



Universidade Estadual de Campinas  
Instituto de Computação



Lucas Zanco Ladeira

Serviço de Sugestão de Rotas Personalizável e Ciente  
de Contexto

CAMPINAS  
2020

**Lucas Zanco Ladeira**

**Serviço de Sugestão de Rotas Personalizável e Ciente de Contexto**

Dissertação apresentada ao Instituto de Computação da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

**Orientador: Prof. Dr. Leandro Aparecido Villas**

Este exemplar corresponde à versão final da Dissertação defendida por Lucas Zanco Ladeira e orientada pelo Prof. Dr. Leandro Aparecido Villas.

CAMPINAS  
2020

Ficha catalográfica  
Universidade Estadual de Campinas  
Biblioteca do Instituto de Matemática, Estatística e Computação Científica  
Ana Regina Machado - CRB 8/5467

L122s Ladeira, Lucas Zanco, 1992-  
Serviço de sugestão de rotas personalizável e ciente de contexto / Lucas Zanco Ladeira. – Campinas, SP : [s.n.], 2020.

Orientador: Leandro Aparecido Villas.  
Dissertação (mestrado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Problema de roteamento de veículos. 2. Sistemas de comunicação móvel. 3. Sistemas inteligentes de veículos rodoviários. 4. Redes ad hoc veiculares (Redes de computadores). I. Villas, Leandro Aparecido, 1983-. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Informações para Biblioteca Digital

**Título em outro idioma:** Personalized and context-aware route suggestion service

**Palavras-chave em inglês:**

Routing vehicles problem

Mobile communication systems

Intelligent transportation systems

Vehicular ad hoc networks (Computer networks)

**Área de concentração:** Ciência da Computação

**Titulação:** Mestre em Ciência da Computação

**Banca examinadora:**

Leandro Aparecido Villas [Orientador]

Edmundo Roberto Mauro Madeira

Heitor Soares Ramos Filho

**Data de defesa:** 24-01-2020

**Programa de Pós-Graduação:** Ciência da Computação

**Identificação e informações acadêmicas do(a) aluno(a)**

- ORCID do autor: <https://orcid.org/0000-0002-8279-2595>

- Currículo Lattes do autor: <http://lattes.cnpq.br/9849459140148624>



Universidade Estadual de Campinas  
Instituto de Computação



Lucas Zanco Ladeira

Serviço de Sugestão de Rotas Personalizável e Ciente de Contexto

**Banca Examinadora:**

- Prof. Dr. Leandro Aparecido Villas  
Instituto de Computação - UNICAMP
- Prof. Dr. Edmundo Roberto Mauro Madeira  
Instituto de Computação - UNICAMP
- Prof. Dr. Heitor Soares Ramos Filho  
Departamento de Ciência da Computação - UFMG

A ata da defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Programa da Unidade.

Campinas, 24 de janeiro de 2020

*Fever dreams,  
Can only haunt you,  
Til the fever breaks,  
They can only haunt you,  
Til the fever breaks,  
They can only haunt you,  
Til the fever breaks,  
They can only haunt you.*

(Dashboard Confessional - Fever Dreams)

# Agradecimentos

Primeiramente eu gostaria de agradecer a minha família que me apoiou emocionalmente e financeiramente para que eu pudesse cursar o Mestrado. O apoio deles foi fundamental durante todo o tempo que passou e os desafios que tive que enfrentar em tal jornada. Eu gostaria de agradecer especialmente aos meus pais Luiz e Silva, irmã Jessica, e avós Sdney, Anezina, Ione, e Luís. Agradeço a minha namorada Ligia que me apoiou e teve paciência durante todo o percurso, mesmo em momentos difíceis de stress e dúvidas. Gostaria de agradecer aos meus amigos do Laboratório de Redes de Computadores, aos outros amigos da Unicamp, e aos de Marília. A todos eles agradeço por terem me dado tantos momentos felizes, e terem me ouvido quando eu mais precisava. Eu não poderia esquecer de agradecer ao meu orientador Leandro por ter me recebido tão bem na Unicamp, ter me ajudado a entender o que é o trabalho de pesquisa, e de ajudar a gerar frutos para a ciência. Por fim, agradeço ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq, processo: 167213/2018-0) pela concessão da bolsa e financiamento do projeto e também, ao programa de pós-graduação em Ciência da Computação do Instituto de Computação (Unicamp), pelo apoio operacional.

# Resumo

Dados contextuais caracterizam regiões distintas da cidade permitindo diferenciá-las de acordo com segurança, entretenimento, serviços, entre outros. O uso de dados contextuais para sugerir rotas ajuda a entender novos aspectos de uma cidade que podem mudar a percepção dos usuários sobre diferentes rotas. O impacto de cada tipo de dado contextual pode variar de acordo com o perfil do usuário, o qual não é levado em consideração na maior parte dos serviços propostos pela literatura. Além disso, é necessário considerar o comportamento dos dados contextuais o qual muda de acordo com o tipo de dado. Por fim, devido à escassez de dados de mobilidade reais disponíveis abertamente, a maioria dos trabalhos consideram traces sintéticos. Os traces sintéticos podem não representar a mobilidade urbana próxima da realidade. Para mitigar os problemas citados anteriormente é proposto um serviço de sugestão de rotas com risco espaço-temporal. O serviço é composto por três módulos, sendo: identificação das janelas contextuais, mapeamento de contexto, e personalização do roteamento. O primeiro módulo é responsável pela análise do comportamento dos dados contextuais para identificar diferentes situações que ocorrem. Para tal, são utilizadas características comuns entre os dados e a distribuição espaço-temporal. O segundo módulo é responsável pelo mapeamento das diferentes situações no cenário através de distribuições de probabilidade distintas. O terceiro módulo é responsável por, de acordo com o perfil do usuário, personalizar a sugestão de rotas. Além do serviço de sugestão de rotas proposto, também foi proposta uma estratégia para geração de padrões de mobilidade utilizando dados reais de fluxo de tráfego. Os resultados da avaliação mostram que o serviço se adapta a sensíveis mudanças no perfil do usuário. Além disso, foram obtidos resultados positivos pela utilização do comportamento dos dados contextuais para evitar requisições desnecessárias. Isso permitiu diminuir em até 50% das requisições feitas ao serviço.

# Abstract

Contextual data characterize distinct regions of the city allowing to differentiate them according to security, entertainment, services, among others. Using contextual data to suggest routes helps to understand new aspects of a city that can change users' perceptions of different routes. The impact of each type of contextual data may vary according to the user's profile, which is not taken into account in most of the services proposed by the literature. In addition, it is necessary to consider the behavior of contextual data which changes according to the type of data. Finally, due to the scarcity of real mobility data available openly, most studies consider synthetic traces. Synthetic traces may not represent urban mobility close to reality. To mitigate the problems mentioned above, a route suggestion service with space-time risk is proposed. The service consists of three modules, namely: identification of contextual windows, context mapping, and customization of routing. The first module is responsible for analyzing the behavior of contextual data to identify different situations that occur. For this, common characteristics are used between the data and the spatio-temporal distribution. The second module is responsible for mapping the different situations in the scenario through different probability distributions. The third module is responsible for, according to the user's profile, personalizing the suggested routes. In addition to the proposed route suggestion service, a strategy for generating mobility patterns using real traffic flow data was also proposed. The evaluation results show that the service adapts to sensitive changes in the user's profile. In addition, positive results were obtained by using the behavior of contextual data to avoid unnecessary requests. This allowed for a reduction of up to 50% of requests made to the service.

# Lista de Figuras

2.1	Alto Nível da Arquitetura. Fonte: [24]. . . . .	21
2.2	Ilustração de tipos distintos de rotas. . . . .	23
2.3	Comparação de rotas mais rápida e rotas mais segura. . . . .	24
4.1	Mapa de calor de Chicago. . . . .	32
4.2	Cenário selecionado de Chicago. . . . .	33
4.3	Distribuição de assalto nos domingos do ano. . . . .	33
4.4	Distribuição do crime de assalto em dias distintos. . . . .	34
4.5	Comparação dos dados da API e do SUMO. . . . .	36
4.6	Mapeamento dos dados de tráfego. . . . .	38
5.1	Ilustração da arquitetura do serviço proposto. . . . .	40
5.2	Pontuação da janela (eixo y) para homicídios em Chicago distribuídos ao longo do dia (eixo x) em 144 partes iguais. . . . .	43
5.3	Comparação entre quantidade de clusters e ocorrências criminais de assalto. . . . .	44
5.4	Comparação entre quantidade de clusters e ocorrências criminais de violência. . . . .	45
5.5	Kernel Density Estimations de diferentes dados contextuais . . . . .	46
5.6	Janelas contextuais distintas ao longo do dia para diferentes tipos de dados contextuais. . . . .	47
6.1	Resultados do sistema de acordo com os dias da semana e horas distintas. . . . .	51
6.2	Comparação da quantidade de chamadas e a quantidade de execuções por configuração. . . . .	52
6.3	Comparação do impacto em dias da semana dos tipos criminais . . . . .	54

# Lista de Tabelas

3.1	Comparação dos trabalhos relacionados sob a perspectiva da identificação de janelas contextuais. . . . .	27
3.2	Comparação dos trabalhos relacionados sob a perspectiva de sugestão de rotas. . . . .	30
6.1	Configurações de roteamento. . . . .	50
6.2	Resultados do roteamento. . . . .	50

# Lista de Abreviações e Siglas

- DBScan - *Density-Based Spatial Clustering of Applications with Noise*
- DSRC - *Dedicated Short-Range Communication*
- DT - *Decision Trees*
- ITS - *Sistemas de Transporte Inteligentes*
- KDE - *Kernel Density Estimation*
- KNN - *K-Nearest Neighbors*
- LR - *Linear Regression*
- MLP - *Multilayer Perceptron*
- ODP - *Open Data Portal*
- OSM - *Open Street Map*
- RF - *Random Forest*
- RCSP - *Resource Constrained Shortest Path*
- SUMO - *Simulation of Urban Mobility*
- SVM - *Support Vector Machines*
- TMS - *Sistemas de Gerenciamento de Tráfego*
- V2V - *Vehicle-to-Vehicle*
- V2I - *Vehicle-to-Infrastructure*

# Sumário

<b>1</b>	<b>Introdução</b>	<b>14</b>
1.1	Motivação . . . . .	15
1.2	Objetivos e Contribuições . . . . .	15
1.3	Publicações . . . . .	16
1.4	Estrutura da Dissertação . . . . .	16
<b>2</b>	<b>Fundamentação Teórica</b>	<b>17</b>
2.1	Dados Contextuais e Janelas Temporais . . . . .	17
2.2	Modelos de Aprendizado de Máquina . . . . .	18
2.3	Sistemas de Transporte Inteligentes . . . . .	20
2.4	Sugestão de Rotas . . . . .	22
2.5	Resumo Conclusivo . . . . .	24
<b>3</b>	<b>Trabalhos Relacionados</b>	<b>25</b>
3.1	Identificação de Janelas Temporais . . . . .	25
3.2	Roteamento Ciente de Contexto . . . . .	26
3.3	Resumo Conclusivo . . . . .	29
<b>4</b>	<b>Ambiente e Cenário das Simulações</b>	<b>31</b>
4.1	Simulador e Cenário . . . . .	31
4.2	Dados Contextuais . . . . .	33
4.3	Mobilidade Veicular . . . . .	35
4.4	Resumo Conclusivo . . . . .	37
<b>5</b>	<b>Serviço de Sugestão de Rotas</b>	<b>39</b>
5.1	Identificação do Contexto . . . . .	40
5.1.1	Caracterização do Contexto . . . . .	40
5.1.2	Identificação de Janelas Temporais . . . . .	41
5.1.3	Filtragem das Janelas Contextuais . . . . .	43
5.1.4	Avaliação da Identificação do Contexto . . . . .	43
5.2	Serviço de Sugestão de Rotas . . . . .	44
5.2.1	Mapeamento do Contexto . . . . .	45
5.2.2	Personalização do Roteamento . . . . .	46
5.3	Frequência de Atualização . . . . .	47
5.4	Resumo Conclusivo . . . . .	48
<b>6</b>	<b>Avaliação do Serviço</b>	<b>49</b>
6.1	Metodologia do roteamento . . . . .	49
6.2	Resultados do roteamento . . . . .	49

6.3	Metodologia da taxa de atualização . . . . .	52
6.4	Resultados da taxa de atualização . . . . .	53
6.5	Resumo Conclusivo . . . . .	53
<b>7</b>	<b>Considerações Finais e Trabalhos Futuros</b>	<b>55</b>
7.1	Conclusões . . . . .	55
7.2	Trabalhos Futuros . . . . .	56
	<b>Referências Bibliográficas</b>	<b>57</b>

# Capítulo 1

## Introdução

O crescimento demográfico nas cidades aumenta o fluxo de veículos, tornando mais comum a ocorrência de congestionamentos e diminuindo a segurança nas vias [9, 5]. Existem várias aplicações que tentam colaborar com o gerenciamento eficiente do tráfego, como também prover diversos serviços durante um percurso, tais quais: predição do tempo de percurso de ônibus [50], transmissão de conteúdo de multimídia [49], identificação de vias congestionadas [8, 1, 35, 31], entre outras. Essas aplicações compõem uma categoria de sistemas conhecida como Sistemas de Transporte Inteligentes (ITS). De acordo com [5], ITSs buscam resolver problemas ligados à mobilidade urbana nas cidades, e para isso, utilizam dados referentes ao tráfego, comunicação entre dispositivos, e computação. Esses sistemas são de extrema importância em grandes centros urbanos devido aos congestionamentos frequentes [5].

Uma especificação dos ITSs é conhecida como Sistemas de Gerenciamento de Tráfego (TMS), a qual foca na identificação de congestionamentos e controle do fluxo de veículos [8]. Esses sistemas são compostos de outras aplicações que mineram, processam, e analisam os dados sobre o tráfego, para prover a inteligência na escolha de rotas e distribuição do tráfego. Com o intuito de diminuir a quantidade de tempo gasto durante o percurso de um local ao outro, as rotas mais rápidas são o foco de grande parte dos sistemas. Essas rotas podem compreender tanto o trajeto feito por um indivíduo em seu veículo, ou multimodal, sendo que o indivíduo pode mudar o meio de transporte durante o trajeto [4]. A escolha das rotas mais rápidas para um determinado veículo, é provida por aplicações, tais como Google Maps e Waze. Essas aplicações fazem o uso de dados de infraestrutura e de sensoriamento participativo para identificar vias congestionadas [16].

No entanto, outros dados podem ser agregados com os dados de tráfego para obter novas rotas. Comumente esses dados tem a característica de serem espaciais pela capacidade de mapeá-los e identificar regiões a serem evitadas / beneficiadas durante a tarefa. Além dessa, outra característica encontrada é a variável temporal dos mesmos, possibilitando identificar as mudanças que ocorrem em uma determinada região durante um período de tempo. Esses dados podem ser provenientes de plataformas de crowdsourcing, tais como: Foursquare, Facebook, Twitter, entre outras. Uma iniciativa importante, chamada Open-DataPortal<sup>1</sup> (ODP), permite a extração de dados abertos de várias cidades. Os dados

---

<sup>1</sup><http://data.europa.eu/euodp/en/home/>

do ODP são estruturados e representam vários aspectos da cidade, tais como: segurança, educação, transporte e saúde. Isso permite a criação de estudos em diferentes cenários e comparação dos resultados obtidos.

Conseqüentemente, podem ser citados alguns trabalhos que apresentam resultados interessantes que justificam essa agregação de dados. O trabalho de [33], propõe a utilização de informações, vindas de redes sociais, sobre os níveis de poluição das regiões. Outros trabalhos, sugerem o uso de informações sobre o que as pessoas pensam sobre cada local [34, 38]. Também é necessário citar, a utilização de dados criminais [37, 7, 15, 10] para identificar regiões com alta incidência criminal. De forma geral, esses trabalhos selecionam rotas com o objetivo de evitar vias que possam prejudicar a qualidade de experiência do usuário.

## 1.1 Motivação

Foi explorado extensamente pela literatura estratégias para a identificação de congestionamentos e re-roteamento de veículos com o intuito de diminuir o tráfego [50, 8, 2, 35, 31]. Todavia, o tráfego não é o único problema encontrado durante um percurso, sendo que, é possível encontrar na mídia relatos de fatalidades que ocorreram por causa de aplicações de sugestão de rotas que levaram usuários para regiões de perigo [14, 32, 17, 11]. Entre os trabalhos encontrados, os que consideram a variável temporal para caracterizar as mudanças nas situações das regiões [41, 37] particionam os dados com um intervalo de tempo fixo, o que pode não compreender o comportamento dos dados. Além disso, não são apresentados resultados com a utilização de múltiplos tipos distintos de dados contextuais de forma a observar o comportamento dos serviços propostos.

No melhor do nosso conhecimento, os trabalhos da literatura que utilizam dados contextuais para sugerir rotas não levam em consideração a identificação do comportamento temporal dos dados para sugerir rotas. Cada tipo de dado contextual pode ter um impacto distinto de acordo com o perfil do usuário. Sendo assim, é necessária a capacidade de atualizar a sugestão de rotas de acordo com esse perfil. Nos trabalhos da literatura, a personalização do serviço está geralmente ligada a frequência da utilização de determinadas rotas pelos usuários [36], portanto é necessário propor uma maneira de, partindo do perfil do usuário, identificar a rota mais próxima das necessidades do usuário.

## 1.2 Objetivos e Contribuições

O principal objetivo desse trabalho é de propor um serviço capaz de identificar o comportamento de tipos distintos de dados contextuais espaço-temporais, e sugerir rotas considerando esse comportamento e o perfil do usuário. O comportamento será mapeado por janelas temporais que podem variar de acordo com o dia da semana, hora, e subtipo do dado. Dessa forma, o serviço contribuirá com a capacidade de mapear diferentes características da cidade e considerá-las para rotear os veículos. Além disso, para simular o serviço é proposta uma estratégia para gerar padrões de mobilidade mais próximos da realidade considerando dados reais de fluxo de tráfego.

### 1.3 Publicações

- (Publicado) - LADEIRA, L. Z.; SOUZA, A. M. ; ROCHA FILHO, G. P. ; SILVA, T. H. ; VILLAS, L. A. Serviço de Sugestão de Rotas Seguras para Veículos. In: Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), 2019, Gramado. XXXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), 2019. v. 37.
- (Publicado) - LADEIRA, L. Z.; SOUZA, A. M. ; ROCHA FILHO, G. P. ; SILVA, T. H. ; FERRARONI, M. S.; VILLAS, L. A. MARTINI: Towards a mobile and variable timewindow identification for spatio-temporal data. In: IEEE Latin-American Conference on Communications, 2019, Salvador.
- (Submetido) - LADEIRA, L. Z.; SOUZA, A. M. ; SILVA, T. H. ; Pazzi, R. W.; VILLAS, L. A. PONCHE: Personalized and Context-Aware Vehicle Rerouting Service. In: IEEE INTERNATIONAL CONFERENCE ON CLOUD COMPUTING, 2020.
- (Submetido) - LADEIRA, L. Z.; SOUZA, A. M. ; SILVA, T. H. ; ROCHA FILHO, G. P. ; PEIXOTO, M. L. M.; VILLAS, L. A. CERVA: Roteamento Contextual para Veículos com Risco Espaço-temporal. In: XXXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), 2020.

### 1.4 Estrutura da Dissertação

Essa dissertação está estruturada em 7 capítulos. O Capítulo 2 apresenta a fundamentação teórica necessária para o desenvolvimento do trabalho, considerando a base teórica e diferentes estratégias utilizadas pela literatura para resolver os problemas pontuados anteriormente na Seção 1.1. No Capítulo 3 são apresentados os trabalhos da literatura relacionados. Esses estão divididos em dois grupos de acordo com a utilização de dados contextuais e a sugestão de rotas. O Capítulo 4 apresenta o cenário, a mobilidade veicular, e o simulador utilizado. No Capítulo 5 é apresentado o serviço proposto dividido em algumas seções, tais são: identificação das janelas contextuais, se refere aos passos utilizados para identificar do comportamento dos dados contextuais, e filtrá-los; a sugestão de rotas, a estratégia utilizada para mapear os dados contextuais no cenário, e personalizar o roteamento de acordo com o perfil do usuário; a estratégia de balanceamento de carga aplicada; e por fim, uma estratégia para diminuir a quantidade de requisições ao serviço. No Capítulo 6 os resultados do serviço são apresentados de acordo com perfis de usuários distintos, como também resultados sobre a estratégia para diminuir a quantidade de requisições ao serviço. Por fim, o Capítulo 7 apresenta as conclusões e trabalhos futuros.

# Capítulo 2

## Fundamentação Teórica

Nesse capítulo são apresentados os conceitos fundamentais para o desenvolvimento do trabalho, tais como: características dos dados utilizados, tipos de janelas temporais, algoritmos de agrupamento, arquitetura geral de sistemas de transporte inteligentes, como é feito o roteamento de veículos, e tipos distintos de rotas.

### 2.1 Dados Contextuais e Janelas Temporais

Como já mencionado no Capítulo 1, o tráfego não é o único problema que pode afetar usuários, sendo assim, é possível utilizar dados contextuais para entender aspectos distintos da cidade. No entanto, é necessário observar características específicas desses dados para entender o comportamento de cada tipo distinto, tais como: ocorrências criminais ou acidentes envolvendo veículos.

Para exemplificar, considerando dados sobre ocorrências criminais, o crime de furto é mais provável em locais com maior concentração de pessoas, pela motivação de subtrair um objetivo sem que a pessoa perceba e deixar o local facilmente. A incidência de crimes, motivada pelas atividades econômicas e sociais do dia-a-dia, pode aumentar em uma determinada região de acordo com o tipo de crime [19]. Sendo assim, é necessário entender cada situação distinta referente a uma determinada região. A situação se refere à distribuição de um determinado dado em cada região.

De acordo com [26, 45, 3, 12], a situação muda de acordo com o tipo de crime, a hora do dia, e o dia da semana. Essa granularidade nos dados permite uma visão aprofundada de cada situação e possibilidade de análise para tipos de dados específicos. O tempo de validade do dado também é de grande importância pelas mudanças que podem ocorrer no contexto urbano. Pois, novos pontos de interesse para criminosos podem aparecer, e portanto, mudar a distribuição de crimes na cidade. No caso de dados contextuais sobre acidentes envolvendo veículos, essa mudança pode ocorrer de acordo com a alteração na velocidade máxima de cada via [43, 44].

Para identificar as mudanças que ocorrem durante o dia, estudos anteriores consideraram uma janela de temporal [51, 7, 37, 28]. Então, cada janela temporal compreende um intervalo de tempo que representa uma determinada situação. Geralmente, esses trabalhos utilizam um tamanho fixo de acordo com um intervalo de horas, mas podem variar

bastante com base no resultado esperado. Existem também outras propostas para filtrar dados [26, 29] e são apresentadas com mais profundidade no Capítulo 3.

Embora uma abordagem de janela fixa considere intervalos de tempo diferentes em um dia, o comportamento (espacial e temporal) dos dados pode mudar de acordo com o tipo de dado [21]. Isso se refere não apenas ao tempo em que a incidência de um evento começa a crescer, mas ao intervalo de tempo em que o evento acontece. Uma janela fixa pode dividir dados enquanto a incidência do evento está aumentando, o que pode impedir a detecção desse crescimento por algum tempo ou a área total ocupada por um ponto ativo. Outro problema refere-se ao tamanho da janela, que pode encontrar pontos de acesso de eventos levando em consideração eventos que ocorreram um tempo considerável. Além disso, se torna mais difícil identificar eventos discrepantes pela quantidade de dados apresentados em tamanhos de janela maiores. Isso se baseia na ideia de que eventos de situações distintas podem ser usados para identificar um ponto de acesso impreciso.

Uma janela de tempo é normalmente aplicada em serviços que desejam usar as informações sobre a situação para tomada de decisão. Por exemplo, um serviço que gerencia forças policiais e precisa identificar onde é necessário auxílio policial, requer essas informações para identificar pontos críticos do crime [29]. Outro exemplo é um serviço de sugestão de rota que precisa dessas informações para caracterizar a criminalidade de acordo com o horário da solicitação de rota [15, 7, 37, 29]. A qualidade desses serviços depende da precisão da situação identificada, o que pode induzir o usuário em erro se ele não tiver uma precisão confiável. Como resultado, se os dados não fornecerem um entendimento claro da situação, um serviço de sugestão de rota poderá levar o usuário a uma área insegura.

A janela temporal apenas separa os dados em grupos distintos mas, não filtra os dados para remover *outliers*. No caso dos tipos de dados contextuais mencionados, ocorrências criminais e acidentes envolvendo veículos, um *outlier* é um evento isolado dentro do conjunto de dados. O intuito de remover esses dados é de considerar apenas regiões com um nível de crimes, ou de acidentes, significativo. Sendo assim, são apresentados na Seção 2.2 algoritmos que podem ser utilizados com esse propósito.

## 2.2 Modelos de Aprendizado de Máquina

Algoritmos de aprendizado de máquina tem o propósito de aprender um padrão em dados para resolver alguma tarefa [6]. Para tal, a maior parte dos algoritmos requer treinamento podendo variar significativamente o tempo necessário de acordo com a estratégia utilizada. Sendo assim, o resultado do algoritmo é um modelo que representa o conhecimento extraído dos dados sobre a tarefa. A tarefa pode compreender a classificação de espécies de plantas, predição de valores de ações, ou agrupamento de usuários em redes sociais. Algoritmos de aprendizado de máquina podem ser agrupados em grupos distintos de acordo com a tarefa a ser executada, nesse trabalho serão apresentados brevemente apenas dois, tais como: aprendizado supervisionado e aprendizado não supervisionado.

Aprendizado supervisionado requer dados previamente rotulados. Esses dados possuem duas partes, as características (*features*) e o rótulo (*class* ou classe). Esse tipo de algoritmo é subdividido em algoritmos de classificação ou algoritmos de regressão. Al-

algoritmos de classificação tentam rotular novos dados de acordo com uma tipificação, ou seja, geralmente o resultado é um valor inteiro (1, 2, 3, 4), ou um texto (A, B, C, D). Como exemplo da utilização de algoritmos de classificação, supondo que a tarefa seja classificar ocorrências criminais em tipos distintos, cada classe se refere a um tipo de crime (furto, roubo, entre outros), e assim, as características de cada ocorrência caracteriza o tipo de crime em questão. Alguns algoritmos de classificação podem ser citados, tais como: K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Decision Trees (DT), Random Forest (RF).

Algoritmos de regressão tentam rotular dados de acordo com valores de ponto flutuante. Como exemplo, dado um conjunto de dados sobre a variação no valor dos barris de petróleo, é desejado prever os valores futuros. O resultado do treinamento é dado por uma função matemática. Alguns algoritmos de regressão podem ser citados, tais como: Linear Regression (LR), Multilayer Perceptron (MLP), Support Vector Machines (SVM), Decision Trees (DT), Random Forest (RF). É possível observar que alguns algoritmos de classificação também são utilizados para regressão. Isso ocorre por utilizarem a mesma estratégia para ambos, mas mudam a forma que o resultado é obtido.

Aprendizado não-supervisionado não necessita de dados rotulados. Esse tipo de algoritmo tem o objetivo de agrupar, ou identificar características comuns em dados. Também são conhecidos como algoritmos de agrupamento (*clustering*). Um exemplo é a tarefa de agrupar usuários de serviços de *streaming* para sugerir filmes de acordo com perfis semelhantes. Alguns exemplos de algoritmos podem ser citados, tais como: KMeans, DBScan, HDBScan, OPTICS, MeanShift. Esse tipo de algoritmo também pode ser utilizado para agrupar dados e então identificar *outliers*.

KMeans utiliza o cálculo da distância entre pontos para identificar grupos. Dado um número  $n$  de grupos, o algoritmo tenta separar os dados em  $n$  grupos com uma variância igual, de forma a minimizar um critério escolhido. Esse critério tenta identificar quais grupos são coerentes. A cada iteração do algoritmo é escolhido um núcleo aleatório para cada cluster. Por considerar núcleos aleatórios o resultado do algoritmo pode variar. Além disso, como é necessário mencionar a quantidade de grupos é restrito em alguns cenários sendo necessário explorar diversas possibilidades.

*Density-Based Spatial clustering of applications with noise* (DBScan), é um algoritmo que utiliza a distância entre os pontos mais próximos para identificar os grupos [38]. Geralmente é utilizado para encontrar grupos quando o tamanho deles possui grande variação. Essa é uma característica interessante pois, os crimes podem ocorrer concentrados em uma região, ou espalhados na mesma. Para encontrar os parâmetros do algoritmo é necessário apenas escolher o tamanho mínimo da vizinhança  $v$  para ser considerado um novo grupo, e a distância máxima  $\varepsilon$  entre dois pontos. Esses dois parâmetros são utilizados para classificar os pontos em 3 tipos distintos, sendo: núcleo, borda, e *outlier*. Os pontos de núcleo possuem uma vizinhança mínima  $v$ . Pontos de borda estão no máximo uma distância  $\varepsilon$  de algum ponto de núcleo. Por fim, os pontos de *outlier*, não estão enquadrados em nenhum dos requisitos anteriores.

Esse algoritmo apresenta algumas vantagens sobre o KMeans, a primeira se refere à possível variação na quantidade de grupos de acordo com uma janela de tempo. No caso do KMeans seria necessário manter fixo o número de grupos, ou utilizar alguma métrica para

tentar variar a quantidade e identificar a melhor. Ademais, não seria possível identificar pontos de *outliers*, o qual é feito pelo algoritmo DBScan. A separação entre os registros é mais clara no DBScan como apresentada em [25].

O algoritmo HDBScan é uma extensão do DBScan, que faz a seleção automática da distância máxima considerada pelo algoritmo [27]. Essa seleção é baseada na escolha de uma distância máxima para cada grupo, seguindo a Equação 2.1 de *reachability*. Nessa equação,  $X_i$  e  $X_j$  referenciam dois pontos quaisquer,  $\kappa$  é a distância para o vizinho mais próximo, e  $d$  é a distância entre  $X_i$  e  $X_j$ .

$$d_{mreach}(X_i, X_j) = \begin{cases} \max\{\kappa(X_i), \kappa(X_j), d(X_i, X_j)\}, & X_i \neq X_j \\ 0, & X_i = X_j \end{cases} \quad (2.1)$$

Então é aplicado o método *Single Linkage Clustering* para identificar os possíveis grupos. Como o objetivo é encontrar áreas com grande densidade de pontos, é gerada uma árvore condensada com as possibilidades de grupos e identificado se é necessário aplicar a separação entre grupos. Para uma descrição mais detalhada do algoritmo é possível verificar o trabalho de [27]. Um problema encontrado nesse algoritmo, é a possibilidade da distância máxima encontrada, separar ou deixar de identificar regiões com alta densidade. Ou seja, a distância escolhida não caracteriza a distribuição dos pontos em uma determinada região, classificando pontos como *outliers*.

No sentido contrário do HDBScan, é possível citar o algoritmo *Agglomerative Clustering*. Esse algoritmo considera que cada ponto é um grupo distinto, e pelo cálculo da similaridade entre pontos, combina esses pontos em um mesmo grupo. O cálculo da similaridade pode ser feito de 4 maneiras diferentes: *ward*, que minimiza a soma do quadrado da diferença entre todos os grupos, similaridade completa, minimiza a distância máxima entre pares de grupos, similaridade média, minimiza a média das distância entre os pares de grupos, e por fim, similaridade única, minimiza a distância entre os pares de grupos mais próximos.

## 2.3 Sistemas de Transporte Inteligentes

Com o objetivo padronizar a forma de comunicação entre os vários dispositivos dentro de um ITS, algumas arquiteturas foram propostas. A arquitetura americana, veja a Figura 2.1, é apresentada brevemente, por servir de referência para grandes empresas como a IBM<sup>1</sup>. Essa arquitetura é dividida em 4 módulos, sendo: *field*, *centers*, *vehicles*, e *travelers*. O primeiro, se refere à infraestrutura utilizada pelo sistema, ou seja, sensores fixos e RSUs. *Centers*, faz o controle do sistema e gerencia a tomada de decisão das outras partes. O módulo *Vehicles* representa os diferentes veículos presentes em uma ITS. Finalmente, *Travelers* define os dispositivos utilizados pelas pessoas. De acordo com [5], alguns problemas dessa arquitetura podem ser citados, tal como, a dificuldade em permitir o uso de várias tecnologias de comunicação. Além disso, a comunicação entre o módulo que faz o controle das aplicações e os veículos é restringida através de uma interface, a qual não permite o uso de novos paradigmas.

<sup>1</sup><https://www.ibm.com/developerworks/industry/library/ind-tmdd/index.html>

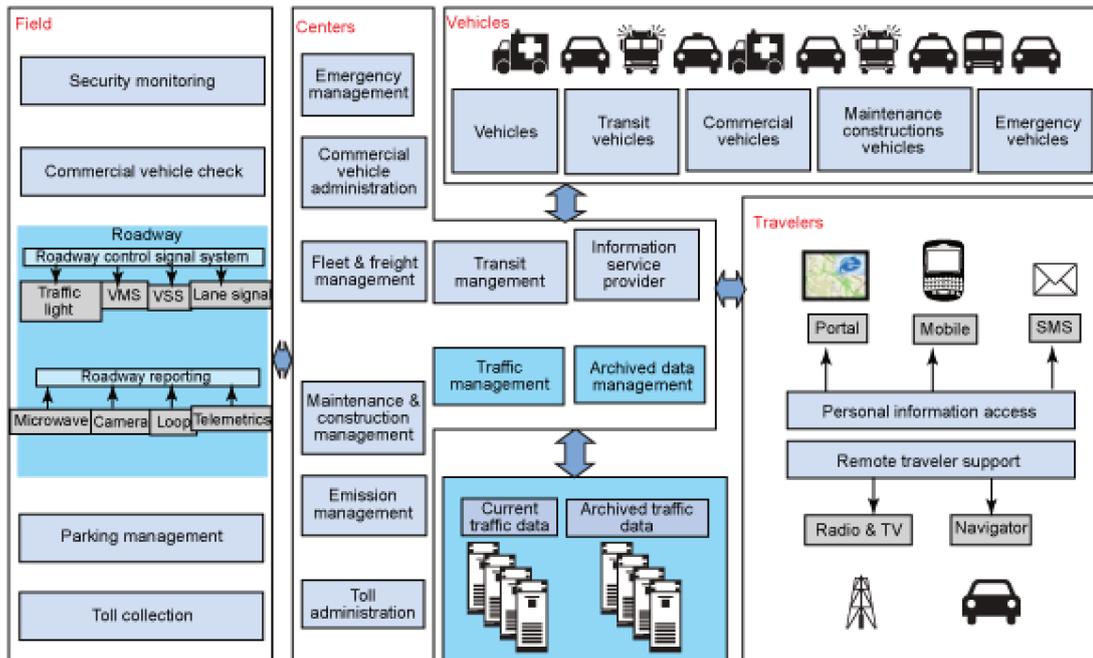


Figura 2.1: Alto Nível da Arquitetura. Fonte: [24].

O arcabouço que compreende os ITSs, aplica inteligência com o intuito de resolver problemas referentes ao transporte, sendo assim, é ampla a quantidade de categorias de problemas que podem ser abordadas. Essas categorias são referentes a segurança dos usuários, o entretenimento, o sensoriamento urbano, e ao gerenciamento eficiente do tráfego. A segurança está ligada a possibilidade de alertar o condutor sobre acidentes, obras na pista, e congestionamentos. Com esse alerta, é esperado que o condutor seja cauteloso e possa diminuir a velocidade do veículo ou desviar de um obstáculo. Além disso, a segurança se refere a capacidade de um sistema de identificar uma provável colisão entre veículos e informar um dos condutores.

A categoria de entretenimento representa a entrega de conteúdo de entretenimento ao condutor, por exemplo, notícias, clima, promoções, estabelecimentos próximos, entre outros. Esses sistemas tentam informar o condutor sobre informações de locais que ele irá passar próximo, ou outras informações relevantes durante o percurso. O sensoriamento urbano, se refere a capacidade de extrair dados dos sensores presentes nos veículos. Isso possibilita entender a mobilidade urbana, e a rotina das pessoas na cidade. Com esses dados é possível identificar congestionamentos, e então, utilizar os dados históricos para prever novos congestionamentos. Essa previsão colabora na distribuição da mobilidade dos veículos, para evitar a ocorrência de congestionamentos em regiões específicas. Além disso, esses dados permitem a identificação de pontos geolocalizados que necessitam de melhoras, compreendendo aumento de veículos coletivos, ou até mesmo, alteração no tempo dos semáforos.

A última categoria se refere ao gerenciamento eficiente do tráfego. O maior exemplo dessa categoria são os TMSs, que buscam processar e analisar dados sobre o tráfego, e então dispor de serviços e aplicações. Para a tomada de decisão inteligente, os TMSs utilizam dados de sensoriamento urbano para identificar a situação do tráfego, e então,

disseminam essas informações na rede. Parte dos trabalhos nessa categoria consideram a comunicação V2V (Vehicle-to-Vehicle) e/ou V2I (Vehicle-to-Infrastructure) através da comunicação sem fio DSRC (Dedicated Short-Range Communication). A estratégia de disseminação dos dados é de grande importância nesses sistemas, por causa da necessidade de informar e re-rotear veículos que iriam passar pelo congestionamento para diminuir o seu impacto.

Os TMSs devem considerar a ocorrência de mudanças no clima, obras na via, demanda de tráfego, eventos como shows, entre outros. Mudanças no clima, se referem a possibilidade de chuvas fortes, a qual diminui a visibilidade dos condutores e pode gerar inundações. Obras na via podem impedir o tráfego em uma determinada faixa, ou até mesmo ocupar uma via por completo, necessitando de sinalização para mover o tráfego por outro caminho. De forma similar do que ocorre em acidentes, o tráfego pode ser levado a uma via com menor capacidade de mobilidade, podendo criar congestionamentos. Isso leva ao próximo problema que é a demanda de tráfego, ou seja, uma via não tem capacidade de compreender todos os veículos que necessitam passar por ela em um determinado momento. Por fim, a ocorrência de eventos como shows, leva uma grande quantidade de pessoas para uma determinada região com formas de locomoção diferentes. Até mesmo em regiões que não possuem histórico de congestionamento pode ser foco desse problema.

## 2.4 Sugestão de Rotas

Para que condutores possam evitar regiões congestionadas, serviços de sugestão de rotas podem ser utilizados. Esses podem tanto funcionar como um módulo de um TMS, ou uma aplicação em um dispositivo móvel. Dois exemplos são: Google Maps e Waze, que tem a capacidade de sugerir diferentes rotas, baseando-se no conhecimento sobre congestionamentos e o tempo de percurso das rotas. A principal rota sugerida é a mais rápida, pelo objetivo de diminuir o tempo gasto pelo condutor em um percurso. No entanto, outros tipos de rotas podem ser derivados de dados distintos, tais como: rota mais curta, rota mais bonita, rota mais silenciosa, rota mais simples, rota mais segura.

A rota mais curta, se refere a minimizar a distância percorrida. Mesmo não sendo muito utilizada no cenário veicular, por não considerar o tráfego, para a mobilidade de pedestres ela é de extrema importância. Isso ocorre porque pedestres geralmente não estão dispostos a percorrer grandes distância até chegar ao seu destino [50]. A rota mais bonita pode ser obtida pela mineração de *feedbacks* de indivíduos sobre regiões da cidade. Além disso, é considerada a existência de parques, monumentos, lugares históricos, entre outros. Essa rota tem o objetivo de mascarar a percepção do condutor sobre a cidade, de forma a levar o indivíduo apenas para locais com que melhorem a sua qualidade de experiência.

A rota mais silenciosa, representa o conhecimento sobre locais da cidade onde ocorrem pouca poluição sonora. Além disso, é possível também considerar a poluição do ar. Ambas possuem o objetivo de identificar regiões que possam piorar a qualidade de experiência do usuário, e então evitá-las. De forma similar ao que é encontrado na rota mais bonita, ocorre a tentativa de mascarar a percepção de poluição de alguma das duas maneiras

mencionadas. A rota mais simples é a que faz menos curvas durante um percurso. Esse tipo de rota busca diminuir o risco de acidentes por maximizar o campo de visão do usuário. A cada curva o campo de visão do usuário deve ser atualizado, pela necessidade de perceber novos elementos. Durante um percurso em uma mesma via, os elementos aparecem de maneira mais suave, diminuindo a chance de alguma surpresa durante o percurso. A Figura 2.2 ilustra tipos distintos de rotas (A, B, C, D) durante uma mesma origem e destino.

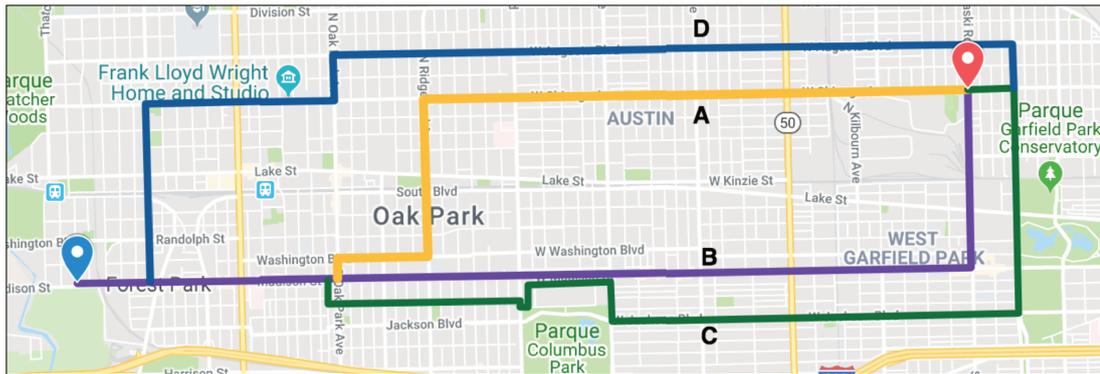


Figura 2.2: Ilustração de tipos distintos de rotas.

Pela análise de dados criminais, é possível identificar regiões com alta incidência criminal, e então sugerir rotas que não passam, ou evitam, essas regiões. A importância desse tipo de rota, é facilmente observada por ocorrências de aplicativos que levaram condutores até regiões perigosas das cidades, que resultaram na morte dos condutores [11]. Para identificar regiões inseguras, é possível utilizar dados de *crowdsourcing* ou dados oficiais disponíveis abertamente. Através do uso de algoritmos de clusterização é possível encontrar focos de crimes e regiões específicas com alta incidência criminal. Então, cada rota disponível deve ser classificada de acordo com o seu nível de insegurança, e então é possível identificar a mais segura. É necessário citar que múltiplos objetivos podem ser combinados para encontrar a melhor rota para cada determinado cenário. Na Figura 2.3, são apresentadas diferentes rotas que podem ser encontradas, dados distritos de origem e destino iguais. Na cor vermelha está representada a rota mais rápida, e em verde a rota mais segura. Os marcadores com a cor azul representam a origem, e os na cor preta o destino. Os polígonos representam regiões com alta incidência criminal.

Para descrever alguns problemas encontrados em serviços de sugestão de rotas é necessário descrever como é tratado o problema de sugerir rotas em um cenário. Por causa da facilidade em espelhar o cenário de uma cidade em um grafo, uma rota é um caminho no grafo, com um conjunto de vértices  $v_i \in V$ , e um conjunto de arestas  $e_i \in E$  conectando um par de vértices. No caso de rotas para veículos,  $G$  é considerado um grafo direcionado, o que difere da mobilidade de pedestres. Um mapa também pode ser descrito como um grafo  $M = (V', E')$ , sendo que, o conjunto de vértices  $V'$  representa as intersecções, e o conjunto de arestas  $E'$  as ruas. Dessa maneira,  $V \subset V'$  e  $E \subset E'$ . Para considerar características diferentes nas rotas, é apenas necessário utilizar um conjunto  $w \in W$  de pesos representando o tempo do percurso, a distância, a segurança, entre outros. Portanto, a rota pode ser definida como um grafo  $G = (V, E, W)$ .

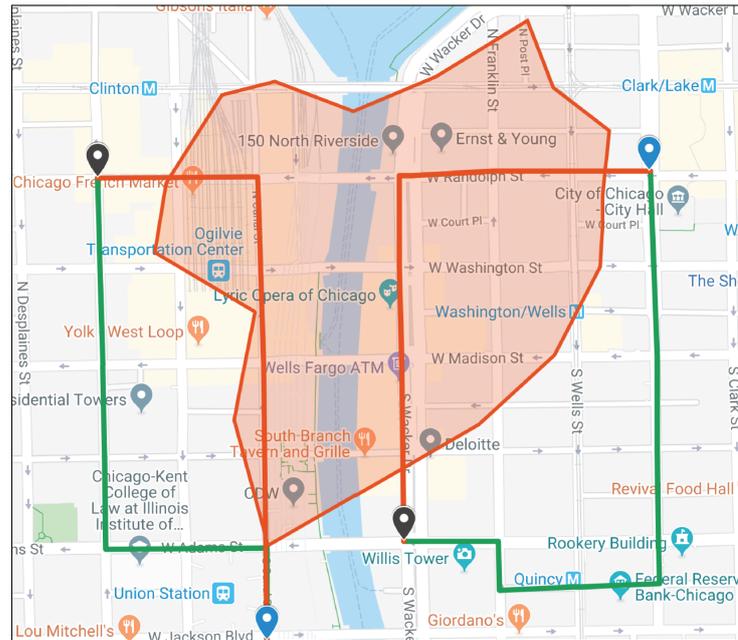


Figura 2.3: Comparação de rotas mais rápida e rotas mais segura.

É necessário aplicar alguma estratégia de balanceamento de carga para não criar congestionamentos ao enviar diversos veículos ao mesmo tempo para uma aresta. Isso ocorre pois é possível que uma mesma aresta possua o menor peso  $w$  para diversos veículos mesmo que o tráfego seja considerado. Além disso, o peso total da rota ( $w$ ) pode continuar enviando veículos para essa aresta. Outro problema é o impacto dos dados contextuais aos usuários, o qual pode variar de acordo com o perfil de cada usuário. Isso ocorre ao utilizar diversos tipos de dados contextuais sendo necessário implementar estratégias para ponderar cada tipo de dado e considerá-los durante a sugestão das rotas. Esse ponderamento é geralmente feito a cada aresta do grafo alterando o seu peso.

## 2.5 Resumo Conclusivo

Neste capítulo foram apresentados os conceitos fundamentais utilizados no desenvolvimento do trabalho. Primeiramente foram apresentadas as características dos dados contextuais utilizados considerando o comportamento observado em diferentes trabalhos da literatura. Como também, alguns serviços que fazem o uso de janelas temporais para identificar características dos dados e para caracterizar situações distintas. Após, foi apresentada a tipificação de algoritmos de aprendizado de máquina, as suas características, e alguns algoritmos foram discutidos. Uma arquitetura de sistemas de transporte inteligentes foi apresentada, mencionando as tecnologias relacionadas, e algumas aplicações. Por fim, foram discutidos alguns tipos distintos de rotas, e a maneira que são formuladas por diversos trabalhos da literatura.

# Capítulo 3

## Trabalhos Relacionados

Nesse capítulo são apresentados os trabalhos relacionados separados em 2 seções distintas, as quais compreendem: i) identificação de janelas contextuais, considera trabalhos da literatura que utilizam janelas temporais para descrever situações distintas de dados espaço-temporais, propondo estratégias de acordo com o tamanho da janela e a distribuição espacial; ii) roteamento ciente de contexto, descreve trabalhos da literatura que utilizam dados de tráfego e/ou contextuais para modificar a mobilidade urbana em cidades.

### 3.1 Identificação de Janelas Temporais

Janelas temporais servem para identificar ou caracterizar situações distintas. A identificação se refere a separar dados temporais ou espaço-temporais com maior granularidade de acordo com a variável temporal. Isso permite observar características, como por exemplo, de períodos distintos do dia. A caracterização de situações se refere a observar o comportamento dos dados e então separar em janelas distintas. Nessa seção serão apresentados trabalhos que compreendem ambos os objetivos, começando pela caracterização das situações, sendo assim, Haberman et al. [19] discutem a variação da incidência de crimes de acordo com diferentes janelas temporais, dias da semana e estações do ano. Além disso, eles analisam a ocorrência de pontos críticos na cidade da Filadélfia nos EUA. Os autores identificaram que a situação do crime muda de acordo com os períodos mencionados e devem ser considerados na identificação dos pontos críticos do crime.

O trabalho de Malleson e Andresen [26] utiliza uma janela de cilindro móvel especial para detectar pontos críticos do crime. Essa estratégia considera as dimensões do espaço como o raio do cilindro e o tempo como o comprimento do cilindro. Além disso, eles usam um Modelo de Poisson Discreto para contabilizar a população em risco, ou seja, a densidade populacional da região. Ao utilizar uma área espacial circular, todos os pontos de acesso têm uma forma circular, o que pode não ser verdade para todos os pontos de acesso. Considerando que um crime pode ser influenciado por um fator econômico, pode assumir qualquer forma de acordo com a disposição de empresas e pessoas em uma cidade.

Uma abordagem para prever crimes é proposta por Mohler e Porter [29]. Uma estratégia de grade foi aplicada para reconhecer diferentes situações criminais de pontos críticos.

A grade ideal para os dados de entrada é selecionada de acordo com um modelo supervisionado treinado para maximizar um índice. Dois parâmetros são definidos no algoritmo, a altura e a rotação da grade. Depois disso, para prever crimes, utiliza diferentes tamanhos de janelas de tempo. Como o objetivo do trabalho é prever crimes e não identificar o comportamento de tipos específicos de crimes, os pontos críticos que apresentam mais crimes são geralmente selecionados, pois têm maior probabilidade de ocorrência de novos crimes. Por esse motivo, a precisão da abordagem de Mohler e Porter [29] não se encaixa nos requisitos de nosso trabalho, considerando as formas inconsistentes de pontos ativos e as diferenças em termos de cada distribuição da criminalidade por dia da semana.

Duas abordagens para a tarefa de sugestão de rota segura são propostas por Souza et al. [7] e Santos et al. [37]. No primeiro, um sistema de gerenciamento de tráfego usa uma estimativa de densidade do *kernel* com base em dados históricos para representar a densidade do crime. Uma janela de tempo de dois anos foi usada para identificar a situação. O segundo trabalho aplica o agrupamento de dados de crimes para identificar regiões com alta incidência de crimes. Um parâmetro de vida útil é atribuído a cada tipo de dados e compreende a janela de tempo utilizada pela plataforma. Os pontos ativos têm forma circular, com as mesmas limitações de Malleson e Andresen [26].

Na Tabela 3.1, é apresentada uma comparação simplificada dos trabalhos relacionados. Essa comparação se refere a aspectos distintos, tais como: granularidade dos dados utilizados, a segmentação dos dados de acordo com variáveis temporais distintas; variável espacial, se o tamanho espacial da janela observada é variável, sendo que, *hotspots* de crimes podem ter tamanhos distintos; variável em formato, em vez de utilizar apenas o formato circular, permitir identificar *hotspots* sem forma definida; variável temporal, as situações identificadas possuem períodos de tempo de tamanhos distintos. Os trabalhos que possuem um *check mark* possuem as características discutidas, e os com um *x mark* não possuem.

Esse trabalho se difere de todos os esforços apresentados anteriormente pois, apresenta uma nova abordagem para identificar janelas contextuais. Para tal, são utilizadas características comuns em dados espaço-temporais, facilitando a sua aplicação em diversos tipos de dados. Essas janelas compreendem às mudanças no comportamento dos dados espaço-temporais. O comportamento distinto de cada tipo de dado também é direcionado nessa abordagem. É possível observar na Tabela 3.1 que são identificadas janelas de acordo com uma maior granularidade, sem formato previamente definido, e qualquer tamanho espacial e temporal.

## 3.2 Roteamento Ciente de Contexto

Há vários trabalhos na literatura que apresentam métodos para sugestão de rotas por meio da utilização de índices de fluxo de tráfego [50, 42, 35, 2]. Esses trabalhos procuram melhorar o tempo de percurso, sugerindo as rotas mais rápidas. Akabane et al. [2] propõe um sistema para diminuir o tráfego nas vias. Esse sistema é dividido em três camadas, sendo: detecção do ambiente, mecanismo de classificação do veículo e, finalmente, decisão de redirecionamento altruísta. A primeira camada obtém as informações de tráfego de

Tabela 3.1: Comparação dos trabalhos relacionados sob a perspectiva da identificação de janelas contextuais.

<b>Trabalhos relacionados</b>	<b>Granularidade de dados</b>	<b>Variável espacial</b>	<b>Variável em formato</b>	<b>Variável temporal</b>
Malleson and Andresen [26]	✓	✗	✗	✗
Mohler and Porter [29]	✗	✓	✗	✓
Souza et al. [41]	✗	✓	✓	✗
Santos et al. [37]	✓	✓	✗	✗
<b>Serviço Proposto</b>	✓	✓	✓	✓

maneira distribuída. O objetivo da segunda camada é identificar a condição de cada estrada próxima por um veículo eleito. A terceira camada baseia-se na disseminação das rotas alternativas escolhidas por cada veículo. Entretanto, as rotas rápidas sugeridas podem permitir que os veículos passem por regiões perigosas, colocando os usuários em risco.

Algumas soluções, tais como Google Maps e Waze, são capazes de sugerir rotas mais rápidas por meio da detecção de níveis de congestionamento de tráfego em tempo real, como também utilizam dados de segurança fixos para evitar locais inseguros. Assim, surge a necessidade por soluções que realizem a sugestão de rotas seguras por meio da utilização de informações contextuais que incluam a segurança. Essas informações sobre a segurança de uma região podem ser compostas por relatórios sobre as atividades criminais, dados de redes sociais e sistemas de sensoriamento participativo.

Os trabalhos [40, 10, 39] propõem sistemas que são baseados em *crowdsourcing* para utilização de informações sobre atividades criminais reportadas pelos usuários para sugerir rotas mais seguras. Já Kim et al. [23], propõe um sistema que sugere rotas seguras baseadas nas informações e percepções de sentimentos dos usuários no Twitter sobre uma localização. Em Utamima et al. [47], é proposta uma aplicação baseada em Web que utiliza uma API do Google para informar e propor novas rotas seguras aos usuários de uma determinada região. Entretanto, os trabalhos citados anteriormente não consideram o fluxo de veículos nas vias para as rotas seguras sugeridas, podendo piorar a eficiência do tráfego em áreas consideradas com maior nível de segurança. Além disso, tais trabalhos possuem uma base de conhecimento sobre crimes possivelmente incompleta e desatualizada em função das informações que são limitadas aos usuários do próprio sistema.

Já Keler et al. [22], propõem a caracterização de área inseguras por meio da utilização de informações de *crowdsourcing*, mas combinado a dados abertos do departamento de polícia sobre históricos de crime da região. Os autores mapeiam as atividades criminais como obstáculos que devem ser desviados a todo custo no momento da sugestão de rotas seguras. Entretanto, além de apresentar o mesmo problema dos trabalhos citados

anteriormente, que não consideram o tráfego dos veículos, esse trabalho não é aplicável em muitos cenários em que não é possível evitar completamente as áreas inseguras por considerá-las como um obstáculos.

Em Galbrun et al. [15], os autores utilizam dados criminais públicos com origem em uma iniciativa do governo dos Estados Unidos<sup>1</sup>, combinados a um modelo de mobilidade que inclui a distância percorrida por pedestres das cidades de Chicago e Filadélfia. Eles utilizam a soma de todas as probabilidades de uma Gaussiana *Kernel Density Estimation* (KDE) para caracterizar a densidade criminal de uma região e obter a rota mais segura. Além disso, eles consideram a melhor solução como uma função multiobjetivo que ponderam tanto distância como a segurança. A primeira limitação desse trabalho está relacionada ao custo computacional, por necessitar processar uma grande quantidade de rotas. Outra limitação é a utilização da mobilidade de pedestre no lugar da real mobilidade veicular, podendo direcionar os veículos para vias congestionadas na intenção de evitar áreas com riscos.

Souza et al. [7] propõe o *itsSAFE*, um sistema de transporte inteligente, que sugere rotas de acordo com conhecimento sobre o tráfego e a segurança das vias. O conhecimento sobre a criminalidade é obtido de uma base de dados policial que apresenta os boletins de ocorrência relatados. A seleção das rotas é mapeada como um problema de *Resource Constrained Shortest Path (RCSP)*, e é utilizado um algoritmo de Programação Dinâmica para resolvê-lo em tempo sub-exponencial. Dessa maneira, é identificada a rota mais rápida com segurança menor ou igual a um *threshold*. Nesse trabalho é considerado o cenário no qual uma região é afetada pela criminalidade, e não vias específicas. Ele não considera a evolução do cenário criminal, tratando apenas de pontos fixos com histórico de crimes.

Ambos os trabalhos em [37, 42] sugerem rotas alternativas por meio da análise contextual dos níveis de risco de segurança das vias e condições do tráfego. Nesses trabalhos, são aplicadas técnicas de clusterização sobre os dados de crimes. Esses dados sobre atividades criminais são oriundos de fontes abertas e servem para a sugestão da rota segura. Enquanto o trabalho de Santos et al. [37] considera o tempo de vida dos dados, o trabalho de Santos et al. [42] não inclui a representação da janela temporal de incidência dos dados criminais, ou seja, a relação de migração da criminalidade ao longo do tempo. Entretanto, representa as áreas afetadas pelas atividades criminais como regiões circulares e não vias específicas.

A criminalidade não é o único tipo de dado contextual que pode contribuir com uma melhor qualidade de experiência do usuário. Para exemplificar outros tipos de rotas, Quercia et al. [34] apresenta o uso de três tipos de dados contextuais, como tais: bonito, silencioso e feliz. Todo tipo de dados contextuais é obtido por meio de um site de crowdsourcing e depende do que as pessoas pensam sobre determinados lugares. Diferentemente, de outras abordagens que tentam evitar determinadas regiões, este trabalho tenta levar os usuários a uma melhor experiência de direção com base em lugares reais. No entanto, cada tipo de dado contextual é usado separadamente; conseqüentemente, três rotas são encontradas para cada par origem x destino. O principal problema é que o mesmo local que pode ser classificado por um tipo de dado pode ser classificado como os

---

<sup>1</sup><http://www.data.gov/about>

outros, e que não é contabilizado para encontrar a pontuação geral dos locais.

Esse trabalho se difere dos apresentados anteriormente, por tratar algumas das limitações dos mesmos propondo um novo serviço de sugestão. Essas limitações se referem a: identificação do comportamento, pela identificação das situações distintas que compõem um cenário; mobilidade veicular, por considerar a direção das vias e não apenas locais acessados por pedestres; tráfego nas vias, por não ignorar o tráfego e a possível ocorrência de congestionamentos; diversidade de dados, pela utilização de diversos tipos de dados contextuais para a sugestão; sugestão personalizável, por permitir que o serviço de altere de acordo com o perfil de cada usuário. Portanto, nesse trabalho é proposto um serviço que considera diversas particularidades da mobilidade veicular, características distintas que compõem um cenário, comportamento dessas características em relação as variáveis espacial e temporal, e a personalização de acordo com o perfil de cada usuário.

### **3.3 Resumo Conclusivo**

Neste capítulo foram apresentados os trabalhos da literatura relacionados ao que é proposto nesse trabalho. Sendo que é explorado o pré-processamento dos dados contextuais para obter uma granularidade menor e identificar mais rapidamente as mudanças nas situações, foram apresentadas estratégias distintas que particionam ou identificam janelas contextuais. As diferenças principais foram discutidas e apresentadas na Tabela 3.1. Em seguida, estratégias distintas foram apresentadas para propor roteamento ciente de contexto, ressaltando as suas limitações, e características da implementação.

Tabela 3.2: Comparação dos trabalhos relacionados sob a perspectiva de sugestão de rotas.

<b>Trabalhos relacionados</b>	<b>Identificação do comportamento</b>	<b>Mobilidade veicular</b>	<b>Tráfego nas vias</b>	<b>Diversidade de dados</b>	<b>Sugestão personalizável</b>
Kim et al. [23]	X	✓	X	✓	X
Utamima et al. [47]	X	✓	X	X	X
Keler et al. [22]	X	✓	X	X	X
Galbrun et al. [15]	✓	X	X	X	X
Souza et al. [7]	X	✓	✓	X	✓
Santos et al. [37]	✓	✓	✓	X	X
Souza et al. [42]	✓	✓	✓	X	X
Quercia et al. [34]	X	✓	X	✓	✓
<b>Serviço Proposto</b>	✓	✓	✓	✓	✓

## Capítulo 4

# Ambiente e Cenário das Simulações

Este capítulo tem o intuito de apresentar as ferramentas utilizadas para simulação, e os dados contextuais e de tráfego. Ele está dividido em 3 partes, as quais compreendem: apresentação do simulador de mobilidade e o cenário escolhido para ser utilizado (Seção 4.1); apresentação dos dados contextuais (Seção 4.2); e por fim, descrição da estratégia proposta para geração de padrões de mobilidade (Seção 4.3).

### 4.1 Simulador e Cenário

O trabalho trata da mobilidade veicular, mais especificamente a sugestão de rotas para distribuir o tráfego evitando pontos de congestionamentos e regiões perigosas. Para tal, é necessário utilizar um simulador de mobilidade urbana que considera diversos fatores, tais como: o posicionamento dos veículos nas vias, semáforos, velocidade máxima da via, entre outros. O Simulation of Urban MObility (SUMO) foi escolhido para tal tarefa considerando a facilidade em configurar, compatibilidade com ferramentas *opensource*, e também, disponibilidade de biblioteca para desenvolvimento de algoritmos na linguagem Python. O SUMO é um simulador *opensource* que permite a criação de diversas ferramentas para geração de cenários sintéticos, geração de padrões de mobilidade aleatórios, conversão de arquivos de cenário para outros formatos. A ferramenta *NETCONVERT* possibilita converter arquivos de cenário do serviço OpenStreetMap (OSM) [13] para o formato que o SUMO utiliza.

Portanto, é possível apenas escolher um cenário no OSM e fazer o download para converter ao formato SUMO. O cenário escolhido para ser tratado nesse trabalho é o da cidade de Chicago, por serem disponibilizados dados abertos [30], além disso, é utilizado por outros trabalhos da literatura. Esses dados compreendem diferentes aspectos da cidade, tais como: dados históricos sobre ocorrências criminais, acidentes envolvendo veículos, posição de corpos de bombeiro e hospitais, educação, saúde, saneamento, entre outros. Os dados de ocorrências criminais e acidentes envolvendo veículos são utilizados nesse trabalho com o intuito de identificar regiões perigosas e diminuir riscos a usuários. Esses serão descritos com mais detalhes na Seção 4.2. Um dado faltante para a cidade de Chicago é a mobilidade veicular durante o dia. Sendo assim, na Seção 4.3 será descrita a estratégia utilizada para gerar essa mobilidade.

Simular o cenário completo de uma cidade demanda de grande poder computacional, o qual não está facilmente disponível. Sendo assim, é necessário selecionar um cenário reduzido. Para tal tarefa, se for selecionada alguma região que não possui qualquer dado contextual, então o serviço não apresentará diferença durante a sugestão das rotas comparado com um serviço que apenas considera o tráfego. Para selecionar com maior assertividade o cenário, é possível observar através de um mapa de calor onde estão concentrados os dados contextuais na cidade. Na Figura 4.1 é apresentado o mapa de calor da cidade de Chicago com cada tipo de dado contextual. Os dados de acidentes estão mais distribuídos pela cidade, diferentemente do que acontece com os dados de crimes.

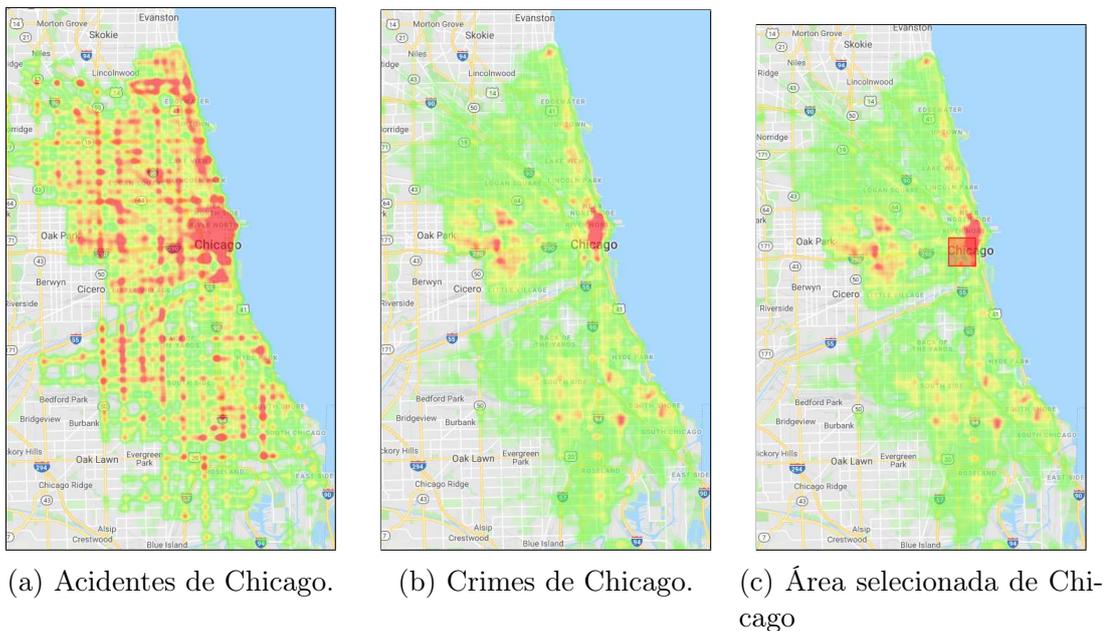


Figura 4.1: Mapa de calor de Chicago.

Portanto, foi selecionada uma área de  $5km^2$  que compreende uma região com maior incidência de crimes e acidentes. Essa região é apresentada na Figura 4.1c compreendendo o quadrado vermelho transparente. Essa região selecionada se refere ao centro da cidade de Chicago, o que já é esperado considerando a densidade de pessoas observada em grandes centros. Mesmo sendo uma intersecção de ambos os dados sobre crimes e acidentes, ainda existem áreas onde é possível percorrer sem cruzar regiões perigosas ou com histórico de acidentes. Dessa maneira, é possível verificar a capacidade do serviço de evitar locais perigosos. Na Figura 4.2 é possível observar essa região através da ferramenta do *SUMO-GUI*. É necessário efetuar uma limpeza nesse cenário pois, como o OSM é um serviço aberto que possibilita pessoas fazerem o mapeamento das vias, existem vias sem conexão com o resto do cenário. Isso permite criar rotas que não saem de um circuito fechado limitado, significativamente menor que o cenário da simulação.



Figura 4.2: Cenário selecionado de Chicago.

## 4.2 Dados Contextuais

Como já mencionado, os dados contextuais de crimes e acidentes foram obtidos da cidade de Chicago através do *Open Data Portal* [30]. A intersecção mais recente desses dados que compreende um ano completo, no momento das simulações, se refere ao ano de 2018. Por ser um ano completo, possibilita verificar a variação desses dados a cada mês. Portanto, esses dados foram escolhidos para serem utilizados. Na Figura 4.3 são apresentados dados contextuais sobre o crime de assalto, com distribuição ao longo dos dias de domingo agrupados. Esses dados estão sub-agrupados em janelas temporais fixas de 2 horas, como exemplo: início 06:00 e fim 07:59. Portanto, existem 12 barras em cada mês do ano. É possível observar a diferença entre os meses do ano em termos da quantidade de crimes e distribuição ao longo do dia, sendo que, quão mais escura a barra maior a quantidade de crimes.

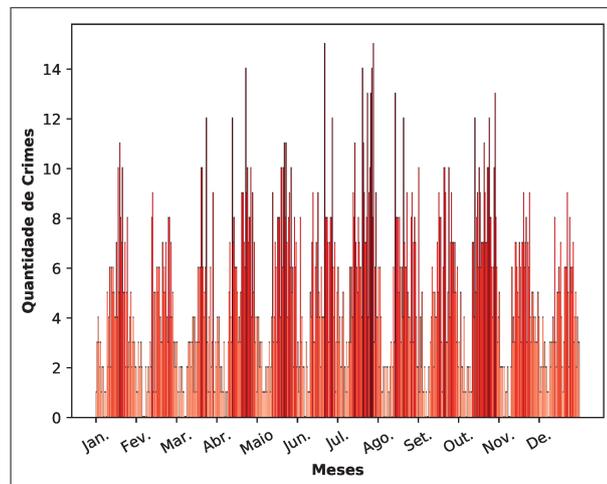
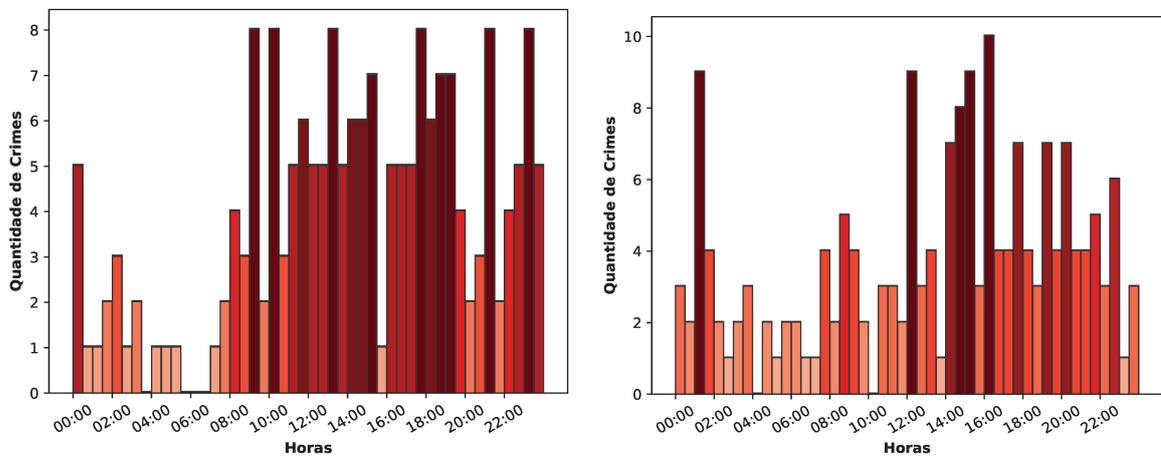


Figura 4.3: Distribuição de assalto nos domingos do ano.

Esse comportamento era esperado considerando o que foi apresentado pela literatura [12, 3, 45, 26]. A Figura 4.4 apresenta a variação na distribuição que ocorre comparando dias da semana distintos. Os dias compreendem um dia da semana e um dia de final de semana, conseqüentemente a variação se torna maior pelo comportamento das atividades cotidianas mudarem. Uma maneira de olhar para dados contextuais espaço-temporais é considerar a distribuição de eventos ao longo de uma janela temporal. Algumas características observadas a cada hora podem não estar presentes no dia seguinte. Além disso, se não houver eventos durante um certo período de tempo, isso deve ser considerado na identificação da janela de tempo. Com base nessa ideia, nesse trabalho os dados são divididos de acordo com os dias da semana, e a sub-tipificação caso seja necessário. Como exemplo, no caso de dados criminais essa sub-tipificação se refere ao tipo de crime.



(a) Distribuição na sexta-feira de Janeiro.

(b) Distribuição no sábado de Janeiro

Figura 4.4: Distribuição do crime de assalto em dias distintos.

Primeiramente os dados são separados em cada dia da semana para permitir uma análise distinta. Neste ponto, os dados da segunda-feira de um determinado mês representam a agregação de todas as segundas-feiras desse mês. De acordo com a literatura [12, 3, 45, 26], os dados do mesmo dia da semana devem comportar-se de maneira semelhante. Então, os sub-tipos de eventos podem estar correlacionados com as características sociais e econômicas das regiões da cidade, por esse motivo, os dados são divididos novamente de acordo com cada sub-tipificação de evento. Essa granularidade de dados permite identificar características mais específicas que se referem ao comportamento dos dados na cidade.

Com essa granularidade de dados é necessário executar simulações distintas para cada dia da semana pois, os dados apresentam comportamentos diferentes. Para tal, é necessário que os dados sobre a mobilidade possuam a mesma granularidade diária. Na Seção 4.3 a geração dessa mobilidade é apresentada pela criação de padrões de mobilidade observando cada situação distinta. Portanto, os dados contextuais e de mobilidade possuem um padrão.

### 4.3 Mobilidade Veicular

Existem algumas estratégias para utilizar mobilidade em simulações, tais como: uso de dados reais [48] e geração de padrões de mobilidade sintéticos. O problema da primeira é a inexistência em muitos casos de dados contextuais disponíveis abertamente, os quais são um requisito para o serviço proposto nesse trabalho. Gerar mobilidade sintética de forma aleatória torna difícil tirar conclusões sobre alguns tipos de serviços, sendo que, não se sabe características específicas dos dados. Por fim, a geração de mobilidade sintéticas pode ser genérica, se baseando em múltiplas simulações para encontrar pontos de congestionamento; ou, podem ser utilizados dados reais sobre o fluxo de tráfego para gerar o padrão de mobilidade.

A última é o caso da estratégia proposta nesse trabalho para mapear dados reais de fluxo de tráfego e então gerar padrões de mobilidade. Para isso, foi utilizada a Traffic API do serviço Here [46] que além de obter os dados de tráfego dos veículos, permite requisitar dados de tráfego considerando uma *bbox*<sup>1</sup>. Essa API dispõe de um zoom semântico para uma melhor precisão dos dados de acordo com o aumento do zoom. O retorno da API é no formato XML contendo o índice de tráfego, o nome da rua, o código da rua no formato *Point TMC Location Code*, a velocidade média no momento, entre outros dados. Com esses dados é possível verificar diferentes aspectos da situação do fluxo do tráfego no momento da requisição.

Os dados do cenário foram descritos na Seção 4.1 (formato *.osm*), e então são convertidos para um formato *.net.xml* do SUMO. Em ambos os casos o cenário é tratado como um grafo, considerando as vias como arestas, e os nós para conectar vias. A maneira que os dados de tráfego e os arquivos *.osm* são organizados seguem padrões de identificação distintos. Ou seja, o identificador de uma via no padrão do HERE não pode ser convertido diretamente para o padrão do SUMO.

Isso também vale para as vias, pois elas possuem tamanhos distintos. No caso do *.osm*, cada intersecção é tratada como uma aresta no grafo do cenário, considerando as intersecções como nós. Já no padrão do HERE, as diferentes arestas que compõem uma determinada via possuem tamanhos distintos sem considerar as intersecções. Para exemplificar algumas diferenças entre os dados, a Figura 4.5 apresenta os dados obtidos da cidade de Chicago por meio da Traffic API comparando-o com um cenário da simulação. Observa-se que há uma maior concentração de dados das vias principais, e menos nas vias dos bairros com menor extensão.

Para mapear os dados é necessário considerar os pontos geográficos de cada aresta e verificar se elas estão sobrepostas. Além disso, como a ordem dos nós se refere a direção das arestas, é possível verificar a angulação para que não seja perdida a informação sobre a direção do tráfego. Por causa disso, dois problemas foram encontrados considerando a precisão das coordenadas geográficas, sendo elas: (i) incerteza em arestas paralelas, os pontos de origem e destino de cada aresta podem estar dispostos próximos, mas não sobrepondo as arestas; e (ii) angulação entre as arestas, que pode apresentar uma pequena variação para mais ou para menos. Para resolver a incerteza, foi considerada uma distância aproximada do mapeamento real de cada aresta de 10 metros. Já para resolver a questão

---

<sup>1</sup>O *bbox* é um conjunto de duas coordenadas: limite superior esquerdo e limite inferior direito.

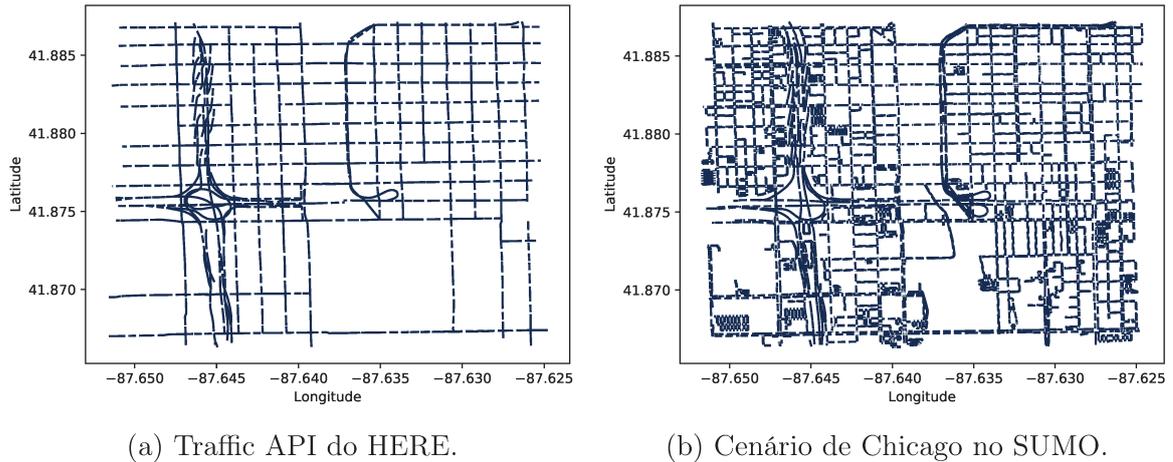


Figura 4.5: Comparação dos dados da API e do SUMO.

da angulação foi considerada uma diferença na angulação aceitável de 20 graus. Esses parâmetros foram obtidos baseados em testes empíricos. O Algoritmo 1 apresenta como é feito o mapeamento dos dados de forma a identificar a relação entre os IDs de cada um.

---

#### Algoritmo 1: MAPEAMENTO DE TRÁFEGO

---

**Entrada:**  $fluxoTrafego$ ,  $grafoCenario$

- 1 **início**
- 2      $mapeados \leftarrow \{ \}$
- 3     **para** cada  $fluxo \in fluxoTrafego$  **faça**
- 4         **para** cada  $aresta \in grafoCenario$  **faça**
- 5              $\backslash\backslash$  Verifica a diferença no angulo em relação a um limite determinado
- 6             **se**  $diferencaAngulo(fluxo, aresta) < difAceitavel$  **então**
- 7                  $\backslash\backslash$  Verifica se as arestas se intercedem
- 8                 **se**  $verificaIntersecao(fluxo, aresta)$  **então**
- 9                      $\backslash\backslash$  Faz o mapeamento entre o fluxo e a aresta
- 10                      $mapeados[aresta] = fluxo$
- 11                 **fim**
- 12             **fim**
- 13         **fim**
- 14     **fim**
- 15     **retorna**  $mapeados$
- 16 **fim**

---

Após o mapeamento dos IDs, é possível utilizar os dados do Here para visualizar o índice de tráfego em cada via do simulador. Esse resultado é apresentado na Figura 4.6. Para gerar os padrões de mobilidade a ferramenta *TrafficModeller*<sup>2</sup> foi utilizada considerando o fluxo de tráfego coletado. Essa ferramenta requer como entrada apenas um arquivo no formato *.net.xml*. Então, é possível selecionar regiões de início e fim de rotas, quantidade de veículos, e horário de saída dos veículos podendo distribuí-los ao longo do

<sup>2</sup>[https://sumo.dlr.de/docs/Contributed/SUMO\\_Traffic\\_Modeler.html](https://sumo.dlr.de/docs/Contributed/SUMO_Traffic_Modeler.html)

tempo. Para decidir quais regiões serão utilizadas, é feita a análise dos fluxos obtidos ao longo do tempo.

Essa ferramenta escolhe pontos aleatórios para início e fim de cada rota dentro das regiões. Em outras palavras, algumas rotas podem não ser válidas considerando a inexistência de um caminho do início ao fim. Isso ocorre devido aos dados do *OpenStreetMap* serem gerados por *crowdsourcing* e conseqüentemente podem possuir erros, como por exemplo, vias sem conexão com o restante do cenário. Utilizando a ferramenta *duarouter* as rotas são validadas de forma a garantir pelo menos um caminho válido em cada rota. Para simular é necessário utilizar conjuntos distintos de rotas para verificar a validade da simulação. Sendo assim, o seguinte processo<sup>3</sup> foi iterado pela quantidade de conjuntos requeridos:

- Escolher regiões no mapa baseado no fluxo observado
- Escolher aleatoriamente uma quantidade de rotas de cada região
- Embaralhar a ordem das rotas, de acordo com mesma origem e destino, para distribuir ao longo do tempo de maneira agrupada (de acordo com as regiões)
- Salvar o resultado de acordo com o padrão utilizado pelo SUMO alocando um ID para cada uma

Considerando a necessidade de avaliar o funcionamento do serviço nos momentos de maior concentração de veículos e de mudanças nos dados contextuais, as rotas são distribuídas no período de 12:00-14:00. Além disso, baseado em testes empíricos foi observada uma variação ínfima entre os resultados das simulações, portanto esse processo foi executado apenas 20 vezes para cada dia da semana. Com esses 140 padrões de mobilidade é possível avaliar a mobilidade em dias distintos e com maior precisão.

## 4.4 Resumo Conclusivo

Primeiramente foram apresentados neste capítulo o simulador de mobilidade utilizado e o cenário escolhido considerando a disponibilidade de dados abertos. Então, algumas características desses dados foram discutidas, considerando as mudanças no comportamento e a necessidade de explorar a granularidade diária dos dados. Por fim, foi apresentada a estratégia proposta para gerar padrões de mobilidade baseados em dados reais de fluxo de tráfego. Essa estratégia é o insumo utilizado em todas as simulações do trabalho.

---

<sup>3</sup>O algoritmo que executa esse processo está disponível no *GitHub* <https://github.com/lucaslzl/sumotripscleaner>



Figura 4.6: Mapeamento dos dados de tráfego.

## Capítulo 5

# Serviço de Sugestão de Rotas

Este capítulo descreve o serviço de sugestão de rotas que permite o mapeamento de dados contextuais e personalização do roteamento. É necessário citar que o serviço não é limitado pelos tipos de dados utilizados no trabalho, conseqüentemente, qualquer dado espaço-temporal que motive evitar determinadas regiões pode ser utilizado. Além disso, o mesmo pode ser utilizado em outros cenários nos quais haja a necessidade de evitar determinadas regiões dentro de uma cidade. O contexto descrito no decorrer do trabalho se refere ao comportamento dos dados que afetam direta ou indiretamente usuários.

A Figura 5.1 ilustra o funcionamento de cada módulo do serviço proposto. Cada módulo tem o objetivo de gerar insumo ao próximo, sendo que, a saída do último é a rota a ser sugerida. Portanto, os módulos são apresentados na ordem de execução. No Rótulo (A) Figura 5.1, é realizada a identificação de janelas contextuais por meio do comportamento dos dados. As janelas contextuais representam janelas de tempo que descrevem uma determinada situação. Sendo assim, dentro de um mesmo dia podem ter diversas janelas contextuais, para cada período do dia, e/ou para cada tipo de dado. O comportamento distinto para cada tipo/subtipo de dado requer a identificação de janelas contextuais de maneira isolada. Durante a sugestão de rotas é necessário verificar quais janelas de tempo descrevem a situação no momento da requisição. Na Seção 5.1 esse módulo é descrito em maiores detalhes.

O Rótulo (B) Figura 5.1 gera o mapeamento do contexto com base em KDEs (*Kernel Density Estimation*) distintas. Ao obter os dados das janelas contextuais, os eventos de cada janela são mapeados em KDEs distintas. Portanto, é necessário considerar cada KDE gerada de acordo com o tipo/subtipo dos dados contextuais. Para os dados sobre ocorrências criminais, é considerada apenas a maior probabilidade obtida pelas KDEs de cada tipo de crime. A qual permite que os dados sejam filtrados previamente apenas para os tipos criminais desejados. A ideia é relacionar com maior precisão o risco de cada tipo de dado com as vias sugeridas aos veículos.

Por fim, o Rótulo (C) Figura 5.1 realiza a personalização do roteamento que permite aos usuários configurarem quanto cada tipo de dado contextual contribui para o risco de acordo com a sua percepção. Com esse objetivo, o algoritmo *Weighted-sum* é utilizado e a ponderação de pesos feita com a estratégia *Rank sum* [18]. Essa estratégia permite que seja dado um peso de 0%-100% a cada tipo de dado, sendo tráfego, crime ou acidentes. No entanto, a soma de todos os pesos deve ser igual a 100%. Sendo assim, quão maior o

peso de um determinado tipo de dado em relação aos demais, mais o serviço tenta evitar a distribuição de probabilidade de cada região. Esse módulo e o anterior são descritos em maiores detalhes na Seção 5.2.

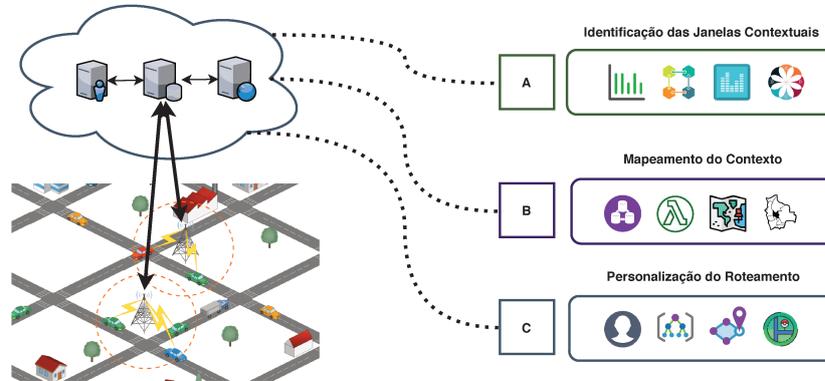


Figura 5.1: Ilustração da arquitetura do serviço proposto.

Como é possível observar na Figura 5.1, o serviço possui uma arquitetura centralizada de forma que possa ser utilizado em um servidor em nuvem, recebendo requisições dos veículos que desejam sugestões de rotas alternativas. No serviço proposto, assume-se que as requisições são transmitidas por meio da infraestrutura LTE. São consideradas 3 entidades na nuvem, as quais compreendem: um servidor que processa as requisições, um que armazena todos os dados, sejam eles históricos ou obtidos em tempo real, e por fim, um que processa os dados e sugere as rotas. É necessário citar que mesmo que sejam utilizados nesse trabalho dados contextuais históricos, é possível utilizar dados obtidos durante o dia e identificar o comportamento da janela contextual no momento da requisição.

## 5.1 Identificação do Contexto

O objetivo desta seção é de apresentar a estratégia para identificar as mudanças que ocorrem nos dados utilizados. Para propor uma abordagem genérica, este módulo considera apenas a densidade de eventos, a distribuição espacial e a sub-tipificação dos dados, se necessário. Observe que outras características de contexto podem estar sendo consideradas, mas isso está fora do escopo deste trabalho. Esta seção está dividida em 4 etapas: caracterização do contexto, apresenta como são identificadas as características dos dados (Seção 5.1.1); identificação de janela, apresenta a divisão das janelas temporais (Seção 5.1.2); filtragem das janelas temporais (Seção 5.1.3), lista as estratégias utilizadas para tal tarefa; e por fim, é feita uma breve avaliação sobre a identificação do contexto proposta na Seção 5.1.4.

### 5.1.1 Caracterização do Contexto

A granularidade dos dados descrita na Seção 4.2 é mencionada nessa e nas próximas seções como um fragmento de dados  $\rho$ . Primeiramente, é contabilizada uma pontuação para analisar a distribuição de eventos em cada  $\rho$ . Cada  $\rho$  compreende uma coleção de

$\varepsilon$  eventos, conseqüentemente  $\rho = (\varepsilon_0, \varepsilon_1, \dots, \varepsilon_n)$ , onde cada  $\varepsilon$  possui três parâmetros que compreendem as coordenadas espaciais ( $\xi_x, \xi_y$ ) e o tempo ( $\lambda$ ). Uma função de distribuição gaussiana  $\gamma$  (Equação 5.1) é utilizada para caracterizar o comportamento da distribuição de eventos de acordo com o tempo. Essa função foi escolhida pois, posteriormente será utilizado o mesmo *kernel* para mapear o contexto, o qual é utilizado por diversos trabalhos da literatura. A pontuação é calculada considerando registros de data e hora predeterminados, por exemplo, a cada  $\tau$  minutos  $\gamma$  é utilizada de acordo com o número e a distância dos eventos para esse registro de data e hora específico. Na Equação 5.2 é apresentada onde  $\delta$  é o *timestamp* referido, e  $\alpha$  é o resultado da pontuação para cada  $\delta$  específico. Pode-se observar que quão maior a quantidade de eventos próximos ao  $\delta$ , maior será a pontuação obtida.

$$\gamma(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (5.1)$$

A distribuição final de um dia compreende uma lista com a soma dos resultados da distribuição gaussiana (pontuações). O comprimento da matriz compreende  $\eta = 1440 \div \tau$ , como resultado, a matriz é representada por  $\kappa = (\alpha_0, \alpha_1, \dots, \alpha_\eta)$ . O número 1440 compreende a quantidade de minutos de um dia inteiro. Finalmente, cada valor de  $\kappa$  é normalizado pela Equação 5.3, pois somente o comportamento é importante, não a soma total. Após calcular o  $\kappa$ , é necessário identificar uma janela de tempo que considere ambos, os picos e os valores baixos dentro de  $\kappa$ . Conseqüentemente, o trabalho considera as mudanças na densidade do evento mesmo quando não existem eventos em um horário específico.

$$\alpha = \sum_{i=0}^n \gamma(\delta - \lambda_i) \quad (5.2)$$

$$norm(\alpha) = \frac{\alpha - min_\alpha}{max_\alpha - min_\alpha} \quad (5.3)$$

### 5.1.2 Identificação de Janelas Temporais

A identificação da janela recebe o parâmetro  $\kappa$  e percorre os valores vizinhos de cada valor de  $\kappa$ , se forem menores que o valor de referência, é um pico. No entanto, se houver dois picos próximos uns dos outros, apenas o maior deles é considerado. Algoritmo 2 apresenta o procedimento de identificação da janela depois de encontrar todos os picos de  $\kappa$ . Depois de encontrar os picos, o valor mínimo entre os picos é selecionado (Algoritmo 2, linha 7), se este valor for 0.0 então, a janela é dividida e é criada uma nova com o intervalo de todos os valores 0.0 até que um valor distinto seja encontrado (Algoritmo 2, linha 9). Se este valor não for 0.0, então a janela é apenas dividida (Algoritmo 2, linha 18). Considerando o exemplo de  $\tau = 10$ , o resultado da janela refere-se a uma lista com índices variando de 0 a 144. Como resultado, cada intervalo entre dois valores vizinhos compreende uma janela específica.

A saída do algoritmo refere-se a uma lista de índices e cada índice compreende a um determinado momento. Por exemplo, cada intervalo entre dois valores vizinhos da janela

---

**Algoritmo 2: IDENTIFICA AS JANELAS CONTEXTUAIS**


---

**Entrada:**  $\kappa$ , *picos*

```

1 início
2   \\Inicialização das variáveis de estado
3   janela  $\leftarrow$  [0]
4   ultimoPico  $\leftarrow$  0
5   para cada pico  $\in$  picos faça
6     \\Percorre todos os picos
7     min  $\leftarrow$  minindex( $\kappa$ [ultimoPico : pico])
8     se  $\kappa$ [min] = 0.0 então
9       janela  $\leftarrow$  min
10      zeros  $\leftarrow$  min
11      repita
12        \\Cria janela sem eventos
13        zeros = zeros + 1
14        até  $\kappa$ [zeros] = 0.0;
15        janela  $\leftarrow$  zeros - 1
16      fim
17      senão
18        janela  $\leftarrow$  min
19      fim
20      ultimoPico  $\leftarrow$  pico
21   fim
22   retorna janela
23 fim
```

---

se refere a uma situação dos eventos que não muda. Ou seja, durante aquele intervalo de tempo as mesmas regiões são afetadas pelo dado contextual utilizado, por exemplo, pela criminalidade. Isso permite que a estratégia procure por regiões de alta incidência de eventos apenas nos intervalos que possuem eventos e siga o crescimento e a diminuição da incidência dos eventos. Além disso, é mais provável que os *hotspots* de eventos apareçam em um intervalo com um pico. O fragmento de dados é então particionado, obtendo o intervalo de tempo correspondente de cada janela de tempo.

O resultado do algoritmo é ilustrado na Figura 5.2. Os dados utilizados são da cidade de Chicago, EUA, crime de homicídio e domingo de janeiro de 2018. Esse tipo de crime foi escolhido para ilustrar o resultado do algoritmo de forma simplificada, sendo que, o comportamento de outros tipos de crimes pode variar. Na figura, pode-se observar a pontuação de densidade durante o dia. Cada seta verde marca o início de uma janela que contém eventos. Além disso, cada seta vermelha marca o final de uma janela que contém eventos. Os intervalos entre uma seta vermelha e verde, nessa ordem, compreendem uma janela que não contém nenhum evento. É possível observar na imagem que durante as primeiras horas do dia ocorre o maior número de eventos criminais, ou seja, durante a madrugada. No entanto, durante o resto do dia é possível encontrar eventos de homicídio com uma quantidade próxima.

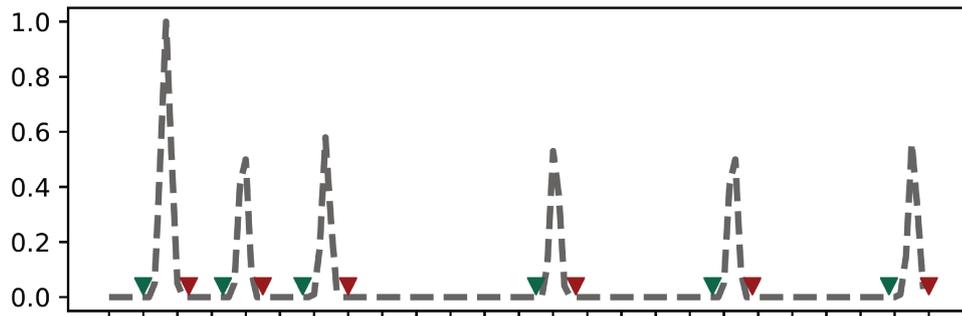


Figura 5.2: Pontuação da janela (eixo y) para homicídios em Chicago distribuídos ao longo do dia (eixo x) em 144 partes iguais.

### 5.1.3 Filtragem das Janelas Contextuais

Após a identificação das janelas temporais, um algoritmo de agrupamento é aplicado em cada intervalo da janela temporal para identificar *outliers* nos dados, ou seja, eventos que possuem uma diferença espacial significativa do resto dos dados. Durante o re-roteamento dos veículos, considerar um evento caracterizado como um *outlier* pode levar à sugestão de rotas que evitem uma determinada região que não representa perigo significativo, sendo que, a aparição de um evento naquela região possa ser raro. Para tal, um algoritmo baseado na densidade e distância tem a capacidade de cumprir essa tarefa. Dois algoritmos podem ser mencionados por serem utilizados extensamente pela literatura e possuem essas características, esses são: DBScan (Density-based spatial clustering of applications with noise) e HDBScan (Hierarchical DBScan) [20].

O HDBScan é uma extensão do DBScan que possui a capacidade de identificar qual distância mínima utilizar para alocar um evento pertencente a um grupo. Essa distância mínima varia a cada fragmento de dado. Um problema identificado é que caso exista uma quantidade de crimes significativa, no entanto, todos distantes, a distância mínima será proporcional. Isso pode fazer com que uma área extensa do mapa seja coberta por esse *cluster*. Para não permitir que isso aconteça, o DBScan é utilizado sendo possível configurar essa distância mínima de acordo com características do tipo de dado contextual utilizado. O resultado do algoritmo se refere a todos os eventos que foi possível agrupar de alguma maneira.

### 5.1.4 Avaliação da Identificação do Contexto

Com o objetivo de avaliar a capacidade de reconhecer as mudanças na densidade da estratégia proposta, a mesma é comparada com janelas temporais de tamanho fixo. Essa comparação se refere a identificar um *hotspot* após o crescimento e diminuição da incidência de crime, sendo assim, foram selecionados alguns fragmentos de dados relevantes e são comparadas a quantidade de clusters e crimes. A seleção é necessária, pois há mais de 25 tipos de crime. Mesmo que a quantidade de crimes não seja proporcional à quantidade de *hotspots*, como muitos crimes podem acontecer e serem adicionados a *hotspots* existentes, eles ainda têm uma correlação. Os fragmentos são divididos em figuras distintas, por isso é mais fácil analisar os resultados. Todos os resultados compreendem a segunda-feira de

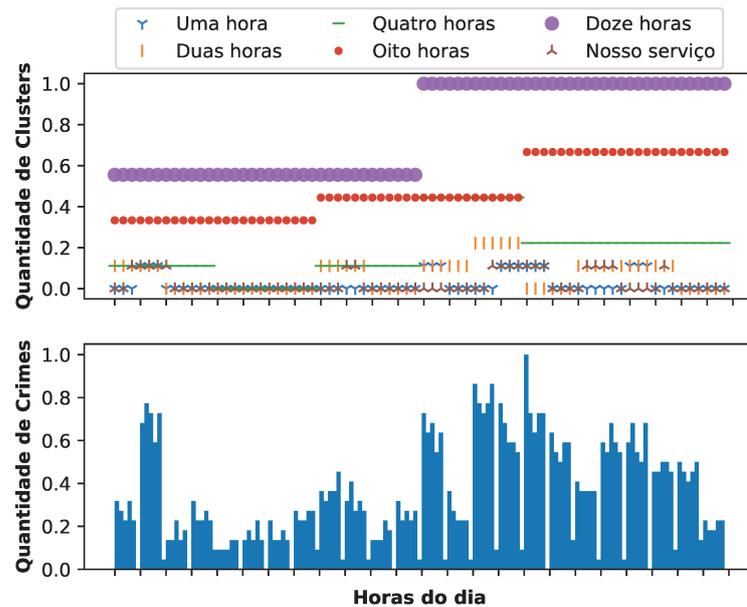


Figura 5.3: Comparação entre quantidade de clusters e ocorrências criminais de assalto.

janeiro para que não beneficiem a estratégia utilizada de forma alguma.

Primeiro, são comparados os resultados para o tipo de crime assalto e esses são apresentados na Figura 5.3. As abordagens com tamanhos de janela de tempo 4 horas ou mais consideram *hotspot* por mais tempo, embora a incidência de crime diminua. Mais especificamente, olhando para os tamanhos de janela de tempo 1 e 2 horas, eles seguem melhor a incidência do crime, porém demoram algum tempo para identificar as mudanças. Diferentemente, como a nossa estratégia considera a granularidade de 10 minutos, reconhece mais precisamente a quantidade de clusters enquanto a quantidade de crime muda. Essa diferença é mais facilmente identificada quando uma região de ponto ativo está começando ou terminando.

Algo que deve ser notado é que mesmo que a incidência do crime cresça, pode não formar mais *hotspots* dependendo de quão esparsas são as ocorrências do crime. O contrário segue a mesma regra, mas de maneira oposta, embora a incidência de crimes entre intervalos próximos tenha diminuído dependendo de quão esparsas as ocorrências do crime possam criar pontos de acesso. Além disso, o fragmento é composto pelos dados de três ou quatro dias, para o propósito de que não terá uma especificidade de apenas um dia de intervalo, permitindo ser utilizado para outros intervalos.

## 5.2 Serviço de Sugestão de Rotas

Nesta seção serão apresentados os módulos de mapeamento do contexto (Seção 5.2.1), o módulo de personalização do roteamento (Seção 5.2.2), e uma estratégia para mudar a taxa de atualização da situação para cada veículo (Seção 5.3). Ao ser requisitada uma rota, o serviço obtém as janelas contextuais identificadas no módulo de identificação do contexto e os transmite para o módulo de mapeamento de contexto. Então, o resultado do mapeamento é transmitido para o módulo de personalização do roteamento que decide

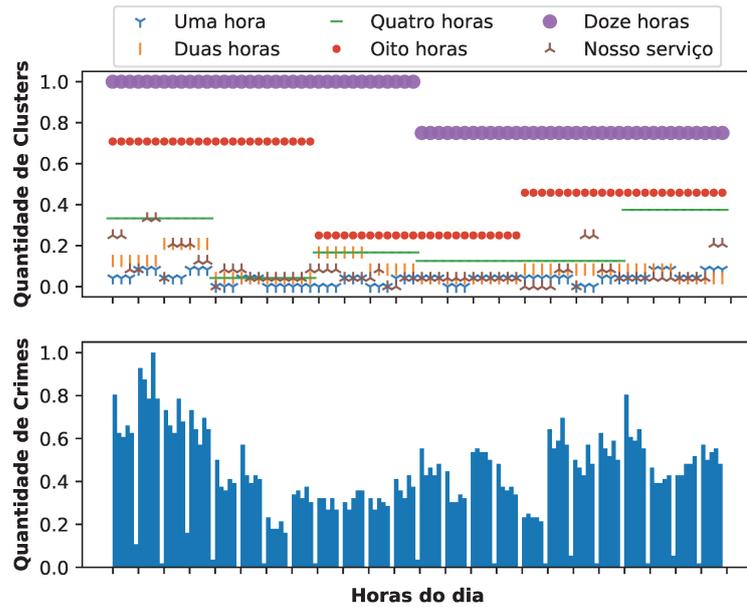


Figura 5.4: Comparação entre quantidade de clusters e ocorrências criminais de violência.

qual é a melhor rota de acordo com o perfil do usuário e a situação no momento da requisição.

### 5.2.1 Mapeamento do Contexto

Esse módulo tem o objetivo de processar as janelas contextuais obtidas durante a sugestão das rotas e gerar distribuições de probabilidade para cada uma. De forma a evitar o processamento repetido, o resultado é armazenado enquanto a janela refletir a situação dos dados contextuais. Dessa maneira, os dados filtrados e separados nas janelas são mapeados no cenário considerando o horário determinado. A distribuição utilizada é uma *Kernel Density Estimation* com Kernel Gaussiano que permite que cada evento descreva uma área de impacto, somando a cada evento próximo para aumentar o risco. Essa área de impacto cresce de acordo com a quantidade de eventos em um mesmo ponto. Seguindo a formulação descrita na Seção 5.1, uma KDE é descrita na Equação 5.4.

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^n K \frac{x - x_i}{h} \quad (5.4)$$

Na equação,  $K$  é o kernel, portanto no caso deste trabalho é o kernel Gaussiano. O valor  $h$  é um parâmetro de suavização, o qual deve ser estritamente maior que 0. O parâmetro  $x$  é o valor de referência, e por fim,  $x_i$  é um evento considerando  $(x_1, x_2, \dots, x_n)$  o conjunto de eventos. Na Figura 5.5, são apresentadas duas KDEs com os dados de criminalidade e acidentes envolvendo veículos da cidade de Chicago. O comportamento observado era esperado considerando o que foi apresentado na Figura 4.1. Na Figura 5.5a é apresentada a KDE considerando todos os tipos de crimes, todavia, durante a execução do serviço é considerada uma KDE distinta para cada tipo criminal.

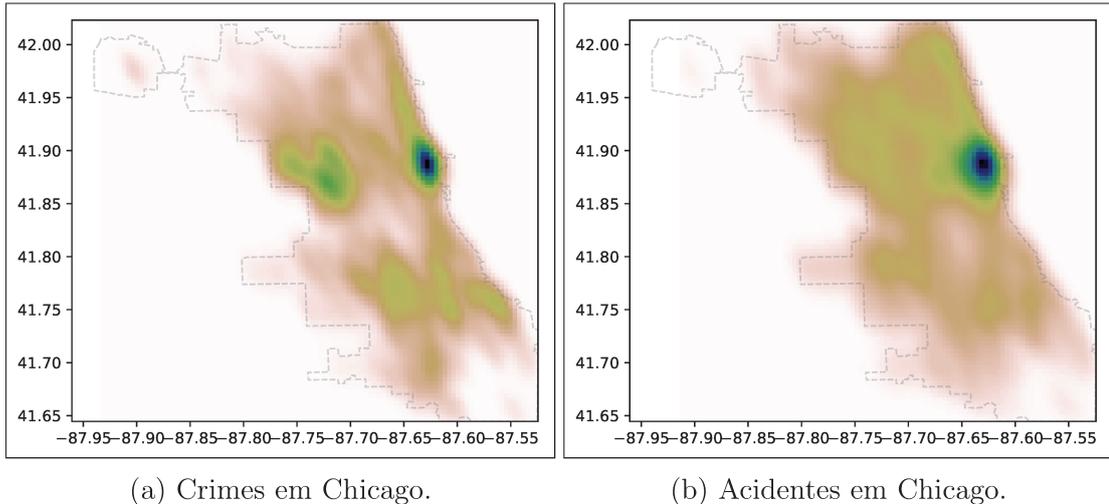


Figura 5.5: Kernel Density Estimations de diferentes dados contextuais

## 5.2.2 Personalização do Roteamento

Como já mencionado, um usuário pode ter uma necessidade diferente para evitar determinadas regiões contextuais. Para isso, esse módulo foi modelado com a estratégia de *Weighted-sum* que permite uma configuração simplificada no serviço proposto. Considerando que essa estratégia possui diversas maneiras de ponderar os pesos de cada tipo de dado, a aplicada nesse trabalho é a *Rank sum* por ser mais controlada a variação dos pesos. Como resultado, esses pesos variam de acordo com o perfil de cada usuário, por exemplo, 70% para um determinado dado de acordo com suas preferências. O intuito é que o perfil usuário descreva quanto que cada tipo de dado contribui para o risco final. O peso de cada tipo de dado contextual utilizado deve possuir soma final de 100%. No entanto, primeiro, é necessário calcular a pontuação de cada tipo de dado contextual com o intervalo entre 0 a 1 para que todos tenham a mesma variação. Nesta etapa, as informações de tráfego coletadas em tempo real são utilizadas para auxiliar a selecionar rotas. A pontuação do tráfego é obtida com base na velocidade máxima e média de cada via, variando entre 0 a 1, sendo 1 uma via congestionada. A Equação 5.5 apresenta a função de normalização utilizada para cada via  $r$ .

$$traffic_r = \frac{maxspeed_r - meanspeed_r}{maxspeed_r} \quad (5.5)$$

Um grafo ponderado de todas as vias é criado com base nas interseções (vértices) de cada via (arestas). A localização espacial de cada via é correlacionada com a pontuação  $p_i$  obtida por cada KDE em um determinado momento. Considerando a ocorrência de mais de um sub-tipo de dado, por exemplo o tipo do crime, apenas a maior pontuação é utilizada. Cada pontuação geral da via ( $r$ ) é calculada de acordo com a estratégia da soma ponderada (Equação 5.6), na qual  $n$  se refere à quantidade de dados contextuais distintos. A etapa de redirecionamento do veículo ocorre a cada  $t$  minutos, quando o veículo exige uma rota atualizada. Usando sua posição atual, o destino e o grafo das vias, o algoritmo Dijkstra é usado para encontrar o caminho mais curto (rota de pontuação

geral mais baixa).

$$geral_r = \sum_{i=1}^n max(p_i) \cdot w_i \quad (5.6)$$

Um problema que pode ocorrer nesse cenário é a criação de novos congestionamentos por levar diversos veículos próximos para uma mesma via. Sendo assim, é necessário aplicar uma estratégia de balanceamento de carga para distribuir esses veículos. A estratégia se resume em contar a quantidade de veículos re-roteados a cada via, considerando a capacidade (quantidade de faixas e tamanho da via) e aumentar o peso da via no grafo proporcionalmente ao seu percentual de carga. Tal estratégia é apresentada na Equação 5.7. Sendo  $tam$  o tamanho da via,  $qtd$  a quantidade de faixas,  $\zeta$  o tamanho médio dos veículos (nesse caso são considerados 5 metros), e  $vload$  quantos veículos já foram enviados a uma determinada via  $r$ .

$$weight_r = \frac{vload}{\frac{tam \cdot qtd}{\zeta}} weight_r + weight_r \quad (5.7)$$

### 5.3 Frequência de Atualização

Na seção anterior é descrito que os veículos podem requisitar rotas alternativas a cada  $t$  minutos, sendo necessário executar todo o processo para cada veículo. No entanto, como cada situação é identificada considerando um tempo de início e fim, é possível utilizar essa informação para diminuir a quantidade de requisições feitas à nuvem. Para manter cada usuário atualizado de acordo com a situação que está contido é preciso requisitar toda vez que ocorre o fim de uma janela contextual. Para diminuir ainda mais a quantidade de requisições é possível dispor de um intervalo máximo que extrapola o fim de cada janela contextual. Como forma de ilustração é possível observar a Figura 5.6, com diversos dados contextuais com situações distintas ao longo do dia. Os resultados dessa estratégia são apresentados na Seção 6.4.

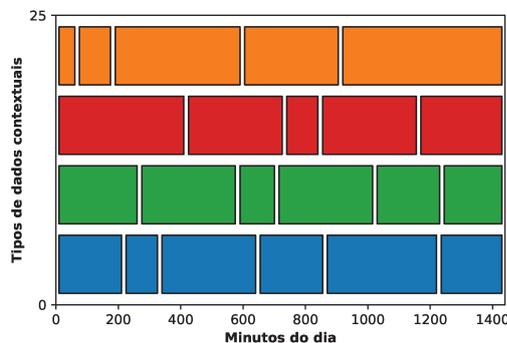


Figura 5.6: Janelas contextuais distintas ao longo do dia para diferentes tipos de dados contextuais.

## 5.4 Resumo Conclusivo

Neste capítulo foi descrito o serviço proposto para sugestão de rotas ciente de contexto e personalizável. Como o serviço é dividido em 3 módulos, cada módulo foi discutido separadamente ressaltando as suas características e a parametrização utilizada. O primeiro módulo apresentou a identificação das janelas contextuais que caracterizam situações e servem de insumo para o módulo de mapeamento de contexto. Esse outro módulo mapeia a distribuição dos dados em KDEs distintas que permitem obter o risco em cada via específica do cenário. O módulo de personalização do roteamento apresentou como foi utilizado o mapeamento de contexto, a ponderação do perfil do usuário, e a estratégia de balanceamento de carga utilizada. Por fim, uma estratégia para diminuir a quantidade de requisições ao serviço foi apresentada, essa baseada nas janelas contextuais identificadas.

## Capítulo 6

# Avaliação do Serviço

Nesse capítulo são apresentados os resultados obtidos em 2 pontos distintos, os quais compreendem: a avaliação do serviço, da mesma maneira como foi descrito no trabalho além da utilização de configurações distintas para observar o roteamento personalizado; e os resultados sobre a execução do serviço com a estratégia apresentada na Seção 5.3, esse explora a utilização de tipos de dados contextuais distintos, sem explorar a existência de tráfego.

### 6.1 Metodologia do roteamento

Na avaliação do serviço proposto, foram consideradas 14 configurações de roteamento distintas, de acordo com pesos para cada tipo de dado. A Tabela 6.1 ilustra cada tipo de configuração e seu peso. Os pesos referem-se a cada tipo de dados contextuais, consequentemente, tráfego (T), crimes (C) e acidentes envolvendo veículos (A). Nas primeiras configurações, apenas os dados contextuais determinados são utilizados ignorando qualquer aspecto dos outros dois. O restante das configurações se refere a combinações de porcentagem de cada tipo de dado, desconsiderando o último que se refere ao *baseline*. O *baseline* compreende não redirecionar os veículos, e usa apenas o padrão de mobilidade gerado inicialmente. Foram executadas 20 simulações para cada configuração distinta. Os resultados obtidos na simulação referem-se ao somatório da distribuição de probabilidade calculada com base em cada tipo de dado contextual. Esse somatório compreende uma simulação completa, ou seja, todos os resultados das distribuições calculadas durante a simulação. Então, foi calculada a média das 20 simulações considerando um intervalo de confiança de 95%.

### 6.2 Resultados do roteamento

Nesta seção, os resultados das simulações são apresentados separados por configuração. Primeiro, as métricas contextuais são discutidas para verificar a execução correta do serviço, alterando os dados contextuais considerados e verificando o impacto dessas métricas na tarefa de re-encaminhamento. Na Tabela 6.2, os resultados dessas métricas podem ser observados. Em **negrito** estão dispostos os melhores resultados de tráfego, crime, e

A			B			C		
Tráfego	Crime	Acidente	Tráfego	Crime	Acidente	Tráfego	Crime	Acidente
1,000	0,000	0,000	0,000	1,000	0,000	0,000	0,000	1,000
D			E			F		
Tráfego	Crime	Acidente	Tráfego	Crime	Acidente	Tráfego	Crime	Acidente
0,500	0,250	0,250	0,250	0,500	0,250	0,250	0,250	0,500
G			H			I		
Tráfego	Crime	Acidente	Tráfego	Crime	Acidente	Tráfego	Crime	Acidente
0,330	0,330	0,330	0,750	0,250	0,000	0,750	0,000	0,250
J			K			L		
Tráfego	Crime	Acidente	Tráfego	Crime	Acidente	Tráfego	Crime	Acidente
0,250	0,000	0,750	0,000	0,250	0,750	0,250	0,750	0,000
M			Baseline					
Tráfego	Crime	Acidente	Tráfego	Crime	Acidente			
0,000	0,750	0,250	0,000	0,000	0,000			

Tabela 6.1: Configurações de roteamento.

acidentes entre todas as configurações utilizadas. Nesse caso, pode ser observado que o menor nível de tráfego foi obtido por considerar apenas o tráfego nas vias (Config A). Uma característica que difere o tráfego dos dados contextuais, é a capacidade em mapear os estados futuros e podendo evitar congestionamentos por um gerenciamento eficiente. Isso é observado no *baseline* por não fazer gerenciamento algum de tráfego, tendo obtido o pior nível tráfego durante as simulações.

A			B			C		
<b>Tráfego</b>	Crime	Acidente	Tráfego	Crime	Acidente	Tráfego	Crime	Acidente
<b>0,806</b>	16,889	28,083	2,205	8,746	14,204	2,218	8,707	14,176
D			E			F		
Tráfego	Crime	Acidente	Tráfego	Crime	Acidente	Tráfego	Crime	Acidente
1,704	8,726	14,36	1,967	8,695	14,198	1,998	8,693	14,145
G			H			I		
Tráfego	Crime	Acidente	Tráfego	Crime	Acidente	Tráfego	Crime	Acidente
1,916	8,681	14,251	1,192	9,519	16,271	1,384	9,122	14,902
J			K			L		
Tráfego	Crime	Acidente	Tráfego	Crime	<b>Acidente</b>	Tráfego	<b>Crime</b>	Acidente
2,008	8,719	14,171	2,197	8,697	<b>14,139</b>	1,845	<b>8,59</b>	14,453
M			Baseline					
Tráfego	Crime	Acidente	Tráfego	Crime	Acidente			
2,203	8,752	14,201	3,362	12,227	19,55			

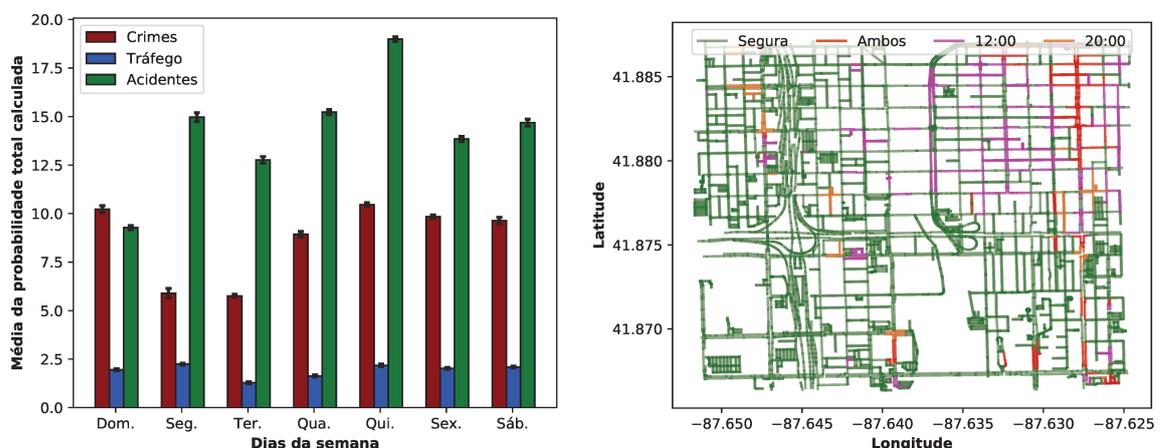
Tabela 6.2: Resultados do roteamento.

O nível de crime mais baixo foi obtido na configuração L, a qual considera a maior parte do peso para a criminalidade e uma pequena fração para o tráfego. Comparando a configuração B, que apenas considera a criminalidade, e a configuração L, foi observado o dobro de desvio padrão. Isso é uma consequência das mudanças na situação criminal que podem ocorrer durante o roteamento. Sendo assim, uma região que apresenta um nível de criminalidade baixo em um determinado momento, pode apresentar um aumento na criminalidade, afetando todos os veículos dentro da região. Algo interessante ocorre para o

nível de acidentes, sendo que, a configuração K apresenta o nível mais baixo. Observando o desvio padrão das outras configurações a maior parte é próximo do encontrado por K, caracterizando uma maior dificuldade em evitar regiões com risco de acidentes. Isso pode ser observado na Figura 5.5 (b).

As configurações podem ser observadas em grupos de acordo com a diferença entre os pesos, tais como: A, B e C (G1); D, E e F (G2); H e I (G3); J e K (G4), e L e M (G5). No caso de G2, é possível verificar que ao alterar os pesos o serviço dá a preferência para o tipo de dado majoritário, isso sem prejudicar os outros. Com o aumento do peso para o tráfego em G3, é observado um aumento no nível dos outros. Em G4, é observada a maior diferença atingindo o menor nível de acidentes envolvendo veículos entre todas as configurações. O mesmo comportamento é observado em G5 para o nível de criminalidade. Considerando tudo o que foi verificado, é possível observar a capacidade do serviço de sugerir rotas personalizadas. Mesmo apenas utilizando nesse trabalho o tráfego, criminalidade, e acidentes envolvendo veículos, o serviço pode ser estendido e utilizado com outros dados contextuais.

É possível também observar cada métrica entre os diferentes dias na semana. Para tal, é utilizada a configuração G que apresenta os mesmos pesos para todos os tipos de dados. Na Figura 6.1a é possível observar a comparação entre as diferentes métricas. A terça-feira é o dia que apresenta os menores níveis para tráfego e crimes, como também é o segundo menor para acidentes. Já a quinta-feira apresenta os maiores níveis para todos os tipos de dados. Observando apenas os crimes, é possível ver uma diminuição durante a segunda-feira e terça-feira, e então ocorre um aumento que se mantém elevado até o resto da semana. Os níveis de acidentes apresentam um maior desvio padrão relativo aos demais tipos de dados. Nesse sentido, também ocorre uma maior variação deles entre dias seguidos. Por fim, mesmo que pequena a diferença nos dados de tráfego comparada com os outros, é possível observar que a maneira que os padrões de mobilidade foram gerados permite diferenciar o tráfego entre cada dia da semana.



(a) Comparação entre os dias da semana. (b) Comparação das vias inseguras de acordo com horários distintos.

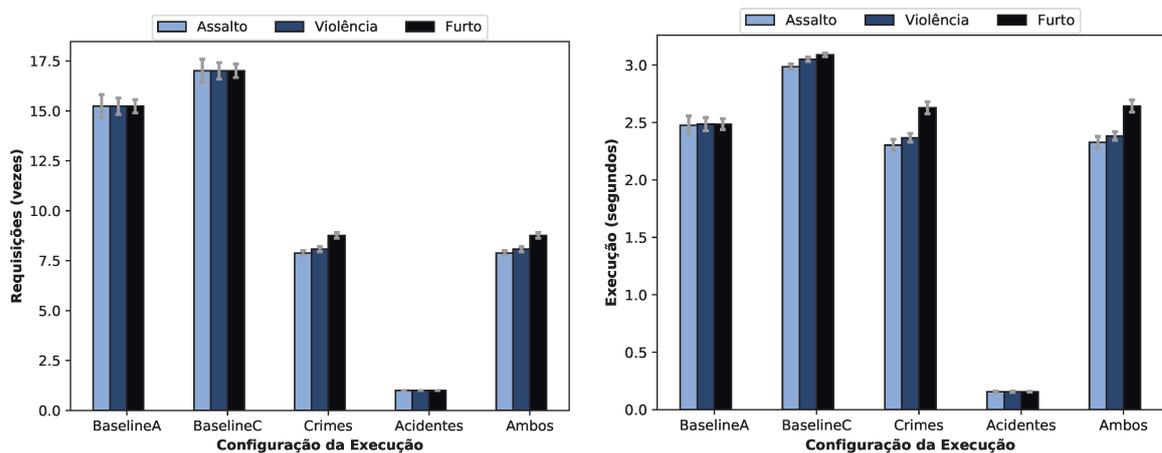
Figura 6.1: Resultados do sistema de acordo com os dias da semana e horas distintas.

Na Figura 6.1b, são apresentadas as vias do cenário, em verde as vias seguras, em rosa

as inseguras às 12:00, em laranja as inseguras às 20:00, e por fim, as em vermelho quando em ambos horários a via é insegura. É possível observar que às 12:00 é mais complicado de evitar vias perigosas em comparação com às 20:00. Portanto, é possível verificar as mudanças durante a sugestão de rotas. No serviço proposto também é utilizada uma estratégia para identificar com maior granularidade as diferentes situações que ocorrem dentro de cada dia. Portanto, o serviço tem a capacidade de identificar a situação para cada tipo de dado contextual e sugerir rotas melhores de acordo com o perfil do usuário.

### 6.3 Metodologia da taxa de atualização

Nessa seção são apresentados os resultados de acordo com a estratégia descrita na Seção 5.3, nela é feita a requisição de rotas alternativas para a nuvem apenas ao fim de alguma janela contextual. Para tal, foram considerados apenas 3 crimes de acordo com as suas quantidades, os quais compreendem: assalto (do inglês, *assault*), violência (do inglês, *battery*), e furto (do inglês, *theft*). São consideradas primeiramente 5 tipos de configurações, essas são: BaselineA, utiliza apenas os acidentes e faz a requisição a cada 5 minutos; BaselineC, utiliza apenas os crimes e faz a requisição a cada 5 minutos; Crimes, considera o fim das janelas contextuais do crime para requisitar; Acidentes, considera o fim das janelas contextuais de acidentes para requisitar; ambos, considera a intersecção entre as janelas contextuais de crimes e acidentes para requisitar. É necessário citar que foi feita uma nova simulação considerando apenas peso de 1.0, para o uso de um tipo de dado contextual, e 0.0, para a exclusão dele. Além disso, não é considerado o tráfego com o objetivo verificar as mudanças nas janelas contextuais.



(a) Quantidade de requisições por configuração.

(b) Tempo de execução por configuração.

Figura 6.2: Comparação da quantidade de chamadas e a quantidade de execuções por configuração.

## 6.4 Resultados da taxa de atualização

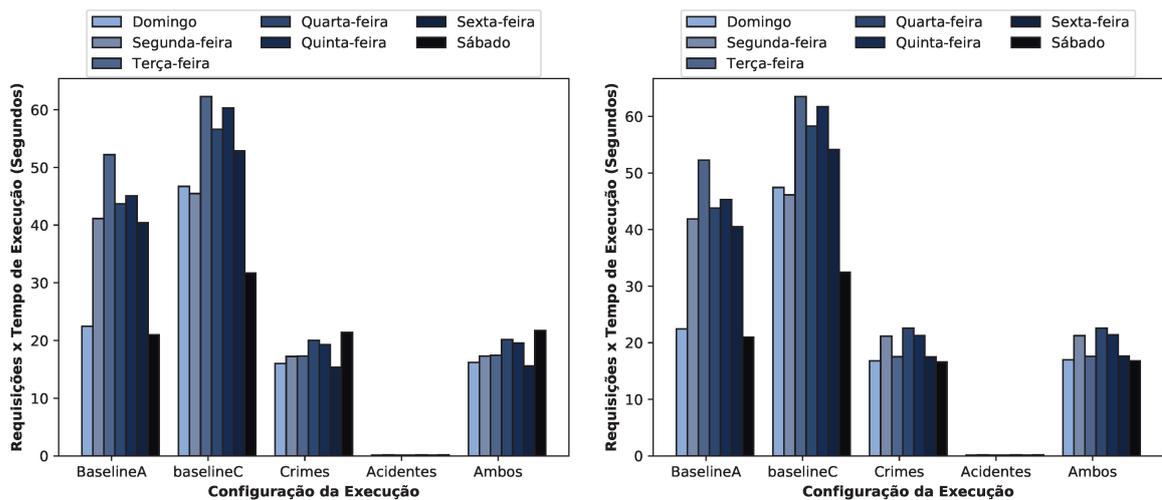
Primeiramente é comparada a quantidade de requisições e o tempo de execução para cada tipo de configuração. Essas comparações são apresentadas na Figura 6.2, e representam a média dos valores durante todos os dias da semana. É possível observar na Figura 6.2a que a quantidade de requisições nas *Baselines* é quase idêntica para todos os tipos de crimes. Isso é um reflexo da periodicidade alta das requisições. Já a quantidade de requisições de crimes e a de acidentes possuem uma grande diferença. Essa diferença exhibe que a situação de acidentes quase não muda, no caso médio, entre 12:00-14:00. Consequentemente, quando é considerada a intersecção entre os crimes e os acidentes o limite é dado pelas janelas contextuais criminais. Ao analisar a diferença entre os crimes é possível verificar que o crime de furto possui uma variação maior nas janelas contextuais, necessitando de mais requisições para manter a situação atualizada. Observando a Figura 6.2b, tem-se que o tempo de execução do crime de furto também é maior que os demais, o que implica em uma maior quantidade de eventos.

A segunda comparação refere-se ao impacto de cada crime em dias da semana distintos, e em cada configuração. O impacto compreende a quantidade de requisições multiplicada pelo tempo de execução, ou seja, o tempo médio total de processamento. Na Figura 6.3, é facilmente observada que a diferença entre as mudanças nas janelas contextuais de crimes sobrepõe ainda mais as mudanças na janelas de acidentes. Como observado anteriormente, os *baselines* apresentam maior custo computacional e de redes comparadas com a utilização de dados criminais e ambos os tipos de dados contextuais. Uma outra observação é que nos *baselines* ficam mais claros os dias que tem maior mudança no impacto, esse são: terça-feira, quarta-feira e quinta-feira. Esse resultado difere do que foi observado na Figura 6.1a, pois não é considerado o tráfego para sugerir as rotas. Os 3 tipos criminais apresentaram comportamentos similares em termos de quantidade de impacto, no entanto, ao observar os dias da semana eles possuem alta em dias distintos. No caso do crime de assalto ele ocorre mais aos sábados. Já no caso do crime de violência ele ocorre mais nas quartas-feiras. E por fim, o crime de furto ocorre mais na quintas-feiras.

Esses resultados mostram a capacidade de diminuir a quantidade de requisições ao considerar o comportamento das janelas contextuais. Além disso, de acordo com o dia da semana o impacto varia substancialmente, sendo que aos domingos e sábados ele quase se iguala ao observado apenas quando é feita a requisição quando ocorre o fim de uma janela contextual. Como não é considerado o tráfego, é possível que tenha ocorrido congestionamentos em diversas vias durante a simulação, mesmo sendo aplicado o algoritmo de balanceamento de carga. Ademais, a identificação do congestionamento se torna mais lenta. Pode-se concluir dos resultados obtidos que a estratégia apresentada na Seção 5.3 permite diminuição na quantidade de requisições, e portanto, no impacto aos recursos disponíveis de rede e computacionais.

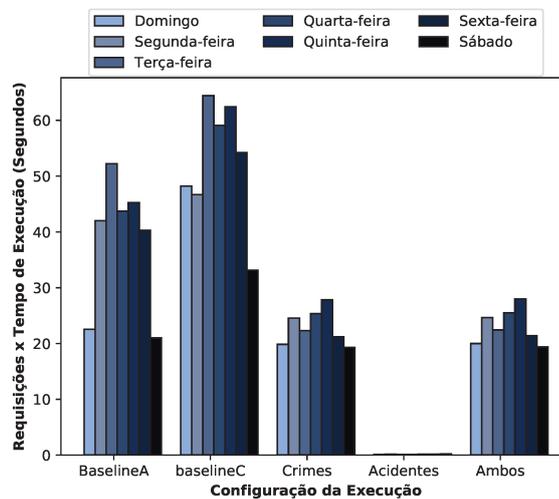
## 6.5 Resumo Conclusivo

Nesse capítulo foram apresentados os resultados obtidos de simulações para avaliar o correto funcionamento do serviço proposto, e a estratégia para diminuir a quantidade de



(a) Impacto do crime de assalto.

(b) Impacto do crime de violência.



(c) Impacto do crime de furto.

Figura 6.3: Comparação do impacto em dias da semana dos tipos criminais

requisições. Foi possível observar a capacidade de identificar rotas distintas de acordo com os diferentes perfis de usuários utilizados. Os resultados também apontam a necessidade de utilizar dados contextuais para sugerir rotas considerando a capacidade de sujeitar usuários a riscos. Além disso, foi possível observar as diferenças que ocorrem durante períodos distintos do dia. Por fim, pelo uso das janelas contextuais identificadas ocorreu uma diminuição relevante no número de requisições ao serviço.

## Capítulo 7

# Considerações Finais e Trabalhos Futuros

### 7.1 Conclusões

Este trabalho apresentou um serviço de sugestão de rotas personalizável e ciente de contexto. Para resumir seus benefícios de acordo com cada módulo, a identificação do contexto é capaz de identificar o comportamento de cada tipo de dado contextual ao longo do dia. Além disso, ele supervisiona as alterações que ocorrem usando uma Função de Distribuição Gaussiana. A filtragem dos dados também remove *outliers* obtendo dados que descrevem melhor a situação. O módulo de mapeamento do contexto caracteriza cada região da cidade com uma distribuição de probabilidade de acordo com cada janela contextual identificada. Isso permite o cálculo de cada índice de tráfego, crime, e acidentes mais acurado. Por fim, a personalização do roteamento permite que os usuários atribuam suas preferências para evitar determinadas regiões contextuais. Além disso, esse módulo balanceia o tráfego para não criar novos pontos de congestionamentos. Os resultados mostraram que o serviço se adapta até a sensíveis mudanças no peso de cada tipo de dados.

Outras contribuições compreendem a proposição de uma estratégia para mapear os dados reais de fluxo de tráfego em cenários do SUMO, e então gerar padrões de mobilidade distintos de acordo com o dia da semana, e o período do dia utilizado na simulação. A disponibilização aberta do algoritmo que aleatoriza as rotas, permite o uso em outros trabalhos de maneira simples, e permite alterações de acordo com a situação desejada. Essa contribuição é relevante para a comunidade por permitir utilizar uma maior quantidade de dados de tráfego em simulações. Portanto, a limitação é a quantidade de dados minerados na API do Here considerando a quantidade de requisições gratuitas mensalmente. É facilmente observado nos resultados a clara distinção entre os dias na semana utilizando essa estratégia. Por fim, é possível observar que utilizando as janelas contextuais o impacto nas infraestruturas de rede e computacionais é diminuído, podendo melhorar em até 50% do recurso utilizado.

## 7.2 Trabalhos Futuros

A seguir são apresentadas algumas limitações e ideias para trabalhos futuros:

- Utilizar os dados de mobilidade para gerar automaticamente rotas sem a necessidade da ferramenta TrafficModeller. Como a ferramenta não possui documentação, automatizar essa tarefa e disponibilizar o código documentado contribuiria para a disseminação da estratégia proposta.
- Considerar a situação futura no cálculo das rotas de forma a entender as mudanças durante o percurso todo. Isso permitiria diminuir a ocorrência de aumento nos níveis de criminalidade ou acidentes abruptamente durante um percurso.
- Explorar outras técnicas, além da *Weighted sum*, que permitam a personalização do serviço enquanto buscam um resultado positivo. Outros métodos podem ser aplicados para obter resultados que tentam minimizar o custo de percorrer uma via de acordo com os tipos de dados distintos.
- Simular trabalhos da literatura para comparar com o serviço proposto.

## Referências Bibliográficas

- [1] A. T. Akabane, R. L. Gomes, R. W. Pazzi, E. R. M. Madeira, and L. A. Villas. Apolo: A mobility pattern analysis approach to improve urban mobility. In *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, pages 1–6, Dec 2017.
- [2] A. T. Akabane, R. Immich, E. R. M. Madeira, and L. A. Villas. imob: An intelligent urban mobility management system based on vehicular social networks. In *2018 IEEE Vehicular Networking Conference (VNC)*, pages 1–8, Dec 2018.
- [3] Martin A. Andresen and Nick Malleson. Intra-week spatial-temporal patterns of crime. *Crime Science*, 4(1):12, Jul 2015.
- [4] Paolo Campigotto, Christian Rudloff, Maximilian Leodolter, and Dietmar Bauer. Personalized and situation-aware multimodal route recommendations: The favour algorithm. *Trans. Intell. Transport. Sys.*, 18(1):92–102, January 2017.
- [5] Felipe Cunha, Guilherme Maia, Clayson Celes, Bruno Santos, Paulo Rettore, Andre Campolina, Daniel Guidoni, Fernanda Souza, Heitor Ramos, Leandro Villas, Raquel Mini, and Antonio Loureiro. *Sistemas de transporte inteligentes: Conceitos, aplicações e desafios*, 2017.
- [6] Rodrigo Fernandes de Mello and Moacir Antonelli Ponti. *Machine Learning: A Practical Approach on the Statistical Learning Theory*. Springer, 2018.
- [7] A. M. de Souza, L. L. C. Pedrosa, L. C. Botega, and L. Villas. Itssafe: An intelligent transportation system for improving safety and traffic efficiency. In *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, pages 1–7, June 2018.
- [8] Allan Mariano de Souza and Leandro Aparecido Villas. A fully-distributed traffic management system to improve the overall traffic efficiency. In *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, MSWiM '16, pages 19–26, New York, NY, USA, 2016. ACM.
- [9] G. Dimitrakopoulos and P. Demestichas. Intelligent transportation systems. *IEEE Vehicular Technology Magazine*, 5(1):77–84, March 2010.
- [10] Samuel Elsmore, Irwan Fario Subastian, Flora Dilys Salim, and Margaret Hamilton. Vdim: Vector-based diffusion and interpolation matrix for computing region-based crowdsourced ratings: Towards safe route selection for human navigation. In *Proceedings of the 13th International Conference on Mobile and Ubiquitous Multimedia*, MUM '14, pages 212–215, New York, NY, USA, 2014. ACM.

- [11] Adriana Farias. As roubadas enfrentadas por motoristas ao seguir atalhos de aparelhos de gps. <https://vejasp.abril.com.br/cidades/roubadas-motoristas-dicas-rota-gps-waze/>, 2018. Accessed: 2018-11-21.
- [12] Marcus Felson and Rémi Boivin. Daily crime flows within a city. *Crime Science*, 4(1):31, Oct 2015.
- [13] OpenStreetMap Foundation. Open street map. <https://www.openstreetmap.org>. Accessed: 2019-07-24.
- [14] G1Rio. Morte após erro de caminho no rj repercute na imprensa internacional. <http://g1.globo.com/rio-de-janeiro/noticia/2015/10/morte-apos-erro-de-caminho-no-rj-repercute-na-imprensa-internacional.html>, 2015. Accessed: 2018-12-22.
- [15] Esther Galbrun, Konstantinos Pelechris, and Evimaria Terzi. Urban navigation beyond shortest route: The case of safe paths. *Information Systems*, 57:160 – 171, 2016.
- [16] Jose Gonsalo. How google maps helps developers to predict travel time using a new api. <https://www.toobler.com/how-google-maps-helps-developers-to-predict-travel-time-using-a-new-api/>, 2016. Accessed: 2019-07-23.
- [17] Jefferson Graham and Brett Molina. Waze sent commuters toward california wildfires, drivers say. <https://www.usatoday.com/story/tech/news/2017/12/07/california-fires-navigation-apps-like-waze-sent-commuters-into-flames-drivers/930904001/>, 2017. Accessed: 2019-06-06.
- [18] Nyoman Gunantara. A review of multi-objective optimization: Methods and its applications. *Cogent Engineering*, 5(1):1502242, 2018.
- [19] Cory P. Haberman, Evan T. Sorg, and Jerry H. Ratcliffe. Assessing the validity of the law of crime concentration across different temporal scales. *Journal of Quantitative Criminology*, 33(3):547–567, Sep 2017.
- [20] Michael Hahsler, Matthew Piekenbrock, Sunil Arya, and David Mount. Density based clustering of applications with noise (dbscan) and related algorithms. <https://cran.r-project.org/web/packages/dbscan/dbscan.pdf>. Accessed: 2019-03-14.
- [21] Yujie Hu, Fahui Wang, Cecile Guin, and Haojie Zhu. A spatio-temporal kernel density estimation framework for predictive crime hotspot mapping and evaluation. *Applied Geography*, 99:89 – 97, 2018.
- [22] Andreas Keler and Jean Damascene Mazimpaka. Safety-aware routing for motorised tourists based on open data and vgi. *Journal of Location Based Services*, 10(1):64–77, 2016.

- [23] Jaewoo Kim, Meeyoung Cha, and Thomas Sandholm. Socroutes: Safe routes based on tweet sentiments. In *Proceedings of the 23rd International Conference on World Wide Web, WWW '14 Companion*, pages 179–182, New York, NY, USA, 2014. ACM.
- [24] Christopher M. Laffoon and Magda Mourad. Build intelligent transportation systems with the traffic management data dictionary standard. <https://www.ibm.com/developerworks/industry/library/ind-tmdd/index.html>. Accessed: 2018-11-21.
- [25] Scikit learn developers. Clustering. <https://scikit-learn.org/stable/modules/clustering.html#clustering>. Accessed: 2018-11-21.
- [26] Nick Malleson and Martin A. Andresen. Spatio-temporal crime hotspots and the ambient population. *Crime Science*, 4(1):10, May 2015.
- [27] L. McInnes and J. Healy. Accelerated hierarchical density based clustering. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 33–42, Nov 2017.
- [28] George Mohler. Marked point process hotspot maps for homicide and gun crime prediction in chicago. *International Journal of Forecasting*, 30(3):491 – 497, 2014.
- [29] George Mohler and Michael D. Porter. Rotational grid, pai-maximizing crime forecasts. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 11(5):227–236, 2018.
- [30] City of Chicago. Chicago data portal. <https://data.cityofchicago.org/>. Accessed: 2019-07-24.
- [31] J. Pan, I. S. Popa, and C. Borcea. Divert: A distributed vehicular traffic re-routing system for congestion avoidance. *IEEE Transactions on Mobile Computing*, 16(1):58–72, Jan 2017.
- [32] Dom Phillips. How directions on the waze app led to death in brazil’s favelas. [https://www.washingtonpost.com/news/worldviews/wp/2015/10/05/how-directions-on-the-waze-app-led-to-death-in-brazils-favelas/?utm\\_term=.d49739a0c4ce](https://www.washingtonpost.com/news/worldviews/wp/2015/10/05/how-directions-on-the-waze-app-led-to-death-in-brazils-favelas/?utm_term=.d49739a0c4ce), 2015. Accessed: 2019-06-06.
- [33] Daniele Quercia, Rossano Schifanella, Luca Aiello, and Kate Mclean. Smelly maps: The digital life of urban smellscapes. *Proceedings of 9th International AAAI Conference on Web and Social Media (ICWSM2015)*, 05 2015.
- [34] Daniele Quercia, Rossano Schifanella, and Luca Maria Aiello. The shortest path to happiness: Recommending beautiful, quiet, and happy routes in the city. In *Proceedings of the 25th ACM Conference on Hypertext and Social Media, HT '14*, pages 116–125, New York, NY, USA, 2014. ACM.

- [35] M. Rezaei, H. Noori, D. Rahbari, and M. Nickray. Refocus: A hybrid fog-cloud based intelligent traffic re-routing system. In *2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI)*, pages 0992–0998, Dec 2017.
- [36] Christian Rudloff and Maximilian Leodolter. Personalization of routing services: User classification as a method to overcome the cold start problem for personalized models. *Transportation Research Record*, 2666(1):38–46, 2017.
- [37] F. A. Santos, D. O. Rodrigues, T. H. Silva, A. A. F. Loureiro, R. W. Pazzi, and L. A. Villas. Context-aware vehicle route recommendation platform: Exploring open and crowdsourced data. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–7, May 2018.
- [38] Frances A. Santos, Thiago H. Silva, Antonio A. F. Loureiro, Azzedine Boukerche, and Leandro A. Villas. Identificação da reputação de Áreas urbanas externas com dados de mídias sociais. *Anais do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*, 36, 2018.
- [39] Thamiris Martins Secron, Eliel Roger da Silva, Claudio Miceli de Farias, and Tiago Cruz. Sigaciente: Uma ferramenta para inferência do trânsito e de rotas seguras baseada em dados sociais. In *ERSI'2016*, pages 58–65, 2016.
- [40] Sumit Shah, Fenyue Bao, Chang-Tien Lu, and Ing-Ray Chen. Crowdsafe: Crowd sourcing of crime incidents and safe routing on mobile devices. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '11, pages 521–524, New York, NY, USA, 2011. ACM.
- [41] Allan Souza, Torsten Braun, and Leandro Villas. Efficient context-aware vehicular traffic re-routing based on pareto-optimality: A safe-fast use case. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2905–2910, 11 2018.
- [42] Allan Mariano de Souza, Torsten Braun, Leonardo C. Botega, Raquel Cabral, Islene C. Garcia, and Leandro A. Villas. Better safe than sorry: a vehicular traffic re-routing based on traffic conditions and public safety issues. *Journal of Internet Services and Applications*, 10, 2019.
- [43] G1 SP. Acidentes caem nas marginais 1 ano após redução de velocidade, diz cet. <http://g1.globo.com/sao-paulo/noticia/2016/07/acidentes-caem-nas-marginais-1-ano-apos-reducao-de-velocidade-diz-cet.html>. Accessed: 2019-09-11.
- [44] G1 SP. Marginais têm alta de acidentes nos 3 meses após aumento de limites de velocidade em sp. <https://g1.globo.com/sao-paulo/noticia/marginais-tem-alta-de-acidentes-nos-3-meses-apos-aumento-de-limites-de-velocidade-em-sp.html>. Accessed: 2019-09-11.

- [45] Lisa A. Tompson and Kate J. Bowers. Testing time-sensitive influences of weather on street robbery. *Crime Science*, 4(1):8, Apr 2015.
- [46] Here Traffic. Here traffic api. [https://developer.here.com/documentation/traffic/topics\\_v6.1/resource-parameters-flow.html](https://developer.here.com/documentation/traffic/topics_v6.1/resource-parameters-flow.html), 2019. Accessed: 2019-11-24.
- [47] Amalia Utamima and Arif Djunaidy. Be-safe travel, a web-based geographic application to explore safe-route in an area. *AIP Conference Proceedings*, 1867(1):020023, 2017.
- [48] Christian Varschen and Peter Wagner. Tapas cologne scenario. <https://sumo.dlr.de/docs/Data/Scenarios/TAPASCologne.html>, 2019. Accessed: 2019-12-11.
- [49] X. Zhang and J. Wang. Joint heterogeneous statistical-qos/qoe provisionings for edge-computing based wifi offloading over 5g mobile wireless networks. In *2018 52nd Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6, March 2018.
- [50] Y. Zheng, Y. Zhang, and L. Li. Reliable path planning for bus networks considering travel time uncertainty. *IEEE Intelligent Transportation Systems Magazine*, 8(1):35–50, Spring 2016.
- [51] Y. Zhuang, M. Almeida, M. Morabito, and W. Ding. Crime hot spot forecasting: A recurrent model with spatial and temporal information. In *2017 IEEE International Conference on Big Knowledge (ICBK)*, pages 143–150, Aug 2017.