(a), temos a taxa de entrega das disseminações efetuadas, incluindo os veículos que não estão no interior da região de interesse. Pode-se observar que a porcentagem total de veículos alcançados pelo TBD é inferior aos demais protocolos avaliados. Isso se deve ao fato de a mensagem percorrer um caminho no mapa, levando a mensagem até o destino, transmitindo o mínimo aos veículos que não são parte do caminho e que não estão na região de interesse. Observa-se que os protocolos CARRO, AID e UV-CAST apresentam uma taxa de entrega bem maior.

Ao variar a densidade da rede, observa-se que os demais protocolos mantêm a proporção de veículos alcançados. No entanto, o TBD mantém a taxa de recepção da mensagem apenas no caminho da disseminação e na região de interesse, ou seja, o número de mensagens recebidas total para o protocolo TBD será sempre menor e mais perceptível enquanto maior for o cenário. Nota-se também que as três versões apresentam o comportamento semelhantes, com pouca variação quando a densidade de veículos é aumentada.

O gráfico da Figura 5-(b) apresenta a taxa de entrega no interior da região de interesse. Importante ressaltar que nos cenários avaliados, o tamanho da região de interesse representa aproximadamente 10% de toda a rede. Pode-se observar que os protocolos UV-CAST, AID e TBD com atualizações de densidade a cada 1 s. apresentam comportamento semelhante, entregando a mensagem em toda a região de interesse. Em razão do maior intervalo entre atualizações de densidade, o protocolo TBD com atualizações a cada 5 e a cada 9 segundos sofrem um pouco para entregarem na região interesse. Isso ocorre devido

a falta de precisão nas informações de localização, o que pode gerar atrasos e gargalos durante o encaminhamento da mensagem até a região.

O gráfico da Figura 5-(c) apresenta o número de mensagens duplicadas por veículos em todo o mapa. É possível observar que o protocolo TBD apresenta os melhores resultados, limitando o número de mensagens transmitidas a poucos veículos. De acordo com o seu funcionamento que define um cone que direciona a retransmissão das mensagens, a quantidade de veículos retransmissores é reduzida. Mesmo com informações menos precisas, o protocolo TBD com atualizações a cada 5 e 9 segundos reduzem o número de mensagens duplicadas a metade, comparando com os demais protocolos. Nota-se também que os demais protocolos, por não implementarem nenhum mecanismo de contenção na disseminação, apresentaram valores bem superiores, destacando CARRO e AID com as maiores taxas.

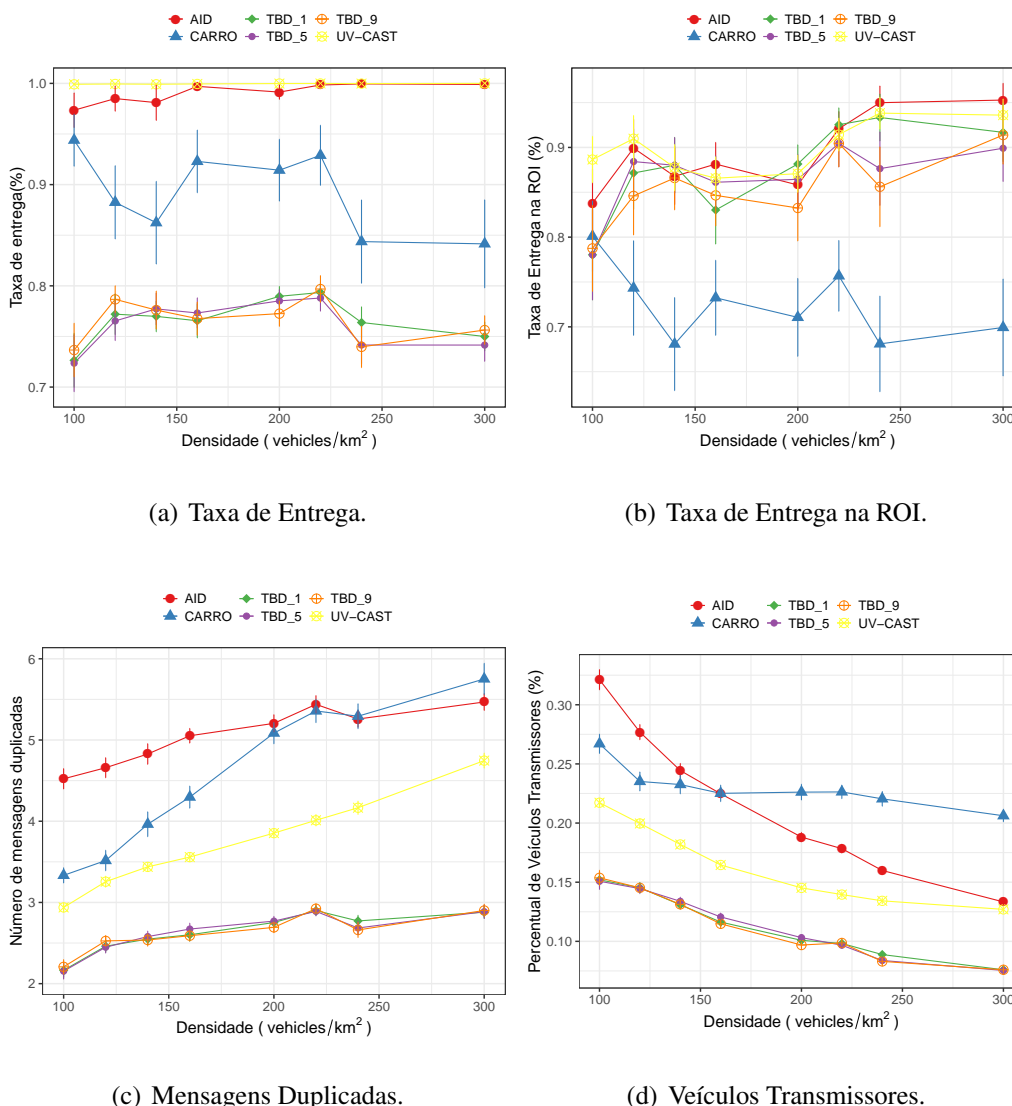
Ao observar a taxa de veículos transmissores no gráfico da Figura 5-(d), pode-se observar conformidade com os resultados já apresentados. A quantidade de veículos envolvidos na disseminação pelo protocolo TBD é inferior em todas as densidades, e reduz conforme a densidade de veículos aumenta. Ao comparar com as demais soluções avaliadas, pode-se verificar que cerca de 25% dos veículos estão envolvidos na disseminação dos dados, o que explica os resultados obtidos com as mensagens duplicadas.

Os gráficos da Figura 6 apresentam resultados das simulações do ponto de vista da eficiência na entrega e do número de saltos. Para todos os protocolos foi definido que o maior tempo de espera para uma retransmissão é 500ms. Esta escolha acompanha o trabalho de [Viriyasitavat et al. 2010], em que o autor mostra que o atraso deve ser cuidadosamente escolhido, e o valor 500ms pode ser definido como limite superior. Para o autor, o assunto deve ser mais estudado em trabalhos futuros.

Pode-se observar os resultados para o atraso médio na Figura 6-(a). Nota-se que o protocolo TBD apresenta resultados bastante satisfatórios comparado aos demais da literatura, igualando ao protocolo AID que também apresenta um atraso baixo. Tanto o protocolo CARRO e UV-CAST adicionam atraso nas retransmissões como mecanismo de contenção para o problema de *broadcast storm*. Por esse motivo esses protocolos apresentam valores superiores. Ao analisar o comportamento do TBD, pode-se notar um bom desempenho do protocolo em todas as suas três versões, independente da densidade de veículos avaliada.

Ao analisar o número médio de saltos efetuados na disseminação (Figura 6-(b)), pode-se notar que em média o TBD apresenta 2 saltos a menos que os demais. O atraso ser inferior está diretamente ligado à quantidade de saltos da origem ao destino da mensagem, assim, por apresentar um número de saltos inferior, o protocolo TBD apresenta melhores resultados em termos de atraso.

Além dos resultados apresentados sob as métricas definidas na Seção 4.2, percebeu-se que o intervalo de tempo entre os acessos à rede celular para submissão das novas posições pouco influenciou o desempenho do protocolo. Esta informação sugere que conforme o tempo passa, as densidades de veículos no mapa não sofrem mudanças significativas em relação aos segundos anteriores devido aos veículos estarem limitados pelas restrições espaciais. Essa informação contribui com a avaliação de custo financeiro dos acessos à rede móvel, sugerindo que são desnecessários acessos muito frequentes devido a



(a) Taxa de Entrega.

(b) Taxa de Entrega na ROI.

(c) Mensagens Duplicadas.

(d) Veículos Transmissores.

Figura 5. Resultados de simulação considerando o comportamento das mensagens enviadas e recebidas.

não agregarem informações capazes de refletir em desempenho.

5. Conclusão e Trabalhos Futuros

Neste trabalho foi proposto um protocolo de disseminação de dados que levasse a mensagem da origem até a região de interesse e na região de interesse entregasse a todos os veículos. Para este fim, o protocolo proposto utilizou a infraestrutura existente nas cidades, a rede celular, como forma de comunicação de controle. Cada veículo atualiza sua posição via rede celular periodicamente, permitindo que os demais veículos da cidade conheçam as densidades de veículos de cada região. Através desse conhecimento, o protocolo TBD faz com que a mensagem seja transmitida por um caminho de densidades, de forma que a mensagem a ser disseminada na região de interesse não seja entregue e retransmitida por veículos fora da região de interesse ou fora do caminho da disseminação. Os resultados

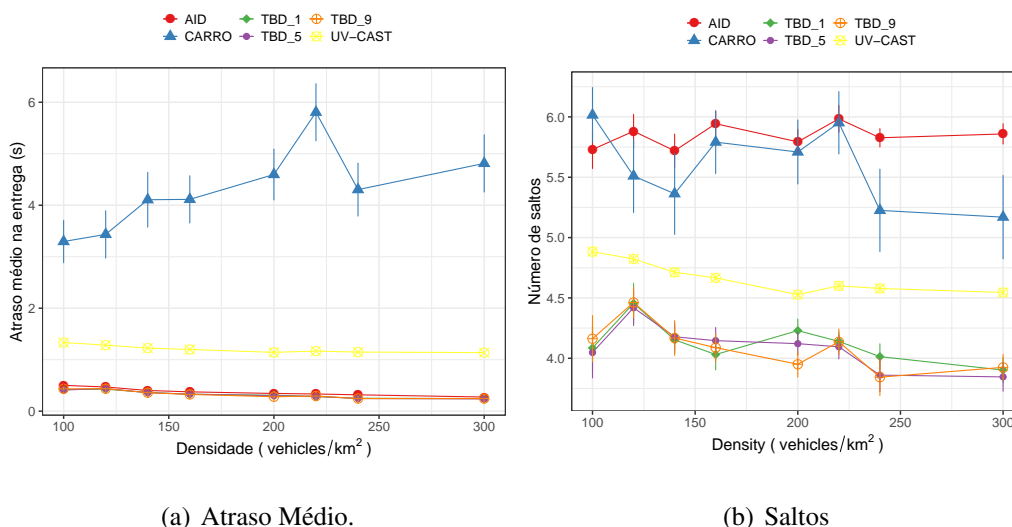


Figura 6. Comparação do número de saltos e atraso no recebimento das mensagens.

de simulações mostraram que o protocolo desempenha bem para o problema proposto, reduzindo de maneira expressiva o número de mensagens duplicadas, bem como o número de veículos não interessados na mensagem que a recebem. Foi verificado também que o percentual de veículos transmissores diminuiu, o que explica a redução no número de mensagens duplicadas, sem mudança significativa na entrega aos veículos no interior da região de interesse. Além disso, foi observado que o intervalo entre atualizações pouco interfere nos resultados de entrega e custo de transmissões.

Para trabalhos futuros, pretende-se utilizar do conhecimento dos veículos sobre seus destinos e velocidades para tornar as atualizações à rede celular dinâmicas, sensíveis às informações locais de cada veículo na rede. Essas informações permitirão a tomada de decisão mais inteligente acerca do intervalo de tempo entre as atualizações de posição, o que pode reduzir o número de acessos à rede. Além disso, pretende-se avaliar novas formas para definição da trajetória, reduzindo a quantidade de veículos transmissores sem comprometer a cobertura da disseminação na região de interesse.

Referências

- [Akabane et al. 2016] Akabane, A. T., Pazzi, R. W., Madeira, E. R., and Villas, L. A. (2016). Carro: A context-awareness protocol for data dissemination in urban and highway scenarios. In *Communications (LATINCOM), 2016 8th IEEE Latin-American Conference on*, pages 1–6. IEEE.
- [Bakhouya et al. 2011] Bakhouya, M., Gaber, J., and Lorenz, P. (2011). An adaptive approach for information dissemination in vehicular ad hoc networks. *Journal of Network and Computer Applications*, 34(6):1971–1978.
- [Behrisch et al. 2011] Behrisch, M., Bieker, L., Erdmann, J., and Krajzewicz, D. (2011). Sumo—simulation of urban mobility. In *The Third International Conference on Advances in System Simulation (SIMUL 2011), Barcelona, Spain*.

- [Costa et al. 2017] Costa, J., Lobato, W., de Souza, A. M., Rosário, D., Villas, L. A., and Cerqueira, E. (2017). Centrality-based data dissemination protocol for vehicular ad hoc networks. In *2017 IEEE 16th International Symposium on Network Computing and Applications (NCA)*, pages 1–4.
- [Cunha et al. 2014] Cunha, F. D., Maia, G. G., Viana, A. C., Mini, R. A., Villas, L. A., and Loureiro, A. A. (2014). Socially inspired data dissemination for vehicular ad hoc networks. In *Proceedings of the 17th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems*, pages 81–85. ACM.
- [Maia et al. 2013] Maia, G., Villas, L. A., Boukerche, A., Viana, A. C., Aquino, A. L. L., and Loureiro, A. A. F. (2013). Data dissemination in urban vehicular ad hoc networks with diverse traffic conditions. In *2013 IEEE Symposium on Computers and Communications (ISCC)*, pages 000459–000464.
- [Maia et al. 2015] Maia, G., Villas, L. A., Viana, A. C., Aquino, A. L., Boukerche, A., and Loureiro, A. A. (2015). A rate control video dissemination solution for extremely dynamic vehicular ad hoc networks. *Performance Evaluation*, 87:3–18.
- [Peng et al. 2018] Peng, H., Liang, L., Shen, X., and Li, G. Y. (2018). Vehicular communications: A network layer perspective. *IEEE Transactions on Vehicular Technology*, pages 1–1.
- [Sommer et al. 2011] Sommer, C., German, R., and Dressler, F. (2011). Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis. *IEEE Transactions on Mobile Computing*, 10(1):3–15.
- [Tonguz et al. 2010] Tonguz, O. K., Wisitpongphan, N., and Bai, F. (2010). Dv-cast: A distributed vehicular broadcast protocol for vehicular ad hoc networks. *IEEE Wireless Communications*, 17(2):47–57.
- [Toor et al. 2008] Toor, Y., Muhlethaler, P., Laouiti, A., and Fortelle, A. D. L. (2008). Vehicle ad hoc networks: applications and related technical issues. *IEEE Communications Surveys Tutorials*, 10(3):74–88.
- [Tseng et al. 2002] Tseng, Y.-C., Ni, S.-Y., Chen, Y.-S., and Sheu, J.-P. (2002). The broadcast storm problem in a mobile ad hoc network. *Wireless networks*, 8(2-3):153–167.
- [Varga 2003] Varga, A. (2003). Omnet++ simulator. *Department of Telecommunications, Technical University of Budapest, Budapest, Hungary*.
- [Villas et al. 2012] Villas, L. A., Ramos, H. S., Boukerche, A., Guidoni, D. L., Araujo, R. B., and Loureiro, A. A. (2012). An efficient and robust data dissemination protocol for vehicular ad hoc networks. In *Proceedings of the 9th ACM symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*, pages 39–46. ACM.
- [Viriyasitavat et al. 2010] Viriyasitavat, W., Bai, F., and Tonguz, O. K. (2010). Uv-cast: an urban vehicular broadcast protocol. In *Vehicular Networking Conference (VNC), 2010 IEEE*, pages 25–32. IEEE.
- [W.H.O. 2013] W.H.O. (2013). Global status report on road safety 2013: supporting a decade of action: summary. *World Health Organization*.

Um Novo Serviço de Gerenciamento de Tráfego para ITS baseado em Computação em Névoa

Celso A. R. L. Brennand^{1,2} e Geraldo P. R. Filho^{1,3}

¹Instituto de Computação – Universidade Estadual de Campinas

²Unidade Acadêmica de Serra Talhada – Universidade Federal Rural de Pernambuco

³Departamento de Ciência da Computação - CIC – Universidade de Brasília

celsobrennand@gmail.com e geraldoprfilho@ic.unicamp.br

Abstract. *Frustrations, monetary losses, lost time, high fuel consumption and CO₂ emissions are some of the problems caused by a traffic jam in urban centers. In an attempt to solve this problem, this article proposes a traffic service to control congestion, named uRbAN – Roteamento veiculAr na Névoa. uRbAN aims to reduce the problems generated by a traffic jam in a distributed way through suggestion of new routes to vehicles. Unlike the literature works, uRbAN is modeled using Fog computing paradigm. With this, it is possible to take advantage of the inherent aspects of this paradigm, such as low latency, load balancing, scalability, geographical location and the reduction of bandwidth usage. In order to validate uRbAN, our performance evaluation consider realistic urban scenario. When compared with others related works, uRbAN shows a reduction in stop time by up to 70 %, CO₂ emissions by up to 29 % and, planning time index by up to 49 % with satisfactory response time.*

Resumo. *Frustrações, perdas monetárias, perda de tempo, alto consumo de combustível e emissões de CO₂ são alguns dos principais problemas causados por engarrafamento nos centros urbanos. Com o intuito de sanar tais problemas, este artigo propõe o uRbAN – Roteamento veiculAr na Névoa, um serviço de tráfego para controlar os congestionamentos nos centros urbanos. O uRbAN visa reduzir os problemas gerados por um engarrafamento de forma distribuída para sugestão de novas rotas para os veículos. Ao contrário dos trabalhos da literatura, o uRbAN é modelado usando o paradigma da computação em névoa. Com isso, é possível aproveitar os aspectos inerentes de tal paradigma, tais como baixa latência, balanceamento de carga, escalabilidade, localização geográfica dos dados e a redução do uso da largura de banda. Para validar o uRbAN, nossa avaliação de desempenho foi realizada em um cenário urbano realístico. Quando comparado com outros trabalhos relacionados, o uRbAN apresenta uma redução no tempo de parada em até 70 %, na emissão de CO₂ em até 29 %, no PTI em até 49 % com um desempenho satisfatório no tempo de resposta.*

1. Introdução

O desenvolvimento não planejado dos centros urbanos está frequentemente associado aos graves problemas socioeconômicos. Tipicamente, tal crescimento causa um estresse significativo nas estruturas da cidade devido à demanda imprevista de vários recursos e

serviços. Nesse cenário, um dos setores mais afetados são os sistemas de transporte urbano, nos quais as ineficiências podem levar a muitas consequências negativas, tais como o aumento na emissão de *CO2* e um número elevado de tempo no trânsito por causa de congestionamento. No cenário internacional, o custo gerado pelo congestionamento é aproximadamente US\$ 461 bilhões [Cookson and Pishue 2017]. O Brasil está na sétima posição de país com maior índice de congestionamento, a cidade de São Paulo ocupa a quarta posição da cidade mais congestionada no mundo [Cookson and Pishue 2017].

Uma das maneiras para auxiliar na redução do problema de congestionamento, é por meio da utilização do Sistema de Transporte Inteligente (ITS). Um ITS utiliza tecnologias de comunicação, processamento e sensoriamento para melhorar o tráfego urbano e consequentemente, o fluxo de veículos nas vias urbanas. Ainda, o ITS pode ser utilizado para prover serviços de informação e entretenimento aos condutores e passageiros, de maneira a tornar o percurso deslocado um processo mais agradável [Brennand et al. 2015, Meneguette et al. 2018]. Um componente importante para a concepção do ITS é a Rede Veicular. Em tal rede, os veículos são equipados com processadores, sensores e interfaces de comunicação sem fio para comunicar entre si e com elementos pertencentes à infraestrutura da rede, formando uma rede *ad-hoc* [Meneguette et al. 2018].

Os serviços de sugestão de rotas desenvolvidos para o ITS possuem características intrínsecas em relação à maneira como processam, armazenam e disseminam dados [Rizzo et al. 2016]. Por exemplo, os dados coletados (i.e., condições de tráfego) de uma região específica podem ser irrelevantes para outras regiões da cidade. Salienta-se que para tal serviço é necessário compartilhar grandes quantidades de informações providas de fontes de dados heterogêneas como sensores no carro (e.g. GPS, velocidade) e nas vias (e.g. fluxo da via) [Rizzo et al. 2016]. Portanto, o uso de uma infraestrutura centralizada de comunicação, como a computação em nuvem [Hou et al. 2016, Ling et al. 2018], não é adequada para os serviços de sugestão de rotas devido aos atrasos na comunicação, as perdas de dados, as interrupções do canal de comunicação, além de existir problemas com a largura de banda e sobrecarga na rede.

O paradigma de computação em névoa surge como uma alternativa promissora para solucionar tais problemas. Isso ocorre em virtude da computação em névoa disponibilizar os recursos computacionais como armazenamento e processamento na borda da rede, deixando-os mais próximo do usuário final (neste caso, os serviços) sem o auxílio da Internet [Bonomi et al. 2012]. Neste trabalho, o paradigma de computação em névoa é baseado em entidades chamadas de *Cloudlet*. Tais entidades possuem capacidades de computação e comunicação (i.e., micro-data centers), além de serem geograficamente distribuídas para estarem mais próximas das redes de acesso [Satyanarayanan et al. 2009], tornando-as ideais para os requisitos exigidos para o desenvolvimento de serviços de sugestão de rotas no ITS.

Diversos trabalhos da literatura exploram o problema do gerenciamento de rotas em centros urbanos [Lin et al. 2017, Shen et al. 2018, Liebig et al. 2017, Pan et al. 2013, Jeong et al. 2016, Brennand et al. 2016, de Souza et al. 2016]. A maioria dos trabalhos realiza o monitoramento e o controle de tráfego utilizando uma arquitetura baseada na coleta de informações dos veículos e das vias. Para tanto, a arquitetura precisa trocar, processar e armazenar uma enorme quantidade de dados gerados pelos sensores incorporados nos veículos e vias. Assim, problemas relacionados ao processamento (i.e.,

balanceamento de carga e tempo de resposta) e a transmissões de dados (i.e., atrasos, perda de dados e interrupção de comunicação) tornam-se uma preocupação na comunidade acadêmica. Além disso, os serviços de sugestão de rotas possuem um tempo de resposta restrito para realizar o processo de tomada de decisão, de modo que a informação ainda seja útil para os condutores realizarem a mudança da rota.

Com base nas limitações mencionadas anteriormente, e com o intuito de avançar no estado da arte, este artigo propõe um serviço de atendimento ao tráfego para controlar o congestionamento baseado no paradigma de computação de névoa, denominado uRbAN – *Roteamento veiculAr na Névoa*. O uRbAN é baseado nas *Cloudlets* para monitorar as condições do tráfego e calcular as rotas dos veículos. Com isso, o uRbAN aproxima o poder computacional onde é mais necessário, distribuindo a carga de processamento e armazenamento do sistema. Com isso, é possível aumentar a escalabilidade geral do sistema, mantendo a capacidade de coletar, processar e armazenar grandes volumes de dados. Para isso, o uRbAN utiliza as *Road Side Units* (RSU) que são espalhadas por toda cidade como *Cloudlets* para administrar dentro de sua região o tráfego dos veículos. Para o seu funcionamento, foi desenvolvido um mecanismo que reúne todos os dados necessários. Tal mecanismo otimiza a taxa de entrega e reduz o número de mensagens. Em seguida, as *Cloudlets* usando os dados coletados, estimam o congestionamento das vias na região correspondente. Portanto, de acordo com as condições das vias que são incluídas nas rotas do veículo, as *Cloudlets* correspondentes calculam uma nova rota como sugestão. O uRbAN é comparado com outras soluções consagradas da literatura, que exploram o problema de gerenciamento de rotas em vias congestionadas. Os resultados simulados mostram que o uRbAN é mais eficiente quando comparado com tais soluções, apresentando redução no tempo de parada, na emissão de *CO2*, no PTI, além de possuir um desempenho satisfatório no tempo de resposta da infraestrutura de comunicação.

O restante deste estudo está organizado da seguinte forma. A Seção 2 apresenta os trabalhos relacionados. A Seção 3 apresenta o uRbAN e seus componentes. A Seção 4 apresenta a avaliação de desempenho do uRbAN, juntamente com a metodologia utilizada e os resultados obtidos. Finalmente, a Seção 5 apresenta as conclusões e os trabalhos futuros.

2. Trabalhos Relacionados

O problema do gerenciamento de rotas nos centros urbanos tem sido pesquisado por vários trabalhos [Lin et al. 2017, Shen et al. 2018, Liebig et al. 2017, Pan et al. 2013, Jeong et al. 2016, Brennand et al. 2016, de Souza et al. 2016] nos últimos anos, e esta seção possui o intuito de apresentar os principais desafios de pesquisa que este trabalho investiga, destacando os que não foram sanados.

Em [Pan et al. 2013], é proposto um sistema para predição, detecção e controle de congestionamentos com o intuito de reduzir o tempo de viagem, o consumo de combustível e as emissões de *CO2*. Para tanto, uma entidade centralizadora é utilizada para coletar dados em tempo real sobre a condição do tráfego de veículos. Em seguida, utiliza-se três estratégias para calcular um novo caminho: (i) *dynamic shortest path*, calcula a rota com base no menor tempo de viagem; (ii) *random k shortest paths*, seleciona uma das *k* rotas de menor caminho aleatoriamente; e (iii) *entropy balanced k shortest paths*, considera o impacto da via selecionada com base no *random k Shortest Paths* para sugerir

a nova rota. Diferente deste trabalho, as estratégias apresentadas por [Pan et al. 2013] possuem limitações, sendo as principais: (i) geração de novos congestionamentos devido a escolha da via mais livre; (ii) rotas mais longas são geradas devido à escolha aleatória; e (iii) sobrecarga de dados e comunicação devido ao uso da entidade centralizadora.

Jeong et al. [Jeong et al. 2016] propôs um sistema baseado em nuvem para otimizar o tráfego, chamado SAINT. No SAINT, os veículos relatam as condições do tráfego rodoviário a central de controle de tráfego hospedado na nuvem. O SAINT vislumbra que as RSU e eNodeB possuem acesso à Internet para comunicação com a nuvem e os veículos são equipados com 802.11p e 4G. Para redirecionar os veículos, o SAINT utiliza o algoritmo de Dijkstra modificado, no qual a função de peso leva em consideração o tempo necessário para o veículo alcançar a próxima via da rota. Assim, a probabilidade de uma rota se tornar muito popular e causar um novo congestionamento é reduzida. No entanto, o SAINT necessita informar constantemente as condições das rotas por meio da Internet para a nuvem. Além disso, a comunicação DSRC usada pelos veículos, não possui nenhum mecanismo para prover escalabilidade na comunicação, não se adequando em cenários de alta densidade como grandes cidades [De Souza et al. 2017].

Meneguette et al. [Meneguette et al. 2016] propõe o INCIDEnT, uma solução para estimar o nível de congestionamento com base em uma Rede Neural Artificial (RNA). A ideia da proposta é maximizar o fluxo de veículos nos centros urbanos. Para tanto, o INCIDEnT utiliza a velocidade do veículo e a densidade da via como entrada para a RNA estimar a condição do tráfego da via e, em seguida, sugerir novas rotas aos motoristas. Entretanto, o INCIDEnT além de não possuir o conhecimento completo do mapa, não possui um mecanismo para evitar as rotas sobrepostas que podem gerar um novo engarrafamento. Outro problema identificado está relacionado com a inexistência de um mecanismo de supressão de transmissão de dados, prejudicando assim sua eficiência, especialmente no cenário de alta densidade.

O trabalho proposto por [de Souza et al. 2016], apresenta o CHIMERA, um sistema de tráfego inteligente para maximizar a utilização da infraestrutura de transporte e reduzir o tempo médio de viagem dos veículos. Por meio de uma comunicação veicular-infraestrutura, os veículos do CHIMERA fornecem suas informações a uma RSU. Com isso, o CHIMERA realiza a detecção da via congestionada usando o algoritmo *K-Nearest Neighbor* (KNN), tendo como entrada a velocidade do veículo e a densidade da via, similar ao [Meneguette et al. 2016]. Como saída, o KNN fornece a condição do tráfego que é usada no *K-Shortest Path* para a escolha da rota. Entretanto, o CHIMERA não propôs um mecanismo de escalonamento de envio de mensagens para evitar os problemas na transmissão de dados. Além disso, a comunicação entre as RSUs no CHIMERA não é ciente das condições de tráfego de outras regiões do mapa, limitando assim a eficiência do sistema de roteamento, diferentemente do uRbAN como será apresentado a seguir.

3. uRbAN – Roteamento veiculAr na Névoa

Nesta seção é apresentado o uRbAN, um novo serviço de gerenciamento de tráfego para sugestões de novas rotas em vias que não estejam congestionadas. O uRbAN é modelado com base no paradigma da computação em névoa para distribuir os recursos computacionais (i.e., processamento, comunicação e armazenamento) entre os componentes do ITS, conforme apresentado na Figura 1. Para isso, no uRbAN, assumimos que cada RSU é

uma *Cloudlet*, e o conjunto das *Cloudlets* é que formam o ambiente computacional em névoa (Rótulo A, Figura 1). As *Cloudlets* são distribuídas no cenário de acordo com o raio de comunicação da RSU. Cada *Cloudlet* é responsável por coletar, armazenar e processar dados de uma área específica (Rótulo B, Figura 1) para computar novas rotas para os veículos em sua região. Com isso, os recursos computacionais no uRbAN são mantidos mais próximos dos usuários finais, resultando em uma melhoria significativa no sistema no que diz respeito ao processamento, comunicação e tempo de resposta. Salienta-se que as características de contexto espacial dos recursos computacionais permitem que cada *Cloudlet* seja independente uma da outra. Ainda, para aumentar a penetração de dados no ambiente e a flexibilidade na comunicação, as *Cloudlets* são equipadas com várias tecnologias de comunicação, tais como LTE e 802.11p (Rótulo C, Figura 1).

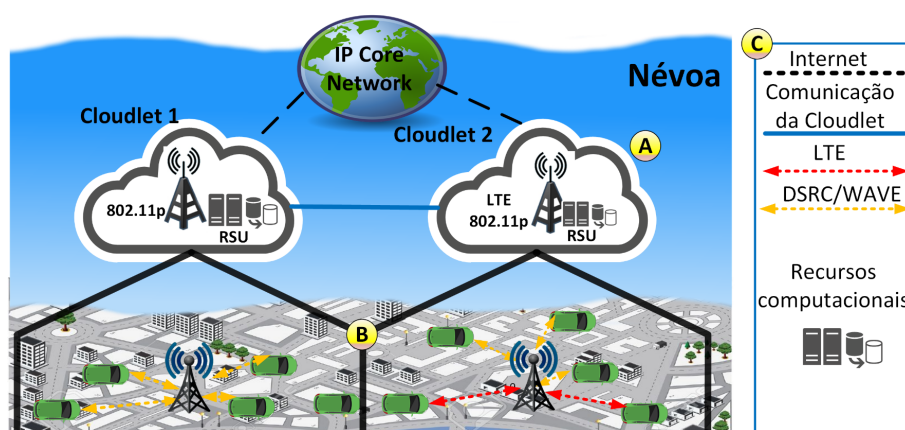


Figura 1. Cenário de funcionamento do uRbAN.

A seguir são apresentadas as três principais etapas para o desenvolvimento do uRbAN, a saber: (i) coleta de dados, Subseção 3.1; (ii) processamento de dados, Subseção 3.2; e (iii) entrega do serviço, Subseção 3.3.

3.1. Coleta de Dados

No uRbAN, um das responsabilidades das *Cloudlets* é coletar os dados dos veículos (i.e., velocidade, posição e densidade) e do ambiente urbano, como a ocupação da via. Com o intuito de obter uma cobertura global do mapa para a coleta dos dados, utilizou-se o algoritmo *Hexagonal Binning* [Carr et al. 1987] que faz uma relação entre o tamanho do cenário e o raio de comunicação das *Cloudlets*. Com isso, foi possível alcançar uma relação eficiente entre a quantidade de *Cloudlets* e o tamanho do mapa da cidade. Tal algoritmo disponibiliza os recursos computacionais mais próximo dos usuários, tornando-o uma estratégia consistente com a computação em névoa.

Para que a coleta de dados funcione adequadamente, as *Cloudlets* e os veículos, enviam *beacons* (i.e., mensagens de controle) periodicamente informando sobre as condições em sua área. A aquisição dos dados é executada de maneira distribuída usando a capacidade de comunicação das *Cloudlets*. Além disso, as *Cloudlets* enviam *beacons* informando sua posição, intervalo de roteamento para os veículos em sua área e a delimitação de sua área de responsabilidade. Um veículo usa tais informações para localizar as *Cloudlets* nas proximidades com o intuito de enviar/solicitar dados (para, por exemplo, saber para quem e quando solicitar uma nova rota). O veículo envia *beacons*

para a *Cloudlet* com dados adquiridos da sua *Onboard Unit* (OBU), informando sua rota atual, posição, velocidade e o tempo gasto para se mover em sua rota. As *Cloudlets* recebem os dados sobre uma área específica e os usam para adquirir conhecimento necessário da condição do trânsito para executar o serviço de tráfego.

Para aumentar a taxa de entrega dos *beacons*, bem como a comunicação das *Cloudlets* com os veículos, um mecanismo de agendamento de mensagens foi desenvolvido com base no padrão de comunicação DSRC/WAVE. A ideia é programar o tempo de envio das mensagens para a camada MAC 802.11p, evitando o problema de resincronização [Donato et al. 2015]. Para isso, o mecanismo agenda a hora de envio das mensagens em duas filas de acordo com o canal de comunicação que será usado para enviar a mensagem. Uma fila para o canal de serviço (SCH) que é usado para troca de mensagens padrões para o serviço (e.g. rotas dos veículos) e outra fila para o canal de controle (CCH) usado para envio de mensagens de emergência e controle do serviço. Ambas as filas são ordenadas de acordo com o tamanho da mensagem, largura de banda da interface de comunicação do veículo/*Cloudlet*, prioridade da mensagem e o intervalo de sincronização dos canais.

3.2. Processamento de Dados

Nesta etapa, o uRbAN processará as informações coletadas da etapa anterior. Como o mapa é coberto por *Cloudlets*, cada *Cloudlet* tem a responsabilidade de processar apenas dados das vias que estão dentro de sua área de cobertura. Com isso, é possível limitar o processamento de dados a uma área menor e, conseqüentemente, reduzir o custo de comunicação no uRbAN. Salienta-se, entretanto, que o tamanho da área de conhecimento que é usada para calcular a nova rota possui impacto direto na qualidade da rota. Portanto, uma área de roteamento com apenas o tamanho do raio de cobertura de rádio da RSU resultará em roteamento ineficiente. Para resolver tal limitação, a *Cloudlet* adquire informações sobre as vias que estão na cobertura de outras *Cloudlets* por meio do modelo de comunicação *request-reply*. Com isso, é possível aumentar a quantidade de informações para utilizar no algoritmo de roteamento. A área de conhecimento adicional contendo informações das vias que estão sob a responsabilidade de outras *Cloudlets* junto com as informações do mapa sob sua responsabilidade, é chamada de Área de Conhecimento (AoK) (círculo azul na Figura 2). Esta divisão é condizente com o paradigma de computação em névoa que é aplicado na modelagem do uRbAN. Em nossa solução, a AoK tem no mínimo o tamanho da área de cobertura da *Cloudlet*.

Salienta-se que a AoK é representada por um grafo ponderado e dirigido $G = (V, E)$, onde $V(G)$ são os vértices e $E(G)$ são as arestas do grafo G . Nesse caso, $V(G) = \{v_1, v_2, \dots, v_i\}$ representa o conjunto de cruzamentos dentro do alcance do AoK. Já $E(G)$ representa o conjunto de vias conectando os cruzamentos. Ainda, o peso $W = \{w_1, w_2, \dots, w_i\}$ de cada via $E_i \in E(G)$ é inversamente proporcional à velocidade com que os veículos viajam na via. Portanto, se a velocidade do veículo estiver próxima da velocidade máxima permitida na via, o W_i da via será menor. O W_i da via é usado pelo algoritmo de roteamento que será descrito na subseção 3.3. Ainda nesta etapa, a classificação da via é feita para ser usada como parâmetro de congestionamento de rota (descrito na subseção 3.3). A classificação da via é baseada nos níveis de serviços (LOS) presente no *Highway Capacity Manual (HCM)* [Board 2010]. O HCM usa a velocidade e a densidade dos veículos nas vias para estimar a capacidade e a qualidade do tráfego.

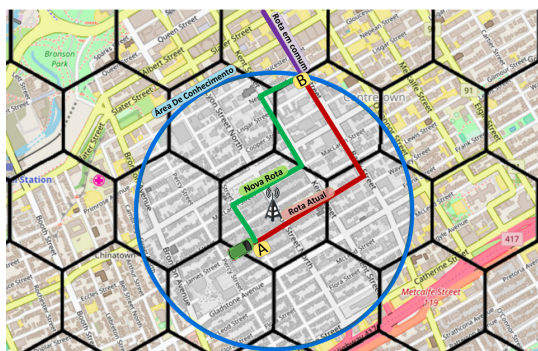


Figura 2. Área de Conhecimento no uRbAN.

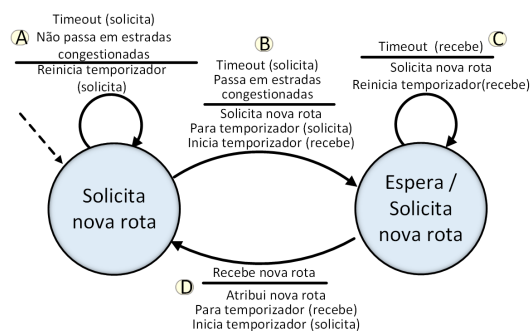


Figura 3. Diagrama de estados: Requisição de nova rota.

A qualidade do tráfego é classificada pelo HCM em 6 níveis (de A ao F), tendo o nível A como fluxo livre e degradando a qualidade até o nível F com o fluxo congestionado. O uRbAN considera uma estrada congestionada quando a velocidade da estrada é classificada como LOS C ou a ocupação da estrada é classificada como LOS D. Tais níveis de limiar foram selecionados pois é uma evidência de um congestionamento emergente. Após a classificação das vias, a *Cloudlet* dissemina por meio de *beacons* as informações das vias congestionadas que estão dentro do seu AoK, como é apresentado na próxima subseção.

3.3. Entrega do serviço

Nesta etapa, cada *Cloudlet* realiza a detecção e o controle do congestionamento calculando rotas alternativas para os veículos. A ideia é maximizar o fluxo de veículos nas vias urbanas. Cada veículo verifica periodicamente se passará por uma via de congestionamento. Para isso, os veículos recebem uma mensagem de alerta enviada pela *Cloudlet* contendo uma lista das vias congestionadas da sua AoK. Assim, a cada intervalo de roteamento, o veículo verifica se a sua rota passa por uma das vias congestionadas/[]]. Para tanto, modelou-se um mecanismo de entrega de serviço, como ilustrado na Figura 3. Se o veículo não passar por um congestionamento (Rótulo A, Figura 3), o intervalo do roteamento será reiniciado. Caso contrário, uma mensagem é enviada para a *Cloudlet* mais próxima com o intuito de solicitar uma nova rota.

Além disso, o tempo de recuperação da solicitação da rota é iniciado (Rótulo B, Figura 3). O tempo de recuperação é um mecanismo de tolerância a falhas que verifica se o veículo recebeu a rota solicitada dentro de um intervalo de tempo adequado¹. Uma nova mensagem de solicitação é enviada em caso de falha no recebimento da nova rota (Rótulo C, Figura 3). Ao receber a nova rota, o tempo do intervalo de roteamento é reiniciado (Rótulo D, Figura 3). Antes de atribuir a nova rota, a OBU do veículo solicitante verifica a satisfação da variável “fator de tamanho da rota”, que determina o quanto maior (em porcentagem) a nova rota tem em comparação com a rota atual. Desta forma, o uRbAN pode limitar o tamanho máximo da rota a ser atribuída, evitando assim o aumento do percurso e de outros problemas de tráfego como a emissão de CO₂.

Vale frisar que as *Cloudlets* calculam uma nova rota para o veículo solicitante no escopo de sua AoK, representado pelo círculo azul na Figura 2. Com isso, o processo de

¹ Definido pelo intervalo de *beacon*, pois é um tempo curto que não compromete o sistema.

re-roteamento não altera a parte da rota fora da AoK (Figura 2 – rota fixa). Como podemos ver na Figura 2, o roteamento de veículos (i.e., carro verde) é executado considerando sua posição atual (Rótulo A, Figura 2) até a última estrada em sua rota atual que está dentro da AoK da *Cloudlet* (Rótulo B, Figura 2).

O fluxograma na Figura 4 apresenta o algoritmo de cálculo da nova rota pela *Cloudlet*. A *Cloudlet* recebe uma solicitação de uma nova rota advinda de uma mensagem enviada por um veículo contendo sua posição atual e rota (Rótulo A, Figura 4). O algoritmo extrai a última via da rota que ainda está dentro da AoK da *Cloudlet* (Rótulo B, Figura 4). O restante da rota que fica fora da AoK é extraída para uso futuro (Rótulo D, Figura 4). Com isso, um conjunto de K caminhos alternativos entre a posição atual e a última via é calculado (Rótulo C, Figura 4). Uma rota deste conjunto é selecionada probabilisticamente baseada no somatório dos pesos (w) de suas vias usando a distribuição de probabilidade de Boltzmann [Kirkpatrick et al. 1983] (Rótulo E, Figura 4). Uma vez selecionada a rota, o sistema verifica se a última via da rota alternativa calculada é o destino do veículo solicitante (Rótulo F, Figura 4). Se não for, a nova rota alternativa é concatenada com o restante da rota original que fica fora da AoK da *Cloudlet* que fez o roteamento (Rótulo G, Figura 4). Entretanto, se o destino estiver na AoK, esta já será a nova rota. Com isso, uma mensagem com a nova rota é gerada e enviada para o veículo solicitante (Rótulo H, Figura 4).

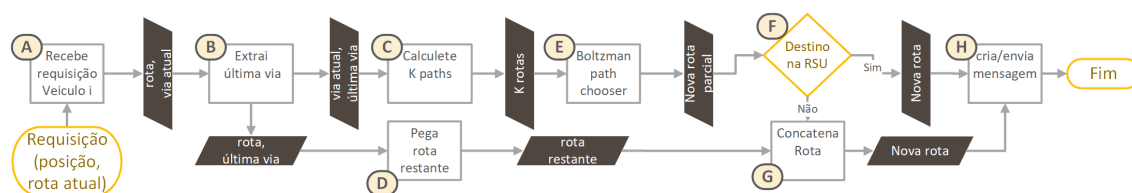


Figura 4. Fluxograma do algoritmo de roteamento do uRbAN.

4. Avaliação de Desempenho

Esta seção avalia o desempenho do uRbAN em termos da eficiência do tráfego e dos recursos computacionais da rede. Para isso, o uRbAN foi comparado com as seguintes soluções: CHIMERA [de Souza et al. 2016], PAN 1 e PAN 3 [Pan et al. 2013], *Dijkstra Shortest Path* (DSP) e com a mobilidade original do cenário, denominado *BASE*. O cenário modelado, as métricas selecionadas e a metodologia utilizada para gerar os resultados são apresentados.

4.1. Metodologia

Para avaliar o desempenho do uRbAN, foi utilizado o simulador de rede OMNeT++ 5.1². Para a simulação de tráfego e mobilidade dos veículos, utilizou-se o SUMO 0.25.0³, um simulador de tráfego aberto que permite modelar e manipular objetos no cenário urbano. Para a rede veicular, utilizou-se o *framework* Veins 4.3⁴ que implementa o IEEE 802.11p e IEEE 1609.4 DSRC/WAVE, além de um modelo de atenuação de sinal que considera obstáculos. Para medir as emissões de CO₂ e o consumo de combustível, foi

²<http://www.omnetpp.org>

³<http://sumo.dlr.de>

⁴<http://veins.car2x.org/>

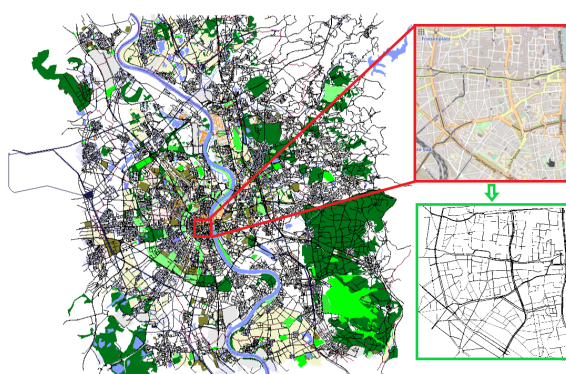


Figura 5. Topologia do cenário de Cologne.

Parâmetros	Valores
Potência de transmissão	2.2 mW
Cobertura	300 m
Taxa de transferência	18 Mbit/s
Intervalo de Beacons	4 s
Rotas alternativas (k)	3
AoK	3000 m
Intervalo de roteamento	120s

Tabela 1. Parâmetros utilizados na simulação.

utilizado o modelo EMIT⁵ do SUMO. Com o intuito de obter um cenário realista das condições de tráfego da cidade para as avaliações, foi utilizado o cenário de TAPAS Cologne [Uppoor et al. 2014] que representa a mobilidade da cidade de Colônia, Alemanha. Foi selecionada uma subárea de 4,5 km² do centro da cidade de Colônia (Figura 5) durante o período compreendido entre 7:00 e 8:00 horas. Período este que possui a maior ocorrência de congestionamento na subárea selecionada. Este cenário possui aproximadamente 14.000 veículos inseridos durante a simulação e uma extensão de 243 km de estradas.

A Tabela 1 sumariza os parâmetros utilizados na simulação. Os parâmetros da rede para todas as simulações foram ajustados para uma taxa de transferência de 18Mbit/s na camada MAC e a potência de transmissão para 2,2mW, resultando em uma cobertura de aproximadamente 300m sob um modelo de propagação de dois raios [Sommer et al. 2011]. O intervalo de roteamento foi definido em 120s para todas as soluções e são usadas três rotas alternativas para o uRbAN, CHIMERA e PAN3 com o intuito de realizar uma avaliação mais justa. Além disso, por meio de experimentos empíricos, notou-se que tal número de rotas alternativas foi o que apresentou os melhores resultados para todas as soluções nos cenários avaliados. Salienta-se que os veículos na solução DSP não divulgam informações sobre o tráfego e usam o comprimento da via como classificação da estrada. A AoK do uRbAN foi definida em 3 km.

A validação do uRbAN foi dividida em duas etapas, eficiência do tráfego e recursos computacionais da rede. Para validar a eficiência do tráfego, utilizou-se as seguintes métricas: (i) **tempo de parada**, que representa o tempo que todos os veículos ficaram preso nos engarrafamentos; (ii) **velocidade média**, que representa a velocidade média de todos os veículos; (iii) **distância percorrida**, que representa a distância média percorrida por todos os veículos; (iv) **emissão de CO2**, que representa a quantidade de CO2 emitido de todos os veículos; e (v) **Planning Time Index**, PTI, que mensura a confiabilidade do tempo de viagem obtida em relação ao fluxo ideal no mesmo caminho. Por exemplo, em uma viagem que duraria 25 minutos em tráfego livre, com o PTI no valor de 2 esta mesma viagem no mesmo trajeto durará 50 minutos. Após validar a eficiência do tráfego, validou-se os recursos computacionais da rede. Para isso, utilizou-se as seguintes métricas: (i) **total de pacotes**, que representa a quantidade de pacotes transmitidos; (ii)

⁵<http://www.hbefa.net>

colisões, que representa a porcentagem de pacotes colididos durante a simulação; (iii) **atraso da rede**, que representa o tempo médio para disseminar os pacotes para todos os veículos; (iv) **rota aceita**, que representa a quantidade de novas rotas aceitas por veículo; e (v) **rota computada por cloudlet**, que representa a quantidade de rotas calculadas por cada *Cloudlet*. Para todos os experimentos, foram executadas 33 simulações com um intervalo de confiança de 95% de acordo com o t-test. Os resultados obtidos são apresentados nas próximas subseções.

4.2. Impacto da eficiência do tráfego

A Figura 6 apresenta os resultados simulados do uRbAN, comparando-os com as soluções implementadas. Como observado na Figura 6(a), as soluções PAN1, PAN3, CHIMERA e uRbAN reduziram o tempo de parada em 62.36 %, 55.49 %, 65.62 % e 73.18 % em relação à solução BASE, respectivamente. Tais resultados refletem em uma velocidade média maior, como ratificado na Figura 6(b). Vale salientar que independente das soluções, o uRbAN apresentou um desempenho superior no que diz respeito à velocidade média, tendo um aumento de 24.92 %. Isso ocorre devido ao mecanismo probabilístico do uRbAN que seleciona a melhor rota das K rotas alternativas, além do uso da classificação da via que representa de maneira mais adequada a condição do fluxo de veículos na via. Nota-se também que o DSP obteve o tempo de parada aumentado em 63 %, e a velocidade média reduzida em 9.51 %, quando comparado com o BASE. Isto faz sentido pois, o DSP calcula apenas um caminho mais curto para todos os veículos, movendo os veículos para uma mesma estrada. Em razão disso um novo ponto de congestionamento é gerado, diferentemente da nossa solução que recalcula uma nova rota quando necessário. Entretanto, ao considerar a métrica velocidade média, o PAN1 com um aumento de 17.74 % obteve um desempenho melhor quando comparado ao PAN3 com um aumento de 13.24 % na velocidade média. Isso é coerente, posto que o PAN3 calcula três rotas alternativas e escolhe uma aleatoriamente. Dessa forma, as segundas e terceiras melhores rotas podem ter sua classificação do trânsito e tamanhos bem diferentes em comparação com a melhor rota. Salienta-se que isso não acontece no uRbAN, uma vez que a rota alternativa é selecionada através do método probabilístico implementado.

Como esperado, o DSP foi o único a reduzir a distância percorrida (2.32 %) entre as soluções avaliadas (Figura 6(c)). Observa-se que o PAN1 e o uRbAN obtiveram as menores reduções nas rotas com 1.35 % e 1.53 % em relação ao BASE, respectivamente. Já o CHIMERA obteve um aumento no tamanho da rota em 3.78 % quando comparado com o BASE, e em 2 vezes em relação ao uRbAN. Isso ocorreu porque o CHIMERA não possui um mecanismo para avaliar o tamanho da rota, diferentemente do uRbAN que implementa tal mecanismo. Na emissão de CO_2 , Figura 6(d), o uRbAN obteve uma redução de 28.25 %, o CHIMERA de 23.85 %, o PAN1 de 24.34 %, e o PAN3 de 21.12 %. Isto é, o uRbAN reduz a emissão de CO_2 independentemente das soluções comparadas. Ainda, observa-se que apesar do CHIMERA obter um tempo de parada e velocidade média melhores que o PAN1, a sua emissão de CO_2 é maior devido à distância percorrida ser maior.

A Figura 7(a) apresenta a qualidade de controle de congestionamento das soluções avaliadas utilizando a métrica PTI. Como pode ser observado na Figura 7(a), o uRbAN obteve o melhor resultado tendo o PTI reduzido em 49 % em relação ao BASE. Já o CHIMERA, o PAN1 e o PAN3 obtiveram o PTI reduzido em 44 %, em 39 % e em 35 %, respectivamente.

respectivamente. Para verificar quais soluções funcionam melhor de acordo com o tamanho da rota, a Figura 7(b) apresenta o PTI para intervalos de 500 metros. Nota-se que o uRbAN obteve um comportamento estável com o crescimento da rota, independente das soluções comparadas que obteve um aumento considerável do PTI. Portanto, tais resultados mostram que o uRbAN é capaz de lidar de maneira mais eficiente com o problema do tráfego da cidade.

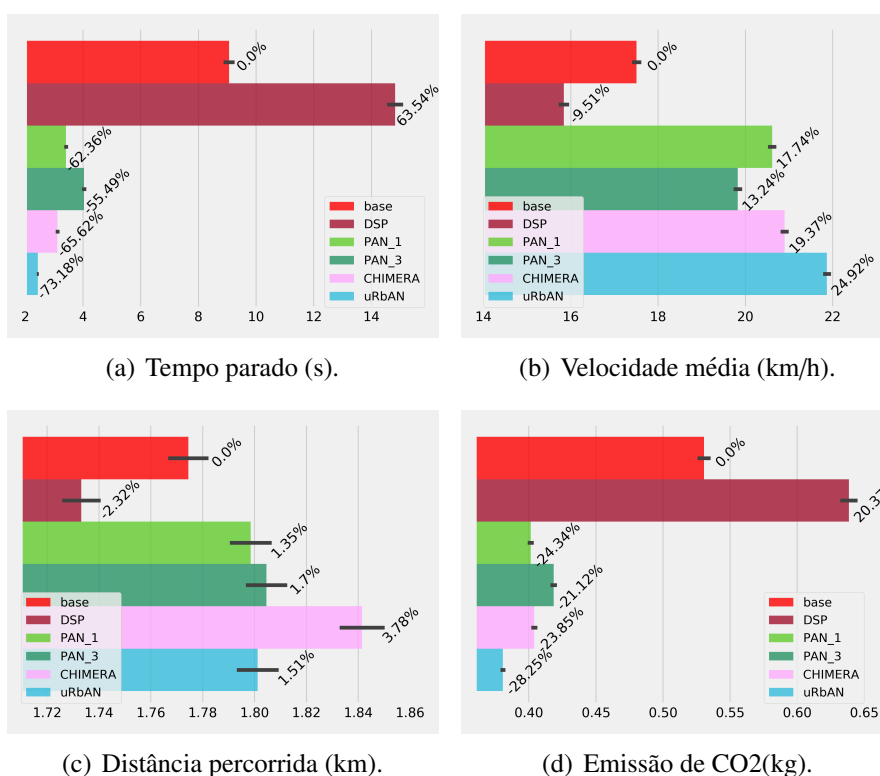


Figura 6. Eficiência do Tráfego.

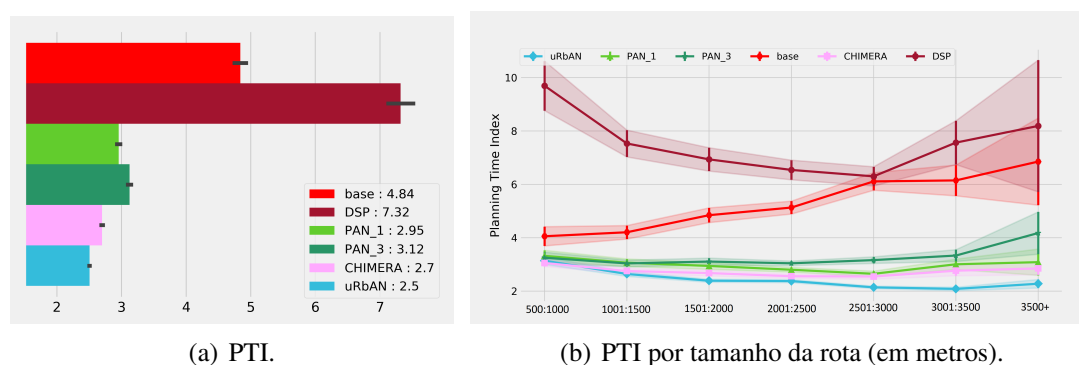


Figura 7. Impacto do PTI.

4.3. Impacto dos recursos computacionais da rede

A Figura 8(a) apresenta o número total de mensagens transmitidas durante a simulação. Observa-se que o uRbAN transmite uma quantidade maior de mensagens quando comparado com o CHIMERA, PAN1 e PAN3. Entretanto, vale destacar que o uRbAN possui

menos colisões em relação ao PAN1, PAN3 e CHIMERA, Figura 8(b). Apesar do número de mensagens transmitidas pelo uRbAN ser maior, o atraso da rede para disseminar os pacotes para todos os veículos não possui diferença estatística significativa entre as soluções, como observado na Figura 8(c). Isso ocorre devido ao mecanismo de agendamento de mensagens para a camada MAC que evita o problema de ressinchronização. Com isso, é possível lidar com a concentração de veículos em uma região que concorre com o mesmo canal de comunicação da rede.

Em relação ao número de rotas aceitas pelos veículos, o uRbAN possui uma média superior, como apresentada na Figura 8(d). No pior caso, o uRbAN possui em média 0.775 vezes mais acertos por veículos re-roteados que o CHIMERA. Já no melhor caso, o uRbAN possui 0.829 vezes mais acertos por veículos re-roteados que o PAN1. De maneira geral, um maior número de veículos roteados de maneira correta, possui um aumento na qualidade do tráfego da cidade. A Figura 8(e) apresenta as médias das rotas

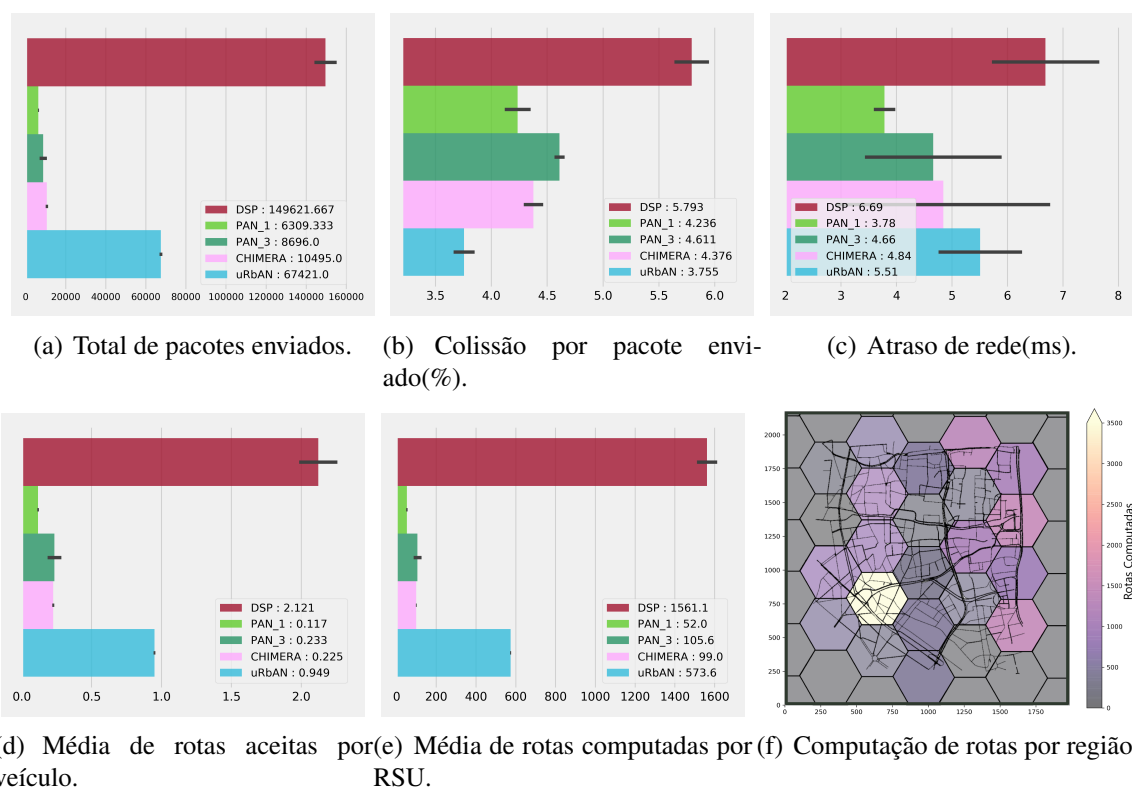


Figura 8. Custo de Rede e Recursos Computacionais.

computadas por cada *cloudlet*. O uRbAN obteve 573.6 rotas computadas, enquanto o PAN1, PAN3 e CHIMERA obtiveram 52, 105.6 e 99 rotas computadas, respectivamente. O elevado número de rotas computadas não prejudica o desempenho do uRbAN por causa do uso do paradigma de computação em névoa, o qual distribui o cálculo das rotas por regiões em várias *Cloudlets*. Isso pode ser confirmado pela Figura 8(f), que apresenta geograficamente a distribuição dos cálculos das rotas do uRbAN no mapa, expondo as áreas com maior demanda por *Cloudlets*. Em outras palavras, o uRbAN distribui a carga de processamento e armazenamento do sistema onde é mais necessário. Portanto, tais resultados mostram que o uRbAN é mais adequado para lidar com congestionamentos de tráfego nos cenários avaliados.

5. Conclusão

Este artigo propôs o uRbAN, um serviço de atendimento ao tráfego para controlar o congestionamento baseado no paradigma de computação de névoa. O uso do paradigma de computação em névoa apresentou diversos benefícios para o uRbAN, como distribuição computacional, fornecendo o balanceamento de carga de processamento e a proximidade dos recursos computacionais com os usuários, diminuindo o tempo de resposta e o uso da largura de banda da rede. Os resultados simulados mostram que o uRbAN é mais eficiente em diversos cenários quando comparado com outras soluções da literatura. Os resultados mostraram ser relevantes pois apresentaram melhorias significativas no tráfego urbano com redução no tempo de parada, emissão de CO_2 e PTI. Em relação ao custo computacional da rede, o uRbAN também mostrou melhorias significativas reduzindo a colisão de pacotes com um menor tempo de resposta. Como trabalho futuro, pretende-se realizar experimentos considerando outros cenários e o uso de outras tecnologias de comunicação como LTE.

Referências

- Board, T. R. (2010). *HCM 2010 - Highway capacity manual*. National Research Council.
- Bonomi, F., Milito, R., Zhu, J., and Addepalli, S. (2012). Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pages 13–16. ACM.
- Brennand, C. A., Cunha, F., Maia, G., Cerqueira, E., Loureiro, A. A., and Villas, L. A. (2016). Fox: A traffic management system of computer-based vehicles fog. In *2016 IEEE Symposium on Computers and Communication (ISCC)*, pages 1–6. IEEE.
- Brennand, C. A., de Souza, A. M., Maia, G., Boukerche, A., Ramos, H., Loureiro, A. A., and Villas, L. A. (2015). An intelligent transportation system for detection and control of congested roads in urban centers. In *2015 IEEE Symposium on Computers and Communication (ISCC '15)*, pages 663–668. IEEE.
- Carr, D. B., Littlefield, R. J., Nicholson, W. L., and Littlefield, J. S. (1987). Scatterplot matrix techniques for large n. *Journal of the American Statistical Association*, 82(398):424–436.
- Cookson, G. and Pishue, B. (2017). Inrix global traffic scorecard.
- De Souza, A. M., Brennand, C. A., Yokoyama, R. S., Donato, E. A., Madeira, E. R., and Villas, L. A. (2017). Traffic management systems: A classification, review, challenges, and future perspectives. *International Journal of Distributed Sensor Networks*, 13(4):1550147716683612.
- de Souza, A. M., Yokoyama, R. S., Maia, G., Loureiro, A., and Villas, L. (2016). Real-time path planning to prevent traffic jam through an intelligent transportation system. In *Computers and Communication (ISCC), 2016 IEEE Symposium on*, pages 726–731. IEEE.
- Donato, E., Maia, G., Duarte, J. M., Loureiro, A. A., Madeira, E., and Villas, L. (2015). Presync: A method for preventing resynchronization in the IEEE 802.11 p standard. In *Computers and Communication (ISCC), 2015 IEEE Symposium on*, pages 457–462. IEEE.

- Hou, X., Li, Y., Chen, M., Wu, D., Jin, D., and Chen, S. (2016). Vehicular fog computing: A viewpoint of vehicles as the infrastructures. *IEEE Transactions on Vehicular Technology*, 65(6):3860–3873.
- Jeong, J., Jeong, H., Lee, E., Oh, T., and Du, D. H. (2016). Saint: Self-adaptive interactive navigation tool for cloud-based vehicular traffic optimization. *IEEE Transactions on Vehicular Technology*, 65(6):4053–4067.
- Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P., et al. (1983). Optimization by simulated annealing. *science*, 220(4598):671–680.
- Liebig, T., Piatkowski, N., Bockermann, C., and Morik, K. (2017). Dynamic route planning with real-time traffic predictions. *Information Systems*, 64:258–265.
- Lin, J., Yu, W., Yang, X., Yang, Q., Fu, X., and Zhao, W. (2017). A real-time en-route route guidance decision scheme for transportation-based cyberphysical systems. *IEEE Transactions on Vehicular Technology*, 66(3):2551–2566.
- Ling, C. W., Datta, A., and Xu, J. (2018). A case for distributed multilevel storage infrastructure for visual surveillance in intelligent transportation networks. *IEEE Internet Computing*, 22(1):42–51.
- Meneguette, R. I., Filho, G. P. R., Guidoni, D. L., Pessin, G., Villas, L. A., and Ueyama, J. (2016). Increasing intelligence in inter-vehicle communications to reduce traffic congestions: Experiments in urban and highway environments. *PLOS ONE*, 11(8):1–25.
- Meneguette, R. I., Robson, E., and Loureiro, A. A. F. (2018). *Intelligent Transport System in Smart Cities: Aspects and Challenges of Vehicular Networks and Cloud*. Springer.
- Pan, J., Popa, I. S., Zeitouni, K., and Borcea, C. (2013). Proactive vehicular traffic rerouting for lower travel time. *IEEE Transactions on vehicular technology*, 62(8):3551–3568.
- Rizzo, G., Palattella, M. R., Braun, T., and Engel, T. (2016). Content and context aware strategies for qos support in vanets. In *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*, pages 717–723. IEEE.
- Satyanarayanan, M., Bahl, P., Caceres, R., and Davies, N. (2009). The case for vm-based cloudlets in mobile computing. *IEEE pervasive Computing*, 8(4).
- Shen, Y., Lee, J., Jeong, H., Jeong, J., Lee, E., and Du, D. H. (2018). Saint+: Self-adaptive interactive navigation tool+ for emergency service delivery optimization. *IEEE Transactions on Intelligent Transportation Systems*, 19(4):1038–1053.
- Sommer, C., German, R., and Dressler, F. (2011). Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis. *IEEE Transactions on Mobile Computing*, 10(1):3–15.
- Uppoor, S., Trullols-Cruces, O., Fiore, M., and Barcelo-Ordinas, J. M. (2014). Generation and analysis of a large-scale urban vehicular mobility dataset. *IEEE Transactions on Mobile Computing*, 13(5):1061–1075.

Realocação em Sistemas de Compartilhamento de Veículos em Uma Via

Felipe Rooke¹, Victor Aquiles Alencar¹, Marcelo de O. C. Machado²,
Alex Borges Vieira¹

¹Departamento de Ciência da Computação - UFJF

²Pós-Graduação em Informática - UNIRIO

felipe.rooke@ufjf.edu.br, victoraquiles@ice.ufjf.br

marcelo.machado@edu.unirio.br, borges@dcc.ufmg.br

Abstract. *The understanding of urban mobility is useful to create more intelligent and sustainable cities. Currently, car-sharing services add more dynamism to transportation in cities. However, some modalities of these services occasionally present imbalances of vehicles in the regions of operation. That is, one region holds too many vehicles, while in others, vehicles are scarce. Thus, this paper presents a characterization of a car-sharing system, operating in Vancouver and nearby regions, revealing some patterns of use and demand of users. In addition, discussions were presented that support the resolution of the problem addressed and that can contribute to the evolution of these systems.*

Resumo. *A compreensão da mobilidade urbana é útil para criação de cidades mais inteligentes e sustentáveis. Atualmente, serviços de compartilhamento de veículos agregam maior dinamismo ao transporte nas cidades. Entretanto, algumas modalidades desses serviços ocasionalmente apresentam desbalanceamentos de veículos nas regiões de atuação. Isto é, uma região detém demasiada quantidade de veículos, enquanto em outras, os veículos estão escassos. Assim, este trabalho apresenta uma caracterização de um sistema de compartilhamento de veículos, que opera em Vancouver e regiões próximas, revelando alguns padrões de uso e de demanda dos usuários. Ainda, foram apresentadas discussões que dão suporte a resolução do problema abordado e que podem contribuir para a evolução desses sistemas.*

1. Introdução

Mobilidade urbana é uma área chave e atrai estudos acadêmicos e investimentos privados. Ela está fortemente atrelada a diversas atividades urbanas, como a demanda por recursos de comunicação. Além disso, a massificação dos dispositivos móveis tornou o acesso a rede ubíquo e centrado no usuário. Nesse cenário, compreender a mobilidade urbana, em específico no trânsito, é importante para uma série de tarefas, desde planejamento das malhas viárias aos recursos de comunicação [Herrera et al. 2010, Ma et al. 2013].

O primeiro passo para compreender padrões de mobilidade urbana é a obtenção de dados. Entretanto, essa é uma tarefa desafiadora, onde em muitos casos, poucas empresas têm acesso a dados acurados e atuais e, na maioria das vezes, esses dados são sensíveis

com relação a privacidade [Ciociola et al. 2017]. Assim, é importante gerar modelos que possam representar os padrões de mobilidade urbana e suas interações sociais.

Atualmente, sistemas de compartilhamento de veículos (*car-sharing*) têm recebido crescente atenção da comunidade acadêmica [Boldrini et al. 2016, Ciociola et al. 2017, Becker et al. 2017]. Esses sistemas permitem que as pessoas agendem o uso de um veículo e realize suas tarefas cotidianas, sem se preocupar com manutenção, estacionamento e impostos. Esses sistemas apresentam grande volume de usuários e podem ser representativos quanto aos padrões de mobilidade urbana. De fato, até 2015 foram contabilizados mais de 1,5 milhão de usuários e mais de 22 mil veículos compartilhados nas Américas – com previsão de crescimento [Shaheen 2016].

Existem três principais modelos de negócio para operação de serviços de compartilhamento de veículos [Nourinejad 2014]: (1) serviços de uma via (*one-way*), onde existem estações bases espalhadas em uma região e o usuário pode alugar um veículo em uma estação e deixá-lo em outra; (2) serviços de duas vias (*two-way*), onde o usuário aluga um veículo e deve deixá-lo na mesma estação que o retirou e (3) serviços flutuantes (*free-floating*), onde não há a necessidade de estações base e o usuário fica livre para alugar um veículo e deixá-lo em qualquer região de operação do serviço [Boldrini et al. 2016].

Um problema associado aos serviços *one-way* e *free-floating* é o desbalanceamento de veículos em determinadas regiões ou estações. Devido a uma demanda de saída elevada de veículos em determinado local, e a uma entrada não equivalente, possivelmente menor, aquele local pode ficar desabastecido. Por outro lado, outras localizações cuja demanda de saída não é elevada, podem ficar com excesso de veículos. Realizar a realocação de veículos pode ser muito custoso para um operador, portanto, nesse contexto, estudos para identificar melhores estratégias de realocação estão sendo desenvolvidos.

Neste trabalho, foi caracterizado o *Evo*¹, um serviço importante de compartilhamento de veículos. As caracterizações conduzidas evidenciam os padrões de uso desse sistema e desbalanceamentos que ocorrem em seu uso. Foram caracterizados aspectos espaço-temporais que servem de substrato para o desenvolvimento de algoritmos de realocação de veículos, com foco na economia e redução de custos operacionais.

O restante do artigo está organizado como segue: a Seção 2 descreve os modelos de compartilhamento de veículos bem como trabalhos relacionados; a Seção 3 expõe a metodologia do trabalho; a Seção 4 propõe uma caracterização relacionada a base de dados da *Evo*; a Seção 5 discute e apresenta propostas de trabalhos futuros para realocação de veículos a partir da caracterização apresentada e, por fim, a Seção 6 conclui o trabalho.

2. Modelos de compartilhamentos de veículos e trabalhos relacionados

De maneira geral, um usuário localiza e agenda um carro através do aplicativo, se desloca até ele, e realiza seu destrancamento também pelo smartfone. Alguns serviços, solicitam que o usuário inspecione o veículo, em busca de algum tipo de dano, antes de iniciar seu uso. Quando um usuário realiza o destrancamento de um veículo, seu aluguel começa a ser contabilizado. Após a utilização, o usuário deve retornar com o veículo a uma estação ou a um estacionamento público, a depender do modelo de negócio do serviço.

De forma ampla, existem dois modelos de negócios de compartilhamento de

¹<https://www.evo.ca/>

veículos: os baseados em estações e os flutuantes. Nos modelos baseados em estações, o usuário deve se dirigir a um local específico onde se encontram os veículos. No modelo flutuante, os veículos não têm localização específica e se encontram espalhados em zonas de atuação do serviço (*home zone*). Ainda, os serviços baseados em estações podem ser de uma ou duas vias.

A dinâmica da demanda dos sistemas de uma via e flutuante pode, em alguns casos, tornar os veículos escassos em algumas regiões. Ocasionalmente, locais onde são originadas muitas viagens, ficam desabastecidos de veículos, enquanto outros, onde ocorre o inverso, muitas viagens se destinam a eles, ficam saturados. Periodicamente, os operadores dos serviços, realocam veículos entre os locais onde existe essa necessidade. Comumente, é definido um limite superior e inferior da quantidade de veículos em cada local, e esse, é verificado para alertar quando as realocações devem ser realizadas [Kek et al. 2009]. Efetuar realocações pode ser muito custoso para os operadores. Portanto, as estratégias de realocação também devem observar se os ganhos da estratégia compensam o custo de realocação [Weikl and Bogenberger 2013].

Atualmente existem três tipos de estratégias de realocação, a realocação baseada no usuário (*User-Based*), a baseada no operador (*Operator-Based*) e uma estratégia híbrida, combinando as estratégias anteriores. A estratégia de realocação baseada no usuário, é comumente realizada mediante bônus e reduções de tarifas no aluguel dos veículos, caso o usuário percorra determinado trajeto, estipulado pela operadora. Essa estratégia tem um baixo custo de implantação. Além disso, apresenta um efeito a médio e longo prazo, por exemplo, condicionando um usuário a incluir a utilização do serviço em sua rotina para determinado trajeto [Weikl and Bogenberger 2013]. A realocação baseada no operador, apresenta uma clara diferença da estratégia anterior, pois a operadora do serviço intervém diretamente na realocação dos veículos. Agentes ficam responsáveis por realizar a movimentação de veículos de um local a outro. Essa estratégia é mais custosa que a anterior, no entanto sua execução ocorre em curto prazo, com operações bem definidas, com hora de início e fim [Weikl and Bogenberger 2013]. Finalmente, tem-se a estratégia híbrida. Um exemplo de cenário onde essa estratégia pode ser adotada é quando há uma situação onde é necessária uma ação a curto prazo e, no entanto, já existe uma realocação baseada em usuários acontecendo. Nesse caso, uma realocação baseada no operador seria realizada [Weikl and Bogenberger 2013].

A Figura 1 apresenta um modelo de dois passos para realocações de veículos em serviços de compartilhamento de veículos flutuantes. Nesse sistema há o *Módulo de Demanda Offline* e o de *Otimização Online*. O primeiro fornece uma análise das demandas de grupos identificados no serviço, e também a predição da demanda. Os resultados desse módulo fornecem informações para a predição da disponibilidade espacial ideal futura dos veículos do serviço. O segundo mede a diferença entre a disponibilidade espacial ideal futura e o estado atual do serviço. Além disso, é responsável por decidir o melhor momento de realocação e qual estratégia utilizar.

O Módulo de Demanda Offline é executado periodicamente. A ideia é realizar a análise de padrões de demanda que são repetidos ao longo do tempo e espaço. O primeiro passo nesse módulo, é a aquisição dos dados a serem analisados. É importante que seja possível identificar características espaciais e temporais nesses dados. O próximo passo, é o agrupamento e análise dos dados. Esse passo pode ser executado de diversas formas,

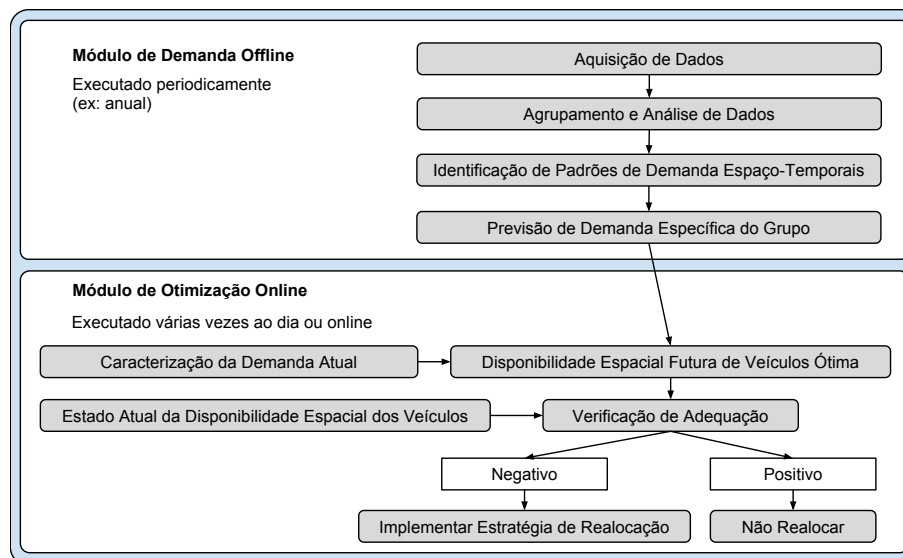


Figura 1. Modelo de dois passos para realocação de veículos em sistemas de compartilhamento de veículos (Adaptação do modelo proposto por [Weikl and Bogenberger 2013])

utilizando algoritmos de agrupamento conhecidos, agrupando por regiões da cidade, utilizando um critério de agrupamento próprio, entre outros. Assim que definido os grupos, o próximo passo é a caracterização, onde são identificados os padrões de demanda espaço-temporais de cada grupo. Por último, é criado um modelo que permite a previsibilidade das demandas específicas de cada grupo.

O Módulo de Demanda Online é executado constantemente. Ele é responsável por verificar a demanda atual do serviço e juntamente com as informações fornecidas pelo módulo Offline, efetuar a tomada de decisão sobre a realocação. Nesse módulo, é feita uma caracterização da demanda corrente do serviço com informações recentes. Essa caracterização, juntamente com as informações fornecidas pelo módulo Offline, definem qual seria a disponibilidade espacial futura ótima dos veículos no momento observado. Uma vez que isso está definido, é possível verificar o estado atual da disponibilidade espacial dos veículos e verificar se está adequada ou não à disponibilidade futura ótima. Caso não esteja, o módulo implementa e sugere uma estratégia de realocação.

Neste trabalho, exploramos partes do Módulo de Demanda Offline, proposto no modelo de dois passos para realocação de veículos em sistemas de compartilhamento de veículos. Utilizamos um método de agrupamento e discretização para definição dos grupos cujas demandas foram posteriormente caracterizadas. Além disso, acreditamos que as características observadas fornecem subsídios para apoiar tanto estratégias baseadas no usuário, quanto estratégias baseadas no operador.

3. Estrutura da pesquisa

Este estudo se baseia em dados de utilização do serviço de compartilhamento de veículos *Evo*. Esse é um serviço conhecido e que opera na cidade de Vancouver (Canadá) e regiões próximas. O *Evo* não oferece uma API pública para pesquisadores. Por isso, foram coletados dados que estão disponíveis publicamente em seu portal web. A cada minuto, uma lista de todos os veículos existentes no serviço foram coletadas. Nesse mesmo intervalo de

tempo foram coletadas informações daquele instante (*snapshot*) do serviço, descrevendo quais veículos estão estacionados, onde estão e se estão disponíveis para viagem.

Todos os *snapshots* do sistema foram processados para inferir os momentos em que o veículo estava ocupado (alugado) ou ocioso (estacionado em uma estação). Durante um *snapshot*, se um veículo está listado entre os veículos do sistema, mas não estiver estacionado em nenhuma estação, infere-se que o mesmo está em uso. Então, foi configurado o ponto de partida de uma nova viagem como a última estação em que o veículo estava estacionado. Analogamente, o ponto final da viagem será a próxima estação em que o veículo aparecer disponível em um *snapshot* futuro. O tempo total da viagem é contabilizado como a diferença entre os tempos dos dois *snapshots*.

Assim, a coleta dos dados segue os seguintes passos: inicialmente, coleta-se o *snapshot* do serviço *Evo* com todos os carros disponíveis no minuto corrente. Depois, é feita uma consulta à base de dados capturando os veículos disponíveis no minuto anterior. Para cada carro presente no minuto anterior, é verificado se o mesmo desaparece no minuto atual, nesse caso inicia-se um novo registro de viagem, com as últimas coordenadas conhecidas do veículo. Posteriormente, para cada carro do minuto atual, é verificado se ele não estava presente no minuto anterior, nesse caso, ele reapareceu. Dessa forma, os passos seguintes do algoritmo servem para guardar o destino final do registro de viagem.

Os dados coletados do *Evo* compreendem um período de 5 meses de coleta, de 1º de Março de 2018 a 16 de Julho de 2018. É importante observar que, para não violar a privacidade dos usuários, não foram expostas nenhuma informação pessoal dos mesmos. Foram coletados aproximadamente 143 milhões de *snapshots* do serviço e 645 milhões de registros de viagens. O serviço apresenta 130 estações e 1237 veículos. Note que neste serviço não é necessário um grande número de estações devido a possibilidade de parada em estacionamentos públicos dentro da área delimitada de atuação do serviço. Esta área é chamada de *home zone* e cobre Vancouver e algumas cidades vizinhas.

A fim de compreender a demanda e as dinâmicas na área de atuação do serviço, buscou-se delimitar as regiões cujas características eram semelhantes. Isto é, foram utilizados dados que representam padrões de utilização dos veículos e não a divisão geográfica da cidade. O método K-Means² foi utilizado para identificar essas regiões através de agrupamentos e, posteriormente, para discretização considerando o grupo mais predominante em certa região. Para o agrupamento foram considerados os seguintes parâmetros: duração da viagem, distância percorrida, período do dia em que a viagem foi iniciada (manhã, tarde, noite e madrugada) e coordenadas de início e fim das viagens. Entretanto, para realização do agrupamento, foi necessária uma pré etapa de limpeza, adequação e normalização nos dados. Portanto, foram removidas viagens que duraram menos de 30 minutos e percorreram menos de 3 metros uma vez que, tais características são atribuídas à viagens canceladas. Então, os dados foram divididos em dias de semana e fins de semana. Além disso, a hora de início das viagens foi discretizada para o formato manhã, tarde, noite ou madrugada.

Para determinar um número mais adequado de grupos foi utilizado o método *Elbow*. A Figura 2 mostra a execução do método para dias de semana e finais de semana.

²Em mineração de dados, o k-means é um método de clusterização que objetiva particionar n observações dentre k grupos onde cada observação pertence ao grupo mais próximo da média.

Para cada k foram executadas 30 iterações do K-Means, ao todo foram testados 25 valores diferentes para k . Como pode ser observado, ambos períodos obtiveram figuras semelhantes, assim foi escolhido o número 7 para o valor de K , pois a partir desse valor as distâncias dos erros quadráticos se estabilizam.

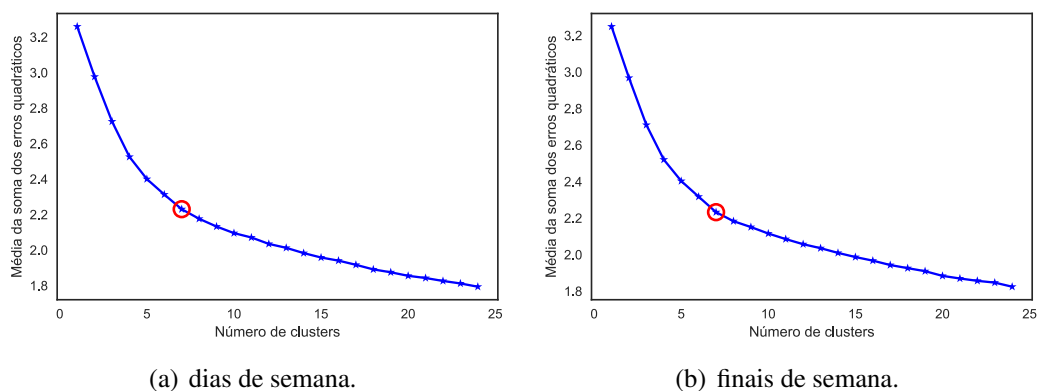


Figura 2. Método Elbow para clusterização K-Means.

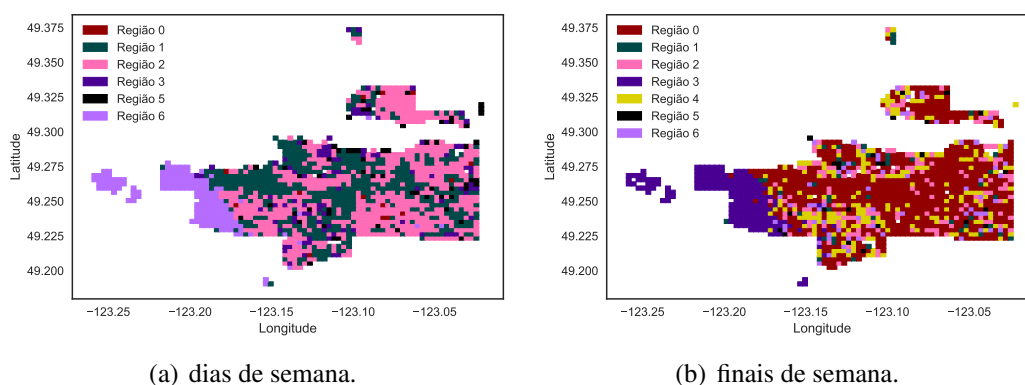


Figura 3. Regiões de atuação por grupo para dias de semana e fins de semana.

Para uma visualização mais clara dos grupos no mapa, os pontos correspondentes a cada grupo foram discretizados de acordo com o grupo de maior predominância em dada região. Foi montada uma malha dividida em quadrantes de 200x200 metros sobre a cidade. Após isso, foi identificado, em cada quadrante, qual grupo teve mais ocorrências de viagens iniciadas dentro da área delimitada pelo quadrante. Assim, definiu-se que a região do quadrante trata-se de uma área de atuação do grupo identificado. A Figura 3, apresenta as regiões discretizadas pelo método anterior, onde a região associada ao “Grupo 0” é identificada por “Região 0”, a região associada ao “Grupo 1” é identificada por “Região 1” e assim sucessivamente. Nesse processo de discretização, o Grupo 4, que deveria ser representado pela Região 4 foi eliminado. Isso aconteceu porque em nenhum quadrante, o Grupo 4 teve uma quantidade predominante de viagens.

4. Resultados

Nesta seção, é apresentada uma caracterização dos dados que apoiam o Módulo de Demanda Offline (Seção 2). A partir da análise dos grupos foi possível identificar padrões

de demanda espaço-temporais. Todas as análises e informações levantadas poderão ser utilizadas para a criação de modelos de previsão e também para a alimentação de um Módulo de Otimização Online.

A Figura 3-a, apresenta três grandes regiões e outras duas mais dispersas pelo mapa. Além disso, nota-se que o grupo 4 desaparece, pois, durante o processo de discretização ela não obteve maioria de ocorrências em nenhum quadrante analisado. A Figura 3-b, mostra que aos finais de semana, as regiões tendem a ser mais dispersas, apresentando somente duas regiões majoritárias e, nesse caso sem perda de nenhum grupo pelo processo de discretização.

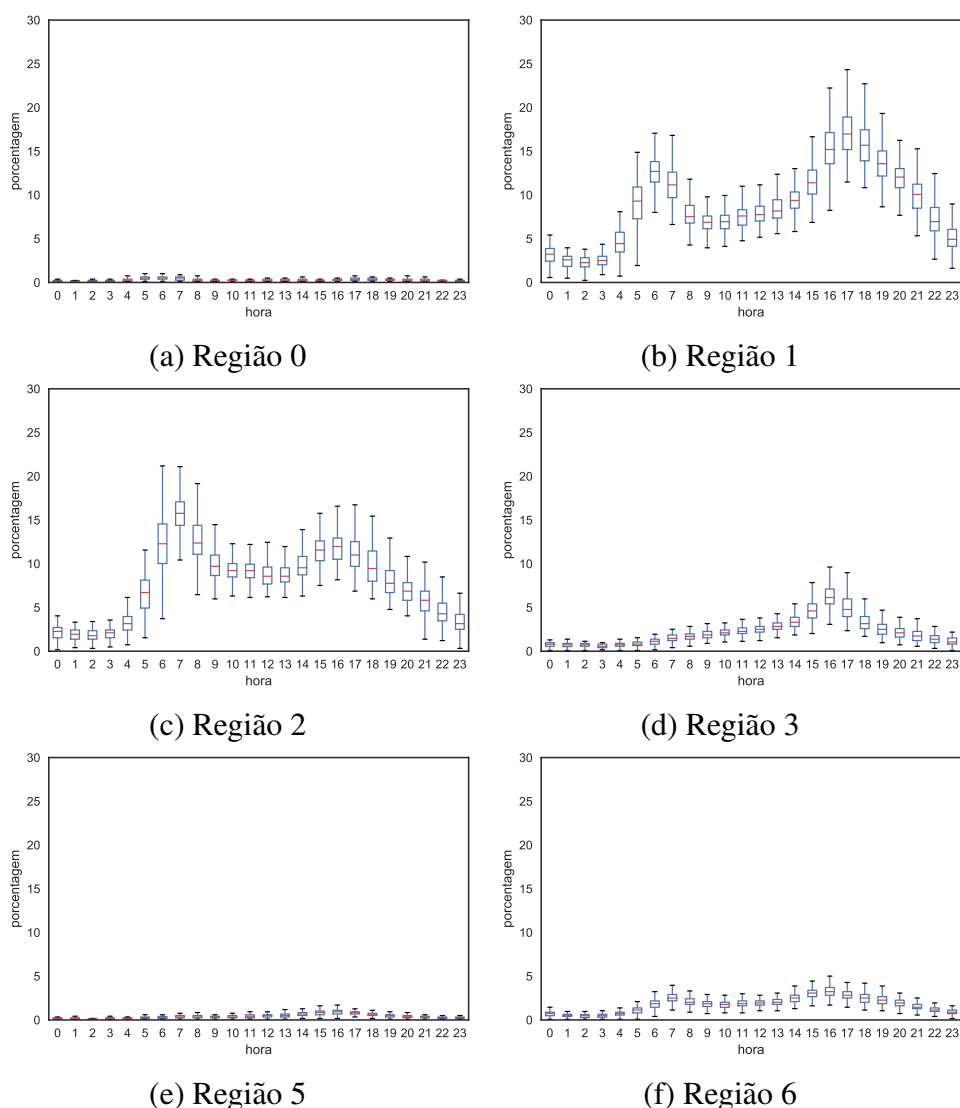
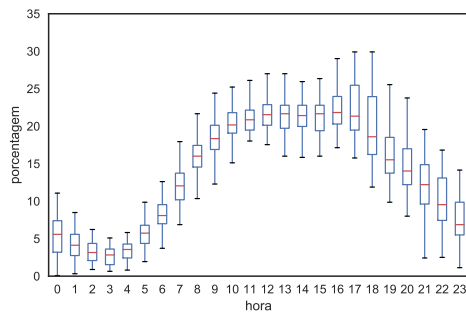
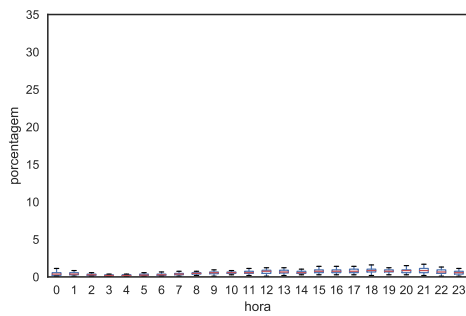


Figura 4. Percentual de veículos ocupados em dada hora para cada região em relação a área total observada, em dias de semana.

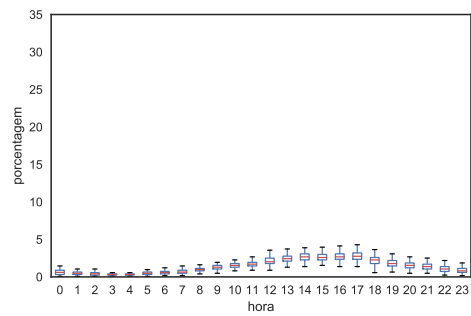
As Figuras 4 e 5 apresentam os aspectos temporais da demanda dos veículos do serviço dentro de cada região. Elas apresentam a porcentagem de ocupação de veículos a cada hora em relação ao todo. Tal abordagem contribui com informações sobre padrões de demanda de veículos ao longo do dia e picos de utilização. A Figura 4 mostra que



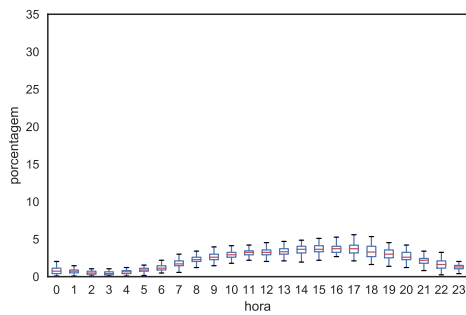
(a) Região 0



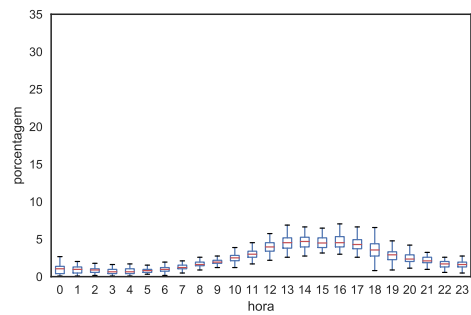
(b) Região 1



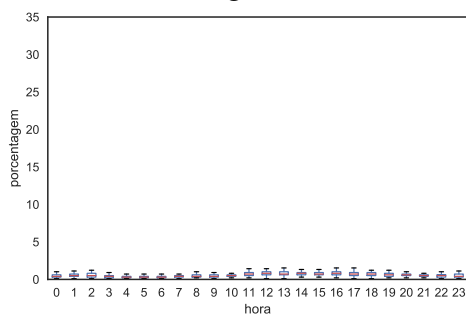
(c) Região 2



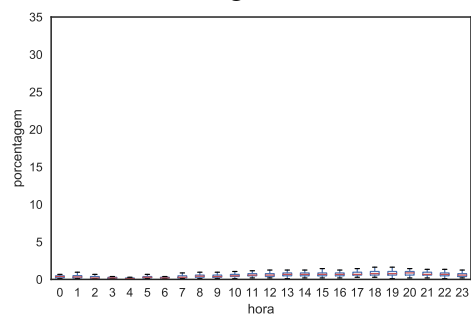
(d) Região 3



(e) Região 4



(f) Região 5



(g) Região 6

Figura 5. Percentual de veículos ocupados em dada hora para cada região em fins de semana.

as regiões 1 e 2 apresentam picos em horários coincidentes a rotinas laborais em dias de semana. No entanto a região 1 detém um pico mais significativo no período da tarde e a região 2 no período da manhã. A Figura 5 mostra que a utilização dos finais de semana tem um comportamento crescente ao longo do dia, com um pico de utilização que se

prolonga durante todo o período da tarde – como pode ser visto na região 1. Também é possível notar que a porcentagem de ocupação tem uma relação proporcional ao tamanho de cada região, tanto para dias de semana quanto fins de semana. De fato, quanto menor a região, menor é a porcentagem do total de veículos do serviço ocupados nela.

A Figura 6 apresenta a duração das viagens iniciadas em cada região. Na Figura 6-a, percebe-se que todas as regiões seguem um padrão semelhante de tempo de ocupação, onde 50% das viagens detêm menos de 30 minutos de duração. Um padrão semelhante também pode ser observado na Figura 6, que representa os fins de semana. Assim, nota-se que o padrão de duração de viagens nas regiões definidas são homogêneos e que suas viagens são majoritariamente, de curta a média duração.

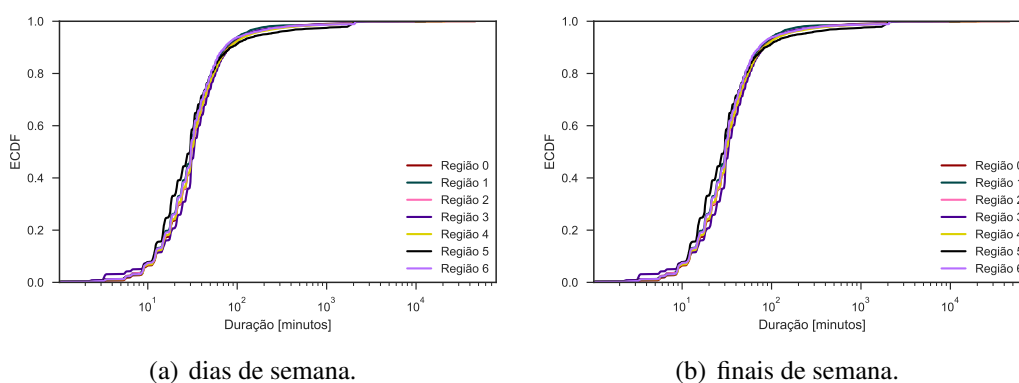


Figura 6. Função de Distribuição Acumulada do tempo de ocupação de veículos.

Além da ocupação, a ociosidade dos veículos dentro de cada região também foi analisada. Compreender o padrão de ociosidade de cada região a auxilia na descoberta da dinâmica de alocação dos veículos. Em outras palavras, uma maior ociosidade de veículos em regiões, cujo padrão de ociosidade é baixo, é um indicativo da necessidade de realocação dos veículos dessas regiões. Portanto, foram geradas as Figuras 7 e 8. A Figura 7, apresenta, como esperado, picos inversos a ocupação (Figura 4) nas regiões de maior movimentação, além disso, percebe-se que a taxa de ociosidade ao longo do dia é alta. Na Figura 8, nota-se, novamente, a predominância de uma região com elevadas quantidades de veículos ociosos ao longo do dia.

Os tempos de ociosidade dos veículos de cada região são apresentados na Figura 9. Para ambas figuras, o comportamento entre regiões é semelhante. A Figura 9-a, mostra que 60% dos veículos ficam até 1 hora e 40 minutos ociosos. Um comportamento semelhante acontece na Figura 9-b, que representa finais de semana. Menos de 10% dos casos tem tempos de ociosidade maiores que 24 horas. Esses valores também podem ser utilizados para a identificação de anormalidades e serem transformados em gatilhos para algoritmos de realocação. Como exemplo, tem-se a situação em que mais de 10% dos veículos de determinada região se encontram ociosos por mais de um dia, indicando uma anormalidade e uma possível necessidade de realocação.

A movimentação entre regiões é apresentada na Figura 10. Esta figura apresenta uma matriz de origem-destino das regiões anteriormente estabelecidas. Os dados foram normalizados em uma escala de 0 a 1, utilizando o método *min-max*. O entendimento da dinâmica entre as regiões oferece suporte a possíveis estratégias de realocação.

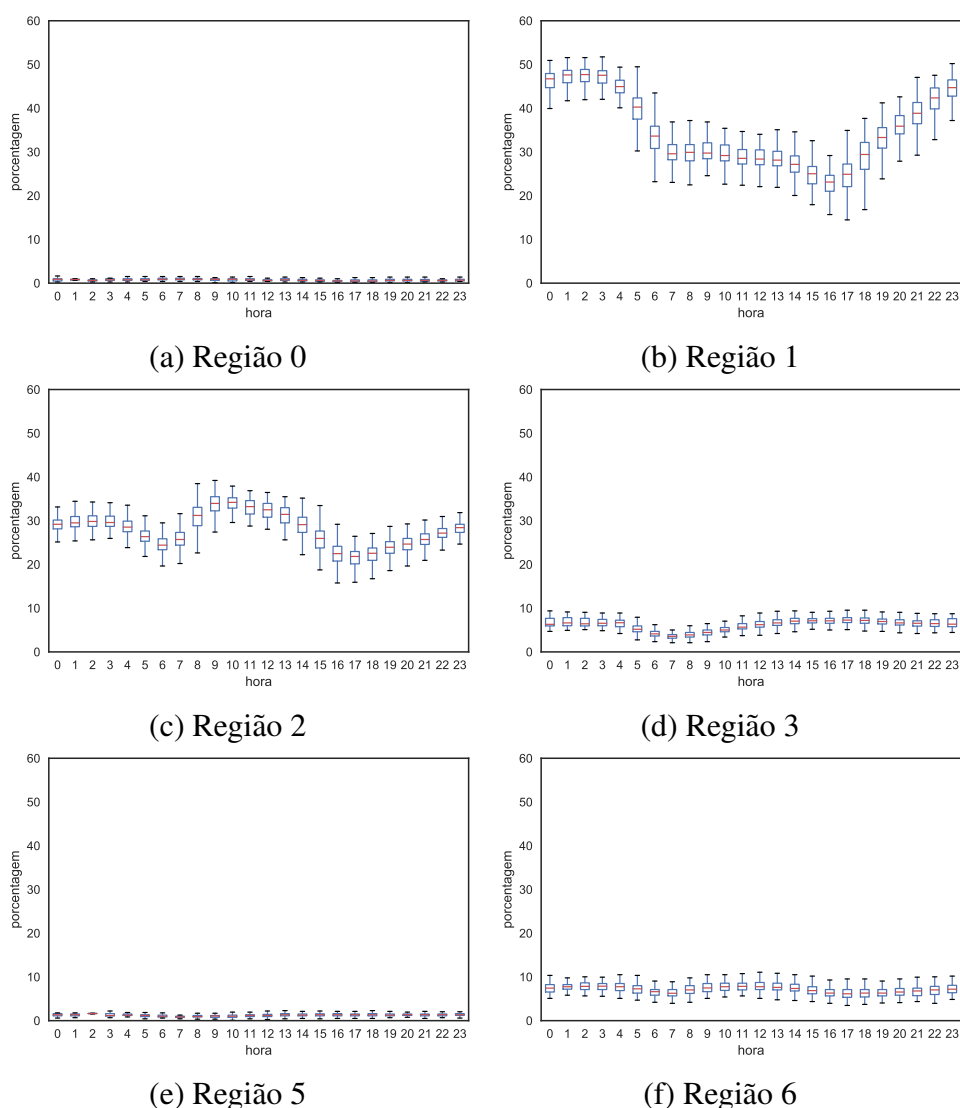
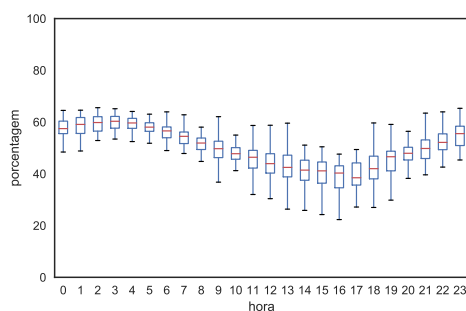


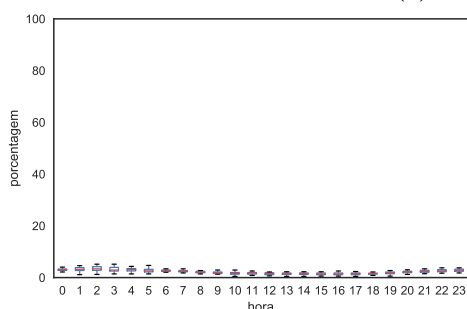
Figura 7. Percentual de veículos ociosos em dada hora para cada região em dias de semana.

Para dias de semana, a Figura 10-a, mostra um forte movimento de saída da região 1 e 2 comparado as demais, contudo não há grandes movimentações de retorno. Para fins de semana, a Figura 10-b, apresenta uma única região com maior movimentação de saída. Ainda, a Região 3, detém uma grande movimentação sobre si mesma, assim, novamente tem-se grandes movimentações de saída de determinada região, porém sem grandes fluxos de retorno. Além disso, é importante notar que há grandes fluxos dentro das regiões para dias de semana e pouco para fins de semana. Tal comportamento aos fins de semana pode gerar problemas de alocação dos veículos sobre o início dos dias de semana.

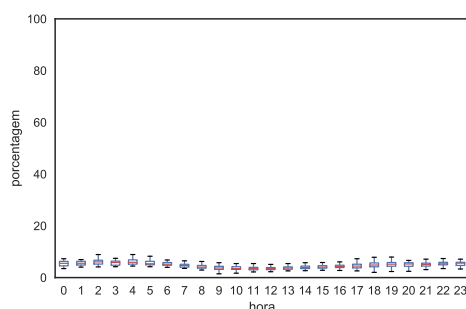
As distâncias percorridas nas viagens realizadas são apresentadas na Figura 11. É importante ressaltar que, para o cálculo das distâncias, foi utilizado a distância geodésica entre os pontos de início e fim das viagens. Essa distância representa um limite inferior da distância real percorrida. Não é possível conhecer a distância real, uma vez que o serviço não disponibiliza os trajetos dos veículos alugados. A Figura 11-a, mostra que em 60% dos casos, as regiões 2, 3 e 5 seguem um padrão similar, onde os trajetos apresentam até



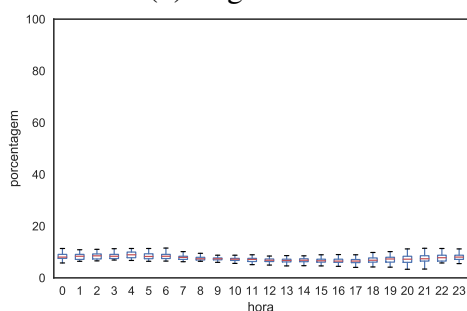
(a) Região 0



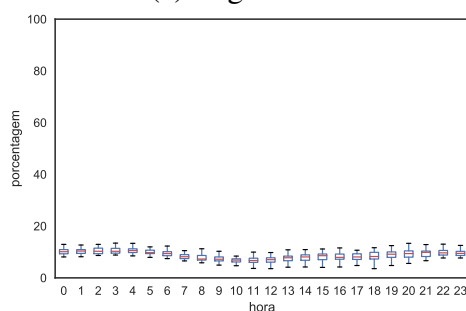
(b) Região 1



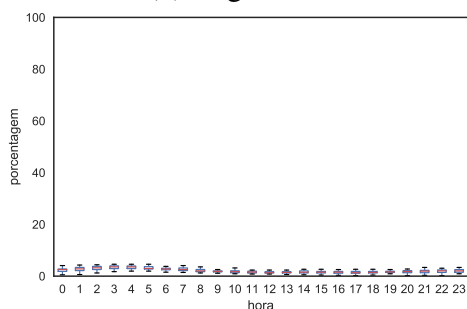
(c) Região 2



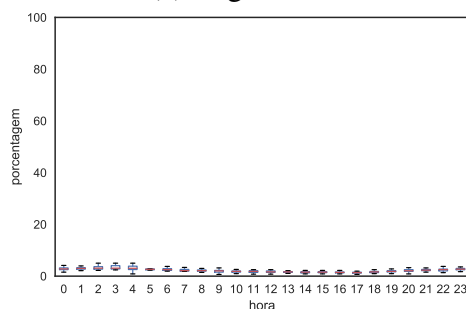
(d) Região 3



(e) Região 4



(f) Região 5



(g) Região 6

Figura 8. Percentual de veículos ociosos em dada hora para cada região em fins de semana.

3 Km. A região 0 e 6 começam a destoar das demais aos 30%. Notadamente, apresentam trajetos mais longos que as demais. Por exemplo, em 60% dos casos a região 0 apresenta viagens com trajetos de até 4 Km enquanto na região 6, esse numero cresce para até 6 Km. Na Figura 11-b observa-se que as regiões seguem um padrão semelhante, onde 60% dos viagens tem trajetos de até 3 Km. A Região 3, nesse caso, apresenta viagens mais longas

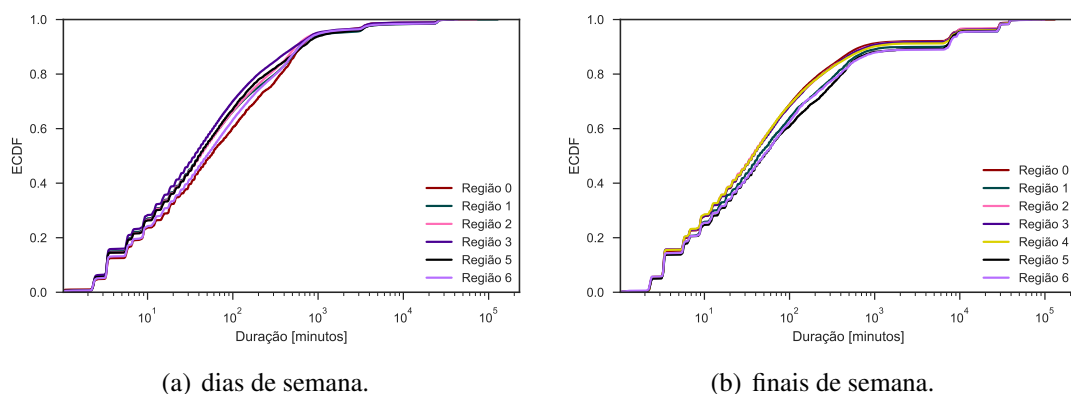


Figura 9. Função de Distribuição Acumulada do tempo de ociosidade de veículos.

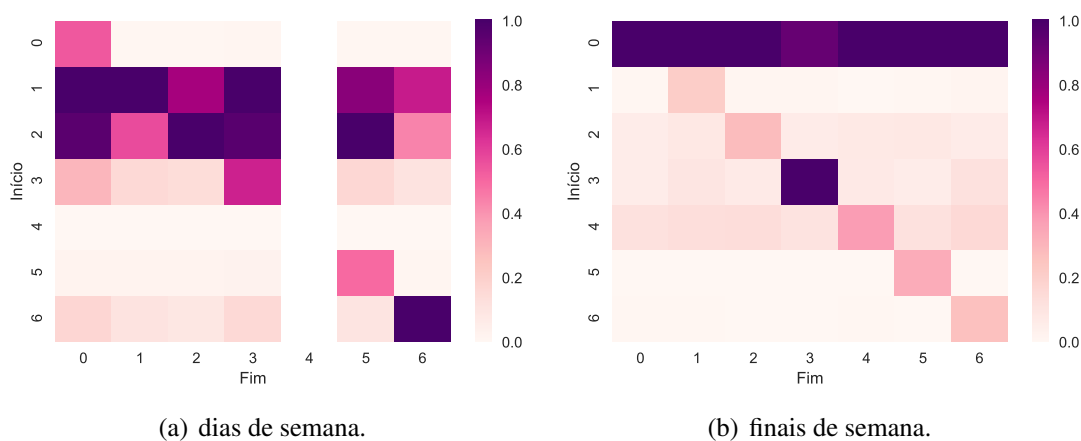


Figura 10. Matriz de origem-destino das regiões.

que as demais. Por exemplo, 60% dos viagens tem trajetos de até 5 Km. Por fim, é interessante observar que tanto a Região 6, nos dias de semana, quanto a Região 5, aos finais de semana, tem comportamentos semelhantes. Conjectura-se que a característica de utilização para viagens mais longas, possa estar associada ao público universitário, característico das áreas compreendidas por essas regiões.

5. Discussão e Trabalhos Futuros

A partir das caracterizações apresentadas na seção anterior é possível reafirmar o problema de realocação bem como criar estratégias visando melhor direcionamento das realocações através de cada região, dia da semana e horário.

Em um primeiro momento, a partir das observações, uma solução plausível seria incluir operadores nas principais regiões em momentos de ociosidade de veículos que antecedam os horários de pico. Esses operadores seriam responsáveis por transportar os veículos entre regiões de forma que o sistema se mantenha estável em relação a demanda e oferta. Assim, os operadores deveriam ser munidos de um *dashboard* que contemple, em tempo real, os dados da caracterização. Todavia, a intervenção em um sistema de compartilhamento de carros é sempre acompanhada de custos. Portanto, a estratégia baseada

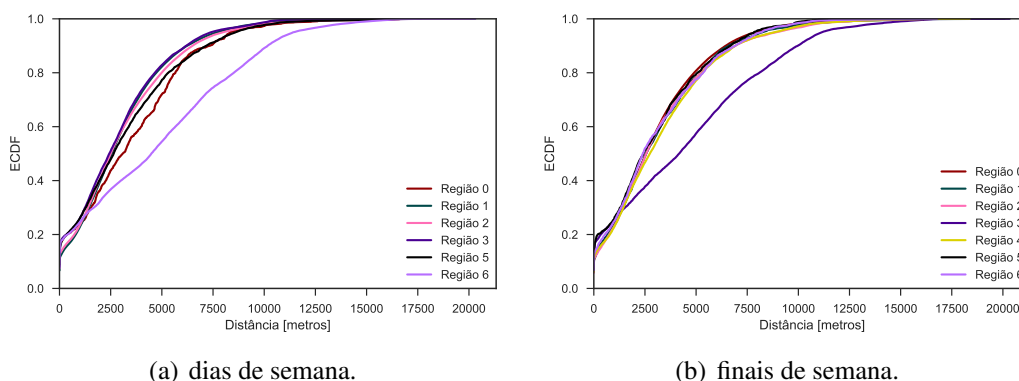


Figura 11. Função de Distribuição Acumulada da distância percorrida pelos veículos.

em operadores deve ser cautelosa. Para esse cenário, vislumbra-se que os serviços de compartilhamento de carro, poderiam também ofertar serviços de transporte privado para pontos específicos. Ou seja, operadores poderiam trabalhar como motoristas particulares em pontos específicos amenizando, através desses ganhos, o custo de mão de obra com recursos humanos. Essa ideia, pode servir como um agente de *marketing* para o serviço, englobando uma nova parcela de interessados.

Visando uma realocação, na maioria das vezes mais econômica, cogita-se a utilização de mão de obra do próprio usuário. Assim, vale-se apoiar em estratégias de *gamificação* onde o usuário recebe recompensas por prestar serviços. De acordo com a caracterização apresentada seria interessante oferecer descontos em locação de veículos àqueles usuários que estejam em regiões propensas a terem sobrecarga. Portanto, esses usuários, seriam notificados a partir de seus *smartphones* sobre descontos para transportar um carro de uma região A para uma região B. Neste cenário, obter informações de contexto do usuário a partir de outros serviços, tal qual o Google Maps se mostra uma estratégia interessante visando ofertas mais assertivas. Isto é, o usuário X deverá receber propostas de deslocamento, diferentes do usuário Y baseado nos seus trajetos cotidianos e localização atual. Comparada à abordagem baseada em operadores, vale-se, inclusive, a oferta de corridas gratuitas para os pontos críticos.

Uma possível implementação das concepções anteriores, deverá utilizar as restrições demonstradas na caracterização e tomar decisões com relação a uma estratégia baseada no operador, usuário ou híbrida. Portanto, o algoritmo deverá receber como parâmetros o estado de cada região – outros serviços também podem ser utilizados, por exemplo, APIs de previsão do tempo – com quantificadores, o horário, o dia da semana, entre outros. O algoritmo retornará que realocações devem ser realizadas para atender a demanda recebida. Com esse retorno o sistema irá tomar decisão e contactar os usuários através de notificações (inteligentes), oferecendo trabalhos e/ou recompensas.

6. Conclusão

A mobilidade urbana se torna cada vez mais um corpo de pesquisa interessante em diversas áreas do conhecimento. Todavia, a aquisição de dados atuais e acurados ainda é um desafio neste cenário, dificultando a criação de propostas mais assertivas. Os sis-

temas de compartilhamento de veículos representam um tópico interessante relacionado à mobilidade urbana. Entretanto, devido a uma demanda de saída elevada de veículos em determinados locais, e a uma entrada não equivalente, possivelmente menor, alguns locais podem ficar desabastecidos. Por outro lado, outras localizações cuja demanda de saída não é elevada, podem ficar com veículos em excesso. Neste trabalho foram apresentados aspectos relacionados ao problema de realocação de veículos, bem como a caracterização do *Evo*, um serviço de uma via híbrido. Nessa caracterização foram discretizados parâmetros importantes que servirão de apoio a tomada de decisão em propostas de soluções do problema. Finalmente, foram apresentadas estratégias e trabalhos futuros visando posteriores implementações. Assim, este trabalho representa um importante passo para o entendimento e evolução de sistemas de compartilhamento de veículos, fornecendo um apanhado de informações que podem ser utilizadas para criação de modelos e *Frameworks* genéricos relacionado a esse tipo de sistema.

Referências

- Becker, H., Ciari, F., and Axhausen, K. W. (2017). Comparing car-sharing schemes in switzerland: User groups and usage patterns. *Transportation Research Part A: Policy and Practice*, 97:17–29.
- Boldrini, C., Bruno, R., and Conti, M. (2016). Characterising demand and usage patterns in a large station-based car sharing system. In *Computer Communications Workshops (INFOCOM WKSHPS), 2016 IEEE Conference on*, pages 572–577. IEEE.
- Ciociola, A., Cocca, M., Giordano, D., Mellia, M., Morichetta, A., Putina, A., and Salutari, F. (2017). Umap: Urban mobility analysis platform to harvest car sharing data. In *2017 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computed, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCOM/IOP/SCI)*, pages 1–8.
- Herrera, J. C., Work, D. B., Herring, R., Ban, X. J., Jacobson, Q., and Bayen, A. M. (2010). Evaluation of traffic data obtained via gps-enabled mobile phones: The mobile century field experiment. *Transportation Research Part C: Emerging Technologies*, 18(4):568–583.
- Kek, A. G., Cheu, R. L., Meng, Q., and Fung, C. H. (2009). A decision support system for vehicle relocation operations in carsharing systems. *Transportation Research Part E: Logistics and Transportation Review*, 45(1):149–158.
- Ma, S., Zheng, Y., and Wolfson, O. (2013). T-share: A large-scale dynamic taxi ridesharing service. In *Data Engineering (ICDE), 2013 IEEE 29th International Conference on*, pages 410–421.
- Nourinejad, M. (2014). Dynamic optimization models for ridesharing and carsharing. Master’s thesis, University of Toronto.
- Shaheen, S. A. (2016). Mobility and the sharing economy. *Transport Policy*, 51(Supplement C):141 – 142.
- Weikl, S. and Bogenberger, K. (2013). Relocation strategies and algorithms for free-floating car sharing systems. *IEEE Intelligent Transportation Systems Magazine*, 5(4):100–111.

Speed: Uma solução inter-veicular para detectar e controlar vias congestionadas no TMS

Geraldo P. Rocha Filho¹, Rodolfo I. Meneguette², Gustavo Pessin³,
Jó Ueyama⁴, Leandro A. Villas⁵

¹Departamento de Ciência da Computação, Universidade de Brasília – UnB

²Instituto Federal de São Paulo – IFSP

³Instituto Tecnológico Vale – ITV

⁴Instituto de Ciência Matemáticas e de Computação, Universidade de São Paulo – USP

⁵Instituto de Computação – Universidade Estadual de Campinas – UNICAMP

geraldof@unb.br, meneguette@ifsp.edu.br, gustavo.pessin@itv.org

joueyama@icmc.usp.br, leandro@ic.unicamp.br

Abstract. *This work proposes a solution with inter-vehicular communication to estimate and maximize the congestion in TMS, named Speed. For this, Speed is modeled based on an ensemble of classifiers to increase reliability when classifying vehicle traffic. Based on a communication mechanism, such classification is used to carry out the dissemination of information. When compared with other works of the literature, Speed showed to advance in the state of the art by having an increase in the precision of congestion levels classification while reducing: (i) the emission of CO₂ by 3.37%; (ii) fuel consumption by 3%; and (iii) travel time by 1.02 times faster than the other works.*

Resumo. *Este artigo propõe uma solução com comunicação inter-veicular para estimar e maximizar o congestionamento no TMS, nomeado Speed. Para isso, o Speed é modelado com base em um comitê de classificadores que tem como objetivo aumentar a confiabilidade no momento da classificação do tráfego de veículos. Com base em um mecanismo de comunicação, tal classificação é utilizada para realizar a disseminação da informação. Ao comparar o Speed com outros trabalhos da literatura, a nossa solução mostrou avançar o estado da arte por possuir um aumento nos acertos dos níveis de congestionamento ao mesmo tempo que reduz: (i) a emissão de CO₂ em 3,37%; (ii) o consumo de combustível em 3%; e (iii) o tempo de viagem em 1.02 vezes.*

1. Introdução

Atualmente, observa-se um rápido aumento do número de veículos que trafegam nas grandes metrópoles. Esse aumento, atribuído ao crescimento populacional e aos cidadãos que usam seus próprios veículos como meio de locomoção [Junior et al. 2015], ocasiona diversos problemas no sistema de transporte, sendo o congestionamento um dos fatores a ser destacado. Ainda por causa desse crescimento, também surgem outros problemas que atingem o trânsito [Meneguette et al. 2018], como o desvio de tráfego e

a falta de segurança dos condutores devido a um evento na via urbana, por exemplo. Tais problemas atingem gravemente a economia quando vista como um todo. No Brasil, o custo estimado do congestionamento nas cidades ultrapassa os R\$ 80 bilhões por ano [Cintra 2013]. Já na União Européia é de aproximadamente 2% do seu Produto Interno Bruto (PIB) [H. Allen and Stonehill 2013] e nos EUA o custo é de mais de US\$ 160 bilhões [Texas Transportation Institute 2015].

Salienta-se que o problema de congestionamento nas cidades não é uma tarefa trivial de ser resolvida. Apesar da ampliação da rede de transporte auxiliar no problema do congestionamento, essa expansão pode ser inviável devido ao elevado custo financeiro, às restrições ambientais e geográficas impostas, bem como ao tempo elevado para finalizar a melhoria da infraestrutura de transporte. Tais limitações podem ser minimizadas pelo Sistema de Gerenciamento de Tráfego [Souza et al. 2017, Meneguette et al. 2018, Cunha et al. 2018] (TMS, do inglês *Traffic Management Systems*) que surge como uma alternativa promissora para auxiliar no problema do congestionamento nas cidades. O TMS pode ser definido como um tipo especial de serviço das Redes Ad-Hoc Veiculares (VANETs, do inglês *Veicular Ad-Hoc Networks*) munido com tecnologias de comunicação, detecção e processamento para coletar dados do tráfego com o intuito de lidar com a sua melhoria, não apenas em relação a determinados veículos, mas também no que tange todo o sistema de transporte da cidade.

Diferentes soluções foram propostas para lidar com o problema do congestionamento de tráfego nas cidades. Algumas soluções sugerem as melhores rotas individuais para evitar o congestionamento [Doolan and Muntean 2013, Doolan and Muntean 2017], o que pode gerar novos congestionamentos em outras áreas. Outras soluções têm se limitado em detectar congestionamento [Wongcharoen and Senivongse 2016, Nottle et al. 2017] ou redirecionar veículos [Pan et al. 2012, Pan et al. 2017] ou propostas para cenários específicos [Meneguette et al. 2015, Severino et al. 2018] como urbano ou rodoviário, resolvendo apenas uma parte do problema. Ainda há trabalhos que modelam algoritmos de aprendizagem de máquina para estimar o congestionamento da via [Araujo et al. 2014, Souza et al. 2015, Meneguette et al. 2016]. Entretanto, tal modelagem deve lidar com uma grande quantidade de dados obtidos de maneira implícita providos do TMS para monitorar o tráfego.

Ainda que haja um esforço crescente conquistado no TMS, aumentar a precisão de accertos das vias congestionadas para maximizar o fluxo de veículos na infraestrutura de transporte, traz uma nova questão de pesquisa, a saber: Como prover uma maior precisão para estimar o nível de congestionamento da via, maximizando o fluxo de veículos na infraestrutura de transporte? Tal questão é uma nova problemática que está sendo abordada na literatura e que este artigo investiga.

Este trabalho apresenta o Speed, uma solução inter-veicular para estimar o congestionamento e maximizar o fluxo de tráfego de veículos no sistema de transporte. O Speed é baseado em um comitê de classificadores que tem como objetivo aumentar a confiabilidade no momento da classificação do tráfego de veículos. Com isso, tal classificação é utilizada para realizar a disseminação da informação com base em um módulo de comunicação *Publish/Subscribe* que, além de encaminhar as mensagens para os grupos de interesse, evita o problema de *broadcast storms*. Para certificar a eficiência do Speed com relação ao problema de congestionamento, a solução proposta foi com-

parada com outras soluções consagradas na literatura que mais assemelham a esta pesquisa [Araujo et al. 2014, Souza et al. 2015, Meneguette et al. 2016]. Os resultados obtidos por meio de simulações, evidenciam um aumento nos acertos dos níveis de congestionamento com redução no tempo de viagem dos condutores, no consumo de combustível dos veículos e na emissão de CO_2 .

O restante deste artigo está organizado da seguinte forma. Na Seção 2, são apresentados os principais trabalhos relacionados a detecção de congestionamento de veículos no TMS. A Seção 3 apresenta a modelagem do Speed, enquanto que sua avaliação de desempenho com os resultados obtidos são descritos na Seção 4. Por fim, na Seção 5, são apresentadas as conclusões e trabalhos futuros.

2. Trabalhos relacionados

Diversas soluções foram propostas para lidar com o problema de congestionamento nos últimos anos. Além de trazer maior conforto e segurança para os condutores [Pessin et al. 2014, Meneguette et al. 2018], as soluções propostas tem como objetivo otimizar o tráfego de veículos no sistema de transporte, reduzindo o tempo de viagem, o consumo de combustível e a quantidade de CO_2 emitidos na atmosfera. Entretanto, modelar um mecanismo para estimar o nível de congestionamento da via com um aumento nos acertos das vias congestionadas, e com um tempo aceitável para alertar os condutores não é uma tarefa trivial.

As soluções apresentadas em [Bauza et al. 2010, Bauza and Gozávez 2013, Araujo et al. 2014], propõem serviços de gerenciamento de tráfego para detectar o congestionamento da via com base na lógica fuzzy. Baseado nos trabalhos [Bauza et al. 2010, Bauza and Gozávez 2013], os autores [Araujo et al. 2014] propõem o CARTIM, um protocolo para detectar e reduzir o congestionamento de veículos no sistema de transporte. Para isso, o CARTIM utiliza como entrada para a lógica fuzzy a velocidade e a densidade dos veículos na via para classificar o nível de congestionamento localmente, disseminando-o de maneira cooperativa entre os veículos. Em seguida, caso a via esteja congestionada, o CARTIM adota uma heurística para alterar as rotas dos veículos. Embora o tempo de inferência da lógica fuzzy seja rápido o suficiente para alertar os motoristas, a sua precisão para detectar o nível de congestionamento depende dos valores de entrada. Ainda, como os veículos não possuem conhecimento global da condição do tráfego, novas áreas podem ficar congestionadas, ou a heurística de sugestão de rotas pode indicar rotas inapropriadas.

Em [Souza et al. 2015], é proposto o CO-OP, uma solução para prevenção e controle de congestionamentos com base em um roteamento cooperativo. O CO-OP modela um mecanismo de classificação para detectar diferentes níveis de congestionamento na via utilizando *K-Nearest Neighbor* (KNN). Para isso, a densidade da rua combinada com a velocidade média da via é utilizada como parâmetro de entrada para classificar a condição do tráfego. A informação da classificação é disseminada para os veículos que estão dentro de uma área de roteamento. Tal área, é utilizada para evitar roteamentos desnecessários para outros veículos. Com isso, apenas os veículos que possuem maior probabilidade de congestionamento recebem a informação. O KNN apresenta um tempo rápido de inferência devido a sua baixa complexidade quando comparado com a lógica fuzzy. Entretanto, quando comparado com os demais mecanismos de classificação, o

KNN possui menor precisão para detectar a condição do tráfego, como constatado nesta pesquisa.

Outra solução que se assemelha com esta pesquisa é apresentada em [Meneguette et al. 2016]. Os autores [Meneguette et al. 2016] propõem o IN-CIDEnt, um protocolo de detecção, disseminação e controle de congestionamento para ambientes urbanos e rodoviários. Diferente do CARTIM e CO-OP, o INCIDENT é baseado em uma Rede Neural Artificial do tipo *MultiLayer Perceptron* (RNA-MLP), a qual é modelada para detectar e classificar os níveis de congestionamentos nas vias. Ainda, a classificação por meio da RNA-MLP é utilizada para realizar a disseminação da informação na rede, além de ser aplicada no mecanismo de sugestão de novas rotas para evitar que os motoristas passem em vias congestionadas. Embora tal mecanismo de classificação permita um auto aprendizado, a configuração pode afetar no desempenho do sistema. Com isso, a maneira como a RNA-MLP é modelada poderá impactar no tempo de decisão que é levado em conta para realizar a sugestão da rota, bem como na precisão da classificação da condição do tráfego.

Diferentemente dos trabalhos da literatura que apresentam um único método de classificação, dos quais a precisão para detectar o congestionamento dependerá da configuração do método, neste trabalho será modelado um comitê de classificadores que busca maximizar a precisão na classificação sem impactar o tempo de decisão da rota. O uso do comitê visa o aperfeiçoamento da taxa de acerto da classificação, além de permitir uma maior adaptação de novos eventos e condições que possam ocorrer. Ainda, a solução proposta permite a classificação dos níveis de congestionamento de tráfego de veículos sem uma sobrecarga de informações de controle na rede, evitando assim colisões de informações de controle, dando maior vazão nas informações. Salienta-se que o mecanismo de classificação proposto é uma ferramenta fundamental que é utilizada para a realização da disseminação da informação na rede.

3. Uma Solução inter-veicular para detectar e controlar vias congestionadas

Esta seção apresenta o Speed, uma solução inter-veicular para estimar o congestionamento e maximizar o fluxo de tráfegos de veículos nas vias. O Speed é composto por um comitê de classificadores para classificar e detectar o nível de congestionamento localmente, além de aumentar a precisão de acertos das vias congestionadas. Ainda, implementou-se no Speed um módulo de comunicação *Publish/Subscribe* (Pub/Sub) capaz de disseminar o conteúdo de acordo com o nível de congestionamento da via, o que evita o problema de *broadcast storms*. Além disso, um mecanismo de planejamento de novas rotas para maximizar o fluxo de tráfegos da via foi modelado no Speed. O Speed tem como objetivo principal prover maior precisão para estimar o nível de congestionamento da via, tendo como meta específica maximizar o fluxo de tráfego dos veículos no sistema de transporte.

No Speed, assume-se que cada veículo é equipado com uma On-Board Unit (OBU) para obter sua velocidade e localização. A Figura 1 sumariza o funcionamento do Speed, o qual detecta o congestionamento e em seguida realiza a troca de rota para maximizar o fluxo de tráfego no sistema de transporte. Para tanto, o veículo que detectou o congestionamento (Rótulo 1, Figura 1), inicia uma disseminação de aviso de congestionamento para notificar os motoristas. Tal aviso chega para o motorista distante (Rótulo

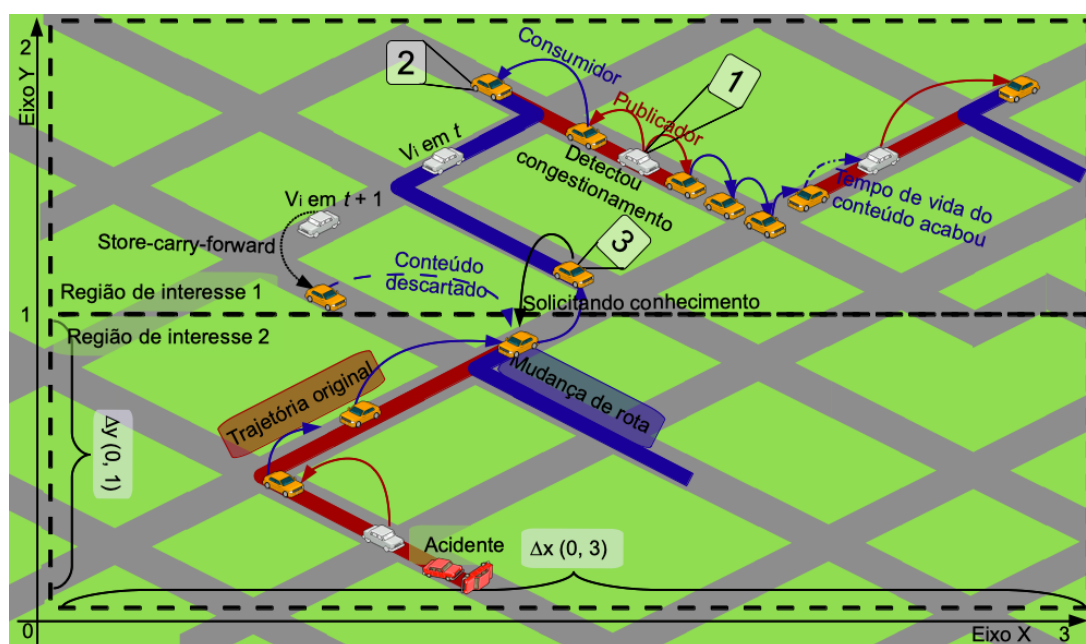


Figura 1. Funcionamento da solução inter-veicular para maximizar o fluxo de tráfegos no sistema de transporte.

2, Figura 1) que ainda não entrou na via congestionada. Com isso, é possível recalcular um novo plano de rota (Rótulo 3, Figura 1) para os veículos evitarem a via congestionada. Para um melhor entendimento do Speed, a solução proposta é detalhada em quatro etapas: (i) modelo de dados para estimar o congestionamento; (ii) mecanismo para detectar congestionamento; (iii) mecanismo para disseminar a informação do congestionamento; e (iv) mecanismo de recomendação de rotas.

3.1. Modelo de dados para estimar o congestionamento

Estimar o nível de congestionamento em uma via não é processo trivial [Bauza et al. 2010, Rao and Rao 2012], uma vez que é necessário correlacionar e lidar com métricas qualitativas (acidentes, infraestrutura da via e eventos especiais como reuniões sociais em massa) e quantitativas (velocidade e tempo de viagem). Tal estimativa ainda é mais problemática, quando se trata de dados obtidos de maneira implícita providos do TMS. Para este trabalho, foi adotado o modelo de classificação estendido do Skyscomp [SKYCOMP 2008], o qual foi aplicado em [Bauza et al. 2010, Araujo et al. 2014, Souza et al. 2015, Meneguette et al. 2016]. O Skyscomp fornece um *Level-Of-Service F* (LOS), o qual representa uma métrica para classificar o fluxo de tráfego de uma via, definida pelo *Highway Capacity Manual* (HCM) [Board 2000].

O modelo de classificação adotado na *LOS F*, considera como atributo a velocidade do veículo e a densidade de veículos na via para fornecer o nível do congestionamento do tráfego, como apresentado na Tabela 1. Nesse caso, foram estabelecidos três níveis de congestionamento, sendo eles: (i) livre; (ii) moderado; e (iii) congestionado. Fundamentado em [Meneguette et al. 2016], o Speed também obtém a velocidade do veículo por meio da OBU. Já a densidade do veículo é obtida pela quantidade de

veículos vizinhos que estão em sua área de cobertura. A seguir, é apresentado o mecanismo para detectar o nível de congestionamento da via.

Tabela 1. Relação entre a densidade da via e velocidade do veículo para estimar o nível de congestionamento [Bauza et al. 2010].

Nível de congestionamento	densidade	velocidade
Livre	[29-37] vel/km/ln	[48-81] km/h
Moderado	[37-50] vel/km/ln	[24-64] km/h
Congestionado	≥ 50 vel/km/ln	[0-40] km/h

3.2. Mecanismo para detectar congestionamento

O mecanismo de detecção tem por objetivo estimar o nível de congestionamento da via. Há diversos trabalhos que utilizam algoritmos de classificação para detectar o nível de congestionamento. Particularmente em [Araujo et al. 2014, Souza et al. 2015, Meneguette et al. 2016], são utilizados a lógica fuzzy, o *K-Nearest Neighbors* (KNN) e a Rede Neural Artificial do tipo *MultiLayer Perceptron* (RNA-MLP), respectivamente. Entretanto, o uso de classificadores individuais além de depender de ajustes nos parâmetros para a classificação, possuem erros característicos que podem resultar em classificações errôneas, dependendo do problema ou do *dataset* utilizado [Canuto 2001, Santos 2008].

Para tratar tal limitação, o mecanismo proposto para estimar o nível de congestionamento nas vias no Speed foi modelado por meio de um comitê de classificadores. Para facilitar o entendimento do comitê no Speed, a Figura 2 apresenta o seu fluxo de funcionamento. O comitê foi modelado com base nos algoritmos de classificação adotados em [Araujo et al. 2014, Souza et al. 2015, Meneguette et al. 2016], sendo eles, respectivamente: (i) Lógica fuzzy; (ii) KNN; e (iv) RNA-MLP. O comitê utiliza como atributo de entrada a velocidade do veículo e a densidade de veículos vizinhos e, mediante uma agregação de respostas dos algoritmos de classificação, estima o nível de congestionamento da via. A agregação da resposta ocorre por meio de um critério de votação dado pela fórmula: $\bar{W}x_i = \frac{APx_i}{\sum_j^n APx_j}$, onde \bar{W} são os pesos atribuídos para cada algoritmo x de acordo com sua taxa de precisão AP . Portanto, o comitê considera o mérito de cada x para aumentar a precisão nos acertos em estimar o nível de congestionamento da via, uma vez que não é persuadido por decisões errôneas de apenas um algoritmo.

A resposta final do comitê foi normalizada entre 0 e 1. Com isso, foram estabelecidos três níveis de congestionamento: (i) livre, entre o intervalo $[0, \frac{1}{3}]$; (ii) moderado, entre o intervalo $[\frac{1}{3}, \frac{2}{3}]$; e (iii) congestionado, entre o intervalo $[\frac{2}{3}, 1]$. Empiricamente, tal divisão possui uma convergência mais rápida para estimar o nível do congestionamento. A seguir, é apresentado o mecanismo para disseminar a informação que foi detectada na via.

3.3. Mecanismo para disseminar dados

Após detectar o nível de congestionamento, o Speed dissemina a informação entre os veículos. Vale salientar que as informações são entregues para múltiplos veículos que são consumidores de informações e não há conexões diretas com os produtores de informações (isto é, veículo que detectou o congestionamento). Por isso, o mecanismo

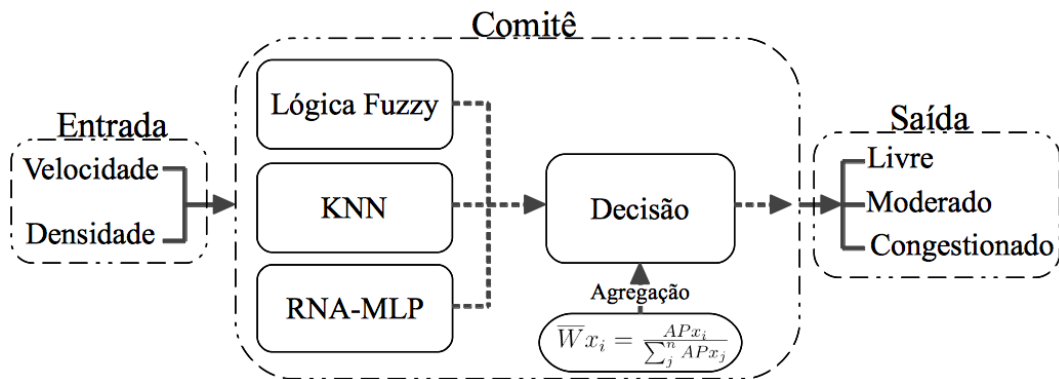


Figura 2. Funcionamento do comitê de classificadores modelado para estimar o nível de congestionamento.

de disseminação do Speed é modelado com base no paradigma Pub/Sub. O Pub/Sub permite a disseminação das informações entre os veículos de maneira assíncrona. Ainda, o Pub/Sub permite que a conexão entre os veículos seja fracamente acoplada, o que não traz dano em uma topologia dinâmica como as das TMSs, uma vez que tais veículos podem perder a conexão por diversos motivos.

Antes de iniciar o processo de publicação, assume-se que o mapa é dividido em regiões de interesse (RI), sendo que cada RI é independente uma da outra. Isso irá evitar o problema da *broadcast storm*. Entende-se por RI, como uma parte do mapa em que a informação relevante é propagada aos veículos que necessitam dela. Com isso, cada veículo tem conhecimento da sua RI (por exemplo, nível de congestionamento ou acidente). Para adquirir o conhecimento de um outra RI, um veículo pode tentar compartilhar dados com outros veículos que estão nas outras RIs por meio de um modelo *request-reply* via *Pub/Sub*. O RI pode ser ajustável dependendo da característica do ambiente, por exemplo, em um horário de pico em uma determinada via o RI pode ser menor para garantir a escalabilidade das TMSs. Entretanto, para esta solução, considerou-se o tamanho do mapa e as coordenadas para formar as RIs (Figura 1). Por exemplo, em um mapa de 2 km por 3 km, uma RI seria formada pelas coordenadas cartesianas Δx_i e Δy_i com deslocamento no eixo $x(0, 3)$ e eixo $y(0, 1)$. Por meio da OBU os veículos conseguem localizar-se, e consequentemente saberem em quais RIs do mapa eles estão. Para estabelecer a comunicação entre os veículos nas suas respectivas regiões, considerou-se duas funções que são implantadas nos veículos:

- *Produtor de informações*. Coleta os dados do tráfego, em seguida detecta o nível de congestionamento, disseminando-o para os veículos consumidores de informações. O primeiro veículo que detecta se a via está congestionada é considerado produtor de informação daquela região.
- *Consumidor de informações*. Recebe a informação, compartilhando-a para a sua RI.

O processo de disseminação ocorre quando o veículo realiza a estimação do nível do congestionamento por meio do comitê. Com isso, tal veículo torna-se produtor de informação V_{pi} . O V_{pi} cria uma mensagem de publicação contendo o seu ID, a sua localização, o conteúdo do evento (isto é, o nível de congestionamento) e o tempo de vida

do conteúdo. O V_{pi} dissemina o conteúdo de acordo com o nível de congestionamento da via. Por exemplo, se o veículo está em uma via congestionada, além do veículo deslocar mais lentamente, não há mudanças significativas no conteúdo (por exemplo, velocidade e localização) para ser enviado. Desta forma, pode-se utilizar um tempo maior (menor) para disseminar o conteúdo quando o nível de congestionamento é alto (baixo). Com isso, é possível definir a disseminação pelo intervalo do nível de congestionamento, como definido na Seção 3.2, para evitar o problema da *broadcast storm*, o qual é minimizado pela divisão do mapa em RIs.

Após o V_{pi} publicar o conteúdo, os seus veículos vizinhos recebem o evento publicado. Tais veículos tornam-se consumidores de informações, V_{ci} . Para os V_{ci} receberem o conteúdo, tem-se que subscrever no barramento de evento do pub/sub, com uma mensagem contendo o ID, o conteúdo a ser recebido, a coordenada de GPS do inscrito e o caminho que o inscrito passará. Além disso, o V_{ci} também se inscreve em outros veículos da região para receber dados de outros V_{cn} . Com isso, o V_{ci} obtém a informação do conteúdo local de onde ocorreu o congestionamento e outras informações da RI. Ao receber o conteúdo, o V_{ci} verifica se está na RI. Em seguida verifica se o tempo de vida do conteúdo acabou ou se o conteúdo é uma duplicata. Somente em caso afirmativo, o V_{ci} descarta o conteúdo. Do contrário, o conteúdo é repassado para os veículos que estão na região de acordo com o nível de congestionamento estimado. Neste caso, pode ocorrer fragmentação da rede devido, por exemplo, a falta de conectividade entre os veículos, resultando em mensagens perdidas. Por isso, foi utilizado um mecanismo de *store-carry-forward* que usa os *beacons* enviados dos veículos. O *store-carry-forward* armazena a informação e aproveita a mobilidade do veículo para transportá-la para diferentes partes da região. O processo para a troca de rotas é descrito a seguir.

3.4. Mecanismo de recomendação de rotas

Nesta seção, é apresentado o mecanismo de recomendação de rotas do Speed para evitar que os veículos passem por uma via congestionada. A recomendação ocorre quando os veículos recebem a informação que a via está congestionada por meio de um *beacon*. Com isso, o veículo verifica se passará pela rota congestionada. Caso positivo, o veículo realiza a troca de rota e atualiza tal informação em sua mensagem. Entretanto, se o veículo recebe a informação que o trânsito está livre ou moderado, o veículo apenas atualiza tal informação em sua mensagem e não realiza a troca de rota.

4. Avaliação de Desempenho e Metodologia

Esta seção apresenta os resultados da avaliação de desempenho, bem como a metodologia utilizada para gerar os resultados. Para isso, o Speed foi validado em duas etapas, a saber: (i) avaliação de desempenho para determinar a capacidade do Speed em estimar o nível de congestionamento das vias; e (ii) avaliação de desempenho da gerência de tráfego no TMS usando o Speed. Com tais avaliações, foi possível comparar o desempenho do Speed frente aos trabalhos da literatura. Os cenários modelados, as métricas utilizadas e os resultados alcançados juntamente com os parâmetros selecionados são apresentados nas próximas subseções.

4.1. Avaliação de desempenho para estimar o nível de congestionamento

Nesta subseção, é avaliado o desempenho do comitê que foi modelado no Speed para estimar o nível de congestionamento nas vias, comparando-o com os algoritmos

Fuzzy, KNN e RNA-MLP que foram adotados em [Araujo et al. 2014, Souza et al. 2015, Meneguette et al. 2016], respectivamente. Para a obtenção dos modelos de classificação, utilizou-se a base de dados que foi construída com base no HCM. Após a construção da base de dados, utilizou-se a técnica *k-fold cross-validation* com $k=10$ para separar o conjunto de teste e treino. Com isso, foi usado $k-1$ subconjunto para treinamento e o restante do subconjunto para teste. De maneira rotativa, o subconjunto de treinamento é alterado k vezes para fornecer uma estimativa mais confiável dos modelos de classificação construídos. Como o objetivo é avaliar o quão correto é estimado o nível de congestionamento, utilizou-se como variável de respostas as métricas: (i) precisão; (ii) recall; (iii) F-Measure; e (iv) Erro Médio Quadrático, EMQ. Tais métricas são calculadas a partir da matriz de confusão, como apresentado na Figura 3, na qual a linha representa a classificação gerada pelos algoritmos com base nas perdas causadas, e a coluna os dados de referência rotulados.

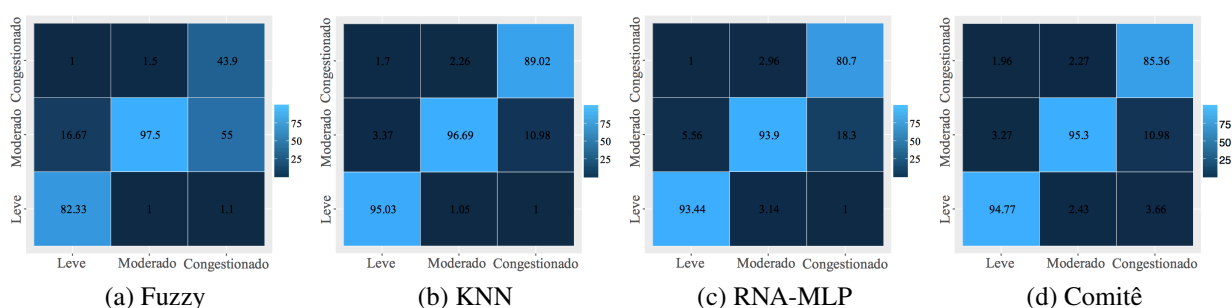


Figura 3. Matriz de confusão em porcentagem utilizada para calcular as métricas da Tabela 2.

A Tabela 2, derivada da Figura 3, sumariza os resultados das métricas em função dos algoritmos avaliados. Os melhores resultados estão destacados em negrito. Nota-se que o comitê modelado no Speed foi o que obteve os melhores valores, independente das métricas utilizadas. Em particular para a métrica precisão, o comitê foi 0.98 vezes mais preciso no pior caso e 1.06 vezes mais preciso no melhor caso. Isso faz sentido, pois o comitê modelou o problema de estimar o nível de congestionamento por meio da combinação de classificadores. Em razão disso, os erros gerados por um classificador isolado não são cometidos. Portanto, existe um ligeiro aumento nos acertos para estimar o nível de congestionamento e, conseqüentemente sua generalização no processo de decisão para a detecção da via congestionada. Portanto, o mecanismo proposto possui uma melhor taxa de acertos para estimar o nível de congestionamento quando comparado com os modelos de classificação individualizados.

4.2. Avaliação de desempenho da gerência de tráfego

Esta subseção avalia o fluxo de tráfego de veículo usando o Speed, apresentando sua eficiência no TMS. Para isso, o Speed é comparado com o INCIDENTEnt [Meneguette et al. 2016] e OVMT¹ que é o tráfego original de mobilidade do veículo. Os experimentos foram realizados por meio de simulações utilizando o OMNeT++ 5.0². Para o cenário urbano, utilizou-se um fragmento de 1km² da cidade de Ma-

¹OVMT, do inglês *Original Vehicle Mobility Traffic*

²OMNeT++, *Discrete Event Simulator*, <http://www.omnetpp.org>

Tabela 2. Avaliação dos algoritmos de classificação.

Algoritmos	Métricas			
	Precisão	Recall	F-Measure	EMQ ¹
Fuzzy	0,91±0,041	0,92±0,043	0,91	0,26
KNN	0,89±0,038	0,90±0,036	0,89	0,21
RNA-MLP	0,93±0,031	0,94±0,033	0,93	0,13
Comitê	0,94±0,027	0,96±0,024	0,95	0,11

¹EMQ - Erro Médio Quadrático.

nhattan, Nova Iorque, Estados Unidos, que foi gerado com o auxílio do SUMO 0.28.0³. O SUMO permite criar o modelo de mobilidade dos veículos, bem como a construção do cenário que foi extraído do OpenStreetMap⁴. No cenário gerado, os veículos viajam em ambas as direções ao longo da rua com uma velocidade de 0-60 *km/h*. Para prover uma condição de tráfego com vias congestionadas, a densidade da mobilidade dos veículos no cenário é de 300 veículos/km². Como é realístico assumir que há mais carros que ônibus e caminhões em uma via, considerou-se as seguintes quantidades de veículos: (i) 1/2 para carros; (ii) 1/4 para ônibus; e (iii) 1/4 para caminhões. Para realizar a comunicação inter-veículos, utilizou-se o framework Veins 4.5⁵ que foi integrado no OMNeT++. Além disso, o Veins possui implementado o IEEE 802.11p e modelos de atenuação de sinal para simular obstáculos na via. A Tabela 3 apresenta o conjunto de parâmetros que foi estabelecido para realizar a simulação.

Tabela 3. Conjunto de parâmetros utilizados na simulação.

Parâmetro	Valor padrão
Cenário	Manhattan
Área de simulação	1km ²
Densidade	300 veículos/km ²
Potência de transmissão	0.98 mW
Raio de transmissão	200 m
Taxa de transmissão	18 Mbit/s
Camada MAC	802.11p

Como o objetivo da avaliação é determinar o desempenho da gerência de tráfego usando o Speed, utilizou-se as seguintes métricas: (i) **tempo de viagem**, representa o tempo médio de viagem gasto de todos os veículos a partir de uma origem até o destino; (ii) **emissão de CO₂**, representa a média de CO₂ emitido durante o tempo de viagem de todos os veículos; e (iii) **consumo de combustível**, representa a média do consumo de combustível durante o tempo de viagem de todos os veículos. Para obter tais métricas, utilizou-se o EMIT que foi integrado no SUMO. O EMIT é um modelo estatístico que é derivado do HBEFA⁶ que estima o tempo de viagem, as emissões de CO₂ e consumo de combustível considerando a velocidade do carro e sua aceleração. As simulações fo-

³SUMO, *Simulation of Urban MObility*, <http://sumo.dlr.de/index.html>

⁴OpenStreetMap, <http://www.openstreetmap.org/>

⁵Veins, <http://veins.car2x.org>

⁶HBEF, *Handbook Emission Factors for Road Transport*, <http://www.hbefa.net>

ram replicadas 33 vezes com 95% de confiança com base na distribuição *t-student*. Os resultados obtidos da simulação são apresentados a seguir.

4.2.1. Análise dos resultados obtidos

A Figura 4a apresenta os resultados pertencentes à métrica tempo de viagem, realizando uma análise comparativa do Speed com o INCIDENT e o OVMT. Nota-se que o Speed possui um tempo de viagem menor independente das soluções, INCIDENT e OVMT. No pior caso que é usando o INCIDENT, o Speed realiza o tempo de viagem em média 1.02 vezes mais rápido. Isso ocorre pois o Speed utiliza um comitê de classificadores para estimar o nível de congestionamento. Com isso, é possível ter mais acertos na estimativa do tráfego, como destacado na Tabela 2, e conseqüentemente haverá mais acertos na sugestão de novas rotas. Em relação ao OVMT, nota-se que o Speed é 1.11 vezes mais rápido. Isso está relacionado com o mecanismo de troca de rotas que o OVMT não possui e, por isso, os veículos ficam mais tempo confinados no congestionamento.

Após avaliar o tempo de viagem, analisou-se a emissão de CO_2 durante a viagem dos veículos no TMS, como apresentado na Figura 4b. Observa-se que o Speed possui uma redução significativa de 14.17% na emissão de CO_2 quando comparado com o OVMT. Isto faz sentido, pois o OVMT não possui um mecanismo de troca de rotas. Com isso, os veículos que estão na via congestionada demoram mais tempo para finalizar o percurso. Já em relação ao INCIDENT, nota-se uma redução de 3.73% na emissão de CO_2 . Ainda, outro aspecto relevante está relacionado com a baixa dispersão da emissão de CO_2 do Speed quando comparado com as soluções implementadas. Isto é identificado pela amplitude interquartil do boxplot. Portanto, o Speed possui uma redução estatisticamente significativa na emissão de CO_2 . Vale salientar que tal redução pode ser explicada não apenas pelo mecanismo de disseminação de dados que entrega a mensagem em um tempo aceitável, mas também pelo uso do comitê de classificadores que possui uma alta precisão na classificação da via para realizar a alteração da rota.

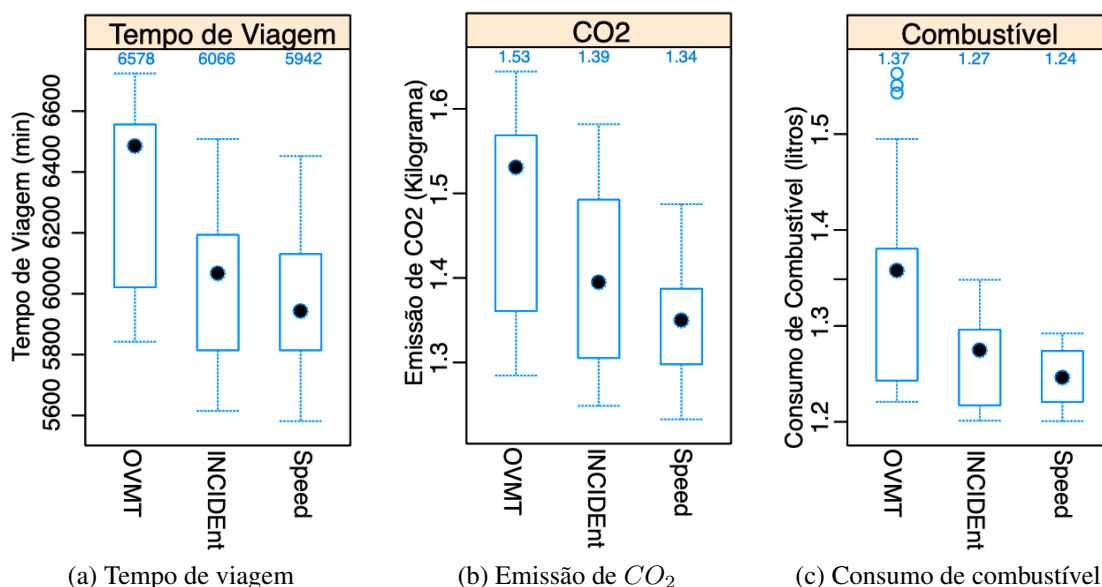


Figura 4. Impacto de desempenho do tempo de viagem, CO_2 e Combustível.

Na Figura 4c, tem-se o objetivo de avaliar a eficiência do Speed no que diz respeito a quantidade de combustível consumida durante o tempo de viagem dos veículos. Observa-se que a quantidade de combustível consumida pelo Speed é ligeiramente menor independente das soluções comparadas (isto é, OVMT e INCIDENT). Ao comparar com o OVMT e o INCIDENT, o Speed obteve uma redução de 10.48% e 3% no consumo de combustível, respectivamente. Isto ocorre pois, o Speed estima a via com mais precisão e no tempo adequado para sugerir novas rotas. Portanto, o Speed além de reduzir o consumo de combustível e CO_2 , possui um menor tempo de viagem. Isso é coerente devido a classificação correta do nível de congestionamento da via com um tempo aceitável na disseminação da informação.

5. Conclusão

Com o objetivo de atender os problemas encontrados na literatura e superá-los, este artigo propôs o Speed que, além de aumentar a precisão de acertos das vias congestionadas, tem como objetivo detectar o congestionamento da via e maximizar o fluxo de veículos no TMS. Para comprovar a eficiência do Speed, foi realizada uma avaliação de desempenho considerando dois cenários. Os resultados da simulação mostraram que o Speed possui um desempenho superior para todas as métricas avaliadas, quando comparado com outras abordagens da literatura. Salienta-se, portanto, que o Speed evidenciou avanço no estado da arte ao obter uma significativa taxa de acerto para detectar a via congestionada com redução no tempo de viagem, consumo de combustível e emissão de CO_2 . Como trabalhos futuros, planeja-se além de ampliar o estudo deste trabalho, realizar uma avaliação de desempenho do custo da rede no que diz respeito a disseminação dos dados.

Referências

- Araujo, G., Queiroz, M., Duarte-Figueiredo, F., Tostes, A., and Loureiro, A. (2014). Um protocolo de identificação e minimização de congestionamentos de tráfego para redes veiculares. Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos-(SBRC).
- Bauza, R. and Gozávez, J. (2013). Traffic congestion detection in large-scale scenarios using vehicle-to-vehicle communications. Journal of Network and Computer Applications, 36(5):1295–1307.
- Bauza, R., Gozalvez, J., and Sanchez-Soriano, J. (2010). Road traffic congestion detection through cooperative vehicle-to-vehicle communications. In IEEE Local Computer Network Conference, pages 606–612.
- Board, T. R. (2000). Hcm 2000 - highway capacity manual.
- Canuto, A. M. d. P. (2001). Combining neural networks and fuzzy logic for applications in character recognition. PhD thesis, University of Kent at Canterbury.
- Cintra, M. (2013). A crise do trânsito em são paulo e seus custos. GVExecutivo, 12(2):58–61.
- Cunha, F., Maia, G., Ramos, H. S., Perreira, B., Celes, C., Campolina, A., Rettore, P., Guidoni, D., Sumika, F., Villas, L., Mini, R., and Loureiro, A. (2018). Vehicular Networks to Intelligent Transportation Systems, pages 297–315. Springer Singapore, Singapore.

- Doolan, R. and Muntean, G.-M. (2013). Vanet-enabled eco-friendly road characteristics-aware routing for vehicular traffic. In Vehicular Technology Conference (VTC Spring), 2013 IEEE 77th, pages 1–5. IEEE.
- Doolan, R. and Muntean, G.-M. (2017). Ecotrec—a novel vanet-based approach to reducing vehicle emissions. IEEE Transactions on Intelligent Transportation Systems, 18(3):608–620.
- H. Allen, K. M. and Stonehill, M. (2013). A summary of the proceedings from the united nations climate change conference in doha, qatar, and their significance for the land transport sector, copenhagen: Bridging the gap (btg) initiative.
- Junior, G. D., Frozza, R., and Molz, R. F. (2015). Simulação de controle adaptativo de tráfego urbano por meio de sistema multiagentes e com base em dados reais. Revista Brasileira de Computação Aplicada, 7(3):65–81.
- Meneguette, R., E. De Grande, R., and A. F. Loureiro, A. (2018). Intelligent Transportation Systems, pages 1–21. Springer International Publishing, Cham.
- Meneguette, R. I., Geraldo Filho, P., Guidoni, D. L., Pessin, G., Villas, L. A., and Ueyama, J. (2016). Increasing intelligence in inter-vehicle communications to reduce traffic congestions: experiments in urban and highway environments. PLoS one, 11(8):e0159110.
- Meneguette, R. I., Ueyama, J., Krishnamachari, B., Bittencourt, L., et al. (2015). Enhancing intelligence in inter-vehicle communications to detect and reduce congestion in urban centers.
- Nottle, A., Quintana-Amate, S., Harborne, D., Alzantot, M., Braines, D., Tomsett, R., Kaplan, L., Srivastava, M., Chakraborty, S., and Preece, A. (2017). Distributed opportunistic sensing and fusion for traffic congestion detection. In First International Workshop on Distributed Analytics Infrastructure and Algorithms for Multi-Organization Federations, pages 1–6.
- Pan, J. S., Khan, M. A., Popa, I. S., Zeitouni, K., and Borcea, C. (2012). Proactive vehicle re-routing strategies for congestion avoidance. In 2012 8th IEEE International Conference on Distributed Computing in Sensor Systems, pages 265–272. IEEE.
- Pan, J. S., Popa, I. S., and Borcea, C. (2017). Divert: A distributed vehicular traffic re-routing system for congestion avoidance. IEEE Transactions on Mobile Computing, 16(1):58–72.
- Pessin, G., Souza, J. R., Osório, F. S., Faiçal, B. S., Geraldo Filho, P., Ueyama, J., Vargas, P. A., and Wolf, D. F. (2014). Investigation on the evolution of a robotic controller for autonomous vehicle navigation. Journal of Intelligent & Fuzzy Systems, 27(6):3047–3058.
- Rao, A. M. and Rao, K. R. (2012). Measuring urban traffic congestion-a review. International Journal for Traffic & Transport Engineering, 2(4).
- Santos, A. d. M. (2008). Uma análise da aplicação do modelo de rede neural repart em comitês de classificadores. Master's thesis, Universidade Federal do Rio Grande do Norte.

- Severino, J. N., Cristiani, A. L., Nakamura, L. H., and Meneguette, R. I. (2018). Urbtrans: A system for mobility management of school vans for cities in the interior. In Proceedings of the 10th Latin America Networking Conference, pages 10–17. ACM.
- SKYCOMP (2008). Major highway performance ratings and bottleneck inventory. Visitado em Outubro de 2018, http://www.skycomp.com/MDSHA/resources/Spring_2008.pdf.
- Souza, A. M., Brennand, C. A., Yokoyama, R. S., Donato, E. A., Madeira, E. R., and Villas, L. A. (2017). Traffic management systems: A classification, review, challenges, and future perspectives. International Journal of Distributed Sensor Networks, 13(4):1550147716683612.
- Souza, A. M., Guidoni, D., Botega, L. C., and Villas, L. A. (2015). Co-op: Uma solução para a detecção, classificação e minimização de congestionamentos de veículos utilizando roteamento cooperativo. Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos-(SBRC).
- Texas Transportation Institute, D.L. Schrank, W. T. J. (2015). 2015 urban mobility scorecard, mobility report.
- Wongcharoen, S. and Senivongse, T. (2016). Twitter analysis of road traffic congestion severity estimation. In Computer Science and Software Engineering (JCSSE), 2016 13th International Joint Conference on, pages 1–6. IEEE.

**III Workshop de Computação Urbana
SBRC 2019
Sessão Técnica 2
Modelagem e Mineração de Dados Urbanos**

Analyzing Patterns of a Bicycle Sharing System for Generating Rental Flow Predictive Models

Johnattan Douglas Ferreira Viana¹, Oton Crispim Braga¹,
Lenardo Chaves Maia², Francisco Milton Mendes Neto³

¹Programa de Pós-Graduação em Ciência da Computação - PPgCC
UFERSA/UERN, Mossoró, RN, Brazil.

²Departamento de Engenharias e Tecnologia - DETEC
Universidade Federal Rural do Semi-Árido, Pau dos Ferros, RN, Brazil

³Centro de Ciências Exatas e Naturais - DCEN
Universidade Federal Rural do Semi-Árido, Mossoró, RN, Brazil

{johnattandouglas, otoncbraga}@gmail.com,

{lenardo, miltonmendes}@ufersa.edu.br

Abstract. *Urban mobility has been highlighted as one of the most relevant themes in Smart Cities. Alongside this, following a principle of resource optimization and seeking greater sustainability, Bicycle Sharing Systems (BSSs) have stood out as a resource that can be used to assess urban mobility. The correct analysis of these data and the understanding of the dynamics in these systems can aid in decision making, in addition to optimize the complex urban mobility system. Thus, this work analyzes a BSS dataset, which is enriched for us with meteorological and seasonal information. In order to achieve our results, we recognize cyclist activity patterns related to date and climate information, as well as we identify a set of parameters that influences bicycle rental flow. Finally, we explore the relationship between these parameters and patterns, in order to present predictive regression models for rental flow prediction. In our results, Random Forest algorithm was the best approach for the creation of an effective regression model, explaining 95% of the explanatory variables.*

1. Introduction

Assuming the current growth trend of the metropolises, it is estimated that by 2050, more than 80% of the world population will be concentrated in urban centers [Zheng et al. 2014]. This expansion, which often happens uncontrolled, has been problematic in several aspects (i.e., energy consumption, traffic, safety). For that reason, it requires solutions that are consistent with current mobility challenges, such as the rising number of vehicles, increasingly time-consuming congestion and physical exhaustion of citizens due to chaotic traffic, resulting in loss of productivity.

These problems bring new opportunities for innovative technologies in Smart Cities, a concept focused on the integration of urban infrastructures and services to Information and Communication Technologies (ICTs) [Randhawa and Kumar 2017]. The context information collected by these scenarios has been an advance that has allowed the emergence of several applications that can do the integration of services in order to assist

decision-making in these scenarios. In this context, urban mobility has been highlighted as a challenging theme for public management, as well as one of the most relevant study areas in the ambit of Smart Cities [Georgescu et al. 2015].

In this sense, cycling has stood out as a great alternative to urban mobility problem, since the adoption of the bicycle reduces traffic congestion [Hamilton and Wichman 2018], besides impact positively the health of the population and the environment [Souza and Gomes 2014]. As a result, the number of bicycles in cities has grown [Fishman et al. 2013].

In addition, the use of bicycles has been enhanced with the emergence of Bicycle Sharing Systems (BSSs), an evolution of traditional bicycle rental systems, where the entire process of registering, lending and returning is automatic. These systems have evolved rapidly and are already in their fourth generation [Mátrai and Tóth 2016]. BSSs are shown as a complement to traditional public transport services such as buses and subways [Jäppinen et al. 2013] and, for that reason, the number of cities implementing them is growing [Correa et al. 2010].

BSSs have several stations throughout the city, where users rent a bicycle for a certain period and return it to a station at the end of the journey. The practicality of the systems encourages new cyclists, once it dispenses maintenance and parking concerns, which helps to minimize the use of cars on shorter journeys [Fishman et al. 2013]. In addition to the practical benefits of BSS, the data generated by these systems make them attractive for research, since variables such as travel time, departure time and arrival positions are explicitly recorded. Unlike other transportation services, this feature makes BSS a resource that can be used to assess urban mobility by identifying patterns and monitoring collected data.

How could we explore information from these systems to understand the dynamism of citizens' habits in order to improve urban mobility in Smart Cities? How can managers properly analyze dataset from public and private BSSs in order to provide better decisions in the strategic and operational context of their companies? The correct analysis of these data and the construction of predictive models, such as those that will be shown in this paper, can help answer these questions and assist in decision-making solutions. For this purpose, this work has the following contributions:

- enrichment of a database with context information and making it available for other researches;
- identification of attributes that influence bicycle rentals;
- recognition of patterns in the relationship of identified attributes;
- generation of a predictive model capable of inferring hourly rental flow.

The remainder of this paper is organized as follows: Section 2 shows some related works, which are focused on data from BSSs; In Section 3 the methodology of this work is presented; in Section 4 the results are discussed; finally, Section 5 presents the conclusions obtained in this work, as well as directions for future work.

2. Related Work

This work was inspired by [Fanaee-T and Gama 2013] that proposes an alternative for labeling events using background knowledge. They enriched 2011 records from a BSS

with weather and seasonal information. We replicate this approach with a more recent dataset for another purpose, with the aim of generating predictive models. On the other hand, some researches have already addressed the analysis of BSS data, mainly through a spatial-temporal perspective, in order to support in decision-making [Vogel et al. 2011]. For instance, after processing more than 18 million bike rides in Mexico City from 2010 to 2015, [Moncayo-Martínez and Ramirez-Nafarrate 2016] carried out an analysis of the mobility patterns using clustering to understand users' behavior related to usage in each stations. [Chen and Jakubowicz 2015] presented a model capable of inferring patterns of travel behavior, evaluating real data from a series of stations in Washington D.C.

[Borgnat et al. 2011] and [Kaltenbrunner et al. 2010] used statistical predictive models for different purposes, at Lyon and Barcelona stations, respectively. [Borgnat et al. 2011] used these models to predict the number of rents at a given time. [Kaltenbrunner et al. 2010] also used statistical predictive models to indicate the amount of free bikes to rent at certain stations. However, unlike these approaches, we use contextual information to generate an enriched dataset. We also identify the parameters that influence the quantity of rentals, finally generating a flow inference model.

[Razzaque and Clarke 2015] proposed an intelligent bicycle-sharing scenario based on Internet of Things. In addition, a number of services are proposed for the enhancement of current BSSs. Therefore, it is a model for the next generations of BSS using real-time information. For that, [Razzaque and Clarke 2015] evaluated an existing system, analyzing factors such as climate.

In our work, the analysis is not focused on the geo-referenced parameters of these data, but rather on the understanding of cyclists' temporal activity patterns related to contextual information, such as meteorological and seasonal variables.

3. Methodology

In order to extract useful knowledge from BSS' data, our work followed a methodology inspired by the classic process of Knowledge Discovery in Database (KDD) [Düsing 2000], using Artificial Intelligence techniques to aid in the decision-making process [Wickham and Grolemund 2017]. We scraped some contextual information from Web (i.e., weather) and integrated them to a BSS dataset. The integration step represents data enrichment process, resulting in a enriched dataset. We import, tidy, integrate, transform, visualize and model this dataset [Wickham and Grolemund 2017] to identify patterns and generate predictive rental flow models (Section 4). The methodology adopted is represented by Figure 1.



Figure 1. Methodology overview.

In this paper, the case study was applied to Capital BikeShare (CBS), the largest BSS in the United States across 6 jurisdictions: Washington, DC.; Arlington, VA; Alexan-

dria, VA; Montgomery, MD; Prince George’s County, MD; and Fairfax County, VA. This system has been running since September 2010 and currently has more than 500 stations and about 4300 bicycles¹. CBS has two types of users: *registered*, who is annual member, 30-day member or day key member; and *casual*, who does single trips or uses 24-hour pass, 3-day pass or 5-day pass. For practicality, in figures 2, 7 and 8, *registered* and *member* refer to the same user type.

CBS conducts studies that aim at continuous improvement such as removing the poorly used stations or adding new ones. For these surveys, stations have a computerized system that stores rental data. These data are publicly available, motivating the analysis and exploitation of these information. The data provided by CBS contains trip duration, start time, end time, info about start station and end station, bike identification and user type [System 2018]. The dataset used in this work represents rental flow from January 1 to December 31, 2017. It has about 3.75 million records, with approximately 35.4% of casual users and 64.6% of registered users.

In order to generate our enriched database using 2017 records, we performed a Web Scraping using a Node.js script, retrieving weather information from Freemeteo Website². For the integration process, it was considered the closest meteorological measurement to beginning of trip. In addition to meteorological data, information about Washington D.C holidays was collected from the Department of Human Resources (DCHR) Website³.

To extract as much information as possible from the *start_date* parameter, it was divided into *hour*, *day*, *month*, *season*, *weekday*, *workday* (Monday to Friday, excluding holidays). *start_station*, *start_station_name*, *end_station* and *end_station_name* parameters represent where the cyclist’s trip begins and ends, containing the numbering and the station name where bike was rented and returned. The *duration* parameter is the subtraction of *start_date* and *end_date*. These data were enriched by aggregating the lowest distance values between *start_station* and *end_station* (*m_distance*, in meters) and the probable duration using this shorter route (*m_duration*, in minutes), both calculated by Google Maps API⁴.

We used MySQL⁵ commands to integrate and manipulate these additional information. In this work, we investigate this enriched dataset we generated⁶. Some parameters are listed in Table 1, except variations from *start_date* parameter.

In order to facilitate the patterns identification, an hourly grouping of the records was carried out. In sequence, statistical analysis was performed and several descriptive graphics (bars, dispersion and curve) were generated. For this purpose, the *ggplot* library [Wickham 2016] for R language⁷ was used. Predictive models were also generated in the RStudio⁸ tool.

¹www.capitalbikeshare.com

²freemeteo.com

³<https://dchr.dc.gov/>

⁴<https://cloud.google.com/maps-platform/>

⁵<https://www.mysql.com/>

⁶Available in <https://goo.gl/FEFeXf>

⁷<https://www.r-project.org/>

⁸<https://www.rstudio.com/>

Table 1. Attributes from enriched database.

<i>bike*</i>	<i>start_longitude*</i>	<i>duration*</i>	<i>temperature**</i>
<i>type*</i>	<i>end_longitude*</i>	<i>m_duration</i> [◊]	<i>r_temperature**</i>
<i>start_station*</i>	<i>start_latitude*</i>	<i>m_distance</i> [◊]	<i>wind**</i>
<i>end_station*</i>	<i>end_latitude*</i>	<i>holiday</i> [•]	<i>humidity**</i>
<i>start_station_name*</i>	<i>start_date*</i>	<i>weather_id**</i>	<i>dew_point**</i>
<i>end_station_name*</i>	<i>end_date*</i>	<i>cut_description**</i>	<i>pressure**</i>

Data origin: CBS^{*}, Google Maps API[◊], DCHR[•] and Freemeteo^{**}.

4. Results and Discussions

After data enrichment, the generated dataset was examined to identify some rental flow patterns (Section 4.1) and to create predictive models using algorithms based on Machine Learning (Section 4.2). Besides that, it is provided a comparative evaluation to assess what is most efficient algorithm for this purpose (Section 4.3).

4.1. Pattern Identification

Analyzing the *type* parameter, it was possible to observe a notable discrepancy in the rental flow of *casual* and *registered* users. In total, the number of rents made by *registered* users (2.775.979) represents almost triple the amount made by *casual* ones (981.798). It indicates that 64.63% of rides were made by *registered* users. The average *duration* of registered rents is approximately 12 minutes, while that of casual ones is approximately 39 minutes. Thus, although the amount of casual rentals in general is lower, they tend to be longer, during on average three times more than registered rents.

We also identified a relationship between the *type* of rent and weekdays. It has been noticed that casual-type rents occur in greater numbers on Sundays and Saturdays. Among the rents of registered users, the amount is higher in the middle of the week (specifically on Wednesdays). Figure 2(a) represents the number of rents per weekday.

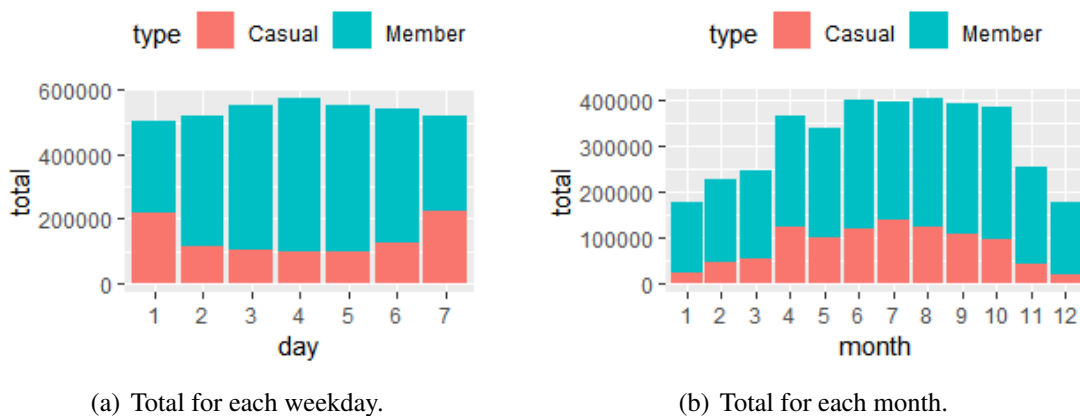


Figure 2. Total of rents per type user.

Analyzing the amounts of rents per month, it was observed that June, July, August, September and October are the months with the highest amount of rents, as shown in Figure 2(b). January and December represent the months with the lowest amounts of

rents. We presume that this decrease is directly related to the average temperature of these months, which are usually the coldest of the year.

Investigating the rental flow per *season*, it was verified that in the winter occurred 572,440 rents (15,2%), with an average of 266 rents per hour and about 6,360 rents per day. In the spring there were 1,072,608 rents (28,5%), with an average of 487 rents per hour and about 11,658 rents per day. In the summer, 1,197,255 (31,9%) were recorded, with an average of 542 rents per hour and approximately 13,013 rents per day. In the fall, 915,474 (24,4%) rents occurred, averaging 419 per hour and about 10,060 rents per day. Thus, winter is the season with less rents. On the other hand, summer is the season with more rents, justifying the peak of rents in these months (Figure 2(b)).

There is a proportional relationship between the *duration* and *season*, since the average duration of rents is usually higher in seasons of the year with more rents and lower in seasons of the year with less rents. Hence, rents tend to be longer in the summer. Figure 3 shows that in most months the amount of rents varies during the month. The months with the most continuous flow of rents are June, July, August and September, those that are part of the summer.

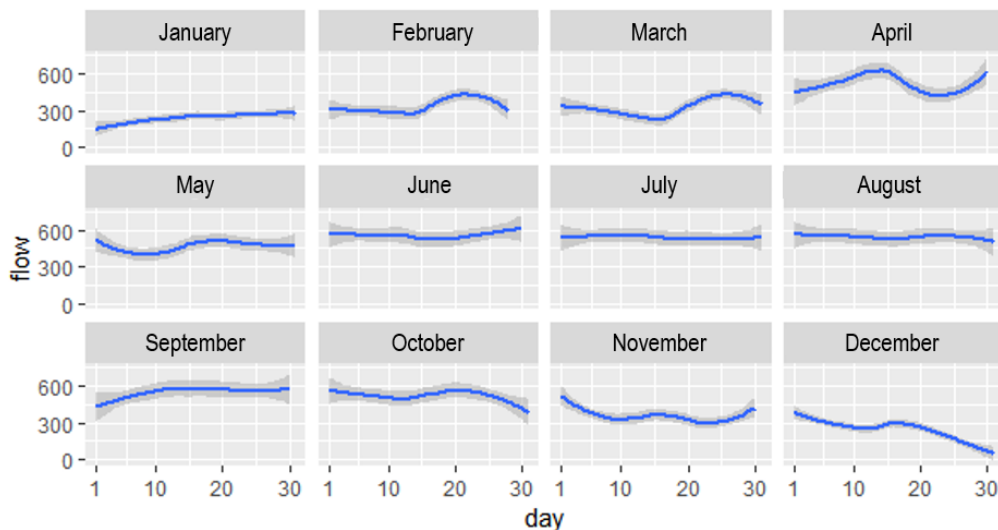


Figure 3. Quantity of rents per day of the month.

In addition, a similarity is seen in the rental flow per hour in each season. Figure 4 presents two peak times around 07:30 and 17:30. In winter, even with the lowest total amount of rents, the rental line flow still resembles the other seasons. These times are usually related to when people go to work and return home. This evidence that, regardless of season, most CBS users use it for the home-work route.

Regarding *hour*, rental flow is higher at times of greater movement in the city. Equivalently, as expected, the rental flow is not high at dawns. It was also noted an hourly patterns in weekends and weekdays (Figure 5). From Monday to Friday, it is clear two rental spikes, which happen around 07:30 and 17:30. On Saturdays and Sundays, there is a continuous increase in rents from 7:30 and a decrease after 15:00.

On holidays, the hourly patterns are similar to those found on weekends. Holidays are also included in Figure 4. However, because they were minor in relation to the working

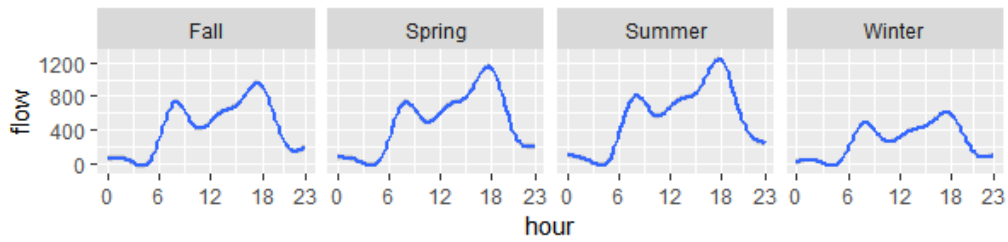


Figure 4. Amount of rents per hour in each season.

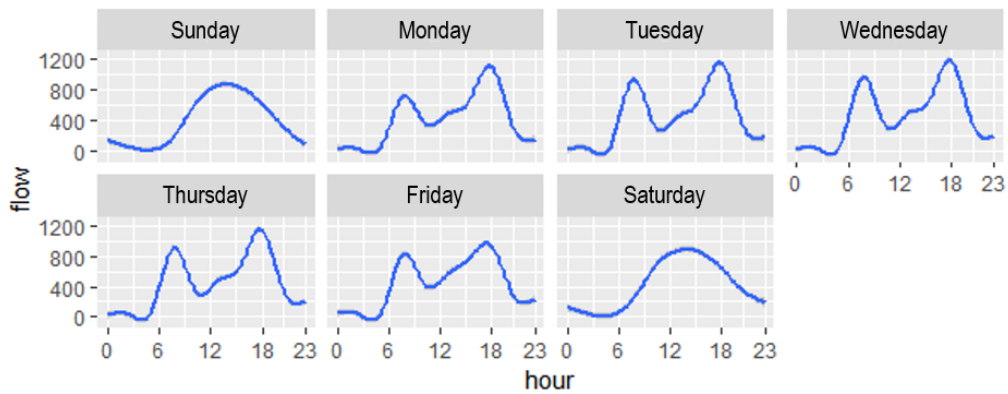


Figure 5. Amount of rents per hour in each weekday.

days, they were not enough to influence the curves in Figure 4. This same pattern of peak in rental flows resembles the hourly peak shown in Figure 5. In addition, rental flow has a relation with the climatic variables. Table 2 details the minimum, median, mean and maximum values of the climatic variables in dataset.

Table 2. Statistical summary of climatic variables.

Label	Min	Median	Average	Max
<i>temperature</i>	-9	17	15.83	36
<i>r_temperature</i>	-17	17	15.24	43
<i>wind</i>	0	13	14.11	63
<i>humidity</i>	13	66	65.13	100
<i>dew_point</i>	-19	10	8.60	27
<i>pressure</i>	990	1016	1017	1042

The analysis of the *temperature* (in °C) and the feels like temperature - real temperature (*r_temperature*, in °C) revealed a great relation of these two variables with the rental flow. As shown in Figure 6, the highest rent flow happens in spring and summer, which are periods of mild temperatures.

The analysis of the mean rental curve in each season shows weather intensity increases or decreases rental flow. In other words, users tend to do less rentals when it is extremely cold or too hot. Figure 6 demonstrates although there is a slight variation between the *temperature* and *r_temperature* parameters. In general, the average rental flow is higher between 15°C and 20°C.

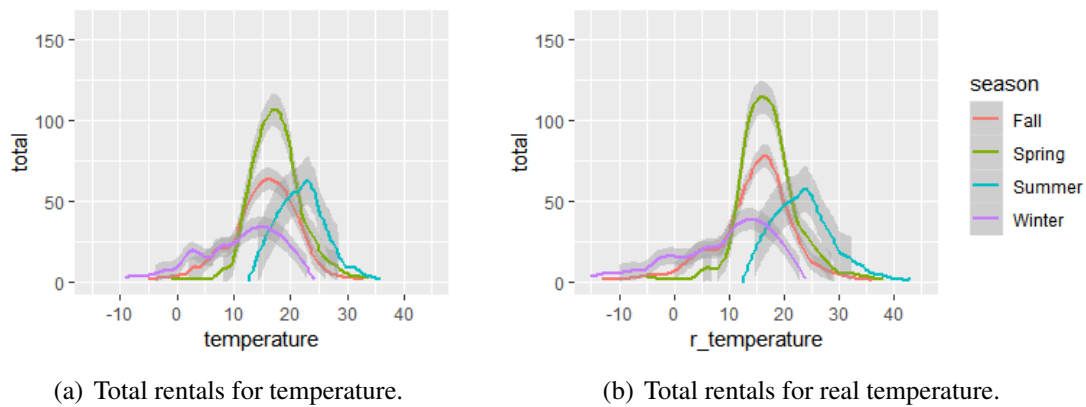


Figure 6. Relation of climate condition with rental flow.

In order to understand the influence of holidays on the rents dynamics, a comparison was made between the types of users for each of these days (Figure 7). However, on some holidays, the amount of casual rents approaches or exceeds the amount of rents recorded, unlike the pattern shown in Figure 2. Perhaps this happens because of the increase of visitor flow in the city during those days, considering the tourism potential of those dates. On some holidays, such as Independence Day and Memorial Day, there is a significant increase in rents, especially at stations close to historic or tourist monuments such as the White House and the Park National. For instance, during Independence Day, National Park Service (SPN) strongly encourages visitors of this holiday to use public transport for activities in the city, since public parking is extremely limited and cars are not allowed inside or around National Mall. In addition, as numerous roads are interdicted around George Washington Memorial Parkway, this is believed to be the main reason for the amount of rents on this day.

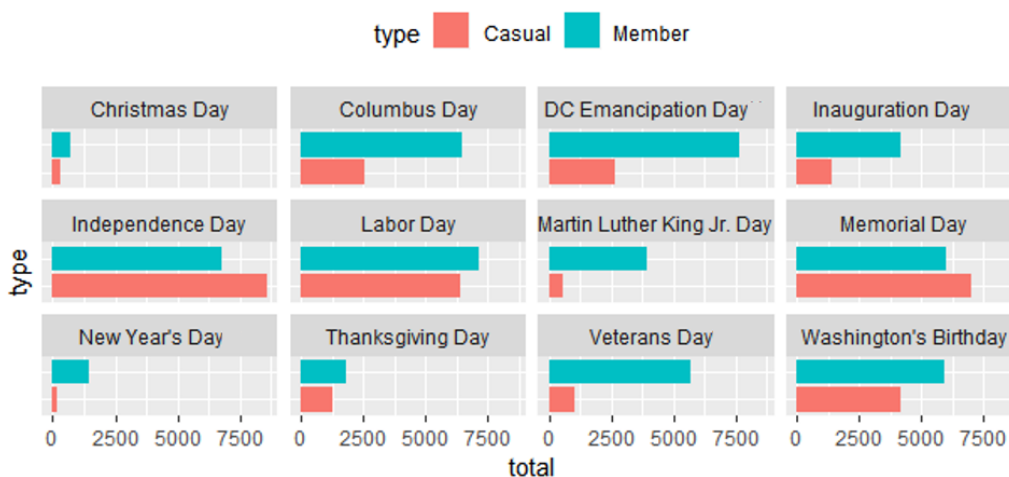


Figure 7. Amount of rent per type on each holiday.

In addition, in some holidays, the amount of rentals decreases rather than increasing, especially at Christmas and New Year. It is believed that in addition to the low temperatures, the type of holiday strongly influences this flow reduction, since, in general, people travel to spend these holidays as a family.

Another pattern found represents the relation between the rent duration ($m_duration$) and the distance covered ($m_distance$). In this relation, shown in a scatter plot in Figure 8, it was identified two rental profile groups, one that is usually faster and that travels farther, and another that usually takes longer and travels smaller distances.

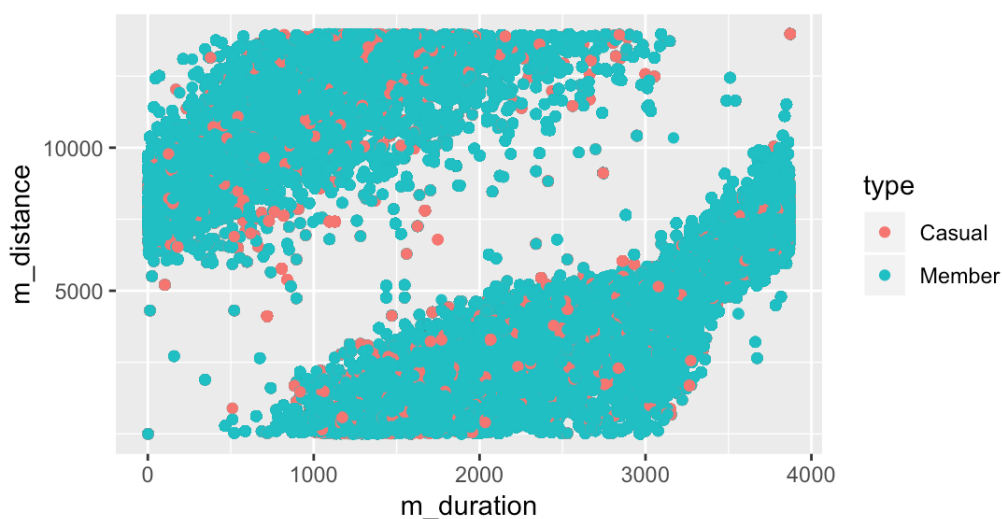


Figure 8. Relationship between distance and duration of rent.

It was believed that these two groups had correlation with *type*. The user who rents a bicycle to go to work, for example, wants to make the journey as fast as possible. On the other hand, it is thought that other users (i.e. tourists), take longer rents, but return the bicycle at nearby stations, and may even return it to the same place as the rent, resulting in shorter distances. However, as Figure 8 evidences, *type* does not seem to be the variable responsible for differentiating these two groups. In our analyzes, we also tested *day*, *weekday* and *season* variables, however, none of them effectively explains the grouping. When comparing the average rental $m_duration$ with $m_distance$ based on *start_station* and *end_station* (calculated using the coordinates), it was realized that many rents took much more time than expected. This happens mainly at stations near parks, during holidays or weekends. Thus, we believe that these rent profiles are related to the rental purpose, although eventually registered users do leisure trips and casual users do home-work-home trajectory.

4.2. Definition of the Predictive Model

The patterns identified in the data analysis process were used to apply algorithms based on Machine Learning techniques in order to generate an predictive hourly rental flow model. For that, as said previously, it was done the hourly grouping and total rent in each hour was saved in a *qtd* variable. In total, the new dataset⁹ after grouping contains 8737 records. To generate the predictive model, we selected 13 variables divided into two sets: *season*, *month*, *day*, *weekday*, *hour*, *workday*, *holiday*, that derive from *date*; and *temperature*, *r_temperature*, *wind*, *humidity*, *dew_point*, *pressure*, that derive from the weather. The parameters with average hourly measurements were calculated for the grouping of climatic values. Section 4.1 showed some analysis of these variables.

⁹Available in <https://goo.gl/FEFeXf>

In this case, we search for the number of rents (*qnt*), a discrete value. Thus, three Machine Learning algorithms were tested and evaluated:

Linear Regression is a statistical predictive model that treats scattered data in a linear way. The algorithm generates a better fit line in the scatter plot, which represents the relation of the explanatory variables to the dependent variable. Using the best fit line it is possible to predict a value for a dependent variable given a new input [Long et al. 1993]. Equation 1 represents the best fit line.

$$Y_i = \alpha + \beta X_i + \epsilon_i \quad (1)$$

Y_i is the dependent variable; α is a constant, which represents the intercept of the line with the vertical axis; β is another constant, which represents the line slope (angular coefficient); X_i is the explanatory variable (independent) that represents the explanatory factor; and, finally, ϵ_i is a random variable that represents the possible measurement errors.

Decision Tree is an algorithm based on gaining information, which is calculated using categorization techniques (i.e. Gini, Chi-square, entropy). The gain is expressed by Equation 2, which follows:

$$G(A) = I(p, n) \sum_{i=1}^v \frac{p_i + n_i}{p + n} I(p_i; n_i) \quad (2)$$

where

$$I(p, n) = -\frac{p}{p+n} \log_2 \frac{p}{p+n} - \frac{n}{p+n} \log_2 \frac{n}{p+n} \quad (3)$$

being p and n , respectively, positive and negative instances of a dichotomous classification variable [Long et al. 1993].

Random Forest is an assembly-based algorithm that generates a forest of sub trees. These trees are generated by a random factor and the combination of learning models increases the overall result [J.Ham and Kamber 2011].

4.3. Training and Testing

To test and compare the algorithms, a training/test method based on cross validation was applied [Kassambara 2018]. This approach divides the dataset in k parts and performs training and test iterations with each one. Each iteration separates the k -th part to test the trained model. Then the training part (k_i) is returned to the dataset while the next part (k_{i+1}) is separated for training. Thus, at the end of the process, all k parts are trained and tested. For this work, each algorithm was tested using cross-validation with 10 parts.

In order to assess the best model for purpose of this work, we considered the following metrics to calculate the effectiveness of the algorithms:

RSquare (R^2) corresponds to an adjustment measure of a generalized linear statistical model in relation to the observed values, varying from 0 to 1. This metric represents how much the model can explain the observed values. The higher this metric, the more explanatory the model will be, and the model better describes the samples [Kassambara 2018].

Mean Absolute Error (MAE) measures the average magnitude of errors in a set of predictions. As shown in Equation 4, this metric refers to the average over the test sample of the absolute differences between the prediction (y_j) and the real observation (\hat{y}_j), in which all individual differences have equal weight [Kassambara 2018].

$$MAE = \frac{1}{n} \sum_{j=1}^n |y_j - \hat{y}_j| \quad (4)$$

Root Mean Square Error (RMSE) is a quadratic scoring rule that also measures the mean magnitude of the error. It represents the square root of the mean square differences between the prediction and real observation (Equation 5).

$$RMSE = \sqrt{\frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2} \quad (5)$$

As shown in Equation 5, it is important to note that the square root of the mean squared errors has some interesting implications, since the errors are high before the average, assigning a relatively high weight to large errors. In this sense, this justifies the utility of this metric in our approach, once substantial errors are undesirable [Kassambara 2018].

Both *MAE* and *RMSE* express the average predictive model error. They can range from 0 to ∞ and are indifferent to the error directions. In addition, they have negatively oriented scores. So, since they represent error rates, lower values correspond to better models. Table 3 shows the performance of each algorithm in relation to the metrics described.

Table 3. Comparison among predictive models.

Algorithm	R ²	MAE	RMSE
Linear Regression	0.3825272	242.9403	320.544
Decision Tree	0.5607824	191.7552	270.3072
Random Forest	0.9560645	53.45990	87.02509

According Table 3, Linear Regression results were the worst in all metrics. We expected that because rental flow does not obey an expressive linearity, as shown in figure 4 and 5. It was possible to perceive the dynamicity of the flow in relation to several attributes, mainly *hour* and *weekday*, that present a certain non-linear pattern. On the other hand, the algorithms based on the gain of information showed better results, mainly the Random Forest, since it was able to explain 95% of the explanatory variables. The information gain strategy adopted by tree-based algorithms has shown promise for the problem faced. Therefore, despite having a higher computational cost, the model of prediction using Random Forest was chosen to infer the hourly rental flow prediction.

In addition, from the random trees generated by the algorithm, we could calculate the relevance of the attributes by the percentage of Increased of Mean Square Error (*IncMSE*), that describes the predictive capacity of the Mean Square Error with variables randomly exchanged. If this permutation drastically changes the predicted value, then the variable is considered critical.

The Increased Impurity Node was also calculated (*IncNodePurity*) that measures the loss function when the best tree nodes are selected, revealing variables that are most significant for the prediction [Echeverry-Galvis et al. 2014]. Figure 9 shows the relevance of the variable in the predictive model constructed with Random Forest algorithm.

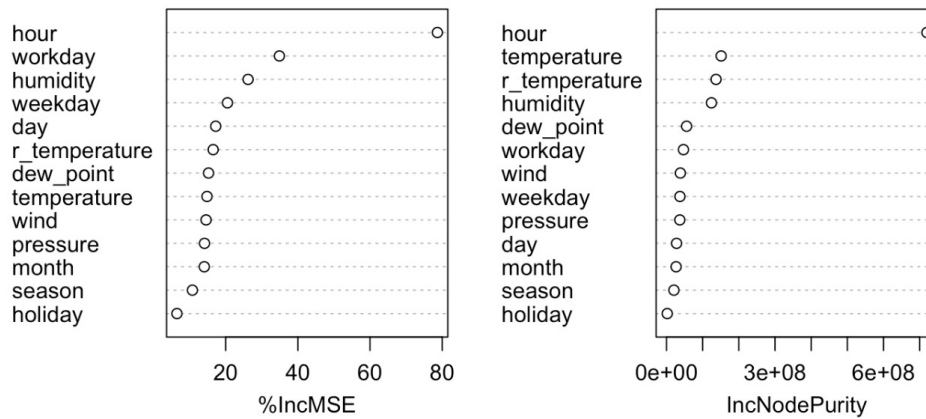


Figure 9. Relevance of the variables in predictive model providing by Random Forest algorithm.

As can be seen, the attributes identified as relevant in the pattern identification process (Section 4.1) are the most influential in the decision tree. It can also be seen that the values of *IncMSE* and *IncNodePurity* differ greatly between themselves, with the exception of *hour*, which is the tree root. That explains why the Random Forest had much better results than Decision Tree.

In this work, we also generated station rental flow prediction and daily rental flow prediction, but the results achieved using the algorithms presented in this work were not as significant as hourly rental flow prediction. From the obtained predictive model, a Web Scraping script was built for obtaining real-time weather information and so predict the amount of rents at a given time.

5. Conclusions

In this current work, Machine Learning algorithms and Data Science techniques supported the understanding of CBS dataset and the evaluation of bicycle rental flows. Certainly, this work approach can support decision making in BSSs, and it can be also applied in other systems that deal with urban mobility, such as rentals car and motorcycles. In addition, applying context information benefits in identifying patterns in these systems, besides enabling to generate rental flow predictive models. In this sense, patterns were observed in the relation between the temporal factors and the rental flow, mainly, if considered the hour that the rent occurred. For an example of practical application, these patterns can assist the managers of these systems in identifying better times for bicycle maintenance, considering less flowing hours.

In relation to the Machine Learning techniques tested in this work, Random Forest algorithm proved to be possible to apply temporal and climatic variables to create efficient predictive models. In our analyzes, this algorithm demonstrated a better result for the purposes of this work, explaining 95% of the explanatory variables.

As future works, we intend to perform a cluster analysis to describe which additional factors influence the relationship between *m_distance* and *m_duration*. Finally, we intend to apply Artificial Neural Networks to infer the rental flow, expecting to enhance our prediction by day, season and year.

Acknowledgment

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

References

- Borgnat, P., Robardet, C., Rouquier, J.-B., Abry, P., Fleury, E., and Flandrin, P. (2011). Shared Bicycles in a City: A Signal Processing and Data Analysis Perspective. *Advances in Complex Systems*, 14(3):415–438.
- Chen, L. and Jakubowicz, J. (2015). Inferring bike trip patterns from bike sharing system open data. In *2015 IEEE International Conference on Big Data (Big Data)*, pages 2898–2900.
- Correa, R., da Cunha, K. B., and Boareto, R. (2010). *A bicicleta e as cidades: como inserir a bicicleta na política de mobilidade urbana*. Instituto de energia e meio ambiente, 2 edition. [In Portuguese].
- Düsing, R. (2000). Knowledge discovery in databases. *Wirtschaftsinformatik*, 42(1):74–75.
- Echeverry-Galvis, M., Peterson, J., and Caceres, R. (2014). The social network: Tree structure determines nest placement in kenyan weaverbird colonies. *PloS one*, 9:e88761.
- Fanaee-T, H. and Gama, J. (2013). Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence*, pages 1–15.
- Fishman, E., Washington, S., and Haworth, N. (2013). Bike share: A synthesis of the literature. *Transport Reviews*, 33(2):148–165.
- Georgescu, M., Pavaloaia, V., Popescul, D., and Tugui, A. (2015). The race for making up the list of emergent smart cities. an eastern european country’s approach. *Transformations in Business and Economics*, 14:529–549.
- Hamilton, T. L. and Wichman, C. J. (2018). Bicycle infrastructure and traffic congestion: Evidence from dc’s capital bikeshare. *Journal of Environmental Economics and Management*, 87:72 – 93.
- J.Ham and Kamber, M. (2011). *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers is an imprint of Elsevier, 3 edition.
- Jäppinen, S., Toivonen, T., and Salonen, M. (2013). Modelling the potential effect of shared bicycles on public transport travel times in greater helsinki: An open data approach. *Applied Geography*, 43:13 – 24.
- Kaltenbrunner, A., Meza, R., Grivolla, J., Codina, J., and Banchs, R. (2010). Urban cycles and mobility patterns: Exploring and predicting trends in a bicycle-based public transport system. *Pervasive and Mobile Computing*, 6(4):455 – 466. Human Behavior in Ubiquitous Environments: Modeling of Human Mobility Patterns.

- Kassambara, A. (2018). *Machine Learning Essentials: Practical Guide in R*. STHDA.
- Long, W. J., Griffith, J. L., Selker, H. P., and D'agostino, R. B. (1993). A comparison of logistic regression to decision-tree induction in a medical domain. *Computers and Biomedical Research*, 26(1):74–97.
- Moncayo-Martínez, L. A. and Ramirez-Nafarrate, A. (2016). Visualization of the mobility patterns in the bike-sharing transport systems in Mexico City. In *2016 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pages 1851–1855.
- Mátrai, T. and Tóth, J. (2016). Comparative assessment of public bike sharing systems. *Transportation Research Procedia*, 14:2344 – 2351. Transport Research Arena TRA2016.
- Randhawa, A. and Kumar, A. (2017). Exploring sustainability of smart development initiatives in India. *International Journal of Sustainable Built Environment*, 6(2):701 – 710.
- Razzaque, M. A. and Clarke, S. (2015). Smart management of next generation bike sharing systems using Internet of Things. In *2015 IEEE First International Smart Cities Conference (ISC2)*, pages 1–8.
- Souza, L. C. and Gomes, E. T. A. (2014). O uso da bicicleta como meio de transporte: Mobilidade urbana na cidade do Recife. In *Anais do I Congresso Brasileiro de Geografia Política, Geopolítica e Gestão do Território*, pages 384–395. Letra1. [In Portuguese].
- System, C. B. S. (2018). Capital bike sharing trip history data.
- Vogel, P., Greiser, T., and Mattfeld, D. C. (2011). Understanding bike-sharing systems using data mining: Exploring activity patterns. *Procedia - Social and Behavioral Sciences*, 20:514 – 523.
- Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York.
- Wickham, H. and Grolemund, G. (2017). *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data*. O'Reilly Media, 1 edition.
- Zheng, Y., Capra, L., Wolfson, O., and Yang, H. (2014). Urban computing: Concepts, methodologies, and applications. *ACM Trans. Intell. Syst. Technol.*, 5(3):38:1–38:55.

Avaliação da Qualidade do Transporte Público no Acesso a Unidades de Saúde de Curitiba

**Clauciane D. Lima¹, Altieris M. Peixoto¹,
Luiz Gomes-Jr¹, Ricardo Lüders¹, Keiko V. O. Fonseca¹**

¹CPGEI / PPGCA / DAINF – Universidade Tecnológica Federal do Paraná (UTFPR)
Av. Sete de Setembro 3165 – 80.230-901 – Curitiba – PR – Brazil

Abstract. *This work evaluates the quality of public transport provided to users of basic health units of Curitiba city. Two databases have been used: one for health attendances, and another for public transport data. Part of the transport data is coded into a geographic graph database which allows to represent the transport network as well as to compute distances between points of interest. The quality of transport is evaluated by Accessibility and frequency of buses at near bus stops. Accessibility is measured by the walking distance from the stopping point to the nearest Health Unit, considering distances from 100, 300 to 500 m. The results show that 97% of units have good accessibility, but few bus lines serve more than five US directly.*

Resumo. *Este trabalho avalia a qualidade do transporte público oferecido aos usuários de unidades básicas de saúde (US) da cidade de Curitiba. Para tanto, são utilizadas duas bases de dados abertas: uma contendo os atendimentos das US e outra com os dados do transporte público. Parte dos dados do transporte é representada em uma base de dados de grafos georeferenciada que permite representar a topologia da rede de transporte e associar distâncias aos pontos de interesse. A qualidade do transporte é avaliada segundo a Acessibilidade e a frequência dos ônibus nas paradas próximas às US. A acessibilidade é mensurada pela distância de caminhada do ponto de parada à Unidade de Saúde mais próxima, para isso usou-se distâncias de até 100, 300 e 500 m. Os resultados mostram que 97% das US têm boa acessibilidade, mas poucas linhas de ônibus atendem mais do que cinco US diretamente.*

1. Introdução

Nas últimas décadas, as cidades brasileiras têm apresentado um crescimento elevado, impactando no sistema de transporte. Esse crescimento gera um aumento em distâncias que não podem ser percorridas a pé ou de bicicleta e dessa forma a dependência do transporte motorizado é inevitável, exigindo um planejamento da oferta e operação dos meios de transporte, principalmente os relacionados ao transporte público [Vasconcellos 1995].

O crescimento da população que mora em áreas urbanas faz com que as cidades enfrentem problemas para fornecer um transporte público de qualidade, surgindo problemas de congestionamentos, acidentes de trânsito, impactos ambientais e perda da qualidade de vida. Segundo [Ferraz and Torres 2004], o equacionamento adequado do transporte urbano é hoje uma preocupação de todos os países, pois a maioria da população mora nas cidades.

Desde a década de 70, Curitiba tem procurado por inovação buscando investir em obras que priorizem a mobilidade por meio do transporte público, implantando canaletas exclusivas que estão sendo complementadas por faixas exclusivas de ônibus. Além das vias segregadas, o serviço de transporte também evoluiu com a implantação de diferentes categorias de linhas que interagem com os corredores conectando diversas regiões da cidade, que é conhecida como RIT- Rede Integrada de Transporte, onde os usuários podem efetuar a troca de coletivo no interior de terminais sem a necessidade de pagar uma nova tarifa.

De acordo com [Gomide 2003], a existência de um serviço de transporte coletivo eficiente, com custo acessível e de qualidade, que proporcione acessibilidade da população a todo o espaço urbano, pode aumentar consideravelmente a disponibilidade de renda e tempo dos mais pobres, propiciando ainda o acesso aos serviços sociais básicos. A operação do transporte público é planejada para fornecer ao usuário uma determinada qualidade de serviço. Esta qualidade é inferida a partir de dados levantados junto à população, tais como rotas preferenciais, número de passageiros a serem transportados e horários de maior demanda. Com base nestas informações, linhas de ônibus são definidas, assim como sua programação horária. Assim, a qualidade do transporte coletivo é um conceito amplo que pode ser aferida por meio de diversos indicadores de qualidade. Este trabalho avalia a qualidade do transporte em relação à acessibilidade. O conceito de acessibilidade segundo [Ferraz and Torres 2004] está associada a facilidade do usuário chegar ao local de embarque no transporte coletivo e de sair do local de embarque até o destino final da viagem. De acordo com [Vasconcellos 2000], acessibilidade é a medida mais positiva dos efeitos de um sistema de transporte, podendo ser entendida como a facilidade de atingir os destinos almejados. Dessa maneira a acessibilidade pode ser avaliada por parâmetros como o tempo de espera nas paradas, frequência das linhas, distância a ser percorrida, entre outros.

O objetivo deste trabalho é avaliar a qualidade do transporte coletivo por meio da acessibilidade propiciada às unidades de saúde pública de Curitiba, segundo critérios de distância e frequência dos ônibus das paradas mais próximas das unidades. Com isso, espera-se contribuir para a identificação de áreas cujo atendimento possa ser melhorado, elevando a qualidade de atendimento da população.

Este artigo está organizado da seguinte forma. A Seção 2 apresenta conceitos básicos e trabalhos relacionados, a seção 3 apresenta a metodologia utilizada e descreve as bases de dados consideradas. A Seção 4 apresenta os resultados obtidos e a Seção 5 as conclusões e trabalhos futuros.

2. Conceitos Básicos

Para [Vasconcellos 1995] a acessibilidade é vista como a habilidade do indivíduo se movimentar, locomover e atingir o destino desejado, ou seja, representa a capacidade de movimentação do indivíduo levando em consideração as condições físicas e econômicas da pessoa.

Segundo [Raia 2000], a acessibilidade está relacionada com a oportunidade que o indivíduo possui em realizar uma atividade particular ou uma série de atividades. Está relacionada à mobilidade do indivíduo, localização espacial mais conveniente em relação ao ponto de partida do indivíduo, além das oportunidades de acesso às atividades diversas

e também na disponibilidade destas atividades. O autor ainda afirma que a acessibilidade é também o potencial disponibilizado pelo sistema de transporte público, de forma a favorecer à todos o desenvolvimento de suas atividades.

De acordo com [Ferraz and Torres 2004], a acessibilidade está associada à facilidade de chegar ao local de embarque do transporte coletivo e de sair do local de desembarque a fim de alcançar o destino final da viagem. Um item que reflete a acessibilidade do sistema de transporte de acordo com [Aguiar 1985] é o número de estações ou pontos de embarque, uma vez que quanto maior for esse número maior será a área coberta pelos serviços de transporte, levando-se em conta que cada ponto possui a sua área de influência própria.

A oportunidade de alcançar um destino é expressa pelo índice de acessibilidade que a relaciona à parâmetros como custo de viagem, tempo de viagem e distância, entre outros.

Para [Pilon and Xavier 2006], a acessibilidade é considerada um dos principais fatores na avaliação da qualidade do transporte público por ônibus. Segundo [Ferraz and Torres 2004], são 12 os principais fatores que influenciam na qualidade do transporte público. Dentre os quais pode-se citar:

- Acessibilidade está associada à facilidade de se chegar ao local de embarque, de deixar o local de desembarque e alcançar o destino final;
- Frequência de atendimento está relacionada ao intervalo de tempo da passagem dos ônibus do transporte público;

A Tabela 1 (adaptado de [Ferraz and Torres 2004]) mostra os atributos que caracterizam a acessibilidade e a frequência do ponto de vista dos usuários, como um serviço de qualidade bom, regular ou ruim.

Tabela 1. Padrões de qualidade para o transporte público de ônibus

Fatores	Parâmetro de avaliação	Padrões de qualidade		
		Bom	Regular	Ruim
Acessibilidade	Distância de caminhada	< 300	300 a 500	> 500
Frequência de atendimento	Intervalo entre atendimento (min)	< 15	15 a 30	> 30

2.1. Acessibilidade e os serviços de saúde

No caso das unidades de saúde pública da cidade de Curitiba, objeto deste estudo, muitas delas foram projetadas em épocas em que a acessibilidade não era tão relevante quanto hoje. Como o conceito de acessibilidade é amplo, nesse tópico abordaremos o conceito de acessibilidade sobre a perspectiva dos serviços de saúde.

[Scatena et al. 2009] realizou um estudo sobre as dificuldades de acesso aos serviços de saúde para o diagnóstico específico da tuberculose. No estudo ele aborda dois fatores principais negativos do acesso aos usuários: deslocamento até a unidade de saúde e, serviço de atendimento falho.

[Raia and Pereira 2001] propõe uma metodologia que serve de ferramenta de planejamento, a fim de avaliar a eficácia e eficiência dos equipamentos públicos urbanos existentes. Como resultado, os autores obtiveram um diagnóstico sobre a localização espacial, a acessibilidade aos serviços básicos de saúde para os núcleos de saúde localizados na cidade de Bauru/SP.

[Garcia and Raia 2015] realizou um estudo sobre a acessibilidade aos hospitais na cidade de São Carlos. O estudo aborda a acessibilidade considerando três modos de viagem: a pé; automóvel e transporte público (ônibus). Este estudo se baseia em indicadores de tempo para medir a acessibilidade e dados do censo para estimar o rendimento médio de setores censitários, a fim de verificar se a parcela da população que mais necessita do serviço de saúde possui níveis de acessibilidade adequados.

[Ferreira and Raffo 2013] apresentam um estudo baseado em Sistemas de Informação Georeferenciados (GIS) que mostra a acessibilidade dos habitantes rurais da cidade de Registro (SP) aos postos de saúde e hospitais na região. Os autores afirmam que a análise da acessibilidade é de grande importância para a população, pois todos precisam garantir o direito à assistência médica-sanitária e hospitalar.

Assim, diversos autores abordam a acessibilidade do usuário do sistema de saúde às questões geográficas, ou seja, à localização do serviço de saúde. De forma similar, este trabalho avalia aspectos de posicionamento geográfico das unidades de saúde, mas se concentra no posicionamento das paradas de ônibus próximas, que são o meio de acesso ao transporte público e propiciam a diversidade de oferta de transporte para diferentes origens e destinos da área urbana.

2.2. Transporte público de Curitiba

Curitiba é considerada uma referência no transporte público. A cidade conta com a Rede Integrada de Transporte (RIT) desde 1974, onde é possível trocar de linhas em determinados pontos pagando-se uma única tarifa. Atualmente, possui uma frota de 1410 ônibus operantes mais reserva, que procura atender um público de 1.389.731 passageiros por dia com 251 linhas de ônibus, 329 estações e 21 terminais [URBS 2019]. A administração é realizada pela Urbanização de Curitiba S/A (URBS), empresa de economia mista que fiscaliza as empresas particulares prestadoras de serviços.

Os terminais são pontos de integração localizados em locais estratégicos da cidade que visam atender usuários que moram nas regiões vizinhas que chegam até um dos 21 terminais por meio de "linhas alimentadoras". De dentro dos terminais os usuários do transporte público podem escolher qualquer percurso pagando uma tarifa única, o que ajuda a diminuir o gasto da população com o transporte, permitindo realizar o traslado entre linhas nos terminais.

Para [Vasconcellos 2005], os terminais de transporte devem ser confortáveis para os usuários e também devem organizar as chegadas e partidas dos ônibus de forma a minimizar o tempo de transferência dos passageiros. Os ônibus da Linha Expresso têm a função de complementar o trajeto até o centro da cidade por canaletas exclusivas. Curitiba conta ainda com linhas interbairros e linha diretas, que tem a finalidade de conectar outros pontos da cidade.

3. Metodologia

A metodologia utilizada identifica os pontos de ônibus mais próximos de cada unidade de saúde e a frequência de atendimento destes pontos pelas respectivas linhas de ônibus. Para tanto, são utilizadas as bases de dados de atendimentos fornecida pela prefeitura de Curitiba¹ e as bases de dados do transporte público fornecidas pela URBS. Esta última é utilizada para construir uma base de dados de grafos que contém tanto a topologia da rede de transporte quanto dados georeferenciados de operação.

3.1. Base de dados das Unidades Básicas de Saúde

Para este trabalho foi utilizada a base de dados do sistema de saúde de Curitiba “Perfil de atendimento de Enfermagem nas Unidades Municipais de Curitiba”, disponibilizados pela prefeitura de Curitiba², referente aos meses de outubro, novembro e dezembro de 2017.

Após uma filtragem, a base utilizada passou a ter 502.403 linhas e 7 colunas com informações de data do atendimento (dia e hora), data de nascimento, sexo, tipo da unidade (nome da unidade), descrição da unidade (Unidade Básica de Saúde ou Unidade de Pronto Atendimento), meio de transporte, bairro.

Para este trabalho, foram consideradas as informações de data do atendimento, tipo da unidade, descrição da unidade e meio de transporte. Apenas atendimentos das Unidades Básicas de Saúde (US) foram considerados. Quanto aos dados referentes ao meio de transporte uma grande parcela apresentou mais de uma opção por resposta, dessa forma adotou-se “ônibus” para todos os atendimentos que mencionaram este meio de transporte, exceto para as respostas que havia ônibus e carro que contabilizou 50% para ambos.

3.2. Base de dados do transporte público

Através do portal de dados abertos de Curitiba é disponibilizado diariamente em formato JSON, dados sobre a rede de transporte público da cidade. A rede do transporte público de Curitiba foi mapeada em um banco de dados de grafos Neo4J³ com as informações georeferenciadas. Assim, foi possível construir um grafo cujos vértices são as paradas de ônibus e as arestas são as ligações entre paradas definidas pelas rotas das linhas de ônibus em operação. Após, foram inseridos vértices correspondentes a cada uma das 111 unidades de saúde.

Em um banco de dados PostgreSQL foram carregadas as informações das US juntamente com os dados das linhas e horários de ônibus do transporte público. Foram utilizados os dados do dia 18 de outubro de 2017 disponibilizados pela URBS⁴ (Urbanização de Curitiba S/A).

No banco de dados Neo4J foi possível identificar as paradas mais próximas de cada US em um raio de até 500 m. Em seguida, foram identificadas as linhas de ônibus que atendem estas paradas e a respectiva programação horária de cada uma com as informações do banco PostgreSQL.

¹<http://www.curitiba.pr.gov.br/dadosabertos/consulta>

²<http://www.curitiba.pr.gov.br/dadosabertos/consulta>

³<https://neo4j.com/>

⁴<http://dadosabertos.c3sl.ufpr.br/curitibaurbs/>

3.3. Tempo de espera nas paradas

A metodologia adotada para o cálculo da acessibilidade através do tempo de espera T do transporte para cada US foi proposta por [Garcia and Raia 2015]. Este trabalho apresentou um estudo sobre a acessibilidade de transporte a hospitais realizado na cidade de São Carlos (SP) utilizando a Equação (1),

$$T = \frac{a}{100} \cdot \frac{\bar{h}_a}{2} + \frac{b}{100} \cdot \frac{\bar{h}_b}{2} \quad (1)$$

sendo que a é o percentual de linhas que passam pela US, \bar{h}_a é o *headway* médio (intervalo de tempo entre dois ônibus de uma mesma linha) de todas as linhas que passam pela US, b é o percentual de linhas que não passam pela US e \bar{h}_b é o *headway* médio de todas as linhas que não passam pela US. Os *headways* médios \bar{h}_a e \bar{h}_b são obtidos pela média aritmética dos *headways* individuais das linhas correspondentes. Estes *headways* individuais são obtidos pelo número de ônibus programados por hora para percorrer a linha em determinado horário.

4. Resultados

4.1. Distância das paradas e frequência dos ônibus

Neste trabalho serão analisados os indicadores de acessibilidade relacionados à distância dos pontos de parada mais próximos das US e a frequência das linhas de ônibus nestes pontos. A cidade de Curitiba possui um total de 6500 pontos de parada de ônibus.

Um resultado importante é o número de pontos de ônibus existentes na cidade distantes das US até 100, 300 e 500 metros. De acordo com a Tabela 2, 47% das US possuem pelo menos um ponto de ônibus distante até 100 m e 50% possuem pelo menos um ponto de ônibus distante entre 100 e 300 m das US. Assim, 97% das US têm acessibilidade boa de acordo com a Tabela 1. Apenas 3% tem acessibilidade regular, pois estão distantes entre 300 e 500 m de um ponto de ônibus. Portanto, 100% das US de Curitiba (111 no total) são atendidas por pelo menos uma parada de ônibus até 500 m de distância. Ou seja, nenhuma US tem acessibilidade ruim do ponto de vista da distância de um ponto de ônibus. Entretanto, isso é uma aproximação, visto que a distância calculada não é de caminhada, mas entre as coordenadas das US e dos pontos de ônibus.

Tabela 2. Acessibilidade das US segundo a distância (m) da parada mais próxima.

Raio < 100	$100 \leq \text{Raio} \leq 300$	$300 < \text{Raio} \leq 500$
47%	50%	3%

A Figura 1(a) mostra as US (cruzes) e os respectivos pontos de ônibus (círculos) distantes até 100 m. A Figura 1(b) mostra às opções de linhas disponíveis nos respectivos pontos de ônibus (quanto mais escura a região mais opções de linhas o usuário tem para chegar à US). Similarmente, a Figura 2, mostra as características das paradas distantes até 300 m e a Figura 3 as paradas distantes até 500 m das US.

Em até 100 m já é possível encontrar paradas de ônibus, em que todas as opções de linhas já foram detectadas para o usuário chegar até a US. Essas US apresentaram poucas

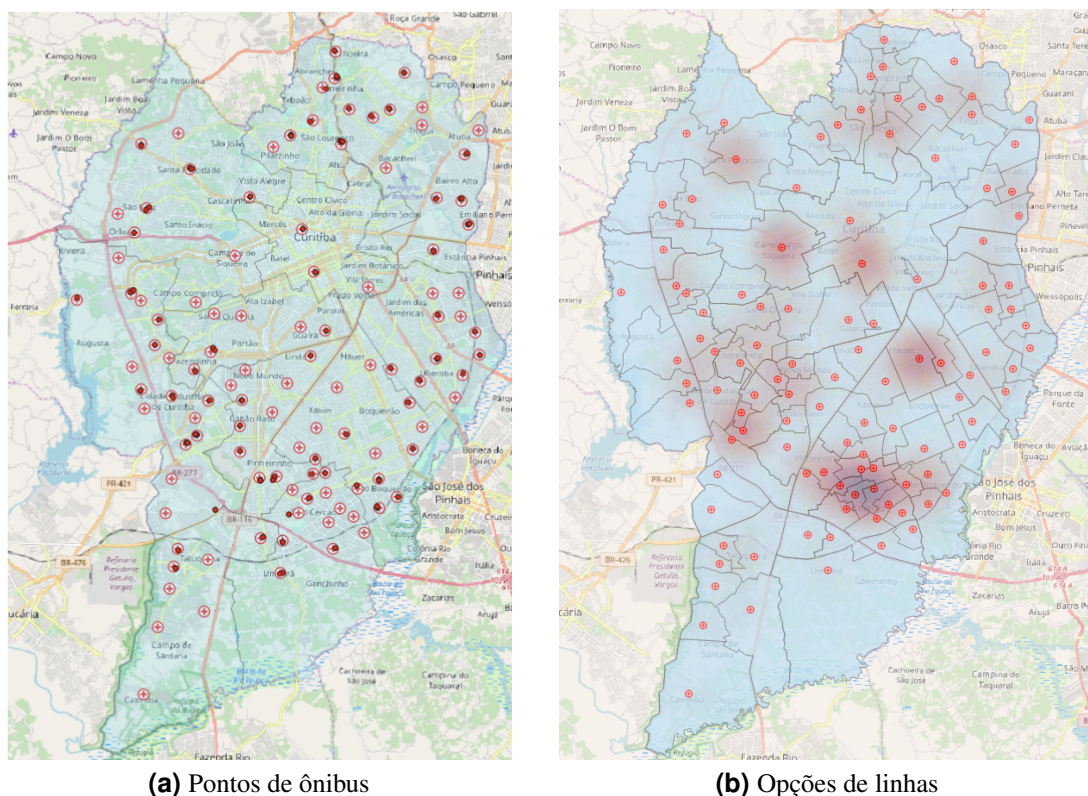


Figura 1. Mapa temático das US com paradas de ônibus até 100m

opções de linhas aos usuários. Os dados foram agregados para análise até 300 m, 500 m, mostrando que para algumas US é necessário percorrer distâncias maiores que 100 m para obter mais opções de linhas. Até 300 m o número US com 100% das opções de linhas detectadas cresce de 2,7% para 24,3%. Porém, a grande maioria das paradas de ônibus ainda possuem opções de linhas para distâncias entre 300 e 500 m.

É possível concluir que existem várias US com poucas opções de linhas, o que pode gerar uma lotação no transporte público, caso o tempo de espera pela linha não seja considerado bom.

O indicador de qualidade da frequência de atendimento está relacionado ao intervalo de tempo entre ônibus, conforme a Tabela 1. Este intervalo de tempo é dado pelo tempo de espera T da Equação (1) em função de \bar{h}_a e \bar{h}_b para cada US.

Para avaliar este tempo de espera no período das 7:00 às 19:00 foi necessária a descrição das linhas em um raio de 500 m de cada US, o intervalo de tempo entre ônibus de uma mesma linha e os pontos de paradas. O valor encontrado para \bar{h}_a variou de 16 a 60 min e o valor de \bar{h}_b de 33 a 34 min. Considerando $1 \leq a \leq 15$ e $85 \leq b \leq 99$, o tempo médio de espera nas paradas próximas às US variou de 16 a 19 min, ou seja, de qualidade regular segundo a Tabela 1.

A análise referente à qualidade do transporte oferecido pelas linhas distintas que servem às US, foi calculado com base no *headway* de cada linha. A maioria delas foram classificadas como regulares e ruins, conforme a Tabela 3.

Em seguida, tem-se uma análise da quantidade de US que uma linha de ônibus

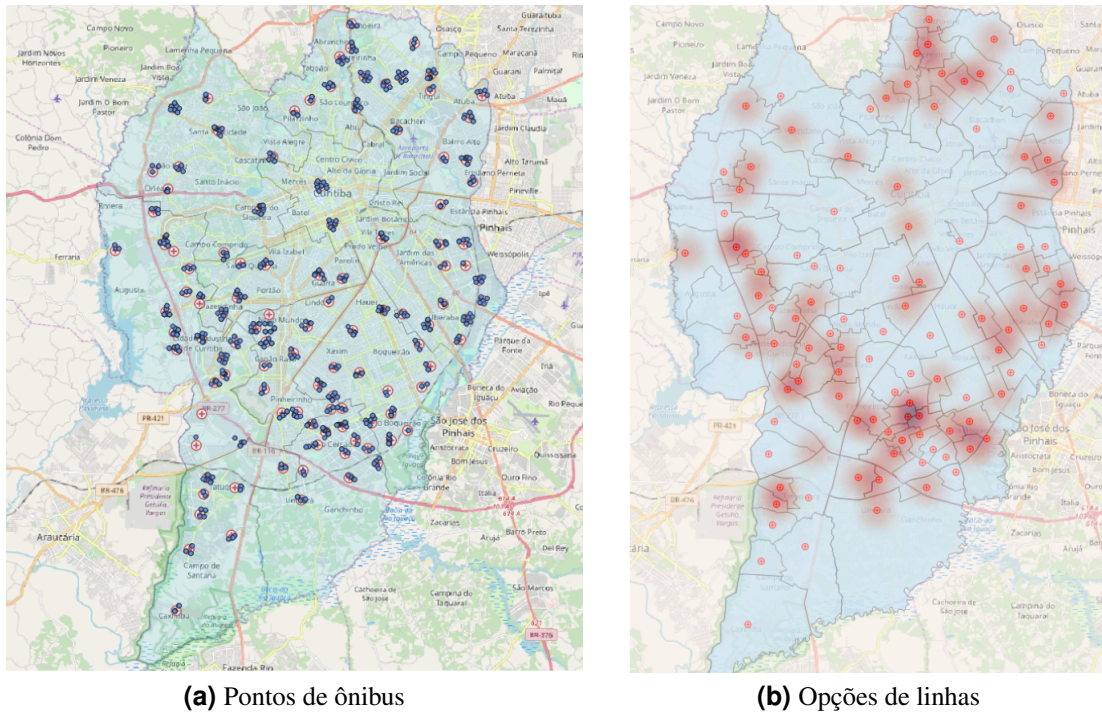


Figura 2. Mapa temático das US com paradas de ônibus até 300m

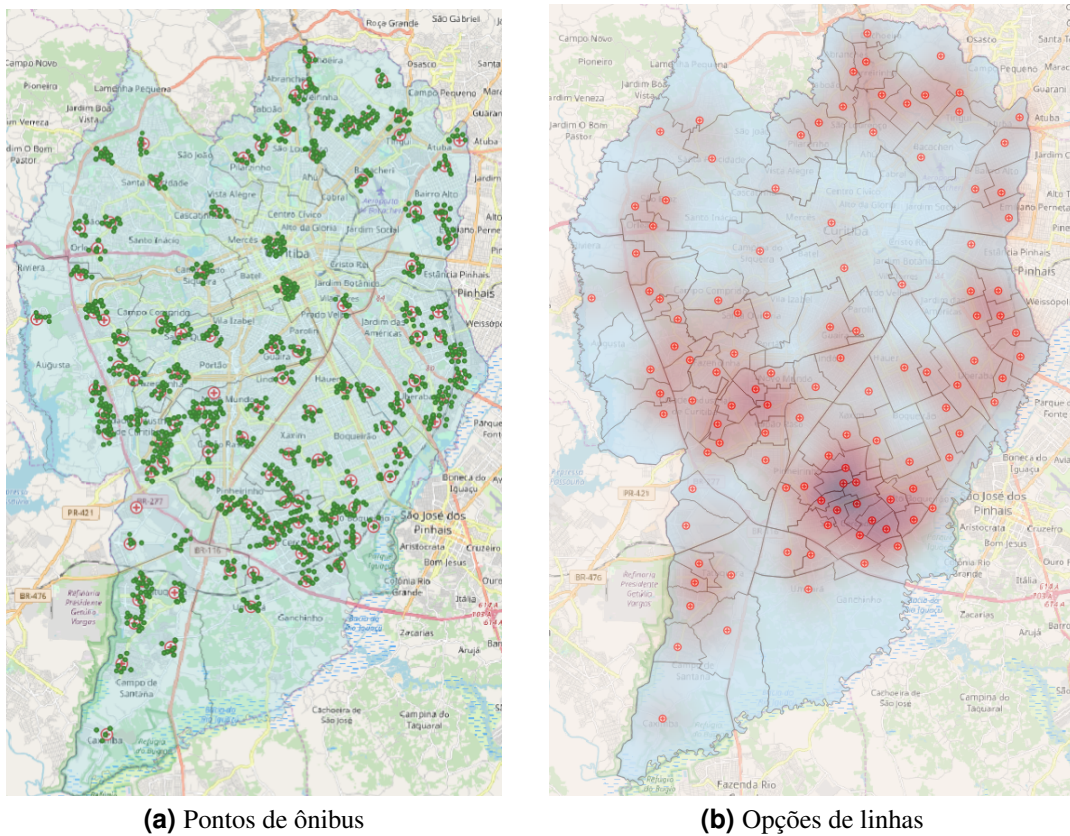
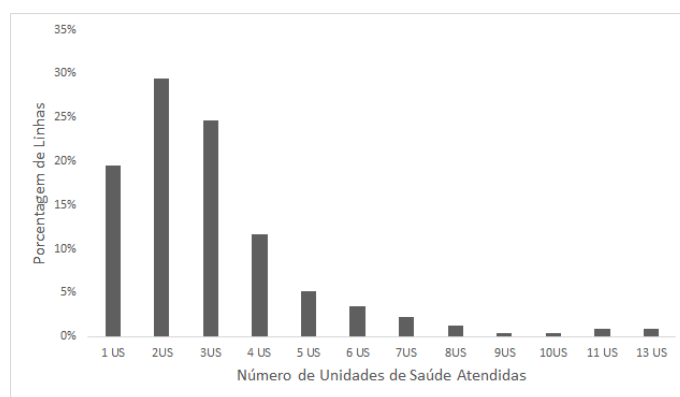


Figura 3. Mapa temático das US com paradas de ônibus até 500m

Tabela 3. Qualidade do transporte segundo o tempo de espera.

Bom	Regular	Ruim
8%	49%	43%

pode servir. De acordo com a Figura 4, a maioria das linhas de ônibus de Curitiba passam próximas (até 500 m) de poucas US. Cerca de 20% das linhas identificadas passam próximas de uma única US e aproximadamente 30% passam próximas de duas US. Ou seja, aproximadamente 50% das linhas de ônibus de Curitiba atendem 1 ou 2 US. Isso é ruim para a população, pois limita a escolha da linha de ônibus para se chegar a uma US, a não ser que o usuário realize traslado em um terminal aumentando o tempo de viagem. Mesmo assim, cerca de 15% das linhas apresentam conexões diretas com 5 ou mais US.

**Figura 4. Distribuição das linhas de ônibus de Curitiba em relação ao número de US que atendem**

4.2. Distritos Sanitários

Curitiba é dividida em dez distritos sanitários que agregam bairros e US. A Tabela 4 mostra a qualidade das linhas que passam próximas (até 500 m) das US em relação ao intervalo de atendimento dos ônibus dessas linhas agrupadas por distrito sanitário. A maior parte dos distritos são servidos por linhas consideradas regulares ou ruins, segundo a Tabela 1. Além disso, na rede de transporte é possível encontrar linhas que servem mais do que 4 distritos sanitários. Porém, a maioria serve apenas um distrito como pode ser visto na Tabela 5.

A partir da base de dados das US, levantou-se o número de atendimentos durante os meses de outubro, novembro e dezembro de 2017, de segunda à sexta-feira, das 7 h às 19 h. O resultado por distrito sanitário é mostrado na Figura 5, sendo Cajuru o distrito de maior volume de atendimentos registrados no período considerado. Conforme a Tabela 6, o número de atendimentos por distrito pode ser relacionado à média das distâncias de cada US à parada de ônibus mais próxima (distância mínima). A Tabela 6 mostra que a maior parte dos distritos sanitários possuem US distantes em média 100 m aproximadamente das paradas mais próximas. Porém, estas distâncias mínimas têm uma variabilidade elevada de acordo com os respectivos desvios padrões. Além disso, os distritos Matriz e Santa Felicidade com os mais baixos números de atendimentos possuem as menores médias de distâncias mínimas (linhas em negrito na Tabela 6).

Tabela 4. Qualidade das linhas de ônibus por distrito sanitário segundo frequência de atendimento

Distrito sanitário	Bom	Regular	Ruim
Boa Vista	9%	36%	55%
Cajuru	14%	55%	32%
Matriz	8%	53%	39%
Boqueirão	26%	56%	18%
CIC	3%	66%	31%
Portão	18%	58%	24%
Santa Felicidade	21%	38%	41%
Bairro Novo	16%	56%	28%
Pinheirinho	7%	58%	35%
Tatuquara	0%	60%	40%

Tabela 5. Porcentagem de linhas que servem de 1 até 4 ou mais distritos

% Linhas	Número de distritos
53%	1
34%	2
9%	3
3%	4
1%	> 4

5. Conclusão

Este artigo apresentou uma avaliação da qualidade da acessibilidade às unidades de saúde públicas da cidade de Curitiba usando o transporte público de ônibus. A qualidade foi mensurada segundo critérios de acessibilidade das paradas dos ônibus às US e frequência de circulação dos ônibus nestas paradas. O número de atendimentos em cada US também foi considerado. Para tanto, foram necessários dados tanto dos atendimentos nas US quanto do sistema de transporte, representado em uma base de grafos georeferenciada. Em relação à distância dos pontos de parada até a unidade de saúde, a acessibilidade de 97% das US foi considerada boa, com pontos localizados a uma distância (euclidiana) menor ou igual a 300 m. Já a avaliação do tempo de espera, mostrou acessibilidade regular das US, para um tempo médio de espera entre 16 a 19 min calculado para cada US. Entretanto, este tempo de espera é uma média que considera a oferta de transporte nos pontos de parada de maneira geral. Caso o usuário necessite de uma linha específica, o tempo de espera pode variar bastante dependendo da linha. Por isso, se fez necessário avaliar a qualidade das linhas individuais nas paradas próximas às US. Como resultado, a maioria delas foi considerada regular ou ruim em relação ao tempo de espera. Em relação à disponibilidade de linhas para acesso às US, os resultados mostram que aproximadamente 50% das linhas de ônibus de Curitiba servem uma ou duas US. Isso é uma limitação relevante nas opções de linhas de ônibus para acesso às US. Em uma análise por distrito sanitário, as linhas de ônibus também foram avaliadas como regulares ou ruins, atendendo em sua maioria apenas um único distrito. Além disso, os distritos com menor número de atendimentos possuem US com menor distância em média das paradas mais próximas. Isso

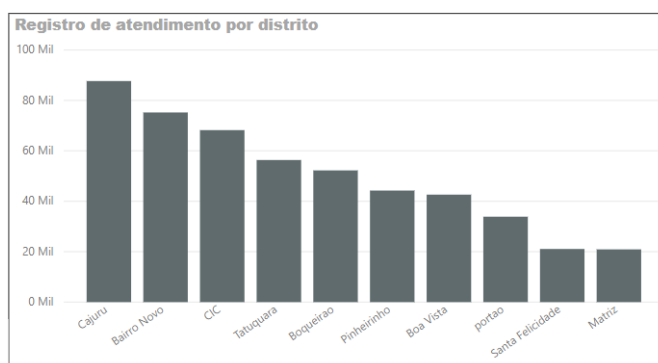


Figura 5. Número de atendimentos nas US por distrito sanitário

Tabela 6. Relação entre a média das distâncias (mínima) de cada US à parada de ônibus mais próxima e o número de atendimentos por distrito sanitário

Distrito sanitário	Média das distâncias mínimas	Desvio padrão	No. atendimentos
Boa Vista	105,3	58,3	42623
Cajuru	136,7	75,7	87673
Matriz	63,3	45,1	20910
Boqueirão	102,1	55,1	52233
CIC	101,2	87,0	68207
Portão	115,0	15,2	33874
Santa Felicidade	95,6	63,5	21069
Bairro Novo	108,3	43,4	75218
Pinheirinho	112,7	91,2	44244
Tatuquara	136,3	69,7	56352

pode significar que distritos com menor população são melhor atendidos pelo transporte público. Porém, outras análises são necessárias para verificar, por exemplo, se estes distritos apresentam populações concentradas em determinadas áreas. Como trabalho futuro, pretende-se considerar indicadores para tempo de viagem necessário para se deslocar até uma US e áreas de abrangência das US. A área de abrangência de uma US é aquela que pode ser atingida por uma linha de ônibus que possui uma parada próxima da US. Futuramente, pretende-se avaliar também a relação entre a qualidade do transporte público disponibilizado nas US e a situação sócio-econômica das populações que são atendidas.

Referências

- Aguiar, E. M. (1985). *Análise crítica dos indicadores de eficiência e eficácia propostos para avaliação de sistemas de transporte público urbano*. Tese de doutorado, EESC/USP, São Carlos.
- Ferraz, A. C. P. and Torres, I. G. E. (2004). *Transporte público urbano*. Rima Editora.
- Ferreira, R. V. and Raffo, J. G. (2013). Cartography of geographic access to rural health care services of the region of registro. *Confins-Revista Franco – Brasileira de Geografia*.

- Garcia, P. B. M. and Raia, A. A. (2015). Análise da acessibilidade a hospitais: Estudo de caso para as cidades de são carlos (sp) e rio claro (sp). *Revista Brasileira de Gestão Urbana*, 7:21–47.
- Gomide, A. A. (2003). Transporte urbano e inclusão social: Elementos para políticas públicas. *Brasília, IPEA*.
- Pilon, J. A. and Xavier, A. A. P. (2006). O emprego de microônibus na melhoria da acessibilidade ao sistema municipal de transporte coletivo da cidade de vitória-es. *XXVI ENEGEP - Fortaleza, CE*.
- Raia, A. A. and Pereira, C. A. (2001). Metodologia para simulação de mapas de equidade e acessibilidade a equipamentos e acessibilidade a equipamentos públicos urbanos. *Relatório de pesquisa São Carlos*.
- Raia, A. A. J. (2000). Acessibilidade e mobilidade na estimativa de um índice de potencial de viagens utilizando redes neurais artificiais e sistemas de informação geográfica. *Tese (Doutorado em Engenharia Civil)- Departamento de Transportes, USP/EESC, São Carlos*.
- Scatena, L. M., Villa, T. C. S., Netto, A. R., Kritski, A. L., Figueiredo, T. M. R. M., Vendramini, S. H. F., Assis, M. M. A., and Motta, M. C. S. (2009). Difficulties in the accessibility to health services for tuberculosis diagnoses in brazilian municipalities. *Rev Saúde Pública*.
- URBS (2019). Urbs em números <https://www.urbs.curitiba.pr.gov.br/transporte/rede-integrada-de-transporte>. Online; acesso em 09/03/2019.
- Vasconcellos, E. A. (1995). *A crise do planejamento de transportes nos países em desenvolvimento: reavaliando pressupostos e alternativas*, volume 3, pages 7–26.
- Vasconcellos, E. A. (2000). *Transporte urbano nos países em desenvolvimento*. Unida Editora, são Paulo.
- Vasconcellos, E. A. (2005). *A cidade, o transporte e o trânsito*. Pro livro, São Paulo.

A Framework for Scalable Data Analysis and Model Aggregation for Public Bus Systems

Mayuri A. Morais¹, Raphael Y. de Camargo¹

¹Centro de Matemática, Computação e Cognição – Universidade Federal do ABC (UFABC)

Rua Arcturus, 03 – 09606-070 – São Bernardo do Campo – SP – Brazil

{mayuri.morais, raphael.camargo}@ufabc.edu.br

Abstract. *Urban mobility through quality public transportation is one of the major challenges for the consolidation of smart cities. Researchers developed different approaches for improving bus system reliability and information quality, including travel time prediction algorithms, network state evaluations, and bus bunching prevention strategies. The information provided by these approaches are complementary and could be aggregated for better predictions. In this work, we propose the architecture and present a prototype implementation of a framework that enables the integration of several approaches, which we call models, into scalable and efficient composite models. For instance, travel time prediction models can use estimators of bus position, network state, and bus headways to deliver more accurate and reliable predictions. We evaluate the scalability of the framework, the CPU usage of the framework components, and the predictions of the travel time models. We show that real-time predictions using this framework can be feasible in large metropolitan areas, such as São Paulo city.*

1. Introduction

Providing efficient urban mobility is one of the major challenges facing large metropolitan centers today. An efficient way to reduce congestion is with the provision of quality public transport systems. Rail transport systems using trains and subways are very efficient, cover long distances, can connect cities close to capitals, are fast and generally suffer few delays and loss of time. However, the investments required for the creation or expansion of a railway line are high, with long maturation and grace periods (until the first line operation occurs). Thus, although rail transport is very efficient, its cost of implementation and extension is very high, so that most of the demand for travel in public transport is served by urban bus systems, which are more straightforward to implement and with lower cost.

Public transport systems using buses in Metropolis, such as São Paulo, are complex systems that continuously interact with the dynamics of the city [Cascetta 2009]. Buses are delayed due to congestion and overcrowding, as well as having their flow interrupted by traffic lights. Understanding the behavior of this system in different contexts, such as weekdays, hours of the day and holidays, is vital for better

This research is part of the INCT of the Future Internet for Smart Cities funded by CNPq proc. 465446/2014-0, Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001, FAPESP proc. 14/50937-1, and FAPESP proc. 15/24485-9.

planning of these systems. Dealing with this complexity is difficult and may require sophisticated models that encompass different aspects of the bus system, such as bus network state [Zhang et al. 2016], bus position estimation [Adachi et al. 2015], travel time predictions [Mori et al. 2015, Choudhary et al. 2016] and bus headway evolution [Yu et al. 2017]. Existing studies evaluate these aspects separately, despite the clear interactions between them, such as the influence of headway evolution on travel time predictions. There are only a few efforts in this direction of combining aspects of bus system analysis [Mazloumi et al. 2011].

One way to improve this interaction is by providing a framework for bus data collection, analysis, and visualization, which contains a set of models, each contemplating a specific aspect of the bus system. Such a system contains some lower level models, such as the graph representing the bus network and the estimation of the network link states (congestion level), bus positions and bus headways. On top of these, higher level predictions models, such as for travel time, bus headway evolution, and network link state evolution, can be built. Moreover, the framework should allow the deployment of different types of models for each aspect, with different computational complexities, which can be selected based on available computational resources and processing time requirements.

In this work we propose the architecture and a prototype implementation of this framework, containing a network model of one hundred bus routes from São Paulo city, an estimator for bus positions, and three travel time prediction models, which use historical and real-time data on bus positions. We evaluate the scalability of the framework with three workstations, the CPU usage of the framework components, and the predictions of the travel time models. We show that real-time predictions using this framework can be feasible even for large metropolitan areas, such as São Paulo city.

2. Related Work

In the literature we find two distinct types of predictions for bus travel time: predictions of total travel time and predictions of time / arrival time at a specific point, where efforts are more concentrated in this second type of prediction. Proposals for the prediction of total bus travel time were developed using Kalman filter [Chen et al. 2004], non-parametric regression [Chang et al. 2010] and SVM [Yu et al. 2010]. One of the first methodologies for predicting bus arrival time was proposed by [Lin and Zeng 1999] using regression and real-time GPS data. After this study, several other studies developed methods and models for the prediction of arrival time, using neural networks [Chien et al. 2002, Jeong and Rilett 2004], Kalman filter [Chen et al. 2004, Shalaby and Farhan 2004], *Support Vector Machine* (SVM) [Bin et al. 2006, Yu et al. 2008], *crowdsourcing* [Zhou et al. 2012] and *Finite State Machine* [Zuo and Wang 2013, Fu et al. 2014].

Some works propose combining different methods to achieve better predictions. [Mazloumi et al. 2011] proposed a framework using data from different sources to predict mean travel time and its variability with two neural networks, using these values to compose a Gaussian distribution. [Zhenliang et al. 2011] combined SVM and H-Filter to combine historical and real-time data to predict bus arrival time. [Kumar et al. 2017a] used the k-NN classification algorithm to identify typical patterns in historical data. The results of this algorithm were used in a Kalman filter to predict travel times. Similarly,

[Kumar et al. 2017b] used k-NN and Kalman filter for travel time predictions, but now applying an exponential smoothing factor to the model used in the Kalman filter. The smoothing factor parameters were dynamically estimated and updated using recent measures. Still extending the latter work, [Kumar et al. 2017c] developed a model using equations of velocity conservation in terms of the flow theory, applying a spatiotemporal discretization to these equations. The discretized model is applied in the Kalman filter to predict the speed in a connection.

All of the above works use a single bus line to carry out predictions. [Yu et al. 2011] proposed a model considering eight lines that go through a common link to make the prediction using SVM and data only in real time. [Yin et al. 2017] used historical travel time information from two bus lines on a common link to predict arrival time at a bus stop. [Gal et al. 2017] investigated the combination of methods of the Theory of Queues and Decision Trees for the prediction of travel times considering historical data and real-time for four lines simultaneously. They also considered dividing the bus route into links according to the bus stops.

These works all focus on travel time predictions and use ad-hoc integration of different types of models. Recently, [Zhang et al. 2016] proposed a complex citywide network of 261 bus lines but focusing their work on analyzing the static topological properties of the city bus network, and without travel time prediction method.

In our approach, we do not focus on a specific model for bus network modeling or travel time prediction. Instead, we propose a general framework for the integration of models, which allows easy development of new models that can be built over existing ones and that can scale over multiple machines.

3. Framework architecture

We propose a general framework for scalable and distributed processing of large amounts of historical and real-time data from thousands of bus lines. This framework would be useful for different applications, such as predicting bus travel times, evaluation of the flow along the bus routes, and preventing the occurrence of bus bunching.

3.1. Models

The core framework concept are `Models` (Figure 1), which can represent: (i) different views of the bus system state, pre-processed by some data analysis algorithm, such as estimated bus position, estimated links states, estimated bus headways; or (ii) predictions of future developments, such as travel time predictions in each link, evolution of link states, and likelihood of bus bunching occurrences.

There can be interdependence between models, such as the bus bunching prevention algorithm, which requires a model of the evolution of bus headways and estimation of bus positions and distances. In these cases, the model that requires the result from other models can use the results of the last execution of these models, or request a new execution of these models and wait for the results.

Models maintain information that can be used by other models and keeps this data in shared databases. After finishing its estimation or prediction algorithm, the model updates its state and writes the new state in shared databases, from which other models

can read. These shared databases improve processing scalability since it decouples the executions of different models.

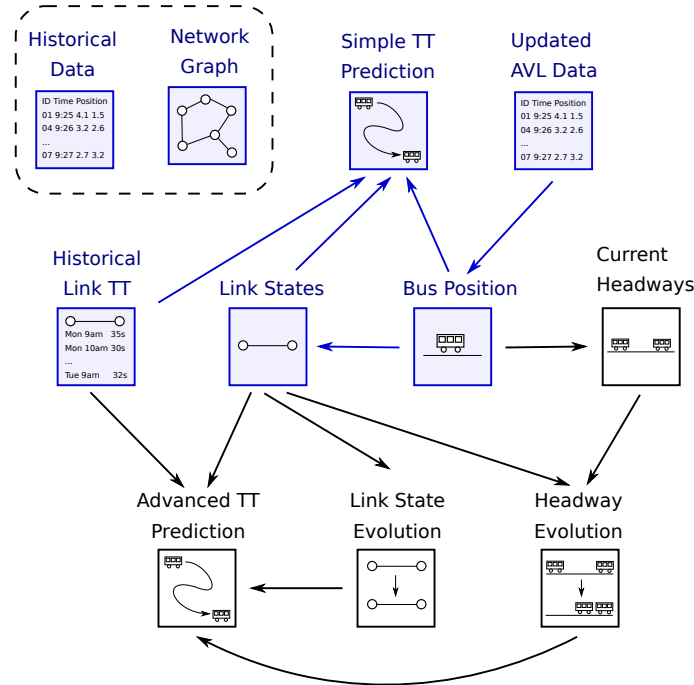


Figure 1. Set of models with direct data dependencies depicted with arrows. Historical Data and Network Graph models are used by several models and are showed separately. The models in blue are part of a possible sequence of model executions, triggered by an update in the AVL Data model.

3.2. Model Execution

The execution of each model is independent of other models, as each can have different execution schedules. This independence permits the execution of models of different complexities. For instance, there can be bus travel time prediction using simple heuristics, such as the mean travel time of the last three buses in the line, to more computationally demanding ones, such as executing deep neural networks models for predicting the evolution of link states in the following two hours.

The framework updates these models periodically, respecting some updating schedule, which can be periodical, for example, every 10 minutes, or started by some event, such as the retrieval of updated bus position information.

Large cities such as São Paulo may have tens of thousands of buses and to generate real-time predictions and estimations requires large computational power. The organization of the framework as independent tasks that updates models allows the processing to occur in distributed machines, which can also access distributed databases. The types of models that can be executed in real-time will depend on the hardware available, but the framework permits a natural expansion of processing capabilities by adding new computational resources, located in a private cluster or the cloud.

3.3. Bus System as a Graph

The `Network Graph` model maintains the representation of the bus system as a graph, with vertices representing points of interest, such as bus stops, and edges (links) representing the path between these points of interest. It keeps a list of links and their properties, such as length, position, and list of bus lines that use it.

Most models use this graph representation. For instance, the `Bus Position` model maps position estimations into graph links. Other models establish link properties, such as the mean historical travel time, the travel time of the last k -buses, the current link state, among others.

3.4. Types of Models

There are two main types of models: estimation and prediction models. Estimation models deal with the evaluation of the bus system current state and include models on bus positions, bus headways, and link states. Estimation models usually are fast to evaluate and, if required, can be computed whenever the bus positions are updated.

Predictions models deal with the prediction of future states of the model, such as bus travel times, bus bunching predictions and link state evolution. The framework can schedule the update of some simpler prediction models, such as the prediction of travel times from the current link state or from mean travel times, whenever bus positions are updated. However, for other models, such as predictions on the evolution of the link states or more complex bus travel time predictions, using neural networks or Hidden Markov Models, the framework may schedule their updates less frequently.

Additional models can also be added, such as trip planning models, which could generate trip proposals based on bus routes and schedules and travel time predictions, and data analysis models, which can gather bus bunching information and generates useful statistics of these events.

3.5. Composite Model Dependency Graph

Composite models can be constructed from simple models using shared databases. The framework updates the models when updated position data arrives. It then follows a dependency graph of model updates, respecting the dependencies in processing. It first updates the bus position estimation, then the models that depend on it, such as the link states and the bus headways. It then updates other models, such as bus travel time estimates that depend on this information.

The dependency graph can be long and contain branches, but not all models must update their state on every bus position update. For instance, complex models can generate new predictions about the evolution of links states in the next hours or about the city-wide evolution of bus headways. The framework can schedule these models for updating with a minimum interval of 10 minutes or more. If these models appear in a dependency graph but have updated their state more recently than the minimum update interval, the downstream models in the dependency graph use the data from the last update of these models. This design provides more flexibility for constructing dependency graph with models of different complexities and different update frequencies.

3.6. Input and Output Data

Real-time data can be obtained directly from APIs available at transit authorities and fed into the framework models. These data can generate updates in the bus position estimation model, which in turn can generate new updates to the states of the links, which causes new travel time estimates for buses.

Historical data usually is available as data files or in databases formats and contain several inconsistencies, missing fields, and other errors. At this moment, we are not dealing with the initial treatment of this data, but instead, consider that they have been already cleaned.

The framework can be accessed by external clients, which can request, for instance, predictions of travel times for bus lines, or the state of all the links from the city for visualization purposes. However, the purpose of the proposed framework is not to deal with individual requests from millions of users. In this case, we consider that an external system would perform this task, by periodically requesting the full model data to the framework and treating the individual requests locally.

4. Implementation

4.1. Distributed Execution using Dask

For the framework implementation we used Python with Dask.distributed, a lightweight library for distributed computing. This library consists of a centralized scheduler and distributed workers which can communicate with each other. Dask provides scalability by permitting the inclusion of more workers as demanded by the framework and supports complex workflows dependency graphs beyond those of map/filter/reduce.

The scheduler sends each task to a different worker which can be running on one or more machines (Figure 2a). Dask manages data transfer from the coordinator process to each worker. Each model stores the data it generates in a local or remote database, and models that use this data can access it directly on the databases. The databases are the primary interaction point among models, and we are using MongoDB non-relational database in the current implementation. Writes to single documents are atomic in MongoDB, preventing race conditions.

We implemented dependency graphs using *future* objects, which hold a promise of a result from a task. Each task executes a complete model or parts of it (Figure 2b). Tasks that depend on the results from other tasks wait for all upstream tasks to complete and then start its computation. Notice that the framework can execute many dependency graphs concurrently, exploiting the available computational resources. Moreover, when constructing the dependency graph, multiple workers can be assigned to a single task, improving the parallelization of the execution. Finally, models that have a minimum interval between executions can, if necessary, skip their execution, and the downstream tasks then get the results from the last execution from the shared database.

4.2. Implemented Models

We already implemented some models in the framework, which we used as a prototype implementation to validate the framework architecture.

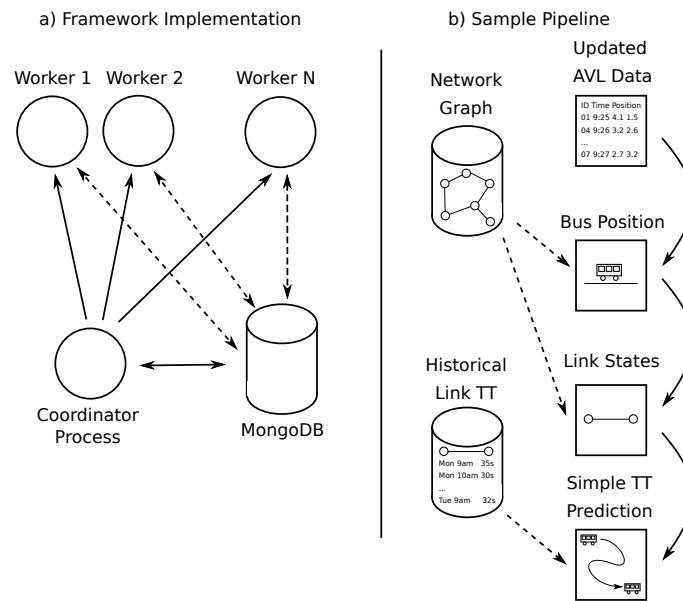


Figure 2. Distributed task execution and management using Dask. a) A coordinator process launches the tasks that are executed by workers (solid lines). The worker read data from other models and write their updated results in the MongoDB database (dashed lines). b) A sample graph of models. Models downstream can wait for other models to finish their execution (solid lines) or obtain data from other models that were already present in the database (dashed lines).

Bus Position: This model estimates the distance traveled by each bus in its predetermined route, interpolate this data and estimate the passing time at any arbitrary point of the bus route.

Graph Model: This model transform general transit feed specification (GTFS) files into a citywide graph containing all bus routes from a city. It defines the midpoints between bus stops of a bus route as vertices and the paths between consecutive vertices as links. The model first defines subgraphs for individual bus routes and then concatenates them into a citywide graph, identifying common links to multiple bus routes.

K-last Buses Model: This model determines the mean travel time of the last k buses that crossed a link in the city graph. The model retrieves data from the Bus Position model, querying for the positions of all buses in the last few hours. It then finds the last k travels on each link and calculates their mean travel times. Each call to this model evaluates a different link in the graph.

Mean Travel Time Model: This model evaluates the historical mean travel time of buses in all graph links. The model retrieves data from the Bus Position model, querying for the positions of all buses for a specific time range (e.g., from 9 am to 10 am), a specific weekday (e.g., Mondays), and a specific calendar range (e.g., from January 2017 to July of 2017). From this data, it estimates the time required for each bus to transverse the path

between the vertices of a link and then evaluates the mean travel time on the link for all buses. Each call to this model evaluates a different link in the graph.

Combined Model: This model combines the results of the Mean Travel Time and K-last Buses models by calculating a weighted mean between these two values, giving weight 1 to the historical mean and weight 2 to the K-last mean. In the current implementation, it waits for the *futures* from the Mean Travel Time and K-last Buses models.

Total Travel Time Model: This model predicts the total travel time in a bus route by summing the individual link travel times from the complete route. Each call to this model evaluates a single bus line, and in the current implementation, it waits for the *futures* from the corresponding models used for link travel time estimation.

4.3. Extending the Framework

One can include new models by writing new Dask tasks that perform the required model computations and access data from other models through the shared databases. Developers can create dependency graphs for composite models by making downstream tasks wait for the results of previous tasks in the graph using Dask *futures*.

The current framework prototype does not yet contain a model to collect real-time AVL data, although we easily implement an AVL Data model task that accesses an online API, such as SPTrans Olho Vivo¹. We could then construct a dependency graph, which would call the tasks that execute the Bus Position model, the link travel time models, and the Total Travel Time model.

We will also implement in the framework an API that permits external clients to perform queries to obtain model results, including travel time predictions and bus arrival times. The responses to the queries will be asynchronous to the model updates since they will only return the most up to date results.

5. Experimental Setup

We performed an experimental evaluation over the implemented prototype. We simulated a real-time travel time prediction scenario, using three travel time prediction algorithms, implemented in the K-last Buses Model, Mean Travel Time Model, and Combined Model, described in the Implementation section. We used $k = 3$ buses in the K-last Buses Model.

In the experiments, we did not execute the online retrieval of AVL data and the Bus Position estimation model. The models obtained the interpolated bus position data directly from a shared MongoDB database. These steps of obtaining AVL data and updating the bus positions should not take longer than the travel time predictions steps.

We executed a simulated real-time prediction of the total travel time of each bus, considering a set of 110 routes. We consider a single business weekday, from 6:00 am. to 11:50 pm., and generated a new prediction every 10 minutes. We used three machines, each with 2 CPUs model *Intel(R) Xeon(R) CPU E5-2620v2@2.10GHz* with 6 (12) physical (virtual) cores per CPU and 128GB of RAM. One machine held the shared database

¹<http://www.sptrans.com.br/desenvolvedores/>

and the task scheduler, with the workers distributed in up to the three machines (A, B, and C), depending on the evaluated scenarios.

First experiment: Execution time. We executed the framework with different setups to evaluate the processing time, CPU usage, and scalability of the framework. In all setups, we executed MongoDB and the scheduler in machine A. We considered scenarios with 6 to 72 workers, distributed as follows:

1. Machine A, with 6 workers;
2. Machine A, with 12 workers;
3. Machine A, with 24 workers;
4. Machines A and B with 48 workers, 24 on each;
5. Machines B and C with 48 workers, 24 on each;
6. Machines A, B, and C with 72 workers, 24 on each.

Note that in configuration 5 we used 3 machines since machine A was executing MongoDB and the scheduler.

We measured the execution time for the combined prediction of the bus travel from all lines. We evaluated predictions for a single time, an hour interval (6 predictions), and a complete day. The last two were used to evaluate the scalability better. We repeated the execution of each setup five times. During the execution, we also collected the total machine CPU usage and the MongoDB Server CPU usage.

Second Experiment: total travel time prediction accuracy. Besides the execution times, we also collected the predictions generated by the framework and compared with the actual bus travel times to measure the quality of the predictions. We collected the following predictions for 110 bus lines (to a total of 1977 links):

- Mean Travel Time Model: the model estimated the mean travel time (MTT) on each link using historical data from 7 months (January to July of 2017). For each link, the model gathered data from the corresponding weekday and full hour from MongoDB, and generated a mean predictor for every hour from 6:00 am. to 11:00 pm. for all graph links.
- K-last Buses Model: the model estimate the travel time on each link by obtaining the last $k = 3$ bus travels to cross each link in a 2-hour time frame. It then evaluated the k-last mean travel time (KTT) of all links. Different from the MTT, we generate a KTT prediction every 10 minutes from 6:00 am. to 11:50 pm.
- Combined Model: this model combines both MTT and KTT predictions using a weighted mean, with $mtt_{weight} = 1$ and $ktt_{weight} = 2$ to calculate the final prediction for every link.

We compared all bus travel time predictions to actual bus travel times from 110 bus lines in the August, 1st, chosen as the first weekday in August.

6. Experimental Results

6.1. Execution Time and Scalability

Our evaluations show that the processing time decreased as we increased the number of workers, except for 72 workers (Figure 3). When evaluating travel times at full hours (e.g.,

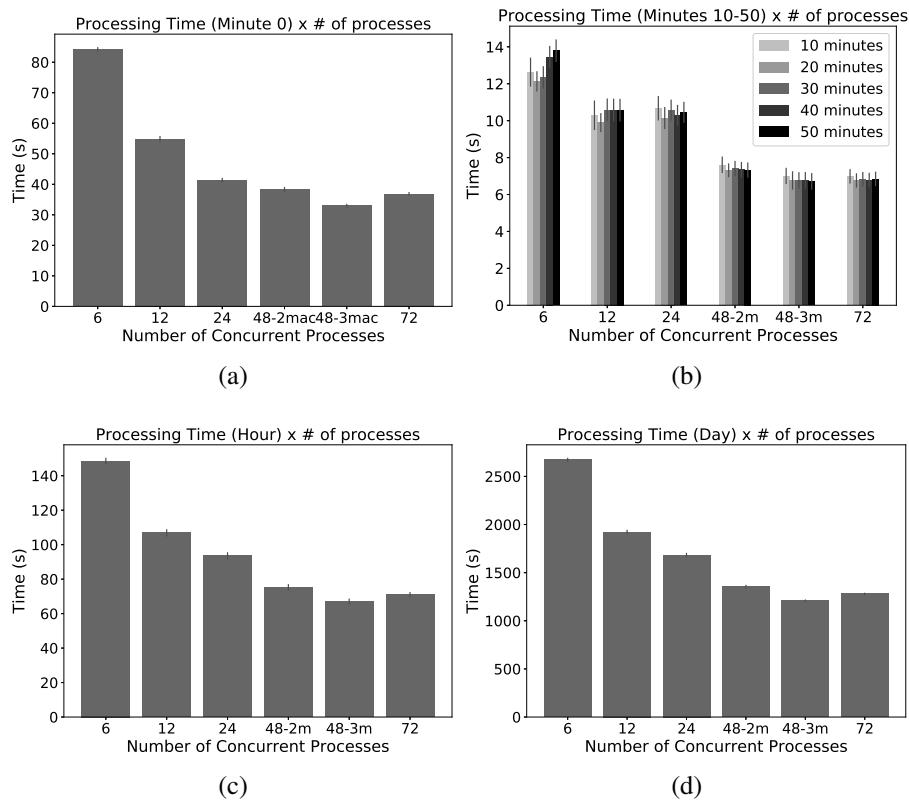


Figure 3. Execution time for predicting the travel times of 110 bus lines, using one machine (6, 12, and 24 workers), two machines (48 workers), and three machines (48 and 72 workers).

at 7 am), the framework executed both MTT and KTT models and required 40 seconds with using 24 workers (Figure 3a) or 30 seconds with 48 workers and three machines (48-3mac). When evaluating minutes 10 to 50, it executes only KTT and the processing time can be lower than 8 seconds for every 10 minutes (Figure 3b). We also evaluated the total time to calculate a whole hour (Figure 3c) and day (Figure 3d). We can see that it follows the same trend as the smaller computations.

The improvement in processing time is less evident when using more than 24 workers. To evaluate this phenomenon, we monitored the overall CPU usage on Machine A and Mongo CPU usage. The collected data shows that with up to 24 workers, running in the same machine, the CPU usage is slightly below 100% (Figure 4c), where 100% represent full usage of the 24 virtual cores in the machine, with MongoDB using about 75

The most interesting case is when machine A only executes MongoDB and the scheduler (Figure 4f), which was the configuration with the smallest total execution time. In this case, MongoDB used almost all CPU resources. When executing 72 workers (Figure 4d), MongoDB had to divide the CPU of machine A with workers and the total execution time increased.

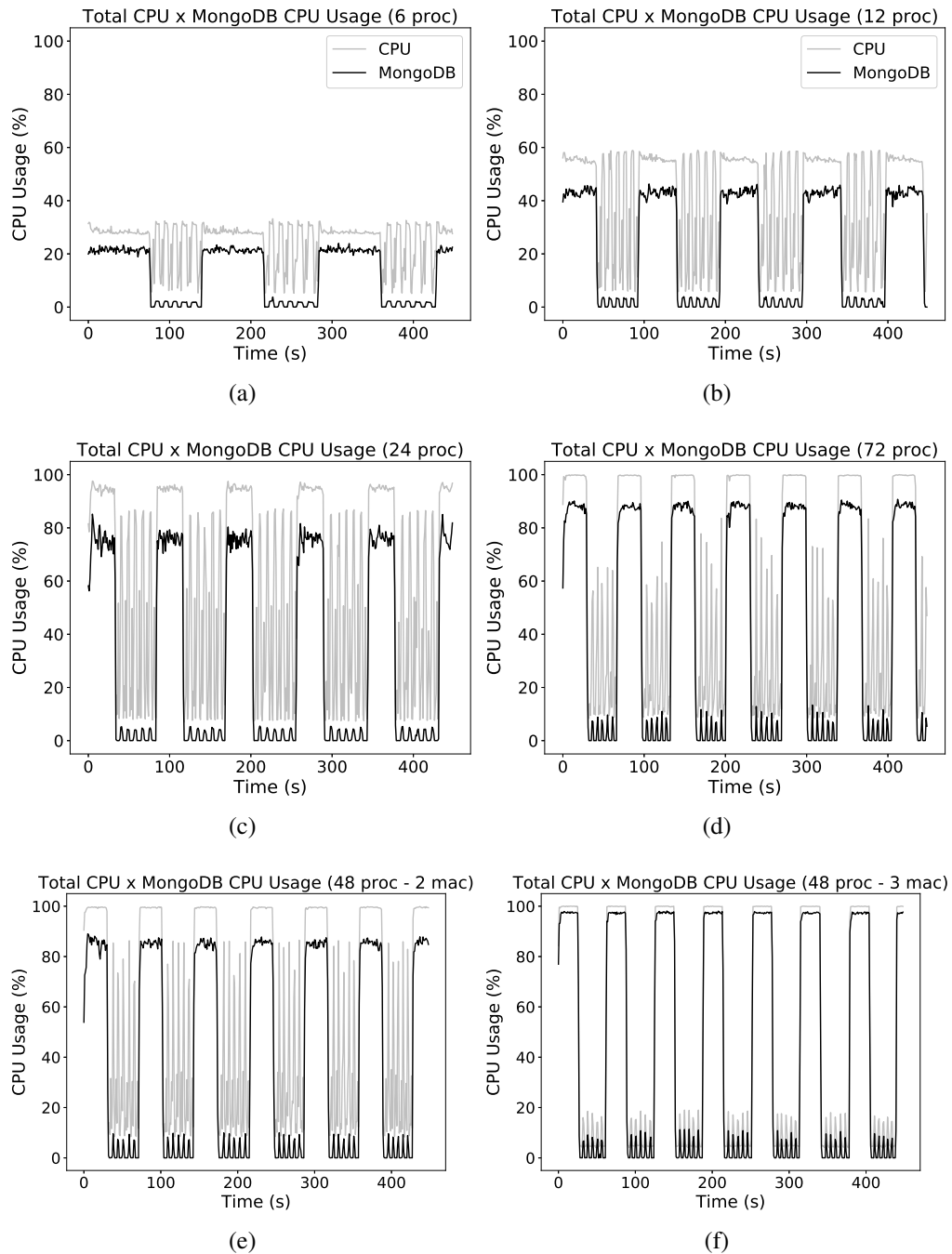


Figure 4. CPU usage measurement for machine A in the six experiments setups: (a) 6 workers, (b) 12 workers, (c) 24 workers, (d) 72 workers, (e) 48 workers in 2 machines, and (f) 48 workers in 3 machines

6.2. Travel Time Prediction Accuracy

In the third experiment, we collected the predictions generated by our prediction algorithms and compared to real travel time data, to evaluate the accuracy of these models. Note that our focus in this paper is the architecture and prototype implementation of the distributed framework and the prediction algorithms used here are rather simple. We calculated three measures: Rooted Mean Squared Error (RMSE), Mean Absolute Error

(MAE) (in minutes) and Mean Absolute Percentage Error (MAPE).

The model KTT performed slightly worse than MTT and combined models (Table 1). This performance probably happened because KTT used a few data points to make each prediction (only the latest three travels before the target travel). Although it captures better the real-time state of each link, it is more sensitive to errors in GPS measurements and unexpected behaviors, for instance, from an unusual delay in one of the latest three buses. Combining both MTT and KTT models with a weighted mean improved RMSE the predictions by a minimal margin.

Algorithm	RMSE (min.)	MAE (min.)	MAPE (%)
MTT	9.21	6.70	11.12
KTT	12.05	8.37	13.95
COMBINED	8.97	6.68	11.22

7. Discussion

We presented a framework for the scalable execution of composite models of public bus systems, describing its general architecture and a prototype implementation using Dask and MongoDB. We showed that it could successfully process data from more than one hundred routes in São Paulo in near real time, using a single node with 12 physical cores.

The experimental results indicate that the database can be a significant bottleneck in data dependent models, such as determining the historical mean travel times on network links. However, on more CPU-bound models, such as neural networks or hidden Markov models, the database may not be a critical bottleneck.

There are several possible solutions for improving the scalability of the database. A simpler one is to replicate MongoDB in multiple machines, with multiples copies of data that is highly accessed. For static data this extension is simple, but dynamic data may require updates on multiple sites to guarantee consistency. However, since the results of model processing generate smaller amounts of data, this may not be an issue. Another approach is to use columnar databases for storing data for some models that depend on the data aggregation, such as the mean travel times model. We are evaluating these approaches for applying in the framework.

References

- Adachi, H., Suzuki, H., Asahi, K., Matsumoto, Y., and Watanabe, A. (2015). Estimation of bus traveling section using wireless sensor network. In *2015 Eighth International Conference on Mobile Computing and Ubiquitous Networking (ICMU)*, pages 120–125. IEEE.
- Bin, Y., Zhongzhen, Y., and Baozhen, Y. (2006). Bus arrival time prediction using support vector machines. *Journal of Intelligent Transportation Systems*, 10(4):151–158.
- Cascetta, E. (2009). *Transportation Systems Analysis: Models and Applications*, volume 29. Springer US.

- Chang, H., Park, D., Lee, S., Lee, H., and Baek, S. (2010). Dynamic multi-interval bus travel time prediction using bus transit data. *Transportmetrica*, 6(1):19–38.
- Chen, M., Liu, X., Xia, J., and Chien, S. I. (2004). A dynamic bus-arrival time prediction model based on apc data. *Computer-Aided Civil and Infrastructure Engineering*, 19(5):364–376.
- Chien, S. I.-J., Ding, Y., and Wei, C. (2002). Dynamic bus arrival time prediction with artificial neural networks. *Journal of Transportation Engineering*, 128(5):429–438.
- Choudhary, R., Khamparia, A., and Gahier, A. K. (2016). Real time prediction of bus arrival time: A review. *Proceedings on 2016 2nd International Conference on Next Generation Computing Technologies, NGCT 2016*, (October):25–29.
- Fu, J., Wang, L., Pan, M., Zuo, Z., and Yang, Q. (2014). Bus arrival time prediction and release: System, database and android application design. In *International Conference on Algorithms and Architectures for Parallel Processing*, pages 404–416. Springer.
- Gal, A., Mandelbaum, A., Schnitzler, F., Senderovich, A., and Weidlich, M. (2017). Traveling time prediction in scheduled transportation with journey segments. *Information Systems*, 64:266–280.
- Jeong, R. and Rilett, R. (2004). Bus arrival time prediction using artificial neural network model. In *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*, pages 988–993. IEEE.
- Kumar, B. A., Jairam, R., Arkatkar, S. S., and Vanajakshi, L. (2017a). Real time bus travel time prediction using k-NN classifier. *Transportation Letters*, 7867(August):1–11.
- Kumar, B. A., Vanajakshi, L., and Subramanian, S. C. (2017b). A hybrid model based method for bus travel time estimation. *Journal of Intelligent Transportation Systems*, 2450(December):1–17.
- Kumar, B. A., Vanajakshi, L., and Subramanian, S. C. (2017c). Bus travel time prediction using a time-space discretization approach. *Transportation Research Part C: Emerging Technologies*, 79:308–332.
- Lin, W.-H. and Zeng, J. (1999). Experimental study of real-time bus arrival time prediction with gps data. *Transportation Research Record: Journal of the Transportation Research Board*, 1666:101–109.
- Mazloumi, E., Rose, G., Currie, G., and Sarvi, M. (2011). An integrated framework to predict bus travel time and its variability using traffic flow data. *Journal of Intelligent Transportation Systems*, 15(2):75–90.
- Mori, U., Mendiburu, A., Álvarez, M., and Lozano, J. A. (2015). A review of travel time estimation and forecasting for Advanced Traveller Information Systems. *Transportmetrica A: Transport Science*, 11(2):119–157.
- Shalaby, A. and Farhan, A. (2004). Prediction model of bus arrival and departure times using avl and apc data. *Journal of Public Transportation*, 7(1):3.
- Yin, T., Zhong, G., Zhang, J., He, S., and Ran, B. (2017). A prediction model of bus arrival time at stops with multi-routes. *Transportation Research Procedia*, 25:4627–4640.

- Yu, B., Lam, W. H., and Tam, M. L. (2011). Bus arrival time prediction at bus stop with multiple routes. *Transportation Research Part C: Emerging Technologies*, 19(6):1157–1170.
- Yu, B., Yang, Z. Z., and Wang, J. (2010). Bus travel-time prediction based on bus speed. In *Proceedings of the Institution of Civil Engineers-Transport*, volume 163, pages 3–7. Thomas Telford Ltd.
- Yu, B., Yang, Z.-Z., and Zeng, Q.-C. (2008). Bus arrival time prediction model based on support vector machine and kalman filter. *China Journal of Highway and Transport*, 2:016.
- Yu, H., Wu, Z., Chen, D., and Ma, X. (2017). Probabilistic prediction of bus headway using relevance vector machine regression. *IEEE Transactions on Intelligent Transportation Systems*, 18(7):1772–1781.
- Zhang, X., Chen, G., Han, Y., and Gao, M. (2016). Modeling and analysis of bus weighted complex network in qingdao city based on dynamic travel time. *Multimedia Tools and Applications*, 75(24):17553–17572.
- Zhenliang, M., Jianping, X., Shihao, Y., and Yubing, W. (2011). An Aggregation Method for Dynamic Bus Arrival Time Prediction. *Traffic*, 3(July):37–48.
- Zhou, P., Zheng, Y., and Li, M. (2012). How long to wait?: predicting bus arrival time with mobile phone based participatory sensing. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 379–392. ACM.
- Zuo, Z. and Wang, L. (2013). Bus arrival time forecasting and real-time information publication technology. *Journal of Transportation Systems Engineering and Information Technology*, 1:012.

Integração de Dados Públicos de Saúde e Transporte: Caracterização para Modelagem Multicamadas

Priscila L. L. Santin¹, Anelise Munaretto¹, Mauro S. P. Fonseca¹, Thiago H. Silva²

¹Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial (CPGEI)

²Programa de Pós-Graduação em Computação Aplicada

Universidade Tecnológica Federal do Paraná (UTFPR)
80.230-901 – Curitiba – PR – Brasil

{priscilasantin, anelise, maurofonseca, thiagoh}@utfpr.edu.br

Abstract. *The abundance of data from different contexts has been driving a revolution in the digital data industry. The information from these data can help public managers to provide better assistance to their citizens in different areas such as health care and urban mobility. Problems in the quality of public transport affect people's ability to travel, making it difficult for citizens to access basic health services provided by governments. This paper presents the characterization of the public transport and health system of Curitiba aiming at the employability of a MultiAspect Graph (MAG) data structure for the integration of these systems.*

Resumo. *A abundância de dados de diferentes contextos vem impulsionando uma revolução no setor de dados digitais. As informações provenientes desses dados podem auxiliar gestores públicos a prover melhor assistência aos seus cidadãos em diferentes áreas, como no cuidado à saúde e a mobilidade urbana. Problemas na qualidade do transporte público afetam a capacidade de deslocamento das pessoas, dificultando a acessibilidade dos cidadãos aos serviços básicos de saúde disponibilizados pelos governos. Este trabalho apresenta a caracterização do sistema público de transporte e de saúde de Curitiba visando a empregabilidade de uma estrutura de dados MultiAspect Graph (MAG) para a integração desses sistemas.*

1. Introdução

A quantidade de dados gerados, coletados e compartilhados por administrações públicas, empresas, entidades sem fins lucrativos e até mesmo pela comunidade científica vem aumentando consideravelmente nos últimos anos [Agarwal e Dhar 2014]. A análise dessas informações digitais e a inteligência para usá-las têm proporcionado uma significativa diferença nos negócios e na forma de viver das pessoas [Labrinidis 2012][Mervis 2012][Nature Editorial 2008]. Alguns trabalhos propõem maneiras pelas quais os governos podem utilizar-se de dados estratégicos para prover melhor atendimento aos seus cidadãos e lidar com outros desafios, tais como custos com cuidados de saúde, criação de emprego, desastres naturais e mobilidade urbana [Manyika et. al. 2011] [Pena 2018].

O debate sobre a mobilidade urbana no Brasil tem sido um tópico bastante discutido entre administradores públicos de grandes cidades. A mobilidade urbana refere-

se à facilidade de deslocamento da população e de bens no espaço geográfico das cidades, em função das complexas atividades nelas desenvolvidas, constituindo um componente da qualidade de vida aspirada por seus habitantes [Araújo et. al. 2011]. Segundo Pena (2018), a principal causa dos problemas de mobilidade urbana no Brasil relaciona-se ao aumento do uso de transportes individuais em detrimento da utilização de transportes coletivos, embora esses últimos também encontrem dificuldades com a superlotação. Neste contexto, problemas de infraestrutura e de qualidade do transporte comprometem a mobilidade e a capacidade de deslocamento das pessoas, dificultando a acessibilidade, ou seja, a facilidade em distância, tempo e custo em alcançar fisicamente os destinos desejados [Araújo et. al. 2011], tais como as unidades de saúde públicas.

Atualmente menos de um quarto da população brasileira possui plano de saúde privado [Oliveira 2017], o que significa que a maioria da população faz uso da rede pública de atenção à saúde. Dessa maneira, entender como acontece o deslocamento da população das grandes cidades até as unidades de saúde poderia contribuir com propostas que facilitariam o atendimento a esses cidadãos no que diz respeito à saúde, evitando, inclusive, que os mesmos utilizem o sistema de transporte público fazendo deslocamento desnecessário. Assim, este trabalho analisa o uso do transporte público e das unidades de saúde de Curitiba por meio dos registros dos usuários desses dois domínios. Para o transporte público foram analisados os registros dos cartões transporte (bilhetagem eletrônica) e o deslocamento da frota do sistema. A junção dessas informações foi realizada com o objetivo de aprimorar o entendimento dos dados estudados. No domínio da saúde pública, foram analisados os registros de atendimentos médicos das unidades de saúde da cidade.

Uma caracterização foi realizada nas bases de dados consideradas neste estudo de forma a entender melhor as questões referentes à utilização dos sistemas em questão, tais como o uso por períodos e faixas etárias, além da dinâmica de deslocamento por meio do sistema de transporte público de Curitiba. Entende-se que é fundamental verificar as propriedades desses sistemas para uma adequada integração de dados. Diante da caracterização realizada, o trabalho propõe a empregabilidade de uma estrutura de dados *MultiAspect Graph* (MAG) para analisar a dinâmica dos sistemas. Acredita-se que essa modelagem é fundamental para auxiliar na proposição de melhorias na mobilidade e no atendimento à saúde da população de grandes cidades, além de possibilitar a geração de dados sintéticos mais próximos da realidade dos sistemas estudados, possibilitando a simulação de usos em diversos cenários.

O restante do trabalho está organizado da seguinte forma. A seção 2 descreve alguns dos principais trabalhos relacionados, enquanto na seção 3 são apresentados os dados utilizados no trabalho e como foi realizado o tratamento desses dados. A seção 4 apresenta a análise e caracterização desses dados. A seção 5 apresenta a empregabilidade do MAG no modelo multicamadas incluindo a modelização do sistema nas bases estudadas. Por fim, a seção 6 apresenta as considerações finais.

2. Trabalhos Relacionados

A caracterização de dados se mostra relevante tanto para se obter um melhor entendimento dos sistemas analisados, bem como para aprimorar a qualidade do atendimento aos seus usuários. Nessa direção, o trabalho de Braz et. al. (2018) utiliza as bases do transporte público de Curitiba para estimar a ineficiência nos tempos de viagem

realizadas no sistema de ônibus da cidade. Já o trabalho de Ahmad (2019) apresenta um estudo direcionado à população idosa do Paquistão de modo a compreender as características de mobilidade existentes dos idosos, suas necessidades e fatores limitantes. O trabalho de Zhang et. al. (2018) propõem um quadro analítico para comparar os padrões de mobilidade urbana extraídos de duas fontes de dados, os cartões de transporte do sistema de transporte público de Singapura e as trajetórias dos táxis dessa cidade. Resumidamente, o estudo demonstra a importância de análise comparativa de dados para pesquisa urbana e de transporte.

Com a caracterização de dados é possível implementar de forma mais adequada modelos que representem os sistemas estudados. Nesse contexto, os modelos multicamadas têm sido empregados em diferentes estudos com esse objetivo. Kurant e Thiran (2006) apresentam em seu trabalho um modelo em camadas para facilitar a descrição e a análise de pequenas redes complexas, que são partes de sistemas maiores, onde uma série de topologias coexistentes interage e depende um do outro. Já Hristova et. al. (2016) apresentam uma rede interconectada de pessoas e lugares. Foram definidas quatro métricas pelos autores que concluem que a análise feita por eles espelha a relação entre a prosperidade das pessoas e dos lugares. O trabalho de Rodrigues et. al. (2017) propõem uma ferramenta chamada *SMAFramework* para analisar padrões de mobilidade baseados em MAG. Os autores utilizaram-se de duas fontes de dados diferentes e propuseram um algoritmo para análise de mobilidade.

Este presente trabalho se diferencia dos anteriores em diversos aspectos. Foi feita uma caracterização de dois importantes sistemas públicos de uma grande cidade brasileira, o que pode ser útil, por exemplo, para a geração de dados sintéticos mais realistas. Além disso, a caracterização permite encontrar de maneira mais apropriada a relação entre os dados analisados. Por exemplo, a dimensão temporal se mostra essencial para o problema estudado, indicando que a construção do modelo multi-aspecto MAG temporal é mais adequada. Esse também é um ponto de diferenciação de trabalhos relacionados anteriores, pois nenhum leva esse aspecto em consideração no modelo multicamadas.

3. Descrição dos Conjuntos de Dados

Nesta seção descrevemos as bases de dados estudadas, sua disponibilização e processamentos dos dados realizados para a busca do conhecimento.

3.1. Bases de Dados Referentes ao Transporte Público

O controle do sistema de transporte público de Curitiba é feito por uma empresa de economia mista, a URBS (Urbanização de Curitiba). É a URBS quem disponibiliza as bases de dados referentes ao transporte público utilizadas neste estudo.

3.1.1. Base do Cartão Transporte

A base de dados dos usuários do cartão transporte de Curitiba é uma base de dados disponibilizada exclusivamente para convênios com universidades e outras instituições conveniadas à URBS. Por meio desta base de dados são disponibilizados diariamente os

registros dos cartões transporte utilizados no dia anterior em arquivos *.json*¹. As informações contidas nessa base de dados estão distribuídas em 7 campos: Código da Linha, Nome da Linha, Código do Veículo, Número do Cartão, Data de Utilização, Data de Nascimento e Sexo.

3.1.2. Base do Transporte Público de Curitiba

Disponibilizada pela URBS para o público em geral, trata-se de uma base com diversos arquivos de informações, tais como as linhas de ônibus existentes e os pontos de ônibus que as linhas atendem, com GPS e a tabela horária dos veículos. Há dados estáticos, atualizados uma vez ao dia, e outros atualizados praticamente em tempo real. Todas essas informações são disponibilizadas por meio de arquivos *.json*. Neste estudo foram utilizados dois arquivos referentes ao transporte público de Curitiba: (I) *Veículos* – arquivo com a localização de todos os veículos que circularam por Curitiba, com informações do Código do Veículo, Latitude, Longitude, Data de Utilização e Código da Linha; (II) *Pontos* – relação de todos os pontos de ônibus existentes em Curitiba com informações como Nome do Ponto, Número do Ponto, Latitude, Longitude, Sequência, Grupo, Sentido, Tipo, Código da Linha.

3.1.3. Informações das Bases Referentes ao Transporte Público

A Tabela 1 apresenta um resumo das informações contidas nas bases de dados do transporte público estudadas neste trabalho.

Tabela 1 – Resumo das informações das bases do transporte público

<i>Descrição</i>	<i>Valor</i>
Número de leituras de cartões	22.703.308
Quantidade de cartões distintos	390.383
Quantidade de linhas operantes	407
Quantidade de pontos de ônibus	6.770

3.1.4. Aprimoramento dos Dados do Transporte Público

Alguns pré-processamentos foram realizados para que uma melhor análise do uso do transporte público fosse possível.

Como principal base de dados de estudos foi utilizada a base de dados dos cartões transporte, porém um aprimoramento foi feito a essa base acrescentando as coordenadas latitude e longitude, por meio do cruzamento das informações das bases apresentadas nessa seção. Além disso, com as informações de latitude e longitude inseridas foi possível identificar os bairros de registro do cartão transporte. Calculou-se, por meio da fórmula de *Haversin*² o ponto mais próximo da linha utilizada com base nas informações geoespaciais. Com as informações da latitude e longitude do ponto mais próximo, buscou-se por meio da função *geocode* do *Google Maps API*³ o bairro de localização daquele ponto.

¹ *JavaScript Object Notation*. Formato compacto, de padrão aberto independente, de troca de dados simples e rápida entre sistemas, que utiliza texto legível a humanos, no formato atributo-valor.

² $haversin\left(\frac{d}{R}\right) = haversin(\Delta\phi) + \cos(\phi_1) \cos(\phi_2) haversin(\Delta\lambda)$

³ <https://cloud.google.com/maps-platform/?hl=pt-BR>

3.2. Base de Dados da Saúde

Para atendimento médico, Curitiba possui atualmente 6 centros de especialidades, 112 unidades municipais de saúde (UMS), 9 unidades de pronto atendimento (UPA) e um hospital de atendimento exclusivo para idosos.

Obedecendo a Lei da Transparência⁴, a Prefeitura Municipal de Curitiba (PMC) disponibiliza na *Internet* a base de dados referente ao Sistema E-Saúde Médicos que contém o registro dos atendimentos médicos nas unidades municipais de saúde. Essa base é disponibilizada mensalmente e contempla o registro dos atendimentos dos três meses anteriores à sua publicação em formato *.csv*⁵.

3.2.1. Campos de Preenchimento das Bases

Cada registro da base possui 38 campos de preenchimento: Data do Atendimento, Data de Nascimento, Sexo, Código do Tipo de Unidade, Tipo de Unidade, Código da Unidade, Descrição da Unidade, Código do Procedimento, Descrição do Procedimento, Código do CBO, Descrição do CBO, Código do CID, Descrição do CID, Solicitação de Exames, Qtde Prescrita Farmácia Curitibana, Qtde Dispensada Farmácia Curitibana, Qtde de Medicamento Não Padronizado, Encaminhamento para Atendimento Especialista, Área de Atuação, Desencadeou Internamento, Data do Internamento, Estabelecimento Solicitante, Estabelecimento Destino, CID do Internamento, Tratamento no Domicílio, Abastecimento, Energia Elétrica, Tipo de Habitação, Destino Lixo, Fezes/Urina, Cômodos, Em Caso de Doença, Grupo Comunitário, Meio de Comunicação, Meio de Transporte, Município, Bairro, Nacionalidade.

Analisando as informações contidas nesses registros observa-se que alguns desses campos são de preenchimento obrigatório, já outros não. É possível observar também a duplicidade de informações. Pode-se citar como exemplo os campos ‘Código do Tipo de Unidade’ e ‘Tipo de Unidade’. Esses campos estão correlacionados, ou seja, há apenas um código para cada tipo de unidade.

3.2.2. Informações da Base

Na Tabela 2 é apresentado em resumo das informações obtidas na base da saúde estudada.

Tabela 2 – Resumo das informações da base da saúde

<i>Descrição</i>	<i>Valor</i>
Número de registros	960.062
Quantidade provável de pacientes distintos	355.697

Uma dificuldade encontrada ao trabalhar com a base de dados do Sistema E-Saúde que contém o registro dos atendimentos médicos nas unidades municipais de saúde é a falta de identificação do paciente. Desde 1º de março de 2012, o cadastro e uso do Cartão Nacional de Saúde (CNS) passou a ser obrigatório para usuários do Sistema Único de Saúde (SUS) segundo portaria do Ministério da Saúde⁶. Cada usuário do SUS possui um

⁴ Lei Nº 12.527, de 18 de novembro de 2011.

⁵ *Comma Separated Values*. Formato de arquivo texto que usa a vírgula para separar os dados de uma planilha permitindo a exportação a um programa editor de planilhas ou ainda utilizar alguma linguagem de programação para ler o arquivo e executar operações.

⁶ Portaria Nº 940, de 28 de abril de 2011.

número único de cartão, e a disponibilização desse número já contribuiria para a identificação precisa do número de pacientes distintos atendidos, assim como ocorre no trabalho de Jasmim [Jasmim et. al. 2017], ao invés do número de atendimentos.

Até o momento os dados disponibilizados pela Secretaria Municipal de Saúde de Curitiba não trazem a informação do número único do paciente. Assim, a técnica utilizada nesse trabalho combina informações referentes aos pacientes das unidades de saúde de Curitiba por inferência e calcula um valor único por meio de uma função *Hash*. Lembrando que uma função *hash* mapeia dados grandes e de tamanho variável em dados pequenos de tamanho fixo.

3.2.3. Processamento da Base da Saúde

Uma pré-análise dos registros da base considerada neste trabalho apontou que alguns desses registros são repetidos. Essas repetições às vezes são no mesmo instante de tempo, em outras ocorrem com diferença de pequenos segundos de tempo, porém apresentam o mesmo conteúdo em todos os demais campos preenchidos. Esses registros repetidos foram retirados da base, e representavam menos de 0,2% do total de registros da base. A Tabela 2 já traz no campo ‘Número de registros’ a quantidade sem as repetições.

4. Caracterização das Bases de Dados

Nesta seção identificamos as principais características do uso do transporte público e das unidades de saúde de Curitiba por meio de seus usuários.

4.1. Caracterização do Uso do Transporte Público

De posse dos dados do cartão transporte e realizados os processamentos informados na seção 3.1.4 foi possível obter uma visão da utilização do transporte público de Curitiba ao longo dos 3 meses analisados.

Observou-se que o mês de agosto/2018 foi o que apresentou o maior número de registros (325.728). Já os meses de setembro/18 e outubro/18 apresentaram uma pequena queda nos registros (294.756 e 312.965, respectivamente). Esse comportamento pode ser justificado se observarmos que esses dois últimos meses contemplam dias de feriados, reduzindo o uso do transporte público.

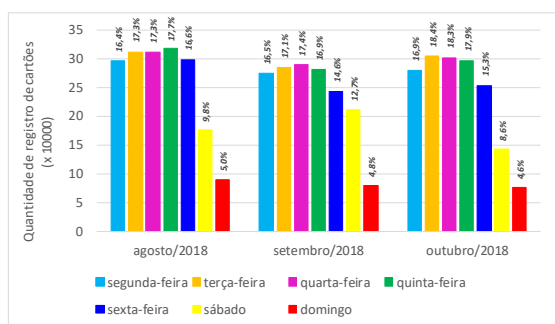


Figura 1 – Utilização de cartões transporte no período de análise

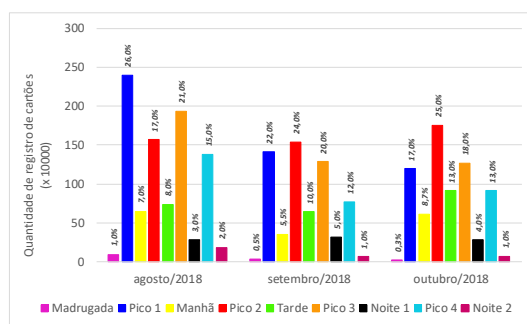


Figura 2 – Utilização de cartões transporte por período

Ainda com relação à utilização do transporte público por meio dos cartões transporte, observa-se na Figura 1 o comportamento do usuário ao longo dos dias da semana. Pode-se verificar que os dias de maior utilização são os dias úteis da semana.

Observa-se também uma queda maior nas sextas-feiras do mês de setembro/18 e outubro/18, coerente com os dias de feriado desses meses (Independência do Brasil, Dia de Nossa Senhora da Luz dos Pinhais – padroeira de Curitiba, Nossa Senhora Aparecida – padroeira do Brasil e Dia dos Professores).

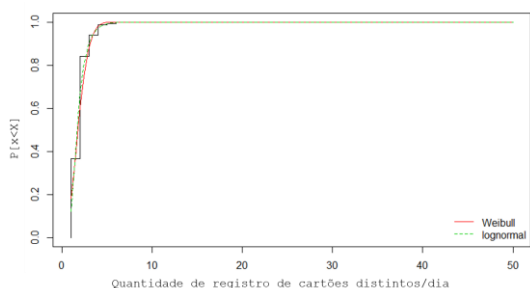


Figura 3 – Registro de cartões por dia

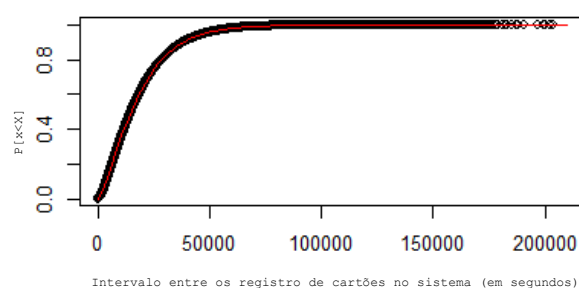


Figura 4 – Intervalo entre os registros de cartões

A Figura 3 mostra a CDF (*Cumulative Distribution Function*) da quantidade de registros de cartões distintos para um mesmo dia. Percebe-se que 80% dos usuários do cartão transporte se registram até 2 vezes por dia no sistema. Observa-se também que há o registro de até 50 vezes no sistema por um mesmo cartão, um padrão inesperado para a utilização do transporte público. A distribuição *Lognormal*⁷ ($\mu = 0.525$ e $\sigma = 1.492$) ou a distribuição *Weibull*⁸ ($\alpha = 2.143$ e $\delta = 2.122$) melhor representam os dados reais analisados. Já a Figura 4 mostra a CDF do intervalo entre os registros sucessivos de um mesmo cartão. Observa-se que cerca de 75% dos usuários de cartão transporte faz o uso do sistema com um intervalo de 8h. Esse valor corrobora com os dados levantados estatisticamente, já que a grande maioria dos usuários do cartão transporte são os adultos, que na média trabalham 8h por dia. O intervalo médio entre os registros de um mesmo cartão é melhor representado por uma distribuição *Gamma* ($\alpha = 1.53$ e $\beta = 8.32e-05$).

4.1.1. Horários de Uso do Transporte por Período

A URBS organiza o transporte público de Curitiba conforme o horário. Atualmente existem 9 períodos considerados pela gestão dos transportes: Madrugada, Pico 1, Manhã, Pico 2, Tarde, Pico 3, Noite 1, Pico 4 e Noite 2.

A Figura 2 apresenta o número de registros de cartões por período de utilização. Observa-se que mesmo havendo variância quanto ao número de registros, os horários de maior utilização do sistema de transporte público de Curitiba obedecem à distribuição de períodos estabelecidos pela URBS, o que vem ao encontro também do aumento da frota nestes períodos, com a operação de contingência, que pode ser observada na base de dados do cartão transporte.

Vale salientar aqui que não se trata de leitura de cartões distintos, uma vez que um mesmo cartão pode ser registrado repetidas vezes num pequeno intervalo de tempo.

$$^7 f(x) = \frac{1}{x\sigma\sqrt{2\pi}} \exp\left(-\frac{(\ln x - \mu)^2}{2\sigma^2}\right)$$

$$^8 f(x) = \frac{\delta}{\alpha^\delta} x^{\delta-1} \exp\left[-\left(\frac{x}{\alpha}\right)^\delta\right]$$

Mesmo o cartão transporte sendo de uso pessoal e intrasferível⁹, observa-se na Figura 4 que há repetidos registros com intervalo de tempo de poucos segundos para um mesmo cartão.

4.1.2. Uso do Transporte por Sexo e Faixa Etária

Cerca de 60% do uso do transporte público de Curitiba é feito por pessoas do sexo feminino, conforme pode ser observado na Figura 5. Analisando os dados do DETRAN-PR (Departamento de Trânsito do Paraná) [DETRAN 2016], 65% de pessoas habilitadas para dirigir são do sexo masculino no levantamento de 2016, o que talvez justifique o maior número de mulheres utilizando o transporte público.

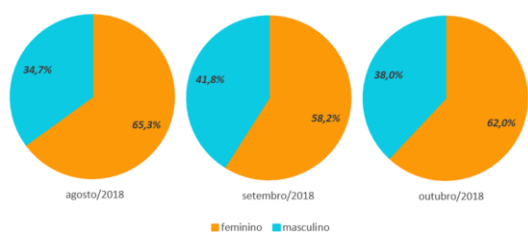


Figura 5 - Utilização de cartões transporte por sexo

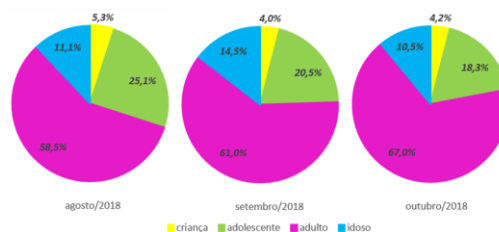


Figura 6 - Utilização de cartões transporte por faixa etária

Para uma interpretação ainda melhor da utilização do transporte público, para esse estudo, a população de Curitiba foi dividida nas 4 faixas etárias, baseadas no Estatuto da Criança e do Adolescente [ECA2018] e no Estatuto do Idoso [EI2018]: Crianças, Adolescentes, Adultos e Idosos. Na Figura 6 observa-se o número de cartões distintos por faixa etária. As crianças representam o menor número de cartões. Cabe ressaltar que até os 6 anos de idade elas possuem gratuidade no uso do transporte público, e costumam não passar pelas catracas dos veículos. Mesmo assim, há na base de dados dos cartões transporte registros de crianças com idade inferior aos 6 anos. Tais registros se justificam por essas crianças possuem algum tipo de benefício para o uso do transporte público, como gratuidade por necessidades especiais, devendo então já possuírem o cartão. As crianças são seguidas pelos Idosos na utilização de cartões. Neste caso, mesmo os Idosos maiores de 65 anos, que por lei¹⁰ possuem gratuidade no uso do transporte, devem possuir o ‘Cartão Isento’ para se utilizarem do sistema. Ao verificar a utilização do sistema de transporte público observou-se que cerca de 1% dos registros não incluíam a data de nascimento do usuário. Esses registros foram desconsiderados nessa caracterização pois a data de nascimento é uma das chaves de identificação do usuário do sistema.

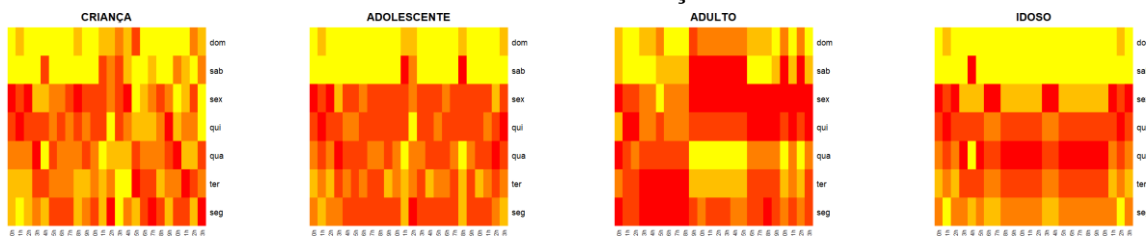


Figura 7 - Utilização do transporte por dia, horário do dia e faixa etária

⁹ <https://www.urbs.curitiba.pr.gov.br/utilidades/cartao-transporte>

¹⁰ Artigo 230 da Constituição Federal Brasileira.

A Figura 7 apresenta um resumo do uso do transporte público por faixa etária, baseado nos dias da semana e horários dos dias. O gradiente das cores varia do amarelo (menor valor) ao vermelho (maior valor). Observa-se que nos finais de semana há uma redução no uso do transporte público por todas as faixas etárias, com exceção dos adultos, que fazem um maior uso entre o período compreendido entre 9h e 15h. Ao apresentar a utilização do sistema separado por horário e faixa etária o gradiente de cores utilizado foi o mesmo para todas as faixas etárias. Para análise da Figura 7 deve-se observar o percentual de pessoas em cada faixa etária para que não haja erro de interpretação. Se observarmos que dentro do total de usuários do sistema de transporte apenas 5% dele representa as crianças, ou seja, em torno de 20.000 crianças, o maior gradiente das crianças representa, percentualmente, um número bem pequeno se comparado às demais faixas etárias, e pode ser justificado como um aumento de demanda pontual para aquele período. Da mesma forma acontece com os adultos, uma pequena variação da utilização aparenta visualmente uma grande queda no gradiente.

4.2. Caracterização do Acesso à Serviços de Saúde

Os mesmos 3 meses analisados na base do transporte público foram analisados para o sistema de saúde de Curitiba. Observou-se que o mês de agosto/2018 foi o que apresentou o maior número de registros (345.242). Já os meses de setembro/18 e outubro/18 apresentaram uma pequena queda nos registros (308.135 e 306.685, respectivamente).

4.2.1. Horários de Uso do Sistema de Saúde

Por convenção, as horas de um dia são divididas em quatro períodos: Madrugada, Manhã, Tarde e Noite.

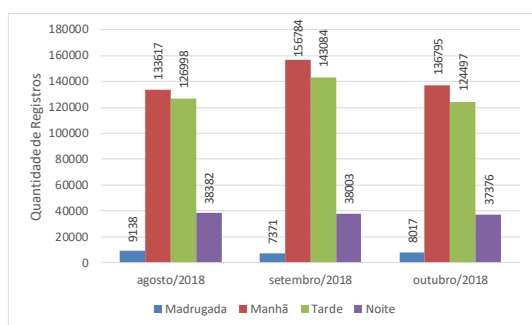


Figura 8 – Registro de Atendimentos por período

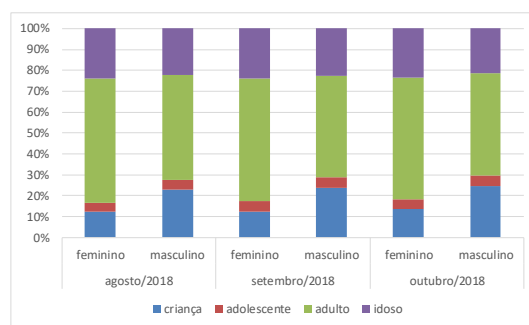


Figura 9 – Atendimentos por faixa etária e por sexo

A Figura 8 apresenta o número de registros de atendimentos por período de utilização do sistema. Observa-se que os horários de maior registro de atendimentos ocorrem no período da manhã e da tarde. Isso pode ser explicado, porque são nesses horários que a maioria das Unidades Municipais de Saúde estão em seu período de funcionamento. Nos períodos da madrugada e da noite somente algumas unidades, e as UPAs, é que se encontram abertas ao público.

4.2.2. Uso do Sistema de Saúde por Sexo e Faixa Etária

Cerca de 60% dos registros de atendimento de saúde de Curitiba são feitos por pessoas do sexo feminino, conforme pode ser observado na Figura 10. Esse comportamento também foi verificado em estudo do IBGE (Instituto Brasileiro de Geografia e Estatística)

[IBGE16], onde mostra que foi observado esse percentual maior de procura por atendimento entre as mulheres.

Na Figura 11 observa-se número de registros por faixa etária. Os adolescentes representam o menor número de atendimento. O maior número de registros de atendimentos é para os adultos, seguidos dos idosos e por fim das crianças.

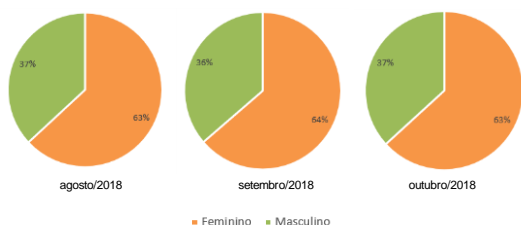


Figura 10 – Utilização do sistema de saúde por sexo

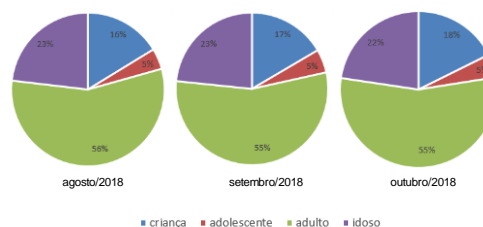


Figura 11 - Atendimentos por faixa etária

Ao fundir as informações dos registros por sexo e faixa etária (Figura 9), observa-se que as crianças apresentam um comportamento inverso ao número de atendimento por sexo. Neste caso, há um maior registro de atendimento para crianças do sexo masculino em todos os meses observados. Segundo dados do IBGE [IBGE19], anualmente há um número maior de nascimento de crianças do sexo masculino. É também na infância que há uma maior dependência dos pais, responsáveis por proporcionar o atendimento à saúde de seus filhos.

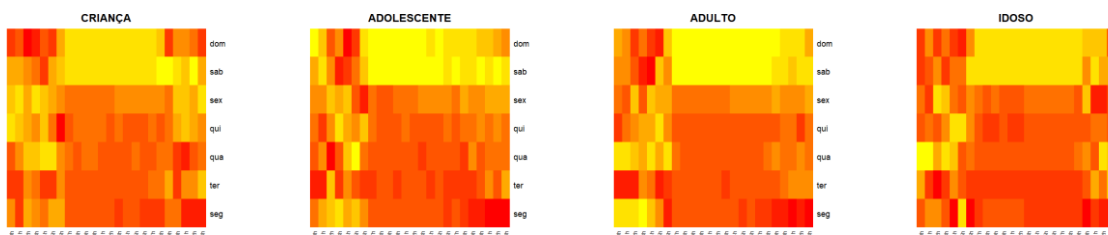


Figura 12 - Utilização do transporte por dia, horário do dia e faixa etária

A Figura 12 apresenta um resumo do registro de atendimentos por faixa etária, baseado nos dias da semana e horários dos dias. Observa-se um comportamento padrão entre as faixas etárias quanto a busca de atendimento à sua saúde, sendo que nos finais de semana há uma redução no número de registros para todas as faixas etárias, ficando os dias úteis e períodos matutinos e vespertinos com maior utilização do sistema.

5. Utilizando o *MultiAspect Graph* para a Criação de um Modelo Multicamadas

Este trabalho também busca a integração de diferentes bases de dados para a criação de um modelo multicamada. A caracterização é importante para a escolha dos aspectos do *MultiAspect Graph*, representando melhor a dinâmica dos sistemas.

5.1. MultiAspect Graph

O *MultiAspect Graph* (MAG) é uma generalização de grafos capaz de representar redes variantes no tempo, redes multicamadas e redes simultaneamente multicamadas e variantes no tempo. Há diferentes descrições do MAG na literatura, porém os modelos

propostos por Kivelä et. al. (2014) e Wehmuth et. al. (2016) são mais bem aceitos pela comunidade acadêmica.

Uma rede multicamada apresenta um conjunto de vértices V igualmente a um grafo, porém acrescentam-se outras camadas. É permitido que cada vértice pertença a qualquer camada, e as arestas podem combinar qualquer conexão entre vértices ou camadas. Além dessa propriedade, o MAG permite a representação de diferentes características dos dados por meio de aspectos. O aspecto pode ser um conjunto de vértices, camadas, instantes de tempo ou qualquer outro recurso independente. Em uma rede multicamada vários aspectos podem ser incluídos. Desta forma, não é possível restringir a rede em um único conjunto de camadas. Por exemplo, em uma rede onde o primeiro aspecto é o tipo de interação e o segundo aspecto é o tempo, é necessário ter um conjunto de camadas para tipos de interação e um segundo conjunto de camadas para o tempo.

Como se pretende construir um modelo multicamadas, as camadas do modelo são formadas por informações extraídas das bases estudadas nesse trabalho e apresentadas na seção 3. Por se tratar de um modelo heterogêneo, cada camada pode ser formada por dados descrevendo aspectos diferentes, tais como localização geográfica, usuário, dados de especialidade ou ainda um híbrido entre esses aspectos. Esse modelo utiliza-se de localização geográfica, identificação do usuário e análise no tempo para criar a relação entre as camadas.

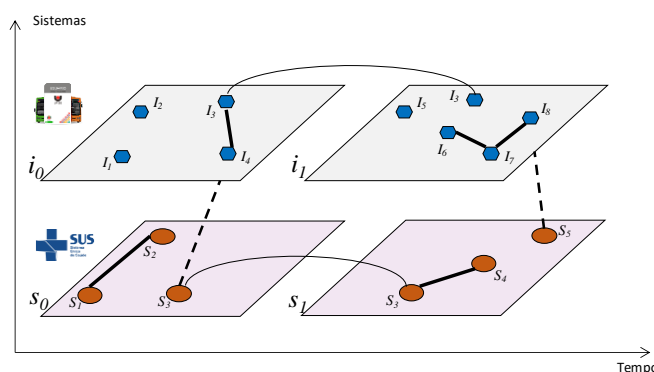


Figura 13 – Representação de um MAG para as bases de estudo

O caso de uso apresentado na Figura 13 traz a representação de um MAG baseado nos dados desse estudo. Neste mapeamento são apresentados três tipos de relacionamentos em dois aspectos (sistemas e tempo). A linha contínua fina representa a movimentação de um vértice no aspecto tempo (no caso de uso, um paciente do sistema de saúde e um usuário do sistema de transporte). A linha tracejada representa o vértice em uma camada diferente. Aqui é importante ressaltar que os dados trabalhados neste estudo não possuem essa informação explicitamente. A conexão para esse aspecto deve ser criada utilizando-se de métodos que avaliem característica similares, localização GPS e compatibilidade de horário entre os sistemas. Por fim, neste modelo, a linha contínua grossa representa a relação dentro da mesma base de dados. Acreditamos que essas três representações no MAG são importantes para o objetivo deste estudo.

5.2. Descrição Formal do Modelo

Para o modelo, os dados em uma camada devem conter os seguintes atributos: t : intervalo de tempo em que os dados foram criados ou observados; l : localização GPS onde os dados

foram gerados ou observados, sendo possível também trabalhar com um raio r a partir da localização informada para minimizar erros; e : informações especiais; u : identificador do usuário que gerou os dados.

A Figura 14 apresenta o mapeamento para a aplicação do MAG. A cidade de Curitiba fornece os dados públicos de uso dos sistemas de saúde e transporte. Esses dados são coletados e armazenados num Banco de Dados de Entrada. Os registros armazenados no banco de dados contêm as informações das bases descritas na seção 3. Todos os registros são organizados temporalmente, e após o pré-processamento apresentado na seção 3.1.4, os dados do cartão transporte são enriquecidos com as informações de latitude e longitude provenientes das informações dos ônibus, formando um novo registro.

Após organização, processamento e caracterização dos registros, esses são armazenados no Banco de Dados do MAG para criar a estrutura MAG apresentada na Figura 13. Os vértices dessa estrutura MAG são elementos formados conforme a seguinte descrição. Seja $U = \{u_1, u_2, u_3, \dots, u_n\}$ um conjunto de usuários de um sistema. No mapeamento proposto neste trabalho trata-se do sistema de transporte e do sistema de saúde, assim, seja $O = \{o_1, o_2, o_3, \dots, o_n\}$ um conjunto de tipo de transporte, neste caso o ônibus, e $C = \{c_1, c_2, c_3, \dots, c_n\}$ um conjunto que represente informações de deslocamento, provenientes das informações do cartão transporte. Cada usuário $u_i \in U$ utiliza-se de um tipo de transporte $o_j \in O$ compartilhando informações de deslocamento $c_k \in C$. Cada n -ésima informação de deslocamento compartilhada é representada pela tripla $I_n^{c_k} = \langle t, u, o \rangle$, onde t se refere ao registro de data e hora quando o usuário $u_i \in U$ do sistema de transporte utiliza-se do ônibus $o_j \in O$. De maneira análoga, o m -ésimo usuário de um sistema de saúde é representado pela tripla $S_m^{u_i} = \langle t, l, e_x \rangle$, onde t se refere ao registro de data e hora em que esse usuário $u_i \in U$ foi registrado em uma das unidades de saúde l , apresentando alguma informação especial de atendimento e_x . Observa-se que as triplas $I_n^{c_k}, S_m^{u_i}$ são os vértices de cada uma das camadas i e s , representadas na Figura 13.

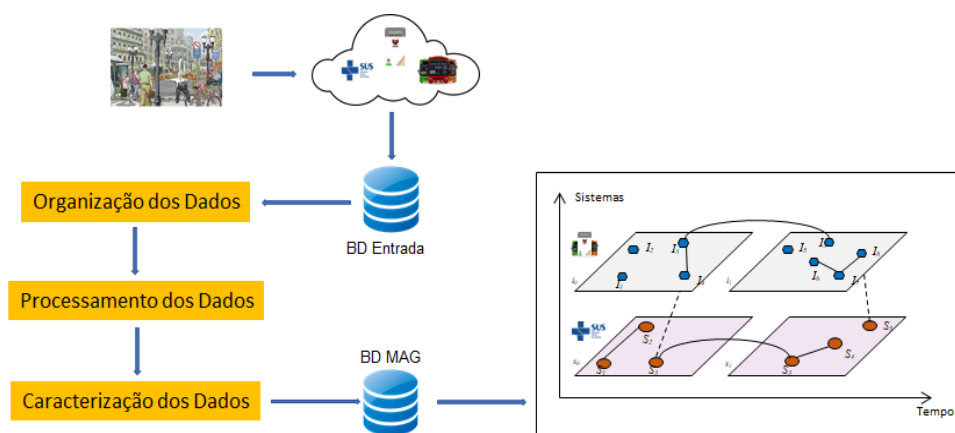


Figura 14 – Mapeamento para a Criação da Estrutura MAG

A rede multicamadas formada contém 2 aspectos que contém uma sequência de conjuntos de camadas elementares. Camada elementar representa um elemento do conjunto de camadas, ou seja, há um único conjunto de camadas elementares L_a para cada aspecto a ($L_{s_0}, L_{s_1}, L_{i_0}$ e L_{i_1}). O conjunto de todas as camadas no grafo é o produto cartesiano de todas as camadas elementares $L_{s_0} \times L_{s_1} \times L_{i_0} \times L_{i_1}$. Qualquer um dos vértices $V = \{I_n^{c_k}, S_m^{u_i}\}$ pode estar em qualquer camada, assim, o conjunto de vértices que

estão presentes no grafo é representado por $V_M \subseteq V \times L_{s_0} \times L_{s_1} \times L_{i_0} \times L_{i_1}$. As arestas do grafo devem conectar vértices diferentes em qualquer camada ou aspecto. Desta forma, o conjunto de aresta do modelo é representado por $E_M \subseteq V_M \times V_M$. Com essas informações, o modelo possui a forma $M = (V_M, E_M, V, L)$.

Essa representação no MAG é um passo chave para entender como acontece o deslocamento da população das grandes cidades até as unidades de saúde. Isso pode contribuir com propostas que podem facilitar o atendimento a esses cidadãos no que diz respeito à saúde, desde a implantação de especialidade médica nas unidades, evitando o deslocamento desnecessário, ou até mesmo a criação de uma nova rota de ônibus na busca de um atendimento especializado.

6. Considerações Finais

Esse trabalho estuda a utilização dos sistemas de transporte público e de saúde da cidade de Curitiba. Uma caracterização desses sistemas foi realizada durante um período de 3 meses, entre agosto e outubro de 2018. Esse estudo inicia apresentando os principais resultados acerca da caracterização dos dados dessas duas grandes e representativas bases de estudos da cidade de Curitiba. Finalmente, esse estudo também apresenta como contribuição uma formalização do MAG para implementação do modelo pretendido. Como trabalhos futuros, pretende-se utilizar o modelo proposto para auxiliar na resposta de perguntas complexas relacionadas com a relação entre o sistema de transporte público e de saúde, como propagação de doenças na cidade entre outras.

7. Referências

- Agarwal, R.; Dhar, V.. “Editorial – Big Data, Data Science, and Analytics: The Opportunity and Challenge for IS Research”. *Information Systems Research*, Volume 25, Issue 3, pg. 443–448. September 2014.
- Ahmad, Z.; Batool, Z.; Starkey, P.. “Understanding mobility characteristics and needs of older persons in urban Pakistan with respect to use of public transport and self-driving”. *Journal of Transport Geography*. Volume 74, January 2019, Pages 181-190.
- Anuário Estatístico 2016 – DETRAN-PR. Disponível em <<https://bit.ly/2Wpsbdd>>. Acessado em 28 de novembro de 2018.
- Araújo, M. R. M. et. al. “Transporte público coletivo: discutindo acessibilidade, mobilidade e qualidade de vida”. *Psicologia & Sociedade*, Volume 23, pg. 574-582. 2011.
- Braz, T. et. al. “Estimating Inefficiency in Bus Trip Choices From a User Perspective With Schedule, Positioning, and Ticketing Data”. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 19, No. 11, November 2018.
- Estatuto da Criança e do Adolescente. Disponível em <<https://bit.ly/1NqqzW6>>. Acessado em 14 de dezembro de 2018.
- Estatuto do Idoso. Disponível em <<https://bit.ly/2CFDyqa>>. Acessado em 14 de dezembro de 2018.

- Hristova, D. et. al. “Measuring Urban Social Diversity Using Interconnected Geo-Social Networks”. Proceedings of the 25th International Conference on World Wide Web, pg. 21-30. April 16.
- Instituto Brasileiro de Geografia e Estatística. Pesquisa Nacional de Saúde. Disponível em <<https://biblioteca.ibge.gov.br/visualizacao/livros/liv97329.pdf>>. Acessado em 24 de janeiro de 2019.
- Instituto Brasileiro de Geografia e Estatística. Sistema IBGE de Recuperação Automática. Disponível em <<https://sidra.ibge.gov.br/tabela/2679>>. Acessado em 04 de fevereiro de 2019.
- Jasmim, L. O.; Ziviani, A. et. al. Caracterização de Atendimentos em uma Rede de Atenção à Saúde. 17º Workshop de Informática Médica. Julho 2017.
- Kivelä, M. et. al. “Multilayer networks”. Journal of Complex Networks, Volume 2, Issue 3, pg. 203–271. September 2014.
- Kurant, M.; Thiran, P.. “Layered Complex Networks”. Physical Review Letters, Volume 96. April 2006.
- Labrinidis, A.; Jagadish, H. V.. “Challenges and Opportunities with Big Data”. Proceedings of the Very Large Data Bases Endowment, Volume 5, Issue 12, pg. 2032-2033. August 2012.
- Manyika, J. et. al. “Big Data: The next frontier for innovation, competition, and productivity”. Available in <<https://goo.gl/2jX7W5>>. Acessado em 17 de janeiro de 2018.
- Mervis, J.. U.S. Science Policy: Agencies Rally to Tackle Big Data. Science, Volume 336, Issue 6077, pg. 22. April 2012.
- Nature Editorial. Community cleverness required. Nature, Volume 455, Issue 7209, pg. 1. September 2008.
- Oliveira, N.. Gastos com saúde crescem mesmo em meio à crise e atingem 9,1% do PIB. Agência Brasil. Disponível em <<https://goo.gl/4ijGnF>>. Acessado em 03 de abril de 2018.
- Pena, R. F. A. Mobilidade urbana no Brasil. Brasil Escola. Disponível em <<https://goo.gl/n3CkrS>>. Acessado em 03 de abril de 2018.
- Rodrigues, D. O. et. al. SMAFramework: Urban Data Integration Framework for Mobility Analysis in Smart Cities. 20th ACM International Conference on Modelling, Analysis and Simulation of Wireless and Mobile Systems. November, 2017.
- Wehmuth, K.; Fleury, E.; Ziviani, A.. On MultiAspect graphs. Theoretical Computer Science, Volume 651, pg. 50–61. 2016.
- Zhanga, X.; Xub, Y.; Tuc, W.; Rattid, C.. Do different datasets tell the same story about urban mobility — A comparative study of public transit and taxi usage. Journal of Transport Geography 70 (2018).

Uma Infraestrutura para o Monitoramento e Predição de Rotas e Paradas de Ônibus no Transporte Universitário

Paulo Miranda e Silva Sousa¹, José Robertty de Freitas Costa¹
Abilio Castro e Silva¹, Wilton Ribeiro Crispim Neto¹
Carla Ilane Moreira Bezerra^{1,2}, Emanuel Ferreira Coutinho^{1,2}

¹Campus de Quixadá – Universidade Federal do Ceará (UFC)
63.902-580 – Quixadá – CE – Brasil

²Programa de Pós-Graduação em Computação (PCOMP)

{paulomirandamss, abilio, robertty, wiltonribeiro}@alu.ufc.br

{carlailane, emanuel.coutinho}@ufc.br

Abstract. *Due to the growth of urbanization, cities have faced social, economic and environmental transformations. In addition, many vehicles currently have several sensors and actuators, capable of performing not only the sensing of the condition of vehicles, but also the environment around them, and this data can be used for various services. The environment of a large university may resemble urban environments, considering that these institutions compare to cities in various aspects, especially in relation to infrastructure problems. The objective of this work is to develop an infrastructure for the monitoring and prediction of bus routes and stops in university transportation. Tests were performed on real bus routes, and the results, even if preliminary, were attractive and have potential for research.*

Resumo. *Devido ao crescimento da urbanização, cidades têm enfrentado transformações sociais, econômicas e ambientais. Além disso, muitos veículos atualmente possuem diversos sensores e atuadores, com capacidade de executar não só o sensoriamento da condição dos veículos, mas também do ambiente ao seu redor, sendo que esses dados podem ser utilizados para diversos serviços. O ambiente de uma universidade de grande porte pode se assemelhar com ambientes urbanos, considerando que essas instituições se comparam às cidades em vários aspectos, principalmente em relação aos problemas de infraestrutura. O objetivo deste trabalho é desenvolver uma infraestrutura para o monitoramento e predição de rotas e paradas de ônibus no transporte universitário. Foram realizados testes em rotas de ônibus reais, e os resultados, mesmo que preliminares, se mostraram atrativos e com potencial para pesquisa.*

1. Introdução

A partir do crescimento populacional e da urbanização, cidades de maneira geral têm enfrentado transformações sociais, econômicas e ambientais, tais como o aumento da desigualdade social, acesso limitado aos serviços públicos básicos, problemas de mobilidade urbana e segurança [de M. Neves et al. 2017]. Neste contexto, o conceito de Cidades Inteligentes (*Smart Cities*) surge como um novo paradigma que visa encontrar

soluções sustentáveis para esses problemas crescentes. Diversas soluções foram propostas e desenvolvidas para os campi universitários, considerando que essas instituições se comparam às cidades em vários aspectos, principalmente em relação aos problemas de gestão, falta de infraestrutura e consequente insatisfação dos seus usuários [Sungur et al. 2015, Saad et al. 2018, Feng et al. 2018].

Atualmente, muitos veículos possuem uma gama muito grande de sensores e atuadores, com capacidade de executar não só o sensoriamento da condição dos veículos, mas também do ambiente ao seu redor [Rodrigues et al. 2018]. As informações coletadas por esses sensores podem ser utilizadas para diversos serviços, que poderão levar um maior conforto e segurança aos passageiros e aos próprios condutores do veículo.

A cobertura de toda uma área urbana com uma rede de sensores pode ser proibitivamente custosa [Cruz et al. 2018]. Uma opção para a redução dos custos e ainda assim obter uma significativa cobertura espacial é utilizar nós móveis de sensoriamento, onde cada nó é responsável por sensoriar uma região. Uma forma mais barata de prover mobilidade nos nós de sensoriamento é embarcá-los em ônibus urbanos [Castro et al. 2017].

Em um cenário ainda mais específico, onde alguns campi universitários possuem convênios com empresas ou prefeituras para auxílio em algum tipo de serviço, como transporte de alunos, uma solução que otimizasse rotas e paradas, gerenciasse horários e lotação dos ônibus, ou aumentasse ou diminuísse a frota conforme a necessidade, seria de grande valor para a comunidade acadêmica [Feng et al. 2018].

Nesse contexto, o objetivo deste trabalho é desenvolver uma infraestrutura para o monitoramento e predição de rotas e paradas de ônibus no transporte universitário. A partir dessa infraestrutura, informações sobre o transporte dos universitários de duas universidades, Universidade Federal do Ceará (UFC) e Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE), ambas localizadas na cidade de Quixadá, Ceará, serão tratadas em uma aplicação e sistema embarcado. Também será possível apresentar por meio da infraestrutura desenvolvida a localização em tempo real dos ônibus universitários e prever o horário do próximo ônibus em uma determinada parada. Foram utilizadas técnicas de aprendizado de máquina para prever o horário do próximo ônibus em um determinado ponto de parada, obtendo-se uma margem de erro muito pequena a partir dos dados coletados nas rotas.

O artigo está dividido nas seguintes seções, descritas a seguir. Na Seção 2, são apresentados conceitos aplicados a este trabalho. Na Seção 3, são apresentados alguns trabalhos relacionados presentes na literatura. Em seguida, na Seção 4, é detalhada a solução proposta para o sistema de monitoramento e predição do transporte universitário. Na Seção 5, são apresentados o planejamento do experimento, a execução dos testes, os resultados e análises do experimento. Por fim, a Seção 6 apresenta as conclusões deste trabalho.

2. Fundamentação Teórica

Nesta seção, apresentamos os principais conceitos para um melhor entendimento do presente trabalho.

2.1. Microcontrolador ATmega328p

Os microcontroladores são pequenos circuitos integrados que possuem memória, processador, periféricos de entrada e saída e entre outros componentes. Os sistemas microcontrolados estão cada dia mais presentes em diversos equipamentos do nosso cotidiano. Segundo [Martins 2005], os microcontroladores estão presentes em quase tudo o que envolve a eletrônica, diminuindo o tamanho, facilitando a manutenção e gerenciando tarefas internas de aparelhos eletroeletrônicos. A redução de custos é um dos principais fatores que contribuem para a popularização e utilização dos microcontroladores.

Neste trabalho, utilizamos o microcontrolador ATmega328p¹ pelo fato de possuir diversas plataformas de desenvolvimento e facilidade de gravação. A plataforma Arduino foi escolhida para o desenvolvimento da aplicação. A plataforma, apresenta um ambiente de desenvolvimento de algoritmos próprio e *upload* através da porta USB de qualquer computador, tornando prático o desenvolvimento de aplicações.

2.2. Aprendizado de Máquina

Aprendizado de Máquina é uma área da computação que tenta prever situações futuras tendo como base informações anteriores. Segundo [Monard and Baranauskas 2003], o objetivo da área de Aprendizado de Máquina é o desenvolvimento de técnicas computacionais sobre o aprendizado bem como a construção de sistemas capazes de adquirir conhecimento de forma automática.

É notório que a área de Aprendizado de Máquina está em constante crescimento e também são encontradas diversas aplicações para a mesma. Esta pesquisa irá trabalhar com aprendizado supervisionado, em específico, com algoritmos de regressão para tentar prever o horário do próximo ônibus em uma determinada parada.

Os algoritmos de aprendizado de máquina supervisionado testados foram *Linear Regression*, *Decision Tree*, *Random Forest* e o *XGBoost*. O modelo de predição foi feito na linguagem de programação Python, onde os três primeiros algoritmos são da biblioteca do *scikit-learn* e o último algoritmo da biblioteca *xgboost*. Esses algoritmos tem se mostrado bastante eficientes em diversos problemas de aprendizado. Muitos desses algoritmos já são utilizados em diversos trabalhos da literatura que lidaram com problemas semelhantes ao que se propõe esse trabalho. Um exemplo disso é o trabalho [Yamaguchi et al. 2018] que tenta prever o tempo de viagem de ônibus em diversas rotas distintas. Para a avaliação do modelo de aprendizado final foram utilizados as métricas MAE (*Mean Absolute Error*) e RMSE (*Root Mean Square Error*), afim de determinar qual melhor algoritmo a ser usado.

3. Trabalhos Relacionados

Existem na literatura diversas soluções para sistemas IoT de monitoramento do transporte urbano, como por exemplo, os sistemas de monitoramento de ônibus. No trabalho de [Castro et al. 2017] são apresentados diversos trabalhos da literatura para sistemas de monitoramento de ônibus, onde são destacadas as tecnologias usadas, implementação, dificuldades e problemas nesses sistemas. Dentre as tecnologias utilizadas pelos trabalhos

¹http://microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf

identificados destaca-se o GPS (*Global Positioning System*) e o GSM (*Global System for Mobile Communication*). Estes trabalhos utilizam o GPS como a tecnologia de rastreamento em conjunto com GSM como tecnologia de comunicação. A utilização conjunta entre GPS e GSM também será utilizada neste trabalho.

Dentre estas aplicações de monitoramento urbano, têm se destacado os sistemas de monitoramento de ônibus dos campi universitários, pelo fato dessas instituições se compararem às cidades em vários aspectos, principalmente em relação aos problemas de gestão, falta de infraestrutura e consequente insatisfação dos seus usuários [Sungur et al. 2015, Miranda et al. 2017, Saad et al. 2018, Feng et al. 2018].

[Sungur et al. 2015] apresentam um sistema inteligente de informações sobre passageiros com paradas para permitir que administradores monitorem o sistema de transporte público e possibilite pessoas que utilizam este sistema observem simultaneamente as informações sobre a localização e o estado desses veículos. Como estudo de caso, utilizou-se as paradas em um campus universitário e outros pontos urbanos da cidade. As tecnologias utilizadas neste trabalho foram um dispositivo embarcado ARM Micro2440 e a posição do GPS, onde os dados são coletados e mantidos em um banco de dados.

No trabalho [Miranda et al. 2017] é apresentado a construção de um sistema de localização de ônibus com registro automático de rotas e acesso em tempo real através de um aplicativo móvel com custo acessível, onde são utilizados os conceitos de sistemas embarcados, redes e geolocalização para aplicar o paradigma de IoT como uma forma de ajudar e aumentar o uso de transporte público em geral. A solução utilizada no trabalho foi *Amazon AWS* para *back-end* em nuvem, unificando os dados transmitidos pelo GPS do ônibus para aplicativos Android. O terminal instalado no ônibus consiste em um Raspberry PiTM usando um receptor GPS e o módulo HSDPA para transmissão de dados. Os dados transmitidos são recebidos no aplicativo Android e são plotados em tempo real usando *Google Maps*TM, permitindo ao usuário saber onde está o seu ônibus desejado, e até mesmo saber o seu ponto em sua rota. A solução proposta em [Miranda et al. 2017] se assemelha com a proposta deste trabalho, pois existem algumas diferenças entre módulos utilizados no hardware que compõe o sistema embarcado mas a funcionalidade de tais módulos em relação à solução aqui apresentada é a mesma.

No trabalho [Saad et al. 2018] é apresentado um sistema de monitoramento em tempo real dos ônibus dos campi. Semelhante a este trabalho, é utilizado o GPS na infraestrutura proposta para capturar dados como latitude, longitude e velocidade. Esse sistema, assim como muitos da literatura, tenta prever o tempo de chegada nas paradas dos ônibus baseado em somente distância e velocidade do ônibus. Como um diferencial da infraestrutura proposta neste trabalho, são utilizados algoritmos de aprendizado de máquina para previsão.

O trabalho de [Feng et al. 2018] realizam um estudo sobre a aplicação da localização de nós da IoT e a conscientização de localização dos ônibus de um Campus Universitário para melhoria da previsão da chegada dos ônibus e integra tecnologias RFID e Zigbee de redes sem fio em um sistema de barramento inteligente. O sistema pode localizar e monitorar o estado de execução do barramento em tempo real, fornecendo para que o barramento colete as informações do tempo nas estações, e melhoria da qualidade, eficiência e satisfação do serviço de ônibus.

Sistemas de transportes inteligentes possibilitam a utilização de sensores e GPS no monitoramento dos sistemas de transportes públicos em cidades inteligentes [Neves et al. 2018]. A captura e processamento desses dados permite, a princípio, a utilização do transporte público com confiabilidade e previsibilidade, o que melhoraria a qualidade de vida da população urbana. Entretanto, diversos fatores provocam que esses dados sejam insuficientes ou de baixa qualidade para uso em tempo real. O trabalho analisa dados obtidos por meio de colaboração coletiva (*crowdsourcing*) para complementar informações. Além disso, o trabalho propõe um modelo de análise de confiabilidade dos dados coletados para o sistema de transporte público (por ônibus) do município de São Paulo.

A dinâmica espacial e social de uma cidade sofre constantes mudanças ao longo do tempo [Wen et al. 2018]. Em consequência, a malha viária e o sistema de transporte público necessitam ser otimizados continuamente para atender a diversas demandas. Uma alternativa para reduzir custos e impactos na avaliação de soluções é o emprego de simuladores com modelos consistentes com a realidade. Este trabalho processou 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 escalável para cidades inteligentes. Um modelo de movimentação baseado em dados reais do serviço de ônibus de São Paulo foi apresentado para tornar o simulador mais eficaz ao recriar cenários de mobilidade urbana.

As condições climáticas adversas representam um fator emblemático afetando negativamente a qualidade do transporte público, especialmente em regiões sujeitas a chuvas [Vianna et al. 2018]. Este trabalho apresenta relações entre eventos de chuva e o comportamento da velocidade média dos ônibus de transporte público na cidade do Recife. Para isto, utilizou-se estatística descritiva na análise dos dados do itinerário, posicionamento e velocidade dos ônibus em relação aos dados de precipitação em estações pluviométricas. Análises de locais conhecidos por problemas de trânsito em dias de chuva foram detalhadas no artigo.

4. Infraestrutura para Monitoramento e Predição de Rotas e Paradas de Ônibus no Transporte Universitário

A solução proposta para a infraestrutura de monitoramento e predição de ônibus no transporte universitário foi dividida em três fases: (i) prototipação do sistema embarcado; (ii) obtenção dos dados; e, (iii) construção do produto final.

De maneira geral, na primeira fase do projeto construiu-se um protótipo do sistema embarcado visando testar os componentes do sistema e o código desenvolvido. Com o protótipo em funcionamento, iniciou-se a obtenção dos dados da rota. Por fim, em posse dos dados, um modelo de aprendizado de máquina supervisionado foi projetado e desenvolvido, e o produto final foi construído.

Na Figura 1, pode-se visualizar a arquitetura do sistema e como cada parte é interligada. No elemento Sistema Embarcado é apresentado o dispositivo responsável por emitir os dados utilizados pelo sistema. Esse dispositivo possui um módulo GPS responsável por capturar dados como coordenadas (latitude e longitude), horário e velocidade. Outro módulo desse sistema é o GSM/GPRS. Tal módulo é responsável por enviar os dados capturados pelo GPS via requisição POST para um servidor WEB. Para enviar

os dados, utiliza-se um chip nano SIM que permite fazer o envio dos dados via internet da rede telefônica. Esses módulos são controlados pelo microcontrolador *Atmega 328p* da *Microchip Technology*.

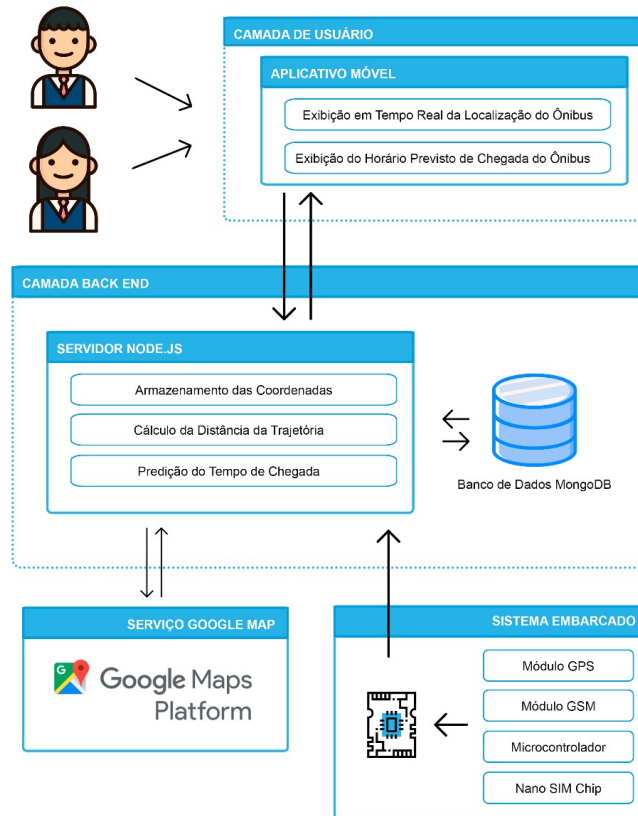


Figura 1. Arquitetura do Sistema

Outra camada importante do sistema é o *Back End*. Essa camada é responsável por receber todas as requisições POST do sistema embarcado e armazená-las em um banco de dados Mongo. Além disso, é nessa camada que é efetuada a predição do tempo de chegada do próximo ônibus em uma determinada parada e todo o suporte necessário para a aplicação do usuário.

Comunicando-se diretamente com o *Back End*, tem-se a camada de usuário. Tal camada ainda encontra-se em fase de desenvolvimento. Esta possuirá uma aplicação *mobile* que exibirá em tempo real a localização dos ônibus e o horário de chegada do próximo ônibus. Além disso, nessa aplicação, as informações sobre o transporte universitário serão concentradas em um *feed* do aplicativo.

Por fim, a camada *Google Maps Service* dará suporte tanto ao *Back End* quanto ao usuário. No *Back End*, serão extraídas informações da cidade como, por exemplo, a menor distância entre dois pontos que respeita todas as sinalizações das ruas. Na aplicação do usuário será utilizada a API do *Google Maps* para que os usuários do aplicativo tenham uma visão mais amigável da posição atual do ônibus, podendo acompanhar em tempo real o deslocamento do mesmo.

Na Figura 2 são apresentados os componentes do sistema embarcado e como eles estão interligados. No rótulo 1 tem-se a antena utilizada pelo módulo GSM/GPRS. No rótulo 2 tem-se um regulador de tensão que converte 6V-12V para 3.3V e 5V. Em 3, é visualizado o módulo GPS *VK2828U7G5LF TTL* que é responsável pelos dados de coordenada, hora e velocidade. Em 4 é possível identificar o módulo GSM/GPRS *SIM 800L v.2* que é responsável por enviar uma requisição POST para um servidor com as informações do módulo GPS. No rótulo 5 tem-se o microcontrolador Atmega 328p responsável por gerenciar todo o sistema. Por fim, em 6, tem-se 3 LEDs (Diodo Emisor de Luz) que são utilizados para exibir o estado atual do sistema. LED vermelho ligado significa que o sistema está sem conexão de internet. LED amarelo aceso sozinho, significa que o dispositivo está aguardando dados do GPS. LED amarelo e verde ligado, os dados estão tentando ser enviado pelo GSM/GPRS. LED verde aceso sozinho, os dados foram enviados com sucesso para o servidor.

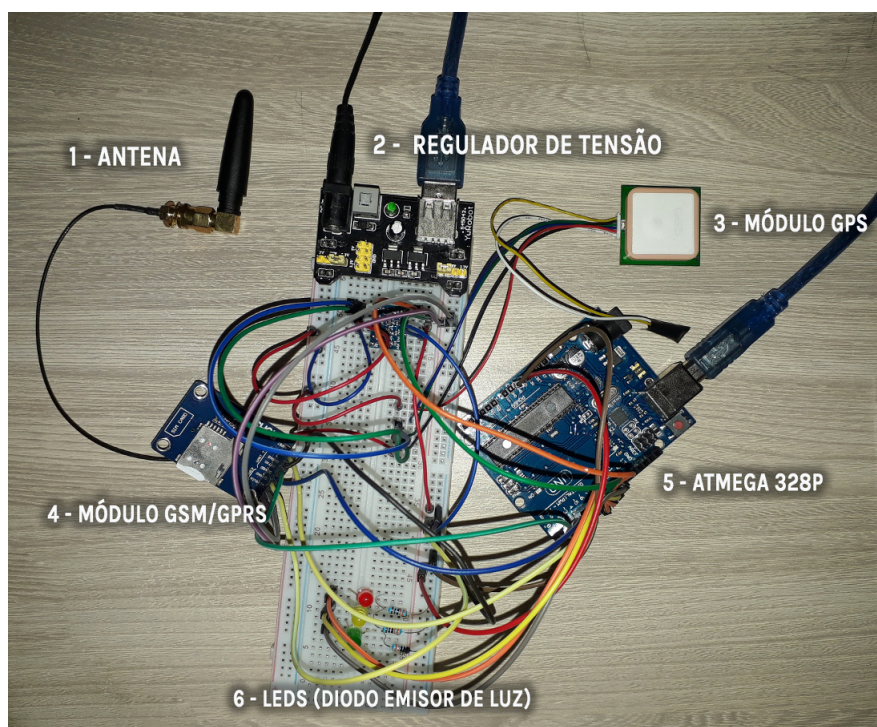


Figura 2. Arquitetura do Sistema Embarcado

5. Planejamento e Execução do Experimento

Nesta seção o planejamento do experimento é descrito, sua execução é detalhada e os resultados apresentados e discutidos.

5.1. Planejamento do Experimento

Os experimentos têm como objetivo verificar a precisão das coordenadas do módulo GPS e o comportamento do sistema em áreas de sinal telefônico fraco.

As coletas de dados foram realizadas em dois trajetos: (i) percurso feito pelo ônibus no trajeto Praça José de Barros - UFC/IFCE; e (ii) percurso feito pelo ônibus no

trajeto UFC/IFCE - Praça José de Barros. Apesar de serem pontos de início e fim invertidos para as duas rotas, os trajetos são diferentes. O sistema embarcado foi transportado durante os trajetos.

Os dados coletados pelo sistema foram as coordenadas em latitude e longitude, horário de emissão e a velocidade do veículo. Todos esses dados são capturados pelo módulo GPS descrito na seção anterior.

Com esses dados, alguns algoritmos da literatura irão prever o horário de chegada do ônibus na parada. Os algoritmos aplicados foram: *Decision Tree*, *XGBoost*, *Linear Regression* e *Random Forest*. Foram utilizadas as métricas MAE (*Mean Absolute Error*) e RMSE (*Root Mean Square Error*) para análise dos resultados.

Para visualização e análise dos dados, mapas com os trajetos, mapas com os pontos de coleta, gráficos de barras e tabelas foram desenvolvidos para suportar o experimento.

5.2. Execução do Experimento

A coleta de dados foi realizada no período de 11/03/2019 até 15/03/2019. No total, 12 coletas foram realizadas, 6 no percurso Praça José de Barros - UFC/IFCE (Figura 3) e 6 na rota UFC/IFCE - Praça José de Barros (Figura 4). Os mesmos ônibus usados no transporte universitário foram utilizados no experimento. O sistema foi ligado antes de entrar no ônibus e desligado logo após a saída. Assim, tentou-se manter os dados o mais próximo possível da realidade.

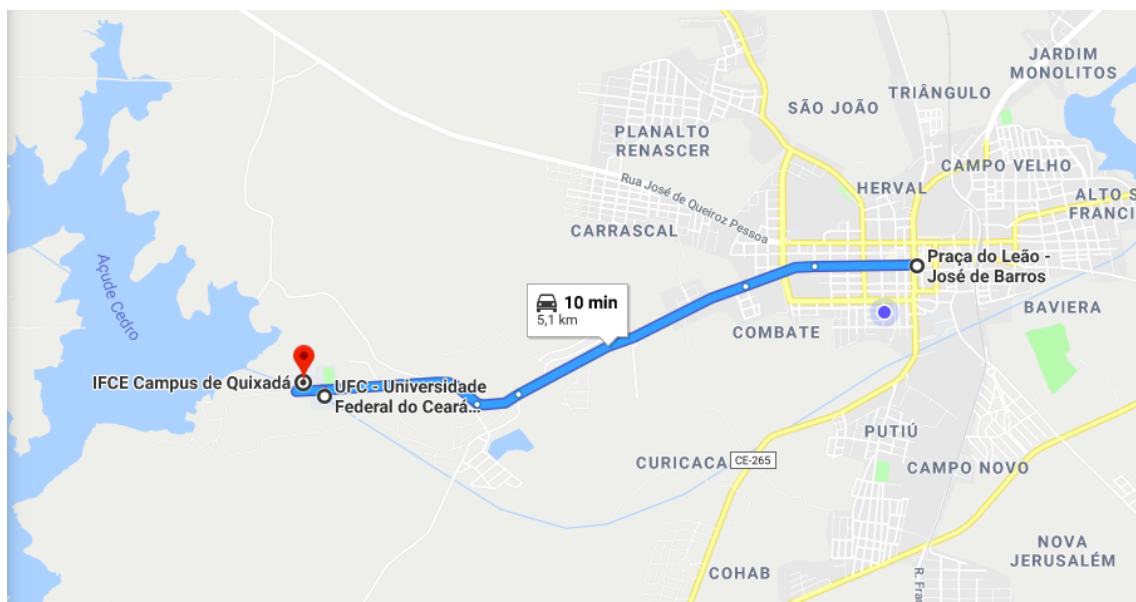


Figura 3. Rota no sentido Praça - UFC/IFCE. Fonte: Google Maps

Durante o experimento a coleta foi parcialmente afetada por falha de conexão do módulo GSM/GPRS à rede. Mesmo com essa condição adversa, a coleta de dados foi proveitosa. Após a coleta de dados, foi montada a base de dados afim de construir o modelo de aprendizado de máquina. Para isso, os dados foram separados em 70% para treino e 30% para teste. Como dito previamente, os algoritmos de aprendizado supervisionado testados foram *Linear Regression*, *Decision Tree*, *Random Forest* e o *XGBoost*.

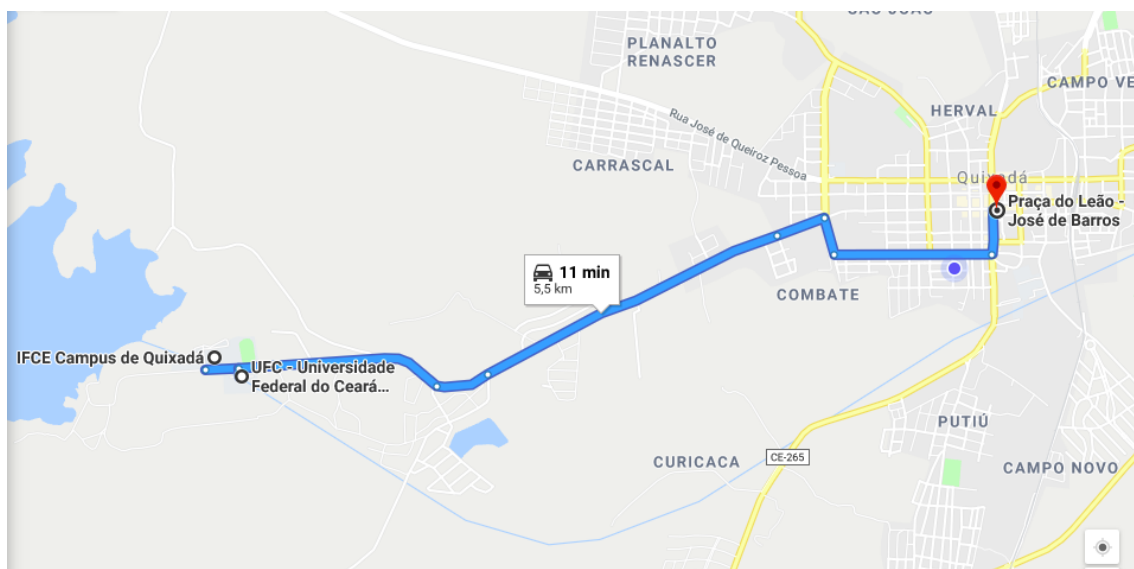


Figura 4. Rota no sentido UFC/IFCE - Praça. Fonte: Google Maps

Para avaliar o desempenho dos algoritmos foram utilizadas duas métricas: MAE (*Mean Absolute Error*), descrita na Equação 1 e RMSE (*Root Mean Square Error*), descrita na Equação 2. Essas técnicas são amplamente utilizadas para testar a eficiência de algoritmos de regressão.

$$MAE = \frac{1}{n} \sum_{j=1}^n |y - \hat{y}| \quad (1)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{j=1}^n (y - \hat{y})^2} \quad (2)$$

As Equações 1 e 2 descrevem, respectivamente, as métricas MAE e RMSE, onde y é o valor real e \hat{y} é o valor predito pelo algoritmo de aprendizado de máquina.

O módulo GPS consegue captar informações como latitude, longitude, horário e velocidade. Foram fixados os pontos de latitude e longitude das paradas do ônibus e calculados o tempo que demorou para o ônibus chegar a uma determinada parada. Foi inserida a distância euclidiana entre o ponto de partida e o ponto de chegada (ponto que representa uma parada do ônibus). Por fim, foram retiradas as coordenadas de latitude e longitude (tanto de partida como destino), pois as mesmas não se mostraram serem atributos relevantes quando os seus valores brutos são inseridos para treinar o algoritmo. Por fim, obteve-se três atributos que são eles: a velocidade do ônibus (em km/h), o horário (em segundos) e a distância euclidiana entre os pontos de partida e chegada. O tempo (em segundos) que demorou para o ônibus chegar na parada é o atributo requerido na previsão.

A Figura 5 apresenta uma amostragem da coleta desses atributos no percurso UFC/IFCE - Praça José de Barros. Nos dados coletados, obtivemos a latitude e longitude do ônibus no momento da coleta, a hora que a localização foi capturada e a velocidade do veículo em Km/h. Observa-se que os dados são gerados, aproximadamente, a cada

10 segundos, o que proporciona uma quantidade de dados satisfatório para a aplicação. Essas informações foram utilizadas no algoritmo de predição.

1	Latitude	Longitude	Hora	Velocidade (Km/h)
2	-4.974.530	-39.034.340	00:40:08	56.06
3	-4.973.767	-39.032.822	00:40:20	57.95
4	-4.973.132	-39.031.536	00:40:30	52.74
5	-4.972.474	-39.030.174	00:40:45	37.21
6	-4.972.107	-39.029.263	00:40:54	46.58
7	-4.971.717	-39.028.191	00:41:03	49.86
8	-4.971.455	-39.027.451	00:41:12	20.72
9	-4.971.333	-39.027.145	00:41:21	16.83
10	-4.971.112	-39.026.531	00:41:30	37.21
11	-4.970.862	-39.025.795	00:41:39	23.39
12	-4.970.808	-39.025.658	00:41:48	0.65
13	-4.970.782	-39.025.635	00:41:57	1.93
14	-4.970.839	-39.025.639	00:42:07	0.07
15	-4.970.850	-39.025.639	00:42:16	1.7
16	-4.970.845	-39.025.642	00:42:25	0.37

Figura 5. Amostragem de coleta no percurso UFC/IFCE - Praça José de Barros

5.3. Resultados e Análises

Nas Figuras 6 e 7 pode-se visualizar, respectivamente, os dados da rota Praça José de Barros - UFC/IFCE e UFC/IFCE - Praça José de Barros. Percebe-se que não foi possível obter uma total cobertura da rota. Isso é ocasionado por dois fatores. O primeiro é o fraco sinal telefônico ao afastar-se do centro da cidade, pois as duas instituições encontram-se distantes do centro da cidade. O segundo motivo é que foi utilizada uma antena para GSM/GPRS com baixo alcance. Apesar dessa limitação inicial do projeto, isso não interferiu no estudo do problema e da viabilidade da solução.



Figura 6. Coleta no sentido Praça (15/03/2019) - UFC/IFCE. Fonte: flopp.net

O módulo GPS utilizado se mostrou bastante preciso, de forma que nenhum ponto foi gerado fora da rota. Este fato contribuiu bastante na coleta e, consequentemente, na predição. Foram realizados diversos testes de predição, com diversos algoritmos e testando empiricamente parâmetros para os mesmos. Também foram testadas

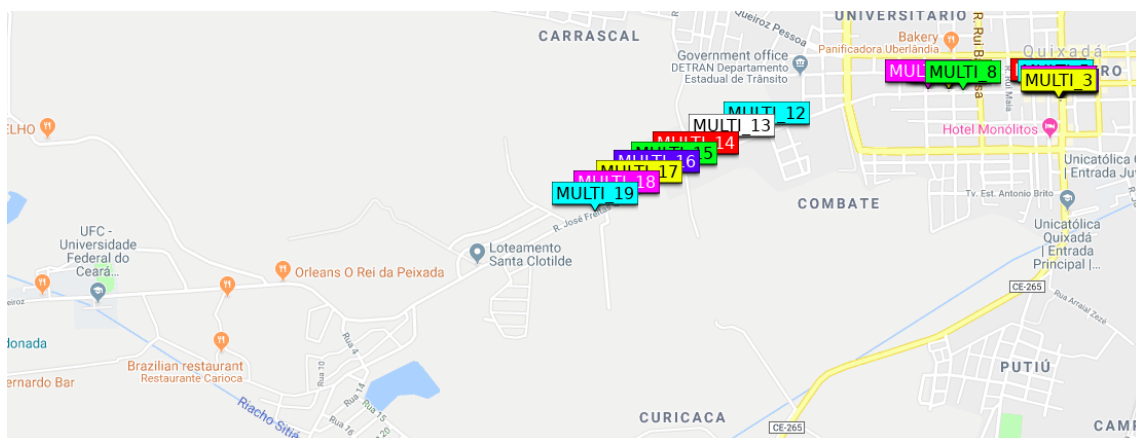


Figura 7. Coleta no sentido UFC/IFCE (15/03/2019) - Praça. Fonte: flopp.net

diversas transformações dos dados, dentre elas a normalização e a padronização. Essas transformações citadas não apresentaram melhores resultados em nossos testes.

Após diversos testes de predição, o gráfico da Figura 8 apresenta uma comparação entre os algoritmos testados. Os testes realizados consistiram em dado o mesmo conjunto de dados de treino para todos os algoritmos, geramos um modelo de aprendizado para cada um dos algoritmos. Após isso, inserimos um mesmo conjunto de dados de teste para todos os algoritmos afim de obter os dados preditos. Por fim, avaliamos os resultados preditos pelos algoritmos utilizando as métricas MAE (*Mean Absolute Error*) e RMSE (*Root Mean Square Error*). A Figura 8 apresenta um resumo dos resultados, onde é exibido os valores das métricas que cada algoritmo obteve nos testes.

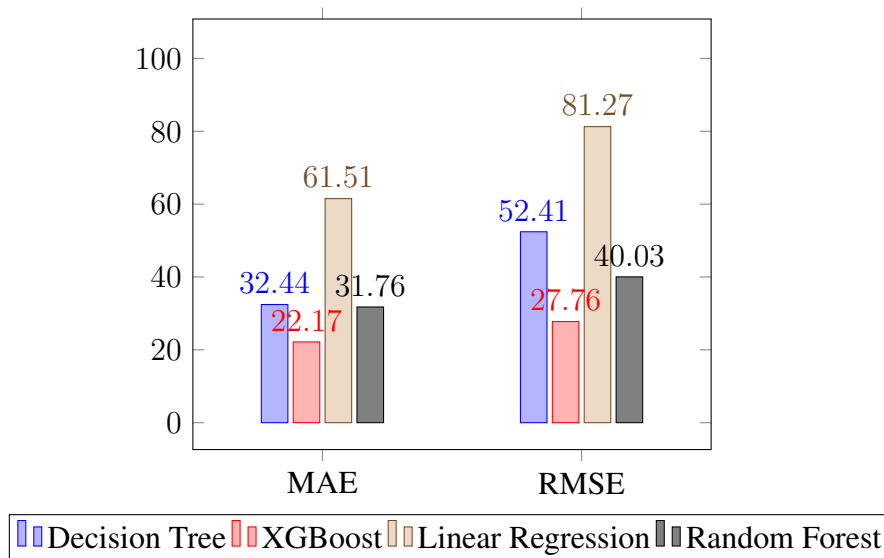


Figura 8. Comparação entre os algoritmos testados utilizando as métricas MAE (*Mean Absolute Error*) e RMSE (*Root Mean Square Error*)

O MAE e o RMSE medem o erro que aconteceu nas predições, dessa forma, o melhor algoritmo em um determinado teste será o que apresentar os menores valores de MAE e RMSE. Pode-se observar diante do gráfico que o algoritmo *XGBoost* foi o que apresentou melhores resultados comparado com os demais algoritmos. Observa-se, tomando

como base a métrica MAE, que em média o *XGBoost* tem erro de aproximadamente 22 segundos, o que é considerado uma precisão muito boa para a aplicação proposta. Pode-se notar que o algoritmo *Linear Regression* foi o que apresentou os piores resultados. Acredita-se que isso se deu pelo fato do tempo não se comportar como uma função linear devido as diversas paradas e o tráfego da via influenciar no tempo.

Os algoritmos *Decision Tree* e o *Random Forest* apesar de apresentar bons resultados não serão utilizados, pois em nenhuma das duas métricas esses algoritmos foram melhor que o *XGBoost*. Diante dos resultados apresentados, foi criado o modelo final utilizando o algoritmo *XGBoost*. Nessa etapa, todos os dados capturados serão utilizados como dados de treino para o algoritmo.

Por fim, foram capturadas informações sobre as *feature importances*, relacionando a importância de cada atributo para a construção do modelo de aprendizado. A *feature importances* apresenta um percentual de importância de um atributo no modelo gerado, ou seja, pode-se observar qual atributo foi mais útil para fazer uma predição mais precisa. Dessa forma, quanto maior o valor para um determinado atributo maior a sua importância para o modelo. A Tabela 1 apresenta os atributos dos dados finais, onde “Dist_euclidiana” é a distância euclidiana entre o ponto de partida e o ponto de chegada, “Tempo” representa o horário que o dado foi capturado e “Velocidade” é a velocidade que o ônibus estava desenvolvendo naquele determinado ponto.

Dist_euclidiana	Tempo	Velocidade
0.28264	0.51920	0.19815

Tabela 1. Importância de cada atributo para o modelo de aprendizado.

Nota-se na Tabela 1 que o tempo é o atributo mais relevante, pois é o atributo que apresenta o maior valor de *feature importances*. Acredita-se que isto está ligado ao fato do tráfego na rota depender diretamente do horário. A velocidade do veículo captada pelo GPS é uma velocidade instantânea. Dessa forma, esse valor pode não ter muita relação com tempo de viagem, já que o veículo pode variar bastante a sua velocidade no intervalo de 10 segundos (que é o tempo para os dados serem capturados). Por esse motivo, acredita-se que a velocidade foi considerada o atributo menos relevante para o modelo de aprendizado final.

6. Conclusão e Trabalhos Futuros

Este trabalho apresentou uma solução baseada em um sistema embarcado e comunicação remota para auxiliar usuários do transporte público ou fretado, especificamente para o transporte de alunos de campi universitários. Adicionalmente uma estratégia para a previsão do tempo de chegada dos ônibus nas paradas foi desenvolvida, baseada em aprendizado de máquina. De maneira geral, a solução atendeu ao objetivo proposto.

Durante o desenvolvimento do trabalho foi analisada a viabilidade do sistema proposto, tanto em custo como em operabilidade. Os testes evidenciaram que em média o algoritmo consegue prever com um erro de apenas 22 segundos, valor baixo comparado ao custo de tempo total de uma viagem (em torno de 30 a 40 minutos).

Como trabalhos futuros pretende-se: implementar o sistema embarcado em um circuito impresso, finalizar o desenvolvimento da aplicação do usuário, adquirir uma

antena com maior alcance e uma placa solar para alimentar o sistema de uma forma ecológica. A partir dessa infraestrutura, pretende-se utilizar as informações sobre o transporte dos universitários em uma aplicação para o usuário, já prevista na arquitetura, concentradas em um *feed* de notícia.

Em relação ao experimento em si, também pretende-se realizar outros trajetos e coletar dados em ambientes totalmente em cidades, com volumes de tráfego e horários variados. Por fim, o desenvolvimento de uma aplicação *mobile* possibilitará ao usuário final obter a previsão de chegada na parada e a localização atual dos ônibus.

O modelo de aprendizado de máquina pode ser aprimorado obtendo outros atributos, tais como, a distância real e o tempo estimado sem paradas que pode ser fornecido pela API (*Application Programming Interface*) do *Google Maps*. A obtenção de mais dados também pode ser encarado como um fator de melhoramento do modelo.

Outra mudança que podemos propor em nosso modelo de aprendizado é a de divisão da rota em diversos trechos de mesmo tamanho e tentar prever o tempo de chegada a partir desses pequenos trechos. Essa abordagem pode ser custosa e não temos garantias de melhorias, mas é apresentada em alguns trabalhos da literatura.

Agradecimentos

Este trabalho foi parcialmente suportado com recursos do projeto **CrOSSiNg: Avaliação da Qualidade de Nuvens Computacionais Apoiadas por Redes Definidas por Software e Virtualização de Funções de Rede**, Edital Universal MCTI/CNPq 01/2016 (processo 422342/2016-5). Agradecemos também ao Programa de Educação Tutorial (PET - Tecnologia da Informação - UFC - Campus Quixadá) pelo apoio no trabalho.

Referências

- Castro, J., Araujo, I., Anjos, E., and Matos, F. (2017). A survey on bus monitoring systems. In *International Conference on Computational Science and Its Applications*, pages 220–231. Springer.
- Cruz, P., Couto, R. S., and Costa, L. H. M. K. (2018). Análise da cobertura espacial de uma rede de sensores baseada em Ônibus urbanos. In *XXXVI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC2018)*.
- de M. Neves, A. R., Sarmanho, K. U., Jr., F. C. N., and Meiguins, B. S. (2017). Iniciativa smart campus: um estudo de caso em progresso na universidade federal do pará. In *Workshop de Computação Urbana (CoUrb2017)*.
- Feng, X., Zhang, J., Chen, J., Wang, G., Zhang, L., and Li, R. (2018). Design of intelligent bus positioning based on internet of things for smart campus. *IEEE Access*, 6:60005–60015.
- Martins, N. A. (2005). Sistemas microcontrolados. *Uma abordagem com o Microcontrolador PIC 16F84*. Editora Novatec Ltda, 1ª edição.
- Miranda, W. M., de Mendonça, R. T. R., da Silva, A. A., de Lima Curvello, A. M., de Souza, F. L. d. S., and da Silva, H. J. (2017). Busme: Automatic bus localization system and route registration. *Procedia Computer Science*, 109:1098–1103.

- Monard, M. C. and Baranauskas, J. A. (2003). Conceitos sobre aprendizado de máquina. *Sistemas inteligentes-Fundamentos e aplicações*, 1(1):1.
- Neves, D., Dias, F. C. A., and Cordeiro, D. (2018). Uso de aprendizado supervisionado para análise de confiabilidade de dados de crowdsourcing sobre posicionamento de ônibus. In *Workshop Brasileiro de Cidades Inteligentes (WBCI2018) - XXXVIII Congresso da Sociedade Brasileira de Computação (CSBC 2018)*.
- Rodrigues, D., Severino, J., Costa, F. T., Nakamura, L. H., and Meneguette, R. I. (2018). Uma nova infraestrutura para captação e comunicação dos sensores embarcados no veículo. In *II Workshop de Computação Urbana (CoUrb2018)*.
- Saad, S. A., Ishak, M. H. I., Fauzi, M. H. M., Baharudin, M. A., Idris, N. H., et al. (2018). Real-time on-campus public transportation monitoring system. In *2018 IEEE 14th International Colloquium on Signal Processing & Its Applications (CSPA)*, pages 215–220. IEEE.
- Sungur, C., Babaoglu, I., and Sungur, A. (2015). Smart bus station-passenger information system. In *2015 2nd International Conference on Information Science and Control Engineering*, pages 921–925. IEEE.
- Vianna, A., Cruz, M., Barbosa, L., and Gama, K. (2018). Análise do impacto de chuvas na velocidade média do transporte público coletivo de Ônibus em Recife. In *Workshop Brasileiro de Cidades Inteligentes (WBCI2018) - XXXVIII Congresso da Sociedade Brasileira de Computação (CSBC 2018)*.
- Wen, M., Rosa, T. O., Souza, M. C., Aleixo, R. P., Silva, C. A., Sá, L. S., Santana, E. F. Z., and Kon, F. (2018). Criação de modelo para simulação de movimentação de Ônibus a partir de dados reais. In *Workshop Brasileiro de Cidades Inteligentes (WBCI2018) - XXXVIII Congresso da Sociedade Brasileira de Computação (CSBC 2018)*.
- Yamaguchi, T., As, M., and Mine, T. (2018). Prediction of bus delay over intervals on various kinds of routes using bus probe data. In *2018 IEEE/ACM 5th International Conference on Big Data Computing Applications and Technologies (BDCAT)*, pages 97–106. IEEE.

Development of a Semantic Representation Model of Criminal Information to Support the Assessment of Risk Situations

Jordan F. Saran^{1,2}, Leonardo C. Botega^{1,2}

¹Programa de Pós-graduação em Ciência da Informação - PPGCI
Universidade Estadual Paulista Júlio de Mesquita Filho - UNESP – Marília, SP – Brasil

²Grupo de Interação Humano-Computador - GIHC
Centro Universitário Eurípides de Marília - UNIVEM – Marília, SP – Brasil

{jordansaran,botega}@univem.edu.br

Abstract. *Situational Awareness (SAW) refers to the level of consciousness that an individual or team holds over a situation. In the area of risk management and criminal data analysis, SAW failures can induce human operators to make mistakes in decision making and pose risks to life or property. In this context, risk assessment processes, which commonly involve data mining, fusion and other methods, present opportunities to generate better information and contribute to the improvement of the SAW of crime and risk analysts. However, the characterization of complex scenarios is subject to problems of representation and expressiveness of the information, which may influence its interpretation due to their quality and significance, generating uncertainties. The state-of-the-art in representation of information on risk situations and related areas presents approaches with limited use of information quality. In addition, the solutions are restricted to syntactic mechanisms for the determination of relations between information, negatively restricting the assertiveness of the results. Thus, this paper aims to develop a new approach to semantic representation of information of risk situations, more specifically creating domain ontologies, instantiated with crime data and information quality. In a case study, real information on crimes, represented by the new semantic model and consumed by computational inference processes, was processed, aiming to characterize robbery and theft situations.*

1. Introduction

In risk management domain, critical system operators need to act decisively with limited time, with overload of information that is often incomplete, contains a high level of psychological pressure among other factors that act in a negative way in the perception and understanding of the situation. Such problems compromise the outcome in responding to a critical situation because decision making can be inaccurate. In this context, situational awareness (SAW) helps human operators of critical systems to measure and improve their levels of perception, understanding and projection of the situation. SAW is an important cognitive process for decision makers in several critical areas and concerns the perception of the presence and disposition of entities of interest in an environment, the understanding of the meaning and importance of their individual actions and In space-time and the projection of their status in the near future [Endsley 1988].

In risk assessment systems, the acquisition and improvement of SAW levels of human operators becomes a complex task, since such systems collect, process and represent data from primarily human intelligence (HUMINT). Such reports can produce data with low levels of quality because its sources are heterogeneous, unpredictable, complex and dynamic. When incomplete data is propagated in decision-making support systems, they produce inaccurate information, which can negatively influence the SAW of the human operator, which depends on the correct specifications to understand the situation and the taking of Assertive decision.

In this context, it was identified that, together with the need to adopt a methodology for assessing the quality of data, the use of ontologies as semantic models can help to improve the representation of information useful to the system and the end user.

To highlight this issue, works such as [Mescherin et al. 2013] use the ontology in conjunction with crises metamodels to increase interoperability and decrease the lack of standardization of vocabularies used in communication between risk systems or the work of [Carvalho et al. 2009] that uses probabilistic ontology (PR-OWL) to represent critical areas characterized by the presence of uncertainty. Both works use ontology to solve problems related to their specific domains, however, do not include the data quality factor as a way of qualification of information in their ontologies, to enrich the representation and Inference of information.

The work [Souza et al. 2018] proposes the construction of an accurate knowledge about traffic conditions and the levels of insecurity in urban roads, calculating faster and safer routes, through the use of trafficking conditions, associated with Mapping and analysis of criminal events. Observing the work [Souza et al. 2018] it would be possible to generate inputs on the traffic conditions from the semantic model, thus enabling better conditions in the calculation of routes.

Thus, this work aims to contribute to the development of the SAW of the users of the system of assessing risk situations using ontologies to represent the information in a clearer and more objective way, demonstrating to the human operator all Possible relationships, entities and their attributes with a risky situation, where criminal data will be used. He will also be using a quality management methodology to quantify and qualify the data used by him in order to help the inferences processes on the risk situation.

2. Development of the Semantic Model

This project follows the development methods and definitions of the JDL pattern of fusion and data representation (Joint Directors Laboratories) [Blasch et al. 2013], and the guidelines of the Quantifymodel of fusion and information representation (Quality-awareHuman-driven information Fusion Model) [Botega et al. 2017] (Figure 1), developed by the human-Computer Interaction Group (HCIG).

This model evidences the need to complement syntactic processes of evaluation of situations with the representation and semantic processing, fueled by the evaluation of the quality of data and information. To develop this work, the methodology applied are based on the following activities: (1) Requirements Survey: Research on the main topics exposed in this work as risk management, situational awareness, management of risk with criminal data, and development and application of a questionnaire to experts in the

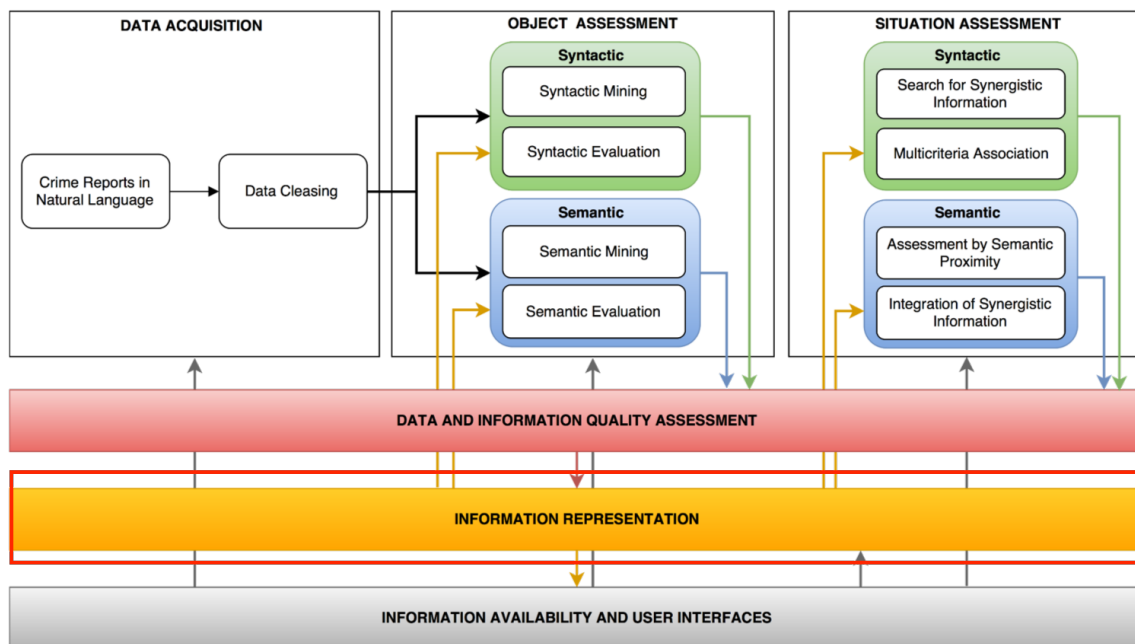


Figure 1. Process of Evaluation of Situations of Risk for the improvement of the Situational Awareness

field; (2) Task analysis aimed at the objective: application of the task analysis directed by the technical Meta(GDTA); (3) Development of an ontology of criminal domain: use of the 101 methodology and the vocabulary DQV (Data Quality Vocabulary) to develop the ontology; (4) Application of Quality Management: use of the IQESA (methodology for assessing the quality of information in the context of the consciousness of the emergency situation), to apply the metrics and dimensions of the quality of the data; (5) Validation of ontology: Validation of ontology developed using the SPARQL query.

2.1. Requirements Survey

After a study of the main themes of this study, a questionnaire was developed and applied to the military police of the state of São Paulo (PMESP). With the answers obtained, it was possible to identify which information is a priority for decision-making in a crime situation, and what are the services and data sources that should be consumed to provide them or deduce them. With this entry, it was possible to construct an attribute tree, this feature becomes an important one for the other phases of development because, it provides the entire hierarchy of attributes through the nodes that compose it, and for the qualification phase of the data that provides inputs to identify requirements that must be quantified and qualified.

From these results, it was possible to update the first version of a goal-driven task analysis Model (GDTA). This model represents beyond the objectives to be achieved, decisions to be taken, tasks to be fulfilled and information requirements to contemplate this demand [Endsley 2001]. The level of importance of each information revealed by the interview also enabled the creation of a scale of informational priorities.

2.2. Goal-Directed Task Analysis (GDTA)

The use of the GDTA methodology [Endsley 1988] aims to identify the main aspects related to the domain being applied. In critical systems, it is used to identify the main objectives to be successfully achieved by the operator, as well as to contribute to the acquisition of SAW and development of such operator. The GDTA is based on goals or goals, and not on simple tasks, according to it, this is because the objectives are the basis for decision-making in many environments with a high level of complexity, such as the military area [Endsley 2001]. Figure 2 shows part of the GDTA developed from the attribute tree.

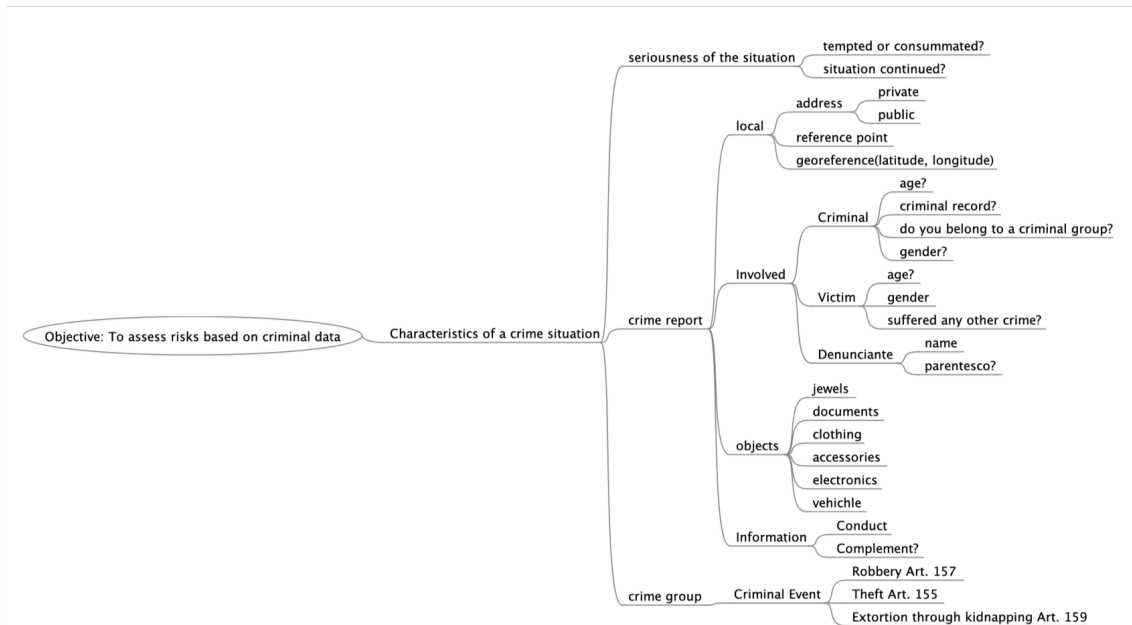


Figure 2. Goal-Directed Task Analysis (GDTA)

2.3. Development of the criminal domain ontology

The methodology adopted for the elaboration and development of ontology will be the 101 methodology [Noy and McGuinness 2001]. Following a development cycle divided into seven stages, this methodology became the most used in the construction of ontologies due to its flexibility and simplicity of being applied in any domain and easy to understand. The steps for the development of ontology in the present study were:

- **Define scope:** The domain of application and study of this ontology are situations of risk, in which for this work are situations of theft and theft exclusively, where their function will be assisting in the generation of richer and more meaningful information for the Development of SAW specialists.
- **Consider the reuse:** No ontology was found related to the same domain of this work that helped in the development of it, especially in the use of Portuguese language and the organization of criminal problems. However, it was identified an external ontology that can be applied in the construction, was the DQV used to apply quality indexes within an ontology (W3C).

- **Enumerate terms:** The execution of this step was through the analysis of the responses obtained from the questionnaire applied to operators of critical systems and public and private security in conjunction with an analysis of criminal records on official bases, thus obtaining a Survey of the main terms found in which they went through the validation of an expert.
- **Define Classes and properties:** Based on the questionnaire applied, it was possible to identify the most relevant information to characterize a situation of laughter in the criminal domain. Based on the result of this questioning, the main classes of ontology were defined: person, place, object, situation and EC (Criminal event). For each of the mentioned classes were created more specific subclasses as well as the properties that characterize them.
- **Define constraints:** After defining classes and properties, it was necessary to establish domain constraints, such as: "Situation at least there will be a location", which receives cardinality 1 – N within the ontology.
- **Create instances:** The ontology was fed with several instances that characterize the information pertaining to the criminal domain, which are presented by Figure 3 after the previous steps.

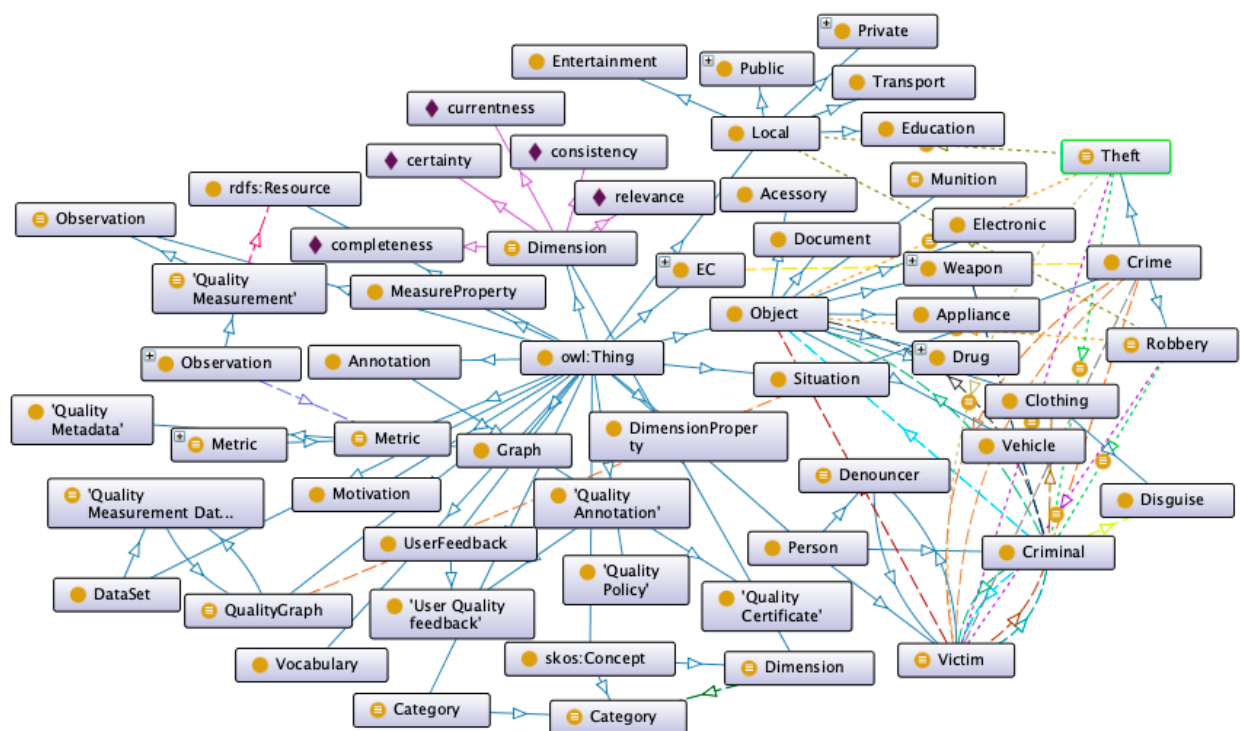


Figure 3. Main classes identified by the methodology 101

2.4. Quality Management Methodology Application

For quality management in conjunction with the ontological development process, functions and metrics of the IQESA methodology were used [Botega et al. 2016]. In addition, the DQV (Data Quality Vocabulary) vocabulary was used, with the objective of identifying the relations between the quality dimensions and the ontological instances. To qualify and quantify the data, the dimensions were used: currentness, completeness, consistency, relevance and certainty.

- **Currentness:** It is a temporal progress of an event, its contribution to the development of SAW systems operators of situations assessment is essential due to the need for updated data [Botega et al. 2016]. Where θ consists of an index composed of the current time and the metric is obtained by two attributes presented in the situation of occurrence, being represented by the Σ of these attributes: (1) Time that the crime occurred and (2) Time that the report was performed. This evaluation has two values as a result: one of them being a quantitative indicator for the existence of two attributes and how many minutes have passed since the report emerged. Equation 1 was adjusted to perform the current quantitative count [BOTEGA 2016].

$$\sum_{y=1}^2 -\theta \quad (1)$$

- **Completeness:** To evaluate the completeness of the criminal domain, a tree of attributes was used, because it has all the priority attributes presented to meet a case report. Equation 2 demonstrates how to perform weight calculation.

$$C = S\left[\left(\frac{\sum\beta \cdot \gamma}{\sum\gamma} \cdot 0.9\right) + 0.1\right] \quad (2)$$

Equation 2 calculates the completeness dimension for the attributes present in the object to be analyzed, and S represents the presence of the object to be evaluated, when present $S = 1$, and when absent $S = 0$; β represents the attribute that describes the object, when present equals 1, and when missing is 0; The γ represents the weight, which, when considered a priority, has a value equal to 2 and when no priority is received the equal value 1. For each present weight, the sum of the multiplication and its result divided by the total of priority attributes should be made. The resulting value of this formula should range from 0 to 100% [BOTEGA 2016].

- **Relevance:** There is no equation to calculate relevance, it consists in identifying the relevant attributes presented in the object, assigning 0 to the irrelevant attributes and 1 to the relevant ones.
- **Consistency:** In the context of this work, the consistency assessment will be applied from the second occurrence regarding the same situation as the system receives, and the value assigned for consistent occurrences will be from 0 to 100%. Below are examples of inconsistent occurrences.

- Report 1: "The victim stopped the vehicle at a traffic light, was surprised by two individuals who were on a motorcycle twister, black color. The croup hit with a gun in the glass and utting serious threat, subtracted the victim's vehicle. The authors escaped with an ignored destination. Victim has no conditions to describe the authors with details. Requested the lock next to Cepol. Nothing more. "
- Report 2: "Two guys on a dark-colored motorcycle stole a car at the lighthouse at the intersection of Mooca Street and Taquari, pointed a revolver and took the woman's car. One of them was in a red coat. "
- Report 3: "An individual stole a gray bike on Mooca Street at the intersection of Fernando Falcão Street. The perp was armed and left sense to the Parque da Sabesp. "

The three occurrences were obtained by PMESP, the first two are consistent because both refer to the characteristics of the criminals, one being reported to have seen two individuals on one motorcycle and the other that the motorcycle was black, the third occurrence is inconsistent due to the location where the crime occurred and the stolen goal being different from the other two occurrences.

- **Certainty:** At the end of the qualification and quantification of all dimensions and attributes, the assessment of the certainty of the situation is made. This dimension consists in determining a global index to represent the assessment of the quality of information as a whole.

2.5. Ontology Validation

To validate the present ontology developed throughout this work will be presented a case study in this section as an objective to demonstrate the quality of data and information, together with the ontology, as a support for critical systems and, in consequence, to help the operators of these systems to improve their SAW, promoting high level abstractions and reducing uncertainties at the time of decision making. To demonstrate such results, the IQESA methodology will be used along with the ontology the DQV vocabulary that represents the quality metrics and dimensions.

A situation of robbery will be addressed in this context, it will be necessary to understand the scenario of the crime situation and to identify some information relevant to the resolution of the case, such as where the crime was held, how many criminals had during the fact, What types of clothing the criminals were using during the crime and what kind of object was stolen. The following is the first occurrence used for this study:

Report 1: "The victim stopped the vehicle at a traffic light, was surprised by two individuals who were on a motorcycle twister, black color. The croup hit with a gun in the glass and uttering serious threat, subtracted the victim's vehicle. The authors escaped with an ignored destination. Victim has no conditions to describe the authors with details. Requested the lock next to Cepol. Nothing more. "

With this information obtained from the report it is possible to extract the information such as the time of the crime, the location, stolen object, characteristics of the victim among other information. After this first contact with the occurrence, the system itself in a data merger process organizes the information based on the process of assessing risk situations illustrated in Figure 1.

Table 1. QUALITY ASSESSMENT 1º REPORT

Entities	Completeness	Relevance	Currentness	Consistency	Certainty
Local	0%	0	-	-	-
Occurrence	90%	3	-	-	-
Victim	91,82%	4	-	-	-
Criminal	30%	1	-	-	-
Crime	100%	1	-	-	-
Object	100%	1	-	-	-
EC	100%	1	-	-	-
Situation	85%	4	375 min	-	28,42%

In a next step, the IQESA methodology is used to qualify and quantify the data and information along with the Completeness, Relevance, Currentness and Consistency dimensions presented in Table 1.

The Consistency dimension is calculated only from the second occurrence on the same crime situation. The calculation of certainty is given by the sum of all the calculated indexes (592%), divided by the amount of incidences in the report(29) resulting in 28.42%. The Completeness, Relevance and Currentness indices are calculated according to the formulas shown in the previous sections:

- **Occurrence:** Of the 6 attributes present in the class, only 5 have values, applying these values the formula of completeness present in Equation 1, the index found is 90%. Of the 5 attributes present, only 3 are relevant.
- **Crime:** It has only one attribute as value, so the completeness index is 100%. It has a value of 1 relevance.
- **Object:** It has only one attribute as value, so the completeness index is 100%. It has a value of 1 relevance.
- **EC(Criminal Event):** It has only one attribute as value, so the completeness index is 100%. It has a value of 1 relevance.
- **Victim:** Of the 7 attributes present in the class, only 6 has value generating a completeness index of 91.82%. Of the 6 attributes present, only 4 are relevant.
- **Criminal:** Of the 6 attributes present in the class, only 1 has value generating a Completeness index of 30%. of the 1 attribute present, only itself is relevant.
- **Situation:** Of the 7 attributes present in the class, only 6 has value generating a Completeness index of 82%. Of the 6 attributes present, only 4 are relevant.

To calculate the currentness, the time the crime described by the victim, in this case at 11:30, was subtracted from the time the victim performed the report at 18:15, thus generating a current value of 375 minutes. The second report of occurrence was made by an anonymous complaint, being done after a few minutes of the crime.

Report 2: "Two guys on a dark-colored motorcycle stole a car at the lighthouse at the intersection of Mooça Street and Taquari, pointed a revolver and took the woman's car. One of them was in a red coat."

It is possible to verify that this report contains more information about the situation in general, e.g. information about the location where the crime occurred and whether the criminals had some kind of firearm. Each new information, referring to the same situation as the system receives, occurs the same process of evaluating the quality of the data. Table 2 presents the new results regarding the dimensions of certainty, completeness, consistency, timeliness and relevance for the situation.

In Table 2 it is possible to obtain consistency, this measure is used to process the information for a situation that is already in progress, to apply a metric that is needed a new information base for a parameter that arrives at the system. The calculation resulted in 100% because the two occurrences contained information about the same crime, some even repeated confirming its consistency. The third report of occurrence was made by an anonymous complaint, being done after a few minutes of the crime equal the second report of occurrence.

Table 2. QUALITY ASSESSMENT 2º REPORT

Entities	Completeness	Relevance	Currentness	Consistency	Certainty
Local	26%	6	-	-	-
Occurrence	82%	3	-	-	-
Victim	82%	4	-	-	-
Criminal	46%	4	-	-	-
Crime	100%	1	-	-	-
Object	100%	1	-	-	-
EC	100%	1	-	-	-
Situation	91%	4	5 min	100%	36,13%

Report 3: "Two men robbed a gray car in the street from Mooca to the side of Santander bank . The men were on a black motorcycle one of them was in a blue jacket and jeans. They left in the direction of the villa lobos hospital."

This last occurrence presents some more important information, in which better characterizes the current situation of the crime as emphasize that the place where the crime could have occurred was in "Mooca street near Santander bank" and that the amount of Criminals who have performed such an act, in fact, are "two misers".

Table 3. QUALITY ASSESSMENT 3º REPORT

Entities	Completeness	Relevance	Currentness	Consistency	Certainty
Local	81%	6	-	-	-
Occurrence	82%	3	-	-	-
Victim	82%	4	-	-	-
Criminal	46%	4	-	-	-
Crime	100%	1	-	-	-
Object	100%	1	-	-	-
EC	100%	1	-	-	-
Situation	100%	4	8 min	100%	44,55%

In Table 3 It is possible to see again the calculation of consistency, where it obtained the result of 100% because the three occurrences contained close information that characterizes a single situation. The Table 4 illustrates the quality values measured for the fusion of occurrences 1, 2 and 3, the process for these calculations were the same as previously shown.

After these calculations on the quality, the ontology was instantiated and demonstrated in Figure 4. Since the ontology has already been conceptually considered a mechanism of representation of information, by being instantiated with the data present in the three occurrences the ontology may present new relationships between the entities, this type of process contributes positively in Inference of knowledge to the human operator of risk assessment systems.

To validate the ontology presented at Figure 4 was utilized SPARQL queries, these queries permit the user of the risk assessment systems a better inference about the crime situation, generating more certainty for an assertive decision-making. Consider the pre-

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX dqv: <http://www.w3.org/ns/dqv#>
PREFIX crimevoc: <http://localhost:7200/crimevoc#>

The first query performed shows information about the Local class, where they are referring to the address, latitude, and longitude that are extremely important in the criminal domain, because considering this information the resources are allocated. The Table 5 demonstrates the results obtained by the query shown at Source Code 2.

Source Code 2. First query

```
SELECT *
WHERE {
    ?situation crimevoc:happens ?crime .
    ?crime crimevoc:occurs_in ?local .
    ?local crimevoc:type_street ?type_street .
    ?local crimevoc:street ?street .
    ?local crimevoc:number_street ?number_street .
    ?local crimevoc:complement ?complement .
    ?local crimevoc:point_reference ?point_reference .
    ?local crimevoc:neighborhood ?neighborhood .
    ?local crimevoc:city ?city .
    ?local crimevoc:state ?state .
}
```

Table 5. RESULT OF THE FIRST QUERY.

Attribute	Value
Situation	crimevoc:robbery
Crime	crimevoc:situation
Local	crimevoc:local
type_street	"Street"
street	"Mooca"
number_street	""
complement	""
point_reference	"Santander bank"
neighborhood	"Mooca neighborhood"
city	"São Paulo"
state	SP

The second query was performed in the Criminal class, the information shown below refers to the types of objectives used by criminals during the crime with the values referring to the dimensions of Completeness and Relevance, provided by data qualification. The Table 6 demonstrates the results obtained by the query shown at Source Code 3.

Source Code 3. Second query

```

SELECT ?Situation ?Crime ?Victim ?robbery_object ?type
WHERE {
  ?Situation crimevoc:have ?Crime .
  ?Crime crimevoc:has ?Victim .
  ?Victim crimevoc:robbery_object ?robbery_object .
  ?robbery_object rdf:type ?type .
  OPTIONAL {
    ?situation crimevoc:has ?graph .
    ?graph dqv:hasQualityMeasurement ?qt .
    ?qt dqv:isMeasurement ?Metric .
    FILTER regex(str(?Metric), 'Metric(SituatuionCertainty)'
    , 'i') .
    ?qt dqv:value ?value .
  }
}

```

Table 6. RESULT OF THE SECOND QUERY.

Attribute	Value
Object	Car
Completeness	92%
Relevance	4

The third query was performed in the Situation class, the information shown below refers to the percentage of certainty that all information contained in the ontology together with the data quality obtained during the construction of a single situation. The Table 7 demonstrates the results obtained by the query shown at Source Code 4.

Source Code 4. third query

```

SELECT *
WHERE {
  ?situation crimevoc:has ?graph .
  ?graph dqv:hasQualityMeasurement ?qt .
  ?qt dqv:isMeasurement ?Metric .
  FILTER regex(str(?Metric), 'Metric(SituatuionCertainty)'
  , 'i') .
  ?qt dqv:value ?value .
}

```

Table 7. RESULT OF THE THIRD QUERY.

Attribute	Value
Certainty	70,28%

3. Conclusion

The present work had the goal of demonstrating the development of a criminal domain ontology aware of data quality that will provide subsidies to achieve the different levels of

situation awareness by the operator in risk management systems. However, the ontology developed incidentally, can only be used to solve crimes of robbery, theft and extortion by kidnapping.

The use of ontologies together with risk assessment systems aims to provide the system with an inference model in which computational processes can hold a greater knowledge about the situation and the relations, this happens when the data used have quality. To the human operator of risk management systems the benefit of the ontology with quality data next to the system extends both the construction of smarter interfaces with more accurate and better represented information as well as the transmission of greater knowledge of the situation even with the dynamicity that it possesses, thus promoting a higher level of trust to the human operator in decision making.

In order to illustrate the results obtained in this work, a case study was developed and presented, in which it can be verified that through queries that meet the operator's needs and the situation he is attending at any given moment, the ontology together with the quality management can contribute positively to the human operator, thus making it possible to identify explicit and non-explicit relationships between the entities present in the situation, such as identifying which object in the situation belongs to the criminal and the victim.. Such knowledge and control of the situation contributes positively to the acquisition and maintenance of SAW levels of such operators, thus allowing safer and more assertive decision making.

References

- Blasch, E., Steinberg, A., Das, S., Llinas, J., Chong, C., Kessler, O., Waltz, E., and White, F. (2013). Revisiting the JDL model for information exploitation. *Proceedings of the 16th International Conference on Information Fusion, FUSION 2013*, pages 129–136.
- BOTEGA, L. C. (2016). *MODELO DE FUSAO DIRIGIDO POR HUMANOS E CIENTE DE QUALIDADE DE INFORMACAO*. PhD thesis, Universidade Federal de São Carlos(UFSCar), São Carlos.
- Botega, L. C., de Souza, J. O., Jorge, F. R., Coneglian, C. S., de Campos, M. R., de Almeida Neris, V. P., and de Araújo, R. B. (2016). Methodology for Data and Information Quality Assessment in the Context of Emergency Situational Awareness. *Universal Access in the Information Society*, 16(4):889–902.
- Botega, L. C., Valdir, A. P., Oliveira, A. C., Saran, J. F., Villas, L. A., and De Araujo, R. B. (2017). Quality-aware human-driven information fusion model. *20th International Conference on Information Fusion, Fusion 2017 - Proceedings*.
- Carvalho, R. N., Laskey, K. B., Costa, P. C. G., Ladeira, M., Santos, L. L., and Matsumoto, S. (2009). Probabilistic ontology and knowledge fusion for procurement fraud detection in Brazil. *CEUR Workshop Proceedings*, 527:3–14.
- Endsley, M. R. (1988). Design and Evaluation for Situation Awareness Enhancement. *Proceedings of the human factors and ergonomics society annual meeting*, 32(2):97–101.
- Endsley, M. R. (2001). Designing for Situation Awareness in Complex Systems. *Proceedings of the 2nd International Workshop on Symbiosis of Humans, Artifacts and Environment*, pages 1–14.

- Mescherin, S. A., Kirillov, I., and Klimenko, S. (2013). Ontology of emergency shared situation awareness and crisis interoperability. *Proceedings - 2013 International Conference on Cyberworlds, CW 2013*, pages 159–162.
- Noy, N. F. and McGuinness, D. L. (2001). Ontology Development 101: A Guide to Creating Your First Ontology. *Stanford Knowledge Systems Laboratory*, page 25.
- Souza, A. M. D., Botega, L. C., Garcia, I. C., and Villas, L. A. (2018). Por Aqui é Mais Seguro : Melhorando a Mobilidade e a Segurança nas Vias Urbanas.

**III Workshop de Computação Urbana
SBRC 2019
Sessão Técnica 3
Estratégias e Aplicações para o
Desenvolvimento Urbano**

Busca de Serviços Baseada em Perfis Sociais dos Objetos em uma Rede SIoT Urbana

Iury Araújo¹, Mikaelly F. Pedrosa¹, Eudisley G. dos Anjos¹,
Fernando Matos¹

¹ Centro de Informática – Universidade Federal da Paraíba

iuryrogerio@ppgi.ci.ufpb.br, mikaelly.felicio@cc.ci.ufpb.br,
[eudisley, fernando]@ci.ufpb.br,

Abstract. *The service interaction provided by objects in IoT networks enables the creation of advanced services to answer application requests. However, the growing number of objects into the IoT network, besides its ad hoc structure, are disturbing some functionalities, such as service discovery. Therefore, when searching for services, the navigability is impaired because the system needs to sweep a great quantity of objects without a previous organization. Social Internet of Things (SIoT) emerged as an alternative to solve several problems faced by IoT through the concept of social networks. In SIoT each object has its own social profile, which contains its characteristics and information, and are organize by relationships. Thus, this research propose a solution for service discovery in a SIoT network. This solution uses the relationships between objects to improve the discovery scalability and considers their social profiles to meet more satisfactorily the requisitions. Simulated results demonstrates the solution performance to answer service requisitions in an urban SIoT network.*

Resumo. *A interação de serviços oferecidos por objetos em redes IoT possibilita a criação de serviços avançados para atender requisições de aplicações. Contudo, o crescimento no número de objetos na rede IoT, além da sua organização ad hoc, acaba dificultando algumas operações, como por exemplo a busca por serviços. Neste caso, ao buscar um serviço, a navegabilidade é prejudicada pois o sistema precisa varrer uma grande quantidade de objetos sem uma organização prévia. A Social Internet of Things (SIoT) surgiu como uma alternativa para solucionar diversos problemas enfrentados pela IoT por meio da utilização do conceito de rede sociais. Na SIoT cada objeto possui um perfil social, contendo suas características e informações, e são organizados através de relacionamentos. Desta forma, este trabalho propõe uma solução para a realização da busca de serviços em uma rede SIoT. Esta solução se utiliza dos relacionamentos entre os objetos para melhorar a escalabilidade da busca e considera os seus perfis sociais para atender de forma mais satisfatória as requisições realizadas. Resultados simulados mostraram o desempenho da solução ao atender requisições por serviços em uma rede SIoT.*

1. Introdução

As áreas urbanas estão cada vez mais populosas o que acarreta no surgimento de diversos problemas relacionados a qualidade de vida, como aumento da poluição, escassez

de recursos, acesso limitado aos serviços públicos, problemas de mobilidade urbana e segurança [Neves et al. 2017]. Dessa forma, existe uma grande procura por soluções inteligentes que, por meio da tecnologia, possam resolver ou diminuir os problemas gerados pela expansão urbana. Muitas dessas soluções são desenvolvidas a partir do conceito de ambientes inteligentes, que pode ser definido como um “pequeno mundo” onde diferentes tipos de objetos inteligentes trabalham para tornar a vida dos seus habitantes mais confortável [Vodanovich et al. 2010]. Dentro desse conceito podemos encontrar diversas áreas de pesquisa como *smart cities*, *smart healthcare* e *smart grid*, que apesar das suas diferenças têm em comum a atuação de diversos objetos inteligentes trabalhando em conjunto para realizar diversas tarefas. Cada vez mais esses objetos adquirem a capacidade de se conectar a Internet, seja para executar ações, adquirir e transmitir dados ou realizar processamento. Essa capacidade facilita a colaboração entre objetos, fornecendo assim uma variedade maior de funcionalidades para as aplicações [Santos et al. 2016].

O cenário descrito é possível através da implementação do paradigma da *Internet of Things* (IoT). A IoT é descrita como a presença pervasiva ao nosso redor de “coisas” ou objetos, que através de esquemas de endereçamento único, estão possibilitados a interagir entre si e cooperar para alcançar um objetivo específico [Atzori et al. 2010]. Esses objetos são os dispositivos tecnológicos do nosso cotidiano, como computadores, *tablets*, celulares, nós sensores, entre outros, que conseguem se conectar a internet. Contudo, o crescente número de objetos conectados é uma preocupação para a IoT. Segundo [Evans 2011], o número de objeto conectados já é maior que a população mundial e deve chegar a 50 bilhões de objetos até 2020. Esse crescimento atinge diretamente o gerenciamento da rede quanto a sua escalabilidade, navegabilidade e a heterogeneidade dos objetos, dificultando a execução de tarefas importantes do paradigma [Jarwar et al. 2018].

O gerenciamento de serviços também é prejudicado com o aumento do número de objetos. Como um objeto pode fornecer um ou mais serviços, o total de serviços na rede pode aumentar vertiginosamente. Tais serviços são abstrações das funcionalidades oferecidas pelos objetos, sem se preocupar com tecnologias ou protocolos. Componentes da arquitetura IoT que lidam com serviços, como os de busca e composição, sofrem diretamente com estes problemas. Como forma de solucioná-los foi proposta a utilização do conceito de redes sociais para organizar os objetos e os seus serviços, surgindo assim um novo paradigma chamado de *Social Internet of Things* (SIoT) [Atzori et al. 2012].

A SIoT introduz a utilização de uma rede social para a organização de objetos, onde cada objeto tem potencial para formar “amizade” com outros objetos através de relacionamentos [Atzori et al. 2011]. A partir da rede social gerada é possível gerenciar a escalabilidade e navegabilidade da rede, a heterogeneidade de objetos e a transmissão de dados, como já é realizada pelas redes sociais humanas [Atzori et al. 2014]. Apesar da SIoT herdar funcionalidades da IoT, muitas delas precisam ser adaptadas para lidar com este novo paradigma, como por exemplo a busca de serviços, que é uma das etapas fundamentais para atender aplicações através de serviços compostos para realizar atividades mais complexas. Outro desafio é utilizar os perfis sociais dos objetos como informação útil para realizar funcionalidades na rede SIoT.

Neste trabalho é proposta uma busca de serviços em uma rede SIoT baseada nos perfis sociais (Busca por Perfil Social - BPS). Nesta busca são utilizados os relacionamentos da rede social para melhorar a sua navegabilidade e escalabilidade, levando também

em consideração as características e informações contidas nos perfis sociais dos objetos para atender de forma mais satisfatória as requisições por serviços. Para verificar a eficiência da solução, foram realizados testes com dados de uma rede SIoT Urbana, onde são comparados os resultados entre o BPS e soluções convencionais da literatura.

O restante deste artigo está organizado da seguinte forma: Na Seção 2 é apresentada uma visão geral das pesquisas realizadas na área de SIoT e suas abordagens para busca de serviços. A Seção 3 expõe os conceitos de SIoT e de sua busca de serviços. Na Seção 4 é apresentada a proposta de busca por perfis sociais. A Seção 5 discorre sobre a implementação da proposta, descreve os testes realizados e apresenta e discute os resultados alcançados. Por último, a Seção 6 conclui o artigo e discute trabalhos futuros.

2. Trabalhos Relacionados

Por ser uma área relativamente recente, a maioria dos trabalhos em SIoT focam em questões conceituais do paradigma, como por exemplo, definições de SIoT, áreas de aplicação, infraestruturas necessárias e propostas de arquiteturas. Poucos trabalhos apresentam implementações do paradigma. Ainda sim, alguns trabalhos focam na capacidade social dos objetos, utilizando suas características, informações e o potencial das interações para fornecer soluções para o gerenciamento de objetos e serviços. Os trabalhos apresentados também se preocupam com as funcionalidades da rede SIoT no que concerne à busca de serviços de forma explícita ou implícita.

O SIoTcampus proposto por [Alves et al. 2015] exemplifica a utilização da SIoT para o disseminamento de informações acadêmicas entre usuários de um campus universitário. Este trabalho propõe uma rede exclusiva para objetos inteligentes em que informações de contexto, como localização, agenda, usuário, etc., são utilizadas como critérios para estabelecer relacionamentos entre os objetos. Como a rede social tem relacionamentos dinâmicos, que podem ser modificados através da aquisição de informações do dono do objeto, é possível propagar pela rede informações, eventos e notícias para um número maior de interessados por aquele contexto.

Uma das possibilidades propostas pela SIoT é utilizar a rede social de objetos como ferramenta para determinar a confiabilidade de objetos na rede. A pesquisa realizada em [Brittes et al. 2017] procura construir uma rede dentro de um ambiente médico para que os objetos possam transmitir e receber dados do sistema hospitalar de forma segura, identificando o nível de confiabilidade dos objetos e bloqueando ataques. A confiabilidade é identificada através da criação de um protocolo de gerenciamento de confiabilidade, que utiliza os relacionamentos e informações dos objetos para determinar o nível de confiabilidade dos mesmos. Em [Wang et al. 2016] também é proposto um modelo que verifica a confiabilidade de objetos na SIoT, sendo baseado no trabalho coletivo de objetos que fornecem operações computacionais e de armazenamento. Neste modelo, tarefas são enviadas aos objetos e ao atendê-las, uma recompensa é entregue aos objetos, ajudando assim a criar relacionamentos e estabelecer confiabilidade entre objetos que cooperam com frequência entre si.

A pesquisa desenvolvida em [Nitti et al. 2016] apresenta uma solução para a implementação da busca de serviços da arquitetura SIoT. Utilizando os relacionamentos construídos pelos objetos da rede social é possível criar um método de busca que com um pequeno número de saltos em uma rede implementada como grafo, encontra objetos

que fornecem os serviços requeridos. Isso é realizado utilizando duas propriedades: a primeira é intrínseca a rede social, baseando-se nas amizades dos objetos; a segunda é externa a rede social, baseando-se na similaridade entre o objeto e a requisição.

A pesquisa em [Chen et al. 2015] propõe a criação de um método para a composição de serviços em SIoT utilizando os relacionamentos formados pelos objetos na rede social. O método encapsula objetos heterogêneos em *web services*. Utilizando o estilo RESTful, as aplicações podem requerer acesso de forma homogênea aos objetos, permitindo a colaboração entre eles para compor processos mais robustos. A composição é possível por meio da modelagem dos relacionamentos entre serviços pela rede social, classificando os relacionamentos em três dimensões: localização, tipo e correlação. Ao se requerer um processo complexo, que precisa ser composto por diversos serviços, o componente realiza uma busca em toda rede social através dos três relacionamentos para encontrar os serviços candidatos, esses candidatos depois são avaliados e os melhores são compostos no processo requerido. O trabalho propõe também um algoritmo de busca que deve ser utilizado para encontrar os serviços utilizando o método de busca em largura em grafos.

Por meio dos trabalhos expostos nessa seção é possível perceber que existem pesquisas preocupadas em endereçar a capacidade social dos objetos, utilizando suas características, informações e o potencial das interações para fornecer soluções para aplicações SIoT, sobretudo dentro dos *smart environments*. Da mesma forma existe uma preocupação crescente em criar funcionalidades da SIoT que se adéquem a utilização da rede social, principalmente para a busca de serviços. Portanto este trabalho apresenta a proposta de um método para a busca de serviços em uma rede SIoT baseado na utilização das informações e características dos perfis sociais dos objetos para melhorar sua eficiência. Diferindo-se de outras soluções por restringir a necessidade de se varrer toda a rede para encontrar os serviços adequados requeridos.

3. Busca de Serviços em uma rede SIoT

A *Social Internet of Things* (SIoT) é um paradigma derivado da IoT que propõe a utilização de redes sociais em conjunto com a infraestrutura já bem estabelecida da IoT. A utilização do conceito de redes sociais em IoT pode levar a grandes vantagens, tais como: (i) a estrutura SIoT pode ser moldada para garantir a navegabilidade da rede, permitindo a eficiência da descoberta de serviços e objetos; (ii) a confiabilidade de objetos podem ser determinadas através da análise da interação com objetos “amigos” e;(iii) modelos utilizados para estudar redes sociais podem ser facilmente aplicados para tratar problemas relacionados a IoT [Atzori et al. 2012]. O conceito formalizado de SIoT determina-o como uma rede social onde cada nó é um objeto inteligente capaz de estabelecer relacionamentos, de forma autônoma, com outros objetos, seguindo regras estabelecidas pelos seus donos [Girau et al. 2017].

Um relacionamento em SIoT é a representação de um vínculo que os objetos podem ter no mundo real, sendo essa representação utilizada para a construção da rede social através da formação de objetos “amigos” que são ligados por relacionamentos. Os trabalhos em SIoT estabelecem cinco tipos de relacionamentos que podem existir entre os objetos [Atzori et al. 2012, Girau et al. 2017, Atzori et al. 2011]: (i) *Parental*: estabelecido entre objetos que pertencem ao mesmo lote de produção, geralmente são objetos

homogêneos, originados no mesmo período e por um mesmo fabricante; (ii) *Co-location*: estabelecido entre objetos que são usados sempre no mesmo local, no caso de sensores, atuadores e objetos de realidade aumentada são considerados os objetos no mesmo ambiente como uma casa ou cidade inteligente; (iii) *Co-work*: estabelecido sempre que objetos cooperam para prover determinada funcionalidade; (iv) *Ownership*: estabelecido entre objetos heterogêneos ou homogêneos que pertencem ao mesmo usuário; e (v) *Social*: estabelecido quando objetos entram em contato, contínuo ou esporádico, porque seus donos entraram em contato durante algum período de suas vidas.

Os objetos SIoT são organizados por meio da rede social de objetos, utilizada em diversos processos do paradigma, como a composição e busca de serviços. A rede social é composta por representações da existência física dos objetos na rede, chamados de perfis sociais. Cada perfil, contém atributos relacionados ao objeto que representa, como sua identificação, os amigos formados pelos seus relacionamentos com outros objetos, os serviços que oferece e suas características e informações [Jadhav and Patil 2016]. As características de um objeto são os dados inerentes a sua constituição e que são geralmente imutáveis, por exemplo seu fabricante, lote de produção, quantidade de núcleos de processamento e de memória primária e secundária, entre outros. Enquanto, que as informações de um objeto são dados sobre o objeto que estão em constante mudança, como capacidade de armazenamento, estado da bateria, confiabilidade do objeto, configurações de segurança, entre outros [Kim et al. 2017].

A busca de serviços geralmente utilizada na literatura, como é o caso de Chen et al. [Chen et al. 2015], utiliza apenas os relacionamentos para procurar na rede por objetos que tenham serviços que atendam às requisições. Devido à peculiaridade de tais relacionamentos, os resultados da busca podem ser diferentes para os mesmos objetos. A Figura 1 apresenta um exemplo de como diferentes relacionamentos afetam a busca em redes SIoT. Cada perfil social é representado por um nó e cada aresta identifica uma relação de amizade entre objetos. São utilizados como exemplos os relacionamentos *Ownership* e *Co-work* (Figuras 1(a) e 1(b), respectivamente). Quando uma requisição é enviada para o componente de busca, ele identifica o objeto que requisitou o serviço (requerente) e o torna o objeto inicial da busca. Isso é possível pois qualquer requerente é um objeto que faz parte da rede social. Em ambas figuras, o requerente é o objeto *D*. A partir do requerente, o método de busca procura saber se os serviços que ele oferece atendem a requisição. Cada cor representa o conjunto de objetos acessados a cada nível de busca. Na Figura 1(a), a partir de *D*, os seus amigos, que são os objetos *B*, *C*, *E* e *G*, são pesquisados. A busca prossegue nos amigos dos objetos anteriores (*A*, *F*, *H*, *I*, *L* e *O*), seguindo assim até a busca ter percorrido todos os objetos na rede. Da mesma forma acontece no grafo da Figura 1(b), onde a busca é realizada utilizando um relacionamento diferente. Neste caso, é possível visualizar que o resultado será diferente, já que não há o relacionamento *Co-work* entre o grupo de objetos *N*, *M* e *I* e o restante da rede.

Em ambos os casos, a busca de serviços é realizada por uma estratégia de busca em largura em grafos, que verifica todos os objetos da rede social para encontrar os serviços que atendem melhor a requisição. Esta abordagem retorna todos os serviços que podem atender à requisição, incluindo diversos serviços que realizam as mesmas atividades ou não atendem requisitos de outras funcionalidades da rede SIoT. Por isso, uma etapa adicional para filtrar os serviços é necessária. Tal etapa pode comprometer o desempenho da

aplicação que realiza a requisição.

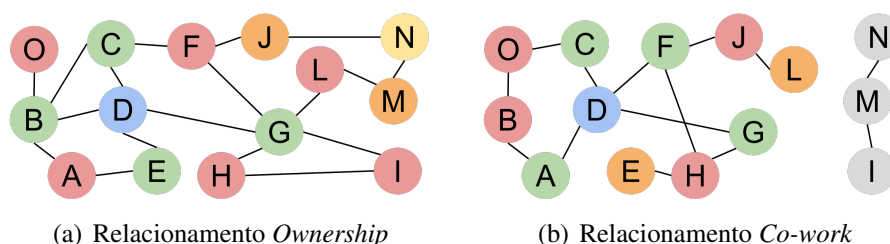


Figura 1. Grafos representando relacionamentos da rede SIoT.

4. Proposta

O método de busca BPS proposto neste trabalho utiliza as características e informações contidas nos perfis sociais dos objetos como critérios adicionais de comparação para verificar se os serviços de um objeto, e ele próprio, atendem aos requisitos da requisição. O BPS também introduz uma nova condição de parada para busca em largura, uma vez que todos os serviços requisitados sejam encontrados, independentemente da quantidade de objetos visitados, a busca é interrompida. Além disso, como na estratégia de busca em largura, o BPS também finaliza sua execução quando todos os objetos da rede já foram pesquisados. Desse modo, o método garante que uma resposta satisfatória para a requisição possa ser encontrada em menos tempo, balanceando a qualidade dos serviços encontrados e o tempo de busca, porém deixando aberta a possibilidade do pior cenário, que acontece quando não é possível encontrar na rede todos os serviços para atender tal requisição.

A Figura 2 apresenta a mensagem de requisição que é recebida pelo BPS, a qual é formada por três atributos: (i) Objeto Requerente (OR), contendo a identificação do objeto que requisitou a busca; (ii) uma Lista de Serviços (LS), contendo os serviços requeridos e que serão buscados pelo BPS; e (iii) a Lista de Requisitos (LR), contendo requisitos que serão verificados e deverão ser atendidos pelas características e informações dos objetos, como por exemplo, tipo de tecnologia de transmissão, quantidade de memória, confiabilidade do objeto, dentre outros.

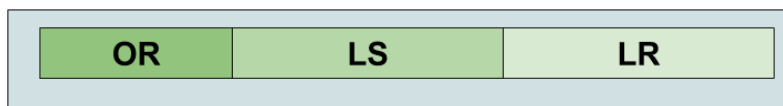


Figura 2. Requisição de busca de serviços e seus atributos.

A Figura 3 ilustra o processo BPS em uma rede com sete objetos que recebe uma requisição por busca de serviços. Como todo objeto que requer uma busca de serviços também faz parte da rede social de objetos, o BPS utiliza o perfil social do requerente na rede como o objeto inicial da busca. Com os outros atributos da requisição o BPS começa a verificar se os objetos amigos de *A* possuem os serviços requisitados (Figura 3(b)). O objeto inicial *A* é marcado como não usável, pois, ou não possui os serviços buscados ou não atende todos os requisitos impostos, mesmo que, tenha o serviço requerido. Ao verificar os objetos *B* e *D*, BPS percebe que os dois tem serviços requeridos e que eles

atendem todos os requisitos, portanto suas identificações são adicionadas a uma lista com o possível resultado. Como não foram encontrados todos os serviços requeridos a busca é expandida para os objetos *F* e *C* (Figura 3(c)), que possuindo os serviços que faltavam e atendendo aos requisitos, completam a execução da busca.

Como exemplo podemos considerar um objeto que pertence a uma rede SIoT. Nessa rede, os objetos possuem como uma das suas características, os tipos de conexões sem fio que podem realizar, como bluetooth, 3G, GSM e Wi-Fi, sendo permitido possuir qualquer combinação dessas quatro conexões. Esse objeto possui somente a capacidade de conectar-se por Wi-Fi, enquanto que a requisição de busca pede por serviços de objetos que podem se comunicar via Wi-Fi e 3G, mesmo que esse objeto tenha um serviço que atenda a requisição ele não será considerado, pois não atende aos requisitos mínimos da busca. Esse tipo de abordagem facilita a manipulação dos resultados da busca por outras funcionalidades da rede SIoT, como a composição de serviços, pois garante que os serviços encontrados não precisam passar por uma nova avaliação de checagem de requisitos.

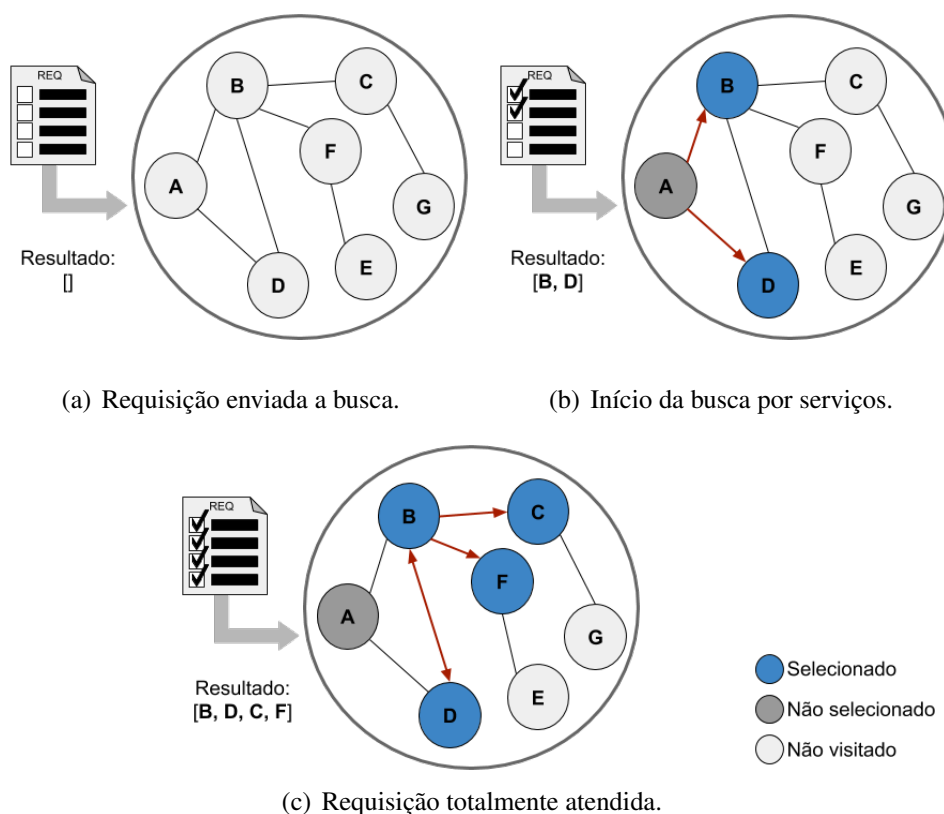


Figura 3. Funcionamento do método BPS.

O Algoritmo 1 descreve o método BPS para busca de serviços considerando os perfis dos objetos. A requisição é tratada como três blocos separados, (i) lista serviços requisitados (*required_service_list*) que representa a lista contendo os serviços devem ser encontrados; (ii) Lista de requisitos (*requirement_list*) que contém todos os requisitos mínimos que os objetos devem possuir; e (iii) Objeto requerente (*requester*) que será utilizado como ponto inicial da busca na rede SIoT. Além da requisição o algoritmo também

recebe como entrada a rede social *social_network* que contém todos os perfis sociais dos objetos na rede

Algoritmo 1 Busca de serviços Baseada em Perfis Sociais (BPS)

Entrada: *required_service_list*, *requirement_list*, *requester*, *social_network*

Saída: *service_list*

Require: *requester* deve ser um objeto pertencente à rede social

```

1: search_list.adiciona(social_network.recupera(requester))
2: while search_list não é vazia do
3:   head ← search_list.removeHead()
4:   if required_service_list foi atendida then
5:     break
6:   else
7:     if head.visualizado = true then
8:       continue
9:     else if head atende required_service_list and head atende requirement_list
       then
10:      service_list.adiciona(head.services())
11:    end if
12:  end if
13:  head.visualizado ← true
14:  search_list.adiciona(social_network.recuperaAmigos(head))
15: end while
16: return service_list

```

Inicialmente, o BPS adiciona o perfil social do objeto requerente a *search_list* e em seguida inicia o laço buscando serviços por toda a rede até que a *required_service_list* tenha sido completamente atendida ou *search_list* esteja vazia. Para cada objeto retirado da lista (*head*) é verificado se o perfil social já foi pesquisado (linha 7). Em caso positivo, verifica-se o próximo objeto. Caso contrário, é verificado se um ou mais serviços da *head* correspondem aos serviços requeridos em *required_service_list* e se aquele objeto atende os requisitos mínimos dispostos na *requirement_list* (linha 9). Os serviços que atenderem ambas as condições são adicionados à *service_list* (linha 10). O BPS marca então o perfil social contido em *head* como já visualizado e adiciona os seus amigos à *search_list* (linha 14), iniciando novamente o processo. Quando o algoritmo termina sua execução por uma das duas condições de parada os serviços encontrados são entregues como resposta da requisição. O resultado final do BPS permite três estados diferentes, (i) não encontrar nenhum serviço que atenda à requisição, (ii) encontrar somente alguns serviços, atendendo então parcialmente a requisição, ou (iii) encontrar todos os serviços de forma a atender por completa à requisição.

5. Ambiente de teste e avaliação

O método BPS foi implementado em Python e testado utilizando o *dataset* de uma rede SIoT urbana disponibilizado por [Atzori et al. 2016], o qual foi criado no contexto do projeto SmartSantander, executado nas cidades de Belgrade, Guildford, Lübeck and Santander. Porém, o único *dataset* disponível é o da cidade de Santander.

O *dataset* é composto por 16216 dispositivos (objetos da rede social) de diversas tecnologias. Ele fornece informações sobre os relacionamentos da rede social e os serviços que cada objeto oferece. Os serviços são representados por números que indicam algum serviço que o objeto pode prover, sendo que cada objeto pode possuir até oito tipos de serviços diferentes de um total de dezesseis distintos. Como o *dataset* não trabalha com os serviços sendo entidades únicas, com sua própria identificação, o BSP foi implementado com a alteração de que ao invés de retornar os serviços encontrados, são retornados os objetos que os provem. O *dataset* apresenta os relacionamentos dos objetos através de matrizes com valores *booleanos* que representam a existência, ou não, de uma “amizade” entre os objetos. O relacionamento *ownership* foi utilizado para realizar os testes deste trabalho. A Figura 4 ilustra a organização de uma matriz de relacionamento onde o número um identifica uma amizade entre dois objetos. Por exemplo os objetos (A,C) e (E,F) são amigos pois a interseção das suas colunas e linhas possuem o valor um.

		Objetos					
		A	B	C	D	E	F
Objetos	A	0	0	1	0	0	1
	B	0	0	0	1	0	0
	C	1	0	0	0	1	0
	D	0	1	0	0	0	0
	E	0	0	1	0	0	1
	F	1	0	0	0	1	0

Figura 4. Matriz de relacionamento do *dataset*.

Para atestar a eficiência do BPS, foram utilizadas duas métricas: tempo de busca e escalabilidade da solução. O tempo de busca é o tempo que o algoritmo leva para responder a requisição, sendo calculado a partir do momento que o algoritmo começa a consultar a rede social para encontrar serviços até o momento em que a lista de objetos encontrados seja retornada. Já a escalabilidade avalia o comportamento da solução à medida que aumenta o número de objetos na rede. Os testes foram realizados em uma máquina com processador i7-8750H hexa-core de 2666 MHz e 16GB de RAM DDR4.

O BPS foi comparado com o algoritmo proposto por Chen et al. [Chen et al. 2015], que utiliza o método de busca em largura. Para testar o tempo de busca, um arquivo foi criado contendo todas as requisições que foram realizadas durante os testes. A utilização do arquivo garante que as requisições que foram realizadas no método BPS eram idênticas às realizadas no algoritmo de Chen. Cada linha do arquivo representa uma requisição, contendo as seguintes informações: o objeto requerente, os serviços que se desejam encontrar e os requisitos que serão utilizados. Esta última informação é utilizada apenas pelo método BPS.

Cada atributo da requisição foi gerado aleatoriamente obedecendo determinados valores mínimos e máximos. O objeto requerente é obrigatoriamente um objeto pertencente a rede. Os serviços desejados são gerados a partir de uma lista de dezesseis serviços

possíveis do *dataset*. Dentre estes, são escolhidos quatro para formar os serviços desejados de cada requisição, de forma que um serviço já escolhido não possa ser escolhido novamente. Para gerar os requisitos, foi necessário primeiro criar os perfis sociais para os objetos do *dataset*, definindo quais características e informações seriam associadas ao perfil e tornariam-se seus atributos, esses, podem ser observados na Tabela 1. O próximo passo foi preencher os atributos contidos nos perfis sociais, seguindo os valores permitidos na Tabela 1. Finalmente, os requisitos foram gerados e adicionados à requisição, sendo esses baseados nos atributos dos perfis sociais. Valores iniciais aleatórios foram utilizados para cada requisito, respeitando determinados limites impostos na Tabela 1. No total, 2380 requisições foram criadas para os testes.

Tabela 1. Atributos dos perfis sociais e requisitos da requisição.

Atributos	Valor Mínimo	Valor Máximo
Memória primária	256 MB	4 GB
Memória secundária	0 MB	512000 MB
Núcleos de processamento	1	8
Conectividade	1	5
Capacidade de armazenamento	15%	75%
Estado da bateria	10%	100%
Confiabilidade do objeto	45	95

O teste de escalabilidade deve verificar como se comporta a solução ao aumentar o número de objetos da rede. Para realizá-lo, foi necessário dividir a rede em diversos recortes que representam redes de tamanhos variados, sendo elas compostas de quinhentos, mil, dois mil, quatro mil, doze mil e dezesseis mil objetos. Esses recortes precisam obedecer a distribuição de probabilidade dos tipos de objetos presentes no *dataset* inicial. A Tabela 2 apresenta as porcentagens de cada tipo de dispositivo.

Tabela 2. Distribuição dos tipos de dispositivos no *dataset*.

Tipo de Dispositivo	Porcentagem
<i>Smartphone</i>	22.45%
Computador Pessoal	20.72%
Carro	13.57%
Impressora	13.07%
Tablet	9.87%
<i>Smart Fitness</i>	5.43%
Estacionamentos	4.17%
Sensores Residenciais	3.70%
Iluminação Pública	3.12%
<i>Smartwatch</i>	1.23%
Transporte	0.88%
Sensores de Ambiente ou de clima	0.86%
Pontos de Interesse	0.58%
Alarmes de Segurança	0.23%
Indicador	0.06%
Caminhão de Lixo	0.04%

Uma vez garantido que os recortes tinham a mesma distribuição do conjunto de dados original, foi necessário lidar com outro problema causado pelos recortes. A aleatoriedade de escolha dos objetos dificulta que objetos que tenham relacionamentos entre

si sejam escolhidos juntos, principalmente para recortes de tamanhos pequenos. Esta situação prejudica a navegabilidade e dificulta a execução da busca, pois é possível que existam diversos objetos ou grupos de objetos isolados na rede. Como solução, foi necessário acrescentar manualmente novos relacionamentos, de forma que todos os perfis sociais tenham pelo menos um relacionamento com outro objeto do mesmo recorte, dessa forma garantindo a navegabilidade da rede.

5.1. Resultados

Foram realizadas dez execuções para cada recorte da rede utilizando cada um dos algoritmos BPS e Chen. Para cada teste a mediana do tempo de busca das 2380 requisições foi calculada, a mediana foi utilizada como forma de evitar que *outliers* impactassem os resultados. Em seguida foi calculada a média de todas as medianas dos testes realizados para cada recorte. A Tabela 3 apresenta a mediana do tempo de busca, em segundos, de cada recorte. Analisando os valores, é possível perceber que o algoritmo BPS consegue responder as requisições de todos os recortes em um tempo expressivamente mais baixo que o algoritmo de Chen. Este comportamento pode ser visualizado principalmente nos recortes 4000 e 8000, onde o tempo de busca do BPS é menor 93,98% e 94,32% do que Chen, respectivamente.

Tabela 3. Mediana do tempo de busca, em segundos, para cada requisição nos recortes de teste.

Recorte (nº de objetos)	Chen (s)	BPS (s)
500	0,02594487	0,01072197
1000	0,10345836	0,01405903
2000	0,38549267	0,03829563
4000	1,37108796	0,08257948
8000	4,28259313	0,2434388
12000	0,56687746	0,00898465
16000	0,00587981	0,00399698

Algumas surpresas foram observadas durante a comparação dos tempos de busca entre os algoritmos, sendo essa comparação apresentada na Figura 5. Entre os recortes de 500 a 8000 é possível perceber que existe um crescimento no tempo médio necessário para que uma requisição seja respondida, sendo esse comportamento já esperado com o aumento de objetos na rede social. Entretanto, nos recortes 12000 e 16000 temos um efeito totalmente reverso com uma diminuição busca do tempo de brusca. Por exemplo, no recorte de 12000, o algoritmo de Chen tem uma queda de tempo na razão de 86,76%, enquanto que o BPS tem uma redução de tempo de 96,30%.

Uma das possibilidades para esse comportamento é que a adição de novos objetos criou uma rede muito mais conectada, tornando-se assim mais fácil para os algoritmos alcançarem objetos que anteriormente tinham poucos relacionamentos e estavam mais isolados. Outra possibilidade é que a aleatoriedade usada para criar relacionamentos e para escolher objetos da rede tenham tornado o seu funcionamento mais complexo. Um grande indício seria o fato de que a maior parte das alterações realizadas na rede foram realizadas no recortes menores, de 500 a 8000, enquanto que poucas alterações foram efetuadas em 12000 e ainda menos no recorte 16000. Sendo este o caso, esses resultados podem ser um claro testemunho da eficiência das soluções SIoT, pois quanto menor for a alteração na rede, provida por [Atzori et al. 2016], melhor é a sua eficiência.

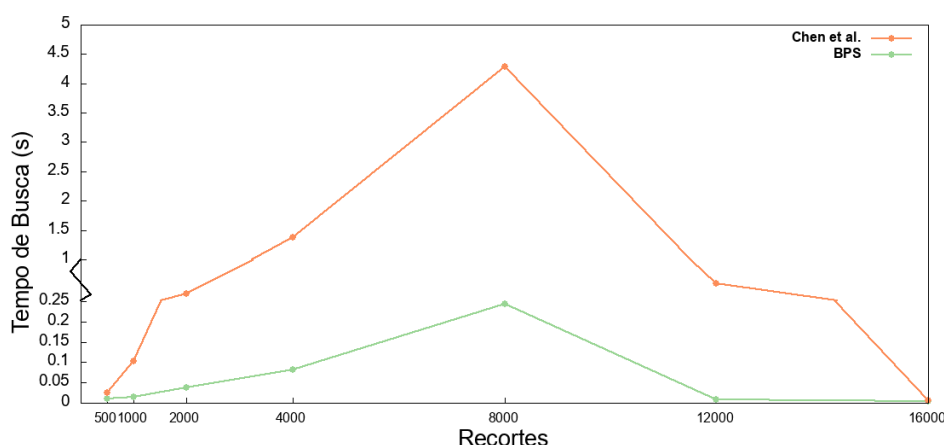


Figura 5. Gráfico comparativo do comportamento do tempo de busca nos algoritmos Chen et al. e BPS.

Ao comparar a execução dos algoritmos por meio da análise da Figura 5 é fácil perceber como o BPS consegue manter o tempo de busca muito mais baixo que o algoritmo Chen et al., sendo geralmente 94,03% mais rápido e não demorando mais que, em média, 0,25 segundos para responder a uma requisição. Outro aspecto importante a ser observado e analisado é o comportamento dos algoritmos em relação a escalabilidade da rede. Obviamente 16000 objetos ainda é um número bem pequeno para testar a escalabilidade desses algoritmos em uma rede que hoje já comporta bilhões de objetos. Entretanto, indícios que o algoritmo BPS é escalável podem ser observados através do crescimento controlado do tempo de busca em relação a expansão do número de objetos na rede e também da queda expressiva dele em recortes maiores.

Em contraste ao BPS, Chen et al. tem um crescimento de tempo de busca bem acentuado até o recorte 8000, onde logo em seguida, cai para valores bem próximos aos do BPS. Apesar de existir uma sugestão que o Chen et al. não é escalável, os resultados observados mostram que é necessário verificar o comportamento desse algoritmo com um número ainda maior de objetos por recorte para ter um veredito final. Entretanto, é possível afirmar que a solução proposta neste trabalho por meio do algoritmo BPS mostrou-se mais escalável que o método proposto por Chen et al. [Chen et al. 2015]. Outra informação fundamental indicada por meio dos testes apresentados na Figura 5 é a necessidade de verificar como a organização dos perfis sociais e seus relacionamentos afetam os algoritmos de busca em uma rede SIoT.

A Tabela 4 apresenta a média de objetos retornados na busca em cada recorte. É possível observar que o número de objetos retornados por Chen é muito maior que o do BPS (em média 99,09% mais objetos). Apesar do método de Chen poder encontrar sempre os melhores serviços para cada requisição, ainda é necessária uma etapa adicional para escolher quais objetos, dentre os encontrados, serão utilizados para satisfazer a requisição, o que pode impactar no desempenho da aplicação que está realizando tal requisição. Já no BPS não é necessária esta etapa adicional, uma vez que o método retorna, assim que encontra, os objetos necessários para atender, por completo, a requisição. Dessa forma, aliado a um tempo de busca mais rápido, o BPS mostra que consegue atender mais satisfatoriamente as requisições de busca de serviços por levar em consideração o perfil social

dos objetos da rede.

Tabela 4. Média do número de objetos retornados por requisição.

Recorte (nº de objetos)	Chen	BPS
500	344	3
1000	679	5
2000	1340	8
4000	2459	17
8000	3502	37
12000	3789	42
16000	4102	39

6. Conclusão

Este trabalho propôs uma solução para a busca de serviços em uma rede SIoT mais eficiente, utilizando as características e informações contidas no perfil social do objeto para atender mais satisfatoriamente as requisições de busca. Para isso um método de busca de serviços, chamado de BPS, foi proposto a partir da modificação de um algoritmo de busca em largura em grafo. O BPS adiciona uma nova condição de parada para melhorar a eficiência da busca e insere a utilização das características e informações dos perfis sociais dos objetos para verificar a qualidade dos serviços encontrados.

Testes simulados utilizando o *dataset* de uma rede SIoT urbana constataram que o algoritmo BPS realiza uma busca de serviços, 94,03% em média, mais rápida que método proposto na literatura. Analogamente, os resultados desses testes também indicam uma boa adaptabilidade do BPS para lidar não só com o aumento do número de objetos na rede, mas também com a complexidade da sua organização. Os testes também indicaram que existe a necessidade de estudar como os relacionamentos entre os objetos estão sendo formados na rede social, sendo um dos pontos que podem levar a melhorias na eficiência e eficácia dos algoritmos de busca de serviços em SIoT.

Como trabalhos futuros, novos testes serão realizados para determinar o comportamento do BPS com requisições mais variadas, contendo um número maior de serviços requeridos e também uma variação maior das características apresentadas. Outro teste que deverá ser essencialmente realizado é a verificação do comportamento dos algoritmos de busca ao lidar com os diversos tipos de relacionamentos possíveis em uma rede SIoT, identificando se existe uma ordem de busca específica entre esses relacionamentos que beneficie a busca de serviços. Além dos testes é possível focar esforços na construção de outras funcionalidades da rede SIoT que dependem diretamente da busca de serviços, como métodos de gerenciamento da rede social e composição de serviços. Outro trabalho futuro pode ser realizado por meio da identificação de outras estratégias de busca de serviços utilizadas na *Internet of Things* e verificar a possibilidade de adaptá-las para a busca em SIoT, incluindo métodos que usem inteligência artificial para isso.

Referências

Alves, T. M., da Costa, C. A., da Rosa Righi, R., and Barbosa, J. L. V. (2015). Exploring the social internet of things concept in a univeristy campus using nfc. In *Computing Conference (CLEI), 2015 Latin American*, pages 1–12. IEEE.

- Atzori, L., Carboni, D., and Iera, A. (2014). Smart things in the social loop: Paradigms, technologies, and potentials. *Ad Hoc Networks*, 18:121–132.
- Atzori, L., Iera, A., and Morabito, G. (2010). The internet of things: A survey. *Computer networks*, 54(15):2787–2805.
- Atzori, L., Iera, A., and Morabito, G. (2011). Siot: Giving a social structure to the internet of things. *IEEE communications letters*, 15(11):1193–1195.
- Atzori, L., Iera, A., Morabito, G., and Nitti, M. (2012). The social internet of things (siot)—when social networks meet the internet of things: Concept, architecture and network characterization. *Computer networks*, 56(16):3594–3608.
- Atzori, L., Nitti, M., and Marche, C. (2016). Social internet of things. In <http://www.social-iot.org/index.php?p=downloads>.
- Brittes, M. P., Schneider Jr, B., and Wille, E. C. (2017). Trustworthiness management through social relationships in internet of medical things. *Journal of Communication and Information Systems*, 32(1).
- Chen, G., Huang, J., Cheng, B., and Chen, J. (2015). A social network based approach for iot device management and service composition. In *Services (SERVICES), 2015 IEEE World Congress on*, pages 1–8. IEEE.
- Evans, D. (2011). The internet of things: How the next evolution of the internet is changing everything. *CISCO white paper*, 1(2011):1–11.
- Girau, R., Martis, S., and Atzori, L. (2017). Lysis: A platform for iot distributed applications over socially connected objects. *IEEE Internet of Things Journal*, 4(1):40–51.
- Jadhav, B. and Patil, S. (2016). Wireless home monitoring using social internet of things (siot). In *Automatic Control and Dynamic Optimization Techniques (ICACDOT), International Conference on*, pages 925–929.
- Jarwar, M., Kibria, M., Ali, S., and Chong, I. (2018). Microservices in web objects enabled iot environment for enhancing reusability. *Sensors*, 18(2):352.
- Kim, J. E., Fan, X., and Mosse, D. (2017). Empowering end users for social internet of things. In *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation*, pages 71–82. ACM.
- Neves, A. R. d. M., Sarmanho, K. U., Nascimento Jr, F. C., and Meiguins, B. S. (2017). Iniciativa smart campus: um estudo de caso em progresso na universidade federal do par. In *Anais do I Workshop de Computao Urbana (COURB 2017)*, volume 1. SBC.
- Nitti, M., Pilloni, V., and Giusto, D. D. (2016). Searching the social internet of things by exploiting object similarity. In *Internet of Things (WF-IoT), 2016 IEEE 3rd World Forum on*, pages 371–376. IEEE.
- Santos, B. P., Silva, L., Celes, C., Borges, J. B., Neto, B. S. P., Vieira, M. A. M., Vieira, L. F. M., Goussevskaia, O. N., and Loureiro, A. (2016). Internet das coisas: da teoria  prtica. *Minicursos SBRC-Simpsio Brasileiro de Redes de Computadores e Sistemas Distribuidos*.
- Vodanovich, S., Sundaram, D., and Myers, M. (2010). Research commentary—digital natives and ubiquitous information systems. *Information Systems Research*, 21(4):711–723.
- Wang, K., Qi, X., Shu, L., Deng, D.-j., and Rodrigues, J. J. (2016). Toward trustworthy crowdsourcing in the social internet of things. *IEEE Wireless Communications*, 23(5):30–36.

Uma Análise dos Protocolos de Comunicação para Internet das Coisas*

Arthur Brito Cosmi¹, Vinícius Fernandes Soares Mota¹

¹Departamento de Informática – Universidade Federal do Espírito Santo (UFES)
Vitória – ES – Brasil

arthurcosmi@gmail.com, vinicius.mota@inf.ufes.br

Abstract. *With the growth of devices connected to the Internet, several application layer protocols specific for the Internet of Things (IoT) appear every day. This high number of protocols bring doubts to developers about what to use in their projects. This work aims to assist developers in the decision making in their IoT projects. This work presents a qualitative analysis of the MQTT, MQTT-SN, CoAP and AMQP protocols. In a qualitative way, the domains of application and requirements of the applications were analyzed to map the protocol choice. After that, the MQTT and CoAP protocols were analyzed quantitatively using devices with low processing and memory capabilities. Results show that MQTT outperforms CoAP when there are several other devices accessing the same access point.*

Resumo. *Com o crescimento de dispositivos conectados à Internet, diversos protocolos da camada de aplicação específicos para Internet das Coisas (IoT) surgem a cada dia. Esta alta quantidade de protocolos trazem dúvidas aos desenvolvedores sobre o que utilizar em seus projetos. O objetivo deste trabalho é auxiliar os desenvolvedores na tomada de decisão em seus projetos IoT. Este trabalho apresenta uma análise qualitativa dos protocolos MQTT, MQTT-SN, CoAP e AMQP. De modo qualitativo, foram analisados os domínios e requisitos das aplicações para mapear a escolha dos protocolos. Após isso, os protocolos MQTT e CoAP foram analisados quantitativamente utilizando-se dispositivos com baixo poder de processamento e memória. Os resultados demonstram que o MQTT tem melhor desempenho que o CoAP quando existe diversos outros dispositivos acessando o mesmo ponto de acesso.*

1. Introdução

A Internet das Coisas (*Internet of Things* - IoT) é uma área tecnológica em constante crescimento nos últimos anos. De fato, são estimados 50 bilhões de objetos inteligentes e conectados à Internet em 2020, propiciando a cada dia dezenas de soluções para melhorar a qualidade de vida das pessoas e automatizar tarefas em residências, empresas e indústrias [Atzori et al. 2010]. Além disso, nos cenários urbanos, os objetos conectados trazem uma nova perspectiva para interação entre residentes e espaços físicos de uma cidade [Zanella et al. 2014].

*O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), do CNPq, da FAPES e do projeto RNP FUTEBOL.

No contexto da computação urbana, o fluxo de dados consiste em sensoriar, transmitir e armazenar para, posteriormente, analisar os dados coletados. Sistemas embarcados, conhecidos como *System-On-Chip* (SOC), permitem a utilização de sensores, atuadores e transmissão de dados em *hardwares* com tamanho restrito. No geral, isto implica em processamento, memória primária e armazenamento limitados. Por outro lado, estes dispositivos apresentam baixo consumo energético.

No entanto, projetos de sensoriamento e transmissão de dados requerem que os desenvolvedores definam a arquitetura do hardware e os protocolos de comunicação que serão utilizados. O projeto de um dispositivo leva em consideração, custo, capacidade de processamento, memória e armazenamento, rede de comunicação e fonte de energia. Já a escolha de protocolos de comunicação dependerá do desempenho da arquitetura bem como de características da aplicação.

Diversos protocolos de comunicação, que atuam na camada de aplicação, específicos para Internet das Coisas têm sido propostos na literatura [Al-Fuqaha et al. 2015]. Os protocolos da camada de aplicação IoT permitem uma melhor interação entre os dados sensoreados pelos dispositivos e à aplicação alvo. Esses protocolos vão de versões similares ao HTTP modificada para o mundo de objetos, como o CoAP, à protocolos de distribuição de mensagens em modo *multicast*, como o MQTT [Naik 2017]. O grande problema é que não existe uma definição muito precisa de quando se deve utilizar determinado protocolo, nem quais os impactos provocados por essa escolha. A partir da análise das aplicações disponíveis no mercado é possível mapear os domínios em que esses protocolos podem ser utilizados de maneira mais eficiente e analisar as métricas de desempenho dos mesmos.

Com o objetivo de auxiliar desenvolvedores na escolha de protocolos de aplicação em projetos que utilizem SOCs, este trabalho apresenta um estudo qualitativo e quantitativo dos principais protocolos. A partir de uma ampla revisão da literatura, apresentamos um estudo quantitativo mapeando como cada protocolo analisado reage a um conjunto de requisitos de aplicações IoT. Após isso, dois protocolos foram implementados para serem analisados e comparados em um dispositivo SOC com processamento e memória limitado, o ESP8266. O *testbed* desenvolvido permitiu analisar e comparar o desempenho dos protocolos quando o dispositivo estava conectado isoladamente à rede sem fio e quando havia tráfego de fundo. Foram analisados a taxa de entrega, o tempo de transmissão e a retransmissão de pacotes. As principais contribuições deste trabalho são:

- Classificar e categorizar, qualitativamente, os protocolos de aplicação de acordo com os requisitos de cada domínio de aplicação. Foram considerados os domínios residenciais, comerciais e industriais;
- Prototipar uma aplicação e analisar, quantitativamente, o desempenho de protocolos de aplicação em relação à taxa de entrega, retransmissão de pacotes e tempo de transmissão, considerando um cenário isolado e um cenário com tráfego de fundo;

O restante deste trabalho está organizado da seguinte forma: A Seção 2 faz uma revisão dos protocolos de aplicação IoT. Os trabalhos relacionados são apresentados na Seção 3. Uma avaliação qualitativa dos protocolos de aplicação é discutida na Seção 4. A Seção 5 apresenta uma comparação quantitativa, utilizando um *testbed*, entre os protocolos MQTT e CoAP. Os resultados dos experimentos são discutidos na Seção 6. Por fim, a Seção 7 conclui este trabalho e apresenta as considerações finais.

2. Protocolos da camada de aplicação

Esta seção introduz os principais protocolos utilizado para transmissão de dados de dispositivos móveis para a base de dados das aplicações em Internet das coisas. Os protocolos que conseguiram aderência da indústria, tais como *Microsoft*, *Amazon*, *Google*, entre outras, são os protocolos MQTT (*Message Queuing Telemetry Transport*), MQTT-SN, CoAP (*Constrained Application Protocol*), AMQP (*Advanced Message Queuing Protocol*). Deste modo, esses protocolos serão descritos brevemente na Seção 4 para apresentarmos uma avaliação qualitativa sobre os mesmos.

O *Message Queuing Telemetry Transport* - MQTT - é um protocolo de comunicação sobre a pilha TCP/IP, criado pela IBM em 1999. O MQTT ganhou popularidade por utilizar o modelo *Publisher/Subscriber* [Mazzer et al. 2015]. Neste modelo, toda comunicação é intermediada por um dispositivo chamado concentrador (*Broker*). Dispositivos chamados *Publishers* enviam suas mensagens para um *Broker* utilizando uma estrutura chamada de *tópico*, que funciona como um assunto de interesse. Dispositivos chamados de *Subscribers* podem se inscrever em vários tópicos junto ao *Broker*. O *Broker* encaminha as mensagens publicadas em um determinado tópico para todos os dispositivos inscritos no mesmo.

O MQTT-SN foi desenvolvido para permitir o uso do MQTT em redes não TCP/IP, tais como *Zigbee* ou UDP/IP. Para isto, o MQTT-SN implementa seu próprio controle de fluxo e retransmissão, além de possuir algumas características interessantes como a criação de um mecanismo de identificadores de tópicos, onde cada cliente pode pedir ao *Broker* que gere um número para o tópico. Com isso, ao invés de utilizar o nome do tópico em UTF-8 para a troca de mensagens, é utilizado o identificador [Truong and Linh 2013].

Message-Oriented Middleware ou MOM é uma arquitetura que funciona como uma camada única de compartilhamento de informações para conectar sistemas distribuídos. O *Advanced Message Queuing Protocol* - AMQP - veio para padronizar essa comunicação entre MOM's de forma a garantir a interoperabilidade e confiabilidade na entrega das mensagens [Appel et al. 2010]. Além disso, este protocolo define como as mensagens serão publicadas e consumidas e o tempo de vida delas. O AMQP tem a vantagem de manter as mensagens no servidor e enviá-las para o consumidor assim que este estiver disponível. Assim, vários dispositivos podem utilizar funções de *sleep* e economizar energia o que é bastante aceitável em aplicações IoT. Outra vantagem é a segurança em cima das mensagens enviadas. A característica de mensagens imutáveis garante o envio das mensagens de ponto a ponto.

O *Constrained Application Protocol* - CoAP - é um protocolo desenvolvido para ser compatível com o modelo HTTP REST. O CoAP utiliza comunicação no modelo requisição/resposta. Neste modelo, dispositivos clientes requisitam uma informação (GET) e os dispositivos, atuando como servidores, respondem a informação (POST) [Mazzer et al. 2015]. O protocolo CoAP foi projetado com um cabeçalho fixo de quatro *bytes* e utiliza o UDP como protocolo de transporte. As requisições são realizadas por meio de URI's. Além disso, o CoAP implementa sua própria forma de confiabilidade para entrega de mensagens e um mecanismo de retransmissão. Para diminuir ainda mais o tráfego de rede, o protocolo permite o envio de mensagens em *piggyback* utilizando *MessageIDs* e *Tokens* para garantir o envio correto da informação [Shelby et al. 2014].

3. Trabalhos Relacionados

Com o aumento das aplicações na área de IoT, vários estudos foram realizados para verificar a eficiência dos protocolos [Mun et al. 2016, Chaudhary et al. 2017]. A heterogeneidade dos dispositivos bem como dos meios físicos de comunicação, tais como redes *wireless*, cabeada e de sinais 2/3/4G, afetam o desempenho dos protocolos. Portanto, a escolha do protocolo de comunicação torna-se um desafio aos desenvolvedores de aplicações IoT.

Em [Mun et al. 2016], os autores compararam cinco protocolos de comunicação (MQTT, CoAP, MQTT-SN, WEBSOCKET, TCP). Para isto, os autores montaram um *testbed* utilizando dispositivos *Raspberry Pi* comunicando-se com uma base de dados local e outra em Tóquio. Foram medidos o tempo de transmissão, consumo total de energia e uso de memória e CPU. Nos cenários analisados, o desempenho do TCP e do *WebSocket* foram superiores. Para pacotes menores que 1024 *bytes*, o CoAP foi melhor que o MQTT. No entanto, o CoAP teve o pior desempenho em memória utilizada quando houveram retransmissões de pacotes. Como conclusão, para que o CoAP seja utilizado em dispositivos restritos, o tamanho do pacote deve ser menor que 1 KB. O MQTT, para o mesmo fim, deve ter até 1,5 KB e o MQTT-SN não está preparado para a aplicação em sistemas externos.

Em [Chaudhary et al. 2017], os autores compararam o desempenho do MQTT, CoAP e AMQP para o envio constante de dados em redes cabeadas, *WiFi* e 2/3/4G. Os autores mensuraram o *Overhead* e a vazão das mensagens. O *testbed* era composto por um servidor de alto desempenho sendo utilizado como *broker* do MQTT e do AMQP e como servidor do CoAP e um *Raspberry Pi-3*, utilizado como cliente MQTT, AMQP e CoAP. Além disto, um computador portátil foi utilizado como cliente. Os testes realizados faziam o envio sucessivo, com intervalos pequenos entre envios, de mensagens com tamanhos diferentes com base 10 (10,100,1000,100000). Os autores concluíram que em aplicações com alta taxa de mensagens, o MQTT QoS1 e 2 utiliza muita largura de banda por causa da retransmissão de pacotes. Já o CoAP consome muito pouco recurso dos dispositivos e trabalha de maneira estável em redes com alta taxa de perdas. Por fim, AMQP não está preparado para aplicações com componentes restritos pela grande memória utilizada pelo dispositivo [Chaudhary et al. 2017].

A necessidade de recursos de processamento no concentrador foi estudada em [Torres et al. 2016]. Os autores mostram que como o concentrador deve gerenciar filas de pacotes, informações sobre a sessão com o dispositivo, entre outras, há uma exigência de recurso dedicado para o mesmo.

Diferentemente dos trabalhos acima, este trabalho utiliza um dispositivo com processamento realmente restrito, com poucos *KBytes* de memória e baixo poder de processamento, para análise de desempenho, replicando um cenário mais real de IoT. Além disso, para caracterizar um cenário mais realístico, considerou-se a influência do tráfego de fundo no desempenho dos protocolos MQTT e CoAP.

4. Avaliação qualitativa dos protocolos de aplicação

A partir dos trabalhos relacionados discutidos anteriormente, esta seção apresenta uma análise qualitativa dos requisitos para as aplicações de Internet das Coisas. Inicialmente, as aplicações IoT foram classificadas em três domínios: residencial, comercial e industrial, que são descritos a seguir.

Residencial As aplicações residenciais se subdividem em aplicações implementadas por entusiastas e as desenvolvidas por empresas [Brush et al. 2011]. As aplicações implementadas por entusiastas utilizam dispositivos de prototipagem e sensores disponíveis no mercado, tais como ESP8266, *Arduínos*, *Raspberries*, etc. Por outro lado, empresas disponibilizam projetos proprietários e, muitas vezes integram suas soluções com sistemas de grandes empresas. Essas soluções geralmente visam sistemas de automatização de ambientes inteligentes, tais como aquecimento ou resfriamento de ambientes, monitoramento de segurança, iluminação inteligente, e automatização de tarefas em horários específicos do dia.

Comercial Neste domínio estão os frameworks comerciais de desenvolvimento de aplicações IoT. Estes frameworks dão suporte aos domínios residenciais e industriais e normalmente, oferecem serviços como armazenamento e processamento de dados [Derhamy et al. 2015]. Desta forma, há uma camada de abstração onde as empresas, residências e indústrias se conectam e utilizam serviços das principais companhias tecnológicas, como *Microsoft*, *Google*, *IBM* e *Intel*, além das dezenas de outras plataformas que disputam o mercado [Mineraud et al. 2016]. De fato, produtos como *Azure IoT* da Microsoft, *Amazon AWS* da Amazon, *Google Cloud IoT* da Google trabalham com mais de um protocolo para gerenciar os dispositivos na borda da rede.

Industrial O domínio industrial trata de sistemas de tempo real e alta confiabilidade. No geral, estes sistemas possuem uma grande quantidade de sensores que precisam comunicar entre si e com um concentrador. Portanto, a utilização da rede de uso comum pode ser facilmente prejudicada devido à grande quantidade de informação que é trocada por eles [Da Xu et al. 2014]. Exemplos de aplicações no domínio industrial são os sistemas de monitoramento de pacientes (*eHealth*), sistemas anti-incêndio, controle e rastreamento de cargas em tempo real, etc.

Para analisar quais protocolos de aplicação IoT atendem cada um dos domínios acima, mapeamos como cada protocolo atende a um conjunto de requisitos, descritos a seguir. As análises se cada protocolo atende ou não cada requisito são baseadas nos experimentos realizados em [Bandyopadhyay and Bhattacharyya 2013], [Chaudhary et al. 2017], [De Caro et al. 2013], [Mun et al. 2016] e [Naik 2017], além das especificações dos protocolos. A Tabela 1 sumariza os resultados obtidos.

- **Restrição severa de processamento:** Este requisito refere-se a dispositivos com restrição de memória e processamento. Consideramos como restrição severa, dispositivos com menos de 80KBytes de memória e 80Mhz de processamento. Os protocolos MQTT, CoAP e MQTT-SN, possuem cabeçalhos pequenos o que ajuda a diminuir a necessidade de *buffers* e de dados processados. O AMQP possui ferramentas adicionais de transmissão e gerenciamento de mensagens o que necessita maior processamento.
- **Restrição energia:** Sensores em locais de difícil acesso precisam ter energia por longos períodos de tempo sem que seja necessária a troca de fontes energéticas.
- **Qualidade de sinal da rede:** A qualidade da rede utilizada para comunicação entre dispositivos em que ele utiliza para comunicação. A escala de respostas pode ser de *Ruim* (com mais de 20% de perda), *Média* (entre 0 e 20% de perda) e *Boa* onde não há perda.

- **Possibilidade de Concentrador:** Essa característica define se a estrutura de rede do local onde os dispositivos estão instalados permite ou necessita de um concentrador de dados. As repostas possíveis são *Sim* e *Não* e estão ligadas diretamente a estrutura de comunicação apresentada pelo protocolo. Aqueles que utilizam o modelo *Publisher/Subscriber* naturalmente necessitam de um concentrador.
- **Recursos do Concentrador:** Essa característica define se o concentrador requer recursos de processamento e memória para gerenciar os dados recebidos. A escala de respostas possíveis trata como *Baixa* quando há a possibilidade de utilizar outro dispositivo com baixo poder de processamento, *Média* quando esse concentrador por ser um *Raspberry* ou equipamentos similares e *Alta* quando há a necessidade de mais processamento.
- **Capacidade de reconhecimento de outros dispositivos:** Essa característica define se, em uma comunicação M2M, o protocolo possui a capacidade de descobrir outros dispositivos, inclusive dispositivos concentradores [Villaverde et al. 2014]. As respostas representam se os protocolos possuem a característica de descoberta de outros dispositivos ativa.
- **Envio de dados em tempo real:** Define se um protocolo é recomendado para uso em situações em que o tempo real é importante. As repostas possíveis são *Sim* e *Não*.
- **Tamanho da carga de dados:** Essa característica mostra se o protocolo está preparado para o envio de mensagens com grandes quantidades de informação (alto *payload*). Esta característica pode ocorrer tanto no nível do protocolo de aplicação ou pode ser implementada a nível da camada de transporte.
- **Confiabilidade na entrega dos dados:** Essa característica define se o protocolo consegue uma entrega confiável de dados para uma aplicação que exija tal requisito. Os protocolos CoAP, MQTT-SN e AMQP possuem seus próprios sistemas de confiabilidade sobre as camadas de transporte TCP e UDP. O MQTT possui nos níveis de QoS 1 e 2, porém no nível mais comum de utilização, o QoS 0, o protocolo deixa esses controles para a camada de transporte.
- **Interoperabilidade:** Essa característica define a possibilidade de interação com plataformas já disponíveis por grandes empresas. O AMQP possui essa característica como fundamento.
- **Envio constante de informações:** Essa característica define a capacidade do protocolo de enviar mensagens de modo constante.

Considerando os requisitos apresentados acima, o MQTT apresenta maior estabilidade devido a utilização do TCP. Aplicações que necessitem troca de mensagens com grande quantidade de dados em redes locais também podem utilizar o MQTT. O MQTT também possui implementações disponíveis para diversos dispositivos, incluindo versões para dispositivos com baixo poder de processamento o que aumenta ainda mais as possibilidades de utilização. Além disto, as nuvens de grandes empresas, como *Amazon* e *Microsoft* fornecem suporte ao MQTT.

Por outro lado, o protocolo de aplicação CoAP apresenta vantagens em todos os domínios. A arquitetura cliente/servidor permite que os dispositivos economizem energia. Além disso, o uso do UDP torna o fluxo de dados menor. Devido a semelhança com o padrão HTTP, o protocolo CoAP tem grande adoção entre desenvolvedores Web.

Tabela 1. Comparação Entre os Protocolos Estudados

Características	MQTT	MQTT-SN	CoAP	AMQP
Restrição severa de processamento	Sim	Sim	Sim	Não
Restrição severa de energia	Não	Sim	Sim	Não
Qualidade do sinal da rede	Boa	Ruim	Ruim	Boa
Possibilidade de concentrador	Sim	Sim	Não	Sim
Capacidade de recursos do concentrador	Média	Média/Baixa	–	Alta/Média
Capacidade de reconhecimento de outros dispositivos	Não	Sim	Sim	Não
Envio em tempo real de dados	Não	Não	Sim	Não
Comunicação com dados muito grandes	Sim	Não	Não	Sim
Confiabilidade na entrega dos dados	Sim	Sim	Não	Sim
Interoperabilidade entre fabricantes	Não	Não	Não	Sim
Envio constante de informações	Sim	Sim	Sim	Sim

5. Análise quantitativa

O MQTT e o CoAP são protocolos com arquiteturas distintas e com ampla adoção da indústria. Por este motivo, este trabalho analisa o desempenho de ambos quando utilizado com dispositivos com recursos restritos de processamento e memória. Para cada protocolo, foram realizados dois experimentos, sendo que cada experimento foi repetido com e sem tráfego de fundo na rede.

Monitoramento em tempo (quase) real: O intervalo mínimo entre mensagens no ESP8266 é de 100ms. Por isto, avaliou-se o monitoramento em tempo quase real com envio pacotes de 30 *bytes* com intervalo entre 100 e 900 milissegundos. Os testes duraram 10 minutos. O conteúdo das mensagens era composto por um identificador do dispositivo, um identificador da mensagem e por um *timestamp*.

Varição no tamanho da mensagem: foram enviados pacotes de tamanho 10, 100, 500 e 1000 *bytes* em intervalos de 1, 5, 10, 30 e 60 segundos durante 1 hora.

A topologia do *testbed* pode ser vista na Figura 1. O ESP8266, representando dispositivos IoT, possui um processador 32-bit RISC com 80 MHz de *clock* e 80KBytes de memória RAM, sendo que aplicações podem utilizar apenas cerca de 50 Kbytes. Foram utilizadas as bibliotecas [Kovalenko 2018] para o MQTT e [Poornima Nagesh 2018] para o CoAP¹. O servidor/concentrador captura as informações da rede com o *Wireshark*.

¹As implementações estão disponíveis em <https://gitlab.com/vmota/mqtt-coap-esp>

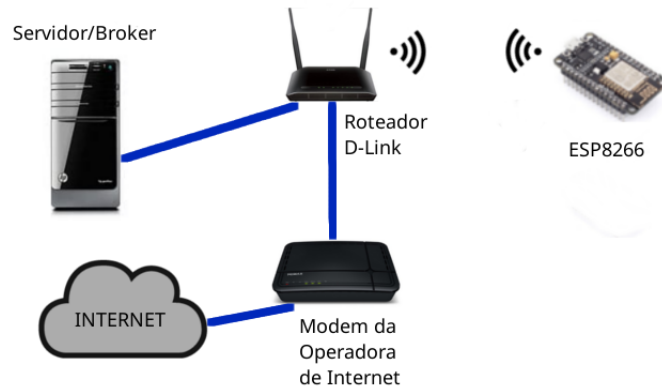


Figura 1. Topologia Utilizada nos Testes

No servidor, o concentrador MQTT *Mosquito Broker* e um servidor CoAP estavam instalados. Para os testes sem tráfego de fundo, isolava-se a rede e apenas o ESP8266 e o servidor estavam conectados a rede sem fio. Já no teste com tráfego de fundo, permitia-se a utilização do roteador sem fio por outros usuários. Foram mensurados quatro métricas:

Taxa de entrega: Quantidade de mensagens recebidas pelo servidor por segundo.

Tempo médio entre recebimento de pacotes: Mostra a média do tempo entre captura dos pacotes no *Broker* e foi utilizada para o MQTT. A média é calculada pela equação:

$$T_{dm} = \frac{\sum_{i=0}^{N-1} t_{i+1} + t_i}{N_{PUB}}$$

sendo, t_i o tempo de recebimento da mensagem i , t_{i+1} o tempo da próxima mensagem e N_{PUB} o número de mensagens publicadas.

Tempo médio de transmissão: Trata do tempo médio entre o envio e o recebimento da mensagem no protocolo CoAP, ou seja, o *Round Time Trip* (RTT). O tempo médio é dado pela equação:

$$T_m = \frac{\sum_{i=0}^{N-1} \frac{t_{ACK} - t_{GET}}{2}}{N_{ACK}}$$

sendo, t_{ACK} o tempo em que a mensagem de confirmação de recebimento foi capturada, t_{GET} o tempo em que a mensagem de requisição foi enviada e N_{ACK} o número de mensagens de confirmação.

Número de retransmissão dos pacotes: número de pacotes que foram retransmitidos até que a mensagem fosse recebida pelo computador.

6. Resultados e Análises

6.1. Monitoramento em tempo (quase) real

Consideramos como monitoramento em tempo quase real aplicações que enviem mensagens com uma taxa de envio menor que um segundo. A primeira métrica que analisamos foi a taxa de entrega.

Para calcular a taxa de entrega, o dispositivo enviava mensagens entre intervalos de 100, 400 e 700 milissegundos. Cada mensagem recebe um identificador inteiro sequencial, o servidor verifica se está recebendo as mensagens na sequência. Em um cenário ideal, o servidor irá receber 10 pacotes por segundo quando o envio tem intervalos for de 100 ms, 2 ou 3 pacotes para intervalos de 400ms e 1 ou 2 pacotes para intervalos de 700ms. Ressalta-se que devido as limitações de hardware do ESP8266, não foi possível enviar mensagens em intervalos inferiores a 100ms.

A Figura 2 apresenta a função massa de probabilidade (*probability mass function* - PMF) dos pacotes recebidos pelo servidor/concentrador a cada segundo em uma rede sem tráfego de fundo.

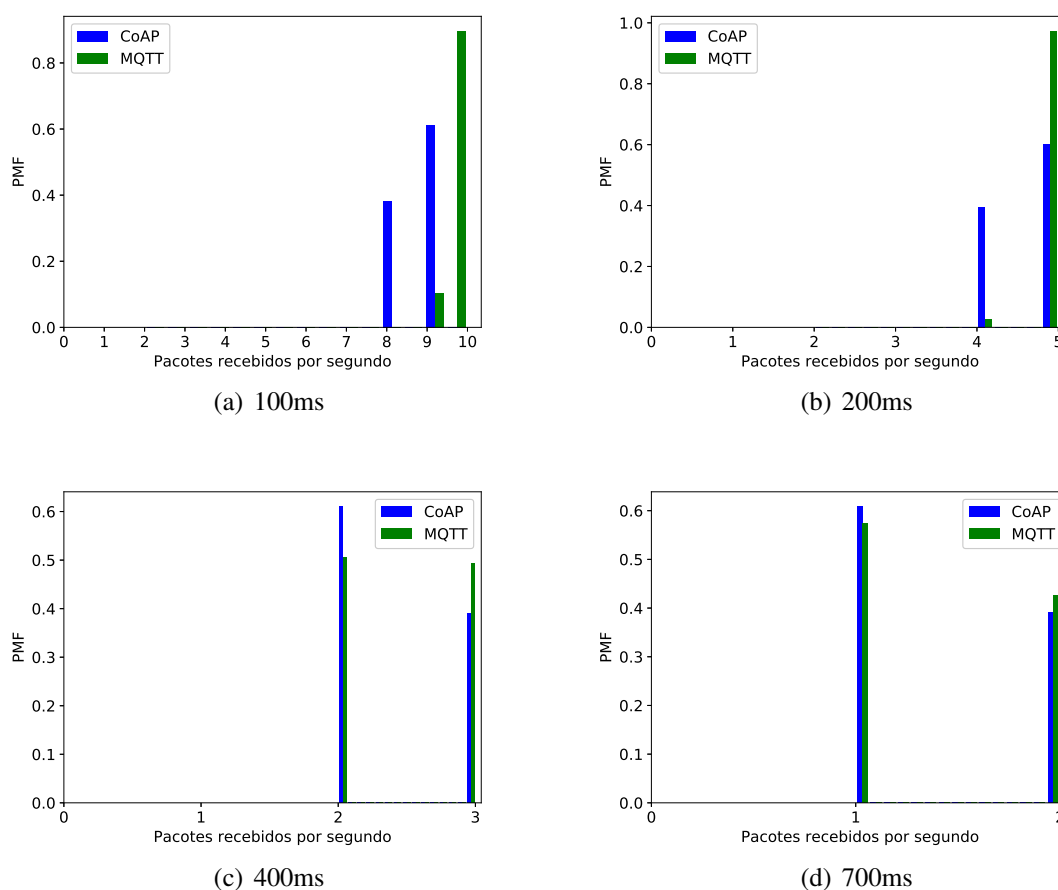


Figura 2. Função massa de probabilidade (PMF) de pacotes recebidos por segundo nos experimentos sem tráfego de fundo dos protocolos CoAP e do MQTT, com envio de mensagens entre 100 e 700 milissegundos.

Observa-se que no cenário de intervalo entre mensagens de 100ms, Figura 2(a), o CoAP não conseguiu entregar 10 mensagens por segundo ao servidor. Como o CoAP utiliza o UDP, não exige confiabilidade na entrega, os pacotes esperam no *buffer* e/ou podem ser descartados.

Para o protocolo MQTT, quanto menor o intervalo entre envios de mensagens, maior a chance de o pacote enviado pelo dispositivo não chegar ao *Broker*. Mesmo utili-

zando o TCP na camada de transporte, observou-se que não há recebimento de mensagens em alguns instantes durante a captura de pacotes.

A explicação para o não recebimento da totalidade de mensagens pelo servidor refere-se ao algoritmo de Nagle, implementado por padrão no *lwIP*. O algoritmo de Nagle pode interromper, por um período de tempo, o envio de mensagens quando uma destas não recebe um ACK. Assim, todas as outras mensagens que poderiam ser entregues permanecem no *buffer* até que ele receba um ACK ou até que o tamanho combinado no *handshake* do TCP seja atingido.

Nos experimentos, o efeito do algoritmo pode ser percebido no *Wireshark* com o recebimento de uma mensagem de erro "*TCP Spurious Retransmission*", seguida de um pacote encapsulando mais de uma mensagem do MQTT.

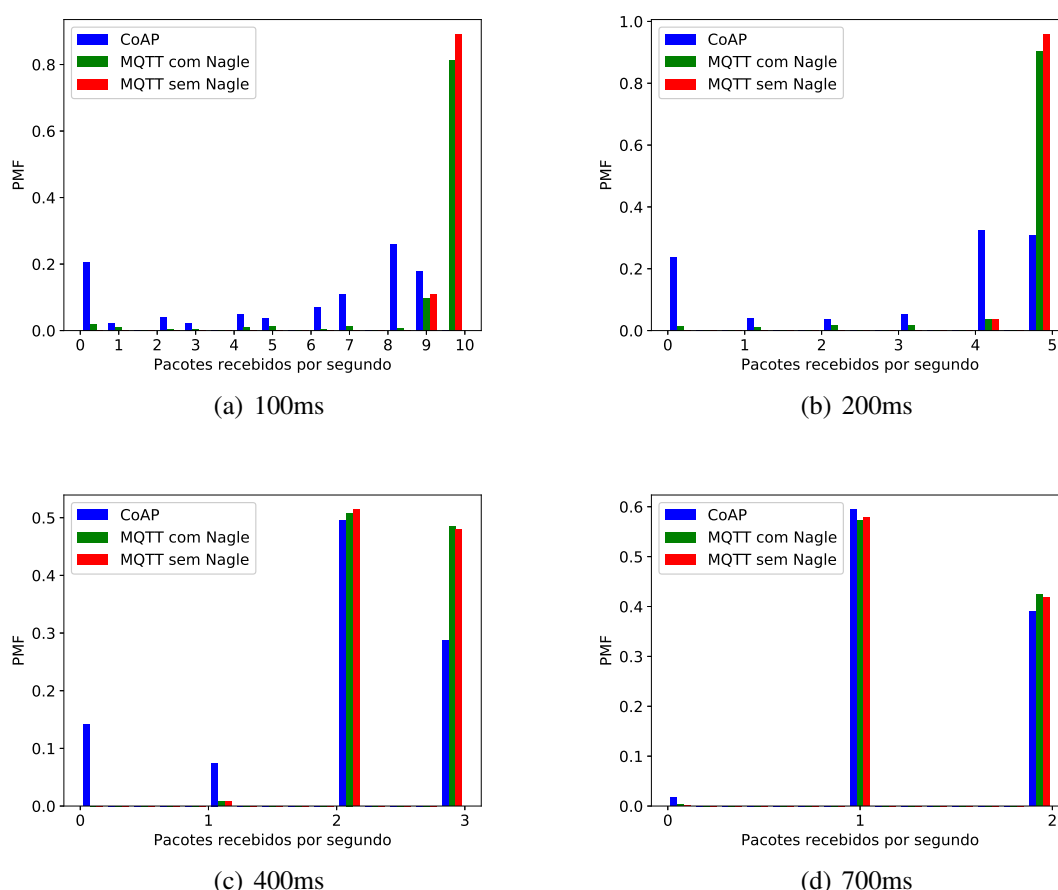


Figura 3. Função massa de probabilidade (PMF) de pacotes recebidos por segundo nos experimentos com tráfego de fundo dos protocolos CoAP, MQTT e MQTT sem Nagle, com envio de mensagens entre 100 e 700 milissegundos.

A Figura 3 apresenta a PMF do recebimento de pacotes quando existe um tráfego de fundo. O tráfego de fundo foi gerado por outros dispositivos conectados ao mesmo ponto de acesso para navegação web, vídeos, músicas, etc. Além disso, neste caso também analisamos os efeitos de habilitar e desabilitar o algoritmo Nagle. O desempenho do MQTT com o algoritmo Nagle desabilitado se assemelha ao desempenho do

MQTT sem tráfego de fundo. Isso acontece por que não há a retenção das mensagens e o fluxo de dados consegue seguir mesmo com perda de mensagens.

Comparando os dois protocolos, observa-se que o MQTT consegue superar o CoAP para intervalos menores que 500 milissegundos. Depois desse valor, até o limite de novecentos milissegundos, a taxa de entrega fica muito próxima.

Em relação ao *número de mensagens retransmitidas*, no MQTT calcula-se o número de mensagens "*TCP Spurious Retransmission*", causadas pelo controle do algoritmo de Nagle. As retransmissões no CoAP acontecem quando o cliente não recebe resposta e faz uma nova requisição.

A Figura 4(a) apresenta a taxa de retransmissão (mensagens retransmitidas por total de mensagens) do CoAP e do MQTT. Observa-se que o protocolo MQTT possui um número menor de retransmissão para intervalos de envio menores que 500ms, que era de se esperar pela confiabilidade do TCP em relação ao UDP. Ao aumentar o intervalo entre envios o CoAP diminui o número de erros, porém o MQTT passa a ser influenciado pelo algoritmo de Nagle, implementado internamente no TCP.

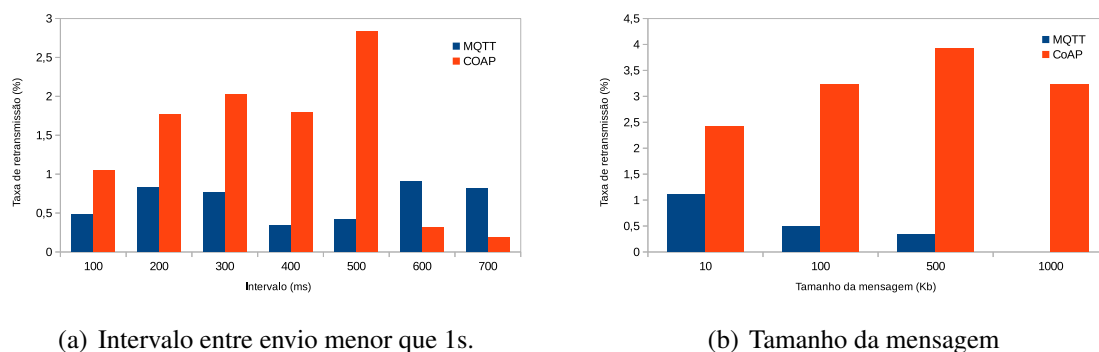


Figura 4. Retransmissões nos protocolos CoAp e MQTT

6.2. Tamanho da mensagem - Payload

O tamanho da carga de dados em cada protocolo foi variado entre {10, 100, 500, 1000} bytes. A biblioteca utilizada para o MQTT conseguiu ser mais estável e enviar mensagens de até 1500 bytes sem que utilizasse toda a memória RAM do ESP. Já para o CoAP, após a alteração da biblioteca, foi possível enviar até 1024 bytes antes que o mesmo efeito acontecesse. De fato, como mostra a Figura 4(b), o número de retransmissões no MQTT diminuiu com o crescimento do tamanho do *payload*. Justificado pelo gerenciamento de *buffer* que ocorre no MQTT.

Tempo médio de transmissão em relação ao tamanho da mensagem para o protocolo CoAP é mostrado na Figura 5. CoAP se mostrou vulnerável ao tráfego de fundo sendo que situações de retransmissão de pacotes com mais de três tentativas aconteceram mais de uma vez o que aumentou consideravelmente o tempo da média.

Para o MQTT, foi calculado o tempo médio de recepção entre mensagens, visto que não é possível inferir o tempo de envio da mensagem. De modo geral, o tamanho do *payload* não interfere na geração e envio de mensagens, como mostra a Figura 6.

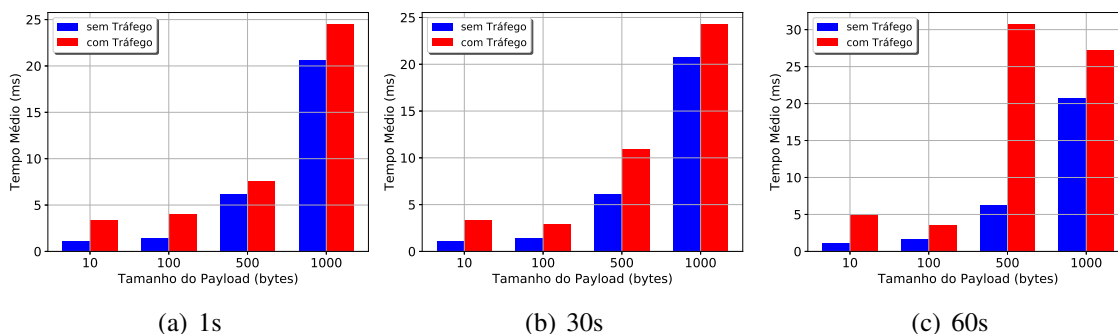


Figura 5. Tempo médio (RTT) requisições-resposta do CoAP com e sem tráfego de fundo, com envio de mensagens entre 1 e 60 segundos

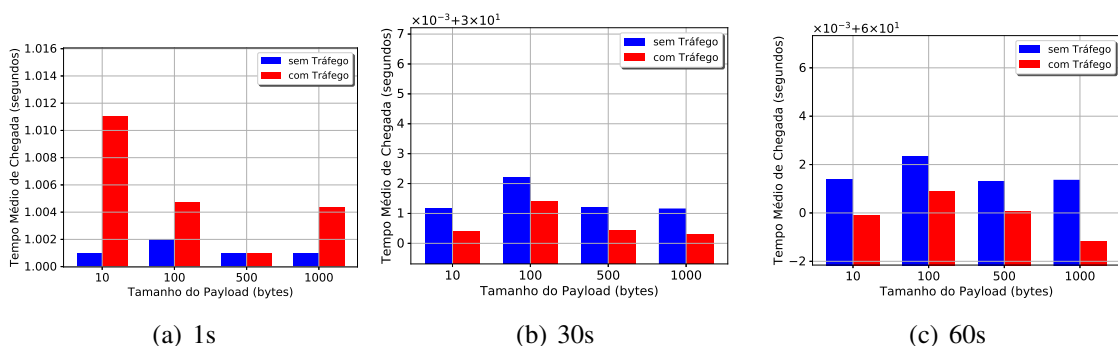


Figura 6. Tempo entre recebimento de pacotes do MQTT com e sem Tráfego de fundo e envio de mensagens entre 1 e 60 segundos.

Entretanto, é possível notar que a diferença média de tempos sem tráfego de rede é maior que a média quando há tráfego de rede, excluindo 6a.

A Figura 6(a) mostra que no momento da captura aconteceram algumas perdas de pacotes na camada TCP e foram feitas algumas retransmissões. Ocorreu também o efeito do algoritmo de Nagle em alguns momentos e o ESP8266 parou de enviar pacotes por uns três a oito segundos influenciando na média dos resultados.

Embora uma comparação direta entre os protocolos não pode ser devida as métricas serem calculadas de forma distinta, nota-se que os dois tiveram problemas ao utilizar uma rede com tráfego de fundo. Observa-se que o número de mensagens retransmitidas pelo CoAP foi maior para todos os tamanhos de carga de dados na mensagem, Figura 5. O CoAP teve um aparente aumento do tempo de resposta com o aumento do *payload*. O MQTT teve um desempenho mais uniforme comparado com os outros tamanhos. Vale ressaltar que o TCP estava fragmentando os pacotes com tamanho máximo de 536 bytes que é implementado como padrão no *lwIP* disponível no ESP8266. Já o CoAP, estava mandando o pacote inteiro como característica do UDP, tendo como máximo valor 1024 bytes devido à limitação do dispositivo.

7. Conclusão

Este trabalho analisou qualitativamente e quantitativamente os principais protocolos da camada de aplicação para Internet das Coisas. Qualitativamente, observou-se que o

MQTT, por sua arquitetura *publish/subscribe* foi adotado por diversas plataformas comerciais de armazenamento de dados IoT. A versão MQTT-SN provê maior versatilidade devido a utilização de UDP no lugar do TCP como camada de transporte. O CoAP é um dos mais versáteis, tendo vantagens em todos os cenários pelo seu modelo de comunicação e compatibilidade com HTTP. Já o AMQP apresenta melhores mecanismos de controle de fluxo e erro, acrescentando confiabilidade e melhorando a interoperabilidade.

Para realizar os testes quantitativos, foi utilizado o SoC ESP8266 e as bibliotecas públicas do CoAP e MQTT. Diferentemente de outras comparações realizadas na literatura, estes dispositivos não suportam o envio de mensagens com payload maiores que 1024 *bytes* e nem o envios de mensagens com intervalos menores do que 100ms. Além disso, a implementação padrão do TCP sobre o lwIP, utilizado nestes dispositivos, habilita o algoritmo Nagle por padrão, para reduzir o número de segmentos muito pequenos enviados, e assim, economizar energia. Isso afeta o desempenho do MQTT para funcionar como um protocolo de tempo-real. Considerando que as bibliotecas públicas continham bugs e/ou incompatibilidades, este trabalho tornou público para desenvolvedores as implementações dos protocolos de aplicação para IoT, CoAP e MQTT.

Quantitativamente, observou-se que o protocolo CoAP foi mais sensível ao aumento do *payload* do que o MQTT nos testes com tráfego de fundo na rede, principalmente com o algoritmo de Nagle desabilitado. Portanto, o MQTT mostrou-se mais estável que o CoAP ao utilizar um dispositivo restrito. Além disso, o tráfego de fundo causou retransmissão de mensagens em ambos os protocolos. Como o CoAP utiliza o UDP na camada de transporte, o CoAP implementa seu próprio mecanismo de controle de erros. Contudo, esse mecanismo pode gerar problemas no envio constante de mensagens, pois o modelo de retransmissão do CoAP pode fazer com que o mesmo aguarde o recebimento de confirmação antes de enviar novas mensagens.

Por fim, espera-se que este trabalho possa auxiliar desenvolvedores a escolherem qual o melhor protocolo da camada de aplicação baseado nos requisitos apresentados. Como trabalhos futuros, pretende-se implementar as especificações dos protocolos MQTT-SN e AMQP para dispositivos de processamento e memória restritos.

Referências

- 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.
- Appel, S., Sachs, K., and Buchmann, A. (2010). Towards benchmarking of AMQP. *Proceedings of the 4th ACM International Conference on Distributed Event-Based Systems, DEBS 2010*, (October 2016):99–100.
- Atzori, L., Iera, A., and Morabito, G. (2010). The internet of things: A survey. *Computer networks*, 54(15):2787–2805.
- Bandyopadhyay, S. and Bhattacharyya, A. (2013). Lightweight internet protocols for web enablement of sensors using constrained gateway devices. In *Computing, Networking and Communications (ICNC), 2013 International Conference on*, pages 334–340. IEEE.

- Brush, A., Lee, B., Mahajan, R., Agarwal, S., Saroiu, S., and Dixon, C. (2011). Home automation in the wild: challenges and opportunities. In *proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2115–2124. ACM.
- Chaudhary, A., Peddoju, S. K., and Kadarla, K. (2017). Study of internet-of-things messaging protocols used for exchanging data with external sources. pages 666–671.
- Da Xu, L., He, W., and Li, S. (2014). Internet of things in industries: A survey. *IEEE Transactions on industrial informatics*, 10(4):2233–2243.
- De Caro, N., Colitti, W., Steenhaut, K., Mangino, G., and Reali, G. (2013). Comparison of two lightweight protocols for smartphone-based sensing. In *Communications and Vehicular Technology in the Benelux (SCVT), 2013 IEEE 20th Symposium on*, pages 1–6. IEEE.
- Derhamy, H., Eliasson, J., Delsing, J., and Priller, P. (2015). A survey of commercial frameworks for the Internet of Things. *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, 2015-October*.
- Kovalenko, O. (2018). Arduinomqtt: Mqtt client library for arduino based on the eclipse paho project.
- Mazzer, D., Frigieri, E., and Parreira, L. (2015). Protocolos M2M para Ambientes Limitados no Contexto do IoT: Uma Comparação de Abordagens. *Inatel.Br*.
- Mineraud, J., Mazhelis, O., Su, X., and Tarkoma, S. (2016). A gap analysis of internet-of-things platforms. *Computer Communications*, 89:5–16.
- Mun, D.-H., Le Dinh, M., and Kwon, Y.-W. (2016). An assessment of internet of things protocols for resource-constrained applications. 1:555–560.
- Naik, N. (2017). Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP. *2017 IEEE International Symposium on Systems Engineering, ISSE 2017 - Proceedings*.
- Poornima Nagesh, L. P. (2018). Esp-coap: This is a arduino library for the esp8266 12e.
- Shelby, Z., Hartke, K., and Bormann, C. (2014). The constrained application protocol (coap). RFC7252.
- Torres, A. B., Rocha, A. R., and de Souza, J. N. (2016). Análise de desempenho de brokers mqtt em sistema de baixo custo. In *Anais do XXXVI congresso da sociedade brasileira de computação. Sociedade Brasileira de Computação*.
- Truong, A. S.-C. and Linh, H. (2013). MQTT For Sensor Networks (MQTT-SN) Protocol Specification. *International Business Machines Corporation (IBM, 1:28*.
- Villaverde, B. C., de Paz Alberola, R., Jara, A. J., Fedor, S., Das, S. K., and Pesch, D. (2014). Service discovery protocols for constrained machine-to-machine communications. *IEEE Communications Surveys and Tutorials*, 16(1):41–60.
- Zanella, A., Bui, N., Castellani, A., Vangelista, L., and Zorzi, M. (2014). Internet of things for smart cities. *IEEE Internet of Things journal*, 1(1):22–32.

Caracterização de Desempenho de uma Rede LoRa em Ambientes Urbanos: Simulação vs. Prática

Fernando M. Ortiz¹, Thales T. Almeida^{1,2},
Ana Elisa Ferreira^{1,3} e Luís Henrique M. K. Costa¹

¹Universidade Federal do Rio de Janeiro - PEE-COPPE/GTA

²Centro Federal de Educação Tecnológica de Minas Gerais - CEFET/MG

³Centro Federal de Educação Tecnológica Celso Suckow da Fonseca - CEFET/RJ

{fmolano, almeida, ferreira, luish}@gta.ufrj.br

Abstract. *LoRa, a low power wide area network technology, enables sensing in Smart Cities increasing coverage and reducing network infrastructure. Nevertheless, works about LoRa lacks simulation evaluations, being limited to experiments with a reduced number of physical devices. Because software developments of the LoRaWAN architecture are emerging, and the necessity to verify the equivalence between practical and simulated experiments, this work compares the results of experimentation using LoRa physical devices and simulations using NS-3 software. The results show that there is strong correlation between the metrics evaluated in each environment. Simulations using an adapted stochastic propagation model demonstrate that are similar to real experiments, with equivalent results for performance tests using the LoRa setup parameters.*

Resumo. *A LoRa, uma tecnologia de baixa potência e longo alcance, possibilita o sensoriamento em Cidades Inteligentes aumentando a cobertura, e reduzindo a infraestrutura da rede. No entanto, a literatura sobre a LoRa carece de avaliações de simulação, estando limitada à experimentação de um número reduzido de dispositivos físicos. Devido ao surgimento de implementações em software da arquitetura LoRaWAN, é necessário verificar a equivalência entre experimentos práticos e simulados. Com foco neste objetivo, este trabalho compara os resultados da experimentação usando dispositivos físicos LoRa e simulações usando o software NS-3. Os resultados mostram que existe uma forte correlação entre as métricas avaliadas em cada um dos ambientes. Simulações usando um modelo de propagação estocástico adaptado mostram que elas são semelhantes aos experimentos reais, com resultados equivalentes para todos os testes de desempenho, usando os parâmetros de configuração da LoRa.*

1. Introdução

Dispositivos com capacidade de se conectar e se integrar têm permitido que múltiplas “coisas” tenham a capacidade de se comunicar através da Internet. A partir desse paradigma, tais dispositivos têm se tornado inteligentes, capazes de capturar, processar e transmitir dados. Estes dispositivos têm a capacidade de serem autônomos, permitindo, inclusive, que o enlace de comunicação entre eles seja gerenciado. Por isso, a quantidade de objetos inteligentes transmitindo informações na Internet é estimada em

50 bilhões de dispositivos até 2020 [Evans, 2011]. Este enorme crescimento leva à necessidade de desenvolver novas tecnologias. As redes de baixa potência e longo alcance (*Low Power Wide Area Network* – LPWAN) como, por exemplo, a tecnologia LoRa (*Long Range*), são uma opção para atender a esta necessidade. Cidades inteligentes, *smart grid* e sensoriamento urbano são aplicações que ilustram bem este cenário com a comunicação de pontos distantes e que requerem baixo consumo de energia.

A LoRa é uma tecnologia de comunicação sem-fio, orientada a complementar as redes de telefonia móvel legadas e as tecnologias sem-fio de curto alcance. É indicada, principalmente, para aplicações de Internet das Coisas (*Internet of Things* – IoT) que requeiram cobertura de grandes áreas [Augustin et al., 2016, Raza et al., 2017]. Para isso, a LoRa implementa uma técnica de modulação por espalhamento espectral de *chirps* (*Chirp Spreading Spectrum* – CSS) [Semtech Corp., 2015], que varia a frequência sem mudar a fase entre símbolos adjacentes [Goursaud e Gorce, 2015], fazendo com que o sinal resultante seja resistente a interferências por ruído ou sinais com frequências próximas. A LoRa também se caracteriza pela redução na complexidade do *hardware*, a diminuição do tamanho dos cabeçalhos e da complexidade da rede em termos de saltos e endereçamento, viabilizando uma comunicação bidirecional simples, com o uso de uma infraestrutura mínima e com baixo consumo de energia. Entretanto, o planejamento de redes LPWAN torna-se complexo pelos múltiplos aspectos da sua implementação, que podem ser avaliadas por meio de simulações. Uma vez que as redes LPWAN utilizam radiofrequência como meio de transmissão, a variação das condições no meio de propagação, as limitações em termos de distância e interferência, além da utilização de um meio de comunicação compartilhado, fazem da simulação de redes LPWAN um desafio importante para avaliar o comportamento dessas e outras variáveis no ambiente simulado. Embora existam outras tecnologias LPWAN [Raza et al., 2017], este documento foca na tecnologia LoRa, por ser uma das mais difundidas atualmente.

Neste trabalho, é investigada a equivalência entre os resultados de simulações executadas no simulador de redes NS-3 e experimentos práticos que envolvem a comunicação de dispositivos reais compatíveis com a tecnologia LoRa. Para isso, é feita uma avaliação de desempenho da comunicação em enlaces compostos por uma unidade transmissora e uma unidade receptora, baseado na metodologia usada em [Ortiz et al., 2018], variando a distância e o fator de espalhamento (*Spreading Factor* – SF). São também implementados protótipos de *hardware* para a comunicação LoRa, onde as mesmas condições de simulação são recriadas. São avaliadas as seguintes métricas: capacidade da rede em termos de vazão, taxa de perda de mensagens, potência do sinal recebido e relação entre vazão teórica e a vazão obtida através dos experimentos. Para as simulações, utilizou-se uma biblioteca para redes LoRaWAN [Magrin et al., 2017], desenvolvida para o simulador de eventos discretos NS-3 (Network Simulator version 3) [Carneiro, 2010, Henderson et al., 2008].

Uma vez que as atuais tecnologias LPWAN (LoRa e SigFox) na camada física são proprietárias, poucas soluções de simulação têm sido desenvolvidas para modelar seu comportamento em diferentes ambientes com linha de visada (*Line-of-Sight* – LoS) ou sem linha de visada (*Non-Line-of-Sight* – NLoS). Assim, vários autores têm desenvolvido experimentos práticos avaliando as suas diversas características (especialmente da LoRa) em vez de simulação. Apesar deste crescente desenvolvimento de soluções práticas im-

plementando a tecnologia LoRa, as variações nas condições do meio de propagação, o curto tempo de contato entre os nós, os problemas de atenuação e os fenômenos de multipercurso tornam estas implementações muito específicas e fazem necessária a avaliação desses comportamentos por meio de simulações que possam ser reproduzidas facilmente e em situações previamente determinadas. Além disso, experimentos de grande porte são mais difíceis de serem realizados no cenário real em comparação aos projetados em ambiente simulado. O tratamento dos dados gerados por simulação possibilita ainda uma melhor compreensão do desempenho de um enlace antes de ser implementado na prática e a identificação de um comportamento anômalo, antes da entrada em operação.

Este trabalho está organizado da seguinte forma. A Seção 2 discute os trabalhos relacionados com experimentos práticos e simulações com a tecnologia LoRa. A Seção 3 descreve a tecnologia LoRa, mostrando a camada física (LoRa PHY). Já a Seção 4 apresenta a metodologia utilizada neste trabalho, assim como os protótipos de *hardware* e *software* implementados para os experimentos. A Seção 5 apresenta os experimentos e resultados obtidos. Por fim, a Seção 6 conclui o trabalho e identifica desafios futuros.

2. Trabalhos Relacionados

Poucos trabalhos têm sido desenvolvidos com o intuito de caracterizar e avaliar o comportamento das redes LPWAN em um ambiente simulado. Bor *et al.* [Bor et al., 2016b] desenvolveram uma ferramenta de simulação de eventos discretos chamada *LoRaSim* usando o arcabouço *SimPy*. Esta ferramenta permite alocar N nós e M gateways em um espaço bidimensional. Cada nó é definido por parâmetros de transmissão específicos da tecnologia LoRa. Assim, a ferramenta *LoRaSim* permite avaliar a quantidade máxima de dispositivos terminais em um enlace LoRa. Para estabelecer os parâmetros de alcance e captura de pacotes, os autores desenvolveram um modelo experimental prático. O alcance da comunicação foi determinado em função do SF e da largura de banda (*Bandwidth* – BW), e a captura de pacotes em função do tempo de transmissão e da potência do sinal. Os autores concluem que múltiplos sumidouros, bem como a configuração dinâmica dos dispositivos, permitem melhorar a escalabilidade na rede. Utilizando o trabalho de [Bor et al., 2016b], Centenaro *et al.* [Centenaro et al., 2016] investigam os efeitos da interferência em um gateway LoRa. Para isso, empregando a ferramenta *LoRaSim*, avaliam duas condições de interrupção em um enlace, uma baseada na relação sinal-ruído e a outra baseada na interferência entre sequências iguais do SF. Partindo dessas condições, os autores concluem que os enlaces são altamente sensíveis às colisões, o que reduz consideravelmente a escalabilidade da rede.

Reynders *et al.* [Reynders et al., 2016] avaliam as diferenças técnicas entre duas tecnologias sem-fio para redes LPWAN: LoRa e SigFox. Diferenciadas pelo tipo de modulação que usam, LoRa - espalhamento espectral por *chirp*, e SigFox - banda ultra estreita (*Ultra Narrow Band* – UNB), ambas as tecnologias podem operar na mesma frequência. Os autores avaliaram a coexistência das duas tecnologias na camada física. Para isso, implementam uma simulação em MATLAB para avaliar a taxa de erro de bit nas técnicas de modulação, CSS e BPSK (usada pela UNB). Já para avaliar a camada de controle de acesso ao meio, os autores comparam o desempenho baseado em interferência e coexistência. Para isso, Reynders *et al.* desenvolveram ambientes de simulação no simulador de eventos discretos NS-3. Para avaliar a interferência, foi projetado um cenário com uma área de 1000 m \times 1000 m com um nó central. Foram simulados cenários avali-

ando a interferência entre duas redes com modulação CSS, e um outro entre UNB e CSS. Os autores concluem que o UNB tem melhor desempenho que o CSS para grandes quantidades de nós a longa distância. Porém, o CSS oferece melhor desempenho em relação à vazão, e a variação do SF permite atingir maiores distâncias.

Magrin *et al.* [Magrin et al., 2017] avaliaram o desempenho de redes LoRa em um cenário de cidades inteligentes. Os autores propuseram uma medição para avaliar o desempenho de um enlace, implementando uma biblioteca LoRaWAN no NS-3. Os autores mostraram que o LoRaWAN fornece uma taxa de transferência mais alta do que o esquema básico do ALOHA, e que as redes LoRaWAN podem ser dimensionadas à medida que o número de *gateways* aumenta. Ao mesmo tempo, estudam o impacto do tráfego de *downlink*. Os autores concluem confirmando que os resultados de dimensionamento da rede são devidos à ortogonalidade entre os SFs. Também concluem que a arquitetura é facilmente escalável, ainda mais quando o número de *gateways* aumenta, melhorando a cobertura e a confiabilidade no enlace de *uplink*. Já Abeele *et al.* [Van den Abeele et al., 2017] implementaram uma biblioteca LoRaWAN no NS-3 que inclui um modelo de erro usado para determinar o alcance, assim como a interferência entre múltiplas transmissões simultâneas. Além disso, o módulo também suporta comunicação bidirecional. Os autores concluem que a configuração dinâmica de parâmetros nos dispositivos terminais é determinante para o desempenho da rede. Também concluem que o fluxo limitado do enlace de *downlink* degrada consideravelmente a taxa de entrega de pacotes, e que aumentando o número de *gateways* o problema pode ser mitigado. As camadas físicas LoRa PHY implementada tanto em [Magrin et al., 2017] quanto em [Van den Abeele et al., 2017] suportam todos os SFs e todas as taxas de código indicadas na especificação.

Petäjärvi *et al.* [Petäjärvi et al., 2015] avaliam uma tecnologia de baixo consumo de energia e longo alcance, implementando a tecnologia LoRa. Para isso, os autores fazem experimentos práticos para avaliar a cobertura em diferentes ambientes: (i) um ambiente aquático, no qual instalaram um dispositivo terminal transmitindo a partir de um barco no mar; e (ii) um ambiente terrestre, na cidade de Oulu, Finlândia. Os autores avaliaram a taxa de perda de pacotes e a intensidade da potência do sinal recebido para as distâncias avaliadas.

Mikhaylov *et al.* [Mikhaylov et al., 2016], baseados nos resultados de [Petäjärvi et al., 2015], fazem uma análise do desempenho do LoRaWAN avaliando o rendimento em termos de taxa de transmissão, capacidade do canal para um número determinado de dispositivos terminais, e alcance entre os dispositivos terminais e os *gateways*. Os autores concluíram que o LoRaWAN, teoricamente, pode possuir milhões de nós conectados e enviando poucos bits, mas que só uma pequena porção desses dispositivos podem se localizar longe do *gateway*. Esta limitação é devida às interferências que podem afetar os enlaces (estruturas, outros sinais). Também concluíram que devido à limitação de ciclo de trabalho (*duty cycle*), o LoRaWAN pode reduzir a latência e o consumo de energia, apesar de ser altamente sensível a colisões de pacotes por conta do método de acesso ao meio (ALOHA).

Adelantado *et al.* [Adelantado et al., 2017] apresentam uma publicação na qual são avaliados os limites do LoRaWAN, com o objetivo de mostrar as limitações da tecnologia. Para isso, os autores discutem as capacidades e limitações da tecnologia no

contexto de casos de uso. Baseado nas limitações do ciclo de trabalho, tempos de transmissão, tamanho do cabeçalho MAC (*Media Access Control*) e número de pacotes recebidos por hora, Adelantado *et al.* mostram que o número de pacotes recebidos por hora diminui conforme aumentam o número de nós, independentemente do tamanho do cabeçalho MAC ou da vazão dos dados úteis (*goodput*). Assim, os autores descrevem as aplicações que podem ser consideradas para a implementação de uma rede LoRa. Já Ortiz *et al.* [Ortiz et al., 2018] apresentam uma análise da tecnologia LoRa através de experimentos práticos, avaliando a capacidade da rede em termos de vazão, a taxa de perda de mensagens, a potência do sinal do enlace e a relação entre vazão teórica e a vazão obtida através dos experimentos. A análise dos experimentos foi feita em função da variação do fator de espalhamento. Assim, foi avaliada a eficácia do enlace para cada fator de espalhamento em um cenário com uma distância máxima de 2 km. Os autores concluíram que o fator de espalhamento aumenta a sensibilidade no receptor, ao custo de menores taxas de transmissão.

O presente trabalho se diferencia dos descritos acima ao investigar a equivalência entre os resultados de simulações e experimentos práticos. Em vez de desenvolver uma nova forma de fazer simulação ou investigar detalhes da tecnologia, busca-se fornecer ao desenvolvedor de soluções que empreguem a comunicação via LoRa o conhecimento do quanto a simulação realmente espelha a realidade. Esta mesma informação pode ser usada para avaliar a simulação em si, contribuindo para o seu aperfeiçoamento e auxiliando pesquisadores a decidir pela sua utilização ou não. A seguir, é descrita brevemente a tecnologia LoRa e algumas das suas características principais.

3. Tecnologia LoRa

A LoRa é uma especificação de camada física (PHY) proprietária da Semtech [LoRa™-Alliance, 2016], desenhada para redes de longo alcance e baixa potência. Assim, esta tecnologia está projetada para permitir a conectividade de objetos inteligentes em distâncias da ordem dos quilômetros, com baixo consumo de energia, requerimento essencial das redes da Internet das Coisas.

A camada física (PHY) da tecnologia LoRa modula sinais em sub-bandas de rádio da faixa de frequências não licenciadas ISM (*Industrial, Scientific and Medical*) na ordem dos MHz. Para o Brasil, de acordo com a Agência Nacional de Telecomunicações (ANATEL), a faixa de frequências regulamentada para ISM está entre 915 e 928 MHz (AU915-928MHz) [ANATEL, 2018]. A frequência de 433 MHz (frequência de radiomador) pode ser implementada para redes com poucos dispositivos [ANATEL, 2016]. O LoRa PHY usa correção antecipada de erros (*Forward Error Correction – FEC*) e uma técnica de modulação de espalhamento espectral proprietária, que é uma variante do espalhamento espectral por *chirp*, que modula pulsos de *chirps* em frequência, a fim de codificar a informação [Bor et al., 2016a, LoRa™-Alliance, 2016].

4. Metodologia dos Experimentos

A metodologia adaptada para os experimentos define os critérios usados para configurar o modelo prático e o modelo simulado. A seguir, são expostos os parâmetros usados para avaliar as métricas propostas no artigo.

4.1. Parâmetros de Configuração do LoRa

Os parâmetros de configuração disponíveis para a customização da camada física do LoRa são os seguintes: a frequência da portadora, a largura de banda, a taxa de código (*Code Rate* – CR), e o fator de espalhamento. A frequência da portadora define a frequência central para a banda de transmissão. É definida de acordo com a região de operação dos equipamentos. Assim sendo, este parâmetro não é, em geral, ajustável de acordo com a aplicação. A largura de banda, por sua vez, define o tamanho da faixa de frequências utilizada, com três valores programáveis: 125 kHz, 250 kHz e 500 kHz. O SF define o espalhamento espectral [Semtech Corp., 2013, Bor et al., 2016a]. Assim, estabelece a razão entre a taxa de bits e a taxa de *chirps*. A especificação LoRa define seis valores diferentes para o parâmetro fator de espalhamento: SF7, SF8, SF9, SF10, SF11 e SF12 [Semtech Corp., 2015], o que permite formar canais ortogonais, fazendo com que enlaces com fatores de espalhamento diferentes não tenham colisões entre si. Um maior SF aumenta a sensibilidade do limiar de recepção em termos de potência, porém, aumenta o tempo de propagação no ar e diminui a taxa de transmissão do enlace [Raza et al., 2017]. A taxa de bits, então, é dada por:

$$R_b = SF \times \frac{BW}{2^{SF}}, \quad \text{com } SF \in \{7, 8, 9, 10, 11, 12\}. \quad (1)$$

A CR define quantos bits são utilizados para redundância na mensagem, para realizar a recuperação de erros. A taxa de codificação T_c [Semtech Corp., 2015] como:

$$T_c = \frac{4}{4 + CR}, \quad \text{com } CR \in \{1, 2, 3, 4\}. \quad (2)$$

Também é especificada uma taxa de transmissão R_b teórica, definida em função do fator de espalhamento e da taxa de código, representando a taxa de bits efetiva como:

$$R_b = SF \times \frac{T_c \times BW}{2^{SF}}, \quad \text{com } SF \in \{7, 8, 9, 10, 11, 12\}. \quad (3)$$

A Tabela 1 exibe os valores teóricos de taxa de transmissão de diferentes fatores de espalhamento, para uma CR igual a 4/5 e uma largura de banda de 500 kHz. Esses valores teóricos são utilizados como referência para a análise de desempenho deste trabalho. A Seção 4 compara esses valores com valores obtidos experimentalmente, a fim de estabelecer bases para as aplicações que utilizam LoRa e LoRaWAN como tecnologias de comunicação.

Tabela 1. Valores teóricos de taxa de transmissão em função do SF.

Fator de espalhamento	Taxa de transmissão teórica (bits/s)
SF7	21875
SF8	12500
SF9	7031
SF10	3906
SF11	2148
SF12	1172

Outro parâmetro necessário para descrever o comportamento do enlace de comunicação sem-fio na simulação é o modelo de propagação. Os modelos de propagação descrevem o comportamento das ondas eletromagnéticas, e como elas são irradiadas desde o transmissor até o receptor. O modelo de propagação que descreve melhor o cenário escolhido para os experimentos práticos é o *LogDistance*. Este modelo é amplamente usado em cenários de larga escala e em enlaces com frequências da ordem dos MHz, e prediz a atenuação do sinal, perto dos valores que foram medidos nos experimentos práticos, usando como parâmetros a potência do sinal e a distância:

$$PL(d) = PL(d_o) + 10n \text{Log}\left(\frac{d}{d_o}\right). \quad (4)$$

Devido ao *LogDistance* implementado no simulador ser um modelo que não considera a variação da atenuação por flutuações do meio de propagação, foi adaptada uma variável de distribuição normal randômica que descreve as perdas ocasionadas pelo fenômeno de desvanecimento [Rappaport, 2001]. Assim, o modelo de propagação usado para a comparação entre os experimentos práticos e simulados foi implementado usando a seguinte equação:

$$PL(d) = PL(d_o) + 10n \text{Log}\left(\frac{d}{d_o}\right) + X_\sigma, \quad (5)$$

onde n descreve o coeficiente empírico de propagação, e X_σ descreve a variação do desvanecimento em termos de potência (dB); o intervalo de valores para X_σ foi obtido a partir dos dados obtidos da potência do sinal nos experimentos práticos.

4.2. Parâmetros dos Experimentos Desenvolvidos

Os experimentos realizados para a avaliação da tecnologia LoRa consistem de medições práticas e simuladas da vazão, da taxa de perda, da potência do sinal e da relação entre a vazão teórica e a vazão obtida através dos experimentos. Todas as métricas são avaliadas em função da variação do fator de espalhamento e da distância. As medições práticas são realizadas por meio da implementação de um protótipo em *hardware*, enquanto as simulações são executadas utilizando a biblioteca LoRaWAN implementada em [Magrin et al., 2017] para o simulador de rede NS-3.

Os parâmetros de configuração dos módulos são descritos na Tabela 2. A comunicação consiste na transmissão de tuplas de dados pela unidade transmissora para cada valor de SF (7, 8, 9, 10, 11) programado. O SF12 não está disponível na biblioteca usada. Para cada transmissão, são feitas 10 rodadas, e em cada rodada são transmitidos 51 pacotes, numerados de 0 a 50. Cada pacote inclui o número de sequência, as coordenadas geográficas (obtidas pelo receptor GPS a cada 250 ms), a altitude, a temperatura e a umidade (ambas medidas pelo sensor). Foram usados 64 bits de preâmbulo. O tamanho total do pacote enviado é de 47 bytes. No cenário escolhido, é feito um estudo comparativo do comportamento dos fatores de espalhamento definidos na camada física do LoRa em função da distância.

Para realizar os experimentos práticos com os dispositivos físicos LoRa, é implementado um protótipo em *hardware* proposto em [Ortiz et al., 2018]. As simulações são

Tabela 2. Parâmetros de configuração dos experimentos práticos.

Frequência	915 MHz
Potência de transmissão	14 dBm
Fator de espalhamento	SF (7, 8, 9, 10, 11)
Largura de banda	500 kHz
Taxa de código	4/5

feitas no NS-3, na versão 28. É utilizada e adaptada a biblioteca pública do módulo de rede LoRaWAN disponível em¹ [Magrin et al., 2017], que simula até a camada de rede. Os modelos de implementação fornecidos são ajustados para compatibilizá-los com os cenários de experimentação planejados para este trabalho. Para isso, é gerado um ambiente de simulação em que os nós se comunicam às distâncias estabelecidas no experimento prático (Seção 4.3). É utilizada na simulação a mesma metodologia de envio de pacotes em função da variação do SF e da distância, assim como os parâmetros mostrados na Tabela 2. É utilizado o modelo de propagação *LogDistance* nativo do NS-3, que se aplica à natureza do ambiente dos experimentos práticos, área urbana com linha de visada desobstruída. Entre os modelos avaliados, o *LogDistance* é o que mais se aproxima do comportamento observado nos experimentos práticos. Entretanto, para descrever melhor a variação do sinal, é acrescentada neste trabalho uma variável de distribuição normal randômica com média $\mu = 0$ e $\sigma^2 = 1,47$, descrita na Equação 5. Estes valores que descrevem a variável X_σ foram obtidos a partir dos experimentos práticos.

4.3. Cenário dos Experimentos

O cenário utilizado para a realização dos experimentos para avaliação prática da tecnologia LoRa compreende pontos localizados na Ilha do Governador e no campus da Universidade Federal do Rio de Janeiro (UFRJ), Ilha do Fundão, selecionados de forma a obter enlaces com linha de visada desobstruída e comprimento variável. Foram usadas duas locações para a unidade receptora: o ponto mais alto do edifício do Centro de Tecnologia e do edifício do Hospital Universitário Clementino Fraga Filho (HUCFF/UFRJ). Já a unidade transmissora foi posicionada em seis pontos pré-definidos a 0, 1000, 2000, 3000, 4000 e 5000 m de distância. A Figura 1 mostra o local dos experimentos.



Figura 1. Visão aérea da Ilha do Governador e do campus Ilha do Fundão da UFRJ, cenário dos experimentos práticos (obtida com o Google Earth).

¹<https://github.com/signetlabdei/lorawan>

5. Resultados

A partir das configurações implementadas nos experimentos práticos e simulados, descritos na Seção 4, são comparados os resultados das métricas de vazão, taxa de perda, potência do sinal e a relação entre a vazão teórica e a vazão obtida dos experimentos. Em todos os cenários, foram feitas 10 rodadas para cada distância e fator de espalhamento, com intervalo de confiança de 95%, representado por barras de erro verticais nos gráficos. Para determinar a correlação dos experimentos práticos e simulados, foi usada a análise correlacional de Pearson, mostrada na Equação 6.

$$r = \frac{\sum(x - \mu_x)(y - \mu_y)}{\sqrt{\sum(x - \mu_x)^2 \sum(y - \mu_y)^2}}, \quad (6)$$

5.1. Análise de Vazão

Para avaliar a vazão, são considerados os tempos entre a primeira e a última recepção do conjunto de pacotes enviado, assim como o tamanho de cada pacote enviado. As figuras 2(a) e 2(b) apresentam a vazão do enlace em segmentos de 1000 m nos experimentos práticos e simulados. Como pode-se observar, nos experimentos práticos e simulados todos os SFs mantiveram uma vazão constante para todos os segmentos, até a qualidade do enlace se degradar pelo aumento da distância, diminuindo a vazão. Esse comportamento está de acordo com o esperado para distâncias maiores e é percebido de forma mais acentuada nos SF7 e SF8. A vazão máxima experimental foi de ≈ 1 kbit/s, semelhante ao experimento simulado. Também se observa que no SF7 e SF8 a degradação do sinal aconteceu um segmento antes que no experimento prático. Isto está associado aos níveis de sensibilidade de recepção. No modelo simulado, há um nível de sensibilidade fixo para cada SF. No modelo prático, há uma faixa de sensibilidade informada para cada SF [Dragino, 2017]. A diferença entre os valores de sensibilidade real e simulado influenciou mais os resultados para o SF7 e SF8.

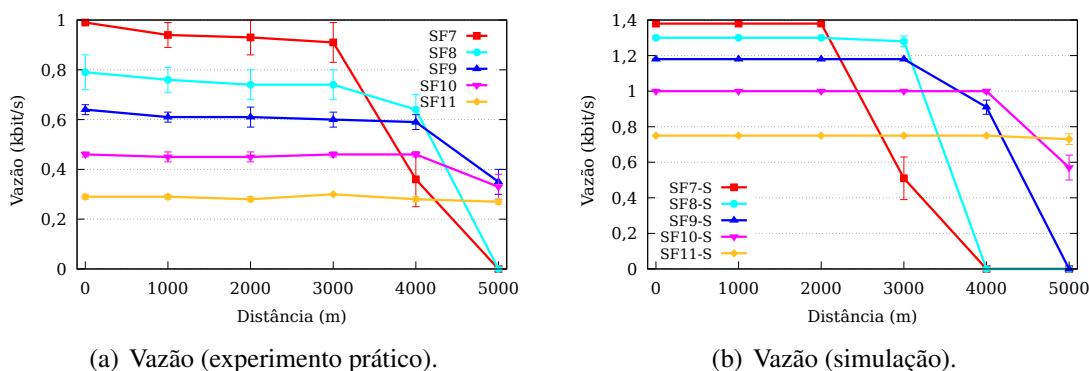


Figura 2. Vazão obtida nos experimentos prático e simulado.

Tabela 3. Correlação da vazão nos experimentos prático/simulado.

	SF7/SF7-S	SF8/SF8-S	SF9/SF9-S	SF10/SF10-S	SF11/SF11-S
r	0,8568606	0,7484335	0,9829023	0,9954955	0,9079594

No entanto, o resultado das curvas obtidas nos experimentos práticos e simulados correspondem ao comportamento teórico da tecnologia, como é mostrado na Tabela 3,

onde é apresentada a correlação entre a vazão prática e a simulada. É possível concluir que qualquer variação na vazão prática está representada na vazão simulada, mostrando um grau de associação alto nos experimentos, com exceção no SF7-S e o SF8-S, que apresenta uma correlação moderada com os outros SFs por conta da diferença na vazão. Isto quer dizer que qualquer variação no experimento prático será refletida no experimento simulado, sempre e quando o cenário for o mesmo.

5.2. Análise da Taxa de Perda

Para a análise da taxa de perda, são considerados os números de sequência que identificam cada pacote no experimento prático. A taxa de perda observada nos experimentos prático/simulado apresenta um comportamento condizente com os resultados da análise da vazão, aumentando com a distância. As Figuras 3(a) e 3(b) apresentam a taxa de perda no enlace nos experimentos práticos e simulados, respectivamente. Comparando as curvas obtidas, a taxa de perda apresenta comportamento semelhante na prática e simulação. No experimento simulado, a perda de pacotes nos SF7 e SF8 inicia 1000 m antes em comparação com o experimento prático. Por outro lado, o SF9 teve uma taxa de perda total em 5000 m, enquanto no experimento prático teve uma perda inferior a 20%. Já o SF11 teve um comportamento semelhante ao SF11 do experimento prático, exceto no ponto de 1000 m, onde teve uma perda de $\approx 10\%$. Esta perda se deve a uma maior atenuação causada pela presença de árvores no enlace. Embora não obstruam a linha de visada, o cenário dos experimentos apresenta variações de ambiente entre os diferentes pontos. Por fim, ao analisar o SF10 e SF11, verifica-se uma perda não superior a 20% nos experimentos práticos, e uma perda não superior a 50% nos experimentos simulados, o que indica que em condições ideais de linha de visada, o enlace pode atingir distâncias maiores usando os fatores de espalhamento mais altos.

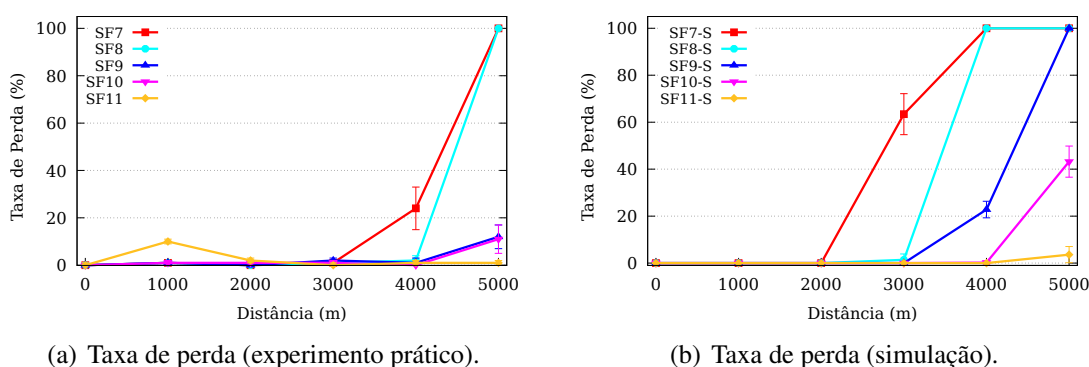


Figura 3. Taxa de perda obtida nos experimentos prático e simulado.

Tabela 4. Correlação da taxa de perda nos experimentos prático/simulado.

	SF7/SF7-S	SF8/SF8-S	SF9/SF9-S	SF10/SF10-S	SF11/SF11-S
r	0,6973851	0,6438531	0,9658178	0,9930800	-0,1705606

A correlação entre os resultados da taxa de perda é mostrada na Tabela 4. Se observa que há uma correlação com grau de associação negativo entre o SF11 e os SFs simulados. Isto acontece porque, embora na prática o SF11 consiga manter a comunicação do enlace em 5000 m, teve perdas recorrentes em todos os pontos, além de uma perda

acentuada no ponto de 1000 m, o que altera a correlação com os outros SFs. Diferente do prático, no experimento simulado, enquanto houve troca de pacotes, a perda foi diferente de 0 para os SF maiores.

5.3. Medições de Potência do Sinal

As medições de potência do sinal foram obtidas das bibliotecas implementadas nos experimentos práticos e nos experimentos simulados, não se utilizando um instrumento de medidas externo. As Figuras 4(a) e 4(b) apresentam a potência do sinal no enlace. O comportamento é muito semelhante para os dois experimentos, apesar da diferença inicial de ≈ -10 dBm. Assim, em 0 m, a potência do sinal no ambiente simulado foi de ≈ -50 dBm, enquanto no experimento prático foi de ≈ -37 dBm. Já em 1000 m, é medida a mesma potência (≈ -110 dBm) para os dois experimentos. Nas distâncias seguintes a potência reduz exponencialmente. O experimento prático apresenta algumas variações de potência devido ao cenário onde foi implementado, mas sem diferença estatística representativa. Se observa essa variação nos pontos de 2000 m e 3000 m. Já nos pontos de 4000 m e 5000 m, a tendência da potência foi semelhante, decrescendo abaixo dos -120 dBm. Nos dois experimentos, há uma queda abrupta nos primeiros 1000 m, e mantendo um decremento exponencial com o aumento da distância. No experimento simulado a potência do sinal vai decrescendo exponencialmente, conforme o descrito na literatura. Nos experimentos práticos, o único SF que mostrou um comportamento anormal foi o SF7 no ponto de 4000 m, onde mediu-se uma melhora na potência do sinal, com respeito aos pontos anteriores. Isso é atribuído a variações na atenuação do enlace e na propagação da onda em si. Uma vez que para atingir este ponto o enlace passa por um espelho de água, que pode eventualmente causar uma composição construtiva entre os raios direto e refletido.

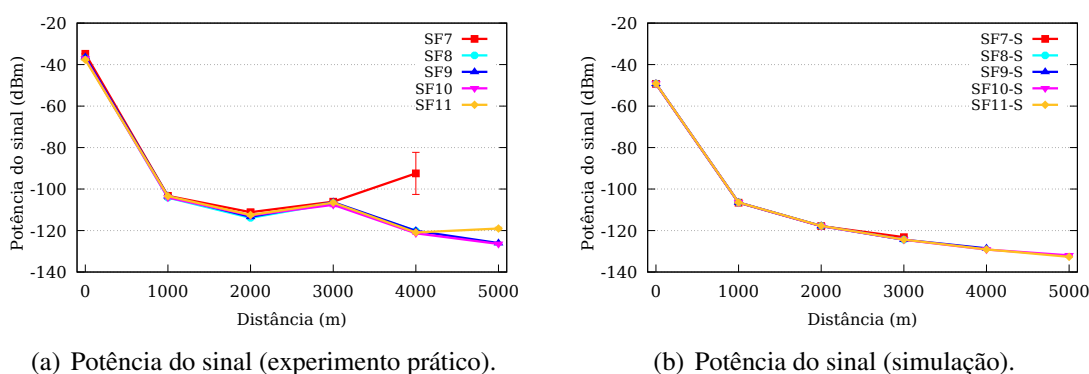


Figura 4. Potência do sinal obtida nos experimentos prático e simulado.

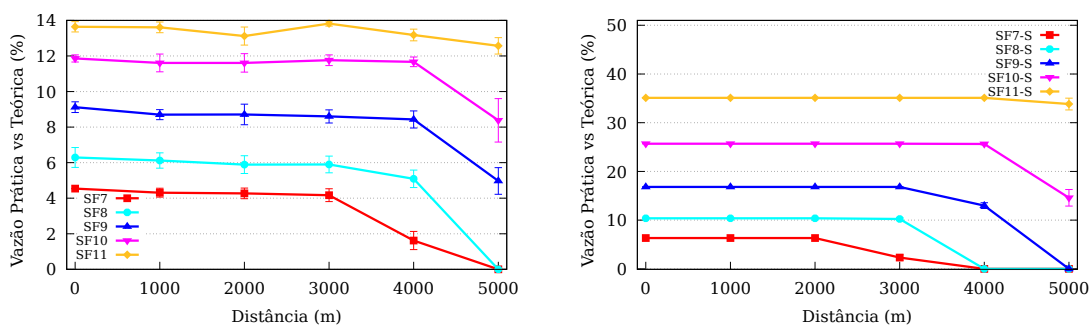
Tabela 5. Correlação da potência do sinal nos experimentos prático/simulado.

	SF7/SF7-S	SF8/SF8-S	SF9/SF9-S	SF10/SF10-S	SF11/SF11-S
r	0,8457486	0,9575271	0,9748970	0,9874096	0,9845240

Condizente com os resultados dos experimentos prático/simulado, todos fatores de espalhamento estão fortemente correlacionados. A Tabela 5 mostra que os resultados de potência do sinal dos experimentos prático e simulado correspondem um com o outro.

5.4. Relação de Vazão Teórica vs. Vazão Prática

Para avaliar a capacidade do enlace, foi comparada a taxa de transmissão calculada na Tabela 1, e os valores obtidos nos experimentos prático e simulado. As Figuras 5(a) e 5(b) mostram o resultado da relação entre vazão teórica e vazão medida. Enquanto no experimento prático a relação varia de 4% a 14%, no experimento simulado a relação assume valores de 6% a 36%. Este resultado mostra que no SF7 há uma relação similar, com a relação em torno de 5% para os experimentos prático e simulado. Nos outros SFs, o experimento simulado apresenta valores que praticamente dobram a relação de vazão teórica *versus* prática em comparação com os valores obtidos no experimento prático. Estes resultados estão associados ao atraso inserido para obter dados de localização atualizados (de 250 ms). Em [Ortiz et al., 2018], a relação de vazão foi maior, porém, foram enviadas tuplas de dados com bits aleatórios para atingir a máxima capacidade do enlace, o que incorreu em uma taxa de perda maior em menores distâncias, e menor alcance.



(a) Vazão prática vs. vazão teórica (experimento prático). (b) Vazão prática vs. vazão teórica (simulação).

Figura 5. Vazão prática vs. vazão teórica nos experimentos prático e simulado.

Os resultados dos experimentos demonstram, como esperado, o comportamento da camada física do LoRa em um enlace. A influência do tamanho do pacote torna-se importante porque o enlace é altamente suscetível a interferências e, portanto, há erros de bit que podem piorar o desempenho da rede na presença de mais dispositivos terminais. No experimento prático, a redução do tamanho do pacote permitiu ter um maior alcance entre a unidade transmissora e receptora, chegando neste trabalho a 5000 m. Por outro lado, devido que a transmissão está sendo espalhada através de diferentes ambientes (vegetação, espelhos de água, variações climáticas), há a tendência de que a qualidade da comunicação do enlace torne-se instável e com fortes variações de potência de sinal e perda de pacotes. Como foi observado na Seção 5.3, as medições de 4000 m e 5000 m, que ocorrem acima da água, apresentam variações na potência do sinal, especificamente nos SF7, SF10 e SF11. O SF11 também apresenta alterações nas medições de 1000 m por conta da vegetação ao redor da antena transmissora, como foi observado na Seção 5.2. Portanto percebe-se que a simulação de redes de longo alcance torna-se fundamental para o planejamento de redes LPWAN, para que seja possível avaliar os enlaces sem estar sujeito a estas diferenças do ambiente. Por isto, é imperativo que o modelo de simulação da camada física do LoRa, projetado para o simulador NS-3, continue sendo aprimorado, de forma a refletir um ambiente mais coerente com aquele obtido com o experimento prático.

6. Conclusão e Trabalhos Futuros

Este trabalho comparou a avaliação de desempenho de uma rede LoRa através de experimentos práticos e simulados. O objetivo foi analisar a correspondência entre os dados coletados nos dois ambientes. Foi utilizado para a simulação um modelo de camada física LoRa implementado no simulador NS-3. Dentro deste modelo, foi utilizado um modelo de propagação adequado para grandes áreas, e calculada uma variável para a análise das perdas por desvanecimento do sinal, característico das redes LPWAN. Os resultados revelaram que todas as métricas avaliadas nos experimentos simulados foram consistentes com os resultados dos experimentos práticos. Por outro lado, as diferenças encontradas nos resultados dos experimentos práticos e simulados sugerem um aperfeiçoamento no modelo usado para a simulação do enlace LoRa. Pode-se concluir que o modelo de simulação pode ser melhorado, procurando uma correlação mais forte com experimentos práticos.

Como trabalhos futuros, a operação da biblioteca do NS-3 lorawan será analisada no contexto de redes de múltiplos saltos, avaliação da vazão máxima da rede, quantidade de dispositivos conectados e testes de dispositivos classe B e C. Também pretende-se avaliar a combinação com outras tecnologias para IoT, a fim de aproveitar as vantagens de cada uma em termos de alcance, consumo de energia e taxas de dados.

7. Agradecimentos

Este trabalho foi parcialmente financiado pelas agências de fomento CAPES, CNPq, FAPERJ e pelos processos nº 15/24494-8 e nº 15/24490-2, da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP).

Referências

- Adelantado, F., Vilajosana, X., Tuset-Peiro, P., Martinez, B., Melia-Segui, J. e Watteyne, T. (2017). Understanding the limits of LoRaWAN. *IEEE Communications Magazine*, 55(9):34–40.
- ANATEL (2016). Plano de atribuição, destinação e distribuição de frequências no Brasil. <http://www.anatel.gov.br/institucional/acervo-documental>. Acessado em fevereiro de 2019.
- ANATEL (2018). Faixas de radiofrequências utilizáveis por equipamentos de radiação restrita. <http://www.anatel.gov.br/legislacao/resolucoes/2018/1220-resolucao-705>. Acessado em fevereiro de 2019.
- Augustin, A., Yi, J., Clausen, T. e Townsley, W. M. (2016). A study of LoRa: Long range & low power networks for the Internet of things. *Sensors*, 16(9):1466.
- Bor, M., Vidler, J. e Roedig, U. (2016a). Lora for the internet of things. Em *International Conference on Embedded Wireless Systems and Networks (EWSN)*, p. 361–366.
- Bor, M. C., Roedig, U., Voigt, T. e Alonso, J. M. (2016b). Do LoRa low-power wide-area networks scale? Em *ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, p. 59–67.
- Carneiro, G. (2010). NS-3: Network simulator 3. Em *UTM Lab Meeting April*, volume 20.

- Centenaro, M., Vangelista, L., Zanella, A. e Zorzi, M. (2016). Long-range communications in unlicensed bands: the rising stars in the IoT and smart city scenarios. *IEEE Wireless Communications*, 23(5):60–67.
- Dragino (2017). Manual. Disponível em: http://wiki.dragino.com/index.php?title=Lora_Shield. Acessado em fevereiro de 2019.
- Evans, D. (2011). The internet of things: How the next evolution of the Internet is changing everything. *Cisco Internet Business Solutions Group (IBSG)*, 1:1–11.
- Goursaud, C. e Gorce, J.-M. (2015). Dedicated networks for IoT : PHY / MAC state of the art and challenges. *EAI endorsed transactions on Internet of Things*.
- Henderson, T. R., Lacage, M., Riley, G. F., Dowell, C. e Kopena, J. (2008). Network simulations with the NS-3 simulator. Em *SIGCOMM Demo*, p. 527.
- LoRa™-Alliance (2016). LoRa™-alliance technology. Disponível em: <https://www.lora-alliance.org/technology>. Acessado em fevereiro de 2019.
- Magrin, D., Centenaro, M. e Vangelista, L. (2017). Performance evaluation of LoRa networks in a smart city scenario. Em *IEEE International Conference on Communications (ICC)*, p. 1–7.
- Mikhaylov, K., Petäjajarvi, J. e Haenninen, T. (2016). Analysis of capacity and scalability of the LoRa low power wide area network technology. Em *European Wireless Conference*, p. 1–6.
- Ortiz, F. M., Cruz, P., de S. Couto, R. e Costa, L. H. M. K. (2018). Caracterização de uma Rede Sem-fio de Baixa Potência e Longo Alcance para Internet das Coisas. Em *Simpósio Brasileiro de Redes de Computadores (SBRC)*.
- Petäjajarvi, J., Mikhaylov, K., Roivainen, A., Hanninen, T. e Pettissalo, M. (2015). On the coverage of LPWANs: range evaluation and channel attenuation model for LoRa technology. Em *International Conference on ITS Telecommunications (ITST)*, p. 55–59.
- Rappaport, T. (2001). *Wireless Communications: Principles and Practice*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2a edição.
- Raza, U., Kulkarni, P. e Sooriyabandara, M. (2017). Low power wide area networks: An overview. *IEEE Communications Surveys & Tutorials*, 19(2):855–873.
- Reynders, B., Meert, W. e Pollin, S. (2016). Range and coexistence analysis of long range unlicensed communication. Em *International Conference on Telecommunications (ICT)*, p. 1–6.
- Semtech Corp. (2013). LoRa™ modem designer's guide. Disponível em: <https://www.semtech.com/uploads/documents/>. Acessado em fevereiro de 2019.
- Semtech Corp. (2015). Lora™ modulation basics an1200.22. Disponível em: <https://www.semtech.com/uploads/documents/>. Acessado em fevereiro de 2019.
- Van den Abeele, F., Haxhibeqiri, J., Moerman, I. e Hoebeke, J. (2017). Scalability analysis of large-scale LoRaWAN networks in NS-3. *IEEE Internet of Things Journal*, 4(6):2186–2198.

Detecção de Eventos no Twitter através de Grafos de Visibilidade Natural

Fernanda Tenório¹, Eduarda T. C. Chagas², Pedro H. Barros², Heitor S. Ramos²

¹ Laboratório de Computação Científica e Análise Numérica (LaCCAN)
Universidade Federal de Alagoas (UFAL) – Maceio, AL – Brazil

² Departamento de Ciência da Computação
Universidade Federal de Minas Gerais (UFMG) – Belo Horizonte, MG – Brazil

fernanda.tenorio@ctec.ufal.br

{eduarda-chagas, phbarros}@ufmg.br

heitor@dcc.ufmg.br

Abstract. *The Internet provides an ever-increasing volume of data and information, helping us to better understand its users and the environment around them. One of the approaches used to detect and understand the events that occur in the world has been an analysis of social networks, such as Twitter, used in this article. Thus, considering the change in the dynamics of data behavior after the presence of an event, we propose a new detection method based on the computation of complex network metrics applied to the bigram extracted from the content of tweets, identifying events through system dynamics changes. To validate our approach, we used two sets of data collected by [Aiello et al. 2013], in which we observed satisfactory results when compared with the techniques already present in the literature.*

Resumo. *A Internet vem nos fornecendo cada vez mais dados e informações, ajudando a compreender melhor os seus usuários e o ambiente que os rodeiam. Uma das abordagens usadas para detectar e compreender eventos que ocorrem ao redor do mundo vem sendo a análise de redes sociais, como o caso do Twitter, usado no presente artigo. Assim, considerando a mudança da dinâmica do comportamento dos dados após a presença de um evento, propomos um novo método de detecção baseado no cálculo de métricas de redes complexas aplicadas aos bigram extraídos do conteúdo de tweets, identificando eventos por meios de mudanças de dinâmica do sistema. Para validar nossa proposta usamos dois conjuntos de dados coletados por [Aiello et al. 2013], no qual observamos resultados satisfatórios quando comparados com as técnicas já presentes na literatura.*

1. Introdução

Nos últimos anos, a popularização da Internet nos fornece grandes quantidades de dados. Ao se tornar mais acessível ao público, tornou-se crescente o número de usuários que possuem como finalidade a comunicação e o entretenimento. Assim, foi imprescindível o crescimento de mídias sociais e o processo de globalização da informação, transformando a internet em uma valiosa fonte de dados, que ao serem tratados e analisados, podem nos

revelar importantes características do comportamento social, político e econômico dos usuários nas redes.

Sendo a 12^o rede social com maior número de membros ativos, o Twitter possui 100 milhões de usuários ativos diários e cerca de 6.000 *tweets* por segundo, atingindo mais de 500 milhões de *tweets* por dia¹. O aumento no fluxo de informações na internet e o conseqüente o crescimento de mídias sociais, dentre elas o Twitter, se dá principalmente pela oferta de smartphones mais baratos no mercado, cobertura Wi-Fi acessível e expansão das redes de 3G e 4G.

Permitindo que os usuários se comuniquem por meio de micro-mensagens, de até 280 caracteres, o Twitter se tornou uma poderosa ferramenta para detectar e compreender eventos que ocorrem ao redor do mundo. [Dou et al. 2012] define um evento como "uma ocorrência que causa mudanças no volume de dados de texto que discutem o tópico associado em um momento específico". Logo, eventos podem ser vistos como um resumo sucinto dos fluxos de informações nas mídias sociais, revelando a evolução de fenômenos sociais ao longo de um determinado período de tempo.

A análise das relações existentes entre o uso de palavras-chave em um intervalo de tempo, eventos e as respostas dos usuários aos eventos são capazes de revelar informações sofisticadas das opiniões das massas sobre um determinado fenômeno. Embora considerada uma valiosa fonte de informações quando comparada às mídias tradicionais de notícias, devemos salientar que existem alguns desafios ao trabalhar com análise de dados gerados pelo Twitter, como por exemplo:

- (i) Os algoritmos devem possuir uma abordagem escalonável para suportar a grande quantidade de dados gerado pelos *tweets*;
- (ii) Os *tweets* são compartilhados em tempo real, logo possuem uma grande relação com o contexto temporal na qual se encontra inserido;
- (iii) Devido a limitação de espaço em suas mensagens, os *tweets* geralmente possuem a característica de apresentar um conteúdo breve e informal, com frases não estruturadas, erros de digitação e abreviaturas, exigindo um tratamento prévio de seu conteúdo;
- (iv) Nem todos os *tweets* apresentam informações úteis. Como já destacado por [Parikh and Karlapalem 2013] "metade dos *tweets* são inúteis e não transmitem nenhuma informação valiosa". Embora tais dados não prejudiquem o tempo de processamento, eles são extremamente nocivos ao resultado final da análise.

Assim, neste trabalho apresentamos um novo método para detecção de eventos no Twitter baseado na interpretação do grafo de visibilidade, onde inferimos a presença ou não de um evento, por meio de uma técnica de aprendizagem adaptativa aplicada na análise de mudança das frequências dos bigramas² (duas palavras consecutivas para um dado elemento textual) presentes nos *tweets*.

Este trabalho está organizado da seguinte forma: Seção 2 apresenta os trabalhos relacionados a detecção de eventos no Twitter; Seção 3 descreve a metodologia utilizada para análise dos dados; Seção 4 apresenta os principais resultados; e Seção 5 conclui esse

¹<https://www.omnicoreagency.com/twitter-statistics/>

²<https://en.wikipedia.org/wiki/Bigram>

trabalho.

2. Revisão da literatura

Há diversos estudos presentes na literatura que possuem como objetivo extrair informações e detectar e/ou sumarizar eventos nas mídias sociais de modo eficiente. Como pode ser visto a seguir, grande parte das técnicas e abordagens focam na análise dos textos compartilhados pelo Twitter usando tópicos emergentes.

[Sakaki et al. 2010] conseguiram por meio da análise de *tweets* detectar com até 96% de precisão terremotos identificados pela *Japan Meteorological Agency* (JMA) com escala de intensidade sísmica igual ou maior a 3. Neste trabalho, construíram um modelo probabilístico com base em recursos como palavras-chave, número de palavras e o contexto dos *tweets*. Cada usuário do Twitter foi modelado como um sensor e aplicado um filtro de Kalman [Evensen 2003] juntamente com um filtragem de partículas [Nummiaro et al. 2003] conseguindo assim encontrar o centro e a trajetória do local do evento.

[Mathioudakis and Koudas 2010] propuseram o *TwitterMonitor* que utilizando um algoritmo de extração de contexto baseado em [Deerwester et al. 1990] que consegue identificar a tendência das palavras-chave e agrupá-las de acordo com suas co-ocorrências, obtendo não apenas a detecção em tempo real de eventos no Twitter, como também fornecendo uma síntese precisa da descrição de cada tópico.

[Cataldi et al. 2010] aplicaram o algoritmo *PageRank* para mensurar o grau de influência de cada usuário sobre os *tweets*. Por meio da análise da energia do conteúdo, a mensagem pode então ser classificada ou não como emergente e assim definida uma janela temporal para determinar seu ciclo de vida.

[Weng and Lee 2011] propuseram EDCoW (Detecção de Eventos com Agrupamento de Sinais Baseados em *Wavelet*), onde aplicaram análise de sinais [Rosso et al. 2009] e *Wavelets* nas palavras, conseguindo descartar aquelas consideradas triviais e detectar eventos usando agrupamento de sinais das palavras em conjunto com o particionamento do gráfico baseado em modularidade.

[Li et al. 2012] propôs o *Tweetvent*, que consegue detectar segmentos de *tweets* em uma janela de tempo fixa e por meio de uma variante do algoritmo Jarvis-Patrick [Jarvis and Patrick 1973] agrupar eventos candidatos, que posteriormente são comparados com artigos da Wikipédia para constatar a veracidade e importância dos resultados obtidos.

[Parikh and Karlapalem 2013] utilizaram a análise do comportamento das frequências dos bigramas extraídos ao longo de um determinado período de tempo e usando métricas de similaridade conseguiram agrupar palavras-chave relacionadas ao mesmo evento.

[Dang et al. 2016] criaram um modelo baseado em Redes Bayesianas Dinâmicas [Murphy and Russell 2002] para detectar palavras-chave emergentes e utilizaram DBSCAN [Ester et al. 1996] se baseando na co-ocorrência para realizar o agrupamento destas palavras, identificando assim tópicos emergentes. Tal método usa como entrada informações sobre o compartilhamento de *tweets* e o relacionamento de seguidores.

[Ester et al. 1996] desenvolveram um classificador dos eventos mais prováveis de um determinado período de tempo. Para classificar e detectar eventos com bom desempenho, consideram como entrada para o algoritmo o número de eventos que está ocorrendo em um intervalo de tempo e utilizam uma métrica baseada na frequência inversa de frequência do documento (TF-IDF) do bigrama.

Sabendo que o Twitter é caracterizado pela mudança incremental da distribuição dos seus dados, uma vez que alguns usuários possuem um tempo de resposta curto a eventos, enquanto outros demoram mais para realizar o *tweet*, nossa abordagem consiste em identificar a mudança da dinâmica do sistema e assim detectar a ocorrência de eventos. Mudanças de dinâmica não devem ser analisadas usando o mesmo conjunto de técnicas aplicadas para sistemas estacionários, sendo neste caso recomendadas abordagens adaptativas [Gama et al. 2014]

Nossa proposta se difere das demais devido ao fato de considerarmos que o sistema sofre uma mudança de dinâmica, ou seja, estamos observando uma mudança de dinâmica através da estrutura do grafo de visibilidade. Logo, buscamos observar mudança de comportamento, e não somente uma detecção de anomalias (*outliers*) como as técnicas encontradas na literatura usualmente fazem.

3. Metodologia

3.1. Conjunto de Dados

Para validar nossa proposta, usamos dois conjuntos de dados coletados por [Aiello et al. 2013], onde os autores reuniram *tweets* relacionados a alguns importantes eventos mundiais que aconteceram em 2012:

1. **FA Cup**³: Este conjunto de dados contém *tweets* sobre a Copa de Futebol Inglês (FA Cup). Esta é a principal competição masculina de futebol no país e pertence à mais antiga associação de futebol do mundo. A Figura 1 mostra a série temporal correspondente aos *tweets* coletados sobre a FA Cup. Cada amostra contém o número de *tweets* coletados em um minuto. Em 2012, Chelsea e Liverpool jogaram a partida final, com gols de Ramirez (11') e Drogba (52') para o Chelsea e Carrol (62') para o Liverpool. Assim, o Chelsea venceu a partida por 2 – 1, tendo esta duração de 90 minutos, mais 15 minutos de intervalo;
2. **The Super Tuesday Primaries**: Nos Estados Unidos, o presidente é eleito em uma eleição indireta, com o vencedor sendo determinado por eleitores do Colégio Eleitoral. *Super Tuesday* refere-se informalmente às terças-feiras no início de uma temporada presidencial dos EUA, quando o maior número de estados realiza eleições primárias. A *Super Tuesday* em 2012 aconteceu em 6 de março de 2012, com 419 delegados (18,3 % do total). Mostramos na Figura 2 a série temporal correspondente aos *tweets* coletados sobre a *Super Tuesday* onde cada amostra contém o número de *tweets* coletados em cinco minutos.

[Aiello et al. 2013] criaram esses conjuntos de dados usando as *hashtags* oficiais do evento. Eles construíram o *ground truth* verificando os principais relatórios de mídia para identificar tópicos significativos para cada conjunto de dados. Foi identificado 13 e 22 tópicos significativos para o FA Cup e *Super Tuesday*, respectivamente. Os detalhes da construção do conjunto de dados podem ser encontrados no artigo original da proposta.

³https://en.wikipedia.org/wiki/FA_Cup

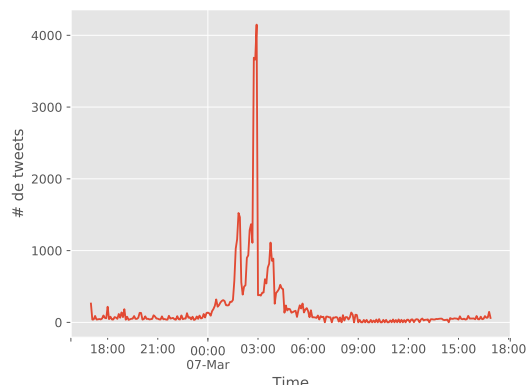
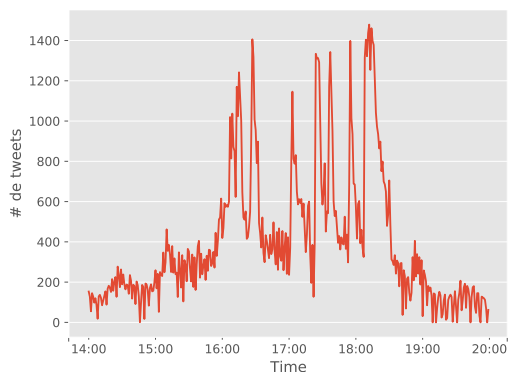


Figura 1. Série temporal do número de *tweets* para o conjunto de dados do FA Cup Figura 2. Série temporal do número de *tweets* para o *Super Tuesday*

3.2. Baseline

A distribuição de Poisson [Poisson 1837] é uma distribuição de probabilidade discreta que expressa a probabilidade de um número de eventos que ocorrem em um período de tempo fixo, dado que esses eventos ocorrem com uma taxa média conhecida e independentemente do tempo desde o último evento [Katti and Rao 1968]. Portanto a distribuição de Poisson é o modelo probabilístico mais comum para detectar eventos incomuns e independentes em uma unidade específica de espaço ou tempo.

Consideramos a taxa de Poisson constante dentro de uma janela de tempo (Processo de Poisson homogêneo), onde foi estimado a taxa de Poisson $\hat{\lambda}$ de um determinado tempo como a média do número de tweets coletados dentro de uma janela deslizante $P^{i:n} = \{p_i, p_{i+1}, \dots, p_{i+n-1}\}$, que corresponde à frequência de todos os tweets em um horário i . Nós usamos 15 unidades de um minuto para calcular a janela deslizante. A taxa de Poisson $\hat{\lambda}$ é estimada como $\hat{\lambda} = \frac{1}{n} \sum_{j=i}^{i+n-1} p_j$.

Um evento é detectado quando observamos que a probabilidade de $k = p_{i+n}$ tweets em $P^{i:n}$ é suficientemente menor que um limite ϵ dentro de uma determinada janela deslizante. Avaliamos a probabilidade de k em $P^{i:n}$ como $\Pr(k; \hat{\lambda}) = \hat{\lambda}^k e^{-\hat{\lambda}} / k!$, com $\hat{\lambda} < k$. Nós detectamos um evento sempre que a distribuição de probabilidade (PDF) em k for menor que um determinado limite ϵ . Isso significa que k provavelmente será uma observação rara, dada a média $\hat{\lambda}$.

Devido à alta sensibilidade apresentada pela distribuição de Poisson, ou seja, é exponencial em λ , consideramos $\epsilon = 10^{-20}$ como um comportamento de anomalia. Se um evento for detectado na janela deslizante $P^{i:n}$, presumimos que o tempo $i+n$ é responsável pela anomalia, ou seja, a hora em que o evento ocorreu.

Após a detecção dos eventos, obtemos os bigramas mais frequentes, agrupando-os com base nos termos (palavras) em comum, formando as palavras-chave relacionadas ao evento detectado.

Observe que, embora a distribuição de Poisson detecte eventos sem conhecer o conteúdo dos *tweets*, nosso método difere pois analisa o assunto presente em cada *tweet*

Bigrama	Valor
game-soccer	4
sunday-game	4
fair-play	4
twitter-now	3
match-today	3
half-time	3
now-win	3
competing-teams	2
soccer-great	2
final-match	2
spirit-team	2
start-match	1

Tabela 1. Exemplo de alguns bigramas para uma dada unidade de tempo.

Bigrama	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7
start-match	1	3	4	2	40	47	62	79

Tabela 2. Série temporal para o bigrama **start-match**.

por meio de bigramas.

3.3. Proposta

3.3.1. Calcular Bigrama

Utilizando a representação mostrada em [Barros et al. 2018], fizemos o mapeamento da contagem do número de bigramas por unidade de tempo. A tabela 1 mostra um exemplo para um valor de tempo t_0 no conjunto de dados do FA Cup.

3.3.2. Criação das séries temporais

À medida que o método é desenvolvido para processamento em tempo real, primeiro coletamos um conjunto de observações dentro de uma janela e depois avançamos pela janela para analisar mais dados. Portanto, primeiro determinamos o mapeamento definido na etapa anterior para a primeira janela e continuamos a determinar esses conjuntos para as janelas subsequentes.

A janela é denotada por $W^{i:n} = \{X_p\}$, onde $p \in \{t_i, t_{i+1}, \dots, t_{i+n-1}\}$ é um intervalo de tempo dentro de W , t_i é o tempo inicial e n é o número de elementos de W , um exemplo para $f(b)$ com o bigrama $b = \text{start-match}$. X_p denota o conjunto de contagens para todos os bigramas no momento p . A janela, com tamanho n , desliza uma unidade de tempo, de forma que a diferença entre janelas consecutivas é exatamente uma observação, como podemos ver na tabela 3, para uma janela de tamanho $n = 4$. Observe que essa janela consideramos apenas a contagem de um bigrama específico.

Tabela 3. Abordagem da janela deslizante.

Bigrama - (start, match)								
Tempo	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7
Contagem	1	3	4	2	40	47	62	79
Contagem ($W^{0:3}$)	1	3	4	2	-	-	-	-
Contagem ($W^{1:4}$)	-	3	4	2	40	-	-	-
Contagem ($W^{2:5}$)	-	-	4	2	40	47	-	-
Contagem ($W^{3:6}$)	-	-	-	2	40	47	62	-
Contagem ($W^{4:7}$)	-	-	-	-	40	47	62	79

3.3.3. Grafo de Visibilidade

O grafo de visibilidade consiste num algoritmo que mapeia séries temporais em grafos, como podemos ver na figura 3 e 4, onde vemos a transformação de uma série temporal (a esquerda) em um grafo (a direita) utilizando a abordagem do grafo de visibilidade. [Lacasa et al. 2008] estuda como os grafos são úteis na caracterização dessas séries temporais, além de mostrar que os mesmos revelam informações não triviais sobre as séries em estudo, a fim de saber se o processo que a gerou pode ser caracterizada usando a teoria dos grafos.

Ainda em [Lacasa et al. 2008], foi mostrado que os grafos de visibilidade que foram mapeados conseguem herdar estruturas das séries temporais utilizadas no mapeamento, ou seja, série periódicas são convertidas em grafos regulares, séries aleatórias em grafos aleatórios e séries fractais em grafos livre de escala.

Logo, devido ao fato de um evento modificar a dinâmica da frequência dos *tweets*, quando um evento não está ocorrendo, a série temporal é mapeada em um grafo aleatório. Na medida que o evento passa a ocorrer, o grafo se transforma num grafo *small-world*.

Com isso, dado a série temporal para um bigrama b obtido da seção 3.3.2, analisamos o grafo gerado pela transformação do grafo de visibilidade, onde se o mesmo for um grafo *small-world*, consideramos que está ocorrendo um evento.

Uma rede *small-world* pode ser quantificada por um coeficiente ω como visto em [Telesford et al. 2011], definido por

$$\omega = \frac{L_r}{L} - \frac{C}{C_r}$$

, onde L e L_r o comprimento médio da rede analisada e da rede aleatória, respectivamente; C e C_r é o coeficiente de agrupamento da rede analisada e da rede aleatória, respectivamente. Uma transformação simples é adotada $\omega' = 1 - |\omega|$ para obtermos valores de ω' entre 0 (rede não *small-world*) até 1 (rede *small-world*).

Existem outras métricas para quantificação de coeficiente de *small-worldness* existem na literatura, por exemplo [Humphries and Gurney 2008], porém essa técnica se torna altamente influenciável pelo tamanho da rede, fazendo com isso que tenhamos preferido a utilização do coeficiente apresentado em nosso trabalho.

Sabendo que ocorreu um evento para um bigrama b , agrupamos os bigramas com termos (palavras) em comum e assim, construímos um grafo de sumarização, que representa o evento. Ou seja, nesse grafo cada vértice representa uma palavra e as aresta identifica que as palavras pertencem ao mesmo bigrama.

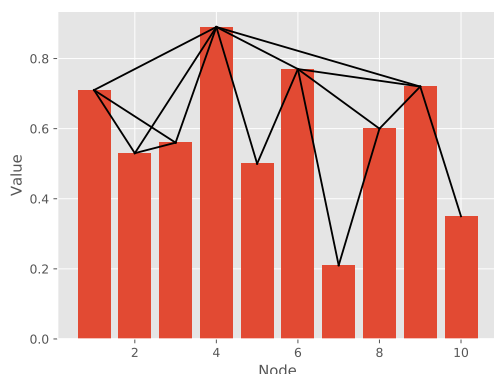


Figura 3. Série temporal

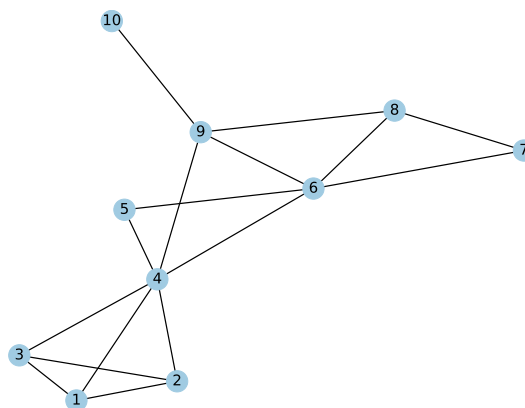


Figura 4. Grafo de visibilidade correspondente a série temporal da Figura 3

3.3.4. Particionar Grafo

Nós dividimos o grafo de sumarização encontrado na etapa anterior para identificar os eventos. Após o particionamento, consideramos que cada grafo conectado com mais de dois elementos encontrados após a partição está associado a um evento.

O Processo de Clusterização Markoviano (MCL), proposto por [Van Dongen 2000] define uma seqüência de processos estocásticos matriciais (inflação e expansão) chamados operadores para particionamento do gráfico.

A idéia principal do algoritmo MCL é simular o fluxo em um grafo normalizado, aumentando o fluxo onde a corrente é forte (muitas visitas do caminhante aleatório para um certo vértice) e abaixando o fluxo onde a corrente é fraca (poucas visitas do caminhante a um certo vértice).

O operador de expansão favorece os caminhos mais curtos, ou seja, passeios aleatórios com poucos passos, favorecendo a visita a novos grupos. Esse operador associa novas probabilidades a todos os pares de nós, diminuindo a probabilidade de caminhos longos e aumentando para caminhos curtos. Assim, o operador de expansão é responsável por permitir que o fluxo conecte diferentes regiões do gráfico.

O operador de inflação é responsável por fortalecer e enfraquecer o fluxo atual. A inflação terá então o efeito de aumentar as chances de passeios dentro do grupo e diminuir as caminhadas entre os grupos. Isso é realizado sem qualquer conhecimento prévio da estrutura de agrupamento.

Decidimos usar esse algoritmo porque ele permite que dois nós estejam em dois grupos separadamente. Em outras palavras, mesmo que dois eventos distintos tenham termos (palavras) em comum, o algoritmo pode segmentá-los de maneira satisfatória.

3.3.5. Avaliação

Para analisar nossos resultados, usamos o conjunto de dados apresentado em [Aiello et al. 2013]. Aplicamos as mesmas métricas utilizadas, para garantir que nossos resultados sejam diretamente comparáveis aos fornecidos.

- *Recall* do tópico (T-Rec): porcentagem de eventos detectados com sucesso, ou seja, a taxa positiva verdadeira para detecção de evento

$$\text{T-Rec} = \frac{\text{ground truth para Tópico} \cap \text{eventos do Tópico detectados}}{\text{ground truth para Tópico}}.$$

- Precisão da palavra-chave (K-Prec): porcentagem de palavras-chave detectadas corretamente sobre o total de palavras-chave de um determinado evento, ou seja, a taxa real negativa para a detecção de palavras-chave

$$\text{K-Prec} = \frac{\text{ground truth para palavras-chave} \cap \text{palavras-chave detectadas}}{\text{palavras-chave detectadas}}.$$

- Recall de palavras-chave (K-Rec): porcentagem de palavras-chave detectadas corretamente sobre o *ground truth* de palavras-chave de um determinado evento, ou seja, a taxa positiva verdadeira de detecção de palavras-chave

$$\text{K-Rec} = \frac{\text{ground truth Palavras-chave} \cap \text{palavras-chave detectados}}{\text{ground truth palavras-chave}}.$$

- F_1 -Score (K-Score): para melhor comparação entre a técnica, adotamos as métricas F_1 -score para palavras-chave

$$\text{K-Score} = 2 \cdot \frac{\text{K-Rec} \cdot \text{K-Prec}}{\text{K-Rec} + \text{K-Prec}}.$$

Observe que essas métricas são calculadas para cada intervalo de tempo.

4. Resultados e Discussões

Logo, como o modelo depende do tamanho da janela deslizante n , nós fizemos uma investigação para determinar qual o valor dessa variável maximizaria a precisão. [Mockus 1975] propôs um novo método chamado Otimização Bayesiana que consiste em otimizar funções como uma "caixa preta", ou seja, quando não conhecemos seu comportamento ou a sua derivada.

O método consiste de, com alguns pontos conhecidos, determinar a forma da função através de uma regressão. Usualmente, essa predição é feita através de um Processo Gaussiano devido a algumas características (escalável para poucos pontos e não paramétrico).

Com isso, tendo como base a regressão do processo gaussiano, ou seja, a possível forma da função que queremos otimizar, é definida uma função utilitária que consiste em achar o próximo candidato para ponto otimizado. Neste trabalho é realizado a discretização para os pontos analisados (devido ao tamanho da janela ser um número inteiro). Utilizamos um ponto inicial aleatório, e posteriormente, 5 rodadas do algoritmo como podemos ver na figura 5, onde foi encontrado o valor de $n = 10$.

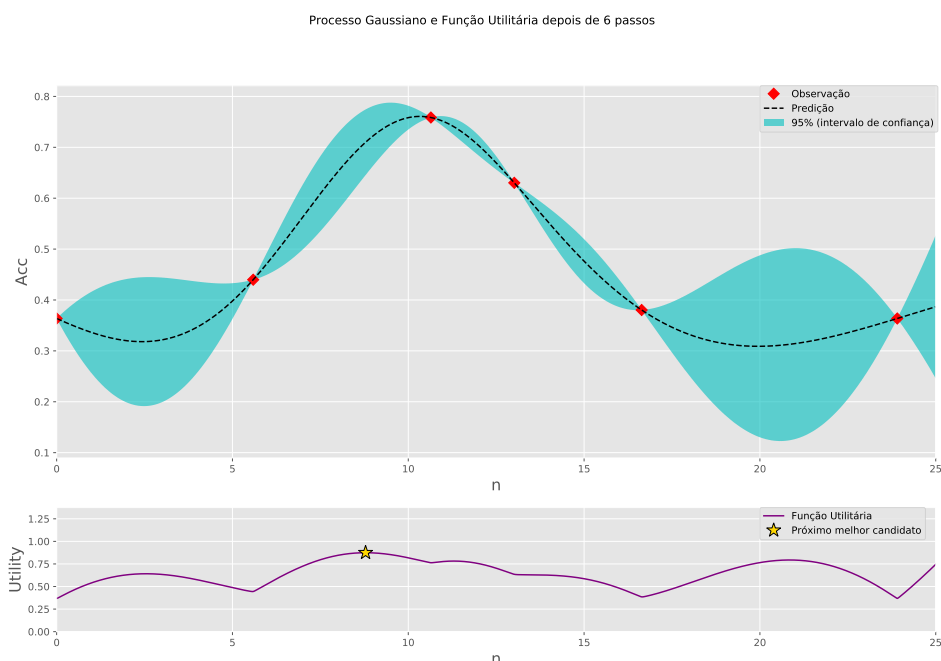


Figura 5. Optimização Bayesiana para o conjunto de dados do FA Cup

Para avaliação dos nossos resultados, nós utilizamos as métricas descritas na seção 3.3.5, como podemos ver nas tabelas 4 e 5, onde comparamos os resultados com algumas técnicas encontradas para literatura que já reportaram resultados para esses conjuntos de dados. Somente os valores de [Choi and Park 2019] e [Barros et al. 2018] foram coletados e copiados dos artigos originais. Os melhores resultados foram apresentados em **negrito**.

Os resultados do método de Poisson já eram esperados. Como a proposta analisa somente o número de *tweets*, sem olhar seu conteúdo, esperávamos que possuísse resultados menos expressivos e por isso, está sendo usada como *baseline*.

Para o conjunto de dados FA Cup, observamos que nossa abordagem é a segunda melhor para a métrica K-Pec e quarta melhor para K-Rec, porém na junção de ambas (K-Score) conseguimos o melhor resultado. Isso indica que nossa proposta possui um balanceamento entre K-Pec e K-Rec, indicando que a mesma consegue um resultado satisfatório para ambas. Para T-Rec nossa proposta é a terceira melhor.

Para o conjunto de dados *Super Tuesday*, nossa abordagem foi a melhor no comparativo com a métrica T-Rec (empatando com duas outras propostas). Obtivemos o quarto e terceiro melhor resultado para as métricas K-Pec e K-Rec, respectivamente. Além disso, considerando o K-Score obtivemos o terceiro melhor resultado.

5. Considerações Finais

Neste trabalho, nós utilizamos o grafo de visibilidade para detectar a ocorrência de um evento em uma rede social. Descobrimos que a ocorrência de um evento apresenta uma mudança de dinâmica que pode ser observada através de métricas em redes complexas.

Tabela 4. Comparação dos métodos usando *FA Cup dataset*

Método	T-Rec	K-Prec	K-Rec	K-Score
Poisson (baseline)	0.308	0.124	0.202	0.154
[Petrović et al. 2010] (Doc-p)	0.692	0.346	0.503	0.410
[Aiello et al. 2013] (FPM)	0.308	0.694	0.512	0.589
[Aiello et al. 2013] (SFPM)	0.615	0.241	0.608	0.345
[Aiello et al. 2013] (BNGran)	0.846	0.310	0.567	0.401
[Aiello et al. 2013] (GFeat-p)	0.238	0.120	0.471	0.191
[Nguyen and Jung 2017]	0.769	0.453	0.548	0.496
[Blei et al. 2003] (LDA)	0.538	0.204	0.643	0.310
[Weng and Lee 2011] (EDCoW)	0.384	0.312	0.357	0.333
[Choi and Park 2019] (HUPM)	0.923	0.320	0.600	0.417
Nossa Proposta	0.769	0.615	0.597	0.605
[Barros et al. 2018]	0.769	0.528	0.596	0.560

Tabela 5. Comparação dos métodos usando *SuperTuesday dataset*

Método	T-Rec	K-Prec	K-Rec	K-Score
Poisson (baseline)	0.091	0.321	0.243	0.154
[Petrović et al. 2010] (Doc-p)	0.182	0.351	0.437	0.389
[Aiello et al. 2013] (FPM)	0.136	0.698	0.372	0.485
[Aiello et al. 2013] (SFPM)	0.273	0.617	0.593	0.605
[Aiello et al. 2013] (BNGran)	0.364	0.522	0.613	0.564
[Aiello et al. 2013] (GFeat-p)	0.091	0.108	0.294	0.158
[Nguyen and Jung 2017]	0.409	0.612	0.714	0.659
[Blei et al. 2003] (LDA)	0.136	0.101	0.212	0.137
[Weng and Lee 2011] (EDCoW)	0.273	0.345	0.381	0.362
[Choi and Park 2019] (HUPM)	0.455	0.420	0.678	0.519
Nossa proposta	0.455	0.525	0.621	0.569
[Barros et al. 2018]	0.455	0.325	0.421	0.367

Com isso, nós propomos um novo método para detectar eventos no Twitter baseando nas séries temporais formadas pela frequência de palavras-chaves extraídas do conteúdo dos *tweets*. Nossa proposta não assume nenhuma informação previa a cerca do *tweet*, e consegue identificar qualquer tipo de evento, sem restrição de idioma utilizado.

Nossa proposta apresenta resultados satisfatórios quando comparadas com o estado-da-arte encontrado na literatura para os conjuntos de dados analisados.

Referências

- Aiello, L. M., Petkos, G., Martin, C., Corney, D., Papadopoulos, S., Skraba, R., Göker, A., Kompatsiaris, I., and Jaimes, A. (2013). Sensing trending topics in twitter. *IEEE Transactions on Multimedia*, 15(6):1268–1282.
- Barros, P., Cardoso, I., A.F. Loureiro, A., and Ramos, H. S. (2018). Event detection in social media through phase transition of bigram entropy. In *IEEE Symposium on Computers and Communications (ISCC)*, Natal, Brazil.
- Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.
- Cataldi, M., Di Caro, L., and Schifanella, C. (2010). Emerging topic detection on twitter based on temporal and social terms evaluation. In *Proceedings of the Tenth International Workshop on Multimedia Data Mining, MDMKDD '10*, pages 4:1–4:10, New York, NY, USA. ACM.
- Choi, H.-J. and Park, C. H. (2019). Emerging topic detection in twitter stream based on high utility pattern mining. *Expert Systems with Applications*, 115:27 – 36.
- Dang, Q., Gao, F., and Zhou, Y. (2016). Early detection method for emerging topics based on dynamic bayesian networks in micro-blogging networks. *Expert Syst. Appl.*, 57(C):285–295.
- Deerwester, S., Duais, S., Furnas, G., Landauer, T., and Harshman, R. (1990). Indexing by latent semantics analysis. *Journal of the American Society for Information Science*, 41(6):391–407.
- Dou, W., Wang, X., Ribarsky, W., and Zhou, M. (2012). Event detection in social media data. In *IEEE VisWeek Workshop on Interactive Visual Text Analytics-Task Driven Analytics of Social Media Content*, pages 971–980.
- Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD'96*, pages 226–231. AAAI Press.
- Evensen, G. (2003). The ensemble kalman filter: Theoretical formulation and practical implementation. *Ocean dynamics*, 53(4):343–367.
- Gama, J., Žliobaitė, I., Bifet, A., Pechevniky, M., and Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, 46(4):44.
- Humphries, M. D. and Gurney, K. (2008). Network ‘small-world-ness’: a quantitative method for determining canonical network equivalence. *PloS one*, 3(4):e0002051.

- Jarvis, R. A. and Patrick, E. A. (1973). Clustering using a similarity measure based on shared near neighbors. *IEEE Trans. Comput.*, 22(11):1025–1034.
- Katti, S. and Rao, A. V. (1968). Handbook of the poisson distribution.
- Lacasa, L., Luque, B., Ballesteros, F., Luque, J., and Nuno, J. C. (2008). From time series to complex networks: The visibility graph. *Proceedings of the National Academy of Sciences*, 105(13):4972–4975.
- Li, C., Sun, A., and Datta, A. (2012). Twevent: Segment-based event detection from tweets. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management, CIKM '12*, pages 155–164. ACM.
- Mathioudakis, M. and Koudas, N. (2010). Twittermonitor: Trend detection over the twitter stream. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, SIGMOD '10*, pages 1155–1158, New York, NY, USA. ACM.
- Mockus, J. (1975). On the bayes methods for seeking the extremal point. *IFAC Proceedings Volumes*, 8(1, Part 1):428 – 431. 6th IFAC World Congress (IFAC 1975) - Part 1: Theory, Boston/Cambridge, MA, USA, August 24-30, 1975.
- Murphy, K. P. and Russell, S. (2002). Dynamic bayesian networks: representation, inference and learning.
- Nguyen, D. T. and Jung, J. E. (2017). Real-time event detection for online behavioral analysis of big social data. *Future Generation Computer Systems*, 66:137 – 145.
- Nummiaro, K., Koller-Meier, E., and Van Gool, L. (2003). An adaptive color-based particle filter. *Image and vision computing*, 21(1):99–110.
- Parikh, R. and Karlapalem, K. (2013). Et: Events from tweets. In *Proceedings of the 22Nd International Conference on World Wide Web, WWW '13 Companion*, pages 613–620, New York, NY, USA. ACM.
- Petrović, S., Osborne, M., and Lavrenko, V. (2010). Streaming first story detection with application to twitter. In *Human language technologies: The 2010 annual conference of the north american chapter of the association for computational linguistics*, pages 181–189. Association for Computational Linguistics.
- Poisson, S. D. (1837). Probabilité des jugements en matière criminelle et en matière civile, précédées des règles générales du calcul des probabilités. *Paris, France: Bachelier*, 1:1837.
- Rosso, O. A., Craig, H., and Moscato, P. (2009). Shakespeare and other english renaissance authors as characterized by information theory complexity quantifiers. *Physica A: Statistical Mechanics and its Applications*, 388(6):916 – 926.
- Sakaki, T., Okazaki, M., and Matsuo, Y. (2010). Earthquake shakes twitter users: Real-time event detection by social sensors. In *Proceedings of the 19th International Conference on World Wide Web, WWW '10*, pages 851–860, New York, NY, USA. ACM.
- Telesford, Q. K., Joyce, K. E., Hayasaka, S., Burdette, J. H., and Laurienti, P. J. (2011). The ubiquity of small-world networks. *Brain connectivity*, 1(5):367–375.
- Van Dongen, S. M. (2000). *Graph clustering by flow simulation*. PhD thesis.
- Weng, J. and Lee, B.-S. (2011). Event detection in twitter. *ICWSM*, 11:401–408.

Um Sistema e-Health Inteligente para Análise e Detecção de Doenças Cardíacas

Diego A. B. Moreira¹, Levy G. Chaves¹, Rafael L. Gomes¹, Joaquim Celestino Jr¹

¹Centro de Ciência e Tecnologia (CCT) - Universidade Estadual do Ceará (UECE)

{diego, levy, rafaellgom, celestino}@larces.uece.br

Resumo. *A indústria de cuidados médicos vem evoluindo a sua prestação de serviços, focando na qualidade do atendimento ao paciente. Tem-se então a necessidade de uma infraestrutura para monitoramento da saúde dos pacientes a partir da análise dos dados destes pacientes, com intuito de detectar possíveis doenças, dentre elas, doenças cardíacas. Dentro deste contexto, este artigo apresenta um sistema e-Health inteligente para analisar os dados biológicos dos pacientes e detectar possíveis complicações cardíacas. No sistema proposto aplica-se três fases: (I) trata-se os dados através de filtros; (II) técnicas para extração das informações úteis; e, (III) os dados são analisados e classificados para detectar doenças cardíacas. Os experimentos realizados utilizando-se uma base dados real mostram que o sistema proposto é capaz de classificar com alta acurácia as anomalias cardíacas.*

Abstract. *The healthcare industry is evolving its service delivery, focusing on the quality of treatment to the patients. There is a necessity to monitor the health of the patients from the data analysis of these patients, aiming to detect possible diseases, among them, cardiovascular diseases. Within this context, this paper presents a e-Health intelligent system to analysis biological data of the patients and to detect possible cardio problems. The proposed system has three phases: (I) filtering the data; (II) useful information extraction; and, (III) analysis and classification to detect possible cardiovascular diseases. The experiments performed used a real database to show the capability of the proposed system to identify cardiovascular anomalies.*

1. Introdução

Devido o grande avanço das tecnologias computacionais, temos diversas áreas de aplicações em expansão. Internet das Coisas (IoT), Computação em Nuvens, sensores de monitoramento, são exemplos de tecnologias utilizadas para melhorar e apoiar a demanda de outras áreas do conhecimento. Ao conectar estes dispositivos a Internet, tem-se uma ampla gama de aplicações que podem facilitar e auxiliar as mais diversas necessidades.

Com uma maior preocupação da saúde humana, uma área em expansão é a e-health que tem como intuito dar suporte à medicina por meio de Tecnologias de Informação e Comunicação (TIC). Esta é uma das áreas mais promissoras e com maior impacto econômico, onde espera-se um crescimento entre \$1.1 a \$2.5 trilhões anualmente pela economia global de 2025, tendo um crescimento superior às demais aplicações [Al-Fuqaha et al. 2015]. Pela interação das Redes IoT com a tecnologia e-health,

esperam-se sistemas capazes de monitorar, analisar e enviar informações sobre determinado paciente em tempo real. Com isso, a aplicação da computação à área médica vem colaborando para melhorar e definir um conjunto de novos processos para a automação na área de saúde. Entretanto, é necessário ter muito cuidado no processo de implantação de sistemas inteligentes por lidar com vidas humanas [Begg et al. 2007].

A grande demanda de cuidados médicos resulta na necessidade urgente de uma infraestrutura de monitoramento de saúde em tempo real para coletar dados biológicos de exames (periodicamente ou em tempo real), analisar estes dados dos pacientes e detectar possíveis doenças, visando aprimorar tratamento dos pacientes e reduzir os casos de alta gravidade [Mendis et al. 2011]. Dentre exames e sinais vitais importantes podemos citar o ECG (Eletrocardiograma), a pressão sanguínea, oxigenação, dentre outros. Adicionalmente, pode-se utilizar os dados biológicos monitorados de forma isolada ou em conjunto a fim de prever possíveis problemas à saúde dos pacientes [Zhao et al. 2018].

Um campo crítico que necessita de grande atenção é o relacionado a Doenças cardiovasculares (DCV), pois representam a principal causa de mortes no Mundo, onde estima-se que quase 18 milhões de mortes em 2016 devido a DCV sendo responsável 31% das mortes mundiais [Organization et al. 2016]. O alerta, a prevenção e o tratamento adequados podem reverter essa grave situação segundo a *World Health Organization* [Organization et al. 2016].

Métodos computacionais foram propostos para automatizar o processo de análise de ondas de eletrocardiograma (ECG), ajudando no processo de tomada de decisão dos médicos, porém poucos trabalhos relatam a detecção e classificação das anomalias referentes a estas ondas. Mesmo que tenham atingido desempenhos razoáveis, vários deles dependem de algoritmos computacionalmente complexos e podem ser muito custosos em termos de processamento e memória.

Dentro deste contexto, este artigo propõe um sistema e-Health inteligente para analisar os dados biológicos coletados dos pacientes e detectar possíveis complicações cardíacas. O principal objetivo é classificar as ondas de ECG, com alta acurácia e rápida resposta, em diversos tipos de anomalias. No sistema proposto aplica-se três fases: (I) trata-se os dados através de filtros, melhorando a capacidade de distinção; (II) técnicas para extração das informações úteis são utilizadas para transformar o sinal original do ECG para um vetor de características significativas; e, (III) os dados são analisados e classificados utilizando Árvores de Decisão, Florestas Aleatórias, Máquina de vetores de suporte e Perceptron Multicamadas (MLP) para detectar doenças cardíacas.

O sistema proposto é avaliado a partir de experimentos que utilizam uma base de dados real [Moody and Mark 2001], onde foram avaliados o tempo de processamento dos dados e a acurácia na detecção de doenças. Os resultados mostram que o sistema proposto é capaz de classificar com alta acurácia 18 diferentes anomalias cardíacas, alcançando uma percentual de 98% de acurácia.

O restante deste artigo está organizado da seguinte forma. A Seção 2 discute vários trabalhos relacionados, levando em consideração a análise de dados cardíacos. A Seção 3 descreve a construção do blocos utilizados para detecção de anomalias e detalha o método proposto, enquanto a Seção 4 descreve os resultados obtidos através do método relatado. Finalmente, a Seção 5 conclui o artigo e apresenta trabalhos futuros.

2. Trabalhos Relacionados

Nesta seção serão descritos os trabalhos relacionados que compreendem a classificação de anomalias e diagnóstico de doenças cardíacas através da onda ECG, com diferentes abordagens.

Um sistema de monitoramento de pacientes é apresentado por Kiranyaz, Ince e Gabbouj (2016). Este trabalho utiliza uma Rede Neural Convolucional (CNN) de uma dimensão para classificação de doenças cardíacas através do ECG. Os dados são analisados separadamente através de redes CNN diferentes, criando assim uma rede personalizada para cada paciente.

Mesmo com bons resultados, este trabalho limita-se na detecção de apenas duas anomalias cardíacas, sendo elas batimento ventriculares prematuros e contrações ventriculares prematuras.

Oresko et. al desenvolveu uma aplicação para smartphones capaz de classificar pontos de importância e doenças cardíacas. Para detecção destes pontos, o trabalho utilizou um algoritmo baseado em [Hamilton and Tompkins 1986] que utiliza um limiar adaptativo. Quanto a classificação dos sinais, Oresko utilizou uma Rede Perceptron de multicamadas (MLP) que como entrada tem-se um sinal de complexo QRS com 51 amostras de segmentos com tempos de 150 ms. A saída da rede MLP é então associada a um dos problemas cardíacos analisados no escopo do trabalho. Para a implementação desta rede utilizou-se o MATLAB e para validação dos resultados, utilizou-se a base de dados MIT-BIH [Moody and Mark 2001].

Este trabalho também apresentou bons resultados para as anomalias analisadas, porém assim como os outros, tem uma limitação na quantidade destas anomalias, trabalhando apenas com quatro, que são: Bloqueio do ramo direito do feixe de His, contração ventricular prematura, batimento cardíaco estimulado por um marcapasso e fusão destes batimentos com batimentos normais.

Em Carrera et al. propõe-se um sistema de monitoramento de ECG, capaz de modelar os batimentos normais de um paciente através de processos de linguagem natural, onde os padrões não conhecidos por este modelo são classificados como anormais. Através de transformações lineares e um conjunto de regras, um dicionário é auto adaptado para os usuários. Através deste modelo, os batimentos anormais são detectados através da distancia entre os subespaços referentes a estes eventos, afirmando como anormal as distâncias fora deste padrão.

Este trabalho apresenta um esquema complexo com ótimos resultados médios na detecção de anormalidade, porém demonstra uma alta variação de precisão de acordo com a alteração da frequência cardíaca, com resultados de falso positivos baixos para determinados pacientes, porém altos para os demais. Além destes fatores, este trabalho relaciona somente duas classes de anomalias, classificando os pacientes com batimentos cardíacos normais e anormais, sem especificar qual a classe da anormalidade.

É apresentado em Xu et al. 2018 um sistema capaz realizar a detecção de arritmias e a análise de dados em larga escala, chamado ECGLens, integrando um sistema de classificação de batidas do coração baseado em redes neurais convolucionais. Para este modelo, foram considerados quatro diferentes anomalias, Ritmo sinusal, batimento ventricular

premature, Atividade Ectópica Supraventricular e fusão sinusal e ventricular prematura. No processo de detecção de anomalias, este sistema implementa um modelo automático de série temporal baseada em modelos multi escala de redes neurais convolucionais. O processo é realizado em três fases, primeiramente utiliza-se técnicas de transformações espectrais e redução por amostragem com o intuito de minimizar os ruídos e melhorar as entradas. Após este processo, os dados são submetidos a um algoritmo convolucional de três camadas seguido por uma camada com a função de ativação ReLU e outra maxpooling. Em um último estágio, o dado é concatenado, e ocorre o envio para duas camadas convolucionais, cada uma seguida por uma outra camada ReLU e outra maxpooling, uma camada totalmente conectada e uma camada probabilística softmax.

Este trabalho propõe um sistema complexo de análise de anomalias em uma onda ECG, porém como resultado, apresenta uma acurácia intermediária de detecção dos batimentos cardíacos. Observa-se também que esta proposta analisa somente quatro diferentes anormalidades.

Tabela 1. Comparação das análises realizadas nos trabalhos relacionados.

Trabalho	Quantidade de Anomalias detectadas	Custo Computacional	Completude da Base de Teste	Numero de atributos
[Kiranyaz et al. 2016]	2	Alto	Parcial	Parte da Onda
[Oresko et al. 2010]	4	Baixo	Completa	6
[Carrera et al. 2019]	2	Baixo	Completa	Parte da Onda
[Xu et al. 2018]	4	Alto	Completa	4
Este trabalho	18	Baixo	Completa	10

Os resumos das principais características de cada trabalho apresentado nesta seção pode ser observado através da Tabela 1, onde destaca-se a quantidade de anomalias detectadas em cada trabalho, seu custo computacional, a completude do trabalho em analisar a base de dados e o número de atributos considerados para a classificação.

3. Proposta

Para realizar a classificação das anomalias em um ECG, é necessário a construção de um modelo de análise da onda que pode ser divididos em três principais processos. Primeiramente os dados recebidos em forma de onda devem ser tratados através de filtros capazes remover artefatos indesejados, como ruídos elétricos ou musculares.

Em um segundo passo, as informações úteis presentes na onda devem ser extraídas através de processos de detecção destes pontos de importância. Por último, as informações extraídas são utilizadas para diferenciar e detectar as diferentes anomalias existentes no exame dos pacientes.

3.1. Filtragem da Onda e Detecção dos Pontos de Importância

A fim de analisar o ECG para detecção dos pontos de importância deve-se inicialmente realizar a filtragem dos dados recolhidos, uma vez que ruídos são facilmente introduzidos em um ECG através de circuitos de alta impedância de entrada e com um alto ganho para detecção de ondas de baixas voltagens produzidas pelo coração [Metshein et al. 2017]. Temos também o ruído introduzido através da rede elétrica local, da atividade muscular e da respiração que é uma das principais causas da variação da linha de base no ECG.

Em Moreira et al. apresenta-se os processos necessários para filtragem dos artefatos indesejados da onda ECG, utilizando para isto, filtros com base em Transformadas

Discretas de Wavelet, que preparam a onda para, na etapa seguinte, realizar a extração dos segmentos de importância.

Com o sinal de ECG já filtrado, os blocos de interesse são então gerados, sendo compostos pelo complexo QRS e uma onda T. Utiliza-se os blocos QRS devido sua grande importância para detecção de doenças cardíacas. Para geração destes blocos, utilizou-se o esquema de médias móveis de acordo com Elgendi (2016). Este esquema utiliza dois períodos de tempo: W_1 , a duração aproximada do evento de interesse, e W_2 , a duração aproximada do ciclo completo de um batimento cardíaco. As médias móveis MA_{evento} e MA_{ciclo} são então computadas através dos tempos W_1 e W_2 utilizando-se a Equação 1 e Equação 2, tendo n como índice de cada ponto dos N pontos de dados e $y[n]$ o valor atual.

$$MA_{evento}[n] = \frac{1}{W_1} \left(y\left[n - \frac{W_1 - 1}{2}\right] + y\left[n - \frac{W_1 - 3}{2}\right] + \dots + y\left[n + \frac{W_1 - 3}{2}\right] + y\left[n + \frac{W_1 - 1}{2}\right] \right) \quad (1)$$

$$MA_{ciclo}[n] = \frac{1}{W_2} \left(y\left[n - \frac{W_2 - 1}{2}\right] + y\left[n - \frac{W_2 - 3}{2}\right] + \dots + y\left[n + \frac{W_2 - 3}{2}\right] + y\left[n + \frac{W_2 - 1}{2}\right] \right) \quad (2)$$

Todas as regiões que $y[n]$ está contido em $MA_{evento} \geq MA_{ciclo}$ são então consideradas como pontos de interesse, sendo calculadas através dos vetores de médias móveis. Os valores de limiar MA_{evento} e MA_{ciclo} podem ser ajustados de acordo com quão ruidoso encontra-se o sinal através da adição de um deslocamento, normalmente, proporcional à média do sinal. Os melhores valores de períodos referentes aos complexos QRS, para detecção dos blocos QRS, foram calculados de acordo com a base de dados MIT-BIH. Para cada valor, calcula-se a média móvel MA_{QRS} do evento e a média móvel $MA_{batimento}$ do ciclo, seguindo a Equação 1 e Equação 2. Em alguns casos, outros eventos, como as ondas P e T, podem ser detectados junto com o complexo QRS, para evitar estes erros, regiões abaixo dos valores W_1 ms são descartadas.

Com a detecção dos complexos QRS concluída, procura-se então detectar os picos Q, R e S. Tendo como maior valor do vetor do complexo, atribuído ao pico R, sendo o pico com maior amplitude do sinal. Os valores máximos da onda à esquerda e direita a partir do pico R, em um sinal quadrático, são atribuídas aos picos Q e R respectivamente. Uma vez já detectado o complexo QRS, este sinal é então removido da onda original, zerando os seus valores correspondentes no vetor, para prosseguir com a detecção dos pontos fiduciais P e T. Após isto, a onda contém apenas os sinais ainda não identificados. Um resultado típico pode ser visto na Figura 1.

Foram também realizados testes para determinar os melhores valores para os intervalos W_1 e W_2 da onda T, levando em considerando a base de dados MIT-BIH. A média móvel relacionada ao evento MA_T e a média móvel relacionada ao ciclo das ondas $MA_{batimento}$ são então calculadas para o novo evento, aplicando a Equação 1 e Equação 2 para o sinal sem o complexo QRS. Remove-se então a onda T do sinal, deixando apenas a onda P e alguns ruídos. O maior valor à esquerda do pico R é considerado o pico P.

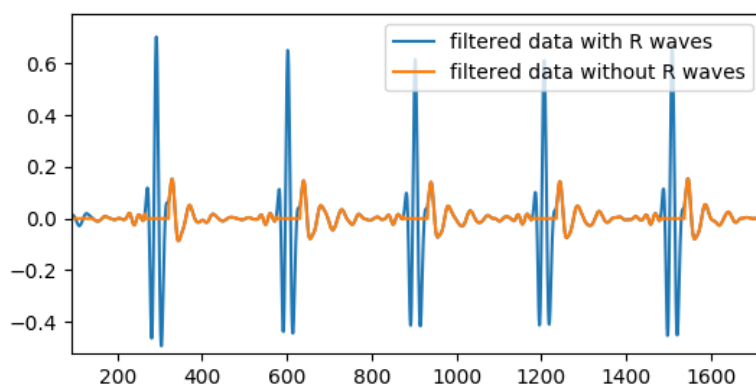


Figura 1. Sinal após remoção do complexo QRS.

Como passo de melhoria realiza-se uma busca dos picos reais das ondas P e T, entre os pontos existentes com base no espaço de ocorrência de cada onda, pelos valores regulares das distâncias entre as ondas PR e entre as ondas RT em um ECG, descartando possíveis valores classificados erroneamente. Em um alcance dentro dessa distância, podemos definir quais pontos são realmente reais, considerando pontos que estão fora deste intervalo como pontos de erro. Realiza-se esta busca somente a partir dos picos R, pois as ondas P e T podem ocorrer em pontos que não são detectados uma onda R, em ondas anômalas.

Como resultado desta fase, produz-se um sinal com todos os picos de importância detectados, visto na Figura 2, que serão então utilizados na fase de classificação. Estes pontos serão então agrupados para então serem utilizados pelos algoritmos de classificação. Cada parte da onda representa um período médio dos batimentos do paciente, possibilitando um diagnóstico segmentado.

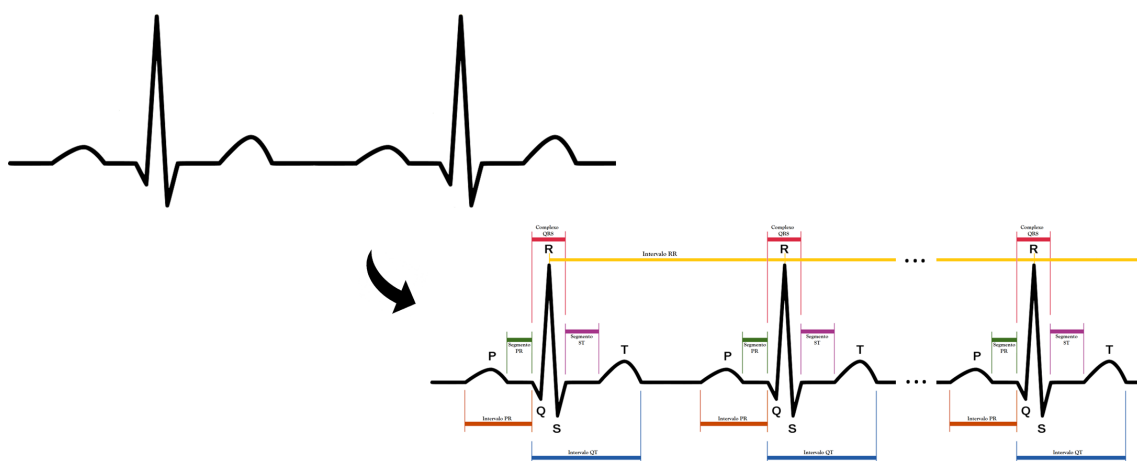


Figura 2. Onda com pontos fiduciais detectados.

A próxima fase, após a detecção dos pontos fiduciais, é a utilização destes pontos para formação dos blocos de interesse.

3.2. Blocos de interesse

Para classificar a onda, definiu-se um sistema de blocos. Cada bloco é formado por um conjunto de pontos de importância, referentes a um período da onda. Entre os pontos de

importância recebidos da etapa anterior, têm-se os pontos P, Q, R, S e T. Cada um destes pontos têm informações da amplitude da onda. A partir dos pontos fiduciais, também são formados segmentos de importância, que representam a distância em amostras entre os pontos. Estes segmentos são definidos segundo os principais segmentos cardíacos existentes, segmento PR, RR, QT, ST e QRS. Desta forma cada bloco pode ser representado por um período que compreende os pontos fiduciais e os segmentos cardíacos entre uma onda P à T, como pode ser observado na Figura 3.

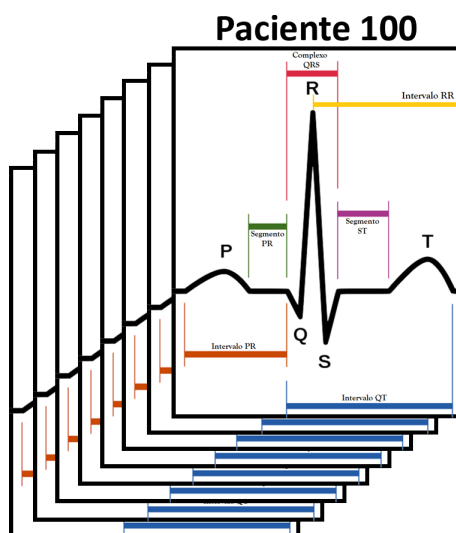


Figura 3. Blocos formados pelos períodos e segmentos compreendidos entre uma onda P à T

O ECG pode ser representado por um conjunto de blocos de interesse. Cada bloco pode ser analisado separadamente ou em conjunto. Esta abordagem possibilita, agora, uma análise não somente da onda completa de ECG, mas como frações de onda. Este processo facilita, principalmente, na aquisição de dados, visto que registros com menos tempo de duração podem ser incorporados ao sistema facilmente. Como também auxilia o diagnóstico em um menor período de tempo, visto que remove a necessidade de uma onda completa para a análise, bastando assim apenas um único bloco do paciente para iniciar o processo de classificação.

Como um último passo, estes blocos são então enviados para as máquinas de aprendizagem, onde realiza-se a classificação, baseando-se nos atributos contidos no bloco recebido.

3.3. Classificação de Anomalias

Para realizar a classificação das possíveis anomalias em um bloco, serão utilizados algoritmos de aprendizagem de máquinas, Árvores de Decisão, Florestas Aleatórias, SVM e métodos de aprendizagem profunda, como Perceptron multicamadas. As doenças são classificadas em relação aos 10 atributos de entrada relacionados a formação do bloco, picos P, Q, R, S, T além dos segmentos e intervalos PR, RR, QT, ST e QRS, todos estes foram propostos por este trabalho. Como entrada para esta fase, pode-se receber, já pré-processado, um bloco isolado de ECG, o conjunto de blocos de um determinado paciente ou um conjunto de clusters formado por blocos agrupados de diferentes pacientes. Estes conjuntos de dados constituem a entrada para os algoritmos de aprendizagem, Figura 4.

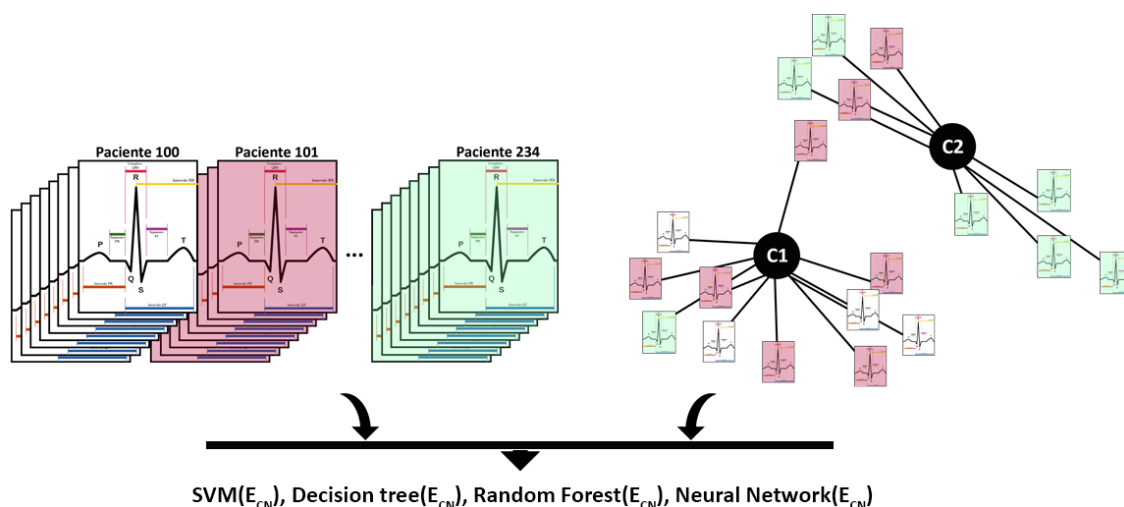


Figura 4. Classificação dos blocos ECG pelos algoritmos de aprendizagem de máquina.

No caso de receber somente um bloco, o objetivo do sistema é classificá-lo como um trecho saudável ou com anomalias, sendo utilizado para análises pontuais ou para reanálise de trechos específicos, este trecho é então re-integrado ao seu contexto para análise do especialista caso necessário.

Para um conjunto de blocos de um único paciente, classificam-se as anomalias deste paciente e se lista todos os blocos com irregularidades, desta forma, cria-se uma máquina especialista nos sinais e problemas do paciente, com isto, pode-se obter ótimos resultados em um exame contínuo de observação. Este cenário é indicado para pacientes que necessitam de uma longa observação pois longos trechos de ECG são difíceis de observar, pelo método proposto, este trecho é facilmente classificado, gerando um retorno menor para a observação do especialista.

Para entradas de um conjunto de blocos de diferentes pacientes, gera-se uma máquina com objetivos amplos, de forma que pode-se ter um melhor resultado na classificação de doenças em geral. Este cenário é indicado para detecção de novas anomalias em diferentes pacientes com uma grande variedade e quantidade de sinais.

Para os dois cenários apresentados e propostos por este trabalho, ambas as máquinas geram resultados capazes de detectar e informar especificamente as anomalias, ou seja, qual a doença está atrelada ao sinal analisado. No próximo capítulo serão apresentados os resultados obtidos através da aplicação da proposta aqui apresentada. Primeiramente verificar o processo de criação e implementação do sistema e depois os resultados quantitativos em comparação com outros trabalhos da área.

4. Avaliação de Desempenho

Esta seção descreve os experimentos realizados para avaliar o sistema e-Health proposto em relação a análise de dados e detecção de problemas cardíacos. A Seção 4.1 apresenta a metodologia aplicada durante a avaliação da proposta, enquanto que a Seção 4.2 analisa os resultados numéricos dos experimentos realizados.

4.1. Metodologia

O desempenho do sistema proposto foi avaliado utilizando o banco de dados real de arritmias *MIT-BIH*, e comparado com o sistema apresentado na referência [Oresko et al. 2010]. O MIT-BIH é composto por quarenta e oito trechos de gravações de ECG, com meia hora cada, de dois canais. Os registros foram obtidos de 47 indivíduos estudados pelo Laboratório de Arritmias do Hospital Beth Israel, em Boston, entre 1975 e 1979 que foram coletados de uma população mista de pacientes internados (cerca de 60%) e ambulatoriais (cerca de 40%) no Hospital Beth Israel e digitalizados em 360 amostras por segundo em cada canal e com resolução de 11 bits em um intervalo de dez (10) mV. Dois ou mais cardiologistas anotaram independentemente os registros para cada batimento cardíaco (aproximadamente 110.000 anotações) incluídos no banco de dados, onde discordâncias foram resolvidas para obter legibilidade.

Durante os experimentos foram analisados todos os trechos de ECG disponíveis no MIT-BIH com as seguintes métricas de avaliação: Acurácia (percentual de acertos em relação ao montante de dados) e tempo de análise da técnica escolhida. Da mesma forma, foram avaliadas as técnicas de aprendizado de máquina Árvore de Decisão, Florestas Aleatórias e SVM implementadas com as bibliotecas python de aprendizagem de máquina scikit-learn [Pedregosa et al. 2011] e MLP implementada utilizando a Biblioteca TensorFlow [Abadi et al. 2015].

Para classificação dos sinais levou-se em conta 19 diferentes classes de eventos cardíacos, com 18 anomalias [Moody and Mark 2001]: Batimento Normal, Bloqueio do ramo esquerdo do feixe de His, Bloqueio do ramo direito do feixe de His, Batimento Prematuro Atrial, Contração Ventricular Prematura, Fusão de batimentos ventriculares e normais, Batimentos Nodais Prematuros, Batimentos atriais prematuros, Batimento supraventricular prematuro ou ectópico, Batimento de escape ventricular, Batimento de escape nodal, Batimento estimulado, Bloqueio atrioventricular do primeiro grau à esquerda ou direita, Onda de fibrilação ventricular, Batimento de escape atrial, Batimento de escape Supraventricular, Onda P não conduzida (APB com clock), Fusão de batimento ritmado e normal, Contração ventricular prematura R-em-T.

4.2. Resultados

Para avaliação da classificação dos pacientes em relação a anomalias cardíacas, recebeu-se como entrada os blocos de importância do sinal ECG. A partir disso, cada bloco é classificado utilizando os algoritmos de árvores de decisão, floresta aleatória, SVM e MLP. Estes blocos foram avaliados em dois principais cenários.

O primeiro cenário realiza a classificação de um conjunto de blocos de um único paciente, ou seja, todos os blocos de um deste paciente são recebidos e classificados de acordo com suas anomalias. Em segundo cenário os blocos de entrada fazem parte de um conjunto de blocos de diferentes pacientes.

Para o treinamento dos algoritmos de classificação utilizou-se a técnica de Validação Cruzada (Cross-validation) [Wong 2015], onde 67% dos dados são utilizados para o treinamento dos algoritmos e os 33% restantes dos dados como entrada para os testes de classificação. Diferentes configurações foram utilizadas para múltiplos testes dos algoritmos de aprendizagem, uma média de 33 execuções de cada algoritmo foi tomada. A

combinação das seguintes configurações, para cada um dos algoritmos, pode ser observada na Tabela 2

Tabela 2. Hiperparâmetros dos Algoritmos de Aprendizagem.

Algoritmo de Aprendizagem	Hiperparâmetros	Valores	Valores com Melhores Resultados
Árvores de Decisão	Critério	Gini, Entropy	Entropy
	Splitter	Best, Random	Best
	Min. Samples Split	$\frac{mss}{10}$; $mss \in [1, 10]$; $mss \in \mathbb{N}$	0.1
	Min. Samples Leaf	$\frac{mss}{10}$; $mss \in [1, 5]$; $mss \in \mathbb{N}$	0.1
	Nº Max. recursos	1,2,3,4,5,6,7,8,9,10, auto, sqrt, log2, None	8
Floresta Aleatória	Critério	Gini, Entropy	Entropy
	Splitter	Best, Random	Best
	Min. Samples Split	$\frac{mss}{10}$; $mss \in [1, 10]$; $mss \in \mathbb{N}$	0.7
	Min. Samples Leaf	$\frac{mss}{10}$; $mss \in [1, 5]$; $mss \in \mathbb{N}$	0.4
	Nº Max. recursos	1,2,3,4,5,6,7,8,9,10, auto, sqrt, log2, None	4
	Estimadores	10, 20	20
SVM	C	0.25, 0.50, 0.75, 1	1
	Kernel	linear, poly, sigmoid, rbf	rbf
	Grau	1, 2, 3	-
	coef0	0.25, 0.50, 0.75, 1	0.25
MLP	Nº de Neurônios - Primeira Camada	128, 256, 512, 1024, 2048	1024
	Nº de Neurônios - Segunda Camada	128, 256, 512, 1024, 2048	1024
	Taxa de Aprendizado	0.001	0.001
	Otimizador	Adam [Kingma and Ba 2014], Gradiente Descendente	Adam [Kingma and Ba 2014]

Para árvores de decisão, os hiperparâmetros utilizados foram o critério Gini e Entropia, medida para mensurar a qualidade da divisão dos ramos. Splitter define a estratégia de divisão em cada nó, melhor ou aleatória, ajustando a melhor divisão ou de modo aleatório, sem critério. O mínimo de amostras para divisão define o número mínimo de amostras que são requeridas para dividir um nó interno.

A quantidade mínima de amostras nas folhas define a quantidade de amostras que são requeridas para ser um nó folha. Por último, o número máximo de recursos, é considerado ao procurar a melhor divisão, este valor pode variar em um inteiro de 1 a 10, que considera uma quantidade fixa de valores, SQRT, ou seja, a raiz quadrada da quantidade de características, log2, definindo o número máximo como o \log_2 e nenhum para o número máximo igual à quantidade de características.

Os hiperparâmetros considerados para Floresta Aleatória compreendem o conjunto de parâmetros das árvores de decisão, com a adição do número de estimadores que define a quantidade de árvores em uma floresta, foram utilizadas 10 e 20 árvores.

No SVM, utilizou-se o parâmetro de erro C, definindo um valor para evitar classificar erroneamente cada exemplo de treinamento, através de uma margem no hiperplano gerado. Os kernels utilizados foram o linear, polinomial, sigmoide e rbf. Para definir o grau do kernel polinomial, utiliza-se os de 1º, 2º e 3º grau. Como termo independente nas funções polinomiais e sigmóides, utilizou-se o parâmetro coef0, com os valores 0.25, 0.50, 0.75 e 1.

Nas MLP foram utilizadas duas camadas intermediárias com o número de neurônios igual à combinação dos valores 128, 256, 512, 1024, 2048. A taxa de aprendizado da rede foi definida como 0.001 e os otimizadores Adam e o Gradiente Descendente foram utilizados na avaliação da rede.

Na classificação do primeiro cenário, onde todos os blocos de treinamento e classificação pertencem apenas ao paciente que terá seus blocos classificados, os resultados obtidos podem ser observados na Tabela 3.

Tabela 3. Resultado dos Algoritmos de Aprendizagem para as 19 diferentes classes de eventos cardíacos, com 18 anomalias em um cenário de análise geral.

Algoritmos de Aprendizagem	Acurácia
Árvore de Decisão	0.9530
Florestas Aleatória	0.9502
SVM	0.9818
MLP	0.9878

Este cenário pode ser utilizado, por exemplo, para monitoramento de pacientes separadamente, onde cada paciente tem seu próprio escopo e uma máquina especialista para classificá-lo, obtendo um resultado de até 98% de acurácia na classificação das anomalias cardíacas. Por meio desta classificação, pode-se por exemplo, realizar uma observação prolongada deste paciente e retornar apenas os períodos com maior quantidade de anomalias detectadas, com o objetivo de monitoramento ao invés de classificação de novas doenças. Neste cenário, observa-se que as máquinas de aprendizado tiveram um bom resultado em geral, visto que foram treinadas de forma especialista, com um menor número de classes possíveis, não apresentando um grande desvio de resultado.

Para o segundo cenário, de classificação dos blocos em um conjunto de outros blocos, não necessariamente de um mesmo pacientes, as médias dos resultados obtidos para estas anomalias podem ser observadas na Tabela 4.

Tabela 4. Resultado dos Algoritmos de Aprendizagem para as 19 diferentes classes de eventos cardíacos, com 18 anomalias em um cenário de análise especialista

Algoritmos de Aprendizagem	Acurácia
Árvore de Decisão	0.7122
Florestas Aleatórias	0.7094
SVM	0.8820
MLP	0.9640

Este cenário pode ser utilizado, por exemplo, em um grande conjunto de dados onde todos os pacientes fazem parte de um único escopo. Neste caso, um paciente com doenças ainda não conhecidas pode ser diagnosticado sem a necessidade de uma máquina especialista ou ser usada em conjunto para descobrir novas doenças em pacientes pertencentes ao outro cenário. Constrói-se uma máquina generalista, capaz de detectar novas anomalias em diferentes pacientes. Diferentemente do cenário anterior, neste observa-se uma grande variação dos resultados entre as máquinas utilizadas, mesmo obtendo bons resultados, visto que com o maior número de classes envolvidas, a técnica de MLP apresentou uma melhor generalização. Os melhores resultados foram obtidos através das configurações apresentadas na Tabela 2 na coluna referente aos valores com melhores resultados, para cada um dos algoritmos testados anteriormente.

Apesar das diferentes abordagens, em ambos os casos os blocos dos pacientes podem ser classificados um a um, provendo um ambiente com resultados imediatos, porém os blocos são disponibilizados em faixas de exame para que o especialista possa avaliá-los.

Para realizar a comparação dos resultados obtidos neste trabalho com os resultados

de Oresko et al. e Xu et al., também utilizou-se os algoritmos de classificação em um escopo de apenas quatro doenças. Onde Oresko levou em consideração as anomalias de Bloqueio do ramo direito do feixe de His, Contração ventricular prematura, ritmos estimulados e fusão de ritmos estimulados com batimentos normais. E Xu as anomalias Ritmo sinusal, batimento ventricular prematuro, Atividade ectópica Supraventricular e fusão sinusal e ventricular prematura. Estes resultados são descritos na Tabela 5 e levam em consideração um cenário composto por seqüências de blocos de diferentes pacientes.

Tabela 5. Resultado dos Algoritmos de Aprendizagem para as 4 diferentes anomalias por bloco

Algoritmos de Aprendizagem	Acurácia
Árvore de Decisão	0.7927
Florestas Aleatórias	0.7886
SVM	0.9218
MLP (Adam)	0.9825
[Oresko et al. 2010] MLP	0.93
[Xu et al. 2018] CNN	0.91

Pode-se observar um ganho de até 7% na classificação das anomalias, se comparado ao cenário de classificação com blocos de diferentes pacientes. Estes resultados devem-se a estratégia de divisão dos blocos de interesse e a utilização do otimizador adaptativo Adam [Kingma and Ba 2014] para aprendizagem profunda através da plataforma para aprendizagem profunda TensorFlow.

Em comparação com os demais trabalhos relatados na literatura, obteve-se um ótimo resultado, levando em consideração o método de MLP. Em comparação com Oresko et al., tem-se um ganho de até 5% na detecção de anomalias, que corresponde a uma melhora média a 232 batimentos classificados corretamente por hora em um único paciente. Quanto a Xu et al. a melhora na classificação das anomalias é de 7%, até 323 batimentos por hora em um único paciente, esta melhoria deve-se principalmente à estratégia de blocos proposta neste trabalho, utilizando-se 10 atributos diferentes para a detecção.

Este trabalho demonstra o alto grau de precisão deste sistema, se comparado até com sistemas com apenas duas classes de classificação, como Kiranyaz et al. e Carrera et al., que relatam uma acurácia máxima de 97,68%. Sabe-se que para o resultado obtido, a parcela negativa foi influenciada pela quantidade de amostras relacionadas a certas doenças que estavam abaixo do necessário para o treinamento da máquina utilizada. Alguns outros casos demonstram erros em relação a proximidade dos sinais saudáveis com os que contém anomalias. Porém, devido ao baixo valor de erro, o sistema é totalmente utilizável, já que os alertas gerados para análise médica em sua grande maioria é correto e com ótima precisão.

Em relação ao tempo decorrido para classificação das anomalias dos pacientes, no pior cenário, com a classificação de todos os pacientes, em média 650,000 amostras e 2314 complexos QRS e utilizando o algoritmo com maior performance de classificação, uma rede MLP, tem-se um resultado em apenas 2.8 segundos para classificação de todos os blocos existentes. Este resultado demonstra a capacidade de tal sistema ser utilizado em situações de tempo real, visto que para gerar 2314 blocos, um ser humano leva em média 33 minutos.

Os resultados foram obtidos tanto para a classificação dos pontos fiduciais de um ECG quanto a classificação destes pontos a partir de uma estratégia de separação por bloco. Cada bloco foi classificado em um conjunto de blocos do mesmo paciente ou de pacientes diferentes, apresentando assim as qualidades de cada abordagem. Também observou-se o tempo necessário para que o método seja utilizado em sistemas de tempo real. No próximo capítulo, apresenta-se as conclusões do trabalho e os trabalhos futuros.

5. Conclusão e trabalhos futuros

Devido ao grande número de mortes por doenças cardíacas, tornou-se importante ter uma preocupação extra para detectar essas doenças. A maneira mais comum é pelo eletrocardiograma (ECG) dos pacientes e de maneira visual. Para isso, é crucial detectar os pontos de importância de um ECG e classificar estes pontos em relação as possíveis anomalias.

Neste trabalho, propõe-se um sistema e-health responsável por detectar características de importância em um ECG e formar uma representação que é então utilizada para classificação de anomalias cardíacas por algoritmos de aprendizagem de máquina. A proposta foi avaliada utilizando a base de dados MIT-BIH que contém trechos de exames ECG reais. A proposta teve um bom desempenho tanto na detecção dos pontos de importância quanto à classificação das anomalias em relação a outros trabalhos importantes nesse domínio.

Como trabalhos futuros, pretende-se estender o sistema proposto para que este seja capaz de, a partir da estrutura base de análise e detecção de doenças apresentada, considerar outros exames (e conseqüentemente alguns outros dados biológicos) e detectar diversos tipos de doenças. Adicionalmente, pretende-se adicionar aspectos de big data, a fim de dar suporte a um grande volume de dados biológicos de diversos pacientes.

Agradecimentos

Os autores gostariam de agradecer a FUNCAP, CAPES e CNPq pelo apoio financeiro.

6. References

Referências

- Abadi, M., Agarwal, A., Barham, P., et al. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.
- 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.
- Begg, R., Lai, D. T., and Palaniswami, M. (2007). *Computational intelligence in biomedical engineering*. CRC Press.
- Carrera, D., Rossi, B., Fragneto, P., and Boracchi, G. (2019). Online anomaly detection for long-term ecg monitoring using wearable devices. *Pattern Recognition*, 88:482–492.
- Elgendi, M. (2016). Terma framework for biomedical signal analysis: An economic-inspired approach. *Biosensors*, 6(4):55.

- Hamilton, P. S. and Tompkins, W. J. (1986). Quantitative investigation of qrs detection rules using the mit/bih arrhythmia database. *IEEE transactions on biomedical engineering*, (12):1157–1165.
- Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Kiranyaz, S., Ince, T., and Gabbouj, M. (2016). Real-time patient-specific ecg classification by 1-d convolutional neural networks. *IEEE Transactions on Biomedical Engineering*, 63(3):664–675.
- Mendis, S., Puska, P., Norrving, B., Organization, W. H., et al. (2011). *Global atlas on cardiovascular disease prevention and control*. Geneva: World Health Organization.
- Metshein, M., Annus, P., Land, R., Krivošei, A., Ojarand, J., Aabloo, A., and Min, M. (2017). Variation of cardiac and respiratory waveform on human thorax in the case of inductive coupling. In *EMBECE & NBC 2017*, pages 671–674. Springer.
- Moody, G. B. and Mark, R. G. (2001). The impact of the MIT-BIH arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine*, 20(3):45–50.
- Moreira, D. A., Chaves, L. G., Lima, B. A., Almeida, T. P., Gomes, R. L., Celestino, J., and Neto, A. J. (2018). Detecção de pontos fiduciais em eletrocardiogramas para sistemas de saúde. *II Workshop de Computação Urbana (CoUrb), 2018, Campos do Jordão. Simposio Brasileiro de Redes de Computadores e Sistemas Distribuidos (SBRC)*.
- Oresko, J. J., Jin, Z., Cheng, J., Huang, S., Sun, Y., Duschl, H., and Cheng, A. C. (2010). A wearable smartphone-based platform for real-time cardiovascular disease detection via electrocardiogram processing. *IEEE Transactions on Information Technology in Biomedicine*, 14(3):734–740.
- Organization, W. H. et al. (2016). Hearts: technical package for cardiovascular disease management in primary health care.
- 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.
- Wong, T.-T. (2015). Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. *Pattern Recognition*, 48(9):2839–2846.
- Xu, K., Guo, S., Cao, N., Gotz, D., Xu, A., Qu, H., Yao, Z., and Chen, Y. (2018). Ecglens: Interactive visual exploration of large scale ecg data for arrhythmia detection. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 663. ACM.
- Zhao, R., Wang, D., Yan, R., Mao, K., Shen, F., and Wang, J. (2018). Machine health monitoring using local feature-based gated recurrent unit networks. *IEEE Transactions on Industrial Electronics*, 65(2):1539–1548.

Energesis: *Fog Smart Meter* para Hospedagem Compartilhada

Rafael Sampaio Pereira¹, José Souza de Jesus², Matheus Thiago Marques Barbosa¹, Eric Bernardes Chagas Barros¹, Maycon Leone Maciel Peixoto¹

¹Programa de Pós-Graduação em Ciência da Computação (PGCOMP)
Instituto de Matemática – Universidade Federal da Bahia (UFBA)
Salvador – BA – Brasil

²Instituto Federal de Educação Ciência e Tecnologia da Bahia (IFBA)
Jequié – BA – Brasil

{rafaspereira, matheus.thiago, eric.bernardes}@ufba.br,

jose.jesus@ifba.edu.br, mayconleone@dcc.ufba.br

Abstract. *Some homeowners host travelers in their own homes, sharing some rooms for a short period. Along with this practice, there is a need for a resource that allows to measure the energy consumption when the property is being shared. In this way, the guest should only pay for the energy consumed by him. The difficulty of delivering this consumption report is because the houses have only a general measuring system. Thus, this article shows the development of a system for monitoring the consumption of electricity for shared hosting. The system was developed under the Internet of Things (IoT) architecture using Fog and Cloud Computing. The System is composed of a Smart Meter, and a Web application. In addition to the construction of the intelligent measurement architecture, the performance evaluation was performed with the purpose of showing the system's ability to measure electricity consumption and be used as a resource for owners in shared hosting.*

Resumo. *Alguns proprietários de imóveis hospedam viajantes em suas próprias residências, compartilhando apenas parte do imóvel por um curto período. Junto com essa prática, existe a necessidade de um recurso que permita mensurar o consumo de energia quando o imóvel está sendo compartilhado. Dessa forma, o hóspede pagará apenas a energia consumida por ele. A dificuldade de entregar esse relatório de consumo é porque as casas possuem apenas um sistema geral de medição. Dessa forma, este artigo mostra o desenvolvimento de um sistema para monitoramento do consumo de energia elétrica para a hospedagem compartilhada. O sistema foi desenvolvido sob a arquitetura IoT (Internet of Things) utilizando a Fog e Cloud Computing. O Sistema é composto por um Smart Meter, e uma aplicação Web. Além da construção da arquitetura de medição inteligente, foi realizada a avaliação de desempenho com o objetivo de mostrar a capacidade do sistema de medir consumo de energia elétrica e ser usado como um recurso para proprietários em hospedagem compartilhada.*

1. Introdução

A economia compartilhada, chamada também de economia de compartilhamento, é uma tendência, de nível global, onde pessoas compartilham bens com outras que

queiram pagar pela experiência do uso temporário e não queiram adquirir esses bens [Ferreira et al. 2016]. No Brasil hospedagem (e.g., **Airbnb**) e transporte (e.g., **99pop**, **Uber**) são os setores onde a economia de compartilhamento já está bem difundida, em expansão também em outros setores e, em médio e longo prazo, pode chegar a corresponder até 30% do Produto Interno Bruto (PIB) brasileiro, referente ao segmento de prestação de serviços [O Globo 2017]. Isso demonstra a dimensão e a força que esta forma de economia vem conquistando, e a rapidez com que se propaga, devido ao fato de ser mantida pela ideia de um consumo colaborativo.

A *Internet* dispõe de opções para compartilhamentos [Goidanich 2016]. As plataformas, isto é, sites e aplicativos, são os mediadores no compartilhamento. “A economia compartilhada é um novo fenômeno [...] cujo foco é o compartilhamento de bens ociosos, como carro ou apartamento, através de aplicativos de celulares e computadores.” [Gurgel 2017, p.8]. Hospedagem compartilhada caracteriza a prática de locar cômodos ou partes de imóveis residenciais, bem como todo o imóvel. Aplicativos apoiam, e até mesmo intermedeiam os contratos entre proprietário e hóspede. Como afirmam [Soares et al. 2017, p.4] “[...] existe um crescimento nas maneiras inovadoras da economia. O turismo observa [...] como a influência desse modo não convencional cresce no segmento através da hospedagem compartilhada [...]”.

A hospedagem compartilhada pode durar menos de 27 dias, período mínimo entre as leituras realizadas pelas concessionárias de energia elétrica para aferir o consumo realizado em um imóvel [ANEEL 2016]. Em algumas situações, a hospedagem acontece de forma parcial, isto é, o hóspede não é responsável por todo o imóvel, apenas por uma parte dele. O ideal é que o hóspede pague pelo consumo de energia elétrica que corresponde apenas à parte do imóvel que ocupou, e que foi realizado durante o período que ocupou, ou seja, o pagamento deve ser proporcional ao consumo realizado.

Quando a hospedagem compartilhada se aplica a todo o imóvel e em período equivalente ao período da leitura feita pela concessionária de energia elétrica, é possível atribuir ao hóspede a fatura emitida pela concessionária, caso contrário não será possível determinar o consumo realizado pelo hóspede, baseando-se apenas na fatura de energia. Para determinar precisamente o consumo elétrico realizado pelo hóspede durante a hospedagem compartilhada, e o valor a ser pago por ele de maneira proporcional ao consumo, faz-se necessário um mecanismo para mensurar consumo elétrico em cômodos específicos por tempo determinado.

Esse mecanismo precisa ser transparente, permitindo ao hóspede o acompanhamento em tempo real do seu consumo. Contudo, considerando a grande quantidade de dados que podem estar sendo transmitidos, a velocidade de conexão do hospedeiro e a quantidade de dados que o serviço pode estar recebendo, esse tipo de acompanhamento pode não ser possível. Assim, é necessário utilizar uma arquitetura onde é possível realizar cálculos em tempo real. Nesse sentido, o conceito de névoa tem sido bastante usado pela comunidade acadêmica, onde procedimentos que antes precisariam ir até a Nuvem(*Cloud*) para serem realizados passaram a serem feitos na borda da rede.

Este artigo descreve o desenvolvimento de um sistema distribuído, chamado de *Energesis*, que atua como medidor de consumo elétrico. Esse sistema pode ser utilizado como ferramenta de apoio ao proprietário ou administrador de um imóvel, pois é capaz

de coletar e armazenar dados do consumo elétrico realizado pelo hóspede durante hospedagem compartilhada. O proprietário desse imóvel pode gerar uma fatura detalhada com dados do consumo e o valor referente ao consumo total do hóspede.

O Energesis foi desenvolvido sob uma arquitetura de 3 camadas: Sensoreamento, *Fog* e *Cloud Computing*, como meio de oferecer agilidade, disponibilidade e escalabilidade. O sistema é composto por um medidor inteligente (*Smart Meter*), uma API *Rest-Full*, uma aplicação *web* e utiliza persistência poliglota. A *internet* é uma rede sem sincronia, onde latência e tempo de resposta são variáveis de difícil controle. O *Smart Meter* se comunica através da rede local com um web service que está na Névoa(*Fog*), evitando que os dados de consumo sejam enviados diretamente para *Cloud* pela *internet*. Na *Cloud* há uma aplicação *web*, onde é possível cadastrar e atualizar dados do imóvel, tarifas e hóspedes.

Para verificar a capacidade do sistema de medir consumo elétrico e a viabilidade de adotá-lo como um recurso aos proprietários que compartilham imóveis, foi realizada uma avaliação de desempenho, onde foram observados o valor de corrente elétrica identificado pelo medidor e o tempo gasto para determinar esse valor. A avaliação foi feita no algoritmo de medição embarcado no *Smart Meter* do sistema e no algoritmo de medição disponibilizado como parte do projeto *open source* OpenEnergyMonitor. Essa avaliação permite afirmar que o algoritmo proposto nesse trabalho possui tempo de execução menor em relação ao algoritmo tomado como *base-line*. Os valores de corrente elétrica identificado por ambos os algoritmos são próximos.

Neste artigo são detalhadas técnicas, conceitos e materiais utilizados para a implementação do sistema, além da avaliação de desempenho no algoritmo de medição. Este trabalho está dividido em 5 Seções. A Seção 1, introdução, contextualiza o trabalho, explica a problemática e introduz a solução proposta. A Seção 2 trata de trabalhos relacionados. A Seção 3 descreve detalhes da solução proposta. A Seção 4 Avaliação de desempenho do algoritmo de medição, apresenta experimentos realizados para verificar a capacidade do sistema de medir o consumo elétrico. A Seção 5 traz considerações finais acerca do trabalho realizado e a Seção 6 apresenta agradecimentos dos autores.

2. Trabalhos Relacionados

É possível encontrar diversos exemplos de sistemas projetados com o intuito de realizar o monitoramento do consumo elétrico em imóveis. Esses sistemas coletam, armazenam e processam os dados de consumo, além de disponibilizar os dados em aplicações *web*.

O OpenEnergyMonitor¹ é um projeto de *hardware* e *software open-source* fundado pelos americanos Trystan Lea e Glyn Hudson. O projeto tem por objetivo oferecer ferramentas de código aberto que auxiliem as pessoas através da visualização detalhada do consumo, o que pode ajudar nas tomadas de decisões em relação a otimização do consumo, a fim de diminuir eventuais desperdícios [OpenEnergyMonitor 2016]. O *emonPi* é um medidor elétrico do OpenEnergyMonitor baseado na plataforma Raspberry Pi. Ele não monitora apenas o consumo, mas também a produção de energia elétrica, podendo ser útil em imóveis que produzem energia por meio de painéis solares [Advances Wales Magazine 2017].

¹Disponível em <https://openenergymonitor.org>

[de Melo and Roque 2016] apresentam o SGC (Sistema Gerenciador de Consumo), que é uma ferramenta capaz de monitorar o consumo elétrico realizado por equipamentos eletroeletrônicos de uma residência. O sistema foi desenvolvido para auxiliar os usuários em decisões que ajudem a economizar energia. Esse sistema utiliza *hardware* baseado na plataforma Arduino e sensores invasivos para realizar as medições do consumo elétrico, além de uma aplicação *web* onde é possível gerar gráficos do consumo elétrico e estabelecer metas de consumo para um determinado período.

[Jaiswal and Chaubisa 2017] apresentam um medidor de energia inteligente (IEM – *Intelligent Energy Meter*). Solução desenvolvida tendo em vista o cenário do fornecimento de energia elétrica indiano, onde, segundo os autores, as medições podem apresentar erros devido ao fato de haver intervenção humana. Para os autores o uso da mão de obra humana no processo de suspensão do fornecimento de energia elétrica, é um risco a vida. Esse sistema visa auxiliar o consumidor e o fornecedor de energia. Dados de consumo e faturamento são enviados por e-mail e por SMS para os consumidores. O conceito de IoT é utilizado para implementar uma solução para suspensão do fornecimento a distância.

[Yan and Su 2016] apresentam como o aumento do número de *smart meters* tem influenciado a produção de dados e assim apontam a necessidade de se criar arquiteturas que consigam lidar de forma adequada com o grande volume de dados. Com isso, foi proposto a utilização de uma infraestrutura que utiliza a *fog* para armazenamento e processamento de dados que são enviados por *smart meters*.

[Zahoor et al. 2018] propõem uma arquitetura em três camadas utilizando uma infraestrutura em Névoa para processar dados privados coletados de *smart meters* e armazenar os dados públicos na *Cloud*. Esse trabalho mostrou que é possível controlar equipamentos eletrônicos através do sistema centralizado e realizar cálculos de utilização de recursos elétricos.

[Pant et al. 2017] demonstra uma abordagem integrada entre a névoa e o *IoT* e como as *smart grids* utilizando esses recursos podem fornecer uma transparência maior entre o consumidor e o distribuidor de energia. Essa abordagem permite estender os recursos da névoa para o usuário final, introduzindo técnicas de balanceamento de carga e de consumo de energia.

A solução proposta neste artigo permite o monitoramento do consumo elétrico realizado em um imóvel sem a necessidade de intervenção da concessionária de energia. Diferentemente dos trabalhos encontrados, o sistema Energesis possibilita o cruzamento de dados do consumo elétrico com os dados de um hóspede a fim de determinar o valor que este deve pagar com base no consumo que ele realizou. A Tabela 1 apresenta um comparativo entre os recursos do sistema Energesis e os trabalhos encontrados na literatura.

3. O Sistema Energesis

Com o objetivo de auxiliar proprietários e administradores de imóveis, onde se pratique hospedagem compartilhada, foi desenvolvido o sistema Energesis. Trata-se de um sistema distribuído, que tem como entrada dados do consumo elétrico, coletados por sensores, que monitoram a circulação da corrente elétrica nos circuitos elétricos de um determinado imóvel. As medições realizadas pelo sistema não são sobre aparelhos específicos,

Tabela 1. Comparativo entre proposta e trabalhos relacionados.

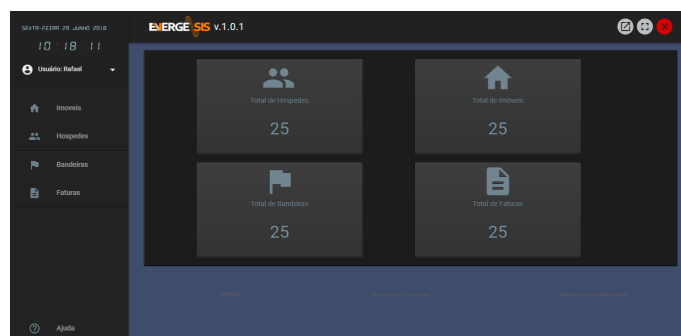
	Monitora Equipamento	Monitora Imóvel	Consumo Compartilhado	Smart Meter	Fog Computing	Acesso Remoto
IEM		✓				✓
emonPI		✓		✓		✓
SGC	✓			✓		✓
Yan and Su 2016		✓		✓	✓	
Zahoor et al. 2018		✓		✓	✓	✓
Pant et al. 2017		✓		✓	✓	
ENERGESIS		✓	✓	✓	✓	✓

tendo em vista que circuito elétrico pode acionar vários aparelhos e um imóvel pode ter vários circuitos. Ao entrarem no sistema os dados são pré-processados e persistidos para que possam ser recuperados e analisados posteriormente. Com os dados armazenados o responsável pelo imóvel pode gerar faturas com informações detalhadas.

O sistema pode ser acessado remotamente por um usuário registrado, por exemplo, o proprietário do imóvel, que doravante será chamado administrador. Desta forma, quando a hospedagem compartilhada englobar todo o imóvel, os responsáveis pelo imóvel poderão emitir um resumo do consumo elétrico ao seu hóspede, sem a necessidade de adentrar ao imóvel que está sob contrato de locação.

O *dashboard* (Figura 1) é a parte da aplicação *web* que permite ao administrador observar dados do consumo, além de cadastrar dados de imóveis, hóspedes, faturas e bandeira tarifária. Para ter acesso a essa parte do sistema, o administrador, precisa estar autenticado utilizando *login* e senha de acesso válidos. As faturas não são cobranças automáticas e não possuem natureza fiscal, sendo apenas listas, onde são detalhados consumo elétrico e o valor a ser pago pelo hóspede. Quando o administrador informar a data, horário de início e termino da hospedagem compartilhada, exibe na tela o valor do consumo elétrico.

No momento do cadastro de faturas, a aplicação *web* calcula o consumo em Kwh realizado pelo hóspede. O calculo toma como base os valores da tensão elétrica do imóvel, da soma das correntes elétricas armazenadas no banco de dados e do tempo total em que houve consumo de energia elétrica durante a hospedagem. Ao aplicar sobre esse valor dados de tarifas de consumo e alíquotas de impostos, o sistema exibirá na tela o valor a pagar em reais (R\$).

**Figura 1. Tela inicial do Dashboard do sistema Energesis.**

3.1. Arquitetura do Sistema

O sistema Energesis é composto por 3 camadas (Figura 2): A camada de sensores que contém um medidor inteligente (*Smart Meter*); a *Fog* que é composta de um Raspberry Pi, que recebe dados do medidor através de uma API *RESTFull*, essa camada responsável por intermediar a transferência de dados entre o medidor e a *Cloud*; e a Nuvem que possui um serviço *web*, um servidor de banco de dados e uma aplicação para visualização dos dados.

A camada mais baixa do sistema utiliza o paradigma de *Internet* das Coisas (IoT), onde ocorre o sensoramento, nela está presente um medidor inteligente que se comunica através do protocolo HTTP e pode ser identificado em rede através de um endereço IP. Esse medidor utiliza um Arduino MEGA 2560 que serve como plataforma para prototipagem de dispositivos eletrônicos [de Souza et al. 2011]. O medidor é responsável por realizar o tratamento dos sinais de consumo elétrico e estabelecer a comunicação com a camada superior, nesse caso, a Névoa (*fog*).

Os sinais de consumo elétrico são coletados por um sensor de corrente alternada que monitora a instalação elétrica de todo o imóvel, ou de um determinado cômodo, através do eletromagnetismo produzido em volta do condutor que está sendo monitorado. Esse projeto utiliza o sensor de corrente TA 12-100, fabricado pela YHDC, capaz de monitorar correntes alternadas de até 5A [YHDC 2015]. Esse sensor não é invasivo, isto é, ele monitora a instalação elétrica do imóvel sem estar eletricamente ligado a ela.

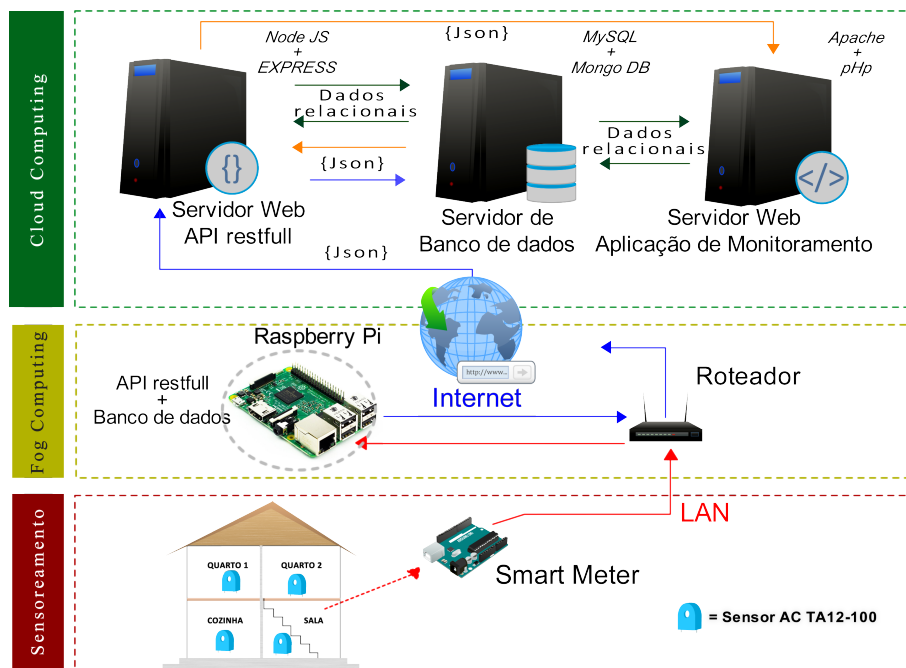


Figura 2. Arquitetura do sistema Energesis.

Os sinais de consumo elétrico coletados pelo sensor são pré-processados pelo sistema embarcado no medidor. Para realizar o tratamento dos sinais o sistema utiliza um algoritmo embarcado que identifica a corrente real (RMS) capturada pelo sensor, ou seja, converte os sinais em valores de corrente elétrica. Esses dados são salvos temporariamente no cartão de memória e depois enviados a um serviço *RESTFull* que está na névoa.

A Fog é camada intermediária, implementada para evitar perda e atrasos na entrega dos dados ao servidor em *Cloud*. A Névoa disponibiliza recursos com baixa latência, distribuição geográfica, reconhecimento de localização, mobilidade, heterogeneidade e capacidade de processamento devido ao alto número de nós [Mukherjee et al. 2018]. Esse modelo já vem sendo adotado em alguns trabalhos relacionados a *Smart Grids* como é o caso de [Okay and Ozdemir 2016, Barros et al. 2018, Kumar et al. 2018]. Assim o medidor envia os dados a um servidor *Web* no Raspberry PI, dessa forma, em um intervalo de tempo programável, os dados são enviados da Fog a *Cloud*. Caso haja atraso ou distúrbios na comunicação os dados são enviados novamente.

Os dados são adaptados para o formato JSON (*JavaScript Object Notation*) e enviados para uma API desenvolvida com a linguagem JavaScript utilizando o Node.js. Os dados de consumo enviados do medidor a Névoa são: corrente elétrica consumida e o instante de tempo, isto é, horário do consumo. Junto com os dados de consumo, o medidor enviará dois dados: um número identificador do imóvel e uma chave de segurança que irão garantir a autenticidade da informação enviada.

Quando a API do servidor na *Fog* recebe os dados e consegue armazená-los sem nenhum erro ou restrição, é enviado um objeto no formato JSON em resposta ao medidor informando se os dados foram armazenados ou não. Quando a resposta recebida for positiva, o medidor apaga os dados temporários. Em caso de resposta negativa ou serviço indisponível, os dados temporários não são apagados e as leituras continuam até que haja uma nova tentativa de envio.

Em caso de erros, indisponibilidade do serviço *web* da *Fog* ou configurações inválidas o medidor exibe, em um *display* LCD, mensagens de alerta. Nesse *display* o usuário é informado acerca último instante em que os dados foram armazenados na *Fog*, ou em caso de erro exibe o instante da ultima tentativa realizada.

O instante em que os dados são recebidos pela *Cloud* não é o instante real em que o consumo foi realizado. Apesar do envio dos dados ocorrer a cada 2 minutos e os dados serem enviados a nevoa antes de serem enviados a *Cloud*, podem ocorrer atrasos na entrega dos dados, uma vez que os dados são enviados através de uma rede assíncrona. O Arduino não possui um relógio e por isso não tem noção do tempo real, apenas conhece o tempo, em milissegundos, decorrido desde a sua ultima inicialização. Para que consumo e tempo sejam associados e armazenados de maneira correta, o sistema utiliza um modulo RTC (*Real-Time Clock*) DS1307 [Maxim Integrated 2015], que permite ao equipamento trabalhar tempo real.

A *Cloud* é a camada do sistema que armazena permanentemente os dados através do paradigma de persistência poliglota, interagindo com um relacional e outro não relacional. Os dados de hóspede, faturam, administrador, bandeira tarifária e imóveis se relacionam entre si, isso explica a presença de um banco de dados relacional. Nessa camada existe uma API que se conecta ao MySQL, solução reacional de licença gratuita, e verifica se o número identificador e a chave de segurança pertencem ao mesmo imóvel. O consumo elétrico é salvo no banco de dados Mongo DB, escolhido pois o formato de documentos aceito por ele(JSON) é o mesmo utilizado pelo *Smart Meter*, desta forma os dados não precisam de tratamentos antes da persistência. Nessa camada existe uma aplicação *web* desenvolvida na linguagem php por onde os dados são visualizados.

3.2. O transformador de corrente TA 12-100

O sistema realiza as medições de consumo elétrico utilizando um transformador de corrente elétrica alternada, que atua como sensor de corrente não invasivo, isto é, ele transforma valores de correntes elétricas através da indução eletromagnética em sinais analógicos de baixa voltagem (Figura 3). Para que o sinal possa ser medido em *volts* é adicionado um resistor de carga em paralelo à bobina do transformador. Esses sinais são enviados a uma entrada analógica do microcontrolador.

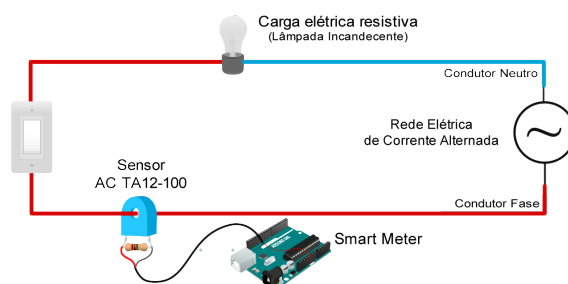


Figura 3. Sensor TA 12-100 atuando em circuito elétrico residencial.

TA 12-100(Figura 4) é um transformador de corrente fabricado pela YHDC, que ao ser utilizado como sensor é capaz de medir correntes de 0A até 5A, e pode ser facilmente substituído por outros transformadores com as mesmas características técnicas. A bobina, isto é, o enrolamento secundário, desses transformadores possui 1000 voltas. O enrolamento primário é o próprio condutor que está sendo monitorado. Para cada 5A que passe pelo primário é disponibilizada uma corrente elétrica de 5mA no secundário. Para compor o resistor de carga foram utilizados resistores em serie com resistência equivalente a 352Ω .

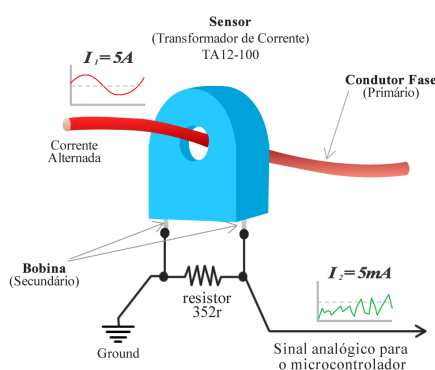


Figura 4. TA 12-100. Características técnicas.

3.3. Algoritmo de medição de corrente elétrica

O sinal analógico enviado pelo sensor de corrente ao microcontrolador precisa ser processado e transformado em valor de corrente para que o consumo possa ser determinado. Nesse trabalho, é proposto um algoritmo desenvolvido em C++, que possibilita ao medidor calcular a corrente elétrica consumida. Para validar a eficácia desse algoritmo foi realizada uma avaliação de desempenho.

3.3.1. A Biblioteca EmonLib

Emonlib é uma biblioteca *open source* disponibilizada como parte do projeto OpenEnergyMonitor (apresentado na Seção 2). Essa biblioteca é composta por um algoritmo dividido em funções que permite dispositivos baseados em Arduino identificar valores de grandezas elétricas, por exemplo, corrente e tensão. A biblioteca não sofreu modificações.

3.3.2. Algoritmo Proposto

O algoritmo proposto para o dispositivo de medição identifica o valor da corrente elétrica convertendo o sinal analógico recebido pelo microcontrolador em voltagem. Para encontrar essa voltagem o algoritmo discretiza o sinal analógico enviado ao ADC do microcontrolador colhendo um determinado número de amostras (Figura 5). O sensor utilizado nesse sistema é unilateral, isto é, monitora apenas um dos sentidos da corrente elétrica alternada, dessa forma são colhidas apenas amostras em um dos ciclos da senoide.

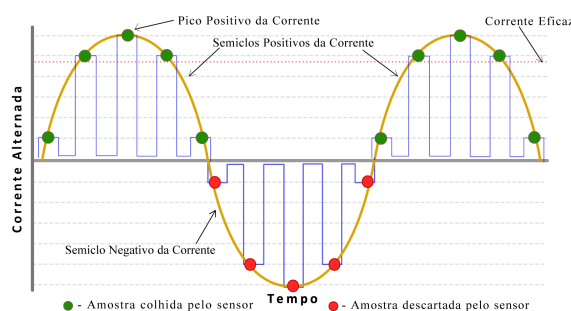


Figura 5. Amostragem realizada pelo algoritmo proposto.

O valor máximo encontrado nas amostras, isto é, o pico de corrente, é o ponto onde o sinal analógico será convertido em *volts*. Conhecido o valor do sinal em *volts*, o algoritmo aplica a regra de eletricidade da primeira lei de *Ohm*. O valor do resistor de carga paralelo ao sensor de corrente é tomado juntamente com o valor em *volts* do sinal analógico para determinar qual a corrente elétrica que passa pelo sensor, utilizando a seguinte formula:

$$V = R \times I,$$

Onde V é o sinal de entrada na porta analógica do microcontrolador, convertido em *volts*, R é o valor do resistor de carga e I é a corrente que passa pelo resistor. O valor de I é multiplicado pelo número de voltas da bobina do transformador. A relação de espiras do transformador é 1000:1, onde há apenas uma volta no primário e mil voltas na bobina do secundário.

$$I = I \times 1000$$

Para encontrar a corrente RMS (*Root Mean Square*), chamada também de corrente eficaz, que foi consumida o algoritmo divide o valor de corrente encontrado pela raiz quadrada de 2, como na formula a seguir:

$$I_{rms} = I / 1.414$$

4. Avaliação de desempenho do algoritmo de medição

O algoritmo de medição de corrente elétrica foi submetido à avaliação de desempenho², onde foram realizados experimentos, para que mediante análise de resultados, seja possível verificar o quão confiáveis são as medições realizadas pelo sistema, além de verificar o tempo gasto pelo algoritmo para adquirir, processar e disponibilizar o valor da corrente elétrica consumida. Dessa forma, o tempo gasto no processamento e o *Throughput* (valor da corrente elétrica na saída do processamento) são as variáveis de resposta a serem analisadas.

4.1. Planejamento de Experimentos

A metodologia utilizada no planejamento dos experimentos realizados foi o Projeto Fatorial 2^k , introduzido por Raj Jain, onde k é a quantidade de fatores e 2 a quantidade de níveis por fator. Esse tipo de planejamento permite determinar qual a influência de um determinado fator sobre as variáveis de respostas que estão sendo observadas [Jain 1996]. Para cada fator foram experimentados dois níveis. Nesse trabalho foram considerados 3 fatores: Algoritmo, Carga Elétrica e Amostra. O projeto fatorial resultou na projeção de 8 experimentos (Tabela 2). Cada experimento foi replicado 10 vezes.

Tabela 2. Matriz de planejamento.

	FATORES		
	A	B	C
EXP.	Amostra	Carga Elétrica	Algoritmo
1	100	Lâmpada	EmonLib
2	1000	Lâmpada	EmonLib
3	100	FerroDeSolda	EmonLib
4	1000	FerroDeSolda	EmonLib
5	100	Lâmpada	Proposta
6	1000	Lâmpada	Proposta
7	100	FerroDeSolda	Proposta
8	1000	FerroDeSolda	Proposta

4.1.1. Amostra

O ADC (Conversor Analógico Digital) do Arduino é capaz de realizar até 10.000 leituras por segundo [Arduino 2018]. Foram escolhidos como níveis 1% (100 amostras) e 10% (1000 amostras) do valor máximo que o Arduino é capaz de ler.

4.1.2. Algoritmo

Será analisado o comportamento de um mesmo sistema utilizando dois algoritmos diferentes: O Primeiro é um algoritmo, que faz parte deste trabalho, chamado de "proposta" durante a avaliação. O segundo é a biblioteca EmonLib, que será utilizada como *base-line* na avaliação de desempenho do algoritmo proposto.

²Video com os experimentos disponível em <https://youtu.be/gSKk8rW39GM>

4.1.3. Carga Elétrica

A carga elétrica acionada durante os experimentos será tomada como fator. Serão acionados uma lâmpada incandescente e um ferro de solda. O fator assume Lâmpada e FerroDeSolda como seus níveis.

4.2. Cenários de execução dos experimentos

Os experimentos foram realizados em dois cenários com tensão elétrica de 127v. No primeiro cenário foi monitorado o consumo de uma lâmpada incandescente com potência de 75w e no segundo cenário, foi monitorado o consumo de um ferro de solda com potência de 25w. As potências dos equipamentos são referências informadas pelos fabricantes. Ambos os cenários utilizaram fio flexível com 1,5 mm de espessura.

Para apurar a precisão das medições realizadas pelo algoritmo embarcado, foi utilizado, simultaneamente, um multímetro em serie e com o circuito (Figura 6), isto é, de forma invasiva ao circuito, enquanto o Smart Meter realizou medições de forma não invasiva. O multímetro utilizado foi o Hikari HM-2044 [Hikari 2013], na escala de 20 Amperes.

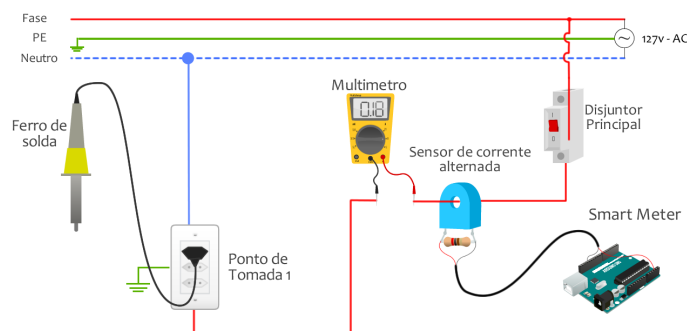


Figura 6. Medição do consumo de um ferro de solda de 25w.

4.3. Análise dos Resultados

O resultado dos experimentos permitem extrair informações sobre o nível de influência que cada fator exerce sobre cada variável de resposta. A avaliação de desempenho não contempla apenas a influencia do algoritmo, e sim a interação do algoritmo com demais fatores e o efeito que essa interação irá produzir na variável de resposta. Os gráficos apresentados na Figura 7 mostram os fatores que mais influenciam nos resultados obtidos em cada variável observada.

O gráfico de efeitos sobre o a variável Tempo Gasto (Figura 7a) mostra que: i) A carga elétrica não influencia no Tempo Gasto; ii) A quantidade de amostras tem maior impacto sobre o Tempo Gasto, ou seja, quanto mais amostras forem colhidas para identificar o valor da corrente elétrica, mais tempo o algoritmo gastará no processamento de dados. 3) O algoritmo Proposto executa, isto é, processa os dados de consumo, mais rápido que o algoritmo *base-line*, a biblioteca EmonLib.

No gráfico de efeitos sobre o a variável *Throughput*(Figura 7b) percebe-se que: 1) A quantidade de amostras não influenciam no *Throughput*. O algoritmo proposto colhe

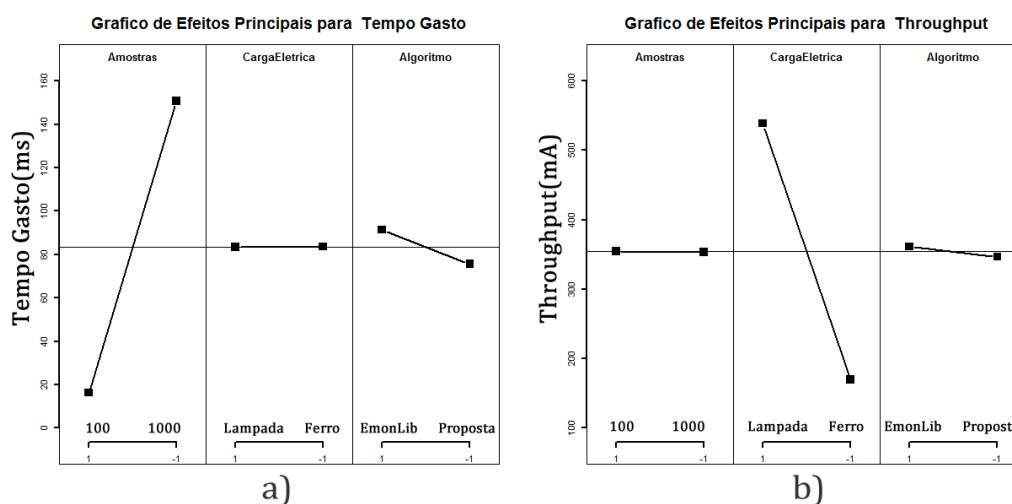


Figura 7. Efeitos dos Fatores nas variáveis. a) Tempo Gasto. b) Throughput.

várias amostras por leitura, porém o valor de corrente eficaz é calculado tomando o valor da corrente de pico, isto é, o mais alto valor lido na amostragem. Dessa forma, independente da quantidade de amostras, a mais alta será considerada, vide item 3.3.2 da seção 3 deste artigo. 2) . A diferença no *Throughput* entre cada algoritmo é pequena. 3) A carga elétrica é o único fator que possui impacto relevante sobre o *Throughput*, ou seja, os algoritmos fazem leituras identificando valores de corrente elétrica próximos. Para que os valores de corrente sofram grandes variações é preciso trocar as cargas elétricas.

5. Considerações finais

Este artigo apresentou o desenvolvimento do sistema Energesis, um solução para medição do consumo elétrico em hospedagem compartilhada utilizando a computação em Névoa, bem como a avaliação do desempenho do algoritmo de medições de consumo elétrico e resultados referentes aos experimentos realizados. Existem na literatura outros Smart Meters que realizam a leitura e processamento de sinais de consumo elétrico a afim de determinar o valor que será pago. Porém em ambientes de hospedagem compartilhada é preciso associar o consumo ao real consumidor. Devido possibilidade de curta duração da hospedagem (e.g. dois dias) em uma semana vários hóspedes podem ocupar o mesmo comodo, sendo essas hospedagens em dias diferentes.

Ao desenvolver um Smart Meter que monitore consumo elétrico uma das etapas mais difíceis é o pré-processamento de sinais de indução eletromagnética. Medições dessa natureza demandam aprofundado conhecimento em conceitos e grandezas da eletricidade. Além disso os sinais elétricos precisam ser convertidos em sinais digitais e discretizados ainda no Smart Meter antes de ser enviado à camada imediatamente superior. A captura e o pré-processamento desses sinais são feitas por um algoritmo embarcado. A calibragem do sensor, isto é, a identificação do resistor de carga é outro passo onde se deve ter extrema cautela, pois ainda que o algoritmo esteja bem projetado, os valores lidos e processados por ele são recebidos através de entradas analógicas do microcontrolador. Essas entradas são muito sensíveis, o que permitiria valores não-reais entrarem no sistema caso o resistor de carga do sensor esteja mal dimensionado. Isso tornaria as medições inconsistentes.

O Smart Meter, desenvolvido para o sistema Energesis, permite associar o valor do consumo ao hóspede. A avaliação de desempenho realizada mostra que o algoritmo proposto nesse trabalho possui menor tempo de execução em relação ao algoritmo tomado como *base-line*, e que os valores de corrente elétrica identificado por ambos os algoritmos são próximos, ou seja, o algoritmo proposto possui eficácia no seu propósito, o que habilita os sistema Energesis para ser utilizado como medidor de energia.

Como trabalhos futuros são sugeridos acrescentar o recurso de gráfico de linha, para representar o consumo elétrico de forma gráfica, utilizar um *Shield Wi-fi* para comunicação sem fio do medidor com a *Fog*, identificar separadamente o consumo de monitorado por cada sensor e desenvolver aplicativos moveis para o administrador e hóspede, permitindo ao hóspede acompanhar o consumo realizado por ele antes mesmo de lhe ser apresentada a fatura.

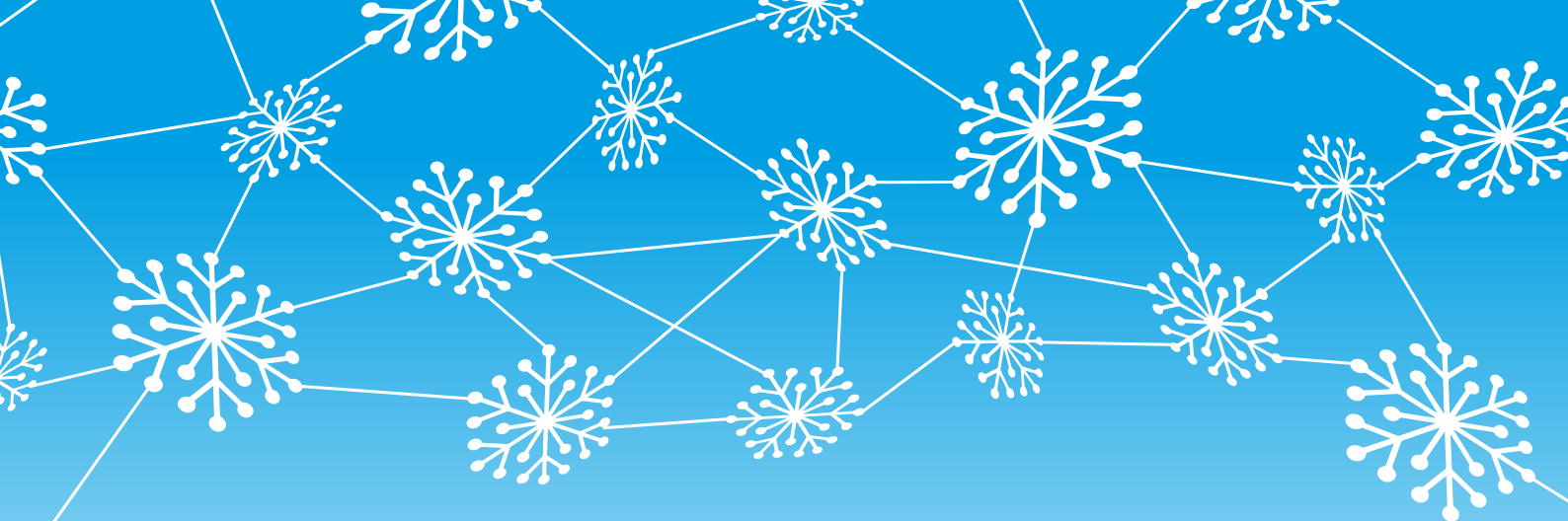
6. Agradecimentos

Os autores agradecem a FAPESB, CAPES, CNPq. Em especial ao MCTI-UFBA, pelo apoio financeiro por meio do Edital PROPCI/PROPG – PROPESQ/UFBA 004/2016.

Referências

- Advances Wales Magazine (2017). Energy monitoring unit for a sustainable future. *Advances Wales Magazine*, (82):18.
- ANEEL (2016). Como resolver problemas com recebimento de fatura. http://www.aneel.gov.br/como-resolver/-/asset_publisher/3SAW3SarixVj/content/problemas-com-recebimento-de-fatura/655804, Acesso em 20 jun. 2018.
- Arduino (2018). Language reference. analog i/o. <https://www.arduino.cc/reference/en/language/functions/analog-io/analogread/>, Acesso em 28 nov. 2018.
- Barros, E., Peixoto, M., Leite, D., Batista, B., and Kuehne, B. (2018). A fog model for dynamic load flow analysis in smart grids. In *2018 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–6.
- de Melo, G. S. and Roque, W. (2016). Sistema de monitoramento de consumo de energia elétrica de aparelhos eletroeletrônicos residenciais. *REVERTE – Revista de Estudos e Reflexões Tecnológicas da Faculdade de Tecnologia de Indaiatuba*. n.14, p.1-19.
- de Souza, A. R., Paixão, A. C., Dias, M. A., Uzêda, D. D., Duarte, S., and de Amorim, H. S. (2011). A placa arduino: uma opção de baixo custo para experiências de física assistidas pelo pc. *Revista Brasileira de Ensino de Física*. <http://sbfisica.org.br/rbef/pdf/331702.pdf>, Acesso em: 6 Dez. 2017.
- Ferreira, K. M., Méxas, M. P., Abreu, W., and Mello, P. (2016). Economia compartilhada e consumo colaborativo: Uma revisão da literatura. In: XII Congresso Nacional de Excelência em Gestão & III INOVARSE – Responsabilidade social aplicada, Rio de Janeiro, p.1-21.
- Goidanich, M. E. (2016). Airbnb e seus anfitriões: Empreendedorismo, comércio e colaboração na economia de compartilhamento. In: VIII Encontro Nacional de Estudos do Consumo, IV Encontro Luso-Brasileiro de Estudos do Consumo, II Encontro Latino-Americano de Estudos do Consumo Comida e alimentação na sociedade contemporânea, Niterói, p.1-20.

- Gurgel, C. G. (2017). A intervenção e regulação estatal sobre o serviço oferecido pelo airbnb no brasil. Master's thesis. Trabalho de Especialização, UFRN, Natal.
- Hikari (2013). Hikari: Soluções de confiança. <http://sawam.com.br/wp-content/uploads/2015/08/catalogo-hikari-2013.pdf>, Acesso: 21 de jun. de 2018.
- Jain, R. (1996). In *The art of Computer Systems Performance Analysis*. John Wiley & Sons Inc.
- Jaiswal, O. and Chaubisa, D. (2017). Arduino mega and iot based intelligent energy meter (iem) to increase efficiency and accuracy in current billing methodology,. International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), Chennai, p. 1901-1904.
- Kumar, S., Agarwal, S., Krishnamoorthy, A., Vijayarajan, V., and Kannadasan, R. (2018). Improving the response time in smart grid using fog computing. In Kulkarni, A. J., Satapathy, S. C., Kang, T., and Kashan, A. H., editors, *Proceedings of the 2nd International Conference on Data Engineering and Communication Technology*, pages 563–571, Singapore. Springer Singapore.
- Maxim Integrated (2015). Ds1307 64 x 8, serial, i2 c real-time clock. <https://datasheets.maximintegrated.com/en/ds/DS1307.pdf>, Acesso em: 30 Jun. 2018.
- Mukherjee, M., Shu, L., and Wang, D. (2018). Survey of fog computing: Fundamental, network applications, and research challenges. *IEEE Communications Surveys Tutorials*, 20(3):1826–1857.
- O Globo (2017). Compartilhamento deverá ser 30% do pib de serviços. <https://oglobo.globo.com/economia/compartilhamento-devera-ser-30-do-pib-de-servicos-22005402>, Acesso: 16 de jun de 2018.
- Okay, F. Y. and Ozdemir, S. (2016). A fog computing based smart grid model. In *2016 International Symposium on Networks, Computers and Communications (ISNCC)*, pages 1–6.
- OpenEnergyMonitor (2016). Solar pv monitoring. <https://guide.openenergymonitor.org/setup/>, Acesso em: 4 jul. 2018.
- Pant, V., Jain, S., and Chauhan, R. (2017). Integration of fog and iot model for the future smart grid. In *Emerging Trends in Computing and Communication Technologies (ICETCCT), International Conference on*, pages 1–6. IEEE.
- Soares, A., Dias, M., and Filho, L. (2017). A experiência do turista e a hospedagem compartilhada através do uso das novas tecnologias no turismo: O caso do airbnb. *Revista Turismo & Desenvolvimento*. n. 27/28, p.1315 – 1324.
- Yan, Y. and Su, W. (2016). A fog computing solution for advanced metering infrastructure. In *Transmission and Distribution Conference and Exposition (T&D), 2016 IEEE/PES*, pages 1–4. IEEE.
- YHDC (2015). Current transformer ta 12-100. product specification. http://www.yhdc.us/ENpdf/TA12-100_YHDC.pdf, Acesso em: 24 Abr. 2018.
- Zahoor, S., Javaid, N., Khalid, A., Yasmeen, A., and Nadeem, Z. (2018). Fog computing based energy management system model for smart buildings. In *International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, pages 719–727. Springer.



Realização



Organização



Apoio Institucional



Patrocínio

Diamante



Ouro



Prata



Bronze

