UNIVERSIDADE FEDERAL DO CEARÁ

CENTRO DE CIÊNCIAS

DEPARTAMENTO DE COMPUTAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

DOUTORADO EM CIÊNCIA DA COMPUTAÇÃO

SAMARA MARTINS NASCIMENTO

PIPE: UM PREDITOR DE TEMPOS DE VIAGEM USANDO FLUXO CONTÍNUO DE TRAJETÓRIAS

FORTALEZA

2018

SAMARA MARTINS NASCIMENTO

PIPE: UM PREDITOR DE TEMPOS DE VIAGEM USANDO FLUXO CONTÍNUO DE TRAJETÓRIAS

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de doutor em Ciência da Computação. Área de Concentração: Ciência da Computação

Orientador: Prof. Dr. José Antônio Fernandes de Macêdo

Coorientador: Prof. Dr. Javam de C. Machado

FORTALEZA

2018

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

N198p Nascimento, Samara Martins.
PIPE: Um Preditor de Tempos de Viagem usando Fluxo Contínuo de Trajetórias / Samara Martins Nascimento. – 2018.
109 f. : il. color.

Tese (doutorado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Ciência da Computação , Fortaleza, 2018.
Orientação: Prof. Dr. José Antônio Fernandes de Macêdo.
Coorientação: Prof. Dr. Javam de Castro Machado.

1. Preditor. 2. Trajetórias. 3. Fluxos contínuos de trajetórias. I. Título.

CDD 005

SAMARA MARTINS NASCIMENTO

PIPE: UM PREDITOR DE TEMPOS DE VIAGEM USANDO FLUXO CONTÍNUO DE TRAJETÓRIAS

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de doutor em Ciência da Computação. Área de Concentração: Ciência da Computação

Aprovada em: 20 de Setembro de 2018

BANCA EXAMINADORA

Prof. Dr. José Antônio Fernandes de Macêdo (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Javam de C. Machado (Coorientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Marco Antônio Casanova
Pontifícia Universidade Católica do Rio de Janeiro (PUC)

Prof. Dra. Ticiana Linhares Coelho da Silva
Universidade Federal do Ceará (UFC)

Prof. Dr. José Maria da Silva Monteiro Filho
Universidade Federal do Ceará (UFC)

Dedico esta tese a Deus, por cuidar sempre de mim, e aos meus pais, pelo amor, pela paciência e por todo apoio que me dão.

AGRADECIMENTOS

Ao Rei dos reis: Jesus. Digno és, Senhor, de receber glória, e honra, e poder; porque tu criaste todas as coisas, e por tua vontade são e foram criadas [Ap 4:11]. Porque desde a antiguidade não se ouviu, nem com ouvidos se percebeu, nem com os olhos se viu um Deus além de ti que trabalha para aquele que nele espera [Is 64:4].

Aos meus pais, Maria Rosimar e Mario Enrique, que sempre me apoiaram e, com amor, me motivaram a superar os obstáculos.

Aos meus irmãos Danielle Martins e Fabio Martins pelo companheirismo, carinho e compreensão.

A minha sobrinha Yasmin Oliveira pelo amor e abraço restaurador em todos os meus retornos a João Pessoa.

Aos meus familiares, em especial aos meus avós, Lais Silva (in memorian), Malvina Martins e Francisco Martins (in memorian), por todo amor que me ofertaram. Estendo essa gratidão aos demais membros da minha família, que sabem como a caminhada é longa. Eu fui muito bem confortada por todos vocês.

Aos meus amigos Juciara Nepomuceno, Julianna Alencar, Ticiana Linhares, Thiago Rique, Dira Vieira, Elias Mediotte, Luiz Márcio Segundo, Wanyne Meira, Laiza Alcântara, Manuela Rezende, Paulianne do Bú e Werton Oliveira. Obrigada pela amizade, que certamente é uma benção [Pr 17-17].

Aos meus amigos da UFC: Mirla, Janaina, Camila, Thiago, Toni, David, Igo, Marcio, Régis, Bruno, Flávio, Lívia, Gustavo e Victor; Aos meus amigos da Grécia: Despina, Panos e Marios; Aos meus amigos da PUC-Rio de Janeiro: Kathrin, Marcos e Alan. Ainda agradeço a todas as pessoas não citadas aqui, mas que estiveram envolvidas nessa jornada, com maior ou menor intensidade. Obrigada pelo apoio e bons momentos compartilhados.

Ao estimado Prof. Dr. José Antônio Fernandes de Macêdo, pela oportunidade de trabalharmos juntos, pela confiança, pelo carisma e pelo investimento nesta que ingressou como aluna e agora termina como professora.

Aos competentíssimos docentes Marco Antônio Casanova, Hélio Côrtes Vieira Lopes, Yannis Theodoridis, Nikos Pelekis e Kostas Patroumpas, por todas as orientações, que me proporcionaram grande crescimento profissional.

Ao Programa de Pós-Graduação em Ciência da Computação, da Universidade Federal do Ceará (UFC) por estabelecer um ambiente de trabalho muito agradável.

Aos profissionais MDCC/UFC pela disponibilidade, colaboração e atenção, que me ofertaram ao longo dessa jornada.

Agradeço ao Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB), instituição onde trabalho, em especial aos amigos do Campus Campina, pelo acolhimento e por acreditar em minha carreira acadêmica.

Agradeço a todos os meus Professores acadêmicos que tive e tenho, o quais tentam passar o conhecimento da melhor forma possível. Em especial, aos professores da Universidade Federal da Paraíba (UFPB), primeira instituição que me incentivou a continuar pesquisando e estudando.

À Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico - FUNCAP, pelo apoio financeiro.

"Uma criança, um professor, um livro e uma caneta podem mudar o mundo. Educação é solução."

(Malala)

RESUMO

Os padrões de movimento ou anomalias no tráfego podem ser compreendidos a partir de análises relacionadas aos objetos móveis. Essas análises podem ser realizadas tanto no contexto histórico, quanto em tempo real, permitindo fazer observações acerca do tráfego, capturando suas mudanças ou detectando, de forma dinâmica, eventos ou incidentes à medida que acontecem. Dentro desse contexto, os dados de trajetória são de importância fundamental na caracterização do comportamento de objetos móveis. No entanto, computar modelos que consigam predizer o tempo de viagem dos objetos, quase em tempo real, é considerado um grande desafio. O principal impedimento está relacionado com a necessidade de apresentar as mudanças relacionadas ao tráfego, quando são recebidos novos fluxos contínuos de trajetórias. Além disso, apesar de sua enorme aplicabilidade, a utilização dos dados de trajetória não se exclui de problemas, o que justifica sua exploração extensiva pela literatura atual, apresentando estudos a respeito do processamento de grandes volumes de dados, tratamento de erros e imprecisões e a construção de modelos preditivos, que consigam, por exemplo, estimar o tempo total para alcançar um destino a partir de uma determinada origem. Este trabalho busca enfrentar o desafio de construir um novo modelo de predição, que consiga computar resultados acerca do tempo de viagem dos objetos móveis, quando reportados fluxos contínuos de trajetórias. Dentro desse contexto, este trabalho de pesquisa propõe o modelo de predição chamado *PIPE: Um Preditor de Tempos de Viagem usando Fluxo Contínuo de Trajetórias*, que pode ser utilizado para estimar o tempo de viagem de um objeto móvel para percorrer um segmento de rua específico, dada uma hora do dia. Assim, esta tese busca responder, de forma geral, duas grandes questões de pesquisa: *(i) É possível criar um modelo de predição, para estimar o tempo de viagem dos objetos, considerando um conjunto de trajetórias?*; e, ainda, *(ii) Como construir um modelo que realize manutenções incrementais, dado o recebimento de fluxos contínuos de trajetórias, e gere, como resultado, funções de predição em tempo real?*. O modelo PIPE é responsável por gerar uma função de predição e atualizá-la, dado o recebimento dinâmico dos dados. A avaliação experimental deste trabalho é conduzida para dois conjuntos de dados reais. Os resultados experimentais, que validaram o modelo proposto, mostram análises relacionadas ao tempo de processamento para construir e atualizar a função diferenciável e, também, discorrem acerca dos resultados relacionados à acurácia da solução.

Palavras-chaves: Preditor. Trajetórias. Fluxos contínuos de trajetórias.

ABSTRACT

Traffic patterns or traffic anomalies can be understood from analyzes related to moving objects. These analyzes can be performed both in historical context, as in real time, allowing you to see about traffic, capturing or detecting its changes, dynamically, events or incidents as they happen. Within this context, trajectory data are of fundamental importance in the characterization of the behavior of moving objects. However, computing models that can predict the travel time of objects, almost in real time, is considered a large challenge. The main impediment is related to the need to present traffic-related changes when new continuous flows of trajectories are received. In addition, despite its great applicability, the use of trajectory data is not excluded from problems, which justifies its extensive exploration in the current literature, presenting studies on the processing of large volumes of data, handling of errors and inaccuracies and the construction of predictive models that can, for example, estimate the total time to reach a destination from a given origin. This work tries to face the challenge of constructing a new model of prediction, which is able to compute results about the travel time of moving objects, when they are reported continuous flows of trajectories. Within this context, this research proposes the prediction model called *PIPE: A Predictor of Travel Times using Continuous Trajectory Flow*, which can be used to estimate the travel time of a moving object to travel through a specific street segment given one hour of the day. Thus, this thesis seeks to answer, in general, two most important research questions: *(i) Is it possible to create a prediction model, to estimate the travel time of the objects, considering a set of trajectories?*; and also, *(ii) How to construct a model that performs incremental maintenance, given the receipt of continuous flows of trajectories, and generate, as a result, prediction functions in real time?*. The PIPE model is responsible for generating a prediction function and updating it, given the dynamic reception of the data. The experimental evaluation of this work is conducted for two sets of data real. The experimental results, which validated the proposed solution, show analyzes related to the processing time to construct and update the derivable function, and also discuss the results related to the accuracy of the solution.

Keywords: Predictor. Trajectories. Continuous flows of trajectories.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ALGORITMOS

SUMÁRIO

1 INTRODUÇÃO

Neste capítulo, detalham-se a contextualização e os desafios relacionados à construção da proposta deste trabalho de pesquisa, assim como suas contribuições. Os desafios mostram alguns problemas relacionados à aplicabilidade dos dados de trajetória, cujos tipos de dados não estão isentos de problemas ou limitações, podendo conter, por exemplo, erros e imprecisões. Os impedimentos listados e suas possíveis soluções explicam a utilização desse tipo de dado em modelos de predições para tempos de viagens. Ademais, discorre-se sobre a proposta desta tese, as questões de pesquisa a ela relacionadas. A conclusão deste Capítulo visa resumir as contribuições obtidas neste trabalho.

1.1 Contextualização

De acordo com a Divisão de População das Nações Unidas (DIVISION POPULATION DISTRIBUTION, 2011), até 2050 espera-se um aumento de oitenta por cento no crescimento da população mundial que vive em áreas urbanas. Este crescimento apresentará grandes desafios para a mobilidade urbana — especialmente para as redes de meios de transporte e de serviços que mantêm o fluxo de pessoas e comércio, de fora e de dentro das cidades. Desta forma, estudar e entender os problemas relacionados à mobilidade urbana é essencial para o crescimento econômico do mundo, a fim de melhorar a qualidade de vida nas áreas urbanas e mitigar o impacto das alterações climáticas. Para lidar com esses desafios, é necessário estudar novos métodos e técnicas que permitam o surgimento de aplicações, as quais realizem mudanças na forma como os serviços de transporte são projetados e ofertados, bem como de inovações em planejamento urbano.

Diante desse cenário desafiador, monitorar e analisar o movimento dos veículos e das pessoas é tarefa fundamental para o entendimento da mobilidade urbana, detectando, por exemplo, padrões de movimento ou anomalias no tráfego. Essas análises podem fornecer importantes *insights* para solucionar os problemas, como detecção de congestionamentos, planejamento de rotas ou, ainda, previsão de tempo de chegada. Neste sentido, a ubiquidade de dispositivos móveis equipados com sistema de localização geográfica (e.g. *Global Positioning System* - GPS) tem permitido a geração de uma grande quantidade de dados georreferenciados no tempo, capturando a coordenada geográfica *(latitude, longitude)*, que se refere à localização do objeto móvel, e a informação temporal *(instante de tempo)* (KINOSHITA *et al.*, 2016; QUDDUS;

WASHINGTON, 2015; BRUSH *et al.*, 2010). Esses dados de localização são utilizados para construir as trajetórias dos objetos móveis e se mostram valiosos para o entendimento da evolução do movimento desses objetos no espaço e no tempo.

As análises de dados de mobilidade podem ser realizadas tanto no contexto histórico, quanto em tempo real, permitindo fazer observações acerca do tráfego e, ainda, caracterizar a movimentação dos objetos. O rastreamento dos objetos em movimento (por exemplo, pessoas e veículos) e de suas posições, que são relatadas continuamente (por exemplo, a cada segundo ou a cada minuto), permite capturar as mudanças no tráfego ou detectar, de forma dinâmica, eventos ou incidentes à medida que acontecem. Considere-se, por exemplo, um engarrafamento como um aglomerado de objetos que exibem movimentos similares no espaço e no tempo; rastreando pequenas mudanças no comportamento de objetos em movimento, é possível detectar em tempo real quando ocorre um engarrafamento e estimar, de certa forma, quanto tempo ele irá durar (LIEBIG *et al.*, 2017).

Computar modelos que consigam predizer o tempo de viagem dos objetos, quase em tempo real, é considerado um grande desafio. O principal impedimento está relacionado com a necessidade de apresentar as mudanças relacionadas ao tráfego, quando são recebidos novos dados de trajetórias. Essas mudanças precisam ser reportadas para uma função de predição, a qual é atualizada, para refletir, no segmento de rua analisado, a atual situação do tráfego. Assim, a motivação dessa tese leva a problemas de pesquisa mais amplos, que surgem quando é necessário predizer informações sobre o tráfego de uma cidade em tempo real, dado que o comportamento em grandes centros urbanos pode mudar.

1.2 Desafios

As análises dos dados de trajetória são aplicáveis em diversos contextos, a exemplo dos sistemas que demandam processamento em tempo real, como a utilização de sistemas de transportes inteligentes, que detectam, de forma dinâmica, eventos ou incidentes. Apesar de sua enorme aplicabilidade, a utilização dos dados de trajetória não é isenta de problemas ou limitações, o que justifica sua exploração extensiva pela literatura atual.

A literatura apresenta diversos estudos a respeito da descoberta de padrões frequentes em dados de trajetórias via mineração de dados, assim como o processamento de grandes volumes de dados, que podem chegar à escala de petabytes ou terabytes. Esses estudos foram extensivamente investigados em (FRENTZOS *et al.*, 2007; PANAGIOTAKIS *et al.*, 2012;

SPILIOPOULOU *et al.*, 2006; GIANNOTTI *et al.*, 2007; LEE *et al.*, 2014; HUANG *et al.*, 2008; CHEN *et al.*, 2005; GUDMUNDSSON; KREVELD, 2006; YUAN *et al.*, 2010; POTAMIAS *et al.*, 2006). Apesar desses inúmeros trabalhos, ainda restam desafios a serem superados na análise de dados de trajetórias, tais como:

- **Tratar Erros e Imprecisões**: os dados de trajetória geralmente são coletados a partir de um dispositivo de rastreamento, a exemplo do GPS, o qual é capaz de capturar um conjunto de dados em tempo real (isto é, a cada minuto ou a cada segundo). No entanto, esses dados podem conter erros de localização, quanto ao correto posicionamento do objeto. Isso ocorre devido a um conjunto de problemas, relacionados com as posições dos objetos e dos satélites, que não são corrigidas no momento em que o dado é obtido. Dentro do contexto de mobilidade de veículos, é possível aplicar técnicas de mapeamento para realizar a interpolação do correto posicionamento do objeto no espaço geográfico. Estas técnicas são mais conhecidas na literatura como *map-matching*, que corresponde a um processo de conversão, dada uma sequência de coordenadas, com *(latitude, longitude)* em uma sequência de segmentos de rua (ZHENG, 2015). Existem algumas abordagens para classificar métodos de *map-matching*, a exemplo de algoritmos *geométricos*, que combinam um ponto de GPS com o segmento de rua mais próximo (PINK; HUMMEL, 2008); e algoritmos *topológicos*, que utilizam a conectividade de uma rede rodoviária para realizar análises, cuja abordagem usa a distância de *Fréchet* para medir o ajuste entre uma sequência de pontos, obtidos via GPS, e uma sequência de segmentos de ruas candidatos (BRAKATSOULAS *et al.*, 2005). Existem, ainda, exemplos de algoritmos *probabilísticos*, que são usados para lidar com trajetórias de taxa ruidosa e de baixa amostragem (QUDDUS *et al.*, 2006; PINK; HUMMEL, 2008). Esses algoritmos realizam predições para o ruído do GPS e consideram diferentes caminhos possíveis, a partir da rede, para predizer o resultado do mapeamento.
- **Detectar e Manter Padrões de Mobilidade em Tempo Real**: o enorme volume de trajetórias abre novas oportunidades de pesquisa para possibilitar a análise dos padrões de mobilidade, considerando os objetos móveis. A literatura mostra que os padrões podem ser determinados a partir de diferentes categorias, como *padrões de movimentos juntos*, que buscam descobrir um grupo de objetos que se movem juntos por um certo período de tempo (TANG *et al.*, 2012; TANG *et al.*, 2013; ZHENG *et al.*, 2013; ZHENG *et al.*, 2014). Esses padrões podem ser diferenciados entre si de diferentes formas, como a densidade

de um grupo ou o número de objetos pertencentes a um grupo. É possível determinar padrões em comboios (JEUNG *et al.*, 2008a; JEUNG *et al.*, 2008b), enxames (LI *et al.*, 2010) ou, ainda, companheiros de viagens (TANG *et al.*, 2012; TANG *et al.*, 2013), o que ajuda no estudo das migrações de espécies ou na detecção de eventos de tráfego. Outra categoria estudada está relacionada com o *agrupamento de trajetórias*, cuja solução é usada para encontrar caminhos representativos ou tendências comuns compartilhadas por diferentes objetos em movimento. Essa estratégia, geralmente, realiza o agrupamento de trajetórias semelhantes em *clusters*. É possível, por exemplo, detectar a similaridade entre duas trajetórias com base no comprimento, na taxa de amostragem, no número de pontos e suas ordens (KHARRAT *et al.*, 2008; CADEZ *et al.*, 2000). Outro exemplo de categoria de padrões de trajetórias é a descoberta de padrões sequenciais de múltiplas trajetórias. Nesse caso, a detecção é realizada considerando certo número de objetos em movimento, que viajam em uma sequência comum, dado um intervalo de tempo similar (ZHENG *et al.*, 2011; YE *et al.*, 2009).

- **Construir Modelos de Predição**: os dados relacionados às trajetórias dos objetos móveis podem mudar continuamente. Nesse caso, é possível utilizar modelos que consigam predizer a duração dos tempos de viagens, quando novos eventos são identificados. Diversos trabalhos na literatura realizaram análises de trajetórias, criando modelos de predição para tempos de viagens, realizando análise dos dados tanto no contexto histórico, quanto em tempo real. O trabalho de (NASCIMENTO *et al.*, 2016b), que corresponde a uma das contribuições deste trabalho de pesquisa, propôs um modelo de predição, que pode ser usado para estimar os tempos de viagem dos objetos, considerando um segmento de rua. Esse trabalho estende o conceito de múltiplas regressões lineares e realiza a predição com base em dados históricos de trajetória. Há, também, trabalhos que manipulam fluxos contínuos de trajetórias para realizar análises de predição, a exemplo de (CAI; NG, 2004), que usa *polinômios de Chebyshev* para obter predições acerca dos tempos de travessia dos objetos. Ademais, (KONG *et al.*, 2016) propôs uma solução que busca estimar o congestionamento de tráfego utilizando o método chamado *Fuzzy Comprehensive Evaluation*, que utiliza uma matriz *fuzzy* de multi-índices, adaptada de acordo com o fluxo do tráfego. A proposta de (PATTARA-ATIKOM *et al.*, 2006) buscou estimar o congestionamento do tráfego usando médias exponenciais da velocidade, as quais foram obtidas a partir das informações coletadas em dispositivos de GPS. Já (YOON *et al.*, 2007) usou as informações de velocidade

espacial e temporal para estimar o status do tráfego nas ruas, considerando as localizações obtidas via GPS, enquanto (CHEN *et al.*, 2007) estimou o estado do tráfego calculando as velocidades médias ao longo do segmento de rua. Este trabalho de pesquisa também apresenta, como contribuição, dois modelos de predição que são utilizados para estimar os tempos de viagens de objetos móveis. Em (NASCIMENTO *et al.*, 2016a), é proposto um modelo incremental descontínuo, que estende a proposta de (NASCIMENTO *et al.*, 2016b). Tal modelo permite atualizar, de forma incremental, a função de regressão que é preditora de tempos de viagem, a partir do recebimento de novos dados de trajetórias. Em (NASCIMENTO *et al.*, 2017), propõe-se um outro modelo de predição, que gera uma função diferenciável; esta solução é apresentada no Capítulo 4 desta tese.

Neste trabalho, busca-se enfrentar especificamente o terceiro desafio, que trata da construção e manutenção de modelos de predição, utilizando fluxos contínuos de trajetórias. Particularmente, propõe-se uma nova solução que consegue predizer o tempo de viagem dos objetos móveis, quando reportados novos fluxos contínuos de trajetórias. O interesse por este problema foi motivado pela necessidade de utilizar modelos preditivos que suportem o processo de tomada de decisão em tempo real, fundamental para a gestão do tráfego em grandes centros urbanos.

1.3 Predição de Tempo de Viagem em Tempo Real

Na literatura, é possível encontrar diferentes propostas de modelos de predição, a exemplo de (SUN; XU, 2011; LIEBIG *et al.*, 2017), que propõem modelos baseados em distribuições Gaussianas para inferir características e comportamentos dos dados de trajetórias. E ainda, trabalhos como (YANG *et al.*, 2013; SCHNITZLER *et al.*, 2014; ASGHARI *et al.*, 2015), que usam distribuições arbitrárias, como soluções probabilísticas e modelos de Markov. Existem, também, as propostas de (KISGYÖRGY; RILETT, 2002), que construíram novos modelos de predição com base em redes neurais e regressões lineares, os quais estimam os tempos de viagens dos objetos que trafegaram em uma via. Além disso, (CORTES *et al.*, 2001; OH *et al.*, 2002; ISHAK; AL-DEEK, 2002; ZHAO *et al.*, 2016; DAI *et al.*, 2015; YANG *et al.*, 2014) propuseram modelos baseados em regressões lineares, para simular os tempos de viagem dos objetos, a partir das variações das velocidades.

O principal problema dos modelos citados anteriormente está relacionado com a necessidade de mudanças, quando são recebidos novos dados de trajetórias, os quais possibilitam

atualizar o modelo para representar a situação atual do tráfego. Nesse caso, existe a necessidade de que o modelo de predição seja alterado incrementalmente, dado que sua representação pode mudar, e ainda que a manutenção incremental ocorra é necessário continuar garantindo a obtenção correta dos resultados, sem impacto no tempo de construção do modelo. É comum encontrar na literatura propostas de aplicações reais, como controles de tráfego e planejamento de viagens, que usam funções baseadas em regressões para predizer o custo para um objeto sair de um ponto de origem e chegar a um ponto de destino (CHEN *et al.*, 2002; KUBRUSLY; LOPES, 2015; NADUNGODAGE *et al.*, 2011; PATEL *et al.*, 2016; SRIMANI; PATIL, 2014). Entretanto, (LUO *et al.*, 2015) afirma que a atualização dos modelos que usam múltiplas regressões lineares pode resultar numa função desconexa, causando uma computação recursiva no processamento da solução, o que pode ocasionar em perdas no desempenho das aplicações que a utilizam, quando este é computado no pior caso.

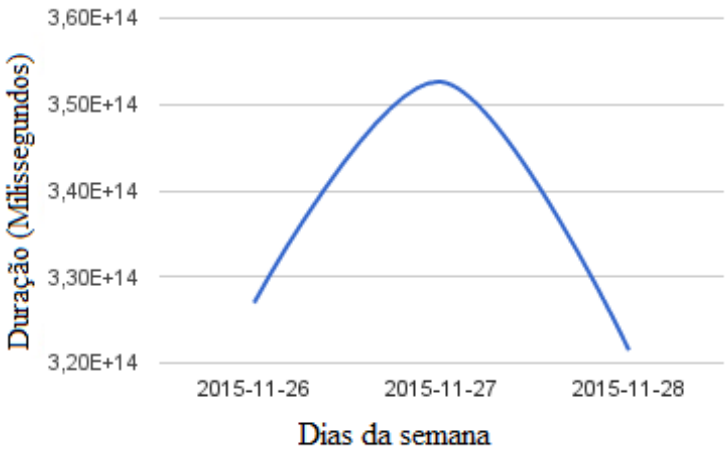
Além de predizer os tempos de viagem dos objetos, é possível utilizar as funções de predição para explicar eventos ocorridos no tráfego de uma cidade, e o resultado pode ser utilizado em sistemas de transportes inteligentes. Observe, por exemplo, a Figura 1, que mostra como uma função preditiva, criada a partir dos tempos de viagem dos objetos, pode explicar um impedimento específico causado a partir de um acidente real, que ocorreu na cidade de Fortaleza/Brasil, entre a Avenida 13 de Maio e a Rua Eusébio Souza, no dia 27 de Novembro de 2015. O *eixo x* mostra cada dia da análise, e o *eixo y* mostra o tempo total de travessia no segmento -– sendo o segmento definido como parte de uma via, pertencente à cidade de análise, e obtido a partir de dois pontos de cruzamento. Essa análise mostra o aumento das durações das viagens no dia do acidente. É possível observar que, um dia antes e um dia após o acidente, o fluxo nas vias seguiu um padrão de tempo de deslocamento. Porém, o tempo de travessia aumentou no dia em que ocorreu a colisão.

Outra análise acerca do tráfego de uma cidade é mostrada na Figura 2, a qual exibe os tempos de viagem dos objetos em avenidas específicas da cidade de Fortaleza. Tanto na Figura 2 (a), quanto na Figura 2 (b), *eixo x* representa o dia da semana em que ocorreu a análise, e o *eixo y* mostra o tempo total de travessia no segmento. O principal impedimento desse cenário está relacionado com a construção de um viaduto entre as Avenidas Raul Barbosa e Murilo Borges, para permitir o acesso da população no sentido Aeroporto/Aldeota. Essa análise compara dois diferentes momentos do ano: um no período de aulas escolares (neste caso, o mês analisado foi novembro de 2015) e outro no período de férias (neste caso, o mês analisado foi dezembro de

2015). É possível observar que, no período de férias, o fluxo do tráfego melhora em demasia e se torna próximo de um comportamento constante; diferentemente, no período de aulas, além de os resultados dos tempos de viagem serem maiores, de maneira geral, eles também oscilam mais.

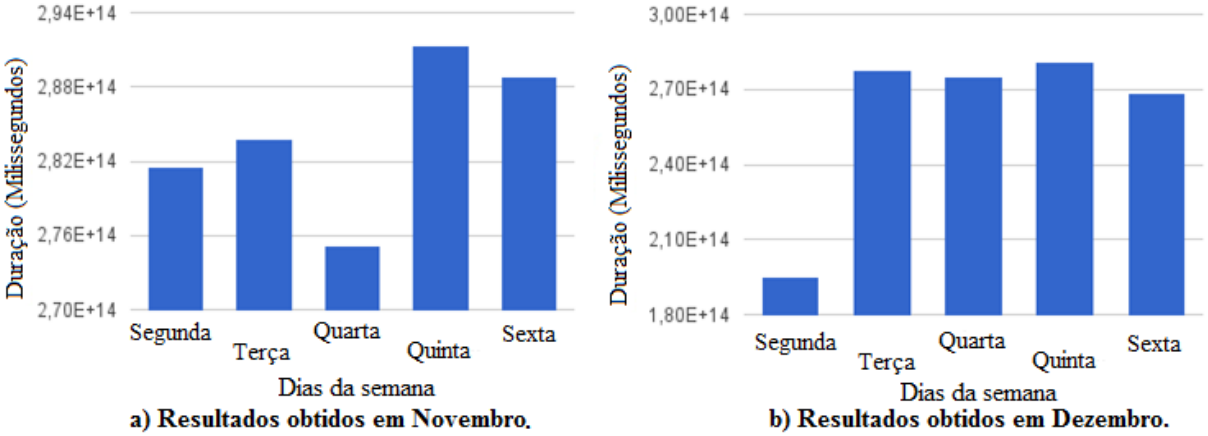
Na próxima seção, serão descritas as questões de pesquisa endereçadas para esta tese.

Figura 1 – Duração média dos tempos de viagens, quando analisado o acidente que ocorreu no dia 27/11/2015.



Fonte: da autora.

Figura 2 – Duração média dos tempos de viagens, quando analisada uma área que passou reengenharia.



Fonte: da autora.

1.4 Questões de Pesquisa

A hipótese dessa pesquisa é apresentada da seguinte forma: *"É possível criar preditores de tempo de viagens para sistemas de tempo real, que apresentem boa acurácia, usando dados de trajetórias volumosos e esparsos, os quais podem conter erros, imprecisões e valores discrepantes."*. Para isso, alguns problemas foram estudados e serão definidos a seguir. O *primeiro problema*, abordado neste trabalho de pesquisa, define um modelo de predição que consiga estimar o tempo total de viagem de um objeto móvel para percorrer um segmento de rua, dado um conjunto de trajetórias. Esse modelo deve computar uma função que recebe como argumento o instante de tempo de partida de um objeto móvel. O *segundo problema* está relacionado com a manutenção incremental do modelo de predição, dado que é preciso representar o comportamento dinâmico do tráfego no tempo. Dentro desse contexto, a proposta deve evitar recomputar o modelo, que foi descoberto anteriormente.

Dentre os desafios, o primeiro a ser superado é realizar a correção dos dados de trajetórias. Este trabalho utilizou um serviço de mapeamento para realizar o processo de limpeza e completude dos dados, dadas as observações realizadas pelos dispositivos de GPS. São descartados dados que têm apenas um ponto de localização por segmento. Isso é realizado porque não é possível inferir, com melhor acurácia, os tempos de viagem dos objetos que trafegam em uma via quando existem objetos sendo representados por apenas um ponto de localização. Outro desafio contido nesta tese está relacionado com a presença de valores discrepantes, que podem influenciar os resultados obtidos pelas soluções. Na solução proposta, que computa fluxos contínuos de trajetórias, os valores discrepantes (também chamados de *outliers*) são removidos, porque esses dados causaram resultados imprecisos de predição e causaram desbalanceamento no modelo proposto.

Considerando as observações acerca dos objetivos e desafios endereçados, é possível resumir as principais questões de pesquisa dessa tese da seguinte forma:

- **Questão de Pesquisa 1:** É possível propor um modelo de predição que compute os tempos de viagens a partir do recebimento de fluxos contínuos de trajetórias, apresentando boa acurácia?
- **Questão de Pesquisa 2:** Qual a influência dos valores discrepantes na construção de modelos preditivos?
- **Questão de Pesquisa 3:** Como manter incrementalmente um modelo de predição, dado o recebimento de novos dados de trajetórias?

- **Questão de Pesquisa 4:** Como garantir a continuidade de uma função, dada uma manutenção incremental no modelo que a gera?

Na próxima seção, serão discutidas as contribuições dessa tese, dados os problemas apresentados.

1.5 Contribuições da Tese

As contribuições relacionadas a esta tese são discutidas a seguir e construídas a partir de um conjunto de operações e análises de algoritmos. Especificamente, é possível listar:

- A construção de um modelo de predição, que possa ser usado em sistemas de tempo real e compute os tempos de viagem dos objetos móveis. A função preditiva, obtida nesta proposta, é diferenciável e tem o objetivo de melhor se ajustar à distribuição dos dados, quando essa é comparada a uma solução competidora (**Questões de Pesquisa 1 e 2)**;
- A manutenção incremental do modelo de predição e a possibilidade de resolver o problema da computação subsequente, encontrado em modelos de predição descontínuos (**Questões de Pesquisa 3 e 4)**.

O Capítulo 5 deste trabalho mostra a descrição da avaliação experimental conduzida para dois conjuntos de dados reais. O primeiro conjunto de dados, considerado para análise, corresponde às informações dos ônibus do Rio de Janeiro, Brasil. Já o segundo conjunto, considera as informações dos taxistas, da cidade de Fortaleza, Brasil. Todas as análises experimentais contêm resultados sobre a acurácia das soluções e comparações acerca dos tempos de processamento para a construção e atualização dos modelos observados.

Os resultados dessa tese foram publicados em conferências nas áreas de Banco de Dados, Sistemas de Informação, Sensores de Dados, Sistemas de Transportes Inteligentes, Análises de Dados, *Data Streams* e *Big Data* (Ver Tabela 3).

Dois modelos foram inicialmente estudados e propostos, cujos resultados não são o foco principal desta tese, mas que serviram como base para construir o modelo de predição descrito no Capítulo 4. O primeiro modelo é chamado *Batch*, que corresponde a uma solução com melhor ajuste sobre a distribuição dos dados, quando esse foi comparado com o modelo de predição competidor. Esse modelo é construído a partir de dados históricos e foi publicado em (NASCIMENTO *et al.*, 2016b). O segundo modelo foi chamado *Incremental Descontínuo*, que estende a solução *Batch* e realiza uma manutenção incremental na função preditiva, dado o recebimento de fluxos contínuos de trajetórias. Essa proposta foi publicada em (NASCIMENTO

et al., 2016a).

Os modelos descritos anteriormente – *Batch* e *Incremental Descontínuo* – serviram como base para construção do terceiro e último modelo, que corresponde a uma solução incremental contínua, chamada *PIPE: Um Preditor de Tempos de Viagem usando Fluxo Contínuo de Trajetórias*. Essa proposta buscou apresentar um melhor resultado quanto à acurácia, quando comparada com outros modelos de predição. Ademais, a solução PIPE apresentou um melhor tempo de processamento, quando modificado de forma incremental. Essa proposta compreende as **Questões de Pesquisa 1, 2, 3 e 4**, garantindo a construção de um modelo preditivo incremental, que gera como solução uma função diferenciável. Além da manutenção incremental, esse modelo não necessita do armazenamento dos dados de trajetórias, liberando espaço em disco. A proposta da solução PIPE foi publicada em (NASCIMENTO *et al.*, 2017). A Tabela 1 apresenta, de forma resumida, as publicações realizadas para este trabalho, e mostra qual melhoria foi acrescentada, a qual justificou a publicação de um novo artigo.

1.6 Proposta

Este trabalho de pesquisa propõe a solução *PIPE: Um Preditor de Tempos de Viagem usando Fluxo Contínuo de Trajetórias*, que corresponde a um modelo de predição cujo resultado busca estimar as durações das viagens a partir da utilização de fluxos contínuos de trajetórias. As respostas de predição, obtidas a partir da solução PIPE, podem ser usadas para realizar análises de mobilidade e tomadas de decisões.

O modelo proposto corresponde a uma solução incremental, que computa o resultado a partir de uma árvore binária e possibilita obter uma função diferenciável. Essa proposta possui uma natureza incremental, cuja solução pode ser modificada dado o recebimento de fluxos contínuos de trajetórias. A função diferenciável é contínua e tem o objetivo de computar, com melhor acurácia e eficiência, o tempo de viagem dos objetos, considerando cada via da cidade, dada uma hora de partida.

Os modelos *Batch* e *Incremental Descontínuo* (NASCIMENTO *et al.*, 2016b; NASCIMENTO *et al.*, 2016a) permitem realizar análises sobre os dados tanto no contexto histórico, quanto em tempo real e são baseadas em múltiplas regressões lineares. O principal impedimento da solução que permite a manutenção incremental – modelo *Incremental Descontínuo* (NASCIMENTO *et al.*, 2016a) – está relacionado com a descontinuidade da função preditiva alcançada, a qual pode causar degradação no desempenho do algoritmo que a utiliza (LUO *et al.*,

Tabela 1 – Catálogo de publicações realizadas para este trabalho

Título do Artigo	**Ano**	**Veículo**	**Resumo/Melhoria**
On computing temporal functions for a time-dependent networks using trajectory data	2016	IDEAS 2016: 20th International Database Engineering & Applications Symposium	1. Modelo *Batch*; 2. Proposta de um modelo de regressão, que gera uma solução contínua; 3. Utiliza exclusivamente dados históricos.
On computing temporal functions for time-dependent networks using trajectory data streams	2016	Proceedings of the 7th ACM SIGSPATIAL International Workshop on GeoStreaming	1. Modelo *Incremental Descontínuo*; 2. Proposta de um modelo de regressão incremental, que gera uma solução descontínua; 3. Utiliza fluxos contínuos de trajetórias.
On computing travel time functions from Trajectory Data Streams	2017	Proceedings of the 8th ACM SIGSPATIAL International Workshop on GeoStreaming	1. Modelo PIPE; 2. Proposta de um modelo incremental, que gera uma solução contínua; 3. Utiliza fluxos contínuos de trajetórias.

Fonte: da autora.

2015). Nesse caso, o autor busca utilizar árvores binárias que armazenam regressões lineares simples, dados intervalos específicos de tempo. O trabalho de (GOLOVCHENKO, 2004) mostra que a natureza incremental dos dados, pode modificar múltiplas regressões lineares, as quais geram a descontinuidade no modelo, dado que não é possível estimar um ponto de quebra em comum entre as regressões, que garanta a continuidade da solução. Dentro desse contexto, este trabalho de pesquisa propõe o modelo PIPE, que tem como resultado uma função diferenciável e estritamente contínua, o qual permite ser modificado, ainda que receba novos fluxos contínuos de trajetórias, cuja solução pode ser utilizada para representar a situação atual do tráfego.

1.7 Organização da Tese

Esta tese está organizada em seis Capítulos, os quais são descritos a seguir:

- *Capítulo 2 - Conceitos Básicos:* Neste capítulo, são apresentadas as informações preliminares utilizadas neste trabalho de pesquisa, que correspondem a definições importantes empregadas ao longo de todo o documento e são essenciais para o entendimento desta tese.
- *Capítulo 3 - Estado da Arte:* Este capítulo apresenta a revisão bibliográfica referente às propostas que se aproximam deste trabalho de pesquisa. Em particular, é apresentado um conjunto de trabalhos dentro do contexto da Análise de Dados de Trajetórias; Funções Preditivas Contínuas e Descontínuas; e Modelos de Árvores, cujos estudos serviram como base para o entendimento da solução obtida nesta tese.
- *Capítulo 4 - PIPE: Um Preditor de Tempos de Viagem usando Fluxo Contínuo de Trajetórias:* Neste Capítulo, é apresentada a solução PIPE, que se baseia na construção de uma árvore binária. O processo de busca na árvore permite obter uma função diferenciável, a qual pode ser modificada, dado o recebimento de fluxos contínuos de trajetórias.
- *Capítulo 5 - Avaliação Experimental:* Este capítulo descreve os experimentos realizados para validar o modelo PIPE e discute os resultados alcançados, os quais estão relacionados às análises do desempenho e da acurácia da solução. Todas as análises utilizaram tipos de dados estritamente reais, os quais correspondem às informações relacionadas com os ônibus da cidade do Rio de Janeiro, Brasil; e os táxis de Fortaleza, Brasil.
- *Capítulo 6 - Conclusão:* Este capítulo traz um resumo do que foi apresentado nesta tese, bem como algumas considerações finais sobre a pesquisa realizada e, também, algumas indicações de trabalhos futuros.

2 CONCEITOS BÁSICOS

Neste capítulo, são apresentadas as informações preliminares utilizadas neste trabalho de pesquisa. Definições empregadas ao longo de todo o documento e consideradas importantes para o entendimento desta tese são detalhadas a seguir. Assim, discorre-se sobre os conceitos de Trajetórias, Fluxo Contínuo de Trajetórias, Mapeamento de Objetos, Modelos de Regressão e Árvores Binárias Rotuladas. Por fim, a Seção 2.6 conclui este capítulo.

2.1 Trajetórias

Os avanços nas técnicas de aquisição sobre a localização de objetos móveis permitiram gerar grandes volumes de dados, que contêm informações sobre a mobilidade de uma diversidade de objetos, como pessoas, veículos e animais. A partir desse conjunto de dados, muitas técnicas foram propostas com o intuito de processar, gerenciar e minerar dados de trajetória, promovendo uma ampla gama de aplicações (ZHENG; ZHOU, 2011). Nesta Seção, vai-se discorrer sobre os principais conceitos relacionados aos dados de trajetórias, sendo esse tipo de dado o foco central das análises realizadas neste trabalho de pesquisa.

O modelo de trajetória tradicional mostra o movimento dos objetos, no espaço geográfico, ao longo do tempo. Assim, uma trajetória pode ser definida como uma evolução espaço-temporal dos objetos móveis. No contexto deste trabalho, uma trajetória é definida como o conjunto de posições sucessivas, ocupadas por um objeto móvel no decorrer do tempo, considerando uma *origem* e um *destino*. A *origem* do deslocamento de um objeto é a posição inicial do percurso; é onde ocorre o início do movimento. Já o *destino* representa a posição final do percurso. Dentro desse contexto, a trajetória é a evolução da posição de um objeto, correspondendo a um conjunto finito de posições no tempo, até atingir o destino final. Os conceitos de deslocamento geográfico e trajetória de um objeto são formalizados a seguir e definidos com base nos trabalhos de (NETO *et al.*, 2017; SILVA *et al.*, 2016a).

Definição 1 (Trajetória de um Objeto) Dada uma sequência $< t_1, \cdots, t_n >$ de instantes de tempo, tal que $t_i < t_j$ para $1 \leq i < j$, sendo $j \leq n$, a *trajetória de um objeto* o_j é uma sequência de pontos ordenados temporalmente $TR_j =< P_1, \cdots, P_n >$. Nesse caso, P_1 representa a *origem* do deslocamento e P_n representa o *destino*. Cada P_i $(1 \leq i \leq n)$ corresponde a um tupla, definida por $P_i = (x_i, y_i, t_i)$.

A partir deste ponto, este trabalho chamará trajetória de um objeto apenas de trajetória. A partir da definição acima, é imediata a definição de sub-trajetória.

Definição 2 (Sub-Trajetória) Considere uma trajetória TR_j. Suponha que de TR_j seja extraída uma *substring* $ST_j =< p_a, ..., p_b >$, where $1 \leq a \leq b \leq n$. Assim, ST_j é uma *Sub-Trajetória* de TR_j.

As consultas realizadas sobre dados de trajetórias podem ser muito demoradas, o que exige a aplicação de técnicas eficazes para o gerenciamento de dados, as quais possam recuperar rapidamente resultados relacionados com os dados de trajetórias. O gerenciamento de dados de trajetória pode lidar com o histórico de viagem de um objeto em movimento. Dentro do contexto das consultas submetidas aos dados de trajetória, é possível citar dois tipos principais: a *Range Queries* e as consultas *K-NearestNeighbor (KNN)*.

Além das consultas, é comum realizar análises acerca das anomalias que envolvem dados de trajetória. Essas anomalias podem estar relacionadas com eventos ou observações de não conformidade com um padrão esperado (por exemplo, um congestionamento no tráfego, causado por um acidente ou construções).

É comum encontrar na literatura propostas que visam detectar valores discrepantes de trajetórias, considerando a análise a partir de um conjunto de dados, como o trabalho de (LEE *et al.*, 2008), que propôs uma estrutura de partição e detecção para encontrar segmentos anômalos de trajetórias. Ou, ainda, (PAN *et al.*, 2013), que propôs um método para identificar anomalias de tráfego de acordo com o comportamento dos motoristas em uma rede rodoviária urbana. Ademais, o trabalho de (CHANDOLA *et al.*, 2009) realiza várias análises sobre métodos gerais de detecção de valores discrepantes e pode ser usado como objeto de estudo.

2.2 Fluxo Contínuo de Trajetórias

Nos últimos anos, os avanços na tecnologia permitiram gerar, de forma automática, dados relacionados às trajetórias dos objetos móveis, os quais podem mudar continuamente (isto é, a cada minuto ou a cada segundo) e reportar novas análises, que são observadas devido ao aumento do volume de dados, cujo crescimento ocorre em ritmo acelerado. A geração de fluxos contínuos de dados tem sido extensivamente pesquisada nos últimos anos. Isso ocorre tanto pelo aumento do número de aplicações, que tratam do comportamento de dados gerados frequentemente, quanto pela transformação de algoritmos mais tradicionais em propostas

ineficientes (ABADI *et al.*, 2003; AGGARWAL, 2007; ELMELEEGY *et al.*, 2009; GAMA, 2010; GUPTA *et al.*, 2013).

Para o entendimento deste trabalho de pesquisa, a definição de fluxo contínuo de trajetórias é fundamental. Tal definição, baseada em (LIBEN-NOWELL *et al.*, 2006; GUHA; HUANG, 2009; SILVA *et al.*, 2016b), é mostrada a seguir.

Definição 3 (Fluxo Contínuo de Trajetórias) Suponha que o tempo é discretizado em janelas de tempo (ou intervalos temporais) de tamanho δt. Seja $T_i = [i \cdot \delta t, i \cdot \delta t + \delta t)$ uma janela de tempo específica, tal que $i \geq 0$. O *Fluxo Contínuo de Trajetórias* em T_i é o conjunto de sub-trajetórias $S_i = \{ST_1^i, ST_2^i, ..., ST_n^i\}$, tal que cada sub-trajetória ST_j^i representa o movimento de um objeto o_j dentro da janela de tempo T_i.

Quando se trata de fluxos contínuos de trajetórias, surgem preocupações relacionadas tanto ao volume de dados, que pode chegar à escala de terabytes ou petabytes, quanto ao processamento de consultas submetidas a essa base volumosa. Dessa forma, é comum encontrar trabalhos que propõem a utilização do modelo *map-reduce*, cuja solução requer que os usuários expressem seu problema em termos de uma função que *mapeia* e processa menores registros de dados e outra função que *reduz* as soluções, mesclando todos os resultados mapeados para produzir um resultado final (HE *et al.*, 2010; YANG *et al.*, 2010).

Além do processamento, os fluxos contínuos de trajetórias também são extensivamente estudados devido à presença de valores discrepantes. (LIU *et al.*, 2011) propôs um algoritmo para investigar a presença de valores discrepantes no tráfego usando árvores, as quais são construídas com base nas propriedades temporais e espaciais dos valores detectados. (AGGARWAL *et al.*, 2003) busca realizar o monitoramento dos valores discrepantes, considerando a descoberta de *clusters* a partir do recebimento de fluxos contínuos de trajetórias. Outras propostas acerca do tratamento de valores discrepantes, considerando dados de trajetórias reportados continuamente, podem ser encontradas no trabalho de (GUPTA *et al.*, 2014).

2.3 *Map Matching*

Os dados de trajetória, geralmente, não são obtidos de maneira precisa. É comum encontrar ruídos ou erros de imprecisão, em que o ponto de localização de um objeto se encontra fora do segmento de rua ao qual pertence. Nesse caso, o ajuste das informações pode ser realizado por algoritmos de *MapMatching*, cuja solução visa mapear corretamente cada posição

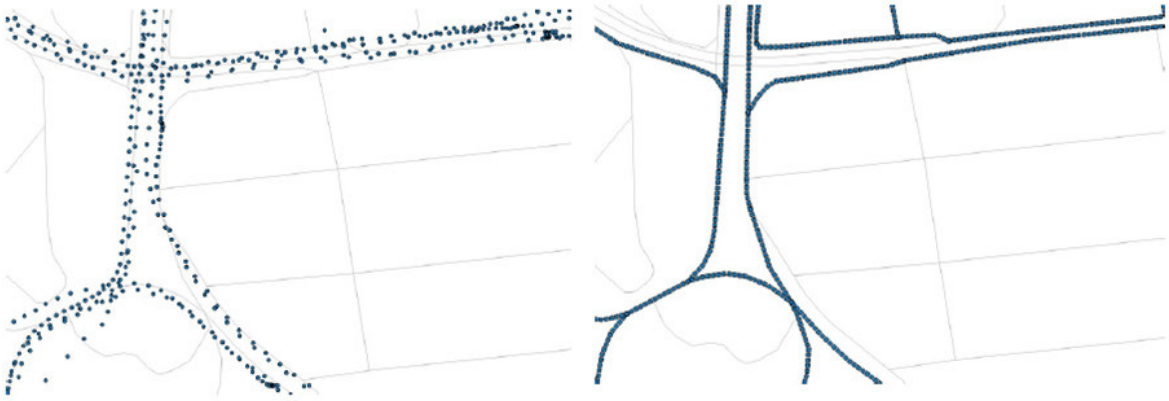
ocupada pelo objeto móvel, no correspondente segmento de rua (ZHENG, 2015; OCHIENG *et al.*, 2003). Dessa forma, o processo de mapeamento tem o objetivo de converter uma sequência de coordenadas geográficas, formadas por *(latitude, longitude)*, em uma sequência de segmentos (ZHENG, 2015). O propósito é realizar a correção e completude das informações, estimando não apenas o trajeto realizado, mas também os possíveis segmentos trafegados, ainda que existam informações faltantes.

As propostas dos algoritmos de *Map Matching* podem mudar de acordo com a variedade de pontos do objeto a ser mapeado. Nesse caso, os algoritmos podem ser classificados em duas categorias: métodos locais/incrementais e globais. Os algoritmos locais/incrementais (CIVILIS *et al.*, 2005; CHAWATHE, 2007) seguem uma estratégia gulosa para computar sequencialmente a solução, a partir de resultados que já foram obtidos anteriormente e que já tiveram o ajuste realizado. Esses algoritmos buscam encontrar um ponto ótimo, baseado na distância e similaridade, do ponto de análise ao correto segmento de rua. Os métodos locais/incrementais são executados de maneira eficiente e geralmente são adotados em aplicações que necessitam do processamento da solução em tempo real. No entanto, quando a taxa de amostragem da trajetória é baixa, é comum usar os algoritmos globais (ALT *et al.*, 2003; BRAKATSOULAS *et al.*, 2005), que pretendem combinar toda uma trajetória com uma rede rodoviária. Nesse caso, os algoritmos, de forma geral, buscam considerar os predecessores e os sucessores de um ponto. A estratégia global é mais precisa, mas menos eficiente do que a estratégia local/incremental, porque busca realizar o mapeamento com base em trajetórias inteiras, as quais já foram previamente computadas.

Algoritmos de *Map Matching*, de maneira geral, têm dois principais passos. O primeiro utiliza os dados dos objetos a serem interpolados no correto segmento de rua, cujas informações são formadas pela tupla *(latitude, longitude, timestamp)*, sendo *(latitude, longitude)* informações que determinam pontos de localização do objeto e *timestamp* a informação que determina a hora do dia. O segundo passo é responsável pelo processamento da correta associação de cada ponto da trajetória com o segmento de rua trafegado.

A Figura 3 mostra os dados antes e após o processamento do *Map Matching*, que foi realizado pelo Algoritmo *Barefoot*. Essa execução interpolou o correto posicionamento dos objetos nos seus respectivos segmentos de rua da cidade. A Figura 3(a) mostra como os objetos estavam posicionados assim que a coleta das informações via GPS ocorreu, e a Figura 3(b) mostra o resultado final do processo.

Figura 3 – *Map Matching* dos dados de trajetória.



(a) Sem Map Matching **(b) Com Map Matching**

Fonte: da autora.

As estratégias de *Map Matching*, existentes na literatura, enquadram-se em três categorias principais: *(i)* uso da média (ou mediana), que estima um valor verdadeiro (desconhecido) para um ponto, com base na média (ou mediana) deste ponto e seus predecessores, dado um instante de tempo (ZHENG, 2015); *(ii)* filtros de *Kalman*, que estima a trajetória a partir de parâmetros como a velocidade (LEE; KRUMM, 2011); *(iii)* detecção dos valores discrepantes baseados numa heurística, que remove pontos de ruído (YUAN *et al.*, 2013).

Uma das principais preocupações quanto aos resultados obtidos, a partir do processamento do *Map Matching*, está relacionada com a precisão desses resultados. Embora os erros de localização venham sendo resolvidos, diferentes trabalhos continuam sendo propostos dentro desse contexto. Dentre eles, é possível citar (LI *et al.*, 2017), que busca propor novos algoritmos que garantam maior acurácia e integridade dos resultados. E, ainda, (LIU; LI, 2017; LUO *et al.*, 2017), que também propuseram estratégias de interpolação, que computam resultados de mapeamento dos objetos a partir de soluções probabilísticas.

2.4 Modelos de Regressão

A teoria de Regressão teve origem no século XIX com Galton (GALTON, 1886). Em um de seus trabalhos, ele estudou a relação entre a altura de pais e filhos, procurando saber como a altura do pai poderia influenciar a altura do filho. Galton percebeu que, se o pai fosse muito alto ou muito baixo, o filho teria uma altura tendente à média. Se isso não ocorresse, as pessoas seriam cada vez mais altas, dado que os filhos de pais altos seriam mais altos ainda. Esse estudo foi chamado de análise de Regressão, por justificar que existe uma tendência de os

dados regredirem à média. Existem vários trabalhos na literatura que usam análises de regressão (KUBRUSLY; LOPES, 2015; LOH, 2011; NADUNGODAGE *et al.*, 2011; PATEL *et al.*, 2016; SRIMANI; PATIL, 2014; TORGO, 2002; CHEN *et al.*, 2002). Inclusive, este trabalho de pesquisa tem como contribuição modelos que utilizaram dois diferentes tipos de análises de regressão – a regressão linear simples e a regressão linear múltipla –, as quais têm como base a teoria de Regressão (NASCIMENTO *et al.*, 2016b; NASCIMENTO *et al.*, 2016a).

A regressão linear, seja ela simples ou múltipla, permite realizar análises para identificar o relacionamento entre variáveis. No caso da regressão linear simples, para se estabelecer uma análise a partir de n pares de valores de duas variáveis, X_i e Y_i (com i = 1, 2, ..., n), admitindo-se que Y é função linear de X, é possível analisar o estudo da relação entre as duas variáveis, onde é procurada uma função de X que explique Y, cujo modelo estatístico é:

$$Y_i = \alpha + \beta * X_i + u_i,$$

São considerados na função os valores de X_i, que corresponde à variável independente ou explanatória; Y_i uma variável dependente ou resposta; β o coeficiente angular da reta, que é também denominado de coeficiente de regressão; α o coeficiente linear da reta, que é também conhecido como o termo constante da equação de regressão; e u_i o valor do erro médio. β e α são parâmetros que recebem uma atenção especial, dado que é preciso identificá-los de tal forma que os desvios entre seus valores observados e estimados sejam mínimos. Para isso, é possível utilizar o método dos *mínimos quadrados*, que corresponde a uma técnica de otimização matemática cujo objetivo é encontrar o melhor ajuste para um conjunto de dados (GUJARATI; PORTER, 2011).

Apesar de buscar uma relação entre as variáveis independentes (X_i) e dependentes (Y_i), em geral, essa associação não é perfeita. Isso ocorre porque os pontos não se situam perfeitamente sobre a função que relaciona as duas variáveis. Segundo (GUJARATI; PORTER, 2011), para estabelecer o modelo de regressão linear simples, é preciso pressupor características como:

- A relação entre X e Y é linear.
- Os valores de X são fixos, isto é, X não é uma variável aleatória.
- A média do erro é nula, isto é, E(u_i) = 0.

De modo geral, os modelos de regressão podem ser usados com diferentes objetivos. (DEMÉTRIO; ZOCCHI, 2006) traz alguns exemplos do seu uso, como em:

1. Predições: corresponde ao motivo mais comum de se encontrar a utilização dos modelos de

regressão. Ao utilizar o modelo de predição, espera-se que a variação de Y seja explicada a partir da variável explanatória (X). Dessa forma, é possível utilizar o modelo de predição para obter valores de Y correspondentes a valores de X que não estavam entre os dados. Em geral, são usados os valores de X que estão dentro do intervalo de variação estudado. Mas é possível encontrar análises que utilizem valores fora desse intervalo, as quais recebem o nome de extrapolação.

2. Seleção de variáveis: frequentemente, não se tem ideia de quais são as variáveis que afetam significativamente a variação da variável dependente (Y). Nesse caso, é comum conduzir estudos que respondam esse tipo de questão.
3. Estimativa de parâmetros: corresponde ao processo de estimar parâmetros ou possibilitar o ajuste de modelos, dada a utilização de um conjunto de dados (amostra) referente às variáveis resposta (Y).
4. Inferência: corresponde à utilização de técnicas como testes de hipóteses ou intervalos de confiança para realizar o ajuste de um modelo de regressão. Nesse caso, além de estimar os parâmetros, buscam-se estratégias para realizar inferências sobre os dados.

Antes de realizar qualquer uma das análises definidas anteriormente, é importante que se plotem os pares de dados em *diagramas de dispersão*. Esses diagramas possibilitam visualizar tanto o tipo de relação que existe entre as variáveis, quanto a presença de valores discrepantes no conjunto de dados, os quais podem resultar em imprecisões.

Além do modelo de regressão linear simples, existe o modelo de regressão linear múltipla, que pode ser identificado quando se admite que a obtenção de uma variável resposta (Y) é função de duas ou mais variáveis explicativas ou regressoras (X).

2.5 Árvores Binárias Rotuladas

As árvores binárias rotuladas estendem a definição de árvores binárias, as quais são conhecidas por sua simplicidade e eficiência quando lidam com grandes volumes de dados. De forma geral, as árvores binárias permitem inferir sobre regras de predição, quando utilizadas em modelos preditivos. Nesta tese, as árvores binárias rotuladas apresentam mudanças no seu comportamento, dados diferentes intervalos de tempo. A definição de árvores binárias rotuladas é dada a seguir.

Definição 4 (Árvore Binária Rotulada) Dado $S_i = \{ST_1^i, ST_2^i, ..., ST_n^i\}$ um fluxo contínuo de

trajetórias, que são recebidas no intervalo de tempo $T_i = [t_i, t_f]$. Uma *Árvore Binária Rotulada T* é uma árvore binária, cujo os nós n_i são rotulados da seguinte forma: $n_i = (\mu_i, \sigma_i, [t_{s_i}, t_{e_i}])$, tal que μ_j é a média e σ_j é o desvio padrão dos tempos de viagens das trajetórias, que pertencem a S_i, $t_{s_i} < t_{e_i}$ e $[t_{s_i}, t_{e_i}] \in T_i$. Cada nó n_i apresenta uma sub-árvore binária rotulada a esquerda T_e e uma sub-árvore binária rotulada a direita T_d. Seja o nó raiz de T_e, $n_{T_e} = (\mu_e, \sigma_e, [t_{s_e}, t_{e_e}])$ e o nó raiz de T_d, $n_{T_d} = (\mu_d, \sigma_d, [t_{s_d}, t_{e_d}])$, $t_{s_i} \leq t_{s_e}, t_{e_e} < t_{s_d}, t_{e_d} \leq t_{e_i}$.

As árvores binárias rotuladas podem ser aplicadas tanto no contexto histórico, no intuito de realizar análises e obter conhecimento, como em tempo real, dado o recebimento de novos fluxos de dados. Neste caso, é analisado cada nível da árvore, que pode determinar um conjunto finito de resultados, e verifica-se a necessidade do crescimento em profundidade desta árvore, dada a computação de uma manutenção incremental, quando recebidos novos dados de trajetória.

2.6 Conclusão do Capítulo

Este capítulo apresentou os conceitos básicos, relacionados a trajetórias e fluxos contínuos de trajetórias, a serem considerados no restante do trabalho. Mostrou-se também o processo de mapeamento, que é realizado a partir do algoritmo *Map Matching*, quando possibilita computar o ajuste do ponto de localização de um objeto, com o correto segmento de rua. Ademais, foram feitas explanações sobre as árvores binárias rotuladas, estendem a definição de árvores binárias. Essas árvores são conhecidas por sua simplicidade e eficiência quando lidam com grandes volumes de dados. Neste trabalho de pesquisa, as árvores binárias rotuladas apresentam no comportamento dos objetos, dados diferentes intervalos de tempo. No Capítulo seguinte, será apresentado o estado da arte, discorrendo-se sobre trabalhos relacionados a esta pesquisa.

3 ESTADO DA ARTE

Este capítulo apresenta uma revisão da literatura relacionada com o estudo realizado nesta tese, cujo foco está centrado em quatro tópicos principais: *(i)* Análise de Dados de Trajetórias, *(ii)* Funções Preditivas Contínuas; *(iii)* Funções Preditivas Descontínuas; e *(iv)* Modelos de Árvores a partir de Dados de Trajetórias. Os tópicos *(ii), (iii)* e *(iv)* apresentam exemplos de técnicas potencialmente utilizadas para gerar modelos de predição que computem tempos de viagens e possibilitem a realização de análises do tráfego de uma cidade, dada uma hora específica do dia. As respostas sobre análises como essas são temas ativos em áreas de pesquisa e, por isso, muitos trabalhos foram desenvolvidos.

A Seção 3.1 discute diferentes trabalhos que realizam análises de dados, considerando o processamento tanto de informações históricas, quanto das reportadas em tempo real. O objetivo de cada trabalho mostrado é obter resultados acerca da utilização e da aplicabilidade dos dados de trajetória. Já as funções preditivas contínuas e descontínuas, discutidas nas Seções 3.2 e 3.3, mostram diferentes trabalhos existentes na literatura, que buscam explicar o deslocamento de objetos móveis, dada a análise dos tempos de viagens, considerando uma hora específica do dia. A Seção 3.4 discorre sobre Modelos de Árvores construídos a partir de dados de trajetórias, com o objetivo de analisar os impactos causados no modelo, dadas as mudanças no comportamento dos dados. Finalmente, a Seção 3.5 encerra este capítulo.

3.1 Análise de Dados de Trajetória

O processamento tradicional de dados de trajetória, realizado por dispositivos de rastreamento e persistido em disco, permite que informações históricas sejam manipuladas para realizar análises sobre uma grande variedade de dados – que podem conter informações sobre carros de passeio, táxis, ônibus ou pessoas – (KINOSHITA *et al.*, 2016). Esse tipo de processamento é vantajoso no sentido de garantir maior precisão às informações obtidas, porque permite construir, por exemplo, um conjunto de regras a partir do processamento controlado sobre o volume de dados (SCHNITZLER *et al.*, 2014) e gerar conhecimento acerca dos dados de mobilidade (ZHENG, 2015) em um menor tempo possível (GABER *et al.*, 2009).

O crescimento de tecnologias, a exemplo das redes de sensores, tem contribuído para o aumento dos fluxos de dados. Diante disso, cabe a pergunta: *"Nós podemos realizar uma análise, em tempo real, para alertar as pessoas sobre as mudanças no tráfego de cidades?"*. A

característica dos dados recebidos em tempo real está relacionada com seu ilimitado volume de dados, rápidas mudanças no comportamento e presença de informações que geram análises valiosas (GABER *et al.*, 2009). O processamento desses dados pode ser classificado em duas categorias: *(i)* o gerenciamento de *data streams*, baseado na sumarização dos dados (BABCOCK *et al.*, 2002; ABADI *et al.*, 2003); e *(ii)* a mineração de dados, que computa técnicas de *Data Mining*, com tempo linear (MUTHUKRISHNAN *et al.*, 2005).

(GABER *et al.*, 2009) construiu uma tabela que mostra a diferença entre o processamento tradicional e o processamento de fluxos contínuos de dados. Algumas das diferenças citadas por (GABER *et al.*, 2009) são mostradas na Tabela 2, e correspondem a conceitos usados para entendimento dos modelos relatados nesta pesquisa; tais modelos abrangem soluções preditivas, que computam seu resultado a partir de dados históricos, gerados por um processamento tradicional; e soluções incrementais, as quais computam a manutenção do modelo, dado recebimento de fluxos contínuos de informações.

Tabela 2 – Fluxo Contínuo de Dados x Processamento Tradicional

Processamento do Fluxo Contínuo de Dados	**Processamento Tradicional**
Processamento de Dados Reportados Continuamente.	Processamento de Dados Históricos.
Geração rápida de dados.	Geração baixa ou regular de dados.
Armazenamento de dados não viável.	Armazenamento de dados é viável.
Resultados aproximados são aceitáveis.	É exigida a precisão dos resultados.
Contextos espaciais e temporais são particularmente importantes.	Os contextos espaciais e temporais são considerados em aplicações geográficas.

Fonte: (GABER *et al.*, 2009)

Os dados reportados continuamente, de modo geral, têm atraído a atenção nos últimos anos. Vários são os trabalhos que buscam desenvolver estratégias para computar esse fluxo de informações (GABER *et al.*, 2009; ARASU *et al.*, 2003; AGGARWAL, 2007; GAMA; GABER, 2007; GAMA, 2010). Isso ocorre porque as informações sobre fluxos contínuos de dados têm trazido novos desafios à comunidade acadêmica, tais como: gerenciar o fluxo contínuo das informações; gerenciar requisitos de memória, os quais requerem rápido processamento das informações; detectar mudanças relacionadas aos dados, para computar resultados de análises; propor novas estratégias para algoritmos de mineração de dados, que descubram conhecimentos

apesar da variabilidade dos dados e do volume na escala de terabytes (HE *et al.*, 2010; ARTHUR; VASSILVITSKII, 2007; YANG *et al.*, 2010; FALOUTSOS *et al.*, 1994).

As aplicações de análises de fluxos contínuos de dados podem variar em diversos cenários, como *Data Mining*, já citado anteriormente; em análises meteorológicas; no monitoramento de tráfego em redes de rodovias; na Estatística, usando testes de hipóteses, os quais buscam encontrar soluções computacionalmente eficientes para análise de dados (AGGARWAL *et al.*, 2003; AGGARWAL *et al.*, 2004; GABER *et al.*, 2005); ou, ainda, em ambientes de aprendizado de máquinas, realizando análises estatísticas em sistemas escaláveis. A ideia é realizar um processo probabilístico para saber quais dados serão ou não processados (DOMINGOS; HULTEN, 2001).

Os dados reportados continuamente correspondem a uma sequência ilimitada de objetos, os quais podem ser gerados em uma taxa rápida (SILVA *et al.*, 2013). Segundo (QIN *et al.*, 2016), existem algumas características típicas de fluxos de dados, como: a chegada contínua de novos dados; a forma desordenada com que geralmente isso ocorre; a possibilidade de os dados reportados terem tamanho ilimitado; e, normalmente, a falta de persistência desses dados após serem processados. Assim, em alguns cenários que computam as análises dos dados reportados em tempo real, as informações não são persistidas dentro do contexto histórico. Isso ocorre porque, além de liberar espaço em disco, esses cenários buscam refletir as análises mais emergentes, que consigam expressar de forma rápida, sem acesso a disco, as mudanças relacionadas à distribuição dos dados.

3.2 Funções Preditivas Contínuas de Tempo de Viagem

As funções preditivas podem ser usadas para estimar os tempos de viagem dos objetos, os quais podem ser considerados um parâmetro essencial para avaliar o comportamento dos objetos que trafegam em redes de rodovias (YE *et al.*, 2015) e o desempenho das condições de tráfego (CHEN *et al.*, 2010). É a partir dessa informação que estratégias são desenvolvidas, tais como o gerenciamento de tráfego, o planejamento de rotas e, ainda, a observação dos objetos em tempo real (SUN *et al.*, 2008; DAILEY, 1993; NAM; DREW, 1996; PETTY *et al.*, 1998). Os tempos de viagem podem ser obtidos direta ou indiretamente. É possível obter tempos de viagem a partir de veículos computadorizados, dispositivos de GPS ou, ainda, a partir de celulares. Dentre os métodos indiretos, é possível citar câmeras de vídeo e/ou sensores, distribuídos em rodovias e instalados em Sistemas de Transportes Inteligentes (DAILEY, 1993; SUN *et al.*, 2008;

SRINIVASAN; JOVANIS, 1996; OH *et al.*, 2002). O tempo de viagem possibilita a realização de estimativas e predições confiáveis sobre as redes de rodovias, provendo uma informação valiosa para que os motoristas reorganizem suas rotas e se desviem dos pontos de congestionamento (ZHANG *et al.*, 2013; KONG *et al.*, 2016).

Embora o conceito de tempo de viagem seja um parâmetro importante para realizar a predição das durações de viagens, existem alguns desafios para computar esse parâmetro e permitir sua utilização na construção de funções preditivas. É possível citar como exemplo a dispersão dos dados, uma vez que muitos segmentos têm informações faltantes sobre a localização do objeto (JENSEN; LARSEN, 2014; MA *et al.*, 2013; YUAN *et al.*, 2010). (WANG *et al.*, 2014) propõe um novo modelo, que visa estimar o tempo de viagem dos objetos nos segmentos a serem percorridos e predizer esse valor para toda a trajetória, e não apenas para um segmento específico. Nesse caso, (WANG *et al.*, 2014) construiu um modelo de concatenação dos segmentos trafegados e usa uma estratégia para descobrir padrões, dada a similaridade das travessias percorridas pelos objetos.

(CHEN *et al.*, 2002), em sua proposta, buscou minimizar tanto o volume de dados, quanto o uso de memória e disco, visando realizar estratégias de compressão de regressões, dado o recebimento de fluxos contínuos de dados. O objetivo de (CHEN *et al.*, 2002) é realizar análises de regressões, dividindo os dados em níveis de granularidade mais finos — que são os dados mais recentes — e em níveis de granularidades mais grossos -– os dados mais antigos. Ainda é possível citar o trabalho de (SRIMANI; PATIL, 2014), cujo propósito foi criar modelos de regressão que minimizassem a utilização da memória, para garantir maior eficiência na computação de dados reportados em tempo real. Esses e outros modelos justificam o surgimento de várias soluções atualizadas, devido ao recebimento de fluxos contínuos de dados, os quais buscam realizar predições sobre os mesmos.

Os tempos de viagem também podem ser considerados parâmetros essenciais para a construção de modelos preditivos. Trabalhos como (SUN *et al.*, 2008; CORTES *et al.*, 2001; KISGYÖRGY; RILETT, 2002; OH *et al.*, 2002; ISHAK; AL-DEEK, 2002) usaram o modelo *Piecewise* para simular a variação dos tempos de viagem e das velocidades médias dos objetos que tenham essa informação desconhecida. Já (SUN *et al.*, 2008) buscou estimar as probabilidades máximas e mínimas da velocidade do objeto, dado um intervalo de tempo e levando em consideração os horários de fluxos livres e os horários já em condições de congestionamento. Em muitos trabalhos, a solução *Piecewise* é amplamente estudada para explicar o comportamento de

dados de trajetória. (SUN *et al.*, 2008) justifica a utilização da função *Piecewise* para dados de trânsito, devido à distribuição dos dados, os quais oscilam em momentos de aceleração e desaceleração durante o trajeto do segmento. Já (CORTES *et al.*, 2001; KISGYÖRGY; RILETT, 2002; OH *et al.*, 2002; ISHAK; AL-DEEK, 2002) buscaram utilizar a *Piecewise* como um método de interpolação para distribuição dos objetos, inferindo-lhes a velocidade – dados detectores de velocidade adjacentes.

Apesar das funções *Piecewise* serem amplamente usadas na literatura, existem autores que explicam o comportamento dos dados de trajetória a partir da utilização de funções Gaussianas, porque conseguem incorporar em sua topologia os diferentes fluxos do tráfego nas vias analisadas. É possível referenciar exemplos desses trabalhos citando (LIAO *et al.*, 2005), que propôs um modelo de Gaussianas para explicar o deslocamento dos objetos, considerando três intervalos de velocidades (caminhada, baixa velocidade e alta velocidade); e ainda (LIEBIG *et al.*, 2017), que propôs uma solução para estimar os valores de sensores a partir de Modelos Gaussianos.

O método de aproximação da *Piecewise*, chamado PLA (*Piecewise* Linear Approximation) (CAMERON, 1966), é largamente usado em séries de dados, devido a sua simplicidade, e pode ser utilizado para manipular tanto dados históricos, quanto dados recebidos em tempo real (FU *et al.*, 2001). A PLA, utilizada a partir dos dados históricos, busca coletar volume de dados antes de o processamento ocorrer e gera um modelo contínuo (ELMELEEGY *et al.*, 2009; HAKIMI; SCHMEICHEL, 1991). Já a PLA, quando modificada a partir do recebimento de dados em tempo real, usa uma estratégia de manutenção incremental (CHEN; WANG, 2013), que pode apresentar uma abordagem problemática, dado que tende a gerar um modelo disjunto, também considerado desconexo por alguns autores. Essa estratégia será discutida na próxima Seção.

3.3 Funções Preditivas Descontínuas de Tempo de Viagem

Apesar de o modelo de regressão contínua resultar numa estratégia que retorna resultados de predição dentro de um melhor tempo de processamento, é possível encontrar nos trabalhos que o utilizam um impedimento em comum: existe perda de informação quando se busca estimar uma aproximação das funções de regressão descontínuas (LUO *et al.*, 2015). A aproximação dos segmentos que o compõem visa construir uma solução contínua. O impedimento dessa aproximação também foi comprovado em um dos modelos propostos neste trabalho

de pesquisa, quando foi construída uma solução cujo problema é *NP-difícil* (NASCIMENTO *et al.*, 2016a). Esse trabalho buscou estimar uma função de regressão contínua, que estende a função *Piecewise*, mas identificou que era impossível determinar, em tempo de execução, um ponto de quebra em comum para regressões lineares adjacentes, sem perda de informação.

(LUO *et al.*, 2015), em seu trabalho, afirma que a solução *Piecewise* desconexa causa computação subsequente do modelo, o que resulta na degradação do desempenho do algoritmo que a usa. Dentro desse contexto, o autor propõe uma solução chamada *mixed-type PLA*, que mistura dois diferentes algoritmos – um que é utilizado para solução de *Piecewise* contínua e outro para estratégia descontínua. O resultado dessa solução gera uma regressão múltipla contínua, dado cada par de regressões disjuntas adjacentes. Além dessa proposta, existem outras nas quais os autores propõem soluções para realizar análises, a partir do recebimento de dados em tempo real, a exemplo de: *Discrete Fourier Transform* (RAFIEI; MENDELZON, 1997), *PiecewiseAggregateApproximation* (YI; FALOUTSOS, 2000), *DiscreteWaveletTransform* (POPIVANOV; MILLER, 2002), *AdaptivePiecewise Constant Approximation* (LAZARIDIS; MEHROTRA, 2003; CHAKRABARTI *et al.*, 2002) e *ChebyshevPolynomials* (CAI; NG, 2004; HEATH, 1997; EPPERSON, 2013; QI *et al.*, 2015), que usaram funções polinomiais.

Outro trabalho que está voltado para a atualização do modelo de predição, dado o recebimento contínuo de novos fluxos de dados, é (NADUNGODAGE *et al.*, 2011). Nesse caso, o autor propõe uma solução incremental que, dinamicamente, recomputa as regressões lineares, gerando uma solução composta por regressões lineares múltiplas, as quais são atualizadas com base nas funções computadas anteriormente. (NADUNGODAGE *et al.*, 2011) propõe a utilização de operações de descarte das informações, não necessitando da persistência dos dados para construção de sua proposta. No entanto, essa estratégia pode constituir em outro problema quando se busca um padrão no modelo, porque a não persistência dos dados pode gerar imprecisão nos resultados (BABCOCK *et al.*, 2003; TATBUL *et al.*, 2003). Ainda assim, o autor defende que, em um ambiente que computa dados recebidos continuamente, apenas aquelas informações mais recentes são significativas para análises. É possível, ainda, citar a proposta de (PATEL *et al.*, 2016), que também utiliza um modelo de regressões lineares múltiplas, as quais são computadas de forma incremental, dado o recebimento de novos dados. Esse modelo, basicamente, estende a regressão linear simples, usando mais de uma variável independente. No entanto, o impedimento da descontinuidade da solução também é encontrado em sua proposta.

3.4 Modelos de Árvores a partir de Dados de Trajetórias

Os modelos de predição baseados em árvores são conhecidos por sua simplicidade e eficiência quando lidam com grandes volumes de dados. As árvores binárias de predição correspondem a um tipo particular de modelo preditivo. A partir desse modelo é possível inferir regras de predição, com o objetivo de obter uma informação útil. Basicamente, realiza-se uma divisão recursiva com dados de treinamento, gerando subconjuntos de dados com volumes menores (LOH, 2011). O uso deste algoritmo é a causa da eficiência desses métodos. Apesar do bom processamento, essa estratégia pode gerar resultados com baixa precisão, devido à obtenção de estimativas a partir de pequenas amostras de dados (LAKSHMI *et al.*, 2013).

(QUINLAN, 2014) combinou os resultados de predição, obtidos a partir de árvores, com amostras de dados obtidas a partir de algoritmos que computam vizinhos mais próximos. Outros autores utilizam o modelo baseado em árvores de predição para obter funções que podem ser constantes, dada a média da amostra prevista para cada nó, ou baseado em regressões lineares simples (KIM; LOH, 2001; KIM; LOH, 2003), todos eles usando dados históricos. As funções baseadas em regressões lineares também foram amplamente estudadas, a exemplo da proposta de (WITTEN *et al.*, 2016), que apresentou um modelo que permite criar a função *Piecewise* a partir de um conjunto de dados históricos, os quais foram usados para realizar análises de predição.

Além de computar informações históricas para obter conhecimento, os modelos baseados em árvores de regressão podem ser utilizados em tempo real. Nesse caso, realiza-se uma manutenção incremental do modelo, dado o recebimento de novos fluxos de dados. O objetivo é particionar os novos dados recebidos em pequenos conjuntos e alcançar predições com melhores acurácias (ZHANG *et al.*, 2015). Alguns algoritmos de árvores operam sobre todo o conjunto de treinamento. Porém, em muitas situações, a abordagem incremental se mostra mais vantajosa. A proposta de (POTTS; SAMMUT, 2005) discorre sobre um modelo incremental, que computa regressões lineares em cada nó não raiz de uma árvore binária. O autor define um conjunto de regras para possibilitar a manutenção incremental, dadas taxas de mudanças previamente definidas. Dentro desse contexto, as regras geradas são usadas como critérios para determinar o crescimento da árvore. Já (ZHANG *et al.*, 2015) propôs uma estratégia que computa múltiplas árvores de decisão, de forma incremental, dado o recebimento de novos fluxos de dados, e o resultado dessa computação permite realizar múltiplas classificações sobre os dados de análise.

O método incremental se apresenta como uma solução viável quando o processa-

mento deve ser realizado, dado o recebimento de fluxos de dados em tempo real. No entanto, para um modelo incremental, é importante determinar não apenas onde, mas também quando a divisão ocorrerá. O trabalho de (UTGOFF *et al.*, 1997) propõe uma estratégia com base em árvores de classificação. Dentre as técnicas mais estudadas no contexto de classificação, é possível citar, popularmente, as chamadas árvores de decisão (GUPTA *et al.*, 2013). Em seu modelo, (UTGOFF *et al.*, 1997) considera que cada nível da árvore pode determinar um conjunto finito de resultados e permite que a árvore cresça em profundidade, dada a computação de uma manutenção incremental. Nas propostas de (QUINLAN, 1993; FRANK *et al.*, 1998) foram adotadas estratégias de divisões, que minimizaram a medida do desvio padrão, dados os valores computados para cada nó não raiz da árvore. Já (TORGO, 2002) buscou propor soluções para considerar a parada de crescimento da árvore, com o objetivo de minimizar o *erro quadrático médio* dos valores computados.

Os modelos de árvores de regressão podem evoluir de outras formas. Por exemplo, (LI *et al.*, 2015) propôs uma estratégia incremental que computa uma árvore de regressão com base nas árvores de regressão precedentes. Dentro desse contexto, um conjunto de árvores de regressão simples é combinado para gerar um novo modelo. Já no trabalho de (ZHANG *et al.*, 2015) foi proposto um algoritmo de classificação supervisionado, para gerar árvores binárias e realizar sua manutenção incremental a partir do recebimento de novos fluxos de dados. O modelo gerado nesse trabalho busca detectar os movimentos dos objetos, dados critérios específicos de classificação. Cada nó da árvore é composto por classificadores que geram regras de indução, as quais são capazes de detectar um tipo de movimento do objeto de análise. Outro trabalho que propõe um modelo incremental é de (GAMA *et al.*, 2005), cuja contribuição utiliza a desigualdade de *Hoeffding* para fornecer o limite superior entre o valor esperado e o valor predito de uma solução. O resultado dessa desigualdade é usado para decidir se a árvore incremental pode ou não ser expandida, criando novos nós.

3.5 Conclusão do Capítulo

Este capítulo apresentou os principais trabalhos correlatos desta tese. Os trabalhos citados foram agrupados em três linhas de análise:

1. A primeira agrupa trabalhos que contemplam informações acerca de dados de trajetória, tanto inseridos no contexto histórico, quanto fluxos de dados reportados continuamente;
2. A segunda contempla informações acerca de funções preditivas, as quais permitem realizar

análises acerca dos resultados obtidos e auxiliam no processo de tomada de decisão com maior acurácia;

3. A terceira, enfim, aborda as estratégias para a criação e a manutenção incremental de árvores com o objetivo de possibilitar o processo de crescimento controlado dessas estruturas de dados.

A Tabela 3 apresenta os principais trabalhos relacionados (descritos nas Seções 3.1, 3.2 e 3.4), comparando-os, também, com esta pesquisa. São analisados alguns itens importantes, que puderam ser selecionados durante a revisão da literatura. Na primeira coluna da Tabela, estão os trabalhos sob análise. A partir da segunda coluna, estão as características consideradas para comparação destes trabalhos. As descrições de cada coluna estão dispostas logo abaixo da Tabela.

- **Dados Históricos:** A predição realizada utiliza dados históricos? Valores possíveis: Sim (S), Não (N) ou Não Informado (NA);
- **Reportados Fluxos de Dados:** A predição realizada utiliza fluxos de dados reportados continuamente? Valores possíveis: Sim (S), Não (N) ou Não Informado (NA);
- **Manutenção Incremental:** É realizada a manutenção incremental no modelo, dado o recebimento de novas informações? Valores possíveis: Sim (S), Não (N) ou Não Informado (NA);
- **Trata Valores Discrepantes:** Realiza-se tratamento de valores discrepantes? Valores possíveis: Sim (S), Não (N) ou Não Informado (NA);
- **Persistência de Dados:** Persistem os dados? Valores possíveis: Sim (S), Não (N) ou Não Informado (NA);
- **Solução Contínua:** Gera-se um modelo contínuo? Valores possíveis: Sim (S), Não (N) ou Não Informado (NA).

Com base nas análises realizadas e mostradas na Tabela 3, é possível afirmar que a proposta da solução PIPE, publicada em (NASCIMENTO *et al.*, 2017) e resultante da contribuição deste trabalho de pesquisa, cobre praticamente todas as características identificadas como importantes para análise dos dados de trajetória, que são reportados como fluxo de dados. A solução PIPE pode ser construída a partir da utilização de dados históricos e ser facilmente manipulada a partir do recebimento de novos dados de trajetória. Assim, é considerada uma solução incremental. A atualização do modelo PIPE ocorre sem perda de informação, e sua estratégia não necessita da persistência de dados para obter resultados de análises.

Apesar de não persistirem os dados, as informações relacionadas ao histórico não são perdidas, porque todas elas estão contidas em nós pertencentes à árvore de predição binária, como será mostrado no Capítulo 4. A função de predição, obtida a partir do modelo PIPE, é contínua e gera uma solução diferenciável. Mas, sua estratégia necessita do tratamento de valores discrepantes, porque a presença desse tipo de dado pode resultar em informações imprecisas ou provocar um desbalanceamento na árvore.

No próximo Capítulo, será detalhado o modelo de predição PIPE, e no Capítulo 5 serão apresentadas as avaliações experimentais que validaram essa proposta.

Tabela 3 – Sumarização dos Trabalhos Relacionados

Trabalho	**Dados Históricos**	**Reportados como Fluxos de Dados**	**Manutenção Incremental**	**Trata Valores Discrepantes**	**Persistência de Dados**	**Solução Contínua**
(KIM; LOH, 2001)	S	N	N	NA	NA	S
(ISHAK; ALDEEK, 2002)	N	S	S	NA	NA	N
(KISGYÖRGY; RILETT, 2002)	N	S	S	NA	NA	N
(KIM; LOH, 2003)	S	N	N	NA	NA	S
(SUN *et al.*, 2008)	N	S	S	NA	NA	S
(ELMELEEGY *et al.*, 2009)	N	S	N	NA	NA	S
(NADUNGODAGE *et al.*, 2011)	N	S	S	N	S	S
(CHEN; WANG, 2013)	N	S	S	N	S	N
(WANG *et al.*, 2014)	N	S	S	NA	NA	S
(QUINLAN, 2014)	S	N	N	NA	NA	S
(LUO *et al.*, 2015)	N	S	S	N	S	S
(NASCIMENTO *et al.*, 2016b) – proveniente desta tese.	**S**	**N**	**N**	**S**	**S**	**S**
(NASCIMENTO *et al.*, 2016a) – proveniente desta tese.	**N**	**S**	**S**	**S**	**S**	**N**
(NASCIMENTO *et al.*, 2017) – solução PIPE - proveniente desta tese.	**S**	**S**	**S**	**N**	**N**	**S**

Fonte: da autora.

4 MODELO DE PREDIÇÃO INCREMENTAL

Este capítulo discorre sobre o *Modelo de Predição Incremental*, proposto nesta tese, chamado *PIPE: Um Preditor de Tempos de Viagem usando Fluxo Contínuo de Trajetórias*, que pode ser usado para estimar os tempos de viagens dos objetos móveis. O PIPE assume que o comportamento dos dados de trajetórias, pertencentes aos objetos móveis analisados neste trabalho, se assemelham com as distribuições Gaussianas, as quais podem representar a aceleração e desaceleração dos objetos móveis dada sua aplicação sobre a distribuição dos dados, como também é considerado no trabalho proposto por (LIAO *et al.*, 2005), quando foram realizadas análises sobre esse tipo de dado.

A estrutura deste capítulo está organizada da seguinte forma. Na Seção 4.1, apresenta-se, em linhas gerais, o objetivo de construir a solução PIPE e a definição do problema a ser resolvido neste trabalho. A Seção 4.2 discorre sobre o modelo PIPE, o qual permite ser atualizado incrementalmente a partir do recebimento de novos dados de trajetórias; e na Seção 4.3, apresenta-se as considerações finais deste capítulo.

4.1 Definição do Problema

Esta seção fornece a definição do problema a ser solucionado neste trabalho e os seus objetivos. Os conceitos definidos aqui são compreendidos a partir do entendimento dos conceitos básicos mostrados no Capítulo 2.

Uma tarefa essencial na construção de *Sistemas de Transportes Inteligentes* é monitorar e analisar o movimento dos veículos dentro ou fora das cidades. Busca-se entender e detectar as possíveis anomalias, congestionamentos, planejamentos de viagens ou, ainda, predizer o tempo de chegada de um objeto móvel no decorrer do tempo, considerando uma *origem* e um *destino*. Em todos esses casos, é necessário computar modelos que consigam predizer o tempo de viagem dos objetos, cujo valor pode continuamente mudar. As abordagens existentes, como, por exemplo, os modelos de regressão ou as distribuições arbitrárias – como soluções *probabilísticas* e *modelos de Markov* –, tendem a gerar soluções descontínuas, quando atualizadas, dado o recebimento de fluxos contínuos de trajetórias, conforme foi mostrado no Capítulo 3.

O impedimento de utilizar uma solução descontínua está relacionado com o tempo de processamento do algoritmo que a computa (LUO *et al.*, 2015). Nesse caso, muitas propostas buscam utilizar técnicas de aproximação para segmentos descontínuos, porque, ainda que

sua atualização resulte em perdas de informações, a solução encontrada consegue garantir a obtenção de uma função preditiva contínua e o processamento desta solução ganha em termos de desempenho (Capítulo 3). Dentro desse contexto, a hipótese deste trabalho é que os modelos de predição que computem os valores preditos dos tempos de viagens, considerando objetos móveis, sejam criados e apresentem como resultado funções de predição contínuas, que tenham boa acurácia, ainda que sejam atualizadas a partir do recebimento de novos dados de trajetórias; tais dados são volumosos, esparsos e podem conter erros, imprecisões e valores discrepantes.

O objetivo desta tese é propor um novo modelo de predição incremental, chamado *PIPE: Um Preditor de Tempos de Viagem usando Fluxo Contínuo de Trajetórias*, o qual possibilita encontrar uma solução contínua e mantida de forma incremental, que seja capaz de predizer os tempos de viagem dos objetos móveis, dada uma hora do dia. A proposta do modelo PIPE evita recomputar a função preditiva *do zero*, dado o recebimento de novos dados de trajetórias. Pretende-se utilizar exclusivamente árvores binárias rotuladas que representem segmentos de ruas de uma cidade, as quais podem ser utilizadas para construir a função de predição. Como a solução proposta considera fluxos de dados (dinâmicos), cada fluxo é manipulado para identificar de que forma eles podem afetar a composição da árvore existente. A definição do problema é mostrada a seguir.

Definição 5 (Definição do Problema) Dada uma janela de tempo T_i e o fluxo contínuo de trajetórias $S_i = \{ST_1^i, ST_2^i, ..., ST_n^i\}$, o problema desta tese é rastrear o nó *não raiz* da árvore binária rotulada T a ser atualizado a partir de S_i, sem recomputar T a cada janela de tempo T_i. E ainda, construir, em tempo de execução, um preditor de tempo de viagem a partir de T, que recebe de entrada um instante de tempo t.

A busca e atualização da árvore binária rotulada T devem ocorrer dentro da próxima janela de tempo T_{i+1}. As avaliações experimentais realizadas para validar o modelo PIPE são abordadas no Capítulo 5 e mostram não só as análises relacionadas ao tempo de processamento para construção, busca e atualização do modelo, como também os resultados relacionados com a acurácia da solução.

4.2 PIPE: Um Preditor de Tempos de Viagem usando Fluxo Contínuo de Trajetórias

O modelo PIPE é incremental, uma vez que permite a criação e a manutenção de funções preditivas a partir do recebimento de novas trajetórias, sem a necessidade de construir

o modelo *do zero*. Em (NASCIMENTO *et al.*, 2016a), foi proposto um modelo de predição de tempos de viagem também incremental. No entanto, tal modelo é descontínuo e gera uma função de regressão desconexa. Esse modelo utiliza árvore de regressão binária para armazenar, em cada nó folha, uma regressão linear específica, que estima a duração das viagens de objetos, dado um intervalo de tempo.

A solução PIPE também utiliza um tipo de árvore binária. No entanto, PIPE garante a obtenção de uma função diferenciável. O modelo PIPE é a solução do problema apresentado nas **Questões de Pesquisa 1, 2, 3 e 4**, vistas na Seção 1.4, cuja proposta foi apresentada em (NASCIMENTO *et al.*, 2017).

As contribuições desse modelo podem ser divididas em duas etapas. Na primeira, propõe-se um algoritmo para construir a árvore binária rotulada, que foi escolhida por ser mais eficiente para computar, simples de interpretar (TØNDEL *et al.*, 2003) e flexível, permitindo utilizar facilmente um esquema que represente regras de decisão (LAGE *et al.*, 2008). Na segunda etapa, um algoritmo que possibilita a atualização da árvore binária rotulada é proposto. Essa atualização ocorre devido ao recebimento de um fluxo contínuo de trajetórias, que pode apresentar um novo comportamento dos objetos que estão se movendo. O algoritmo de atualização recebe como entrada as durações dos tempos de viagens e tem como saída a árvore binária atualizada. Essa estratégia é interessante para o recebimento de dados continuamente, uma vez que é necessário atualizar apenas a árvore, não necessitando os dados persistirem em uma base histórica. Assim, a atualização do modelo ocorre a partir de cálculos matemáticos, que consideram a média e o desvio padrão dos dados relacionados aos objetos de análise, cujos cálculos são definidos de forma semelhante a uma sumarização e serão mostrados posteriormente.

O modelo PIPE estende a proposta (LAGE *et al.*, 2008), no sentido de considerar o comportamento dinâmico dos objetos em movimento ao longo do tempo. A construção e a atualização da árvore binária são processos que, recursivamente, dividem o conjunto de trajetórias em duas partes, usando a hora do dia como referência para essa divisão. A escolha do particionamento em dois intervalos iguais é baseada em (GUESSOUS *et al.*, 2014), considerando a distribuição dos dados de trajetória como uma mistura de distribuições Gaussianas (LIAO *et al.*, 2005), as quais correspondem a distribuições estatísticas.

Inicialmente, a partir do conjunto de dados de trajetórias, são computados os tempos inicial e final, que correspondem à primeira e à última hora do dia. Imagine, por exemplo, que o conjunto de dados contém informações a respeito dos tempos de viagens de objetos que

trafegaram num segmento de rua nas 24 horas do dia. Nesse caso, a primeira parte do conjunto se refere a dados que trafegaram no segmento entre 00h00min e 12h00min. Por outro lado, a segunda parte da divisão se refere aos dados entre 12h01min e 23h59min. Essa divisão pode ocorrer tantas vezes quantas forem necessárias, até que o desvio padrão dos objetos (σ), que estão contidos em um nó não raiz, seja menor ou igual a uma tolerância (δ), previamente definida pelo usuário. A forma de particionamento dos dados também justifica a escolha de árvores binárias. O conceito da árvore binária utilizada nesta proposta é formalizado na **Definição 4**.

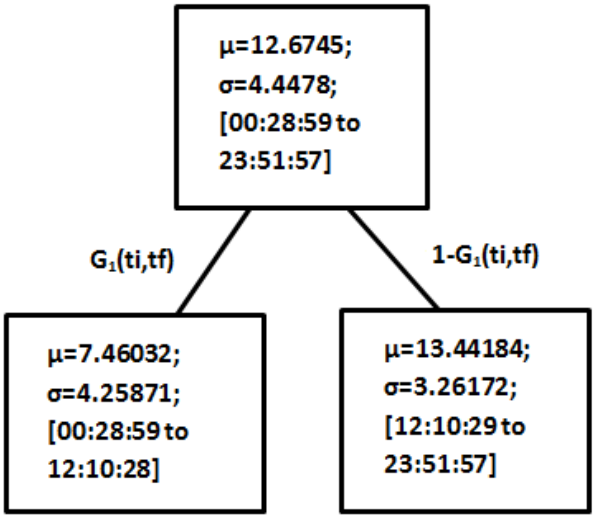
Para cada novo fluxo contínuo de dados, o modelo PIPE computa a média (μ) e o desvio padrão (σ) dos tempos de viagem. Além disso, uma função polinomial também é computada (Equação 4.3), a qual será apresentada no processo de construção da função, obtida a partir do modelo PIPE. Neste modelo, qualquer *nó não raiz à esquerda* computa a função polinomial, cuja solução garante a curva de suavização do modelo. O cálculo do polinômio, obtido anteriormente, é usado para computar a função polinomial do *nó não raiz à direita*. A função polinomial é usada para melhor ajustar a suavização da função preditiva e considera em sua construção a distribuição dos dados de trajetórias. Essa função será detalhada quando descrita a utilização da Equação 4.3.

A Figura 4 mostra o processamento da árvore binária rotulada, quando computada a partir dos dados que representam as localizações enviadas via o GPS dos ônibus, que trafegaram os segmentos de ruas da cidade do Rio de Janeiro/Brasil, entre os meses de agosto e setembro de 2015. A Figura 5 mostra a função gerada pelo modelo PIPE, dada a distribuição dos dados, considerando o tempo de travessia dos objetos, em um segmento de rua, e hora do dia. Assim, no *eixo y* estão as durações de viagens dos objetos, indicando o tempo total de travessia do segmento em análise e, no *eixo x*, a hora do dia. A primeira divisão da árvore ocorre a partir do valor médio dos tempos de partida (isto é, das horas do dia), que, no caso do exemplo ilustrado, acontece às 12h00min.

Cada nó $n_k \in T$, sendo ele raiz ou não, é rotulado pela tupla (μ_k, σ_k, $[t_s, t_e]$), como mostrado na **Definição 4**. Sendo μ_k a média dos tempos de viagens, e σ_k o desvio padrão, ambos calculados a partir de subtrajetórias induzidas pelo intervalo de tempo $[t_s, t_e]$, onde t_s e t_e são instantes de tempo e $t_s < t_e$.

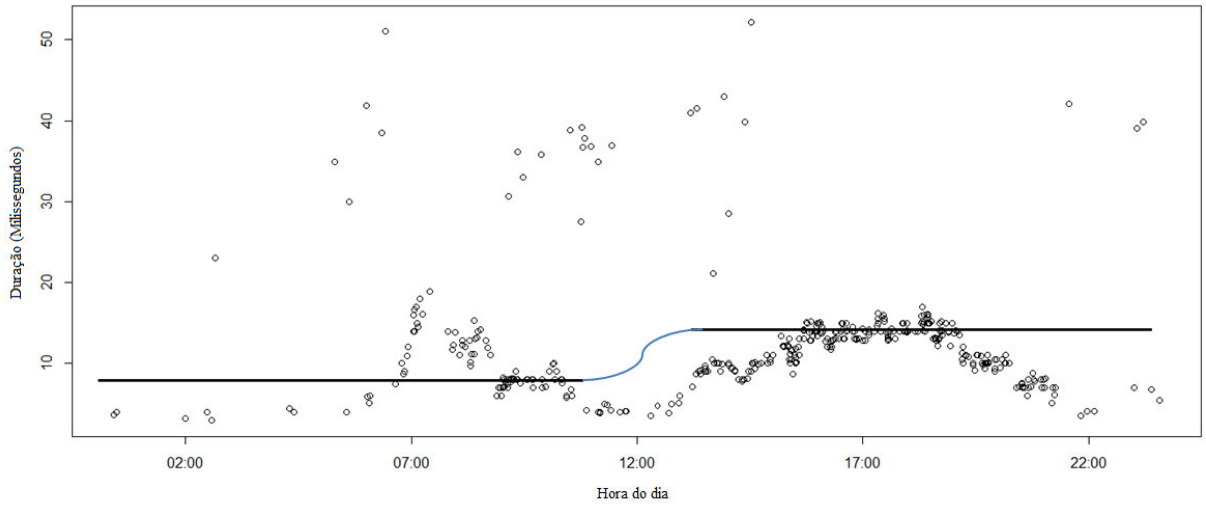
A Figura 5 mostra a função preditiva alcançada a partir da utilização do modelo PIPE, a qual é construída após a criação da árvore binária (mostrada na Figura 4). O processo de divisão da árvore é recursivo e ocorre até que a condição de parada seja atingida, que é

Figura 4 – Árvore Binária Rotulada.



Fonte: da autora.

Figura 5 – Função obtida a partir da Árvore Binária Rotulada (Figura 4).



Fonte: da autora.

até o nó n_k apresentar o resultado relacionado ao desvio padrão das trajetórias (σ_k) menor ou igual à tolerância (δ_k), previamente definida pelo usuário – ou seja, até que $\sigma_k \leq \delta_k$. O valor da tolerância pode ser definido pelo usuário ou pode ser computado a partir da seguinte combinação linear: $\alpha * \sigma + (1 - \alpha) * \sigma'$, onde α é o valor usado para balancear a importância do passado e do presente; e os valores de σ e σ' correspondem aos valores dos desvios padrões anterior e atual, respectivamente. Este cenário é mais realístico, uma vez que as condições de tráfego podem mudar e, nesse caso, a tolerância deve mudar também.

O Algoritmo 1 mostra como a árvore binária é construída. Os parâmetros de entrada para esse algoritmo são: o conjunto de trajetórias TR, cujos valores serão usados para a criação da árvore, e a tolerância (δ), que limitará a profundidade da árvore. A saída do algoritmo é a

árvore T construída. Inicialmente, é criado um nó raiz, que mantém os valores das médias e desvios padrão (linhas 2 a 4). Esses valores são computados com base no retorno da função *getTempoViagemCom()*, cujos parâmetros são formados pelo conjunto de trajetórias que pertencem a um intervalo de tempo. Basicamente, a função *getTempoViagemCom()* calcula a duração das viagens, a partir do recebimento dos dados de trajetórias. Nesse caso, cada ponto de localização, pertencente à trajetória, é considerado pela função, que computa a duração das travessias. A partir desse resultado, os cálculos das médias e desvios padrão dos tempos de trajetórias são realizados, e a condição para a criação de um novo nó é analisada, como mostrado na linha 5. Novos nós são criados apenas se o desvio padrão (σ) do nó n é maior que a tolerância δ. Adicionalmente, o volume de dados, que pertence a esse nó, precisa ser maior que 1. Se ambas as condições forem satisfeitas, dois novos nós (um à esquerda e outro à direita) são criados, e o conjunto de dados é dividido em duas partes. A primeira partição computa a média ($\mu_{esquerda}$) e o desvio padrão ($\sigma_{esquerda}$) para o novo nó à a esquerda (linhas 7 e 8). A segunda partição computa os mesmos cálculos ($\mu_{direita}$ e $\sigma_{direita}$) para o novo nó à direita (linhas 9 e 10). Finalmente, esses nós são criados na árvore, cada um contendo suas respectivas informações (linhas 11 a 14).

Algoritmo 1: Computando a Árvore Binária Rotulada – Modelo PIPE

Input:
- Conjunto de Trajetórias TR, com o intervalo de tempo $[t_i, t_f]$;
- Tolerância δ.

Output: Árvore Binária Rotulada T.

begin
- $\mu \leftarrow$ computeμ(getTempoViagemCom(TR,$[t_i,t_f]$));
- $\sigma \leftarrow$ computeσ(getTempoViagemCom(TR,$[t_i..t_f]$);
- $n_{raiz} \leftarrow$ criaNo(μ, σ, $[t_i, t_f]$));
- **if** *(computeσ(getTempoViagemCom($TR, [t_i,t_f]$)) $> \delta$ AND $TR.lenght > 1$)* **then**
  - $tempo\_medio \leftarrow ((t_f$ - $t_i)$ / 2) + t_i;
  - $\mu_{esquerda} \leftarrow$ computeμ(getTempoViagemCom(TR,$[t_s$,$t_{tempo\_medio}]$));
  - $\sigma_{esquerda} \leftarrow$ computeσ(getTempoViagemCom(TR,$[t_s..t_{tempo\_medio}]$);
  - $\mu_{direita} \leftarrow$ computeμ(getTempoViagemCom(TR,$[t_{tempo\_medio}..t_e]$);
  - $\sigma_{direita} \leftarrow$ computeσ(getTempoViagemCom(TR,$[t_{tempo\_medio}..t_e]$);
  - $n_e \leftarrow$ criaNo($\mu_{esquerda}$, $\sigma_{esquerda}$, $[t_s, t_{tempo\_medio}]$);
  - $n_d \leftarrow$ criaNo($\mu_{direita}$, $\sigma_{direita}$, $[t_{tempo\_medio}, t_e]$));
  - addNoEsquerda(n_{raiz},n_e);
  - addNoDireita(n_{raiz},n_d);
- **end**

end

Fonte: da autora.

O conjunto de trajetórias recebidas, no intervalo de tempo $[t_i, t_f]$, pode atualizar a árvore binária rotulada, e as mudanças causadas por essa atualização podem afetar o resultado da função preditiva. Na literatura, é possível encontrar limitações acerca dos modelos de predição, quando são recebidos novos dados de trajetórias, porque essa atualização pode causar

a descontinuidade da função temporal obtida (NASCIMENTO *et al.*, 2016a). A solução PIPE resolve esse problema porque atualiza apenas a árvore binária e não a função temporal. É necessário lembrar que cada nó n_k da árvore T, retornada pelo Algoritmo 1, seja ele raiz ou não, é composto pela tupla $(\mu_k, \sigma_k, [t_s, t_e])$, onde $t_s < t_e$. Seja $S = \{TR_1, ..., TR_n\}$ um novo fluxo contínuo de trajetórias, o qual é recebido em um intervalo de tempo $[t_i, t_f]$, onde $t_i < t_f$, é possível realizar a manutenção incremental do modelo, para refletir o comportamento dos novos dados – contidos em S_i. Nesse caso, para realizar a manutenção incremental, o modelo PIPE usa duas diferentes equações: uma é responsável por atualizar a média do conjunto de valores, e a outra por atualizar o desvio padrão, ambos pertencentes à tupla (μ_k, σ_k, $[t_s, t_e]$) e modificados a partir do recebimento do novo fluxo de dados S_i. As equações são baseadas em (SONG; WANG, 2005) e apresentadas a seguir.

$$\mu'_j = \frac{\mu_j + (\mu_{S_i} - \mu_j)}{lenght(S_i) + 1} \tag{4.1}$$

$$(\sigma'_j)^2 = \frac{lenght(S_i) \cdot (\sigma_j^2 + \mu_j^2) + \mu_j'^2}{lenght(S_i) + 1} - \mu_j'^2 \tag{4.2}$$

A Equação 4.1 computa a nova média μ'_j para o nó n_j, considerando que as trajetórias contidas em S_i estão no mesmo intervalo de tempo do nó n_j. Temos que μ_j é a média previamente computada no nó de análise e μ_{S_i} corresponde à média dos tempos de viagens, recebidas a partir do fluxo contínuo de trajetórias S_i e *lenght(S_i)* corresponde a quantidade de trajetórias contidas no fluxo de dados S_i. Já a Equação 4.2 computa o novo desvio padrão σ'_j para o nó n_j, onde σ_j e μ_j^2 são, respectivamente, os valores do desvio padrão e da média, previamente computados; e $\mu_j'^2$ é o valor da nova média, obtida a partir da Equação 4.1.

Quando um novo fluxo contínuo de trajetórias é recebido, o modelo PIPE investiga se um nó ou um conjunto de nós deve ser atualizado. Inicialmente, os valores discrepantes são identificados com base na média, no desvio padrão e no tamanho da amostra. Caso esse tipo de dado seja descoberto, ele será removido, porque seu comportamento pode causar um desbalanceamento desordenado na árvore binária, gerando, para esta proposta, análises imprecisas. Após a inspeção acerca dos valores discrepantes, a busca pelo nó a ser atualizado deve ocorrer e é feita com base no intervalo de tempo em que está contido o novo conjunto de dados. Imagine que o fluxo contínuo de trajetórias reflete o intervalo de tempo $[t_i, t_f]$, onde $t_i < t_f$. Ademais, imagine que o nó n_k, ou o conjunto de nós $N = \{n_1, ..., n_m\}$ tem informações de trajetórias contidas no

mesmo intervalo de tempo $[t_i, t_f]$. As informações contidas no nó n_k devem ser atualizadas com base nas Equações 4.1 e 4.2. Se o desvio padrão computado for maior que a tolerância, previamente definida pelo usuário (isto é $\sigma_k > \delta$), dois novos nós serão criados. Dessa forma, cada nó da árvore pode gerar apenas dois outros nós. Essa atualização ocorre recursivamente, até que um critério de parada seja atingido, ou seja, até que o nó n_k apresente $\sigma_k \leq \delta$ ou até que o volume de dados seja menor ou igual a 1. Observe que essa abordagem é simples e fácil de implementar.

O Algoritmo 2 computa a atualização da árvore binária rotulada. O algoritmo para manutenção incremental do modelo PIPE será chamada PIPE*. Esse algoritmo tem como entrada a árvore binária, que sofre as atualizações necessárias, dado um novo fluxo contínuo de trajetórias (S_i) e o valor da tolerância (δ), que é definido pelo usuário e determina o crescimento da árvore. A saída do algoritmo é a árvore atualizada. Inicialmente, o conjunto de dados, recebido como fluxo contínuo de trajetórias, é analisado no sentido de identificar se existem valores discrepantes contidos no mesmo. Caso sejam encontrados, esses valores são descartados porque podem causar um desbalanceamento na árvore a ser construída. Caso não existam, o Algoritmo realiza um processo de busca na árvore para identificar qual nó precisa ser atualizado. Vale lembrar que qualquer novo conjunto de trajetórias se refere ao intervalo de tempo $[t_s, t_e]$. O nó à esquerda será atualizado se o valor do desvio padrão, computado a partir dos dados recebidos, é maior que a tolerância previamente definida, isto é, se $\sigma' > \delta$, onde σ' é o desvio padrão de todo o conjunto de dados (históricos e os obtidos a partir de S_i). Além disso, verifica-se também se S contém dados que ocorrem no intervalo de tempo desse nó (linhas 2 a 7). Se essas condições forem verdade, as linhas de 8 a 10 serão computadas. Os cálculos da média e do desvio padrão são realizados com base no novo conjunto de dados (linhas 8 e 9), os quais usam as Equações 4.1 e 4.2. Se a condição mostrada na linha 7 não for satisfeita, o nó à direita é analisado, como mostrado das linhas 12 a 17, cujo cálculo do desvio padrão se repete, agora considerando os dados pertencentes ao nó à direita. Se a condição para atualizar o nó à direita for satisfeita, as linhas de 15 a 17 serão computadas. O Algoritmo 2 é recursivo (linhas 10 e 17) e continua criando novos nós até que uma condição de parada seja satisfeita – quando $\sigma_k \leq \delta$.

A Figura 6 mostra a atualização da árvore binária da Figura 4, dado o recebimento de novos dados de trajetórias. O particionamento do nó ocorre nos dois nós folhas da Figura 4, gerando quatro novos nós, mostrados na Figura 6, cuja árvore produz a função mostrada na Figura 7. Outro exemplo de atualização da função é mostrado na Figura 8, que identifica

as mudanças do tráfego na cidade do Rio de Janeiro/Brasil e mostra como essas mudanças se refletem na obtenção da função preditiva. Considere o cenário da Figura 8(a), que mostra função preditiva, criada a partir da árvore binária, obtida no Algoritmo 1. Essa árvore representa os tempos de viagens em um segmento de rua específico, no mês de Agosto de 2015. Suponha que essa árvore precisa ser atualizada para representar a atual situação do tráfego. Dentro desse contexto, são reportados, continuamente, novos dados de trajetórias, as quais estão contidas em horários específicos da manhã, compreendendo informações entre o intervalo de tempo relativo das 07h:00min e 09h:00min. É possível observar que, em outros horários, a profundidade da árvore se mantém. O resultado dessas atualizações reflete-se na computação da função temporal, que é mostrada na Figura 8(b).

Algoritmo 2: Algoritmo AtualizaArvoreContinua: Modelo PIPE*

Input:
- n_{raiz}, que é um nó na árvore binária rotulada T;
- Conjunto de Trajetórias S_i;
- Tolerância δ.

Output: Árvore Binária Rotulada T

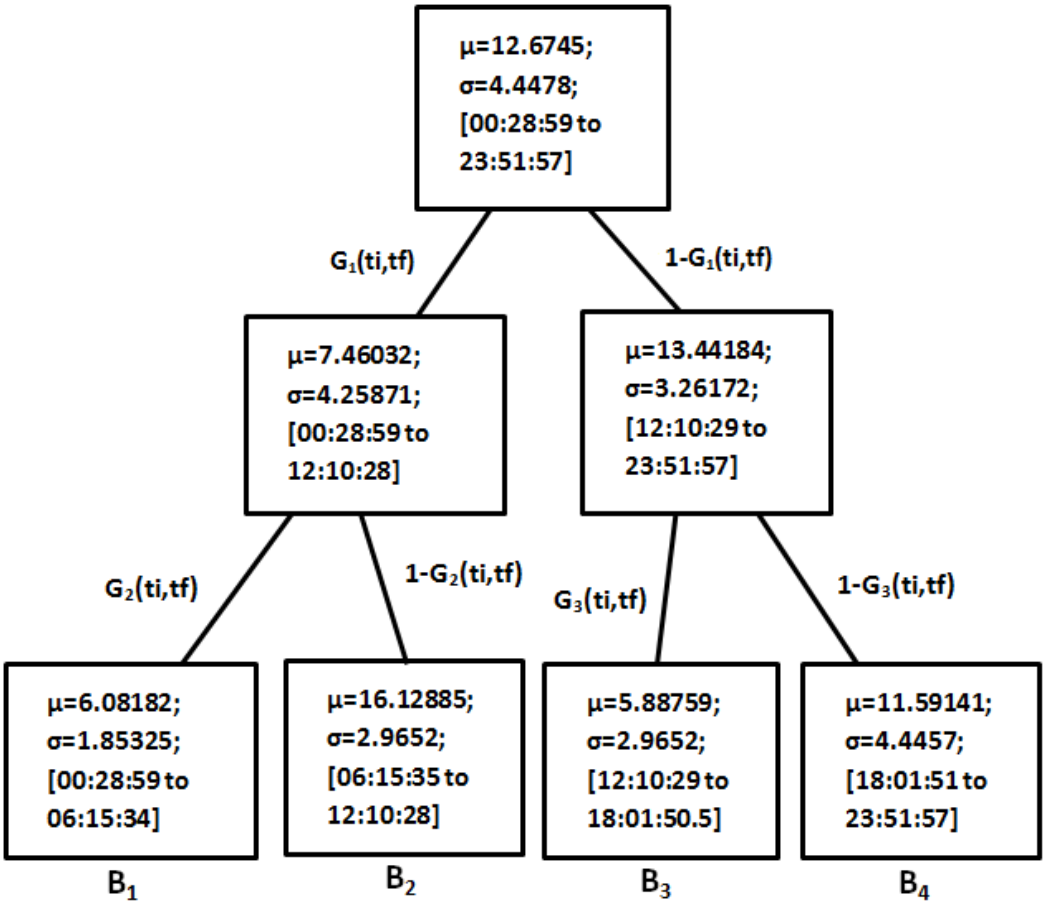
begin
  if *buscaValorDiscrepante(S_i)* **then**
    return AtualizaConjuntoTrajetorias(S_i) ;
  end
  $n_{esquerda} \leftarrow getNoEsquerda(n_{raiz})$;
  $\sigma'_{esquerda} \leftarrow$ compute$\sigma'(n_{esquerda}, S_i)$;
  if *($\sigma'_{esquerda} > \delta$) AND ($n_{esquerda}[t_s] \geq S_i[t_f]$)* **then**
    $\mu_{esquerda} \leftarrow$ computeμ'(getTempoViagemCom(S_i,[t_s,t_e]));
    $\sigma_{esquerda} \leftarrow$ computeσ'(getTempoViagemCom(S_i,[t_s..t_e]);
    return AtualizaArvoreContinua($n_{esquerda}$, S_i, δ)
  end
  $n_{direita} \leftarrow getNoDireita(n_{raiz})$;
  $\sigma'_{direita} \leftarrow$ compute$\sigma'(n_{direita}, S_i)$;
  if *($\sigma'_{direita} > \delta$) AND ($n_{direita}[t_s] \leq S_i[t_f]$)* **then**
    $\mu_{direita} \leftarrow$ computeμ'(getTempoViagemCom(S_i,[t_s,t_e]));
    $\sigma_{direita} \leftarrow$ computeσ'(getTempoViagemCom(S_i,[t_s..t_e]);
    return AtualizaArvoreContinua($n_{direita}$, S_i, δ)
  end
end

Fonte: Fonte: da autora.

A função temporal é computada em tempo de execução, dada a hora do dia t, definida pelo usuário. Nesse momento, realiza-se uma busca na árvore binária até que seja encontrado o nó $n_k = (\mu_k, \sigma_k, [t_s, t_e])$, que armazena a hora t requerida, isto é, $t \in [t_s, t_e]$. O caminho da raiz até a folha nos leva a obter a função temporal f, cuja formalização é definida a seguir.

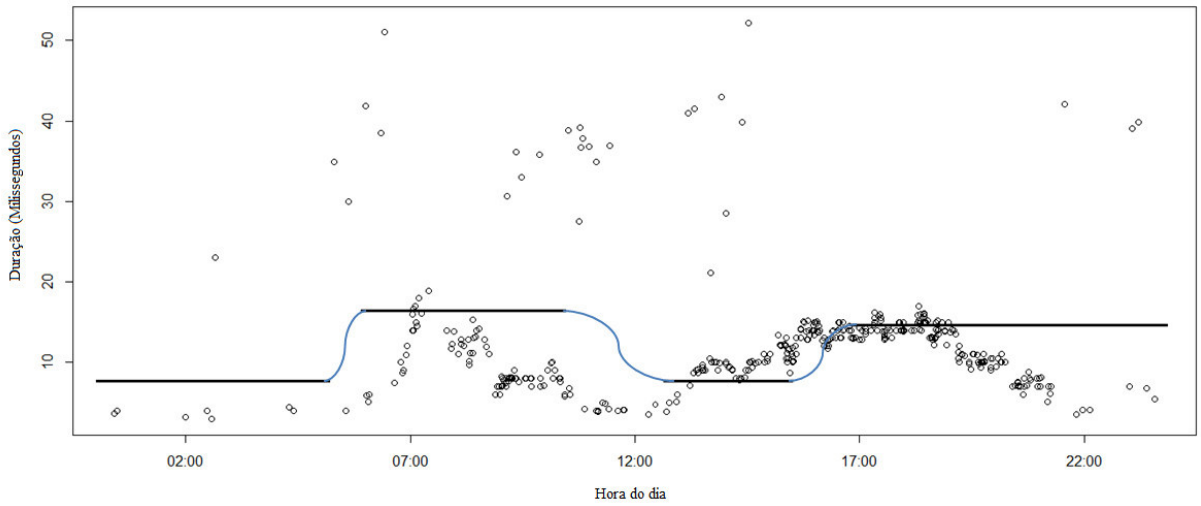
Definição 6 (Função Temporal) Dada a Árvore Binária T, a *Função Temporal* f no instante

Figura 6: Atualização da Árvore Binária Rotulada, a partir do Algortimo 2.



Fonte: da autora.

Figura 7: Função obtida a partir da Árvore Binária Rotulada, criada a partir do Algortimo 1. (Figura 6).



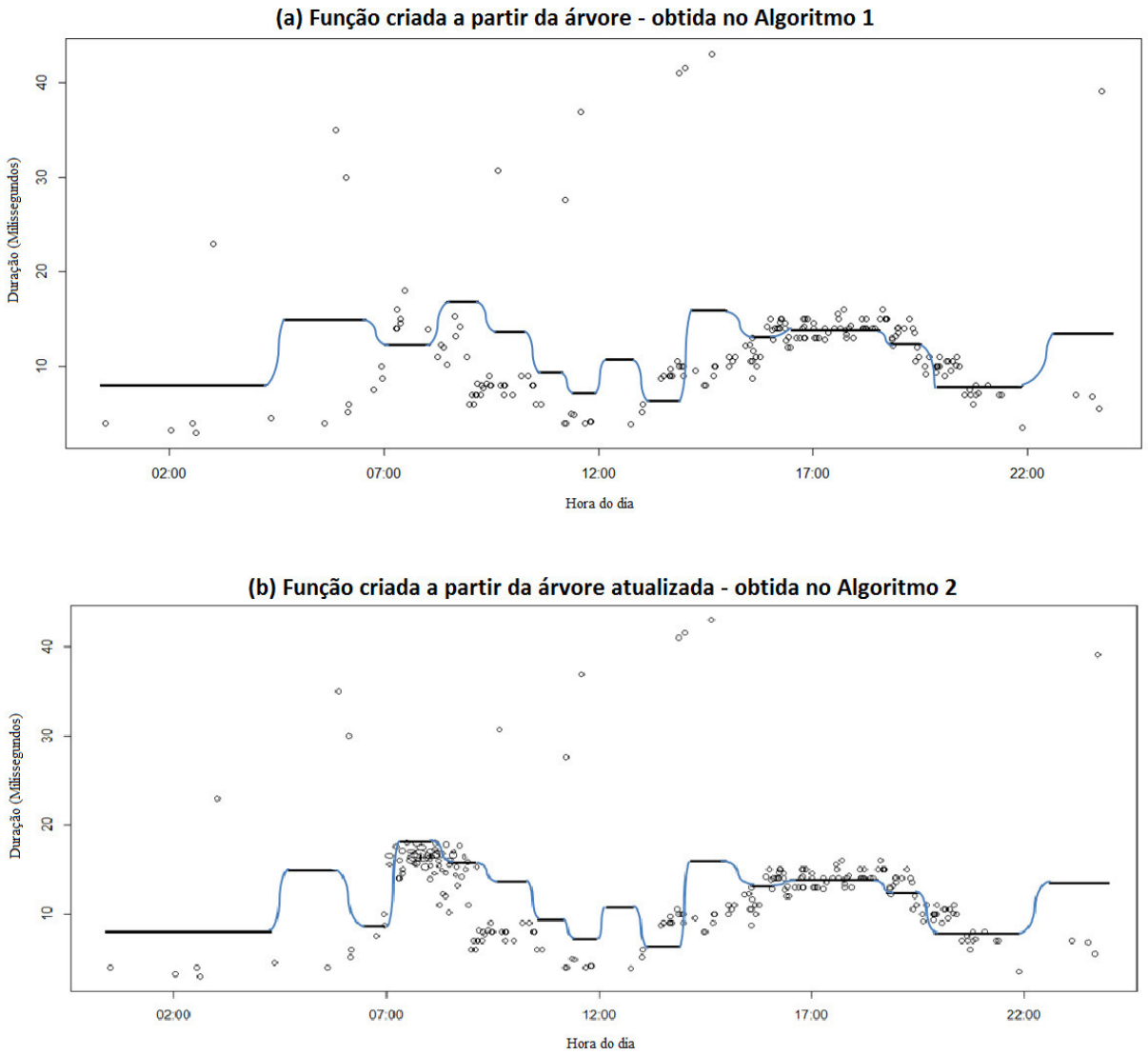
Fonte: da autora.

de tempo t é computada usando a expressão

$$f(t) = \sum_{i=1}^{k} B_i(t) \cdot \mu_i,$$

onde k é o número de níveis da árvore, o qual é composto por $n_i = (\mu_i, \sigma_i, [t_s, t_e]) \in T$ e $B_i(t)$ é a

Figura 8: Atualização da Função, dado o recebimento do Fluxo Contínuo de Trajetórias.



Fonte: da autora.

função que garante a diferenciabilidade de f.

As funções de transição B_i precisam ser computadas para cada nó da árvore. Cada função definida em B_i corresponde ao produto de funções de transição G_k (ou $(1 - G_k)$) associado ao nó, pertencente ao caminho da raiz até a folha. A associação da função de transição segue uma regra: se o caminho percorrido da raiz para o nó k é à esquerda, então a função de transição usada será G_k; caso contrário, será (1 - G_k). Aqui, $G_i(t)$ representa uma família de funções de transição, parametrizadas por λ (LAGE *et al.*, 2008). A proposta de (KUBRUSLY; LOPES, 2015) foi estendida para computar o polinômio, cujo coeficiente de grau 5 foi escolhido, no propósito de melhor ajustar a suavização da função, dada a distribuição dos dados. A função $G_i(t)$ depende da distância do plano da divisão temporal, definida pelo parâmetro λ. $G_i(t)$ pode

ser definida pela equação seguinte.

$$G_i(t) = \begin{cases} 0, \text{se } t \leq -1/\lambda \\ 1, \text{se } t \geq 1/\lambda \\ \frac{1}{2} + \frac{15 \cdot \lambda}{16} \cdot t - \frac{5 \cdot \lambda^3}{8} \cdot t^3 + \frac{3 \cdot \lambda^5}{16} \cdot t^5, \text{caso contrário.} \end{cases} \tag{4.3}$$

O parâmetro λ foi escolhido para cada nó de acordo com a seguinte fórmula: $|((((t_e - t_s)/2) + t_s) - t)|$, criado no intuito de particionar o intervalo de tempo em dois intervalos iguais.

Para exemplificar, considere o exemplo da Figura 6, que apresenta uma árvore binária com quatro nós folhas – isso significa que cada nó folha i apresenta B_i, onde $i = \{1, \ldots, 4\}$. Assim, cada B_i é computado conforme a seguinte Equação:

$$\begin{aligned} B_1(t) &= G_1(t) \cdot G_2(t) \\ B_2(t) &= G_1(t) \cdot (1 - G_2(t)) \\ B_3(t) &= (1 - G_1(t)) \cdot G_3(t) \\ B_4(t) &= (1 - G_1(t)) \cdot (1 - G_3(t)) \end{aligned} \tag{4.4}$$

4.3 Conclusão do Capítulo

Neste capítulo, foi apresentado o modelo *PIPE: Um Preditor de Tempos de Viagem usando Fluxo Contínuo de Trajetórias*, que corresponde a uma solução incremental e gera, como resultado, uma função preditiva contínua. Esse modelo foi proposto devido aos impedimentos encontrados nas soluções existentes na literatura, a exemplo do modelo *Incremental Descontínuo*, publicado em (NASCIMENTO *et al.*, 2016a), cuja solução gera uma função preditiva desconexa e, segundo (LUO *et al.*, 2015), a descontinuidade de uma solução preditiva pode degradar o desempenho do algoritmo que a utiliza.

O modelo PIPE utiliza árvores binárias rotuladas para a obtenção da solução e permite realizar a manutenção incremental dessa árvore, dado o recebimento de fluxos contínuos de trajetórias. Quando novos dados são recebidos, realiza-se uma análise para verificar se o conjunto de informações indicam mudanças na distribuição dos dados; se isso for comprovado, o modelo é atualizado para possibilitar a computação de novas funções preditivas, as quais buscam refletir os resultados relacionados à árvore binária modificada e permitem responder a requisições em tempo real.

No próximo Capítulo, será apresentada a avaliação experimental dessa proposta. Tal avaliação serve para validar o modelo PIPE, mostrando resultados relacionados tanto ao desempenho dos algoritmos, quanto à acurácia desta solução.

5 AVALIAÇÃO EXPERIMENTAL

Neste capítulo, são mostrados os resultados experimentais realizados para avaliar o modelo PIPE, apresentado no Capítulo 4. Essa proposta foi implementada nas linguagens de programação Java e R. Os dados utilizados na avaliação experimental, obtidos via dispositivos de GPS, são reais e correspondem às informações de trajetória dos ônibus do Rio de Janeiro e dos táxis de Fortaleza.

Todos os experimentos foram conduzidos em duas máquinas Intel Core 2, com CPU Q6600 server, 8GB de RAM e 2.40GHz, usando-se o sistema operacional Ubuntu 11.10, de 64 bits.

5.1 Descrição do Conjunto de Dados

O conjunto de dados usados para validar o modelo PIPE se refere aos ônibus da cidade do Rio de Janeiro/Brasil, os quais foram disponibilizados pela Prefeitura da cidade (RIO, 2017c).

Neste trabalho, apenas os dados referentes às terças-feiras de agosto e setembro de 2015 e 2016 foram usados, uma vez que é comum uma via apresentar a mesma velocidade em determinado instante de tempo, em dias da semana, e outra velocidade em fins de semana ou feriados. Dessa forma, neste experimento, espera-se que os tempos de viagem dos objetos sigam uma mesma distribuição ao longo das terças-feiras do mesmo mês e ano, não considerando os feriados ou finais de semana. Além disso, na coleta de dados realizada, foi possível observar um maior volume de informações referentes às terças-feiras dos meses de agosto e setembro, em comparação com os demais dias da semana. Os dados foram obtidos a partir de dois diferentes segmentos de ruas, os quais incluem a Autoestrada Lagoa-Barra e a Avenida Ministro Ivan Lins. Esse conjunto de dados compreendeu em torno de 2.400 trajetórias.

5.2 Metodologia Experimental

A metodologia experimental usada neste trabalho busca investigar a acurácia e a eficiência dos modelos incrementais, que podem ter soluções contínuas e descontínuas. O objetivo é analisar qual modelo tem um melhor ajuste em relação à distribuição dos dados e qual deles tem melhor acurácia, a partir do cálculo do *Raiz Mean Square Error* (RMSE), quando comparados os valores esperados e preditos das soluções. A solução PIPE*, considerada

incremental, consiste na geração de uma função preditiva, criada a partir de um modelo de árvore binária. Essa solução foi comparada com uma solução competidora, chamada *Incremental Descontínua* (NASCIMENTO *et al.*, 2016a), a qual corresponde a um modelo de predição que também é modificado a partir do recebimento de novos dados de trajetórias e retorna uma função de predição. Essa estratégia retorna um conjunto de regressões lineares simples.

O modelo *Incremental Descontínuo* (NASCIMENTO *et al.*, 2016a) foi comparado com a solução competidora *Piecewise* e apresentou como resultado uma função que melhor se ajusta à distribuição dos dados de trajetória. Dessa forma, é aceitável verificar se a solução PIPE* computa melhores resultados que a solução *Incremental Descontínua*. A Tabela 4 apresenta, resumidamente, as diferenças entre esses modelos incrementais.

Tabela 4: *Incremental Descontínuo* x PIPE*

***Incremental Descontínuo* (solução proveniente de (NASCIMENTO *et al.*, 2016a))**	**PIPE* (solução proveniente de (NASCIMENTO *et al.*, 2017))**
Permite o processamento de fluxos contínuos de trajetórias.	Permite o processamento de fluxos contínuos de trajetórias.
Persiste os dados de trajetória.	Não persiste os dados de trajetória.
Função obtida a partir de uma árvore binária.	Função obtida a partir de uma árvore binária.
Retorna uma regressão linear simples, dada uma hora do dia.	Retorna uma função diferenciável, dada uma hora do dia.
Gera uma função de predição descontínua, com base na soma das regressões lineares	Gera uma função de predição contínua.

Fonte: da autora.

Além das diferenças mostradas na tabela anterior, é possível realizar uma comparação experimental. Dessa forma, ambas as soluções foram avaliadas, simulando o recebimento de fluxos contínuos de trajetórias, considerando os dados de todas as terças-feiras de agosto de 2015, os quais foram usados para gerar as primeiras árvores binárias. Já as trajetórias dos dados dos ônibus, que correspondem às terças-feiras de setembro de 2015, também foram simuladas como fluxos contínuos, no entanto, tais dados atualizam as árvores binárias criadas anteriormente. A atualização da árvore pode ocorrer em vários nós incrementalmente, e seu resultado pode refletir a situação real do tráfego.

Outra análise realizada neste capítulo busca comparar e investigar tanto a eficiência, quanto a acurácia, do modelo PIPE. Para essa análise, o processo de avaliação do modelo considerou duas diferentes construções: *(i)* uma abrange todo o conjunto de dados e gera um modelo de árvore binária, criada *do zero*, que continuará sendo chamada PIPE; e o outro *(ii)* abrange a manutenção incremental da árvore binária, atualizada a partir do recebimento de novos fluxos de trajetórias, a qual será chamada PIPE*.

Os dados usados nesses experimentos contêm informações sobre as linhas de ônibus, descritas na Seção 5.1. A solução PIPE foi construída a partir dos dados de agosto, e a solução PIPE*, que corresponde à manutenção incremental da árvore binária, é obtida a partir do recebimento dos dados de setembro. Os experimentos executados tiveram diferentes objetivos. O primeiro buscou investigar como as mudanças causadas na reengenharia do trânsito – devido aos jogos Olímpicos, realizados em agosto de 2016 – afetaram o tempo de travessia dos ônibus para as rotas analisadas. Ademais, esses experimentos buscaram avaliar métricas como precisão e tempo de processamento dos modelos.

Quando a análise da acurácia se refere à função temporal, este trabalho assumiu que o tempo estimado da viagem coincide quando a diferença entre o valor predito e o valor real é de até 2 minutos. O limite de 2 minutos é aceitável porque o tempo de viagem no conjunto de dados variou em até 12 minutos. Para simular o envio de trajetórias por meio do fluxo contínuo em tempo real, o algoritmo *Storm* (STORM, 2016) foi utilizado.

Neste trabalho, o *Map Matching* funciona como um pré-processamento, que estima tanto a correta trajetória dos objetos coletados, quanto a velocidade média do percurso, em cada segmento de rua. O ajuste dos dados de trajetória foi realizado pelo algoritmo *Barefoot* (BAREFOOT, 2017). Dentro desse contexto, a fim de executar o *Map Matching* para os dados de trajetória, descritos na Seção 5.1, também foram utilizados os dados do *Open Street Map* (OSM) do Rio de Janeiro, representados por um conjunto de informações que contêm os segmentos das ruas da cidade.

5.3 Métricas dos Experimentos

Os modelos incrementais foram medidos a partir das análises de desempenho, considerando o tempo de processamento para a atualização das árvores binárias e o tempo de construção das funções incrementais. Ademais, foram realizadas medições acerca da acurácia dos modelos, identificando a quantidade de acertos entre os valores esperados e os valores

preditos.

Especificamente, para a metodologia proposta na Seção 5.2, foi analisado qual modelo melhor explica a distribuição dos dados. Nesse caso, os valores de *Akaike Information Criterion* (AIC) (AKAIKE, 1974) foram computados. Essa análise é importante porque, de acordo com (NASCIMENTO *et al.*, 2016a), o modelo *Incremental Descontínuo* gera uma função preditiva, que tem melhor ajuste quanto a distribuição dos dados de trajetórias, quando comparado à solução competidora *Piecewise*. Porém, resta comparar a estratégia *Incremental Descontínuo* com a solução PIPE*.

O algoritmo $k - fold$ (DUDA *et al.*, 2012) foi utilizado para os testes experimentais. O valor de k foi definido como 10. O conjunto de dados foi dividido randomicamente em 70% para treino do modelo e 30% para teste.

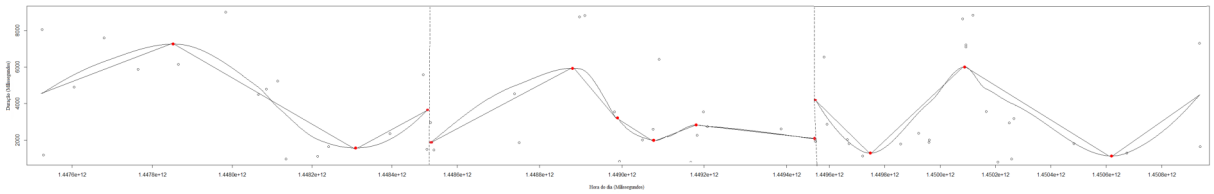
5.4 Comparando a Acurácia e o Tempo de Processamento das Soluções Incrementais

A avaliação experimental referida nesta seção busca investigar a acurácia das soluções incrementais, que são construídas com base na criação de árvores binárias, previamente compostas por dados históricos e mantidas incrementalmente, dado o recebimento de novos fluxos contínuos de trajetórias. Dessa forma, foi realizada uma análise acerca do ajuste dos modelos *Incremental Descontínuo* e PIPE*, para identificar qual deles melhor explica a distribuição dos dados. Os resultados obtidos são mostrados nas Figuras 9 e 10.

A Tabela 5 mostra os resultados dos valores de AIC, computados para as soluções *Incremental Descontínuo* e PIPE*, dado o recebimento de novos conjuntos de trajetórias, com intervalos de uma hora, entre às 05h00 e 07h00 do dia 08 de setembro de 2015. A análise do recebimento das novas trajetórias foi realizada por hora para facilitar a visualização das mudanças geradas. Nesse caso, é possível identificar que a função preditiva, obtida a partir do modelo PIPE*, tem um melhor ajuste, dada a distribuição dos dados, quando comparado com o modelo *Incremental Descontínuo*, uma vez que o valor obtido do AIC é menor. Além disso, a solução PIPE* garante a construção de uma função diferenciável. Isso ocorre porque a manutenção incremental desse modelo atualiza diretamente na árvore binária, diferentemente da estratégia descontínua, que computa as mudanças diretamente na função (NASCIMENTO *et al.*, 2017).

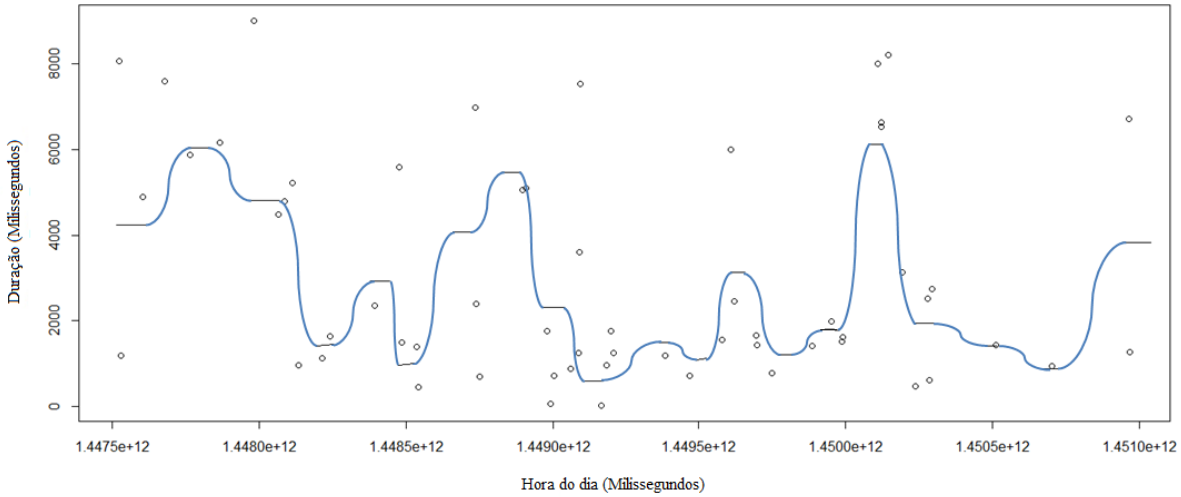
É importante comparar, ainda, a qualidade das funções preditivas, que foram obtidas a partir das construções dos modelos incrementais. Assim, as estratégias incrementais foram

Figura 9: Geração da função preditiva a partir do modelo *Incremental Descontínuo*.



Fonte: da autora.

Figura 10: Geração da função preditiva a partir do modelo PIPE*.



Fonte: da autora.

Tabela 5: Akaike Information Criterion – Modelos Incrementais

	Primeiro Fluxo de Trajetórias	Segundo Fluxo de Trajetórias	Terceiro Fluxo de Trajetórias
Incremental Descontínuo	1103.554	957.423	741.325
PIPE*	785.32	458.245	584.325

Fonte: da autora.

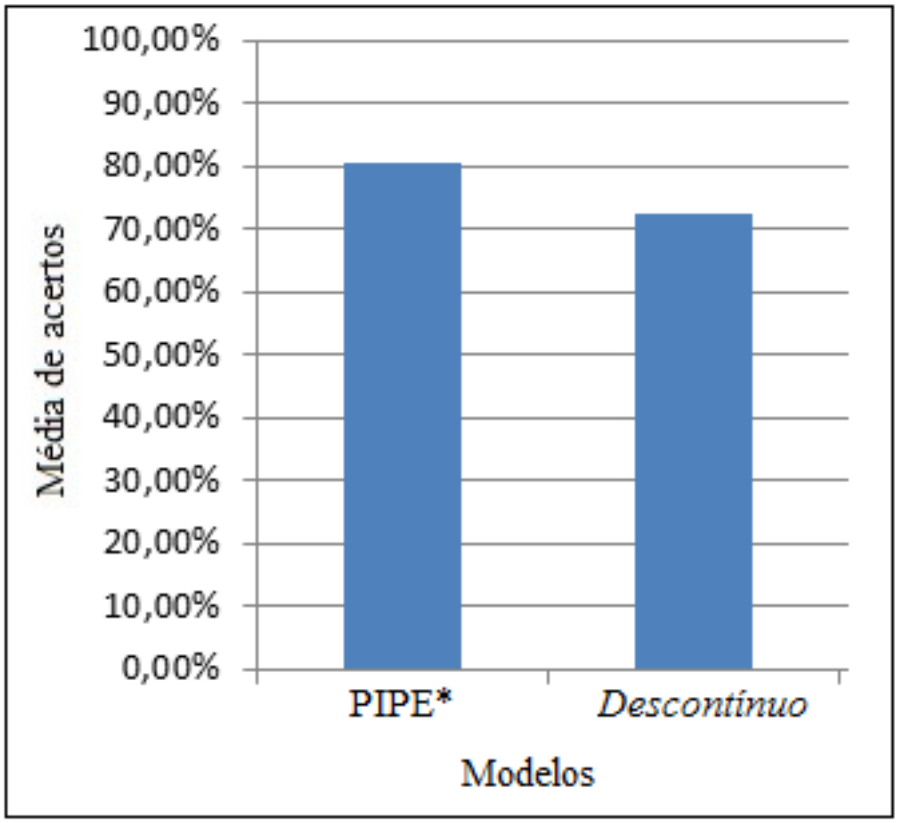
comparadas quanto à acurácia e ao tempo de processamento das soluções.

Para construir a árvore binária da solução PIPE*, o valor da tolerância (δ) foi considerado igual a 0.5. Neste experimento, as árvores binárias foram construídas a partir dos dados de todas as terças-feiras de agosto de 2015 e atualizadas a partir dos dados de todas as terças-feiras de setembro de 2015, que foram simulados como se chegassem em fluxos contínuos de trajetórias. As funções temporais, para esses experimentos, foram criadas em intervalos de 15 minutos.

Para realizar as análises, os dados foram divididos em 70% para treinamento e 30% deles para testes, conforme discutido na Seção 5.3, seguindo a estratégia do Algoritmo *k-fold*. Para reduzir o escopo de análise, os dados utilizados nesses experimentos correspondem ao período de 12h00 a 23h59 (nos dois meses de análise, isto é, agosto e setembro).

A Figura 11 mostra os resultados obtidos, indicando que a acurácia da função temporal para a solução PIPE* atingiu, em média, 81% de acertos, para os dados de teste – enquanto a solução *Incremental Descontínua* obteve, em média, 72% de acertos. Além disso, a Figura 12 mostra as médias dos valores de *RMSE*, obtidas para as 10 execuções (Algoritmo $k - fold$). Observando-se esses resultados, é possível afirmar que a solução PIPE* tem melhor acurácia que a solução *Incremental Descontínua*, pois a primeira erra menos do que a segunda e apresenta resultados de melhor qualidade.

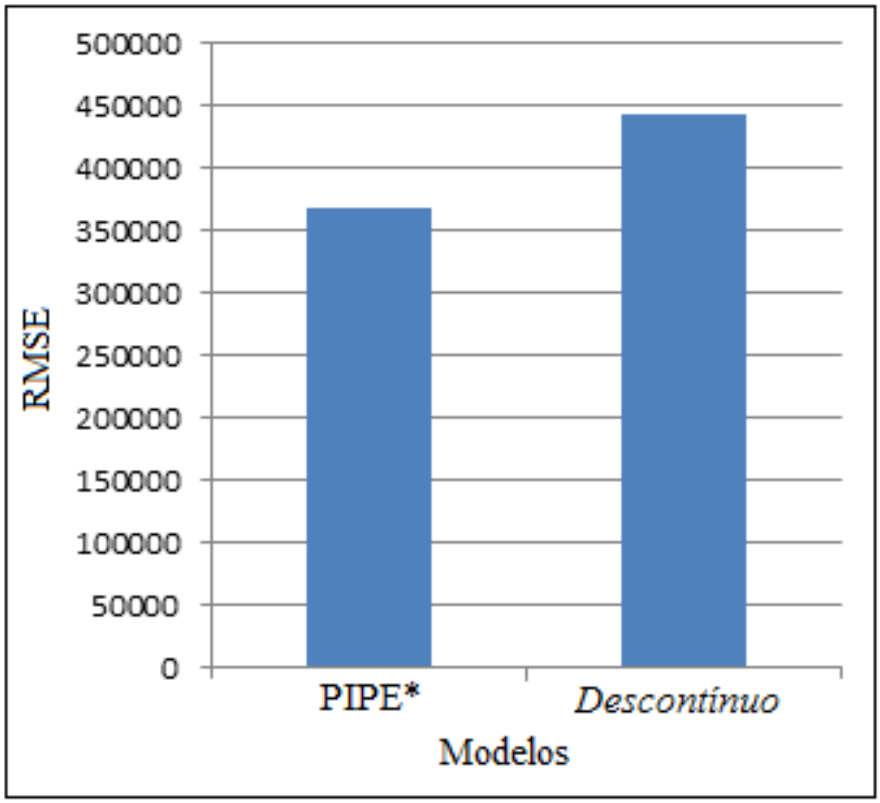
Figura 11: Acurácia das Soluções Incrementais.



Fonte: da autora.

A Figura 13 mostra a comparação entre as médias dos tempos de processamento, para atualização e busca nas árvores binárias. O resultado experimental indica um tempo maior para busca e computação da função da árvore binária mantida pela solução *Incremental Descontínua*, quando comparada à solução PIPE*. Enquanto a primeira usa, em média, 2.846 milissegundos para realizar a busca e predizer o tempo de viagem, a segunda gasta, em média, 1.105 milissegundos. A intuição é que o modelo descontínuo apresenta um pior resultado, quanto aos tempos de processamento, porque a árvore binária desse modelo é formada por um maior volume de dados, sendo ela mais profunda. Isso não ocorre com o modelo PIPE*, porque o

Figura 12: RMSE das Soluções Incrementais.



Fonte: da autora.

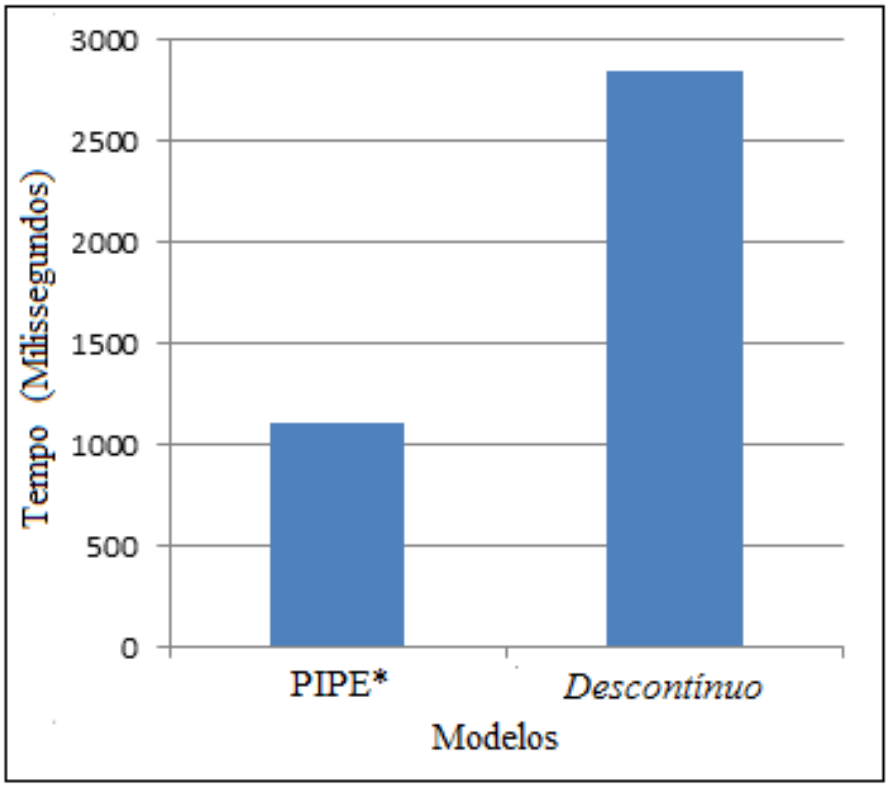
crescimento da árvore obedece a um critério de parada que é alcançado quando o desvio padrão atinge um valor menor ou igual à tolerância previamente definida (isto é, $\sigma \leq \delta$).

5.5 Visão geral das Avaliações Experimentais usando as estratégias PIPE e PIPE*

Para a análise do modelo PIPE foram apresentadas cinco questões principais, que seguem a metodologia descrita na Seção 5.2 e obedecem às métricas, previamente mostradas na Seção 5.3. Cada questão de pesquisa corresponde a um conjunto de experimentos associados a ela. Essas questões estão indicadas da seguinte forma:

1. **Questão I:** As mudanças realizadas na reengenharia do tráfego da cidade do Rio de Janeiro, devido às Olimpíadas de 2016, causaram mudanças consideráveis nos tempos de viagem dos ônibus?
2. **Questão II:** A variação da tolerância e da hora afeta os resultados relacionados com a acurácia do modelo PIPE*? É importante lembrar que o valor da tolerância (δ) é o único parâmetro usado no modelo PIPE* para construir e manter, de forma incremental, a árvore binária.
3. **Questão III:** Como a qualidade (acurácia) da estratégia PIPE, que computa a árvore

Figura 13: Tempos de atualização e busca nas árvores binárias.



Fonte: da autora.

binária *do zero*, difere da solução incremental, PIPE*?

4. **Questão IV:** Qual o ganho ou a perda quanto ao tempo de processamento da estratégia PIPE em relação à PIPE*?
5. **Questão V:** Qual o ganho ou a perda em relação ao tempo de processamento da solução PIPE* quando o recebimento de novos dados de trajetórias ocorre em diferentes intervalos de tempo? Além disso, esse diferente recebimento de novos fluxos contínuos de trajetórias altera o tempo de busca na árvore para construir a função e/ou se reflete, de alguma forma, na acurácia do modelo?

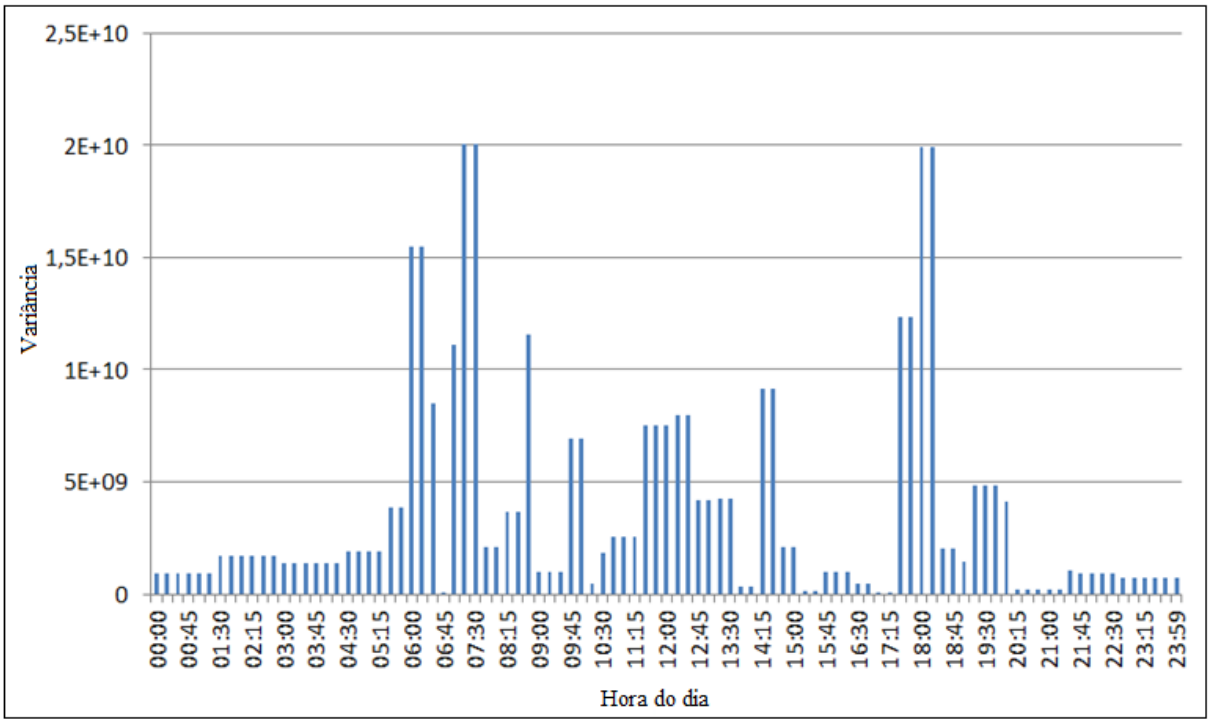
Questão I: Análise da variação dos tempos de viagens em diferentes anos.

Esse experimento avaliou se ocorreram mudanças no tráfego dos ônibus da cidade do Rio de Janeiro, comparando-se as informações antes e após o evento olímpico, que ocorreu entre 05 e 21 de agosto de 2016. A investigação buscou ainda verificar se as interferências na reengenharia da cidade aceleraram ou tornaram mais lentos os tempos de viagens dos ônibus.

Para essa análise experimental, foram considerados os segmentos de rua que compreendem as vias entre a Autoestrada Lagoa-Barra e a Avenida Ministro Ivan Lins, pertencentes

à Barra da Tijuca. Assim, para cada dia (isto é, 18 de agosto de 2015 e 16 de agosto de 2016 – ambos uma terça-feira), foram realizadas duas construções: a árvore binária e a função preditiva. Esses dados foram os mesmos discutidos na Seção 5.1. A função temporal foi computada a cada 15 minutos. A Figura 14 mostra a variância entre os tempos de viagem entre 18 de agosto de 2015 e 16 de agosto de 2016. Os dados de trajetória avaliados não tinham a mesma janela de tempo e, nesse caso, para alinha-los, foi usada a estratégia de interpolação. Para a análise ocorrer, apenas esses dois dias foram suficientes para descobrir que os tempos de viagem dos ônibus variaram nas vias analisadas.

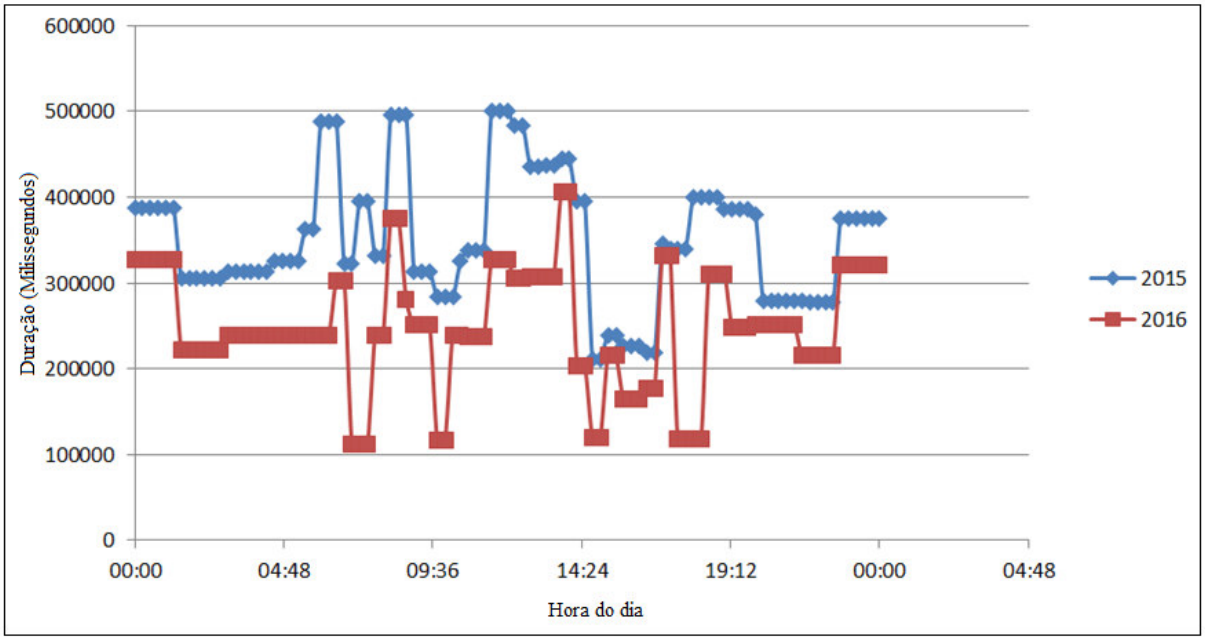
Figura 14: Variância entre os tempos de viagens referentes a Agosto de 2015 e Agosto de 2016.



Fonte: da autora.

Em resumo, houve uma alta variação entre os dias analisados, o que mostra a dinâmica dos tempos de travessia. A Figura 15 mostra a diferença entre os tempos de viagens de 18 de agosto de 2015 e 16 de agosto de 2016. É possível observar que houve uma melhora no tráfego dos ônibus no mês do evento olímpico, correspondente a agosto de 2016. Essa melhora pode ser explicada pelos investimentos realizados na reengenharia do trânsito, cujo objetivo foi melhorar os tempos em segmentos de ruas com difícil acesso. Dentre as melhorias existentes, é possível citar a construção de dois novos viadutos (RIO, 2017b) e a implementação do BRT Transbrasil (RIO, 2017a), outro tipo de transporte público implementado na cidade.

Figura 15: Tempos de viagens dos ônibus em 18 de Agosto de 2015 e 16 de Agosto de 2016.



Fonte: da autora.

Questões II e III: Análise da acurácia dos resultados.

Solução PIPE: Criação do modelo de árvore binária do zero para cada novo fluxo contínuo de trajetórias recebido.

Esta seção contém os resultados da avaliação experimental que buscou investigar como o limiar da tolerância (δ) pode afetar a qualidade da construção da árvore binária, obtida a partir da solução PIPE. É importante relembrar que o valor do parâmetro δ é definido pelo usuário, para possibilitar a construção e a manutenção da árvore binária. No entanto, é a partir do valor de δ que a profundidade da árvore binária é controlada. A intuição dessa análise é mostrar que, quanto maior for o valor de δ, haverá mais chances de a diferença ser maior entre o valor real do tempo de viagem e o valor predito pela função temporal.

Em cada nó não raiz da árvore binária existe um conjunto específico de dados de trajetórias. Para esse conjunto, o valor do desvio padrão é computado, e o resultado obtido é comparado com o valor de δ. Se o valor do desvio padrão for maior que δ, o nó não raiz deve ser dividido, originando dois outros nós. Dessa forma, δ também pode ser visto como um critério de parada para o aumento da profundidade da árvore. Para o propósito desse estudo, três diferentes valores para δ são analisados: 0.5, 1.0 e 1.5.

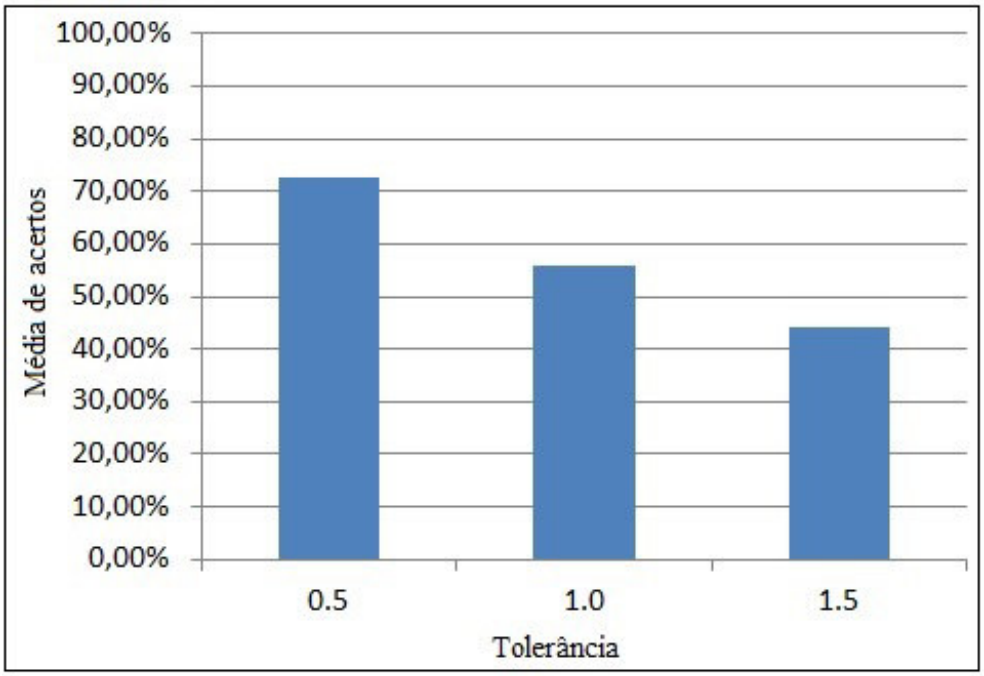
Para construir a árvore binária para solução PIPE, foram utilizadosos dados da trajetória dos ônibus que circularam nas estradas analisadas nas terças-feiras de agosto de 2015.

Para essa análise experimental, o volume de dados foi separado, randomicamente, em dois diferentes conjuntos: um com 70% dos dados para treino e outro com 30% dos dados para teste, cuja explicação dada na Seção 5.3.

Conforme foi discutido anteriormente, a construção do modelo envolve a construção da função temporal. Nesse experimento, a função temporal também é construída em intervalos de 15 minutos. A Figura 16 mostra a acurácia do modelo gerado para o conjunto de teste, quando variado o valor do parâmetro δ. A função temporal, obtida a partir da construção da árvore binária, acerta em média de:

- 73% do conjunto de teste, quando $\delta = 0.5$;
- 56% do conjunto de teste, quando $\delta = 1.0$;
- 44% do conjunto de teste, quando $\delta = 1.5$.

Figura 16: Porcentagem média de acertos variando a tolerância.



Fonte: da autora.

Por meio desses experimentos, é possível afirmar que, quanto menor a tolerância, melhor é a acurácia dos resultados. A intuição é a de que, quanto menor a tolerância, maior a profundidade da árvore e, consequentemente, maior será o número de nós não raiz criados. Vale lembrar que cada nó não raiz contém informações sobre um conjunto de dados de trajetória, e sempre que for aumentado o número de nós não raiz, será diminuído o conjunto de trajetórias descritas por cada nó. Dessa forma, o valor do desvio padrão também tenderá a diminuir.

Outra análise sobre a acurácia dos resultados relaciona-se com a investigação sobre

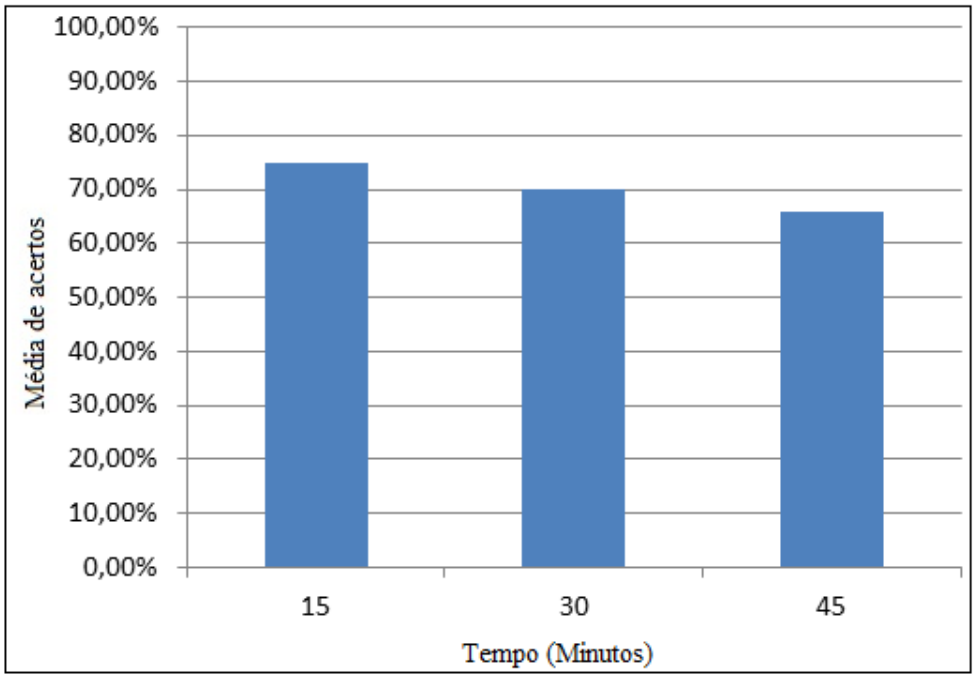
como a variação temporal pode afetar a qualidade da predição, obtida do processamento da árvore binária gerada pela solução PIPE. Sabendo-se que a hora do dia pode afetar os resultados obtidos a partir da função preditiva, foi realizado um conjunto de análises que consideraram diferentes variações temporais. A intuição é que, quanto maior for a variação temporal, maior será a diferença entre o valor real e o valor predito pela função. Dessa forma, o valor do intervalo de tempo para computar a função foi variado, com o intuito de analisar se essa alteração pode causar algum impacto na qualidade do modelo.

Para esse estudo, três diferentes variações temporais foram usadas para computar a função: *(i)* a cada 15 minutos; *(ii)* a cada 30 minutos; e *(iii)* a cada 45 minutos. Nessa análise, a árvore binária foi gerada a partir da solução PIPE, cujo processamento ocorreu considerando todos os dados de trajetória dos ônibus, os quais correspondem ao mês de agosto de 2015.

A Figura 17 mostra a acurácia do modelo gerado, quando a função é computada para cada valor de intervalo de tempo diferente. A função temporal prediz os valores corretos em média de:

- 75% do conjunto de teste, quando gerada a cada 15 minutos;
- 70% do conjunto de teste, quando gerada a cada 30 minutos;
- 66% do conjunto de teste, quando gerada a cada 45 minutos.

Figura 17: Porcentagem média de acertos variando o tempo.



Fonte: da autora.

Por meio desse experimento, é possível afirmar que, quanto menor o intervalo de

tempo, melhor será a acurácia dos resultados. A intuição por trás é que, quanto menor o intervalo de tempo para gerar a função, menor será o volume de dados analisado, o que reduz o desvio padrão entre as informações dos tempos de viagens.

Solução PIPE*: Manutenção incremental da árvore binária para cada novo fluxo contínuo de trajetórias recebido.

A avaliação experimental dissertada nesta seção busca investigar a acurácia da solução PIPE*, que permite atualizar, de forma incremental, uma árvore binária previamente criada, quando reportados novos fluxos contínuos de trajetórias. A ideia de ser incremental é permitir que essa árvore seja atualizada para representar a atual situação do tráfego, cujos valores são reportados continuamente, sem a necessidade de computar esta árvore *do zero*. Dentro desse contexto, é importante comparar a qualidade da função temporal obtida a partir da construção da árvore binária com a solução PIPE, em relação à PIPE*.

Inicialmente, a árvore binária com melhor acurácia foi obtida a partir das análises experimentais realizadas na seção 5.5, cujo valor da tolerância foi de 0.5. Esse valor de δ será mantido para a solução PIPE* nos experimentos desta seção. Ademais, a árvore binária foi construída a partir dos dados de todas as terças-feiras de agosto de 2015 e será atualizada a partir dos dados de todas as terças-feiras de setembro de 2015, que foram simulados como se chegassem no formato de fluxos contínuos. A função temporal para esses experimentos foi criada em intervalos de 15 minutos.

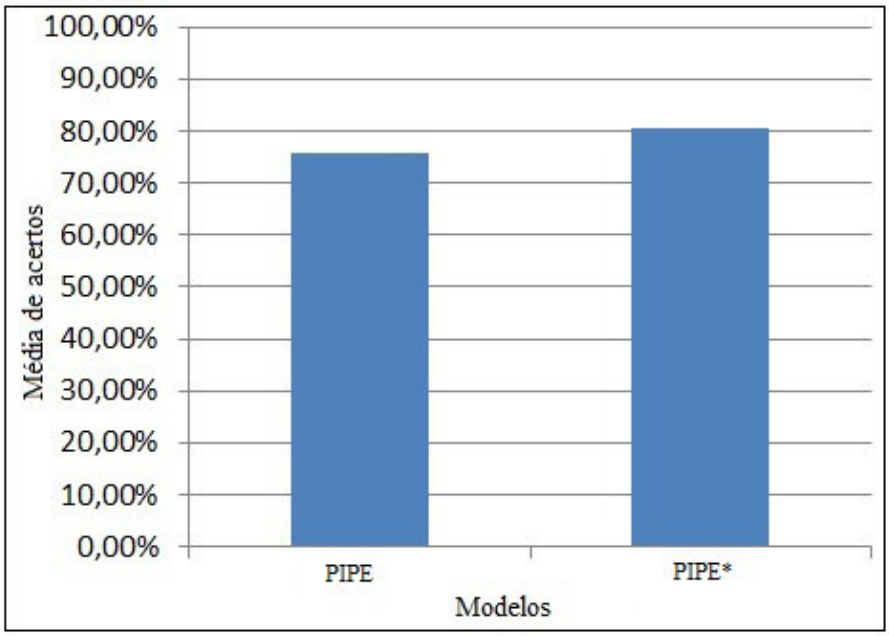
Sabendo que acidentes, obras públicas e outros eventos podem afetar a dinâmica da cidade e mudar os tempos de viagens, esse estudo se torna relevante porque busca avaliar a acurácia de um modelo que responde a tais mudanças. Nessa avaliação experimental, foi analisada como a predição do modelo gerado para as terças-feiras de agosto de 2015 pode sofrer alterações, devido às dinâmicas do comportamento do tráfego de setembro de 2015. Assim, a predição do modelo precisa ser alterada, dado o recebimento de novos fluxos contínuos de trajetória.

Existem duas soluções para analisar a consistência do modelo proposto nesta tese: *(i)* reconstruindo um novo modelo para as terças-feiras de setembro de 2015, conforme discutido na Seção 5.3, dividindo 70% desses dados para treinamento e 30% deles para testes; ou *(ii)* atualizando, de forma incremental, o modelo já existente, dado o recebimento dos dados de setembro de 2015. Nesse caso, a atualização ocorre a partir da árvore binária já construída para

agosto de 2015 – que foi construída com um volume de dados contendo em torno de 940 tuplas, sumarizando os dados de agosto e setembro, onde 690 tuplas foram usadas para atualizá-la, sendo essas referentes aos dados do tráfego de setembro de 2015. A fim de diminuir o escopo da análise, foram consideradas as informações contidas entre 12h00 e 23h59 (nos dois meses de análise, isto é, agosto e setembro). A acurácia dos resultados relacionados com as propostas *(i)* e *(ii)* é descrita a seguir.

A Figura 18 mostra o resultado da análise, indicando que a acurácia da função temporal para a solução PIPE* teve, em média, 81% de acertos para os dados de teste de setembro de 2015. A solução PIPE, que construiu a função temporal a partir da árvore binária – a qual foi construída *do zero*, com dados apenas de setembro de 2015 –, obteve em média a acurácia de 76% para os mesmos dados de testes. A intuição por trás é que a solução PIPE* é composta por um maior volume de informações para construir a árvore binária, uma vez que essa solução usou dados tanto de agosto de 2015, quanto de setembro de 2015. A solução PIPE foi construída usando-se apenas os dados de setembro de 2015, o que produz a criação de um modelo específico para dados de setembro.

Figura 18: Porcentagem média de acertos.



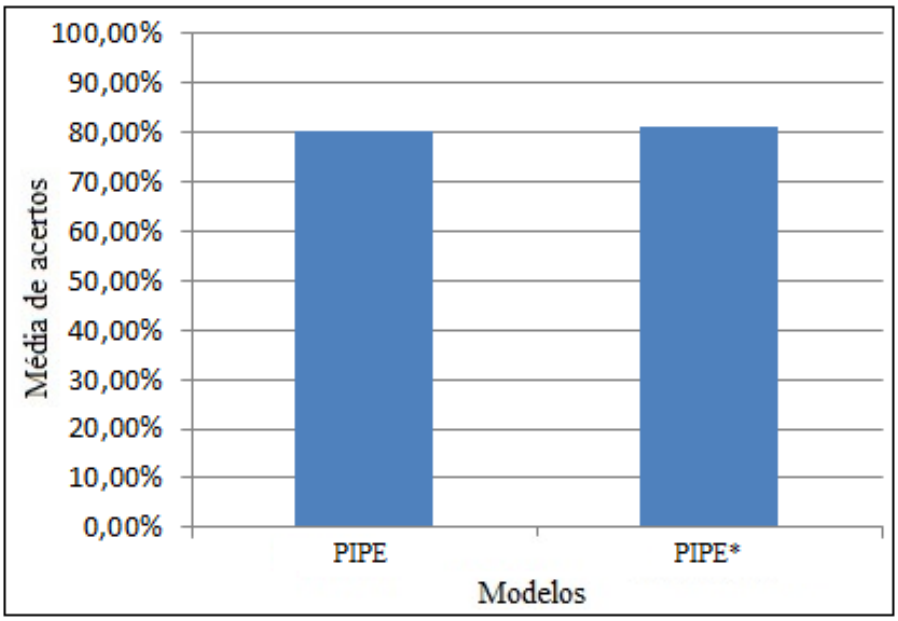
Fonte: da autora.

A solução PIPE* torna-se interessante, principalmente, porque os dados recebidos a cada instante de tempo não precisam persistir em uma base de dados. Além disso, o modelo PIPE* permite a atualização em nós específicos da árvore previamente criada. Essa atualização

se reflete na função temporal, porque será construída em tempo de execução, com base na busca da árvore atualizada. Nesse caso, apesar de a manutenção incremental ocorrer, a função se mantém contínua. O caminho para computar a árvore binária do modelo PIPE é diferente do usado pelo modelo PIPE*, porque a solução PIPE é criada a partir de todo volume de dados – os quais podem já existir em memória ou disco.

Um passo importante é identificar se o modelo PIPE, gerado a partir do mesmo volume de dados que a solução PIPE*, resulta em um modelo melhor ou igualmente bom em termos de acurácia. Nesse caso, dois modelos foram avaliados: *(i)* a construção da árvore binária *do zero*, com 70% dos dados de agosto de 2015 e 70% dos dados de setembro de 2015, onde a criação dessa árvore foi realizada a partir de todos os dados que estão disponíveis em memória ou disco – solução PIPE; e *(ii)* a construção da árvore binária a partir da manutenção incremental, com a sua geração ocorrendo a partir de 70% dos dados de agosto de 2015 e atualizada com 70% dos dados de setembro de 2015 – solução PIPE*. Assim, os dois modelos foram obtidos a partir do mesmo volume de dados, mas usando diferentes tipos de computação.

Figura 19: Porcentagem média de acertos usando o mesmo volume de dados.



Fonte: da autora.

Para comparar o número de acertos gerados pela função temporal, obtidos pelas soluções PIPE e PIPE*, apenas os dados de setembro de 2015 foram utilizados para teste. A acurácia das duas soluções praticamente coincidiu, como mostrado na Figura 19. Enquanto a solução PIPE teve, em média, 80% de acertos, a solução PIPE* teve 80.8%. Esses resultados

mostram que a solução PIPE* gera um modelo tão bom quanto a solução PIPE, se ambos forem criados com o mesmo volume de dados. Entretanto, a solução PIPE*, por ser incremental, tem mais vantagens, porque permite a atualização da árvore apenas quando necessário.

Questão IV: Análise da Eficiência.

Esta seção mostra resultados acerca da avaliação experimental realizada para verificar a eficiência das soluções. Dessa forma, os experimentos foram divididos em dois tipos de análise: *(i)* investigação do tempo de processamento, tanto construindo a árvore binária da solução PIPE, quanto atualizando-a de forma incremental, por meio da solução PIPE*; e *(ii)* investigação do tempo de busca na árvore para construir a função temporal. Nesse caso, foi comparado o tempo de busca nas árvores binárias e identificadas as diferenças entre os resultados obtidos.

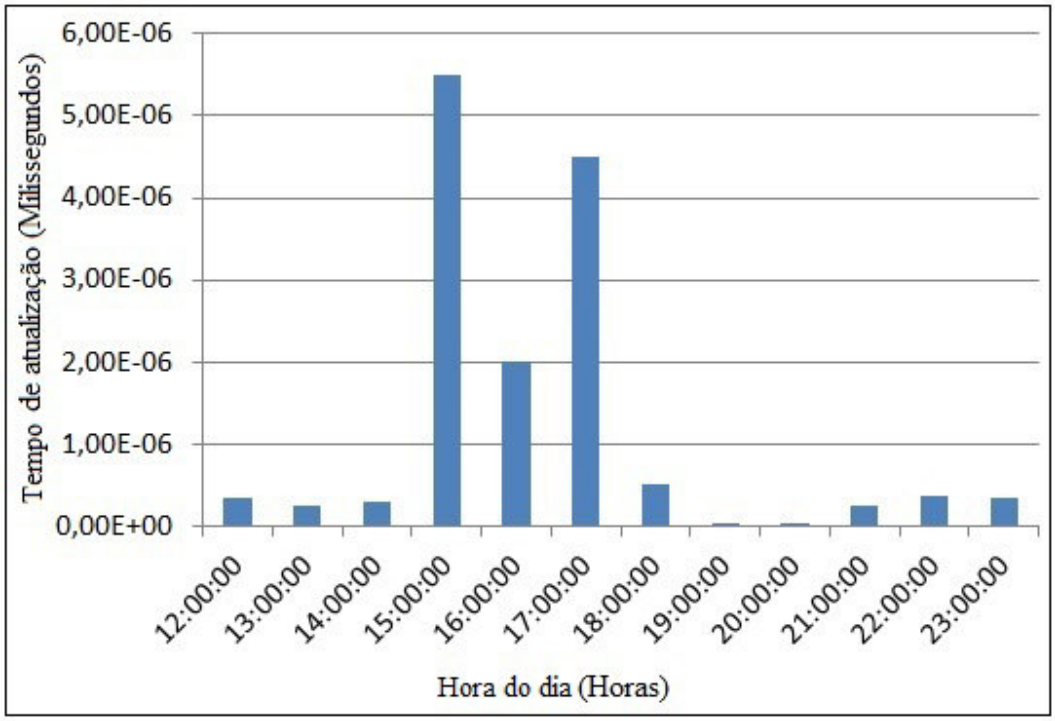
Análise do tempo de processamento para realizar a atualização incremental da árvore binária.

Num ambiente real, em que os fluxos contínuos de trajetórias são recebidos em intervalos de tempo, foi simulada uma análise para verificar o tempo de processamento da manutenção incremental. Dentro desse contexto, as árvores binárias foram construídas a partir dos dados dos ônibus que circularam nos segmentos de análise descritos na Seção 5.1, e na solução PIPE foram consideradas as terças-feiras de agosto de 2015. A manutenção incremental ocorreu a partir do recebimento dos dados do dia 08 de setembro de 2015, no período entre as 12h00min e as 3h59min.

A Figura 20 mostra o tempo de atualização da árvore binária, que já havia sido construída a partir dos dados das terças-feiras de agosto de 2015. Nesse resultado, é possível observar o tempo de processamento para atualização do modelo, quando são recebidos dados apenas do dia 08 de setembro de 2015. Suponha-se que a árvore binária é atualizada para cada hora do dia; nesse caso, nem todos os fluxos contínuos de trajetórias recebidos precisam atualizar a árvore. Isso ocorre porque o desvio padrão computado continua sendo menor que a tolerância previamente determinada pelo usuário. Nesse experimento, é possível observar que apenas os dados de trajetórias recebidos entre as 15h00 e as 18h00, de fato, modificam a árvore binária. Assim, apenas algumas horas do dia exigiram mais tempo de processamento para atualizar o modelo. O tempo total para atualização da árvore binária foi de 0.87 milissegundos. Por outro lado, o tempo total para a construção da árvore *do zero* demandou 284.2 milissegundos,

necessitando-se de um tempo de processamento maior para ser computado, quando esse é comparado com a manutenção incremental. Isso ocorre porque a execução do algoritmo PIPE, para toda janela de tempo recebida via fluxos contínuos de trajetórias, pode levar o modelo a um alto custo de processamento, devido ao recebimento de um grande volume de dados em pouco espaço de tempo.

Figura 20: Manutenção incremental da árvore por hora.



Fonte: da autora.

A solução PIPE* se apresenta como uma boa estratégia para dados reportados continuamente, porque a computação para atualizar o modelo apresenta um melhor desempenho, quando comparada com a solução obtida *do zero* – PIPE. Adicionalmente, a solução incremental – PIPE* – apresenta uma qualidade satisfatória, conforme mostrado na Seção anterior.

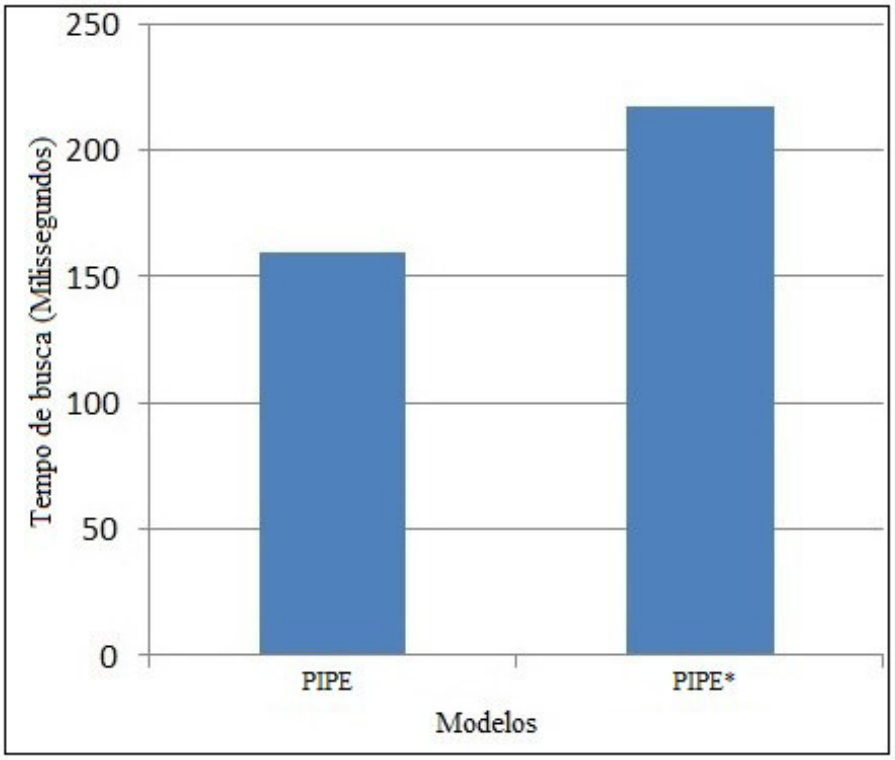
Análise o tempo de processamento de buscas realizadas na árvore binária a partir das criações PIPE e PIPE*.

Outra análise sobre a eficiência dos modelos relata o tempo de busca na árvore para construir a função temporal. O principal objetivo é analisar o tempo de processamento para gerar a função temporal, comparando o tempo de busca na solução PIPE com o tempo de busca na solução PIPE*. As duas árvores binárias já foram previamente criadas na Seção 5.5.

A Figura 21 mostra a comparação entre a média dos tempos de busca na árvore

binária, para encontrar o valor predito da função temporal. As funções foram construídas a cada 15 minutos e comparadas com os dados de teste, cuja análise seguiu a avaliação métrica descrita na Seção 5.3. O resultado experimental, mostrado na Figura 21, indica um tempo de busca maior para a árvore binária mantida com a solução PIPE*, quando comparada à construída a partir da solução PIPE. Enquanto a primeira usa, em média, 217.2 milissegundos para realizar a busca e predizer o tempo de viagem, a segunda gasta em média, apenas, 159.6 milissegundos. Isso ocorre porque o modelo gerado pela solução PIPE* apresenta uma árvore com maior profundidade, com um maior volume de dados, quando comparado à árvore criada a partir da solução PIPE. A intuição por trás é que, quanto maior a profundidade da árvore, maior o tempo de busca para computar o tempo de viagem, obtido a partir da função temporal.

Figura 21: Comparação dos tempos de busca nas árvores.



Fonte: da autora.

Questão V: Análise do tempo para atualização da árvore dado o recebimento de fluxos contínuos de trajetórias.

Nesta seção, são apresentados os resultados quando se varia a janela de tempo para avaliar a solução PIPE*, dado o recebimento de novos fluxos contínuos de trajetórias. A exploração de fluxos contínuos de trajetórias, em diferentes intervalos de tempo, pode prover

uma melhor compreensão sobre o comportamento desses dados e um maior entendimento sobre os valores preditos, gerados a partir da função temporal. Nessa análise experimental, foram utilizados dados de trajetórias dos ônibus do Rio de Janeiro, referentes às terças-feiras de agosto de 2015, os quais foram descritos na Seção 5.1. Para o propósito deste estudo, foram utilizados três janelas de tempo diferentes para o recebimento de fluxos contínuos de trajetórias: 5, 10 e 15 minutos. Nesse experimento, o conjunto de dados foi dividido randomicamente em 70% para treino do modelo e 30% para teste, como explicado na Seção 5.3. Além disso, o valor de $\delta = 0.5$ foi considerado para a construção e a manutenção da árvore, porque essa tolerância apresentou melhor qualidade, dados os resultados mostrados na Seção 5.5. Ademais, a função foi construída por segmento e leva em consideração o parâmetro da hora do dia, cujo intervalos foi de 15 minutos.

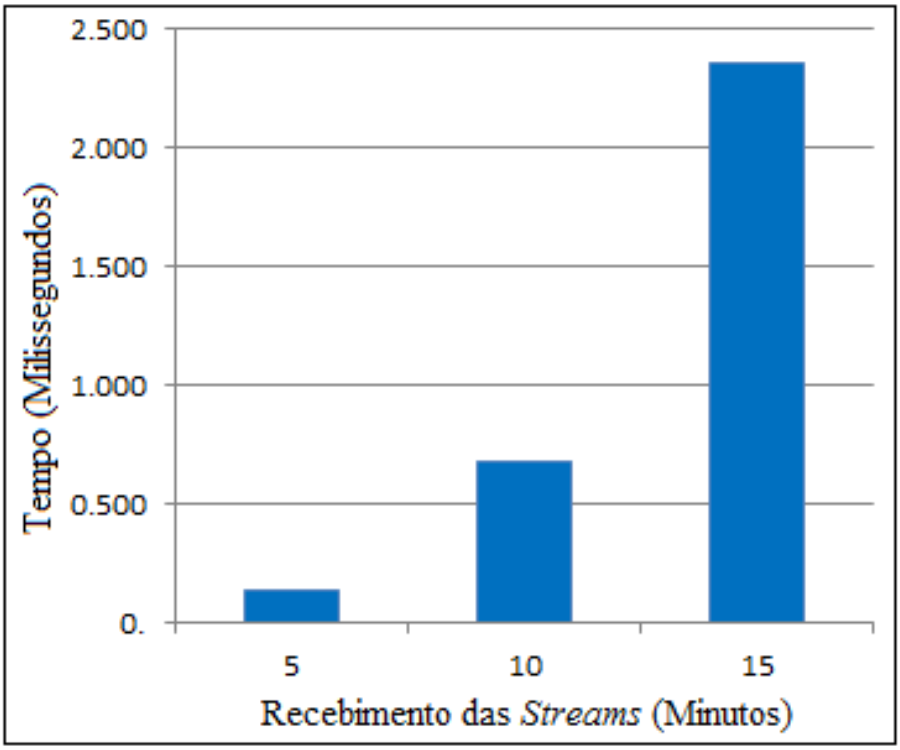
A Figura 22 reporta o tempo de atualização da árvore, dado o recebimento de fluxos contínuos de trajetórias (para os dados analisados nos experimentos desta Seção). Os resultados sobre tempo de processamento mostram uma diferença temporal pequena, definida em milisegundos. No entanto, o ganho em processamento esperado é quando se usa a solução em larga escala. Os resultados da análise, mostrados na Figura 22, mostram que o tempo médio de atualização do modelo tem melhor desempenho quando dados reportados continuamente estão contidos em um menor intervalo de tempo, onde:

- Para fluxos contínuos de trajetórias, que são recebidos a cada 5 minutos, a atualização média da árvore binária é de 0.135 milissegundos;
- Quando os dados são reportados de 10 em 10 minutos, a média de atualização é de 0.680 milissegundos;
- A cada 15 minutos, a média de atualização da árvore é de 2.362 milissegundos.

A intuição por trás desses resultados depende da distribuição dos dados, que neste caso é uniforme. Nessa avaliação experimental, é possível observar que quanto menor o intervalo de tempo para recebimento de novos fluxos contínuos de trajetórias, menor será o volume de dados e, assim, menor também será o tempo de processamento para atualização do modelo. Porém, se mesmo com a janela de tempo sendo modificada a cada 15 minutos, todos os dados chegassem nos primeiros 5 minutos, essa diferença não seria sentida.

Outra análise realizada acerca da acurácia dos resultados está relacionada com a investigação sobre como a variação temporal, dado o recebimento de fluxos contínuos de trajetórias, pode impactar a qualidade da predição. A Figura 23 mostra o tempo de busca

Figura 22: Comparação dos tempos de atualização da árvore.



Fonte: da autora.

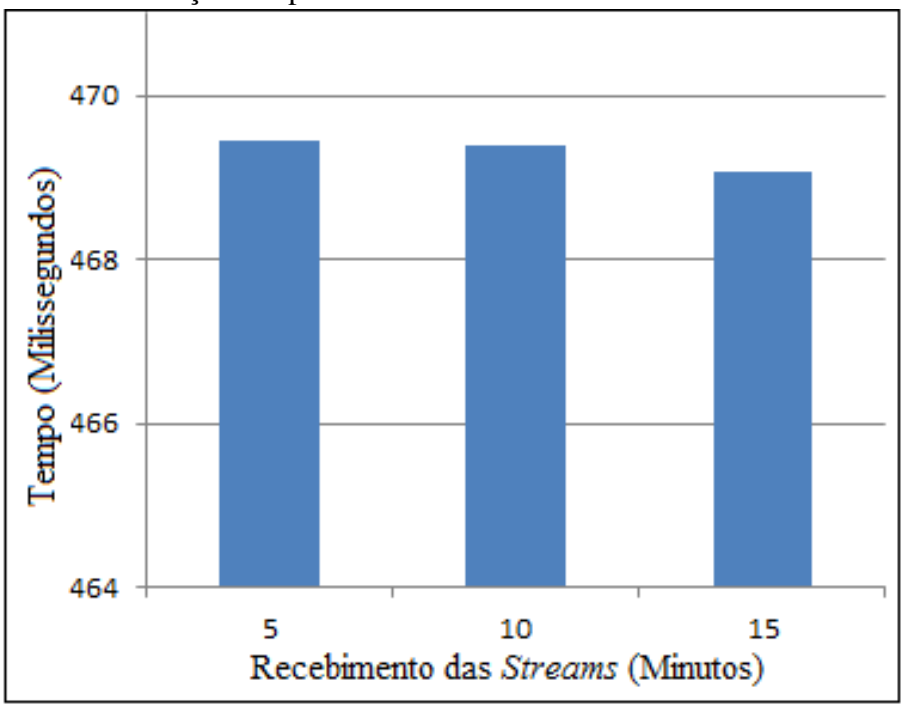
na árvore binária, para construir a função preditiva. Considerando os diferentes tempos para recebimento de novas informações, é possível verificar que o tempo de busca para construir a função temporal se manteve em torno de 469 milissegundos.

A Figura 24 mostra a acurácia do modelo gerado. Nesse resultado, a função preditiva acertou em praticamente 81% do conjunto de testes em todos os intervalos de tempo. Por meio desse experimento, é possível afirmar que, apesar de o recebimento de fluxos contínuos de trajetórias ocorrer em intervalos de tempo diferentes, o tempo para computar a função temporal foi mantido, e a acurácia da solução, também. Isso é justificado porque, no final de todo recebimento do conjunto de trajetórias, o volume dos dados usados para atualizar a árvore binária é o mesmo para todas as execuções.

5.6 Avaliação Experimental da Solução PIPE* em Tempo Real

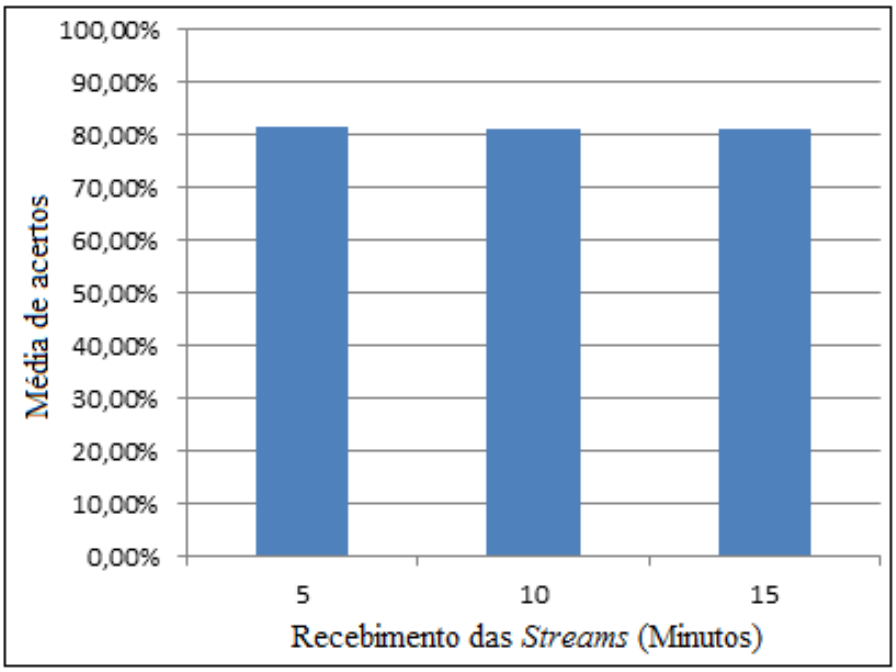
Esta seção mostra os resultados da análise experimental realizada para a solução PIPE*, a qual realiza a manutenção incremental da árvore binária. Os fluxos contínuos de trajetórias foram simulados chegando em consecutivas janelas de tempo, a cada 5, 10 e 15 minutos, e a variação no tamanho das janelas de tempo ocorreu sem sobreposição para todos os experimentos.

Figura 23: Comparação dos tempos de busca na árvore para construir a função temporal.



Fonte: da autora.

Figura 24: Acurácia do modelo dado o recebimento de Fluxos Contínuos de Trajetórias.



Fonte: da autora.

A Figura 25 mostra um exemplo de como foi simulado o recebimento dos novos dados de trajetórias a cada 5 minutos. Imagine que cada linha preta, mostrada na Figura 25, corresponde a uma trajetória, e os retângulos em azul correspondem às janelas de tempo (isto

é ($TW_1, TW_2, ..., TW_n$), as quais são compostos por fluxos contínuos de trajetórias. Os valores recebidos em cada TW_i podem atualizar o modelo de forma incremental, caso o desvio padrão entre os dados de análise seja maior que a tolerância previamente definida.

No exemplo, é mostrado um conjunto de trajetórias correspondentes à Avenida Santos Dumont, localizada em Fortaleza/Ceará. Note que a janela de tempo tem tamanho de 5 minutos e começa a receber dados às 17h00min. Dessa forma, é possível verificar que, se uma dada amostra for recebida no intervalo de tempo entre 17h00min e 17h04min59seg, o próximo fluxo de dados recebido refere-se ao intervalo entre 17h05min e 17h09min59seg, não existindo sobreposição entre janelas.

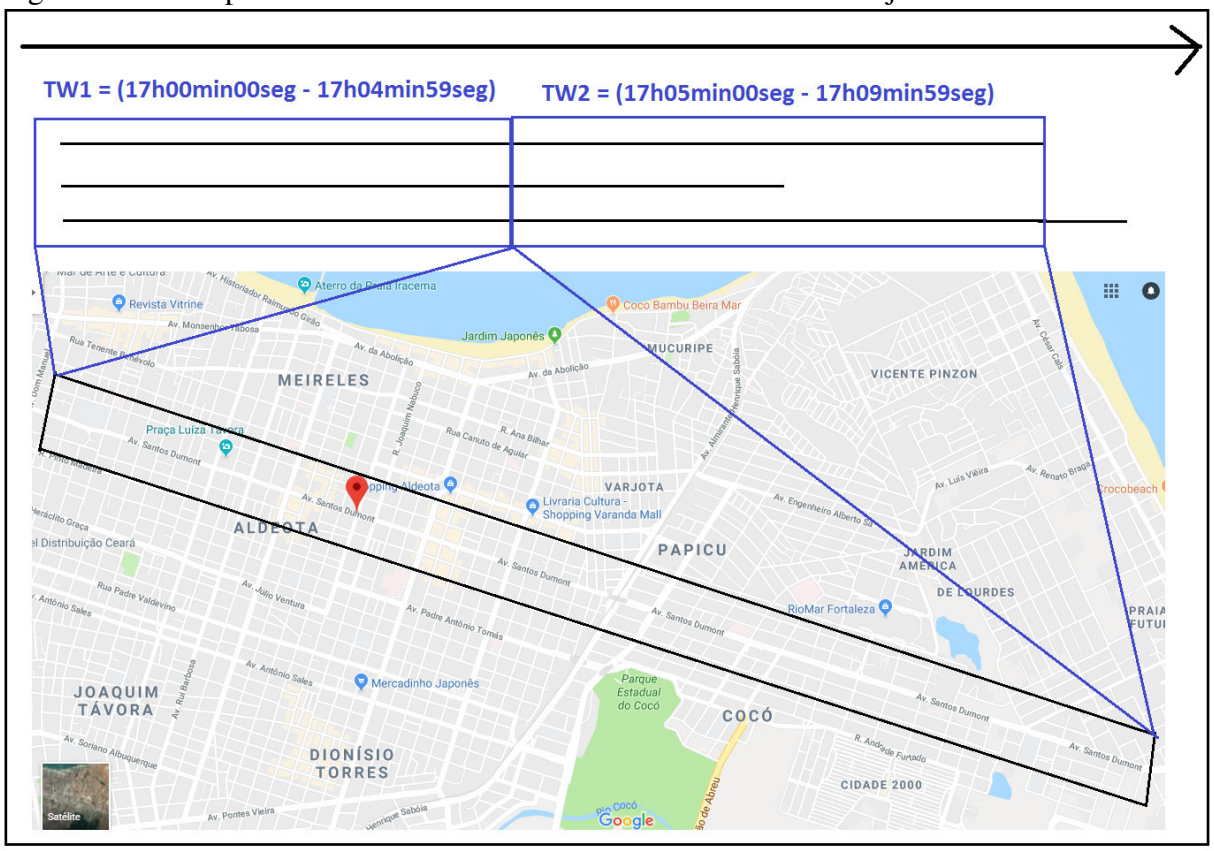
Os dados de trajetória utilizados nesta análise experimental são dos táxis de Fortaleza/Ceará, pertencentes à Empresa Táxi Simples (SIMPLES, 2015), os quais são gerados a partir de um dispositivo de GPS. Esses dados de análise correspondem aos táxis que trafegaram na Avenida Santos Dumont entre os dias 16 de novembro de 2015 a 31 de dezembro de 2015. Dentro desse contexto, o modelo PIPE* cria e atualiza a árvore binária em tempo de execução, para possibilitar a construção da função preditiva para esse trecho específico da cidade. Na análise realizada, foi possível observar que o modelo PIPE* admite um processamento distribuído, caso seja utilizado, por exemplo, um cluster de máquinas, onde cada host possibilitaria processar os dados referentes a um segmento de rua específico da cidade, construindo ou mantendo o modelo a partir do recebimento de fluxos contínuos de trajetórias.

Foi observado que o modelo sofre bastante atualização no intervalo de tempo entre as 17h00min e as 22h00min, quando considerada a Avenida Santos Dumont entre os dias 16 de novembro de 2015 a 31 de dezembro de 2015. Dessa forma, o conjunto de dados pertencente a esse intervalo foi considerado como objeto de análise.

Os resultados mostrados nesta Seção foram obtidos a partir da execução de um conjunto de processos, como mostrado na Figura 26. O Algoritmo *Storm* (STORM, 2016) foi utilizado para simular, continuamente, a chegada de novos dados de trajetória. Enquanto, o Algoritmo *Barefoot* (BAREFOOT, 2017) foi utilizado para realizar o *Map Matching* dos dados reportados continuamente. Após o mapeamento das trajetórias na rede de ruas, a solução PIPE* permite a criação e a manutenção incremental da árvore binária, a qual é utilizada para construir a função preditiva para o segmento de análise. Esta função preditiva recebe de entrada um instante de tempo t.

Ao variar o tamanho das janelas de tempo, variou-se a quantidade de fluxos de

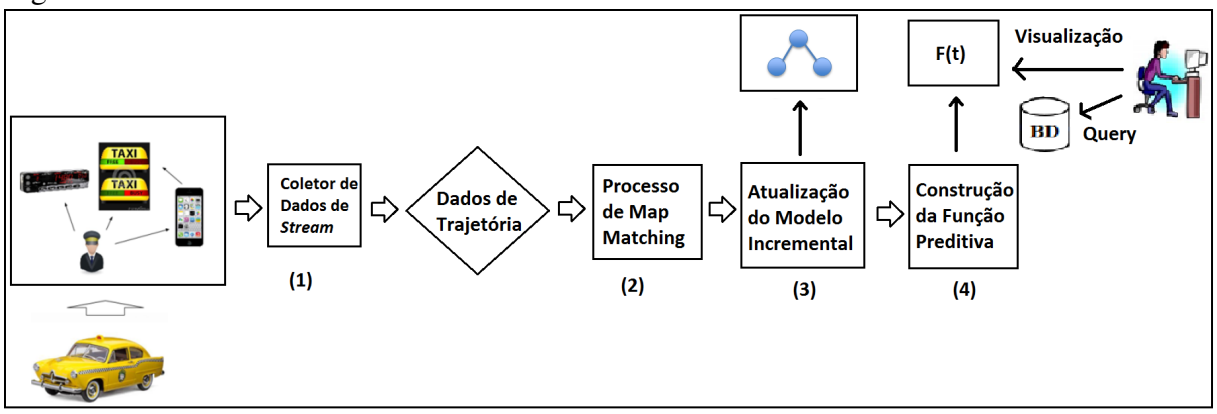
Figura 25: Exemplo do recebimento de novos fluxos contínuos de trajetórias a cada 5 minutos.



Fonte: da autora.

trajetórias a serem processados pela solução proposta. Nessa experimentação foi assumido que os dados têm distribuição uniforme no tempo e os resultados mostraram que, mesmo aumentando o tamanho da janela e processando mais fluxos de dados, não houve perda de eficiência na estratégia proposta. Dessa forma, a solução PIPE*, que permite a manutenção incremental, é computacionalmente viável para construir modelos utilizados em larga escala; além disso, os processos podem ser paralelizados e distribuídos (conforme explicado anteriormente).

Figura 26: Modelo do Processo *Online*.



Fonte: da autora.

A metodologia usada nos experimentos aqui realizados também foi baseada no Algoritmo *k-fold*, cujo valor de *k* foi definido como 10. O volume de dados foi dividido em 70%, para realizar o treinamento, e em 30% para realização de testes. Essa divisão ocorreu para as três janelas de tempo estudadas (isto é, 5, 10 e 15 minutos), resultando em 30 diferentes análises.

Todos os experimentos também foram conduzidos em máquinas Intel Core 2, com CPU Q6600 server, 8GB de RAM e 2.40GHz, usando o sistema operacional Ubuntu 11.10, de 64 bits. As linguagens de programação usadas para implementar as soluções também foram Java e R.

Os resultados das análises mostram as investigações acerca da acurácia e eficiência da solução PIPE*, quando variadas as janelas de tempo. As manutenções incrementais do modelo permitem atualizar a árvore binária de tal modo, que um nó existente pode se transformar em dois novos nós, caso o desvio padrão dos novos fluxos contínuos de trajetórias recebidos seja maior que a tolerância (δ). Essa atualização é importante porque reflete a atual situação do tráfego, a qual é relatada pelos novos dados.

Nos experimentos anteriores, a tolerância (δ) foi definida como um parâmetro a ser informado pelo usuário. No entanto, nesses experimentos, a tolerância foi computada a partir da combinação linear, que visa balancear a importância do passado e do presente, dados os valores relacionados ao conjunto de dados de análise.Para realizar essa análise, foi necessário computar o desvio padrão, para cada janela de tempo recebida e computar tempo estimado da viagem, dado o valor predito, obtido a partir do modelo proposto.

Além dos resultados obtidos acerca da acurácia da solução, foram realizadas análises de desempenho, considerando o tempo de processamento do *map matching*, quando realizado em tempo real; o tempo de processamento para realizar a manutenção incremental – execução da solução PIPE*; e o tempo para construção da função preditiva.

Dessa forma, duas questões, definidas a seguir, guiaram a análise experimental desta Seção:

1. **Questão I:** Ao variar o tamanho da janela de tempo, como varia a acurácia da solução PIPE*?
2. **Questão II:** Ao variar o tamanho da janela de tempo, como varia o tempo do processamento da solução PIPE*?

Questão I: Acurácia da Solução PIPE* ao variar o tamanho das Janelas de Tempo.

Esta Seção contém os resultados da avaliação experimental, que buscou investigar como a variação das janelas de tempo (TW) dos dados reportados como fluxos contínuos de trajetórias pode afetar a qualidade do resultado obtido a partir da função preditiva, construída dado o processamento da solução PIPE*. É importante relembrar que o valor da tolerância (δ) foi computado a partir de uma combinação linear, e esse resultado é essencial para determinar a construção e a manutenção da árvore binária.

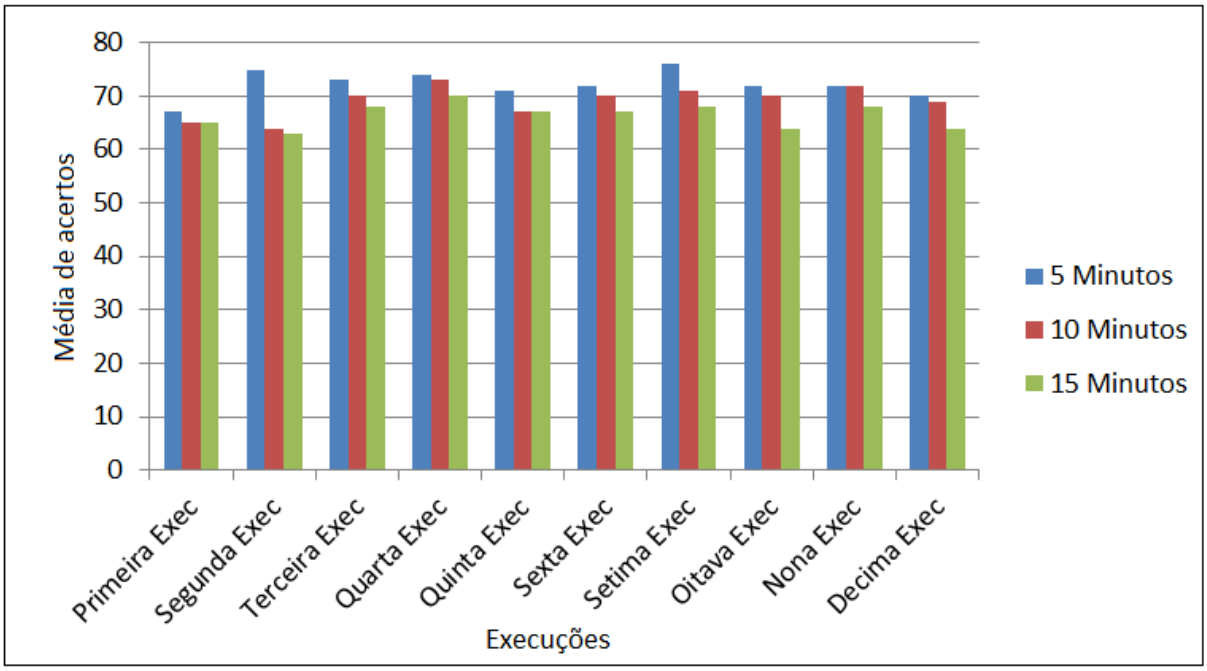
Em cada nó não raiz da árvore binária, existe a tupla (μ_i, σ_i, $[t_i, t_f]$), que corresponde aos valores da média, desvio padrão e intervalos de tempo. Como visto no Capítulo 4, cada valor dessa tupla é computado a partir de conjunto de dados, os quais podem ser reportados continuamente. Cada nó, pertencente à árvore binária, pode ser alterado e se transformar em dois novos nós, se os novos dados recebidos tiverem o valor do desvio padrão maior que a tolerância (δ). Dessa forma, sempre que um novo conjunto de dados é recebido, o valor do desvio padrão é computado, e o resultado obtido é comparado com o valor de δ. Se o valor do desvio padrão for maior que δ, o nó não raiz deverá ser dividido, originando dois outros nós.

O volume de dados utilizado nessa avaliação experimental foi, também, separado em dois diferentes conjuntos – de treino e teste –, que foram obtidos de forma randômica, sobre os quais se discorreu na Seção 5.6. Nesse experimento, a função temporal também é construída dada a janela de tempo de análise, ou seja, dadas as mudanças que ocorreram em intervalos de 5, 10 ou 15 minutos, totalizando 30 execuções, como mostrado na Figura 27. Nas análises realizadas, o modelo proposto computa um valor predito para a função temporal. Para as avaliações realizadas neste conjunto de experimentos, foi assumido que o tempo estimado da duração de viagem coincide quando a diferença entre o valor predito e o valor real é de até 2 minutos, cujo limite aceitável porque o tempo de viagem no conjunto de dados de teste variou em até 13 minutos entre durações de viagens observadas, que coincidiam no mesmo instante de tempo.

A atualização do modelo ocorre a partir da árvore binária já construída, que contém um volume de dados em torno de 285 tuplas, onde 85 tuplas foram usadas para testar o modelo, e as demais, para treiná-lo. Foram considerados os objetos móveis que tinham dois ou mais pontos de localização por segmento observado, os demais, que não estavam contidos nessa restrição, foram descartados. A Figura 27 mostra a acurácia computada a partir do modelo incremental, quando variadas as janelas de tempo (TW). A função temporal, obtida a partir da construção da árvore binária, acerta em média:

- 84% do conjunto de teste, quando TW = 5 minutos;
- 81% do conjunto de teste, quando TW = 10 minutos;
- 78% do conjunto de teste, quando TW = 15 minutos.

Figura 27: Média de acertos da função, quando variada a janela de tempo.



Fonte: da autora.

Por meio desses experimentos, é possível afirmar que a quantidade de acertos da função, quando variadas as janelas de tempo e dado o recebimento de fluxos contínuos de trajetórias, é maior para a janela de tempo correspondente ao menor volume de dados, isto é quando TW = 5 minutos. A intuição por trás é que, quanto menor a janela de tempo, a qual reporta novos dados de trajetórias, menor o volume de dados e, consequentemente, menor o desvio padrão entre os tempos de viagens, considerando, nesse caso, a distribuição uniforme dos dados.

Questão II: Tempo do Processamento da Solução PIPE* ao variar o tamanho das Janelas de Tempo.

Esta Seção mostra os resultados acerca da eficiência do modelo, quando variadas as janelas de tempo para o recebimento de novas trajetórias, as quais causam atualizações na árvore binária. Os experimentos realizados foram divididos em dois tipos de análises: *(i)* a investigação do tempo de processamento, tanto realizando o *map matching*, quanto atualizando a árvore binária; e *(ii)* a investigação do tempo para construir a função temporal, dada a profundidade da

árvore.

Análise do tempo de processamento para realizar o map matching em tempo real.

Simulando um ambiente real, em que novos dados de trajetórias são recebidos continuamente a partir de janelas de tempo previamente definidas, foi realizada uma análise quanto ao tempo de processamento para realizar o ajuste das trajetórias. Dentro desse contexto, os dados de análise foram obtidos em três diferentes intervalos de tempo (isto é, a cada 5, 10 e 15 minutos), os quais trafegaram na Avenida Santos Dumont, em Fortaleza/CE, entre as 17h00min e as 22h00min.

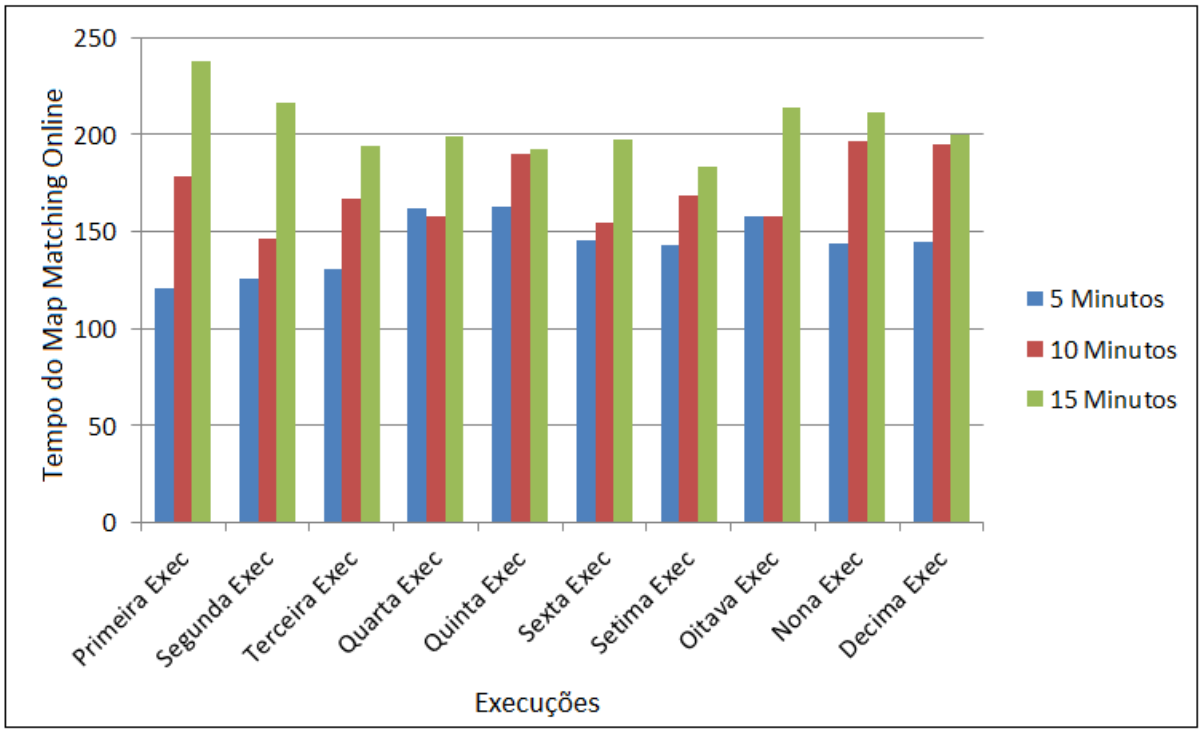
O processamento do *map matching* em tempo real, para diferentes janelas de tempo de análise, e o tempo demandado por ele, para realizar os ajustes necessários nos dados de trajetória, fazem parte do processo para construção e análise do modelo PIPE*. O *map matching* desta análise ocorre antes da atualização da árvore binária e é responsável por disponibilizar a correta localização do objeto na rede de ruas, a partir de uma coordenada geográfica *(latitude, longitude)* e de uma informação temporal *(data, hora)*. Além da correção dos dados, recebidos como fluxos contínuos de trajetórias, é necessário saber se todo o processamento do modelo, desde o recebimento e mapeamento dos dados até a construção da função preditiva, ocorre antes de receber um novo conjunto de informações. Ou seja, se esse tempo para a obtenção do resultado final é menor que o tamanho da janela de tempo.

A Figura 28 mostra os resultados obtidos, quando variadas as janelas de tempo. A análise foi realizada em dez execuções, para cada janela de tempo. Em média, as atualizações para as janelas de tempo iguais a 5 minutos ocorreram em até 143 milissegundos. O *map matching* para janelas de tempo iguais a 10 minutos ocorreu em até 171 milissegundos; e considerando o volume de dados maior, correspondente às janelas de tempo iguais a 15 minutos, o *map matching* foi realizado em até 204 milissegundos. A intuição por trás é que, quanto menor a janela de tempo, menor o volume de dados para a realização do ajuste e, assim, menor o tempo de processamento do *map matching*.

Análise do Tempo de Processamento da Solução PIPE*.

Nesta Seção, a avaliação experimental busca investigar o tempo de processamento para atualizar a árvore binária, construída a partir da solução PIPE*, a qual é criada desde o primeiro recebimento do conjunto de dados. Posteriormente, é preciso mantê-la de forma

Figura 28: Tempo do Processamento do *Map Matching*, variando as Janelas de Tempo.



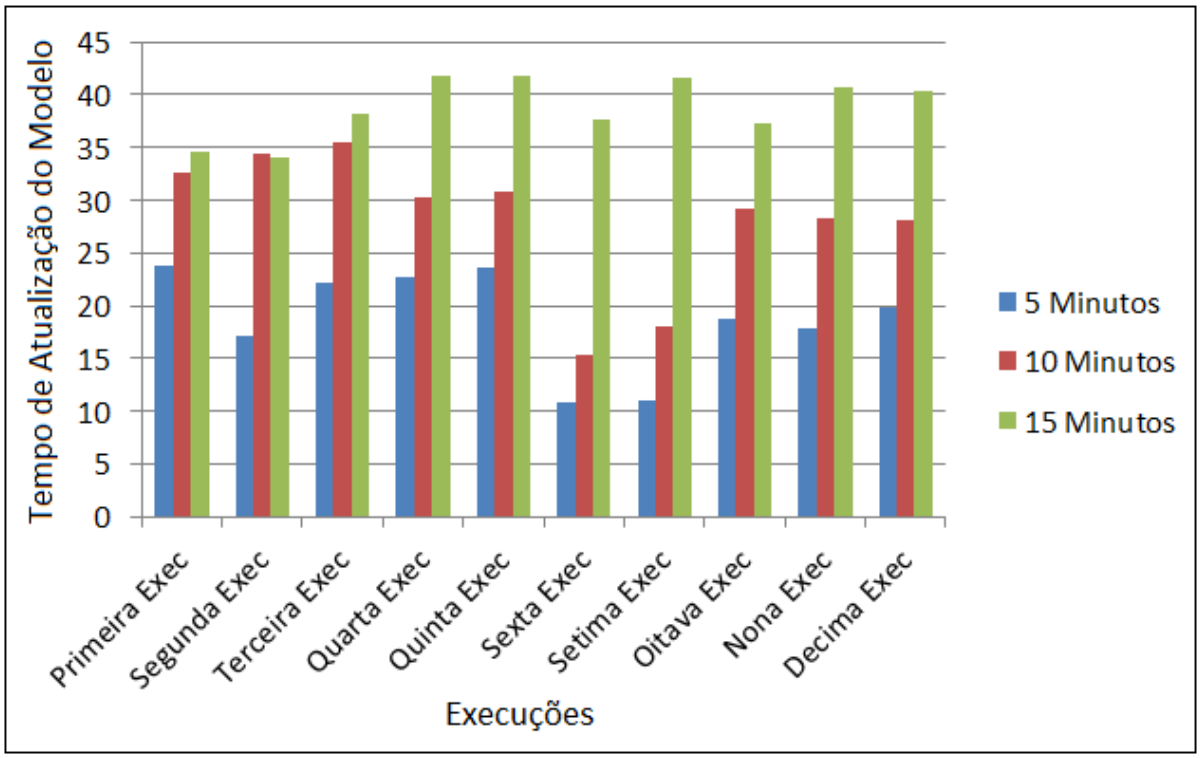
Fonte: da autora.

incremental, dado o recebimento de novos fluxos contínuos de trajetórias.

Foi simulado o recebimento de conjuntos de dados a partir de três diferentes janelas de tempo, os quais foram utilizados para realizar a manutenção incremental – esses dados foram os mesmos descritos na Seção 5.6. O recebimento de cada janela de tempo, para possível atualização, ocorreu entre às 17h00min e às 22h00min.

A Figura 29 mostra os resultados das atualizações para cada janela de tempo, indicando o tempo de atualização em cada uma das dez execuções, seguindo as estratégias de utilização do Algoritmo $k - fold$, conforme descrito na Seção 5.6. Dos resultados obtidos, o tempo total médio para atualização de todos os registros, que modificaram incrementalmente a árvore binária, considerando a janela de tempo igual a 5 minutos, foi de cerca de 18 milissegundos. Já a atualização da árvore, para um volume de dados maior, correspondente à janela de tempo igual a 10 minutos, demandou um tempo médio de até 28 milissegundos; e quando a janela de tempo variou em até 15 minutos, o tempo total médio de processamento subiu para cerca de 38 milissegundos. Com os resultados obtidos, é possível identificar que, quanto maior a janela de tempo, e consequentemente maior o volume de dados, maior também será o custo de processamento para atualização do modelo.

Figura 29: Eficiência da solução PIPE*, dada a variação das Janelas de Tempo.



Fonte: da autora.

Análise do tempo de processamento para realizar a construção da função preditiva.

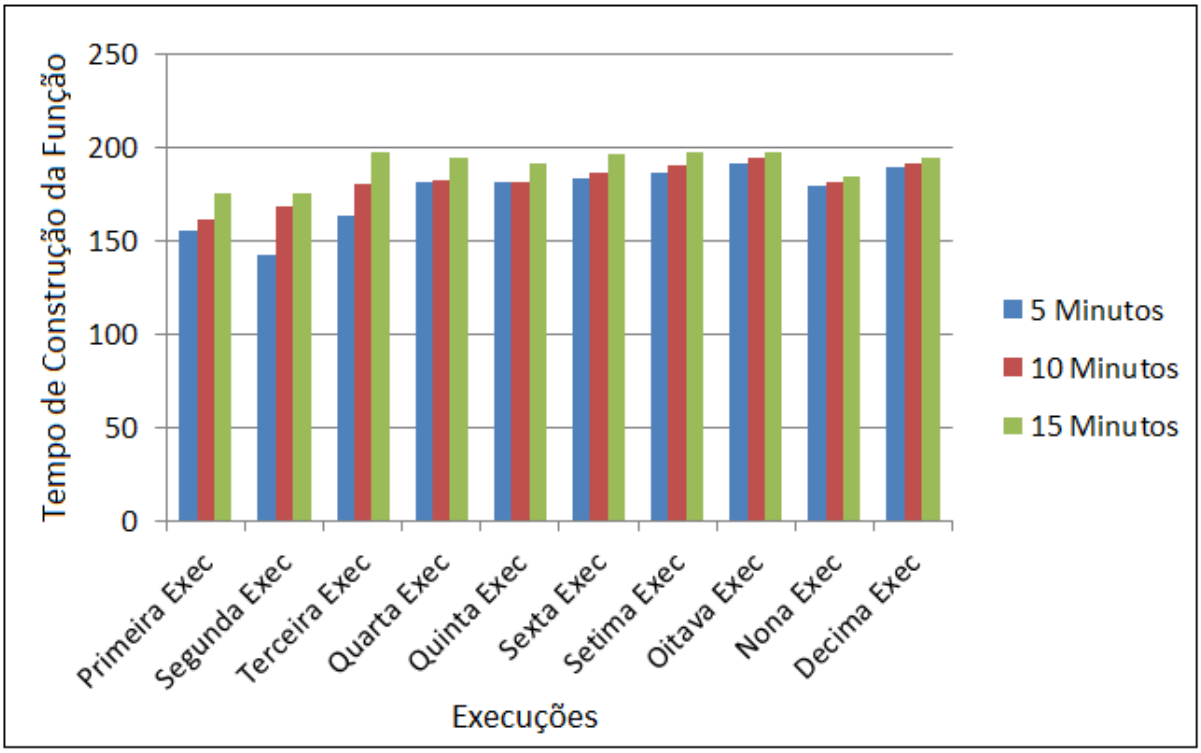
O modelo incremental proposto nesta tese e mostrado no Capítulo 4 é formado por um conjunto de processos, cujo objetivo é permitir a construção de uma árvore binária para um segmento de rua específico. Essa árvore deve permitir a criação de uma função preditiva, dado um instante de tempo t. Sabendo que o modelo é considerado incremental, porque pode ser atualizado a cada novo conjunto de dados recebidos continuamente, é importante investigar se essa atualização causa impactos no tempo de processamento da construção da função.

A avaliação experimental realizada nesta Seção busca investigar de que forma as mudanças nos tamanhos das janelas de tempo causam impacto no tempo de processamento do modelo. Os dados de trajetória reportados continuamente podem viabilizar a manutenção incremental do modelo, aumentando a quantidade de nós que compõem a árvore binária. Assim, um nó da árvore pode se transformar em dois novos nós, caso o desvio padrão dos objetos recebidos seja maior que a tolerância previamente computada.

Novamente, foi simulado o recebimento das três diferentes janelas de tempo para realizar a manutenção incremental, cujos dados foram os mesmos descritos na Seção 5.6 e cujo recebimento de cada janela de tempo também ocorreu entre às 17h00min e às 22h00min. A

Figura 30 mostra os resultados dos tempos de processamento em cada uma das dez execuções, por janela de tempo. Dos resultados obtidos, o tempo total médio para a construção da função a partir da busca realizada na árvore binária, considerando a janela de tempo igual a 5 minutos, foi em média de 175 milissegundos. Já o tempo de busca na árvore, para um volume de dados maior, correspondente ao conjunto de dados recebido a cada 10 minutos, demandou um tempo total médio de processamento de até 181 milissegundos; e quando a janela de tempo variou em até 15 minutos, o tempo total médio de busca subiu para cerca de 190 milissegundos. Apesar da pouca diferença entre os resultados, é preciso atentar para possíveis análises realizadas em larga escala, cujo aumento do volume de dados pode gerar árvores binárias mais profundas, aumentando o tempo de processamento para busca e, consequentemente, o tempo para construção da função, o que resulta numa diferença ainda maior sobre os resultados.

Figura 30: Medição da eficiência para construção da função preditiva, dada a variação das Janelas de Tempo.



Fonte: da autora.

Análise da eficiência para construir a função preditiva, dada a profundidade da árvore binária.

A avaliação experimental dissertada nesta Seção busca investigar a eficiência da solução PIPE*, quando a árvore binária é mantida incrementalmente. Nessa análise, foi observado se o crescimento da árvore, aumentando-se a quantidade de nós e, consequentemente, sua profundidade, causa impacto no tempo de processamento de busca na árvore para a construção da função preditiva.

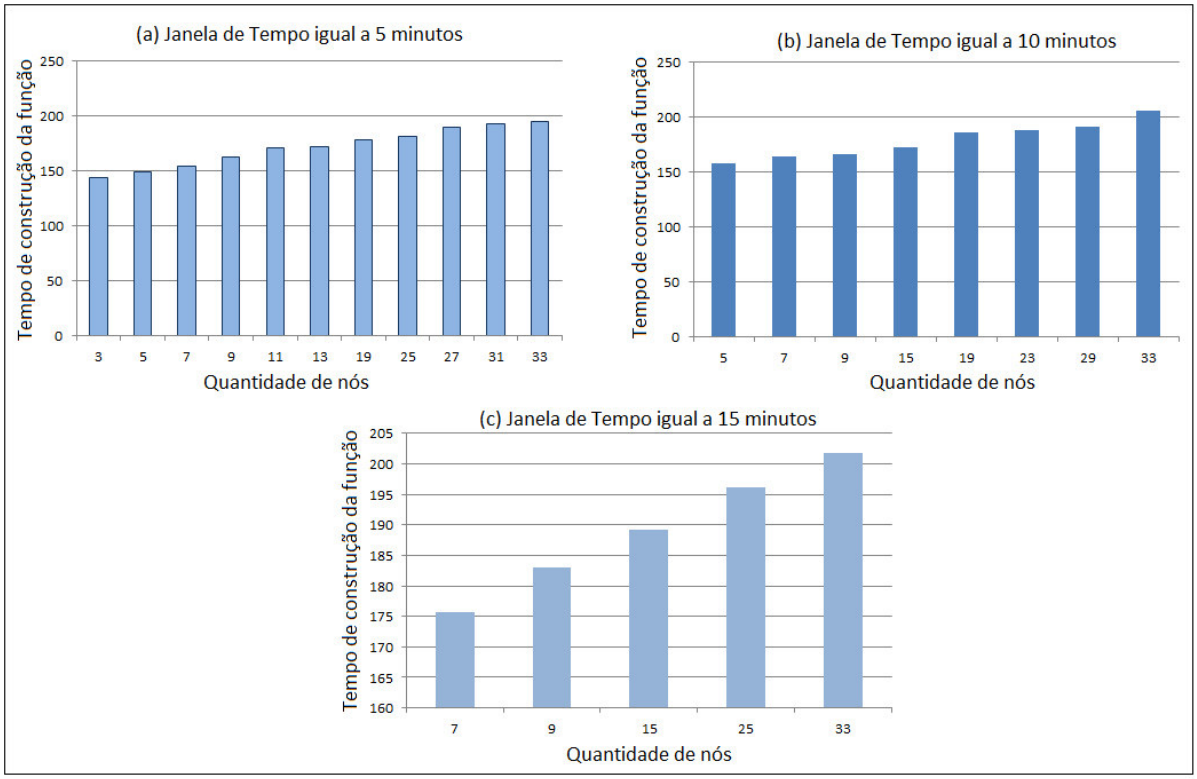
Para a observação dos resultados obtidos, foi verificada a variação dos nós que compõem a árvore binária e o tempo médio de processamento para construção da função. Dentro desse contexto, a Figura 31(a) mostra os resultados para a janela de tempo igual a 5 minutos, os quais indicam que, quando ocorre o crescimento da árvore a partir do número de nós que a compõem, ocorre também o aumento no tempo médio de processamento para a construção da função. No gráfico da Figura 31(a), o tempo gasto para computar a função preditiva é inicialmente, em média, de 144 milissegundos quando a árvore se mantém com apenas 3 nós. Quando a mesma árvore cresce em profundidade, sendo composta por até 33 nós, esse tempo total aumenta, em média, para 194 milissegundos.

Na Figura 31(b), que representa a janela de tempo igual a 10 minutos, também é identificado o aumento do tempo médio de processamento para a construção da função, dado o aumento do número de nós da árvore binária. Nessa execução, a árvore, que inicialmente começa com 5 nós, após o recebimento da primeira janela de tempo, realiza o processamento, para obter a função preditiva, em um tempo médio de até 157 milissegundos. Quando a mesma árvore termina a execução com 33 nós, esse tempo sobe para 201 milissegundos.

A Figura 31(c) mostra os resultados da execução, quando a janela de tempo é igual a 15 minutos. Nessa análise, também é identificado o aumento do tempo médio de processamento para a busca do tempo t requerido e a construção da função, que antes ocorria num tempo médio de até 175 milissegundos, quando a árvore era composta por 7 nós, e sobe para até 206 milissegundos quando ela passa a ser formada por 33 nós, já no final da execução. A intuição por trás de todas as análises é quanto maior o número de nós (isto é, quanto maior a profundidade da árvore), maior também será o tempo de busca por um nó específico, que contém o instante de tempo t, e maior será o tempo para computar a construção da função. As análises mostradas na Figura 31 consideraram o mesmo volume de dados e, por isso, todas as árvores tiveram, no final da execução, a mesma quantidade de nós. Nesse caso, é importante salitentar que existiram, em

certos instantes de tempo, janelas sendo recebidas sem conter pontos de localização.

Figura 31: Medição do tempo de construção da função, dado o crescimento da árvore binária.



Fonte: da autora.

5.7 Conclusão do Capítulo

Neste Capítulo, foram apresentadas avaliações experimentais relacionadas aos modelos PIPE – solução construída a partir de um processamento *do zero* – e PIPE* – que corresponde a uma solução incremental –, ambas geram como resultado uma função preditiva diferenciável. Para realizar a validação desses modelos, foram usadas informações acerca dos pontos de localização dos ônibus do Rio de Janeiro, descritos na Seção 5.1, e também dados referentes aos táxis de Fortaleza/CE, descritos na Seção 5.6. Para esse conjunto de dados, foi realizado um processo de limpeza e completude das informações, utilizando-se o algoritmo de *map matching* do *Barefoot* (BAREFOOT, 2017)

As análises experimentais mostraram resultados tanto de desempenho, quanto da acurácia das soluções. Do mesmo modo, foi investigado se as soluções poderiam ser utilizadas para identificar mudanças na reengenharia do tráfego de uma cidade. Dessa maneira, foi realizada uma experimentação específica, que comparou os dados da cidade do Rio de Janeiro um ano

antes do evento Olímpico e no ano das Olimpíadas (isto é, em 2016). Nesse caso, foi possível observar que as mudanças realizadas na cidade impactaram nos tempos de viagens dos objetos.

Este Capítulo também mostrou os resultados das análises que compararam dois modelos de predição: o *Incremental Descontínuo*, publicado em (NASCIMENTO *et al.*, 2016a), e a solução PIPE*, proposta nesta tese. Sabendo-se que, dada a distribuição dos dados, o modelo *Incremental Descontínuo* teve melhores resultados relacionados ao ajuste da solução, quando comparado com a solução *Piecewise*, foi necessário compará-lo com a solução PIPE*. Dentro desse contexto, os valores de AIC foram computados, com o objetivo de analisar qual deles tem o melhor ajuste quanto à distribuição dos dados. Os resultados mostrados na Seção 5.4 indicam um valor menor de AIC para o modelo PIPE*, indicando que essa estratégia tem um melhor ajuste que a anterior.

Além da análise acerca da bondade do ajuste, foi preciso investigar se o modelo PIPE* ainda apresenta melhores resultados que o modelo *Incremental Descontínuo*. Dentro desse contexto, foram comparados o tempo de processamento e a acurácia das soluções. Em todos os resultados, o modelo PIPE* obteve resultados melhores.

Sabendo-se que o modelo PIPE* atende melhor às necessidades das aplicações em tempo real, quando comparado com modelos competidores, foi realizada outra análise, que buscou investigar se a manutenção incremental era mesmo necessária. Assim, realizou-se uma análise comparativa entre a computação de uma solução *do zero* – chamada PIPE –, a qual gera uma árvore binária a partir de dados históricos, e a solução que realiza a manutenção incremental – chamada PIPE*. O objetivo dessa análise foi investigar se a persistência da informação e a construção da árvore, usando-se a solução PIPE, pode impactar os resultados relacionados tanto com os tempos de processamento, quanto com a acurácia das soluções. Na maior parte dos experimentos, foi possível observar que a solução PIPE*, que possibilita realizar a manutenção incremental, tem resultados melhores que a solução PIPE, quando analisados: *(i)* o tempo de processamento para atualizar a árvore binária, dado o recebimento de fluxos contínuos de trajetórias, e o tempo de processamento para construir o modelo *do zero*; além disso, *(ii)* foi realizada uma análise comparativa sobre a acurácia das soluções, dado o total de acertos observados. Na maioria dos resultados, o modelo PIPE* se comportou como o mais indicado para computar funções preditivas em sistemas que respondem a requisições em tempo real.

6 CONCLUSÃO E SUGESTÃO PARA TRABALHOS FUTUROS

A evolução espacial dos objetos móveis pode ser descrita continuamente ao longo do tempo. Dentre as características desses objetos, é possível citar que suas localizações e velocidades mudam constantemente, dados diferentes períodos do dia, e essas informações podem ser reportadas no formato de fluxos contínuos de trajetórias (LIU; SCHNEIDER, 2011). Atualmente, o serviço de rede sem fio tem possibilitado a captura e o relato sobre o movimento dos objetos, informando sua trajetória a todo instante. A análise desse dado é importante porque traz uma compreensão maior sobre a dinâmica de uma cidade, podendo ser utilizada nos processos de tomada de decisões.

Esta tese teve como contribuição o modelo PIPE*, que pode ser utilizado para construir funções preditivas em tempo real. O escopo de análise para validar este modelo está diretamente concentrado em dados de trajetórias. Dentro desse contexto, foi mostrado que as funções preditivas, geradas a partir do modelo proposto, podem auxiliar no entendimento do fluxo contínuo de trajetórias, considerando cada via específica da cidade.

A distribuição dos dados foi uma característica fundamental para a construção do modelo alcançado. As avaliações experimentais da solução PIPE concentraram-se na análise de dados estritamente reais, correspondentes às informações de duas cidades do Brasil – Rio de Janeiro e Fortaleza –, os quais relatam, respectivamente, o comportamento dos ônibus do Rio de Janeiro e dos táxis de Fortaleza. O modelo PIPE gera funções preditivas, cujo comportamento se aproxima de um conjunto de distribuições Gaussianas, como observado por (LIAO *et al.*, 2005), que afirma que essa distribuição consegue representar a distribuição dos dados de trajetória, devido ao processo de aceleração e desaceleração dos objetos móveis ao longo do dia.

As avaliações experimentais realizadas a partir do modelo PIPE tiveram o objetivo de analisar a acurácia e tempo de processamento da solução. Inicialmente, a solução alcançada neste trabalho foi comparada com um modelo competidor – chamado *Incremental Descontínuo* (NASCIMENTO *et al.*, 2016a). Essa análise buscou comparar os modelos com o intuito de identificar qual deles melhor explica a distribuição dos dados, dada a obtenção dos valores de *Akaike Information Criterion* (AIC) (AKAIKE, 1974). Os resultados alcançados mostraram um valor menor de AIC para o modelo proposto nesta tese, indicando que essa estratégia tem um melhor ajuste em relação à distribuição dos dados, quando comparada com a solução competidora. Outra análise realizada buscou apresentar resultados relacionados com os modelos PIPE – solução construída a partir de um processamento *do zero* – e PIPE* – que corresponde a uma solução

incremental –, as quais geram como resultado uma função preditiva diferenciável. Os resultados, basicamente, relataram sobre dois tipos principais de análises: *(i)* o tempo de processamento para atualizar a árvore binária, dado o recebimento de fluxos contínuos de trajetórias; e *(ii)* a acurácia das soluções. Na maioria dos resultados, o modelo PIPE* se comportou como o mais indicado para computar funções preditivas em sistemas que respondem a requisições em tempo real.

O Capítulo 2 apresentou um resumo dos principais conceitos usados para o entendimento desta tese, o qual envolve informações básicas, que foram usadas ao longo de todo o documento. Dessa forma, discorreu-se acerca de Trajetórias, Fluxos Contínuos de Trajetórias, *Map Matching*, Modelos de Regressão e Árvores Binárias Rotuladas.

O Capítulo 3 tratou do estado da arte, na área de pesquisa desta tese. Uma revisão sistemática foi realizada e foram obtidos trabalhos de autores que publicaram propostas acerca de conceitos relacionados a Análise de Dados de Trajetórias, Funções Preditivas Contínuas, Funções Preditivas Descontínuas e Modelos de Árvores a partir de Dados de Trajetórias. Essa discussão foi importante para investigar se as estratégias propostas neste trabalho melhor se adaptam às necessidades das predições realizadas sobre os dados de trajetórias.

O Capítulo 4 mostrou a solução *PIPE: Um Preditor de Tempos de Viagem usando Fluxo Contínuo de Trajetórias*, que pode ser criada ou atualizada a partir do recebimento de fluxos contínuos de trajetórias. Essa proposta é considerada incremental e foi criada para suprir as necessidades das propostas já existentes na literatura, as quais computam predições sobre esse tipo de dado. Para a construção do modelo, um conjunto de dados de trajetórias é recebido continuamente, dado um intervalo de tempo, com o objetivo de criar ou manter uma árvore binária, a qual será, posteriormente, usada para gerar a função preditiva em tempo de execução.

O Capítulo 5 apresentou a os resultados experimentais obtidos a partir das contribuições alcançadas nesta tese, as quais permitiram realizar análises para investigar a eficiência e a acurácia da solução, dado o recebimento de novos dados de trajetórias. É importante salientar que a avaliação experimental considerou a estratégia do Algoritmo *k-fold*, dividindo a base de dados em duas partes (uma de treinamento e outra de teste). Essa estratégia buscou investigar o total de acertos da função preditiva, dado um valor esperado. o modelo PIPE* se apresentou como mais vantajoso que as soluções competidoras, no sentido de ter melhor acurácia; melhor ajuste, quanto à distribuição dos dados; e melhor tempo de processamento, quando computados seus algoritmos de busca e atualização.

Todas as publicações realizadas, durante o período de execução dessa tese, são apresentadas a seguir.

- Samara Martins Nascimento, Mirla Rafaela Rafael Braga Chucre, José Antônio Fernandes de Macêdo, José Maria da Silva Monteiro Filho, Marco Antonio Casanova. On computing temporal functions for a time-dependent networks using trajectory data. In Proceedings of the 20th International Database Engineering & Applications Symposium (pp. 236-241). ACM, 2016.
- Samara Martins Nascimento, José Antônio Fernandes de Macêdo, Mirla Rafaela Rafael Braga Chucre, Marco Antonio Casanova, Javam de Castro Machado. On computing temporal functions for time-dependent networks using trajectory data streams. In Proceedings of the 7th ACM SIGSPATIAL International Workshop on GeoStreaming (p. 4). ACM, 2016.
- Samara Martins Nascimento, José Antônio Fernandes de Macêdo, Hélio Côrtes Vieira Lopes, Ticiana Linhares Coelho da Silva, Marco Antonio Casanova, Javam de Castro Machado.On computing travel time functions from Trajectory Data Streams. In Proceedings of the 8th ACM SIGSPATIAL International Workshop on GeoStreaming (p. 4). ACM, 2017.
- Mirla Rafaela Rafael Braga Chucre, Samara Martins Nascimento, José Antonio Fernandes de Macêdo, José Maria da Silva Monteiro Filho, Marco Antônio Casanova. Taxi, please! a nearest neighbor query in time-dependent road networks. In Mobile Data Management (MDM), 2016 17th IEEE International Conference on (Vol. 1, pp. 180-185). IEEE, 2016.

Como trabalhos futuros, são propostas algumas estratégias que permitem estender esse trabalho de pesquisa em diferentes direcionamentos, como:

- Possibilitar que o modelo PIPE seja criado a partir de outros tipos de árvores. Essa estratégia possibilita construir árvores menos profundas e, assim, algoritmos de funções preditivas com processamentos ainda melhores — dado que a função temporal é construída em tempo de execução, a partir da realização de uma busca na árvore.
- Estender essa estratégia para análise de séries temporais. O objetivo é identificar se as soluções propostas neste trabalho podem ser usadas para outros tipos de dados, possibilitando alcançar, por exemplo, descobertas de padrões.
- Realizar testes, usando o modelo PIPE dentro do contexto de tráfego, para descobrir padrões relacionados ao tráfego de uma cidade e comparar segmentos de ruas para identificar

possíveis similaridades.

- Realizar testes, considerando o uso do modelo PIPE em redes de ruas, e avaliar sua acurácia e tempo de processamento.
- Usar as funções diferenciáveis, obtidas a partir do modelo PIPE, em uma rede de ruas, a qual possibilite a utilização de consultas que analisam *Algoritmos de Menor Caminho*.
- Comparar a acurácia e os tempos de processamento da solução PIPE com Algoritmos como o *Gradient Boosting* e o *Randon Forest*, os quais podem ser utilizados em problemas de Classificação e Regressão.
- Avaliar a escalabilidade da solução PIPE.

REFERÊNCIAS

ABADI, D.; CARNEY, D.; CETINTEMEL, U.; CHERNIACK, M.; CONVEY, C.; ERWIN, C.; GALVEZ, E.; HATOUN, M.; MASKEY, A.; RASIN, A. et al. Aurora: a data stream management system. *In:* ACM. **Proceedings of the 2003 ACM SIGMOD international conference on Management of data.** *[S.l.]*, 2003. p. 666–666.

AGGARWAL, C. C. **Data streams: models and algorithms.** *[S.l.]*: Springer Science & Business Media, 2007. v. 31.

AGGARWAL, C. C.; HAN, J.; WANG, J.; YU, P. S. A framework for clustering evolving data streams. *In:* VLDB ENDOWMENT. **Proceedings of the 29th international conference on Very large data bases-Volume 29.** *[S.l.]*, 2003. p. 81–92.

AGGARWAL, C. C.; HAN, J.; WANG, J.; YU, P. S. On demand classification of data streams. *In:* ACM. **Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining.** *[S.l.]*, 2004. p. 503–508.

AKAIKE, H. A new look at the statistical model identification. **IEEE transactions on automatic control,** Ieee, v. 19, n. 6, p. 716–723, 1974.

ALT, H.; EFRAT, A.; ROTE, G.; WENK, C. Matching planar maps. **Journal of Algorithms,** Elsevier, v. 49, n. 2, p. 262–283, 2003.

ARASU, A.; BABCOCK, B.; BABU, S.; DATAR, M.; ITO, K.; NISHIZAWA, I.; ROSENSTEIN, J.; WIDOM, J. Stream: the stanford stream data manager (demonstration description). *In:* ACM. **Proceedings of the 2003 ACM SIGMOD international conference on Management of data.** *[S.l.]*, 2003. p. 665–665.

ARTHUR, D.; VASSILVITSKII, S. k-means++: The advantages of careful seeding. *In:* SOCIETY FOR INDUSTRIAL AND APPLIED MATHEMATICS. **Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms.** *[S.l.]*, 2007. p. 1027–1035.

ASGHARI, M.; EMRICH, T.; DEMIRYUREK, U.; SHAHABI, C. Probabilistic estimation of link travel times in dynamic road networks. *In:* ACM. **Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems.** *[S.l.]*, 2015. p. 47.

BABCOCK, B.; BABU, S.; DATAR, M.; MOTWANI, R.; WIDOM, J. Models and issues in data stream systems. *In:* ACM. **Proceedings of the twenty-first ACM SIGMOD-IGACT-SIGART symposium on Principles of database systems.** *[S.l.]*, 2002. p. 1–16.

BABCOCK, B.; DATAR, M.; MOTWANI, R. et al. Load shedding techniques for data stream systems. *In:* **Proceedings of the 2003 Workshop on Management and Processing of Data Streams.** *[S.l.: s.n.]*, 2003. v. 577.

BAREFOOT. **Main Page — Barefoot.** 2017. Acesso em: 10 de Novembro de 2017. Disponível em: https://github.com/bmwcarit/barefoot.

BRAKATSOULAS, S.; PFOSER, D.; SALAS, R.; WENK, C. On map-matching vehicle tracking data. *In:* VLDB ENDOWMENT. **Proceedings of the 31st international conference on Very large data bases.** *[S.l.]*, 2005. p. 853–864.

BRUSH, A.; KRUMM, J.; SCOTT, J. Exploring end user preferences for location obfuscation, location-based services, and the value of location. *In:* ACM. **Proceedings of the 12th ACM international conference on Ubiquitous computing.** *[S.l.]*, 2010. p. 95–104.

CADEZ, I. V.; GAFFNEY, S.; SMYTH, P. A general probabilistic framework for clustering individuals and objects. *In:* ACM. **Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining.** *[S.l.]*, 2000. p. 140–149.

CAI, Y.; NG, R. Indexing spatio-temporal trajectories with chebyshev polynomials. *In:* ACM. **Proceedings of the 2004 ACM SIGMOD international conference on Management of data.** *[S.l.]*, 2004. p. 599–610.

CAMERON, S. H. **Piece-wise linear approximations.** *[S.l.]*, 1966.

CHAKRABARTI, K.; KEOGH, E.; MEHROTRA, S.; PAZZANI, M. Locally adaptive dimensionality reduction for indexing large time series databases. **ACM Transactions on Database Systems (TODS),** ACM, v. 27, n. 2, p. 188–228, 2002.

CHANDOLA, V.; BANERJEE, A.; KUMAR, V. Anomaly detection: A survey. **ACM computing surveys (CSUR),** ACM, v. 41, n. 3, p. 15, 2009.

CHAWATHE, S. S. Segment-based map matching. *In:* IEEE. **Intelligent Vehicles Symposium, 2007 IEEE.** *[S.l.]*, 2007. p. 1190–1197.

CHEN, D. Z.; WANG, H. Approximating points by a piecewise linear function. **Algorithmica,** Springer, v. 66, n. 3, p. 682–713, 2013.

CHEN, L.; ÖZSU, M. T.; ORIA, V. Robust and fast similarity search for moving object trajectories. *In:* ACM. **Proceedings of the 2005 ACM SIGMOD international conference on Management of data.** *[S.l.]*, 2005. p. 491–502.

CHEN, S.; WANG, W.; ZUYLEN, H. van. A comparison of outlier detection algorithms for its data. **Expert Systems with Applications,** Elsevier, v. 37, n. 2, p. 1169–1178, 2010.

CHEN, Y.; DONG, G.; HAN, J.; WAH, B. W.; WANG, J. Multi-dimensional regression analysis of time-series data streams. *In:* VLDB ENDOWMENT. **Proceedings of the 28th international conference on Very Large Data Bases.** *[S.l.]*, 2002. p. 323–334.

CHEN, Y.; GAO, L.; LI, Z.-p.; LIU, Y.-c. A new method for urban traffic state estimation based on vehicle tracking algorithm. *In:* IEEE. **Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE.** *[S.l.]*, 2007. p. 1097–1101.

CIVILIS, A.; JENSEN, C. S.; PAKALNIS, S. Techniques for efficient road-network-based tracking of moving objects. **IEEE Transactions on Knowledge and Data Engineering,** IEEE, v. 17, n. 5, p. 698–712, 2005.

CORTES, C.; LAVANYA, R.; OH, J.; JAYAKRISHNAN, R. A general purpose methodology for link travel time estimation using multiple point detection of traffic. institute of transportation studies, university of california, irvine, reportno. **Irvine, CA, University of California,** 2001.

DAI, J.; YANG, B.; GUO, C.; JENSEN, C. S. Efficient and accurate path cost estimation using trajectory data. **arXiv preprint arXiv:1510.02886,** 2015.

DAILEY, D. J. Travel-time estimation using cross-correlation techniques. **Transportation Research Part B: Methodological,** Elsevier, v. 27, n. 2, p. 97–107, 1993.

DEMÉTRIO, C. G. B.; ZOCCHI, S. S. Modelos de regressão. **Piracicaba: ESALQ,** 2006.

DIVISION POPULATION DISTRIBUTION, U. U. P. Internal migration and development: An international perspective. p. 1, 2011.

DOMINGOS, P.; HULTEN, G. A general method for scaling up machine learning algorithms and its application to clustering. *In:* **ICML.** *[S.l.: s.n.]*, 2001. v. 1, p. 106–113.

DUDA, R. O.; HART, P. E.; STORK, D. G. **Pattern classification.** *[S.l.]*: John Wiley & Sons, 2012.

ELMELEEGY, H.; ELMAGARMID, A. K.; CECCHET, E.; AREF, W. G.; ZWAENEPOEL, W. Online piece-wise linear approximation of numerical streams with precision guarantees. **Proceedings of the VLDB Endowment, VLDB Endowment,** v. 2, n. 1, p. 145–156, 2009.

EPPERSON, J. F. **An introduction to numerical methods and analysis.** *[S.l.]*: John Wiley & Sons, 2013.

FALOUTSOS, C.; RANGANATHAN, M.; MANOLOPOULOS, Y. **Fast subsequence matching in time-series databases.** *[S.l.]*: ACM, 1994. v. 23.

FRANK, E.; WANG, Y.; INGLIS, S.; HOLMES, G.; WITTEN, I. H. Using model trees for classification. **Machine Learning,** Springer, v. 32, n. 1, p. 63–76, 1998.

FRENTZOS, E.; GRATSIAS, K.; THEODORIDIS, Y. Index-based most similar trajectory search. *In:* IEEE. **2007 IEEE 23rd International Conference on Data Engineering.** *[S.l.]*, 2007. p. 816–825.

FU, T.-c.; CHUNG, F.-l.; NG, V.; LUK, R. Evolutionary segmentation of financial time series into subsequences. *In:* IEEE. **Evolutionary Computation, 2001. Proceedings of the 2001 Congress on.** *[S.l.]*, 2001. v. 1, p. 426–430.

GABER, M. M.; ZASLAVSKY, A.; KRISHNASWAMY, S. Mining data streams: a review. **ACM Sigmod Record,** ACM, v. 34, n. 2, p. 18–26, 2005.

GABER, M. M.; ZASLAVSKY, A.; KRISHNASWAMY, S. Data stream mining. *In:* **Data Mining and Knowledge Discovery Handbook.** *[S.l.]*: Springer, 2009. p. 759–787.

GALTON, F. Regression towards mediocrity in hereditary stature. **The Journal of the Anthropological Institute of Great Britain and Ireland,** JSTOR, v. 15, p. 246–263, 1886.

GAMA, J. **Knowledge discovery from data streams.** *[S.l.]*: CRC Press, 2010.

GAMA, J.; GABER, M. M. **Learning from data streams: processing techniques in sensor networks.** *[S.l.]*: Springer, 2007.

GAMA, J.; MEDAS, P.; RODRIGUES, P. Learning decision trees from dynamic data streams. *In:* ACM. **Proceedings of the 2005 ACM symposium on Applied computing.** *[S.l.]*, 2005. p. 573–577.

GIANNOTTI, F.; NANNI, M.; PINELLI, F.; PEDRESCHI, D. Trajectory pattern mining. *In:* ACM. **Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining.** *[S.l.]*, 2007. p. 330–339.

GOLOVCHENKO, N. **Least-squares fit of a continuous piecewise linear function.** *[S.l.]*: August, 2004.

GUDMUNDSSON, J.; KREVELD, M. van. Computing longest duration flocks in trajectory data. *In:* ACM. **Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems.** *[S.l.]*, 2006. p. 35–42.

GUESSOUS, Y.; ARON, M.; BHOURI, N.; COHEN, S. Estimating travel time distribution under different traffic conditions. **Transportation Research Procedia,** Elsevier, v. 3, p. 339–348, 2014.

GUHA, S.; HUANG, Z. Revisiting the direct sum theorem and space lower bounds in random order streams. **Automata, Languages and Programming,** Springer, p. 513–524, 2009.

GUJARATI, D. N.; PORTER, D. C. **Econometria Básica-5.** *[S.l.]*: Amgh Editora, 2011.

GUPTA, M.; GAO, J.; AGGARWAL, C. C.; HAN, J. Outlier detection for temporal data: A survey. **IEEE Transactions on Knowledge and Data Engineering,** IEEE, v. 26, n. 9, p. 2250–2267, 2014.

GUPTA, P.; THAKKAR, A.; GANATRA, A. Comprehensive study on techniques of incremental learning with decision trees for streamed data. **International Journal of Engineering and Advanced Technology (IJEAT) ISSN,** p. 2249–8958, 2013.

HAKIMI, S. L.; SCHMEICHEL, E. F. Fitting polygonal functions to a set of points in the plane. **CVGIP: Graphical Models and Image Processing,** Elsevier, v. 53, n. 2, p. 132–136, 1991.

HE, Q.; ZHUANG, F.; LI, J.; SHI, Z. Parallel implementation of classification algorithms based on mapreduce. **Rough Set and Knowledge Technology,** Springer, p. 655–662, 2010.

HEATH, M. T. **Scientific computing: An introductory survey.** *[S.l.]*: McGraw-Hill, 1997.

HUANG, Y.; CHEN, C.; DONG, P. Modeling herds and their evolvements from trajectory data. *In:* **Geographic Information Science.** *[S.l.]*: Springer, 2008. p. 90–105.

ISHAK, S.; AL-DEEK, H. Performance evaluation of short-term time-series traffic prediction model. **Journal of Transportation Engineering,** American Society of Civil Engineers, v. 128, n. 6, p. 490–498, 2002.

JENSEN, A.; LARSEN, T. Travel-time estimation in road networks using gps data. **Unpublished,** v. 8, n. 8, 2014.

JEUNG, H.; SHEN, H. T.; ZHOU, X. Convoy queries in spatio-temporal databases. *In:* IEEE. **2008 IEEE 24th International Conference on Data Engineering.** *[S.l.]*, 2008. p. 1457–1459.

JEUNG, H.; YIU, M. L.; ZHOU, X.; JENSEN, C. S.; SHEN, H. T. Discovery of convoys in trajectory databases. **Proceedings of the VLDB Endowment,** VLDB Endowment, v. 1, n. 1, p. 1068–1080, 2008.

KHARRAT, A.; POPA, I. S.; ZEITOUNI, K.; FAIZ, S. Clustering algorithm for network constraint trajectories. *In:* **Headway in Spatial Data Handling.** *[S.l.]*: Springer, 2008. p. 631–647.

KIM, H.; LOH, W.-Y. Classification trees with unbiased multiway splits. **Journal of the American Statistical Association,** Taylor & Francis, v. 96, n. 454, p. 589–604, 2001.

KIM, H.; LOH, W.-Y. Classification trees with bivariate linear discriminant node models. **Journal of Computational and Graphical Statistics,** Taylor & Francis, v. 12, n. 3, p. 512–530, 2003.

KINOSHITA, A.; TAKASU, A.; AIHARA, K.; ISHII, J.; KURASAWA, H.; SATO, H.; NAKAMURA, M.; ADACHI, J. Gps trajectory data enrichment based on a latent statistical model. *In:* **ICPRAM.** *[S.l.: s.n.]*, 2016. p. 255–262.

KISGYÖRGY, L.; RILETT, L. R. Travel time prediction by advanced neural network. **Periodica Polytechnica. Civil Engineering,** Periodica Polytechnica, Budapest University of Technology and Economics, v. 46, n. 1, p. 15, 2002.

KONG, X.; XU, Z.; SHEN, G.; WANG, J.; YANG, Q.; ZHANG, B. Urban traffic congestion estimation and prediction based on floating car trajectory data. **Future Generation Computer Systems,** Elsevier, v. 61, p. 97–107, 2016.

KUBRUSLY, J.; LOPES, H. Constructive regression on implicit regions. **Advances and Applications in Statistics,** Pushpa Publishing House, v. 45, n. 3, p. 201, 2015.

LAGE, M.; BORDIGNON, A.; PETRONETTO, F.; VEIGA, A.; TAVARES, G.; LEWINER, T.; LOPES, H. Approximations by smooth transitions in binary space partitions. *In:* IEEE. **Computer Graphics and Image Processing, 2008. SIBGRAPI'08. XXI Brazilian Symposium on.** *[S.l.]*, 2008. p. 230–236.

LAKSHMI, T. M.; MARTIN, A.; BEGUM, R. M.; VENKATESAN, V. P. An analysis on performance of decision tree algorithms using student's qualitative data. **International Journal of Modern Education and Computer Science,** Modern Education and Computer Science Press, v. 5, n. 5, p. 18, 2013.

LAZARIDIS, I.; MEHROTRA, S. Capturing sensor-generated time series with quality guarantees. *In:* IEEE. **Data Engineering, 2003. Proceedings. 19th International Conference on.** *[S.l.]*, 2003. p. 429–440.

LEE, J.-G.; HAN, J.; LI, X. Trajectory outlier detection: A partition-and-detect framework. *In:* IEEE. **Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on.** *[S.l.]*, 2008. p. 140–149.

LEE, P.; LAKSHMANAN, L. V.; MILIOS, E. E. Incremental cluster evolution tracking from highly dynamic network data. *In:* IEEE. **Data Engineering (ICDE), 2014 IEEE 30th International Conference on.** *[S.l.]*, 2014. p. 3–14.

LEE, W.-C.; KRUMM, J. Trajectory preprocessing. *In:* **Computing with spatial trajectories.** *[S.l.]*: Springer, 2011. p. 3–33.

LI, F.; BONNIFAIT, P.; IBANEZ-GUZMAN, J.; ZINOUNE, C. Lane-level map-matching with integrity on high-definition maps. *In:* IEEE. **Intelligent Vehicles Symposium (IV), 2017 IEEE.** *[S.l.]*, 2017. p. 1176–1181.

LI, Y.; ZHENG, Y.; ZHANG, H.; CHEN, L. Traffic prediction in a bike-sharing system. *In:* ACM. **Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems.** *[S.l.]*, 2015. p. 33.

LI, Z.; DING, B.; HAN, J.; KAYS, R. Swarm: Mining relaxed temporal moving object clusters. **Proceedings of the VLDB Endowment,** VLDB Endowment, v. 3, n. 1-2, p. 723–734, 2010.

LIAO, L.; FOX, D.; KAUTZ, H. Location-based activity recognition using relational markov networks,[in proc. 19th int. *In:* **Joint Conf. Artif. Intell., Edinburgh, Scotland.** *[S.l.: s.n.]*, 2005. p. 773–778.

LIBEN-NOWELL, D.; VEE, E.; ZHU, A. Finding longest increasing and common subsequences in streaming data. **Journal of Combinatorial Optimization,** Springer, v. 11, n. 2, p. 155–175, 2006.

LIEBIG, T.; PIATKOWSKI, N.; BOCKERMANN, C.; MORIK, K. Dynamic route planning with real-time traffic predictions. **Information Systems,** Elsevier, v. 64, p. 258–265, 2017.

LIU, H.; SCHNEIDER, M. Querying moving objects with uncertainty in spatio-temporal databases. *In:* SPRINGER. **Database Systems for Advanced Applications.** *[S.l.]*, 2011. p. 357–371.

LIU, W.; ZHENG, Y.; CHAWLA, S.; YUAN, J.; XING, X. Discovering spatio-temporal causal interactions in traffic data streams. *In:* ACM. **Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining.** *[S.l.]*, 2011. p. 1010–1018.

LIU, Y.; LI, Z. A novel algorithm of low sampling rate gps trajectories on map-matching. **EURASIP Journal on Wireless Communications and Networking,** Springer International Publishing, v. 2017, n. 1, p. 30, 2017.

LOH, W.-Y. Classification and regression trees. **Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery,** Wiley Online Library, v. 1, n. 1, p. 14–23, 2011.

LUO, A.; CHEN, S.; XV, B. Enhanced map-matching algorithm with a hidden markov model for mobile phone positioning. **ISPRS International Journal of Geo-Information,** Multidisciplinary Digital Publishing Institute, v. 6, n. 11, p. 327, 2017.

LUO, G.; YI, K.; CHENG, S.-W.; LI, Z.; FAN, W.; HE, C.; MU, Y. Piecewise linear approximation of streaming time series data with max-error guarantees. *In:* IEEE. **Data Engineering (ICDE), 2015 IEEE 31st International Conference on.** *[S.l.]*, 2015. p. 173–184.

MA, S.; ZHENG, Y.; WOLFSON, O. T-share: A large-scale dynamic taxi ridesharing service. *In:* IEEE. **Data Engineering (ICDE), 2013 IEEE 29th International Conference on.** *[S.l.]*, 2013. p. 410–421.

MUTHUKRISHNAN, S. et al. Data streams: Algorithms and applications. **Foundations and Trends R in Theoretical Computer Science,** Now Publishers, Inc., v. 1, n. 2, p. 117–236, 2005.

NADUNGODAGE, C. H.; XIA, Y.; LI, F.; LEE, J. J.; GE, J. Streamfitter: a real time linear regression analysis system for continuous data streams. *In:* SPRINGER. **International Conference on Database Systems for Advanced Applications.** *[S.l.]*, 2011. p. 458–461.

NAM, D. H.; DREW, D. R. Traffic dynamics: Method for estimating freeway travel times in real time from flow measurements. **Journal of Transportation Engineering,** American Society of Civil Engineers, v. 122, n. 3, p. 185–191, 1996.

NASCIMENTO, S. M.; CHUCRE, M. R.; CASANOVA, M. A.; MACHADO, J. et al. On computing temporal functions for time-dependent networks using trajectory data streams. *In:* ACM. **Proceedings of the 7th ACM SIGSPATIAL International Workshop on GeoStreaming.** *[S.l.]*, 2016. p. 10.

NASCIMENTO, S. M.; CHUCRE, M. R. R. B.; MACEDO, J. A. F.; FILHO, J. M. S. M.; CASANOVA, M. A. On computing temporal functions for a time-dependent networks using trajectory data. *In:* **IDEAS 2016: 20th International Database Engineering & Applications Symposium.** *[S.l.: s.n.]*, 2016.

NASCIMENTO, S. M.; MACEDO, J. A.; LOPES, H.; CV; SILVA, T. L.; CASANOVA, M. A. et al. On computing travel time functions from trajectory data streams. *In:* ACM. **Proceedings of the 8th ACM SIGSPATIAL International Workshop on GeoStreaming.** *[S.l.]*, 2017. p. 10.

NETO, F. D. N. et al. Modelos baseados em ppm para previsão de trajetórias utilizando informações contextuais. Universidade Federal de Campina Grande, 2017.

OCHIENG, W. Y.; QUDDUS, M.; NOLAND, R. B. Map-matching in complex urban road networks. **Revista Brasileira de Cartografia,** v. 2, n. 55, 2003.

OH, J.-S.; JAYAKRISHNAN, R.; RECKER, W. Section travel time estimation from point detection data. **Center for Traffic Simulation Studies,** 2002.

PAN, B.; ZHENG, Y.; WILKIE, D.; SHAHABI, C. Crowd sensing of traffic anomalies based on human mobility and social media. *In:* ACM. **Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems.** *[S.l.]*, 2013. p. 344–353.

PANAGIOTAKIS, C.; PELEKIS, N.; KOPANAKIS, I.; RAMASSO, E.; THEODORIDIS, Y. Segmentation and sampling of moving object trajectories based on representativeness. **TKDE,** p. 1328–1343, 2012.

PATEL, S.; HARDAHA, M.; SEETPAL, M. K.; MADANKAR, K. Multiple linear regression model for stream flow estimation of wainganga river. **American Journal of Water Science and Engineering,** v. 2, n. 1, p. 1–5, 2016.

PATTARA-ATIKOM, W.; PONGPAIBOOL, P.; THAJCHAYAPONG, S. Estimating road traffic congestion using vehicle velocity. *In:* IEEE. **ITS Telecommunications Proceedings, 2006 6th International Conference on.** *[S.l.]*, 2006. p. 1001–1004.

PETTY, K. F.; BICKEL, P.; OSTLAND, M.; RICE, J.; SCHOENBERG, F.; JIANG, J.; RITOV, Y. Accurate estimation of travel times from single-loop detectors. **Transportation Research Part A: Policy and Practice,** Elsevier, v. 32, n. 1, p. 1–17, 1998.

PINK, O.; HUMMEL, B. A statistical approach to map matching using road network geometry, topology and vehicular motion constraints. *In:* IEEE. **Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on.** *[S.l.]*, 2008. p. 862–867.

POPIVANOV, I.; MILLER, R. J. Similarity search over time-series data using wavelets. *In:* IEEE. **Data Engineering, 2002. Proceedings. 18th International Conference on.** *[S.l.]*, 2002. p. 212–221.

POTAMIAS, M.; PATROUMPAS, K.; SELLIS, T. Sampling trajectory streams with spatiotemporal criteria. *In:* IEEE. **Scientific and Statistical Database Management, 2006. 18th International Conference on.** *[S.l.]*, 2006. p. 275–284.

POTTS, D.; SAMMUT, C. Incremental learning of linear model trees. **Machine Learning,** Springer, v. 61, n. 1-3, p. 5–48, 2005.

QI, J.; ZHANG, R.; RAMAMOHANARAO, K.; WANG, H.; WEN, Z.; WU, D. Indexable online time series segmentation with error bound guarantee. **World Wide Web,** Springer, v. 18, n. 2, p. 359–401, 2015.

QIN, Y.; SHENG, Q. Z.; FALKNER, N. J.; DUSTDAR, S.; WANG, H.; VASILAKOS, A. V. When things matter: A survey on data-centric internet of things. **Journal of Network and Computer Applications,** Elsevier, v. 64, p. 137–153, 2016.

QUDDUS, M.; WASHINGTON, S. Shortest path and vehicle trajectory aided map-matching for low frequency gps data. **Transportation Research Part C: Emerging Technologies,** Elsevier, v. 55, p. 328–339, 2015.

QUDDUS, M. A.; NOLAND, R. B.; OCHIENG, W. Y. A high accuracy fuzzy logic based map matching algorithm for road transport. **Journal of Intelligent Transportation Systems,** Taylor & Francis, v. 10, n. 3, p. 103–115, 2006.

QUINLAN, J. R. Combining instance-based and model-based learning. *In:* **Proceedings of the Tenth International Conference on Machine Learning.** *[S.l.: s.n.]*, 1993. p. 236–243.

QUINLAN, J. R. **C4.5: programs for machine learning.** *[S.l.]*: Elsevier, 2014.

RAFIEI, D.; MENDELZON, A. Similarity-based queries for time series data. *In:* ACM. **ACM SIGMOD Record.** *[S.l.]*, 1997. v. 26, n. 2, p. 13–25.

RIO, B. **Main Page — Prefeitura.** 2017. Acesso em: 11de Maio 2017. Disponível em: http://www.rio.rj.gov.br/web/smo/exibeconteudo?id=5303190.

RIO, O. **Main Page — Prefeitura.** 2017. Acesso em: 11de Maio 2017. Disponível em: http://prefeitura.rio/web/guest/exibeconteudo?id=1130901.

RIO, P. **Main Page — Prefeitura.** 2017. Acesso em: 11de Maio 2017. Disponível em: http://data.rio/dataset/gps-de-onibus.

SCHNITZLER, F.; LIEBIG, T.; MANNOR, S.; MORIK, K. Combining a gauss-markov model and gaussian process for traffic prediction in dublin city center. *In:* **EDBT/ICDT Workshops.** *[S.l.: s.n.]*, 2014. p. 373–374.

SILVA, J. A.; FARIA, E. R.; BARROS, R. C.; HRUSCHKA, E. R.; CARVALHO, A. C. de; GAMA, J. Data stream clustering: A survey. **ACM Computing Surveys (CSUR),** ACM, v. 46, n. 1, p. 13, 2013.

SILVA, T. L. C. da; ZEITOUNI, K.; MACÊDO, J. A. de. Online clustering of trajectory data stream. *In:* IEEE. **Mobile Data Management (MDM), 2016 17th IEEE International Conference on.** *[S.l.]*, 2016. v. 1, p. 112–121.

SILVA, T. L. C. da; ZEITOUNI, K.; MACÊDO, J. A. F. de; CASANOVA, M. A. On-line mobility pattern discovering using trajectory data. *In:* **EDBT.** *[S.l.: s.n.]*, 2016. p. 682–683.

SIMPLES, T. **Main Page Táxi Simples.** 2015. Acesso em: 5 de Novembro de 2015. Disponível em: https://taxisimples.com.br.

SONG, M.; WANG, H. Highly efficient incremental estimation of gaussian mixture models for online data stream clustering. *In:* **Proc. of SPIE Vol.** *[S.l.: s.n.]*, 2005. v. 5803, p. 175.

SPILIOPOULOU, M.; NTOUTSI, I.; THEODORIDIS, Y.; SCHULT, R. Monic: Modeling and monitoring cluster transitions. *In:* **Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.** *[S.l.: s.n.]*, 2006. p. 706–711.

SRIMANI, P.; PATIL, M. M. Regression model using instance based learning streams. **Indian Journal of Science and Technology,** v. 7, n. 6, p. 864–870, 2014.

SRINIVASAN, K.; JOVANIS, P. Determination of number of probe vehicles required for reliable travel time measurement in urban network. **Transportation Research Record: Journal of the Transportation Research Board,** Transportation Research Board of the National Academies, n. 1537, p. 15–22, 1996.

STORM. **Github — Algoritmo Storm.** 2016. Acesso em: 10 de Dezembro de 2016. Disponível em: https://github.com/nathanmarz/storm/wiki.

SUN, L.; YANG, J.; MAHMASSANI, H. Travel time estimation based on piecewise truncated quadratic speed trajectory. **Transportation Research Part A: Policy and Practice,** Elsevier, v. 42, n. 1, p. 173–186, 2008.

SUN, S.; XU, X. Variational inference for infinite mixtures of gaussian processes with applications to traffic flow prediction. **IEEE Transactions on Intelligent Transportation Systems,** IEEE, v. 12, n. 2, p. 466–475, 2011.

TANG, L.-A.; ZHENG, Y.; YUAN, J.; HAN, J.; LEUNG, A.; HUNG, C.-C.; PENG, W.-C. On discovery of traveling companions from streaming trajectories. *In:* **ICDE.** *[S.l.: s.n.]*, 2012. p. 186–197.

TANG, L.-A.; ZHENG, Y.; YUAN, J.; HAN, J.; LEUNG, A.; PENG, W.-C.; PORTA, T. L. A framework of traveling companion discovery on trajectory data streams. **ACM Transactions on Intelligent Systems and Technology (TIST),** ACM, v. 5, n. 1, p. 3, 2013.

TATBUL, N.; ÇETINTEMEL, U.; ZDONIK, S.; CHERNIACK, M.; STONEBRAKER, M. Load shedding on data streams. *In:* **Proceedings of the Workshop on Management and Processing of Data Streams (MPDS 03), San Diego, CA, USA.** *[S.l.: s.n.]*, 2003.

TØNDEL, P.; JOHANSEN, T. A.; BEMPORAD, A. Evaluation of piecewise affine control via binary search tree. **Automatica,** Elsevier, v. 39, n. 5, p. 945–950, 2003.

TORGO, L. Computationally efficient linear regression trees. *In:* **Classification, Clustering, and Data Analysis.** *[S.l.]*: Springer, 2002. p. 409–415.

UTGOFF, P. E.; BERKMAN, N. C.; CLOUSE, J. A. Decision tree induction based on efficient tree restructuring. **Machine Learning,** Springer, v. 29, n. 1, p. 5–44, 1997.

WANG, Y.; ZHENG, Y.; XUE, Y. Travel time estimation of a path using sparse trajectories. *In:* ACM. **Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining.** *[S.l.]*, 2014. p. 25–34.

WITTEN, I. H.; FRANK, E.; HALL, M. A.; PAL, C. J. **Data Mining: Practical machine learning tools and techniques.** *[S.l.]*: Morgan Kaufmann, 2016.

YANG, B.; GUO, C.; JENSEN, C. S. Travel cost inference from sparse, spatio temporally correlated time series using markov models. **Proceedings of the VLDB Endowment,** VLDB Endowment, v. 6, n. 9, p. 769–780, 2013.

YANG, B.; GUO, C.; JENSEN, C. S.; KAUL, M.; SHANG, S. Stochastic skyline route planning under time-varying uncertainty. *In:* IEEE. **Data Engineering (ICDE), 2014 IEEE 30th International Conference on.** *[S.l.]*, 2014. p. 136–147.

YANG, X. Y.; LIU, Z.; FU, Y. Mapreduce as a programming model for association rules algorithm on hadoop. *In:* IEEE. **Information Sciences and Interaction Sciences (ICIS), 2010 3rd International Conference on.** *[S.l.]*, 2010. p. 99–102.

YE, P.; CHEN, Z.; XU, L. Analyzing travel time variability on transit route using gps data. *In:* **ICTE 2015.** *[S.l.: s.n.]*, 2015. p. 448–456.

YE, Y.; ZHENG, Y.; CHEN, Y.; FENG, J.; XIE, X. Mining individual life pattern based on location history. *In:* IEEE. **Mobile Data Management: Systems, Services and Middleware, 2009. MDM'09. Tenth International Conference on.** *[S.l.]*, 2009. p. 1–10.

YI, B.-K.; FALOUTSOS, C. Fast time sequence indexing for arbitrary lp norms. *In:* VLDB. *[S.l.]*, 2000.

YOON, J.; NOBLE, B.; LIU, M. Surface street traffic estimation. *In:* ACM. **Proceedings of the 5th international conference on Mobile systems, applications and services.** *[S.l.]*, 2007. p. 220–232.

YUAN, J.; ZHENG, Y.; XIE, X.; SUN, G. T-drive: Enhancing driving directions with taxi drivers' intelligence. **IEEE Trans. Knowl. Data Eng.**, v. 25, n. 1, p. 220–232, 2013.

YUAN, J.; ZHENG, Y.; ZHANG, C.; XIE, W.; XIE, X.; SUN, G.; HUANG, Y. T-drive: driving directions based on taxi trajectories. *In:* ACM. **Proceedings of the 18th SIGSPATIAL International conference on advances in geographic information systems.** *[S.l.]*, 2010. p. 99–108.

ZHANG, J.-D.; XU, J.; LIAO, S. S. Aggregating and sampling methods for processing gps data streams for traffic state estimation. **IEEE Transactions on Intelligent Transportation Systems,** IEEE, v. 14, n. 4, p. 1629–1641, 2013.

ZHANG, P.; ZHOU, C.; WANG, P.; GAO, B. J.; ZHU, X.; GUO, L. E-tree: An efficient indexing structure for ensemble models on data streams. **IEEE Transactions on Knowledge and Data Engineering,** IEEE, v. 27, n. 2, p. 461–474, 2015.

ZHAO, H.-y.; LI, G.-x.; ZHANG, H.-l.; XUE, Y. An improved algorithm for segmenting online time series with error bound guarantee. **International Journal of Machine Learning and Cybernetics,** Springer, v. 7, n. 3, p. 365–374, 2016.

ZHENG, K.; ZHENG, Y.; YUAN, N. J.; SHANG, S. On discovery of gathering patterns from trajectories. *In:* **ICDE.** *[S.l.: s.n.]*, 2013. p. 242–253.

ZHENG, K.; ZHENG, Y.; YUAN, N. J.; SHANG, S.; ZHOU, X. Online discovery of gathering patterns over trajectories. **IEEE Transactions on Knowledge and Data Engineering,** IEEE, v. 26, n. 8, p. 1974–1988, 2014.

ZHENG, Y. Trajectory data mining: an overview. **ACM Transactions on Intelligent Systems and Technology (TIST),** ACM, v. 6, n. 3, p. 29, 2015.

ZHENG, Y.; LIU, Y.; YUAN, J.; XIE, X. Urban computing with taxicabs. *In:* ACM. **Proceedings of the 13th international conference on Ubiquitous computing.** *[S.l.]*, 2011. p. 89–98.

ZHENG, Y.; ZHOU, X. **Computing with spatial trajectories.** *[S.l.]*: Springer Science & Business Media, 2011.