

PGCOMP - Programa de Pós-Graduação em Ciência da Computação
Universidade Federal da Bahia (UFBA)
Av. Milton Santos, s/n - Ondina
Salvador, BA, Brasil, 40170-110

<https://pgcomp.ufba.br>
pgcomp@ufba.br

With the growth in the number of vehicles worldwide in recent years, the adoption of technologies to deal with the consequences that this volume of vehicles can generate for large cities, such as increased congestion on highways, has become necessary. Vehicular Ad-Hoc Networks (VANETs) present themselves as a promising technology in this scenario, assisting in the formation of vehicular networks capable of interconnecting vehicles and infrastructure to understand and deal with traffic congestion. Considering this, the amount of data generated by this environment increases proportionally as the number of vehicles on the roads grows. Consequently, sending data from the vehicular environment to the structure that identifies congestion can become increasingly burdensome from a network utilization perspective, leading to overloads and increased latency, thus hindering quick decision-making. Thus, we propose in this work the construction of a Framework aimed at identifying vehicle congestion, with an approach to reducing, at the fog layer, the data generated by a VANET and then sending only the most relevant data to the cloud for decision-making. In addition to congestion detection, with historical data in time series format, we perform congestion prediction using ARIMA. To work with data reduction, the Framework utilizes simple sampling algorithms and clustering techniques (DBSCAN and XMEANS). The results showed that the use of clustering algorithms in this Framework achieved a significant level of accuracy in detecting traffic congestion along with a pronounced reduction in network usage compared to the Baseline algorithm.

Palavras-chave: PData Reduction, Clusterization, Machine Learning, Fog Computing, Ad Hoc Vehicular Networks, Intelligent Transportation System.

Um Framework baseado em Fog Computing para redução de dados em um sistema de detecção de tráfego para VANETs

Adriano Humberto de Oliveira Maia

Dissertação de Mestrado

Universidade Federal da Bahia

Programa de Pós-Graduação em
Ciência da Computação

Dezembro | 2023

MSC | 175 | 2023

Um Framework baseado em Fog Computing para redução de dados em um sistema de detecção de tráfego para VANETs

Adriano Humberto de
Oliveira Maia

UFBA





Universidade Federal da Bahia
Instituto de Computação

Programa de Pós-Graduação em Ciência da Computação

**UM FRAMEWORK BASEADO EM FOG
COMPUTING PARA REDUÇÃO DE DADOS
EM UM SISTEMA DE DETECÇÃO DE
TRÁFEGO PARA VANETS**

Adriano Humberto de Oliveira Maia

DISSERTAÇÃO DE MESTRADO

Salvador
Dezembro de 2023

ADRIANO HUMBERTO DE OLIVEIRA MAIA

**UM FRAMEWORK BASEADO EM FOG COMPUTING PARA
REDUÇÃO DE DADOS EM UM SISTEMA DE DETECÇÃO DE
TRÁFEGO PARA VANETS**

Esta Dissertação de Mestrado foi apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal da Bahia, como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Orientador: Professor Dr. Maycon Leone Maciel Peixoto

Salvador
Dezembro de 2023

Sistema de Bibliotecas - UFBA

M217 Maia, Adriano Humberto de Oliveira.

Um Framework baseado em Fog Computing para redução de dados em um sistema de detecção de tráfego para VANETs / Adriano Humberto de Oliveira Maia – Salvador, 2023.

49p.: il.

Orientador: Prof. Dr. Professor Dr. Maycon Leone Maciel Peixoto.

Dissertação (Mestrado) – Universidade Federal da Bahia, Instituto de Computação, 2023.

1.Ciência da Computação. 2. Redução de Dados. 3. Redes de Computadores.. I. Peixoto, Maycon Leone Maciel. II. Universidade Federal da Bahia. Instituto de Computação. III Título.

CDD – 000

CDU – 004

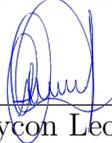
TERMO DE APROVAÇÃO

ADRIANO HUMBERTO DE OLIVEIRA MAIA

UM FRAMEWORK BASEADO EM FOG COMPUTING PARA REDUÇÃO DE DADOS EM UM SISTEMA DE DETECÇÃO DE TRÁFEGO PARA VANETS

Esta Dissertação de Mestrado foi julgada adequada à obtenção do título de Mestre em Ciência da Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação da Universidade Federal da Bahia.

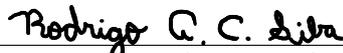
Salvador, 20 de dezembro de 2023



Prof. Dr. Maycon Leone Maciel Peixoto
Universidade Federal da Bahia



Prof. Dr. Roberto Rodrigues Filho
Universidade Federal de Santa Catarina



Prof. Dr. Rodrigo Augusto Cardoso da Silva
Universidade Federal do ABC

Dedico este trabalho à minha família e amigos, pela sua ajuda diária, paciência e confiança, sem as quais este trabalho não teria o mesmo desfecho, aos meus pares de pesquisa e professores que tanto contribuíram para o desenvolvimento deste trabalho.

AGRADECIMENTOS

Registro aqui meus sinceros agradecimentos à minha família, amigos de infância, amigos pesquisadores e profissionais que contribuíram com discussões, ideias, apoio técnico sobre diversos aspectos deste projeto.

Agradeço aos professores da UFBA pela competência e qualidade técnica, além da disponibilidade para mostrar os caminhos que levam ao conhecimento.

Entre estes, agradeço em especial ao Prof. Dr. Maycon Leone, pela sua contribuição na definição dos objetivos, ajuda, compreensão, paciência e, principalmente, sua capacidade de inspirar os seus alunos na busca pelos melhores resultados, fazendo isso com empatia, humildade e dedicação!

E a todos que participaram direta ou indiretamente deste trabalho.

*Acredite em si e chegará um dia em que os outros não terão outra
escolha senão acreditar com você.*

— (Cynthia Kersey)

RESUMO

Com o crescimento da quantidade de veículos no mundo nos últimos anos, se tornou necessária a adoção de tecnologias para lidar com as consequências que esse volume veicular pode gerar para as grandes cidades, como o aumento dos congestionamentos nas rodovias. As redes veiculares Ad-Hoc (VANETs) se apresentam como uma tecnologia promissora nesse cenário, auxiliando na formação de redes veiculares capazes de interconectar veículos e infraestrutura para entender e lidar com congestionamento de veículos. Considerando isso, a quantidade de dados gerados por esse ambiente aumenta na proporção que o número de veículos nas vias cresce. Consequentemente, o envio dos dados do ambiente veicular para a estrutura que identifica o congestionamento pode ser cada vez mais oneroso no ponto de vista de utilização de rede, podendo gerar sobrecargas e aumento da latência e dificultando a tomada de decisão rápida. Dessa forma, propomos neste trabalho, a construção de um *Framework* que visa identificar congestionamento de veículos, com uma abordagem para reduzir, na camada de névoa, os dados gerados por uma VANET e depois enviar para a nuvem somente os dados mais relevantes para tomada de decisão. Além da detecção de congestionamento, com os dados históricos em formato de séries temporais realizamos, a previsão de congestionamento utilizando ARIMA. Para trabalhar com a redução dos dados, o *Framework* utiliza algoritmos de amostragem simples e técnicas de agrupamento (DBSCAN e XMEANS). Os resultados mostraram que, a utilização de algoritmos de agrupamento nesse *Framework* alcançaram um nível significativo de acurácia na detecção de congestionamento de tráfego em conjunto com uma redução acentuada do uso de rede em comparação com o algoritmo Baseline.

Palavras-chave: Redução de Dados, Agrupamentos, Aprendizado de Máquina, Computação em Névoa, Redes Ad Hoc Veiculares, Sistemas de Transporte Inteligente.

ABSTRACT

With the growth in the number of vehicles worldwide in recent years, the adoption of technologies to deal with the consequences that this volume of vehicles can generate for large cities, such as increased congestion on highways, has become necessary. Vehicular Ad-Hoc Networks (VANETs) present themselves as a promising technology in this scenario, assisting in the formation of vehicular networks capable of interconnecting vehicles and infrastructure to understand and deal with traffic congestion. Considering this, the amount of data generated by this environment increases proportionally as the number of vehicles on the roads grows. Consequently, sending data from the vehicular environment to the structure that identifies congestion can become increasingly burdensome from a network utilization perspective, leading to overloads and increased latency, thus hindering quick decision-making. Thus, we propose in this work the construction of a Framework aimed at identifying vehicle congestion, with an approach to reducing, at the fog layer, the data generated by a VANET and then sending only the most relevant data to the cloud for decision-making. In addition to congestion detection, with historical data in time series format, we perform congestion prediction using ARIMA. To work with data reduction, the Framework utilizes simple sampling algorithms and clustering techniques (DBSCAN and XMEANS). The results showed that the use of clustering algorithms in this Framework achieved a significant level of accuracy in detecting traffic congestion along with a pronounced reduction in network usage compared to the Baseline algorithm.

Keywords: Data Reduction, Clusterization, Machine Learning, Fog Computing, Ad Hoc Vehicular Networks, Intelligent Transportation System.

SUMÁRIO

Capítulo 1—Introdução	1
1.1 Hipótese	4
1.2 Questão de Pesquisa	4
1.3 Objetivo Geral	5
1.3.1 Objetivos Específicos	5
1.4 Principais Contribuições	5
1.5 Organização	6
Capítulo 2—Framework Proposto	7
2.1 O Framework	9
2.2 Fluxo de dados entre as camadas	15
2.3 Definição do ambiente de Simulação	15
2.4 Fluxo de execução do Framework	16
Capítulo 3—Resultados	21
3.0.1 Metodologia do cenário e da Simulação	23
3.0.2 Análise das velocidades dos experimentos	23
3.0.3 Análise do LOS dos experimentos	26
3.0.4 Previsão de dados com a utilização do ARIMA	32
3.0.5 Análise das velocidades com ARIMA	33
3.0.6 Análise do LOS com ARIMA	34
3.0.7 Análise de rede	35
Capítulo 4—Conclusões	43
4.1 Trabalhos Futuros	44
4.2 Publicações	45

LISTA DE FIGURAS

2.1	Visão geral da estrutura e do fluxo de dados.	8
2.2	Framework de redução de dados.	9
2.3	Fluxo de Dados Contínuo no <i>Framework</i>	16
2.4	Fluxo de execução do Framework.	17
3.1	Mapa da cidade de Cologne na Alemanha.	22
3.2	Trechos utilizados com base para análise dos resultados.	24
3.3	Análise da Velocidade para 30 veículos.	25
3.4	Análise da Velocidade para 50 veículos.	25
3.5	Análise da Velocidade para 70 veículos.	26
3.6	Análise do LOS para 30 veículos.	27
3.7	Análise do LOS para 50 veículos.	28
3.8	Análise do LOS para 70 veículos.	28
3.9	Detecção de congestionamento de tráfego com densidade de 30 veículos/ km^2	30
3.10	Detecção de congestionamento de tráfego com densidade de 50 veículos/ km^2	31
3.11	Detecção de congestionamento de tráfego com densidade de 70 veículos/ km^2	32
3.12	Análise da Velocidade com predição do Arima para 30 veículos.	34
3.13	Análise da Velocidade com predição do Arima para 50 veículos.	35
3.14	Análise da Velocidade com predição do Arima para 70 veículos.	36
3.15	Análise do LOS com predição do Arima para 30 veículos.	37
3.16	Análise do LOS com predição do Arima para 50 veículos.	38
3.17	Análise do LOS com predição do Arima para 70 veículos.	39
3.18	Comparativo da Largura de Banda e a Quantidade de Veículos.	40
3.19	Estimativa de custo de armazenamento (anual).	41

LISTA DE TABELAS

2.1	HCM Classification.	8
3.1	Parâmetros de simulação	24
3.2	Classificação (w_{ij}) das Condições de Tráfego	29
3.3	Network Usage	37

LISTA DE SIGLAS

UC	Urban Computing.....	1
IoT	Internet of Things.....	2
ITS	Intelligente Transport Systems	1
VANETs	Vehicular Ad Hoc Networks	2
CC	Cloud Computing	3

INTRODUÇÃO

Nos últimos anos, o processo de urbanização vem crescendo de forma acelerada nos grandes centros e com isso gerando aumento problemas relacionados ao congestionamento de veículos, poluição em geral, situações relacionadas à segurança do cidadão, o aumento do consumo de energia, e outros desafios que geram impacto no bem estar social. Mais especificamente na área de transporte rodoviário, existe um desafio diário relacionado ao aumento da densidade de veículos, que pode apresentar um quadro de lentidão parcial ou retenção total do trânsito em muitos momentos do cotidiano. Diante disso, o tempo de deslocamento dos indivíduos está cada vez maior impactando no bem estar social da população, meio ambiente, aumento dos custos e eventuais prejuízos para a economia (HOU et al., 2016; ZHENG et al., 2014; STEFANELLI, 2015).

Para enfrentar esses desafios, são oferecidas tecnologias que auxiliam os usuários na tomada da decisão para lidar com as questões relacionadas ao trânsito, como escolha da melhor rota considerando a situação do tráfego, acidente ou zona perigosa. Essas tecnologias atuam desde a captação de informações através de dispositivos e pessoas, até a disponibilidade de recursos computacionais robustos, escaláveis e distribuídos para tratamento dos dados gerados. A aplicação dessas abordagens de captação de dados de múltiplas fontes, processamento e análise da informação é denominada de Computação Urbana (do inglês, *Urban Computing (UC)*) (ZHENG et al., 2014).

A *UC* está em constante desenvolvimento e em crescente implementação nas cidades modernas. Devido a essa progressão, as estratégias, planos, investimentos e aquisições de estrutura tecnológica, estão cada vez mais direcionados para suprir essa demanda. (ZHENG et al., 2014).

Uma área de destaque da *UC* é a de Sistemas de Transporte Inteligente (do inglês, *Intelligent Transport Systems (ITS)*). Os *ITS* têm como objetivo geral atenuar alguns desafios da mobilidade urbana, utilizando a tecnologia como meio principal para a promoção de um serviço mais consistente para a população. Na prática, esse ambiente utiliza a cooperação entre sensores, dispositivos e estruturas computacionais para obtenção de dados e realização de análises para tomada de decisão (GUIDO et al., 2017), (CERQUEIRA et al., 2018).

De acordo com (CERQUEIRA et al., 2018), aplicações de um ambiente de *ITS*, dispõem de características determinantes como:

- Utilização de múltiplas fontes de dados para originar informação;
- Multiplicidade de tecnologias de comunicação, com variação de alcance, atrasos e tempo de entrega sensíveis;
- Necessidade constante de uma tomada de decisão em tempo real;
- Alto nível de exigência de qualidade de serviço, tendo uma rigidez maior para algumas situações, como detecção, definição e divulgação de congestionamentos para um grupo de veículos, por exemplo.

Essas características das aplicações voltadas para uma estrutura de *ITS*, impulsionaram o estudo e o desenvolvimento de tecnologias importantes como as Redes *Ad hoc* Veiculares (do inglês, *Vehicular Ad Hoc Networks (VANETs)*) e a Internet das Coisas (do inglês, *Internet of Things (IoT)*).

As *VANETs* são redes que utilizam a comunicação sem fio em ambientes veiculares. Esse tipo de rede possibilita o estudo e desenvolvimento de serviços que ajudam os usuários do sistema de trânsito e seus elementos. Por exemplo, aplicações que atuam identificando áreas com restrição de segurança, rotas livres de congestionamento e pontos de acidentes na rota de determinados veículos, podem utilizar essa tecnologia de comunicação (AL-SULTAN et al., 2014). As *VANETs* são dotadas de um rede de sensores para coleta de dados e comunicação entre os elementos de rede. A Internet das Coisas auxilia nessa definição e utilização dessa estrutura de coleta e comunicação de dados.

A *IoT* é um conjunto de tecnologias que visam a conexão de diferentes objetos, equipados com os mais diversos sensores e tecnologias de comunicação, que trocam mensagens para gerar dados mais completos sobre um determinado ambiente ou aplicação e auxiliam na tomada de decisão (ATZORI; IERA; MORABITO, 2010; Zhu et al., 2019). A Internet das Coisas vem evoluído significativamente nos últimos anos e muitos estudos atuais projetam um aumento considerável de sua utilização. Para ilustrar esse avanço, a *International Data Corporation (IDC)* no relatório de 2023 mostra que o número de conexões IoT cresceu 18%, em 2022 chegando a 14,3 bilhões de dispositivos ativos no mesmo ano (IDC, 2023). O mesmo relatório projeta um crescimento de 16%, chegando até 16,7 bilhões de *devices* IoT ativos.

A ascensão dessas tecnologias trouxe um avanço nas capacidades dos sensores, evolução nas tecnologias de comunicação entre equipamentos e uma redução no custo de aquisição, possibilitando a consolidação de uma malha sensorial mais densa e acessível (GUIDO et al., 2017).

Esse avanço favorece a obtenção de uma estrutura mais ampla para aplicações de *ITS* e conseqüentemente uma geração de dados cada vez maior para análise. Portanto, o volume dos dados gerados por um ecossistema de *ITS* pode ser muito grande, sua produção ocorre em alta velocidade e está em constante progressão. Essa progressão de geração de dados pode determinar uma complexidade computacional e uma elevação dos custos relacionados com a comunicação (GUIDO et al., 2017).

Uma questão adicional de destaque nas aplicações destinadas as *ITS* atuais é a tendência de centralização dos recursos computacionais em um ambiente de Computação em Nuvem (do inglês, *Cloud Computing (CC)*), de forma unificada e exclusiva. Essa abordagem tem como objetivo trazer um aumento da confiabilidade, escalabilidade, agilidade no retorno da informação, sistemas e serviços relacionados. Porém, como descrito anteriormente, existem particularidades inerentes ao ambiente de *ITS*, que não são bem aderentes a uma estrutura centralizada, pois muitas vezes dependem de uma resposta célere e com baixa interferência de um enlace computacional (LEITE et al., 2017), (BATISTA et al., 2018).

Os mecanismos do Sistemas de Transporte Inteligente, de forma geral precisam de uma baixa latência e baixo *Jitter* para a comunicação entre cenário urbano e os grandes datacenters da *CC* (YI; LI; LI, 2015), para assim atender requisições em tempo real associadas aos desafios das grandes cidades. Por exemplo, identificar um congestionamento e reportar essa informação com rapidez para um eventual veículo para que a sua rota seja alterada evitando o encontro com essa retenção de trânsito. Além disso, a totalidade dos dados devem ser transportados para a *CC*, podendo representar uma quantidade elevada de dados e aumentando consideravelmente os custos atribuídos a comunicação, armazenamento e processamento, pois a tarifação habitual desses recursos está na carga de dados utilizada (LEITE et al., 2017).

Portanto, tratar um grande volume de dados gerados pelas *VANETs*, equipadas com uma rede de sensores cada vez mais extensa, em um ambiente centralizado e distante do cenário urbano, pode gerar atrasos e comprometer a entrega da informação em tempo real. Além do fato da elevação constante e exponencial dos custos relacionados a operação de aplicações *ITS* (BILAL et al., 2018).

Atualmente existem inúmeras aplicações que trabalham de forma automática e muitas vezes colaborativa, para analisar e apresentar para os usuários as condições do trânsito e suas variações, como acidentes, definição de áreas de risco, picos de congestionamento, suporte a semáforos inteligentes e veículos de emergência (SILVA et al., 2019; Zhu et al., 2019). Pesquisas sobre ferramentas que utilizam sensores nos veículos e nas vias para a propagação veloz das informações relacionadas ao trânsito, estão em constante crescimento. Entretanto, de forma geral esses estudos não estão orientados para tratamento dos problemas oriundos do requisito de disponibilização da informação em tempo real, a exemplo da necessidade da manutenção da baixa latência e baixo *Jitter* (CHIANG, 2016). Outra carência de abordagem nestas pesquisas está na redução dos custos associados a comunicação para esse tipo de aplicação, considerando a projeção de aumento significativo da estrutura e do número de *VANETs* previsto para os próximos anos.

Por conta do progresso significativo da Internet das Coisas, sistemas computacionais para *ITS* estão em pleno desenvolvimento. Um contratempo relevante gerado por essas aplicações é que precisam transmitir enormes volumes de dados heterogêneos, provenientes de ambientes distintos de mobilidade urbana (SOUZA et al., 2018).

Em um cenário de uso exclusivo do paradigma de *Cloud Computing*, se faz necessária a utilização da Internet para transitar esse conhecimento, o que pode gerar severos problemas na eficiência da comunicação de rede, afetando a confiabilidade da latência e gerando uma instabilidade no tempo de atraso (DESAI, 2018; MUKHERJEE; SHU;

WANG, 2018), além do aumento dos custos de banda para transmissão entre os pontos participantes e processamento nos servidores centralizados na Nuvem, pois essas tarifas são realizadas sob demanda.

Apesar das projeções de aumento da rapidez e eficiência com a implantação da tecnologia 5G, esse crescimento exponencial de informação pode causar impactos consideráveis para os ecossistemas de Sistemas Inteligentes de Transporte, incluindo eventuais congestionamentos nas redes de comunicação. Variações de eficácia na entrega da informação em tempo real para tomada de decisão, das aplicações com a arquitetura de comunicação direta com a Nuvem, tendem a ser observadas nesses casos.

Utilizar a *Cloud Computing* de forma centralizada para receber e processar os dados gerados pela rede veicular, sem perder de vista as diretrizes supracitadas, está cada vez mais complexo. Existem situações que o volume de dados gerados pelos sensores pode ter redundância de informação, irrelevância ou pouca alteração de valores ao longo do tempo e esse cenário pode ocupar recurso computacional de forma desnecessária.

A motivação para a construção deste trabalho, está na definição de uma aplicação *ITS* para identificação de congestionamento, com uma abordagem de tratamento dos dados gerados de forma descentralizada, através da adoção da estrutura de Névoa, alocada diretamente no ambiente de computação urbana. Essa camada intermediária, lida com um grande volume de dados, tomando decisões com o foco na redução da quantidade dos dados transmitidos.

Essa disposição de rede, trabalha na *Fog Computing* com algoritmos e técnicas para redução dos dados apresentados pelos sensores, entregando para a *Cloud Computing* apenas informação relevante para identificação do congestionamento. Essa arquitetura multicamada lida com fluxos contínuos de dados.

O objetivo desse processo é detectar o fluxo de trânsito e gerenciar o tráfego utilizando apenas uma fração dos dados que será entregue pela infraestrutura intermediária, visando reduzir os custos de comunicação.

1.1 HIPÓTESE

Um sistema para detecção de congestionamento de veículos que utiliza um Framework para redução de dados pode mitigar a sobrecarga na rede de comunicação sem reduzir a acurácia das avaliações para gerenciamento do fluxo de tráfego.

1.2 QUESTÃO DE PESQUISA

A questão de pesquisa deste trabalho é a seguinte: *A implementação de um framework de redução de dados, utilizando computação em névoa (Fog Computing) e algoritmos de agrupamento, em um contexto de fluxo de dados contínuo, pode efetivamente contribuir para maximizar a eficiência no fluxo de tráfego e, ao mesmo tempo, reduzir os custos de comunicação?*

1.3 OBJETIVO GERAL

Desenvolver uma solução para identificação de congestionamento de veículos, apoiada por um framework de redução de dados baseado em computação em névoa para lidar com os desafios de latência e custos de comunicação durante a identificação de congestionamento, monitoramento e gestão do tráfego.

1.3.1 Objetivos Específicos

- Propor um Framework para redução de dados;
- Utilizar técnicas de redução e algoritmos de agrupamento para reduzir dados na estrutura da Névoa, com posterior entrega apenas dos dados relevantes para a Nuvem.
- Implementar um modelo de classificação de congestionamento veicular com base nos dados recebidos da névoa;
- Implementar um modelo de predição de congestionamento com base em séries temporais, utilizando os dados históricos gerados pelo *framework*.

1.4 PRINCIPAIS CONTRIBUIÇÕES

As principais contribuições desta dissertação estão relacionadas à redução de dados em redes veiculares, permitindo melhor gerenciamento, redução da complexidade e dos custos de comunicação destes ambientes. A seguir, algumas características que se destacaram durante o desenvolvimento desse documento:

1. **Definição do *Framework***: Implementação de uma arquitetura para selecionar, processar e distribuir de forma eficiente o fluxo contínuo de dados recebidos das redes veiculares;
2. **Estratégias de redução de dados**: Adaptação dos algoritmos **DBSCAN** e **X-Means** para trabalhar em uma conjuntura de fluxo de dados contínuo, possibilitando a identificação de pontos de congestionamento a partir de um percentual de dados contidos nos agrupamentos identificados pelas diferentes abordagens;
3. **Redução dos custos de comunicação**: As estratégias de redução de dados desaguam na consequente redução dos custos relacionados ao **OPEX (Operational Expenditure)**, ou seja, com as despesas relacionadas a operação. Mais especificamente, os benefícios aparecem nos fatores de redução no armazenamento de dados e redução dos custos de comunicação;
4. **Identificação do volume de trânsito em fluxo de dados contínuo**: Implementação de um modelo de classificação contínua do estado do tráfego, com base no monitoramento dos dados obtidos da VANET e utilizando a escala **LOS (Level of Services)**;

5. **Utilização de algoritmo de predição de dados:** Fundamentado no histórico dos dados de classificação do trânsito obtidos em tempo real, foi implementado o **ARIMA (Autoregressive Integrated Moving Average)** como mecanismo de predição de congestionamento de veículos.
6. **Submissões:** Realização de submissões parciais e completa desta pesquisa com a finalidade de validar a presente proposta junto à comunidade acadêmica.

1.5 ORGANIZAÇÃO

O restante desse trabalho está organizado da seguinte forma: o **Capítulo 2** apresenta a fundamentação teórica dessa pesquisa. O **Capítulo 3** descreve os trabalhos relacionados com a nossa proposta. O **Capítulo 4** demonstra a proposta, metodologia e resultados. O **Capítulo 5** a conclusão, trabalhos futuros e publicações.

FRAMEWORK PROPOSTO

Os sistemas de transporte inteligente são essenciais para a computação urbana, pois ajudam a lidar com os desafios do tráfego e contribuem para a melhoria da qualidade de vida. O presente trabalho propõe um Framework para identificação de congestionamento de veículos que utiliza computação em Névoa, para executar algoritmos e técnicas para redução dos dados, com foco de diminuir a utilização e os custos de rede.

O *Framework* proposto foi estruturado em camadas de *VANET*, *Fog Computing* e *Cloud Computing*. Os dados são coletados da *VANET* e enviados para a *Fog Computing* em um contexto de fluxo de dados contínuos em conjunto com um abordagem de janelas temporais, resultando em blocos de dados que chegam a névoa e que representam o ambiente veicular em um determinado tempo. A escolha dessa disposição de tecnologias, visa reduzir a quantidade dos dados enviados para tomada de decisão e conseqüentemente diminuir a utilização de rede.

Para facilitar o entendimento da proposta, a visão geral do *Framework* é ilustrada na Figura 2.1. Nesse modelo, podemos identificar o fluxo dos dados em todo o *Framework*. É possível observar a geração dos dados no ambiente veicular, a comunicação dos veículos com a *RSU/FOG*, o tratamento dos dados na névoa e a posterior entrega dos dados para a nuvem, onde o processo de identificação de congestionamento é realizado.

A camada de rede veicular, possui uma *RSU/Fog* que coleta de forma contínua os dados enviados pelos veículos que estão dentro da sua área de cobertura. Os carros enviam sua identificação, posição (latitude e longitude), velocidade, rodovia, data e hora. Esse processo acontece em fluxo de dados contínuo e, para viabilizar a comunicação entre veículos e *RSU*, utilizamos o protocolo IEEE 802.11p.

Os dados oriundos do ambiente veicular, se acumulam dentro de uma janela de tempo e após esse período a *RSU/FOG* aplica alguma técnica de redução ou de agrupamento de dados. Quando a janela de tempo atinge o limite, uma nova janela do mesmo tamanho é criada, e o processo de agregação de dados na *RSU/FOG* acontece novamente. Esse processo se repete até o fim da simulação.

A informação resultante é enviada da névoa para a nuvem utilizando a interface LTE/5G. Na nuvem é realizada a classificação do congestionamento de tráfego, com base

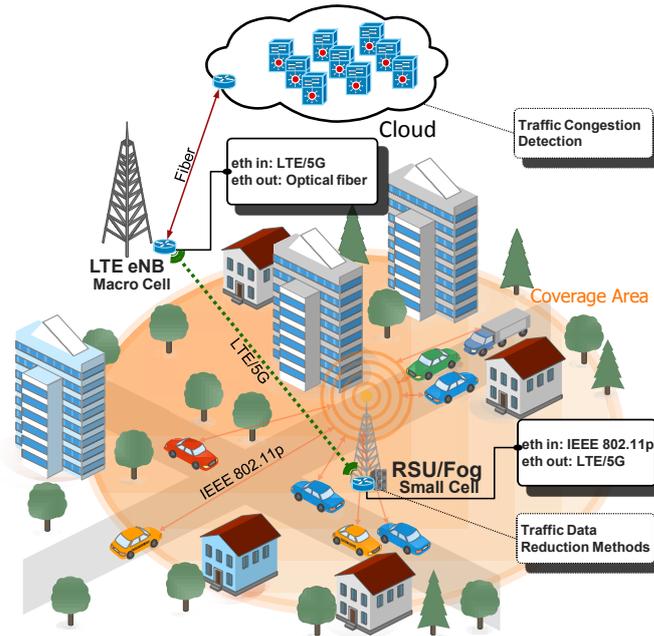


Figura 2.1: Visão geral da estrutura e do fluxo de dados.

no cálculo do Nível de Serviço (*Level of Service - LOS*). LOS foi extraído do Manual de Capacidade da Rodovia (*Highway Capacity Manual - HCM*) (BOARD, 2010). Essa abordagem tem seis níveis diferentes de fluxo de carros, para descrever as condições de trânsito, de acordo com a tabela a seguir:

Tabela 2.1: HCM Classification.

(w_i)	LOS	Traffic Congestion Classification
[0, 0.15]	A	Free-flow
[0.15, 0.33]	B	Reasonably free-flow
[0.33, 0.50]	C	Stable-flow
[0.50, 0.60]	D	Approaching unstable-flow
[0.60, 0.70]	E	Unstable-flow
[0.70, 1.00]	F	Breakdown-flow

Os dados gerados por um ambiente veicular operando com envio contínuo, sem nenhuma técnica de redução, podem apresentar um grande volume para ser processado. Esse cenário pode proporcionar um congestionamento da estrutura de rede, perda de pacotes, aumento do atraso e aumento de custo. A adoção da Fog como extensão da nuvem nesse projeto, pode promover uma redução significativa dos dados enviados para tomada de decisão e consequentemente apresenta uma redução nos custos de comunicação entre a VANET e a Cloud.

A Seção 2.1 descreve o funcionamento do *Framework* suas camadas e os algoritmos relacionados. A Seção 2.2 apresenta como o fluxo de dados contínuos funciona e transita entre as camadas do modelo. Já a Seção 2.3 apresenta o arranjo de simuladores que foram utilizados na pesquisa. Na Seção 2.4 é apresentado todo o sentido de execução do *framework* e os elementos que estão em sua composição.

2.1 O FRAMEWORK

Como podemos observar na Figura 2.2, o framework proposto neste projeto possui três camadas que funcionam de maneira integrada para lidar com redução de latência, redução do volume de dados e as consequências destas reduções nos custos de comunicação. Importante destacar que essa arquitetura pressupõe um fluxo de dados contínuo da camada de mobilidade para as camadas adjacentes do modelo.

Cada camada executa operações independentes e possuem protocolos e tecnologias de comunicações diferentes. Essas camadas foram desenvolvidas de modo a ter um desacoplamento entre elas, garantindo que eventuais ajustes em uma camada não interfiram na outra. Essa arquitetura possibilita a diminuição dos dados enviados para a nuvem, considerando que existe um tratamento de redução dos dados recebidos pela névoa que envia apenas os dados relevantes para a camada superior.

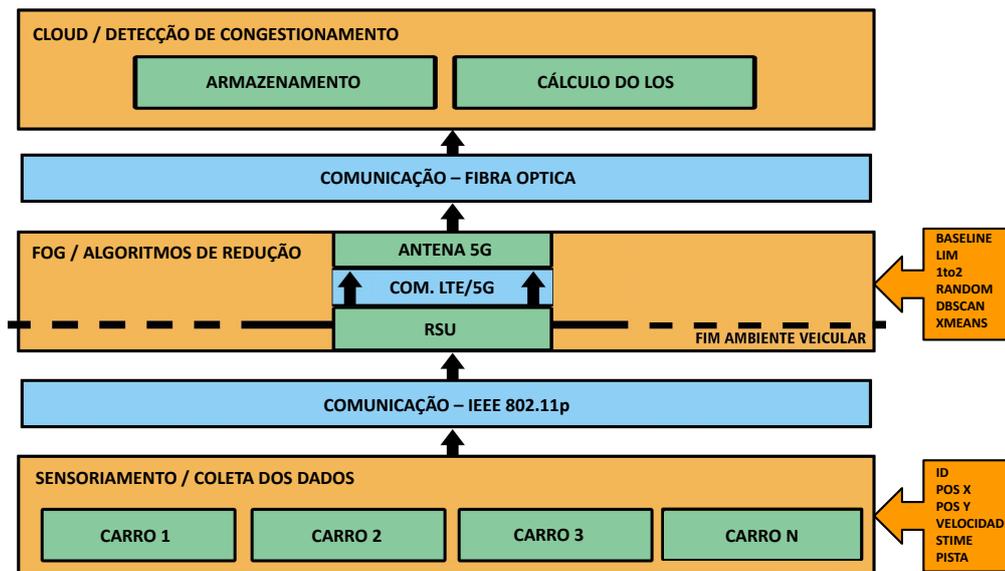


Figura 2.2: Framework de redução de dados.

A primeira camada é a de Coleta de dados que será constituída por veículos localizados no cenário urbano. Cada veículo envia periodicamente seus dados de identificação e posicionamento e por uma RSU que receberá os dados dos elementos que estão em seu alcance. Essa coleta ocorre através da adoção do protocolo de comunicação IEEE 802.11p e dentro de uma janela de tempo definida.

Subsequentemente, a segunda camada é a camada de Fog Computing, sua posição física está localizada no ambiente de rede veicular dentro da própria RSU e fornece

uma alternativa para processamento de dados na borda da rede. Nessa camada serão implementados algoritmos que visam reduzir os dados coletados e concentrados na janela temporal. Após a execução do processo de redução os dados resultantes são enviados para a nuvem. Para enviar os dados a RSU/Fog utilizará a tecnologia LTE/5G.

Por fim, é a camada a nuvem, que nesse projeto tem o objetivo de classificar o fluxo de tráfego de acordo com o LOS e armazenar essa informação.

Na Figura 2.2, os dados coletados na camada de sensoriamento e os algoritmos utilizados na camada de FOG, estão destacados fora dos respectivos retângulos, para demonstrar que eles não são obrigatoriamente fixos nesse Framework, ou seja, em trabalhos futuros pode-se escolher outros dados a serem coletados ou outros algoritmos de redução, mantendo a estrutura de funcionamento inalterada.

Nessa proposta, na camada FOG foram utilizados seis algoritmos para estabelecer algum processo de redução de dados. Esse algoritmos serão descritos a seguir.

O algoritmo **Baseline** trabalha com todos os dados coletados do ambiente de VANETS, sem redução dos dados. Utilizamos esse algoritmo para fins de comparação de acurácia e redução de rede com outras abordagens (SOUZA; VILLAS, 2016; SOUZA et al., 2016).

O processamento do **Baseline** ocorre na camada da FOG e consiste na recuperação e processamento do fluxo de dados contínuo recebido da camada de mobilidade, sem aplicar técnicas de redução. Os resultados podem ser usados como parâmetro para aferir a acurácia das estratégias de redução implementadas. O Algoritmo 1 ilustra o funcionamento desta abordagem.

O algoritmo **1to2** é baseado em amostragem simples, onde um a cada dois dados é sempre enviado, descartando metade dos dados gerados, ou seja, ele utiliza uma taxa de [1 : 2]. Para incorporar esse algoritmo na névoa, foi necessário adaptá-lo para lidar com fluxo de dados contínuo. Algoritmo 2 apresenta o funcionamento desse algoritmo.

Algorithm 1: Baseline

Require: Conjunto de pontos obtidos no fluxo de dados *Dados* contendo todas as informações posicionais do veículo no intervalo de tempo Δt

```

while ( $\Delta t == TamanhoJanela$ ) do
    buffer  $\leftarrow D$ 
end while
Dados  $\leftarrow buffer$ 
Inicia leitura dos dados
while ( $\neg Dados\ IsEmpty$ ) do
    beacon  $\leftarrow Dados$ 
end while
return beacon

```

O algoritmo **Random** (M. Tanizaki; O. Wolfso, 2007) utiliza um modelo de seleção aleatório para decidir se um beacon será selecionado ou não. O mecanismo de escolha está associado a uma função que seleciona os dados apenas quando uma certa probabilidade

Algorithm 2: 1-TO-2

Require: Conjunto de pontos obtidos no fluxo de dados *Dados* contendo todas as informações posicionais do veículo no intervalo de tempo Δt
 $pos = 0$
while ($\Delta t == TamanhoJanela$) **do**
 $buffer \leftarrow D$
end while
 $Dados \leftarrow buffer$
Inicia leitura dos dados
while ($!Dados IsEmpty$) **do**
 if ($pos == 2$) **then**
 $beacon \leftarrow Dados$
 $pos = 0$
 end if
 $pos+ = 1$
end while
return $beacon$

$p > 1$ é atingida. Para incorporar esse algoritmo na névoa, foi necessário adaptá-lo para lidar com fluxo de dados contínuo. Algoritmo 3 apresenta o funcionamento desse algoritmo.

Algorithm 3: Random

Require: Conjunto de pontos obtidos no fluxo de dados *Dados* contendo todas as informações posicionais do veículo no intervalo de tempo Δt
while ($\Delta t == TamanhoJanela$) **do**
 $buffer \leftarrow D$
end while
 $Dados \leftarrow buffer$
Inicia leitura dos dados
while ($rand[0..1]==True$) **do**
 $beacon \leftarrow Dados$
end while
return $beacon$

O algoritmo **Limite** proposto por (WAHID; RAO; GOEL, 2019), define um limite mínimo para selecionar dados com base na velocidade. Valores abaixo do *threshold* são descartadas, formando um subconjunto com um valor mínimo de velocidade estabelecidos.

Esse algoritmo permite que o limite seja definido como um parâmetro, levando em consideração o cenário desejado. No contexto veicular, essa definição pode ser complexa.

Portanto, para trabalhar essa questão, o processamento dos dados é realizado avaliando a velocidade registrada em cada *beacon*.

Assim como os outros algoritmos de redução, para incorporar esse algoritmo na névoa, foi necessário adaptá-lo para lidar com fluxo de dados contínuo. O Algoritmo 4 ilustra o funcionamento desta abordagem.

Algorithm 4: Limite

Require: Conjunto de pontos obtidos no fluxo de dados *Dados* contendo todas as informações posicionais do veículo
no intervalo de tempo Δt

Require: um parâmetro p que indica o limite mínimo para o *limite*.

limite = p

while ($\Delta t == \text{TamanhoJanela}$) **do**
 buffer $\leftarrow D$
end while

Dados $\leftarrow \text{buffer}$
Inicia leitura dos dados

while (*Dados.speed* $\geq \text{limite}$) **do**
 beacon $\leftarrow \text{Dados}$
end while

return *beacon*

O algoritmo **DBSCAN** utiliza agrupamento baseado em densidade. Esse algoritmo foi proposto por (ESTER et al., 1996) e é um método não paramétrico que, quando utilizado em um contexto veicular, agrupa veículos próximos uns dos outros em uma determinada rodovia ou mapa.

O DBSCAN utiliza dois parâmetros principais: ε , que representa a distância máxima entre dois veículos (vizinhança), e *minPts*, que descreve o número mínimo de veículos para formar um determinado *cluster*. Portanto, como nas abordagens baseadas em amostragem, apresentadas anteriormente, propõe-se neste trabalho, uma adaptação técnica e conceitual do algoritmo DBSCAN para definição de números de agrupamentos dentro de uma janela temporal, bem como na forma de trabalhar o fluxo de dados contínuo. O Algoritmo 5 ilustra o funcionamento detalhado desta abordagem.

Conforme apresentado anteriormente, a presente proposta utiliza o algoritmo **XMEANS**, pela diversidade de conjuntos de dados apresentados em um ambiente veicular, esse contexto impossibilita o uso de algoritmos de agrupamento com parametrização prévia, a exemplo do K-means.

O **XMEANS** é inicializado com 2-Means (K-Means com $k = 2$) como uma sub-rotina que tenta subdividir repetidamente os agrupamentos, mantendo as melhores divisões resultantes até que o critério *Bayesian Information Criterion* (BIC) (SCHWARZ, 1978) seja alcançado. De forma similar aos demais algoritmos abordados nessa pesquisa, o algoritmo XMEANS, baseado na métrica BIC, também foi adaptado para atuar em um contexto de fluxo de dados contínuo, sua estrutura é descrita no Algoritmo 6.

Algorithm 5: DBSCAN Adaptado: Redução de Dados em fluxo contínuo

Require: Conjunto de pontos obtidos no fluxo de dados (D) por Δt

Require: $minPts$, ϵ , $percent$

Ensure: agrupamentos (k e k') para cada Δt

```

while  $\Delta t == TamanhoJanela$  do
   $buffer \leftarrow D$ 
end while
 $D \leftarrow buffer$ 
Inicializa o conjunto de clusters  $k = 0$ 
for all novo ponto  $P \in D$  do
  if  $P$  is visited then
     $P_i = P_{i+1}$ 
  else
     $P = visited$ 
  end if
   $NhoodSet = \text{all } P \in \epsilon.neighborhood$ 
  if  $|NhoodSet| < minPts$  then
     $P = noise$  #classifica como ruído
  else
     $k_i = k_{i+1}$ 
     $k = P$ 
    for all  $P' \in NhoodSet$  do
      if  $P'$  is not visited then
         $P' = visited$ 
         $NhoodSet' = \text{all } P' \in \epsilon.neighborhood$ 
        if  $|NhoodSet'| \geq minPts$  then
           $NhoodSet = NhoodSet'$ 
        end if
      end if
    end for
    if  $P' \notin k \leftarrow (k = 0, \dots, k = n)$  then
       $k = P'$  #O ruído de um grupo pode ser absorvido por outro
    end if
  end for
end if
end for
#Seleciona um percentual  $p$  de cada grupo identificado
for all  $k \leftarrow (k = 0, \dots, k = n)$  do
   $\lambda = (\frac{p}{size(K)})$ 
   $k' = k(\lambda)$ 
end for
return ( $k'$ )

```

Algorithm 6: X-Means Adaptado: Redução de Dados em Fluxo Contínuo

Require: Conjunto de pontos obtidos no fluxo de dados (D) contendo todas as informações posicionais do veículo no intervalo de tempo Δt

Require: k

Require: Valor em percentual P dos elementos selecionados em cada *cluster*

Ensure: Conjunto de agrupamentos (k') atribuídos para cada Δt

```

while  $\Delta t$  do
   $buffer \leftarrow D$ 
end while
 $D \leftarrow buffer$ 
Inicializa o conjunto de clusters  $k = 2$ 
while  $\log M_j(D) - \frac{1}{2}kp_j \log n$  OR  $K > D$  do
  if  $BIC(j) == True$  then
     $K \leftarrow BIC(j)$ 
    for all novo ponto  $P \in K$  do
      if  $P$  is visited then
         $P_i = P_{i+1}$ 
      else
         $Update(P)$ 
         $P \leftarrow visited$ 
      end if
    end for
  else
     $Next(k)$ 
  end if
end while
for all  $k \leftarrow (k = 0, \dots, k = n)$  do
   $\lambda = (\frac{P}{size(K)})$ 
   $k' = k(\lambda)$ 
end for
return ( $k'$ )
  
```

Após a execução de 2-Means (K-Means com $k = 2$) a resultante será dois grupos que representam os focos de maior densidade veicular observados na primeira iteração. Em seguida, 2-Means é novamente executado em cada um dos agrupamentos formados na primeira iteração.

Após este processo, o algoritmo define dois centroides deste novo grupo com base no centroide do *cluster* original e compara com os elementos contidos no raio de cobertura. Se algum dos elementos do grupo inicial não tiver uma boa representação dos dados, de acordo com a métrica BIC, estes agrupamentos serão substituídos por seus “filhos”. Se nenhum *cluster* for pior do que seu filho, um membro do *cluster* é escolhido para

representar os dados com base nas métricas adotadas. O processo é repetido até que o número de *clusters* exceda o número máximo fornecido como entrada.

2.2 FLUXO DE DADOS ENTRE AS CAMADAS

Nesta pesquisa, os dispositivos emitem fluxos contínuos de dados que são recebidos por unidades de processamento RSUs. Os dados são reunidos em um fluxo unificado na camada subsequente. Um fluxo de dados (S) é uma sequência de tuplas ilimitadas e contínuas, geradas ao longo do tempo. Assim, $S = \{(S_1, \tau_1), (S_2, \tau_2), \dots\}$, onde S é uma sequência infinita de elementos que traz em sua estrutura diferentes informações de tráfego de veículos. O fator tempo é dado por $\tau \in R_+^* = \{x \in R | x \geq 0\}$. Conforme o tempo limite é alcançado, o conjunto de dados S é apresentado aos algoritmos de redução e de agrupamento para saída dos dados relevantes. Em simultaneidade, novos dados estão compondo uma janela temporal em fluxo de dados.

O resultado extraído dos algoritmos de redução ou agrupamento pode ser representado por uma série de segmentos $S' = \{p_1, p_2, \dots, p_{|S'|}\}$ que estão alinhados com o tamanho da janela temporal. Os elementos na janela deslizante seguem um método de escalonamento *first in, first out* (FIFO), que considera elementos do período atual (τ_i) até um período no passado (τ_{i-j}), onde i é o início e j o fim do intervalo de tempo Δt . Por fim o conjunto de dados S' é enviado para a camada de Nuvem para classificação de congestionamento de tráfego. Para o caso do **Baseline** o conjunto de dados S permanece inalterado e é enviado para a camada de Nuvem para classificação de congestionamento de tráfego.

O funcionamento abstrato do processamento de fluxo de dados contínuos realizado pelo framework desenvolvido nessa pesquisa é ilustrado na Figura 2.3, onde n_i representa a identificação de cada veículo no raio de cobertura de uma RSU.

O conjunto de veículos $\{n_1, n_2, \dots, n_i\}$ e o conjunto de RSUs $\{rsu_1, rsu_2, \dots, rsu_j\}$, que fazem parte da Camada VANET, estão em cinza na Figura 2.3. A névoa com a janela temporal em FIFO e as representações de amostragem e agrupamento estão em vermelho na mesma figura.

2.3 DEFINIÇÃO DO AMBIENTE DE SIMULAÇÃO

Para que a proposta seja validada, se faz necessária a realização de experimentos que contemplem as tecnologias estudadas. Trabalhar com cenários reais pode apresentar uma grande complexidade e custos elevados para a realização desses experimentos.

Diante disso, utilizaremos simuladores de software para desenvolver a estrutura prática do projeto. Esse tipo de abordagem experimental para avaliação da pesquisa é muito utilizada nos trabalhos de redes veiculares e no desenvolvimento de aplicações ITS.

Ao longo da revisão da literatura, buscamos identificar alguns simuladores que possuem ampla utilização em trabalhos de mobilidade urbana, que tem documentação disponível, que possam simular o tráfego de rede e que são open-source.

Os experimentos foram implementados usando três simuladores distintos que operam em conjunto.

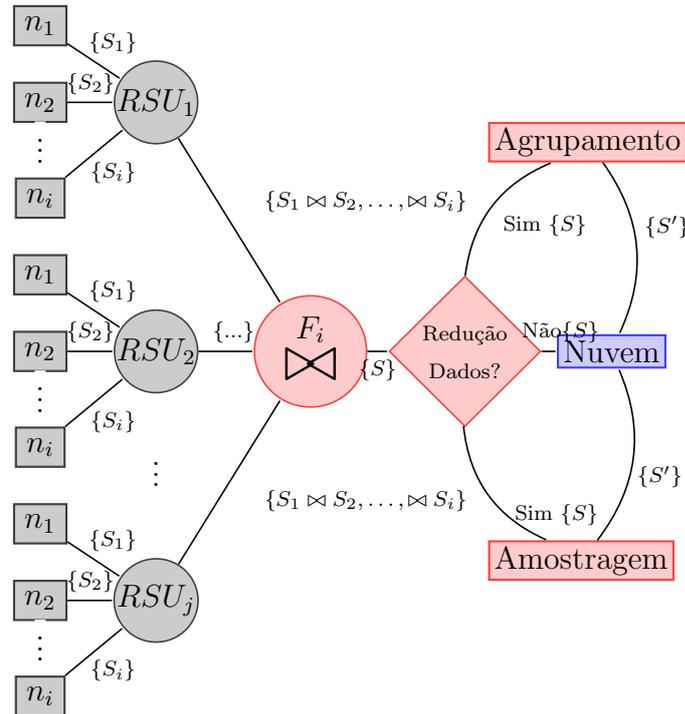


Figura 2.3: Fluxo de Dados Contínuo no *Framework*

O VEINS ¹ versão 4.7 é um Framework de código aberto que trata da comunicação entre veículos e a RSU, além da utilização do protocolo 802.11p.

A parte de Mobilidade Urbana é simulada pelo SUMO ² (*Simulation of Urban Mobility*) versão 0.32, que lida com as rodovias, semáforos e todos os elementos rodoviários do projeto.

O OMNET++ ³ versão 5.3, é um simulador que lida com os aspectos gerais de rede e conectividade. Tem suporte a protocolos de rede, rede de sensores e redes ad hoc.

Esses simuladores trabalham de forma integrada e foram adotados na presente proposta. Essa integração possibilita a verificação das questões relacionadas a redes veiculares, tráfego de rede e mobilidade urbana.

2.4 FLUXO DE EXECUÇÃO DO FRAMEWORK

Além da definição do cenário da solução, estabelecemos o processo de implementação da proposta. Com o objetivo melhorar o entendimento da proposta como um todo, a Figura 2.4 apresenta um fluxograma de implementação da solução. A seguir descreveremos com mais detalhes as camadas, seu funcionamento e os elementos contidos em sua estrutura.

Observando a imagem 2.4, pode-se verificar que, após o início da simulação, começa o fluxo de tráfego, e as mensagens do tipo *beacons* são enviadas em uma frequência contínua de 1Hz pelos veículos contidos em uma rede veicular. Esse mecanismo se baseia

¹<https://veins.car2x.org/>

²<http://sumo.dlr.de/index.html>

³<https://omnetpp.org/>

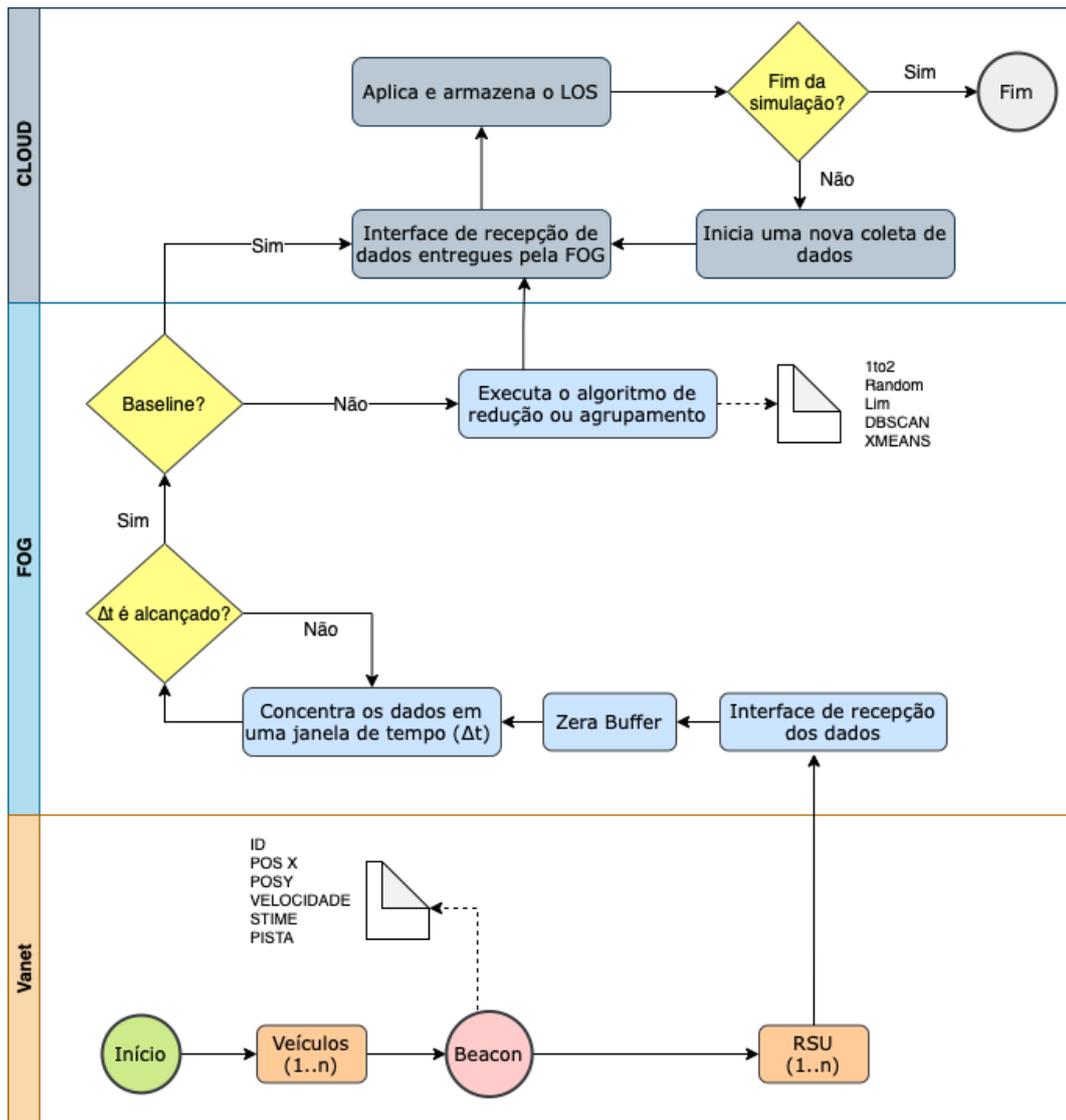


Figura 2.4: Fluxo de execução do Framework.

no trabalho de (Schmidt et al., 2010), que sugere o envio de mensagens pelos veículos a uma taxa de 1Hz, o que é considerado o mínimo para um monitoramento eficiente de pontos de retenção ou congestionamentos.

Vale destacar que os veículos enviam as mensagens independente da garantia de recepção por outros equipamentos. Isso ocorre porque os veículos não sabem se estão sob área de cobertura ou se os beacons estão sendo capturados por uma RSU. Nesse ponto, é importante destacar que o envio em uma alta frequência ou por muitos veículos retidos pode sobrecarregar a rede.

Nesse contexto, as *Road Side Units* (RSU) são responsáveis por encaminhar os fluxos de dados recebidos pelos diferentes veículos. O protocolo IEEE 802.11p, tecnologia de rádio de comunicação de curto alcance (Xinzhou Wu et al., 2013), foi utilizado como forma de viabilizar a trocar dados entre veículos e as unidades de processamento RSU.

Cada unidade de processamento está conectada a uma interface de entrada e saída que controla o fluxo de dados que chega ou sai da camada de mobilidade. Os diferentes fluxos de dados recebidos na RSU são emitidos para uma interface que é responsável por conduzir esses conjuntos às camadas adjacentes do *Framework*. A camada de mobilidade é responsável pela geração de dados em fluxo até que não haja mais dados, encerrando assim a simulação.

A camada de Névoa (FOG) atua como receptora e fornece estratégias de redução de fluxo com base na densidade e distribuição dos veículos nas rodovias. Esses fluxos de dados que são provenientes de diferentes RSUs, são unificados em um único conjunto de dados antes de serem processados. Isso é feito através do uso de uma janela temporal, que estabelece limites flexíveis para o processamento contínuo dos dados.

A janela temporal garante a seleção de um conjunto finito e variável de tuplas que são acumuladas durante um intervalo de tempo específico, como citado anteriormente (TOLPIN, 2016) e (PATROUMPAS; SELLIS, 2006). Ou seja, essas tuplas são encaminhadas para processamento em um intervalo de tempo pré definido.

Ainda observando a Figura 2.4, pode-se verificar que a Fog que inicia uma concentração de informações dentro de um tempo definido (Δt). Quando a janela de tempo é alcançada, ou seja, quando o *tempo_decorrido* > *tempo_limite_janela*, o framework verifica se o algoritmo que está em execução é o **Baseline**, caso positivo o buffer de informações é enviado diretamente para a camada de Cloud. Caso não seja o algoritmo **Baseline** que está em operação, o *buffer* de informações é enviado para o algoritmo de redução ou agrupamento de dados, e o resultado dessa redução é enviado para a camada de Cloud.

O funcionamento detalhado dos algoritmos contidos na *layer* FOG foram apresentados na Seção 2.1 deste trabalho.

Na presente proposta, a camada de computação em nuvem (Cloud) visa unificar o fluxo de dados obtidos pelos diferentes pontos da borda da rede, classificação do fluxo de tráfego com base no modelo LOS e predição das condições de tráfego com base na série histórica do panorama de trânsito com base no modelo ARIMA.

Analizando a Figura 2.4 a camada da Cloud recebe os dados resultantes da FOG e aplica o modelo de classificação do congestionamento de veículos (LOS) e armazena essa informação. A definição e funcionamento do LOS foram apresentados na introdução do presente capítulo.

A medida que os conjuntos de dados são entregues pela névoa, o mecanismo de integração dos dados é iniciado, em busca da criação de uma visão integrada do fluxo de tráfego dinâmica e atualizada. A visão que esse modelo entrega, favorece a tomada de decisão em tempo real, considerando que toda a conjuntura do tráfego está apresentada nessa visualização. Em tempo, destacamos que, por estar em um ambiente dotado de potencial ilimitado de armazenamento e processamento, podemos manter as séries históricas dos dados para a aplicação de algoritmos de predição de dados, como o ARIMA utilizado nesse trabalho.

Em última análise da Figura 2.4, o processo de concentração dos dados na RSU no ambiente veicular, recebimento de um volume de dados contínuos e dentro de uma janela temporal pela névoa, aplicação do processo de redução de dados e a posterior entrega da

informação relevante para a nuvem, se repete até o fim da simulação.

Os dados coletados pelos veículos e os algoritmos de redução e de agrupamento utilizados, estão destacados para demonstrar que não são determinantes para o funcionamento do *Framework*, ou seja, por se tratar de um modelo, podem ser acrescidos ou substituídos sem afetar o funcionamento da solução. Assim como esse modelo é desenhado para trabalhar com um eventual aumento exponencial de geração de dados no ambiente veicular, por exemplo a inclusão de Veículos Autônomos Conectados (CAV).

No próximo capítulo, serão apresentados os resultados das análises dos algoritmos selecionados em comparação com o algoritmo **Baseline**. Essa métrica permitirá comparar as estratégias de identificação de congestionamentos e predição de dados, uma abordagem comumente usada para analisar as variações nas diferentes abordagens implementadas nesta pesquisa (SOUZA; VILLAS, 2016; SOUZA et al., 2016).

RESULTADOS

A avaliação da proposta será realizada através de experimentos que visam comparar uma aplicação ITS que identifica congestionamento. Essas avaliações e ponderações se baseiam na análise de seis algoritmos, dos quais dois são baseados em agrupamento, e três utilizam uma abordagem baseada em seleção de amostragem e um algoritmo *Baseline* utilizado como balizador nas medições de acurácia.

Entre os algoritmos de agrupamento estão o **X-Means** (PELLEG; MOORE, 2000) e **DBSCAN** (ESTER et al., 1996). Nas análises realizadas, buscamos a ótica de desempenho na redução de dados e também na capacidade desses algoritmos determinarem níveis significativos de acurácia em comparação ao conjunto original dos dados. As estratégias baseadas em agrupamento foram comparadas com outras abordagens descritas na literatura como **1to2**, **Rand** (TANIZAKI; WOLFSON, 2007) e **Lim** (WAHID; RAO; GOEL, 2019). Essa tarefa foi realizada com o objetivo de posicionar esta proposta frente a outras abordagens discutidas pela comunidade científica.

Na prática, essa comparação foi feita entre o Framework utilizando todos os dados enviados da camada de rede veicular (**Baseline**) (SOUZA; VILLAS, 2016), (SOUZA et al., 2016, 2018) e o mesmo Framework utilizando algoritmos que reduzem os dados trafegados para a camada que toma a decisão de classificação de trânsito. Essa análise foi de natureza quantitativa e, como mencionado anteriormente, a comparação, focada no desempenho, pois avaliou o nível de acurácia com a economia de uso de rede de todas as técnicas utilizadas com relação ao **Baseline**.

Esses experimentos foram realizados nos simuladores apresentados na Seção 2.3 gerando um ambiente virtual de redes veiculares. Cabe reiterar que, nesse contexto, os veículos emitem *beacons* que são capturados por uma *Roadside Unit* (RSU) e encaminhados para processamento na camada de névoa. Nessa camada, os *beacons* são processados e as estratégias de redução de dados propostas na pesquisa são implementadas sobre eles.

Nos experimentos, foi convencionada a existência de um link de rede entre a névoa e a nuvem com uma taxa máxima de 340 (KB/s). É importante destacar que, no cenário proposto para os experimentos, foi utilizada uma *Roadside Unit* a disposição de todo o

mapa, mas esse modelo pode ser alterado através da utilização de diferentes instâncias do *framework* para cada RSU.

Para a implementação do cenário foi utilizado o mapa da cidade de Colônia, na Alemanha obtido em OpenStreetMap ¹, conforme ilustrado na Figura 3.1.

Esse mapa foi utilizado através do TAPASCologne ² (UPPOOR et al., 2013), que se trata de um projeto open-source, muito utilizado por trabalhos relacionados a redes veiculares, que fornece um grande conjunto de mobilidade urbana baseada no SUMO para elevar o realismo da simulação do ambiente de ITS.

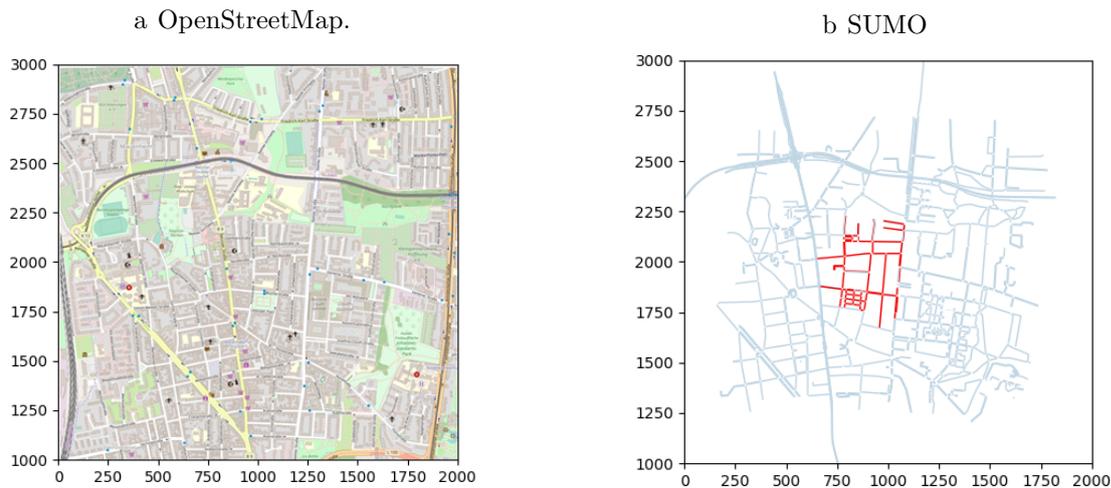


Figura 3.1: Mapa da cidade de Cologne na Alemanha.

Para realizar as simulações, será utilizada a região mais central do conjunto de dados, pois apresenta uma densidade veicular maior. Essa região será apresentada em vermelho na Figura 3.1(b).

Na Figura 3.1(a), pode ser visualizada a região do mapa selecionada no *OpenStreetMap*³, com área de simulação de 4,6 Km. Esse mapa foi exportado do *OpenStreetMap* em um arquivo com extensão .osm e posteriormente convertido para .xml, para ser importado pelo SUMO. O mapa inserido no SUMO pode ser visualizado na Figura 3.1(b).

Os veículos contidos nesse cenário, seguem as características de mobilidade urbana com base no trânsito dessa cidade. Foram utilizadas as densidades de, 30, 50 e 70 veículos ativos no mapa durante o tempo de simulação. Esses carros assumem diferentes rotas e tempo de permanência no mapa a cada mudança de semente de aleatoriedade de geração inicial da simulação.

Neste cenário, cabe ressaltar que foi realizado um conjunto de experimentos com 20 replicações para 5 sementes de aleatoriedade para cada algoritmo, utilizando um intervalo de confiança de 95%. As quantidades de replicações dos experimentos, são realizadas de modo a suavizar grandes variações nos resultados finais.

¹<https://www.openstreetmap.org>

²<https://sumo.dlr.de/wiki/Data/Scenarios/TAPASCologne>

³<https://www.openstreetmap.org>

Os níveis de acurácia foram coletados por meio da classificação do nível de serviço (LOS), velocidade média dos veículos e pelos impactos das estratégias de redução de dados no consumo de recurso de rede para cada uma das técnicas analisadas dentro do *Framework*.

3.0.1 Metodologia do cenário e da Simulação

Com base no resultado dos experimentos realizados, podemos analisar inicialmente o comportamento da velocidade média em km/h ao longo do tempo de simulação. Por conta da variação dos tempos de simulação entre os experimentos, se fez necessária a aplicação de uma normalização temporal em todas as estratégias de redução de dados contidas nessa proposta. Esse mecanismo visa entregar uma fidelidade nas análises comportamentais dos veículos.

Vale reiterar que em todos os experimentos realizados, as densidades foram variadas entre 30, 50 e 70 veículos dispostos no cenário de mobilidade. Os veículos usados na simulação desfrutaram das mesmas características, como tamanho, média de velocidade e desvio padrão.

Como mencionado, a Figura 3.1 apresenta o mapa de Cologne na Alemanha, e destaca em vermelho a região de atuação da simulação. A escolha por esse recorte da região aconteceu em função da área apresentar maior densidade veicular, vias longas e gargalos que poderiam contribuir para a geração de cenários mais realistas.

E visando entender o padrão de ação dos veículos nas rodovias, mais especificamente em pontos que contém características marcantes de trânsito, como cruzamentos, fluxo intenso da via e sentido único do trânsito. Foram definidos, segmentos de pistas específicas nesse mapa. Conforme pode ser observado na Figura 3.2, cada trecho de rodovia recebeu um número de referência (1, 2 e 3) que representa os respectivos segmentos. Esta numeração será referência durante as próximas análises.

Os trechos escolhidos para a realização dos experimentos foram definidos com base em diferentes particularidades, com o objetivo de manter uma fidelidade maior com cenários reais. Portanto, o **Trecho 1** foi escolhido porque é a via mais longa do mapa, com muitas intersecções e também por ser uma via de acesso rápido para outras vias do cenário; o **Trecho 2** foi escolhido devido ao fato de ser uma via alternativa a intersecção para a via principal de trânsito rápido, é uma pista unidirecional e possui um gargalo com propensão a retenções de tráfego. O **Trecho 3** demonstra uma artéria urbana densa, que promove acesso aos veículos ao ponto central do cenário. Essa característica tende a causar retenção nesse trecho, bem como a propagação da retenção para as vias vizinhas.

Além da definição do mapa e dos trechos de análise, destacamos que todas as simulações seguiram os parâmetros apresentados na Tabela 3.1.

3.0.2 Análise das velocidades dos experimentos

Visando definir um panorama geral de análise das variações e tendências das velocidades dos veículos ao longo do processo de simulação, temos as Figuras 3.3, 3.4 e 3.5 que apresentam este comportamento nos três trechos de rodovias predefinidos com as três variações de densidades veiculares listadas.

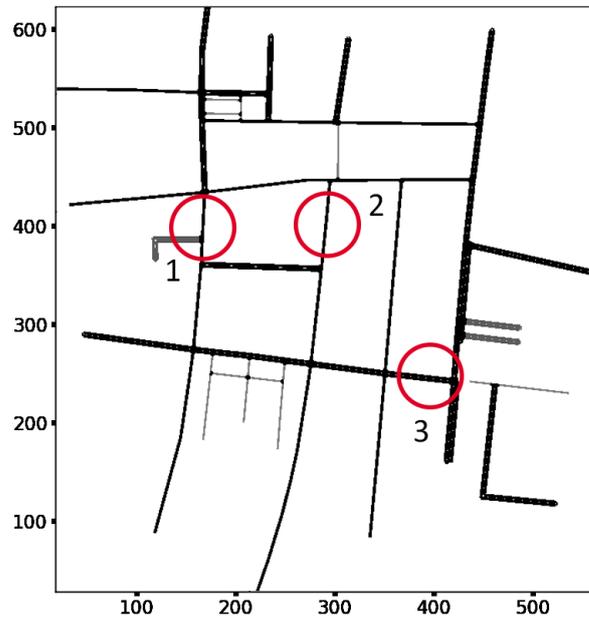


Figura 3.2: Trechos utilizados com base para análise dos resultados.

Tabela 3.1: Parâmetros de simulação

Parâmetros	Valores
Área de Simulação	4.6 km ²
Mapa	TAPASCologne (UPPOOR et al., 2013)
Número de veículos	{30, 50 e 70}
Protocolo de comunicação	IEEE 802.11p
Potência de transmissão	20 mW
Taxa de bits	6 Mbps
Frequência (Beacons)	1Hz

A Figura 3.3 descreve um comportamento semelhante para os algoritmos **1to2**, **Rand** e **Baseline**, mostrando que as linhas de velocidade estão praticamente sobrepostas ao longo do tempo. Pode-se observar ainda, que há uma variação sensível na tendência da velocidade para o algoritmo **Limite** e **XMEANS**, essa variância pode ser oriunda do comportamento natural desses algoritmos para densidades veiculares mais baixas. O algoritmo de amostragem envia poucos dados por conta de não atingir sempre o limite para envio dos dados e o algoritmo de agrupamento forma poucos grupos nesse contexto, afetando um pouco o grau de representatividade dos dados.

Para densidades veiculares menores este comportamento pode ser explicado pelo curto período de tempo em que a estrada se encontrava congestionada. Os picos de congestionamento de tráfego de curto período podem não ser detectados com alta precisão pelos algoritmos de agrupamento, uma vez que esse intervalo pode não ser suficiente para ter

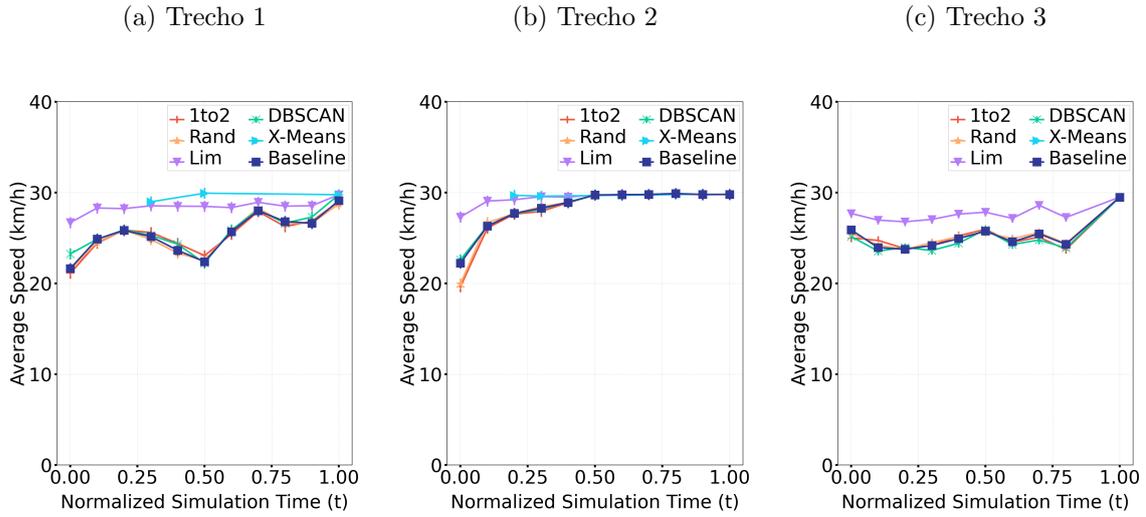


Figura 3.3: Análise da Velocidade para 30 veículos.

todos os dados de tráfego coletados e caracterizar um cenário de alta densidade espacial.

Em tempo, é importante destacar que para todas as circunstâncias verificadas, o algoritmo **Limite** apresenta menor acurácia frente ao *Baseline*. Essa diferença na acurácia pode estar relacionada ao limite estabelecido para a transmissão de dados de tráfego. Considerando isso, ambientes com alta variação de velocidade podem causar certa redução nos níveis de acurácia. Uma forma de mitigar este problema pode ser a utilização de um limite obtido a partir da média de dados de tráfego históricos do mesmo local.

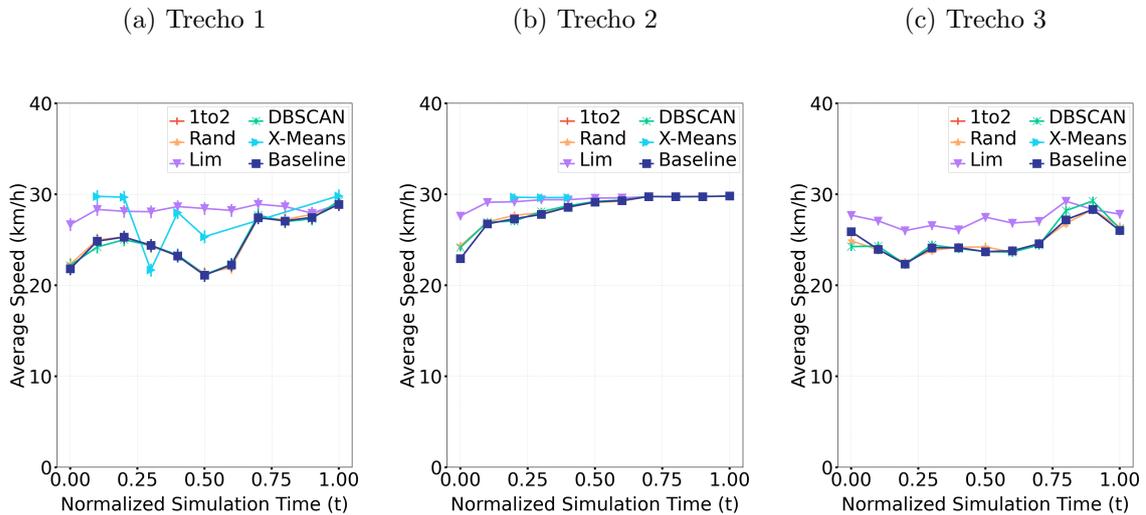


Figura 3.4: Análise da Velocidade para 50 veículos.

Com exceção do algoritmo **Limite** e o **XMEANS**, todos os outros algoritmos ilustrados na Figura 3.4 também mantiveram justo alinhamento no que se refere ao com-

portamento do algoritmo *Baseline*. A divergência na tendência de velocidade desses dois algoritmos nos cenários apresentados se devem aos motivos supracitados.

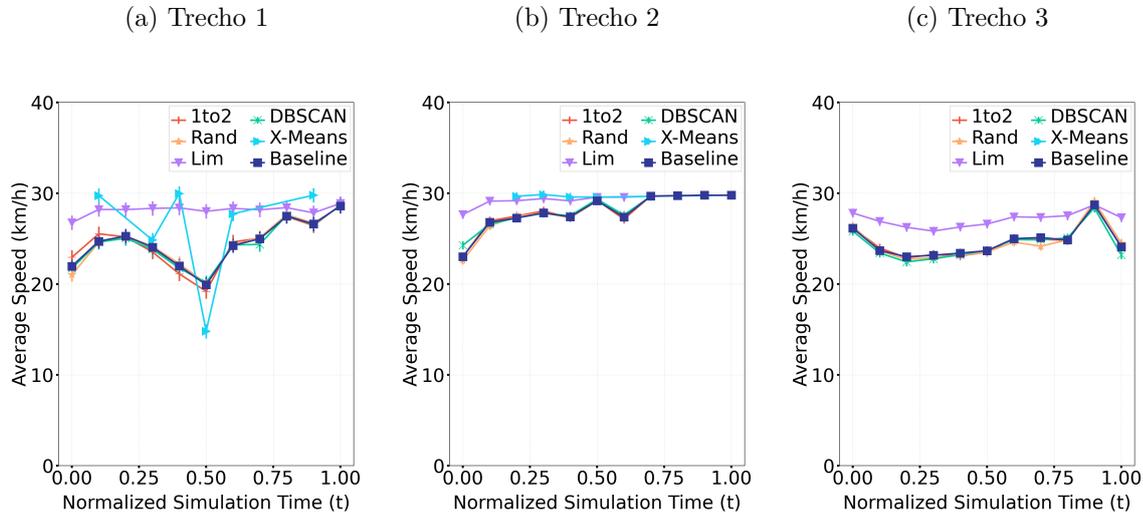


Figura 3.5: Análise da Velocidade para 70 veículos.

Analisando a Figura 3.5, podemos observar que apesar de uma variação de velocidade o algoritmo de agrupamento **DBSCAN** segue a tendência de velocidade do **Baseline**. Já o algoritmo de agrupamento **XMEANS** não apresenta bons resultados mesmo quando a densidade veicular aumenta para 50 e 70, isso ocorre devido ao pequeno número de *clusters* criados durante o processo de simulação e baixa representatividade da amostra em função do baixo número de elementos selecionados.

Os algoritmos **DBSCAN** e **X-Means** funcionam nesse projeto destacando um percentual de elementos obtidos a partir de cada *cluster* formado para proceder com os mecanismos de redução de dados. Essa separação, tem o foco de trazer mais chances de redução de dados, porém em certas circunstâncias, pode não apresentar uma amostra tão representativa para garantir o acompanhamento contínuo do fluxo de dados ao longo dos experimentos. Este cenário fica mais crítico quando o volume de dados analisados é reduzido. Este desafio pode ser superado com a definição de técnicas de seleção capazes de fornecer um conjunto resultante mais representativo frente ao conjunto de dados original.

O estudo de velocidade é fundamental para demonstrar o padrão de ação dos dados de tráfego que cada algoritmo da camada da névoa fornece para a nuvem. Com este tipo de informação de tráfego, o *Framework* proposto pode ser capaz de detectar a fonte e classificar o nível de congestionamento de acordo com sua severidade. Para realizar esta classificação utilizamos a definição do Nível de Serviço (LOS).

3.0.3 Análise do LOS dos experimentos

Conforme apresentado anteriormente, a classificação por nível de serviço (LOS) é um modelo que categoriza o fluxo de tráfego em cada segmento da rodovia com base em

fatores como velocidade, densidade e atraso de veículos durante as viagens. Isso permite definir o estado do fluxo de tráfego em cada segmento da rodovia.

A seguir, as Figuras 3.6, 3.7 e 3.8 ilustram os resultados relativos à aplicação do nível de serviço para os mesmos algoritmos, trechos e densidades veiculares destacadas nesta etapa do experimento.

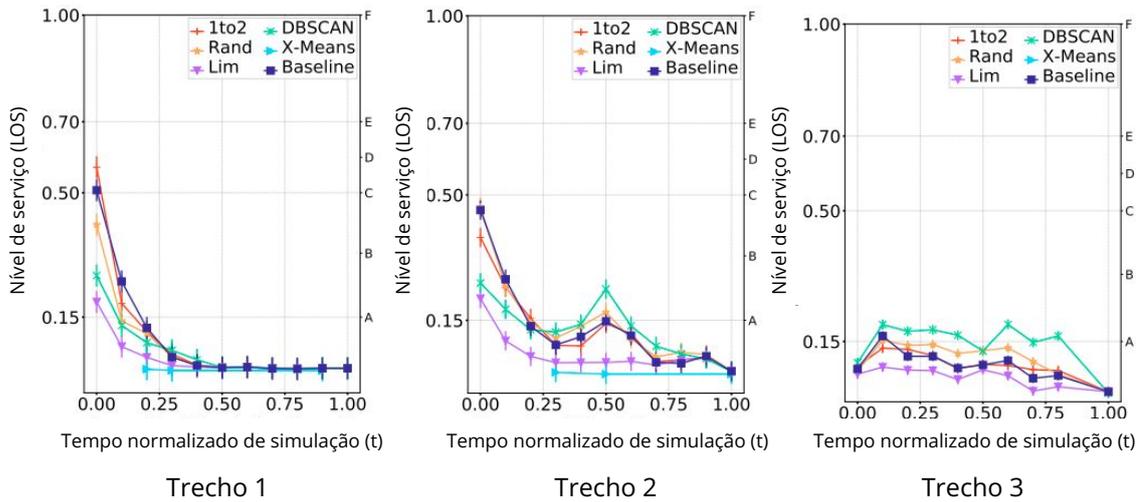


Figura 3.6: Análise do LOS para 30 veículos.

A Figura 3.6 mostra o comportamento do LOS para uma densidade de 30 veículos. Podemos verificar que **Baseline**, **1to2** e **Rand** tem uma similaridade de tendência de LOS. Adicionalmente, podemos ver que no Trecho 1, todos os algoritmos tiveram comportamento parecido, apresentando alta precisão. Nesse contexto, o LOS diminui em função da baixa densidade de veículos. Por outro lado, o trecho 2 apresenta uma perda de acurácia para os algoritmos **XMEANS** e **Limite**, o que ocorre também no trecho 3.

Esses trechos possuem um fluxo de tráfego naturalmente maior e com a densidade de 30 veículos por km^2 , pode gerar picos de congestionamento de tráfego de curto período, variando muito o limite de velocidade afetando a geração de dados do algoritmo **Limite** e criando poucos grupos na janela temporal para o caso do algoritmo **XMEANS**. Essa conjuntura pode afetar diretamente a acurácia dos algoritmos citados.

Ainda na Figura 3.6, podemos observar uma ligeira perda de acurácia para o algoritmo de agrupamento **DBSCAN** em todos os trechos, porém com a manutenção bem próxima da tendência de LOS com relação ao **Baseline**, inclusive ficando na maior parte do tempo, na mesma escala de classificação. Este fenômeno pode ser explicado pela redução na quantidade de dados disponíveis para análise nos momentos finais do experimento nessa densidade veicular, gerando menos grupos e afetando o grau de representatividade da porção desses *clusters*.

A Figura 3.7 apresenta o comportamento dos algoritmos quando aumentamos o número de veículos nas estradas para 50 veículos por km^2 . Dentro deste cenário, enquanto o algoritmo **Limite** mantém uma menor capacidade de detecção de congestionamento, observamos uma melhora na precisão do **XMEANS**, em comparação com Figura 3.6. Isso

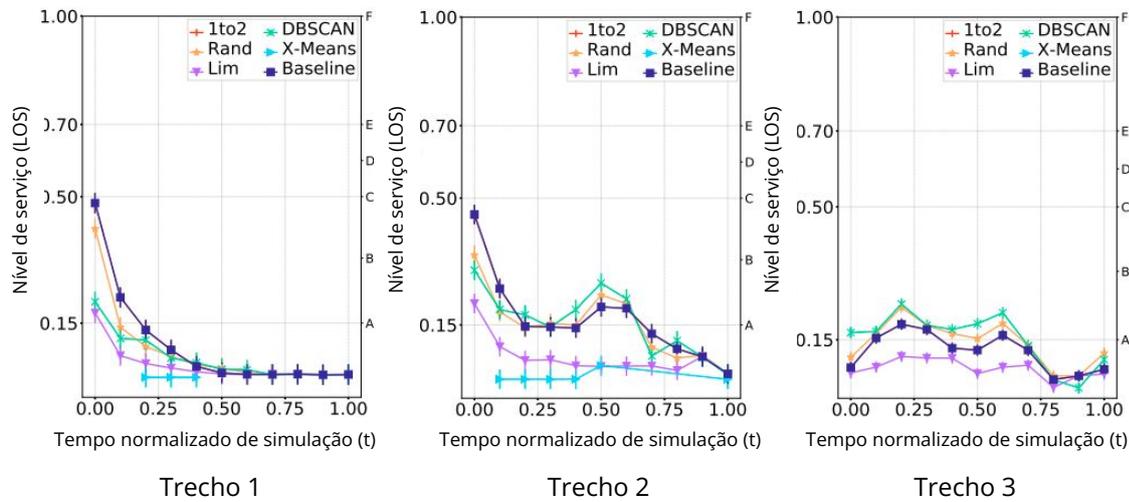


Figura 3.7: Análise do LOS para 50 veículos.

se deve principalmente ao aumento no número de pontos de dados, o que resulta em mais *clusters* no mapa. Este aumento na precisão também é observado para o **DBSCAN**. De modo geral, conseguimos observar uma tendência de comportamento nos trechos para as densidades de 30 e 50 veículos.

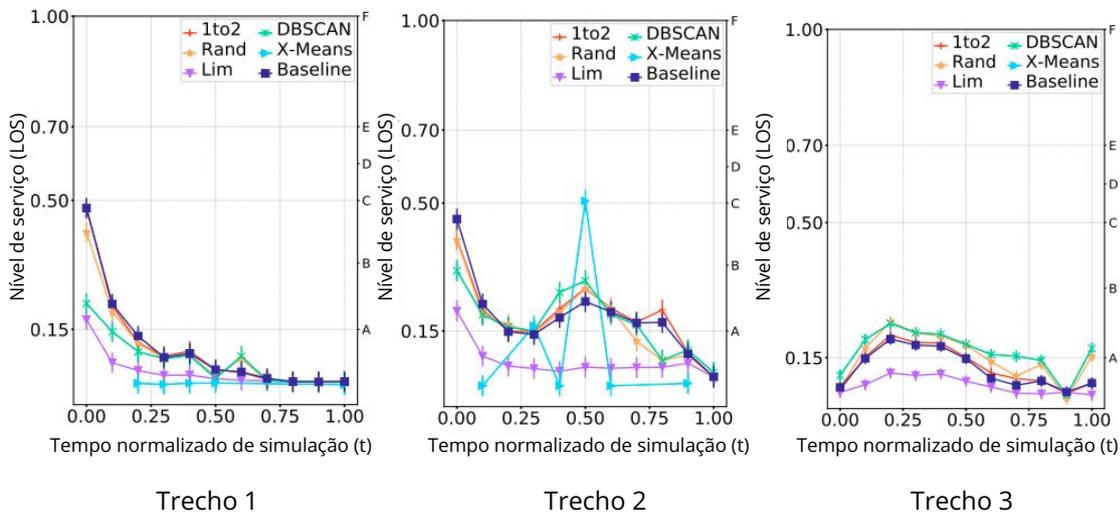


Figura 3.8: Análise do LOS para 70 veículos.

Na Figura 3.8 analisamos o tráfego de 70 veículos por km^2 e podemos observar um aumento no congestionamento. Neste cenário, o algoritmo **Limite** mostrou uma maior eficiência na detecção de congestionamento do que os cenários anteriores, porém ainda com menos acurácia com relação ao algoritmo **Baseline**. Isso sugere que a aplicação dessa técnica tende a ser mais eficiente em situações com maior concentração de veículos. Embora o **XMEANS** tenha apresentado uma ligeira melhoria para este cenário, é o al-

goritmo **DBSCAN** que mantém a alta precisão e consistência com relação a classificação do LOS, mostrando com precisão os pontos e os níveis de congestionamento.

Analisando as três Figuras 3.6, 3.7 e 3.8, podemos ver que os algoritmos **Limite** e **XMEANS** possuem menor representatividade da classificação de congestionamento LOS. Podemos observar que o algoritmo **Limite** não possui todos os dados de tráfego necessários para calcular com precisão o LOS.

No algoritmo **XMEANS**, apenas uma porção dos dados de tráfego de cada *cluster* são enviados da névoa para a Nuvem, levando a uma eventual queda na representatividade dos dados e a consequente falta de informações para classificação do fluxo de tráfego. Importante destacar que mesmo quando o **XMEANS** possui um LOS menor em relação ao Baseline, ele ainda consegue identificar alguns pontos de congestionamento.

A análise do LOS é essencial para garantir a classificação do fluxo de tráfego e determinar a acurácia dos algoritmos em funcionamento no modelo. Considerando essa importância, promovemos uma análise adicional aplicando a classificação por nível de serviço (LOS) em todo o cenário de simulação.

Cabe destacar que os experimentos nesta seção visam medir a precisão das técnicas de classificação, tanto para posicionar a proposta atual em relação a outros estudos analisados quanto para estabelecer uma base confiável de comparação para o uso dessas técnicas em diferentes cenários de computação urbana.

Conforme mencionado anteriormente, a classificação de congestionamento de trânsito (LOS) permite definir, a partir de um modelo de classificação, o estado do fluxo de tráfego em cada trecho da rodovia. Esta análise é capaz de categorizar o nível de fluxo de tráfego com base em fatores como velocidade, densidade e atraso de veículos durante as viagens nas estradas.

Para ajudar no entendimento das próximas visualizações, a Tabela 3.2 de descrição da classificação das condições do tráfego (LOS) será novamente apresentada, porém com a inclusão de cores de referência para facilitar a associação.

Tabela 3.2: Classificação (w_{ij}) das Condições de Tráfego

Peso da Rodovia (w_{ij})	LOS	Fluxo de Tráfego
[0, 0.15]	A ●	Free-flow
[0.15, 0.33]	B ●	Reasonably free-flow
[0.33, 0.50]	C ●	Stable-flow
[0.50, 0.60]	D ●	Approaching unstable-flow
[0.60, 0.70]	E ●	Unstable-flow
[0.70, 1.00]	F ●	Breakdown-flow

As figuras 3.9, 3.10 e 3.11 apresentam o resultado de um conjunto de experimentos nos quais os mapas são populados com diferentes densidades de veículos. Nestes experimentos, as densidades variam entre 30, 50 e 70 veículos, respectivamente.

Podemos observar na Figura 3.9 os resultados da aplicação da abordagem de classificação por nível de serviço LOS utilizando uma densidade de 30 veículos no cenário. Nessa conjuntura, verifica-se que a aplicação da métrica LOS nos algoritmos *Baseline*, **1to2** e **Rand** mantém comportamentos semelhantes, demonstrando, na maioria dos casos, similaridade entre os pontos, tendências, além de manter os mesmos níveis de identificação de congestionamento de tráfego. Podemos notar que o **DBSCAN** consegue identificar muitas variações de densidade quando comparado com o **Baseline**, identificando a maior parte dos pontos de congestionamento formados nas rodovias analisadas. O **Limite** apresenta uma perda significativa de acurácia e o **XMEANS** nesse contexto não teve uma acurácia representativa.

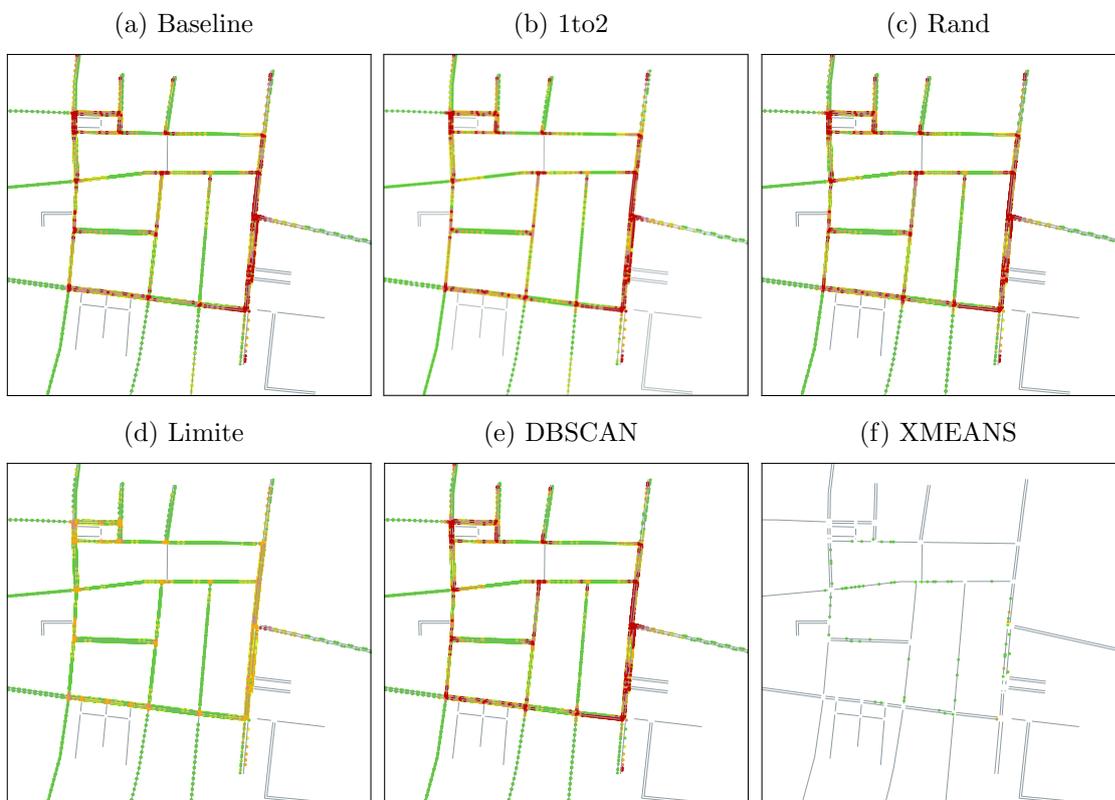


Figura 3.9: Detecção de congestionamento de tráfego com densidade de 30 veículos/ km^2

Já a Figura 3.10 ilustra o comportamento dos algoritmos quando a densidade veicular é aumentada para 50 veículos por km^2 . Nesse novo cenário, o algoritmo **Limite** ainda tem uma capacidade de detecção de congestionamento inferior aos demais algoritmos de amostragem, e há um aumento na acurácia na detecção de pontos de retenção pelo algoritmo de agrupamento **DBSCAN**. Esta melhoria se deve ao aumento na quantidade de dados, o que torna a amostra mais representativa em relação aos experimentos envolvendo 30 veículos, analisados na Figura 3.9. Apesar do algoritmo **XMEANS** ter apresentado uma melhora com o aumento da densidade veicular, ele ainda não teve uma acurácia significativa nesse modelo.

Trabalhando com o fluxo de tráfego de 70 veículos por km^2 , como pode ser visto na

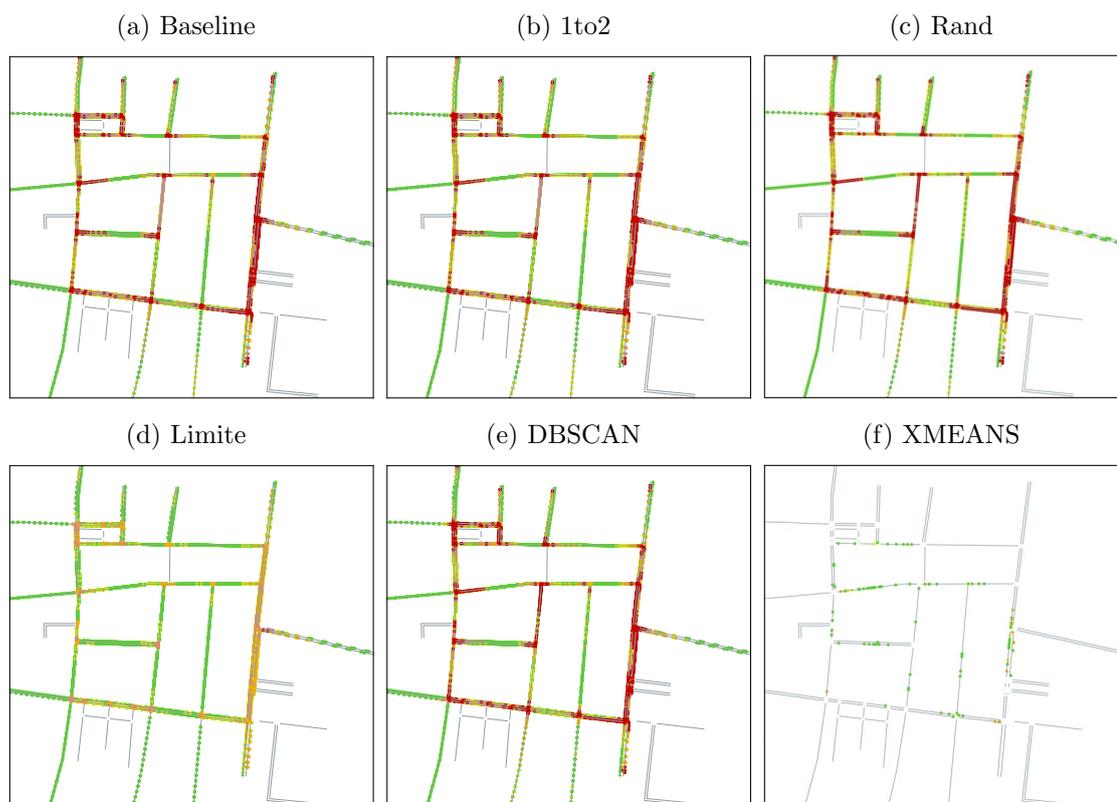


Figura 3.10: Detecção de congestionamento de tráfego com densidade de 50 veículos/ km^2

Figura 3.11, podemos perceber uma grande formação de congestionamento e consequentemente a maior presença dos pesos E e F, do modelo de classificação de congestionamento de tráfego (LOS). Esta classificação pode ser verificada na tabela 3.2.

Nesse cenário, o algoritmo **Limite** demonstrou uma singela melhora na detecção de congestionamento do que os cenários anteriores. O aumento de desempenho sugere que a aplicação dessa técnica pode ser mais eficiente em situações de tráfego intenso com muitos veículos. O algoritmo **DBSCAN** manteve níveis de precisão acurados em todas as variações de densidade, mostrando com exatidão os pontos de retenção e congestionamento nas rodovias. Apesar do algoritmo **XMEANS** ter apresentado mais uma melhora com o novo aumento da densidade veicular, ele ainda não teve uma acurácia significativa com relação ao modelo.

Analisando as três Figuras 3.9, 3.10 e 3.11, simultaneamente, podemos observar que o algoritmo **XMEANS** apresentou a menor representatividade em relação ao nível de serviço LOS em comparação com os outros. Isso pode ser explicado pela baixa aquisição de dados de tráfego, considerando que naturalmente esse algoritmo envia um percentual dos elementos contidos nos *clusters*. Esta baixa representatividade frente ao conjunto de dados original prejudica a aplicação adequada da análise, classificação e uso deste conjunto de dados na tomada de decisão.

Por outro lado, o algoritmo **DBSCAN** que se utiliza de uma estratégia de agrupamento baseada em densidade, manteve significativa representatividade ao longo de todos

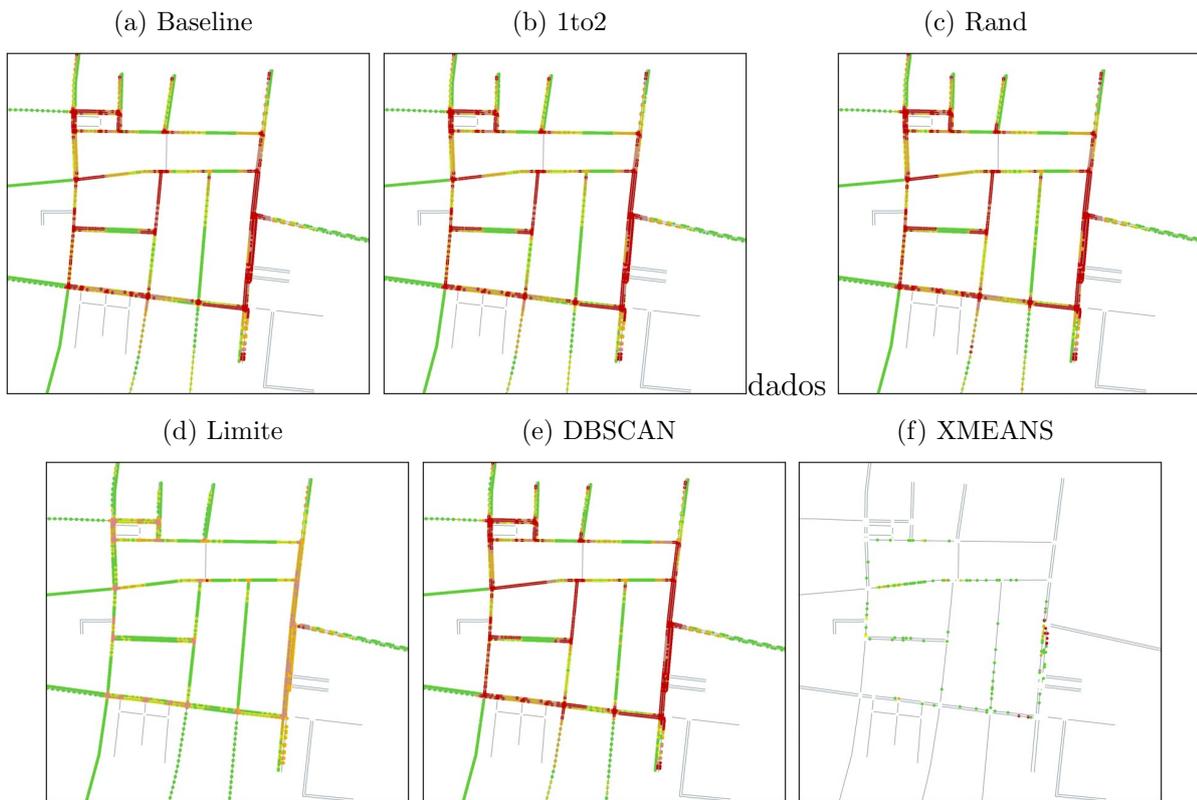


Figura 3.11: Detecção de congestionamento de tráfego com densidade de 70 veículos/ km^2

os experimentos. Ficando bem similar ao **Baseline** no que diz respeito a acurácia de classificação de congestionamento de tráfego.

Importante destacar que este trabalho busca, entre outros aspectos, o uso de algoritmos de agrupamento como forma de garantir a redução de dados mantendo níveis de acurácia satisfatórios. Por se tratar de uma pesquisa e um modelo de trabalho baseado em camadas com um nível de desacoplamento, a verificação que um determinado algoritmo funciona melhor com mais densidade veicular de que outro, pode gerar uma grande contribuição para entendimento do *framework* e para trabalhos futuros.

A seguir vamos realizar as análises de velocidade e LOS para predição de séries temporais utilizando o algoritmo Autoregressive Integrated Moving Average (ARIMA).

3.0.4 Previsão de dados com a utilização do ARIMA

Conforme mencionado anteriormente o modelo ARIMA é amplamente utilizado na previsão de séries temporais. Ele combina a autocorrelação, que analisa a dependência dos valores passados, e médias móveis, que suavizam as variações. Com base nesses recursos, o ARIMA é capaz de fazer previsões futuras com base nos dados históricos.

O presente modelo, trabalha com o fluxo de todos os dados contidos na nuvem para aplicação do algoritmo de previsão de dados. Foram reunidos os resultados do conjunto de experimentos com 20 replicações para 5 sementes de aleatoriedade para cada algoritmo.

Para realizar a predição o modelo ARIMA precisa de uma definição de uma série histórica e o posterior treinamento do modelo com esses dados. Essa série é definida como um ponto de tempo no eixo X e uma unidade de valor no eixo Y. No contexto desta pesquisa o valor será trabalhado em 2 momentos, com a grandeza de velocidade e LOS. O objetivo é confrontar os resultados da predição do algoritmo ARIMA com os gráficos de velocidade e LOS já apresentados e analisados anteriormente.

Para montagem da série histórica, foram realizadas os seguintes passos:

1. Inicialmente, foi realizada uma normalização temporal para tratar as diferenças de tempo entre as simulações para cada replicação, gerando o resultado de 20 replicações por semente, com pontos de tempo em comum;
2. Cada replicação tem em sua composição 10 pontos no tempo, cada um com sua unidade de valor;
3. Posteriormente, foi realizada a concatenação linear das 20 replicações, gerando uma série temporal intermediária com 200 pontos de tempo, cada um com sua unidade de valor. Para esclarecer essa abordagem, podemos definir R como uma sequência ininterrupta de resultados de replicação com normalização temporal e valor $tv1, tv2, \dots, tvN$ de modo que $R = \{tv^i\}_{i=1}^N$, onde N finaliza no último resultado tempo/valor da última replicação;
4. Depois, realizamos mais uma concatenação com as replicações de cada semente de aleatoriedade, gerando por fim a série temporal estacionária e definitiva com 1000 pontos de tempo, cada um com sua unidade de valor. Essa série é submetida ao treinamento para gerar a predição;
5. Quando a série de previsão é gerada, inicia o processo de reversão da concatenação, com o objetivo de restabelecer a estrutura inicial de 20 replicações para 5 sementes com 10 pontos de tempo com seu respectivo valor alterado de acordo com a predição;
6. Com essa nova estrutura é aplicada a média entre as replicações alteradas com o valor predito, com o objetivo de suavizar os resultados e gerar os gráficos de comparação entre o resultado do ARIMA e os gráficos de LOS e Velocidade;
7. Cabe reiterar que todo esse processo acontece em 2 momentos diferentes, um para os rótulos de valor para velocidade e outra para o rótulos de valor para LOS.

3.0.5 Análise das velocidades com ARIMA

Para analisar o comportamento da previsão de dados com o ARIMA no aspecto da velocidade vamos avaliar as três Figuras 3.12, 3.13 e 3.14. Conforme descrito na subseção anterior, para fins de comparação, foram previstos os valores de velocidade e gerados gráficos nos mesmos moldes realizados para a avaliação da acurácia com base na velocidade em fluxo de dados contínuos para o *framework*.

Avaliando o comportamento dos gráficos, podemos constatar uma manutenção nas tendências de velocidade em comparação com os resultados das Figuras 3.3, 3.4 e 3.5, porém com variações de aproximadamente 15% nos valores de velocidade.

Cabe destacar, que o processo de predição acontece na finalização de todos os experimentos e suas replicações que seguem o modelo de fluxo de dados contínuos, ou seja, os dados que vão gerar a série histórica, precisam estar no passado. Diante disso, o ARIMA não contempla eventuais variações em tempo de simulação no modelo de janela temporal.

A utilização de um modelo de previsão de séries temporais nessa pesquisa, foi motivada pela necessidade de atestar que um modelo com um certo nível de desacoplamento das camadas pode abarcar uma variedade de recursos e técnicas. Além do fato de uma adoção de predição de dados pode reduzir ainda mais o tráfego de dados pelo *framework*.

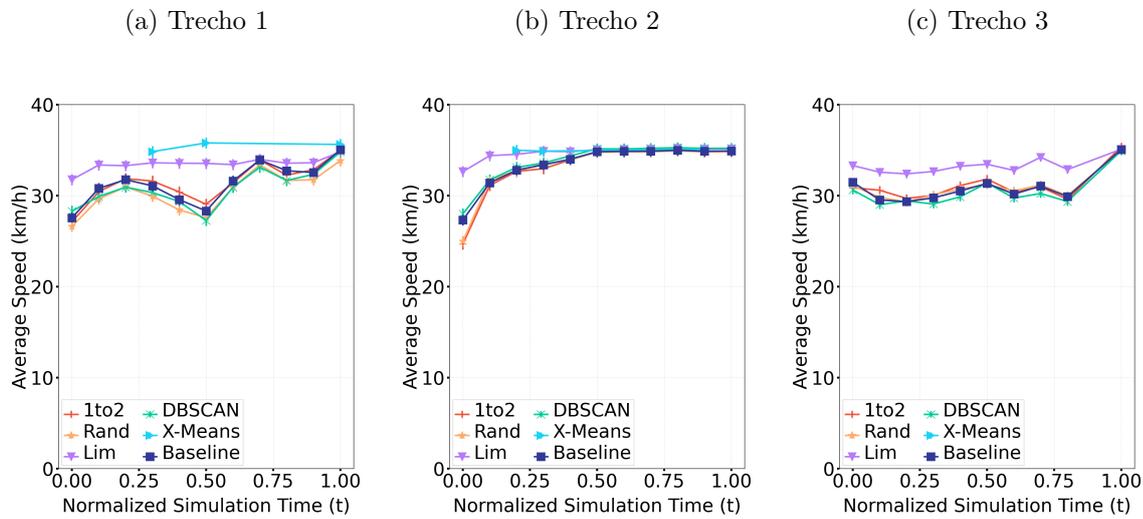


Figura 3.12: Análise da Velocidade com predição do Arima para 30 veículos.

3.0.6 Análise do LOS com ARIMA

Para analisar o comportamento da previsão de dados com o ARIMA no aspecto do LOS vamos avaliar as três Figuras 3.15, 3.16 e 3.17. Conforme descrito anteriormente, para fins de comparação, foram previstos os valores de LOS e gerados gráficos nos mesmos moldes realizados para a avaliação da acurácia com base no LOS em fluxo de dados contínuos para o *framework*.

Avaliando o comportamento dos gráficos, podemos constatar que assim como aconteceu com a velocidade, houve uma manutenção considerável das tendências de LOS em comparação com os resultados das Figuras 3.6, 3.7 e 3.8. Apesar de promissor, os resultados apresentaram variações significativas na escala de classificação de congestionamento de tráfego. Em alguns casos, modificando em até 2 posições o status a definição do fluxo de trânsito.

Outro ponto de análise, foi a variação entre os resultados dos algoritmos, que em alguns casos estavam praticamente sobrepostos no resultado de LOS do fluxo de dados

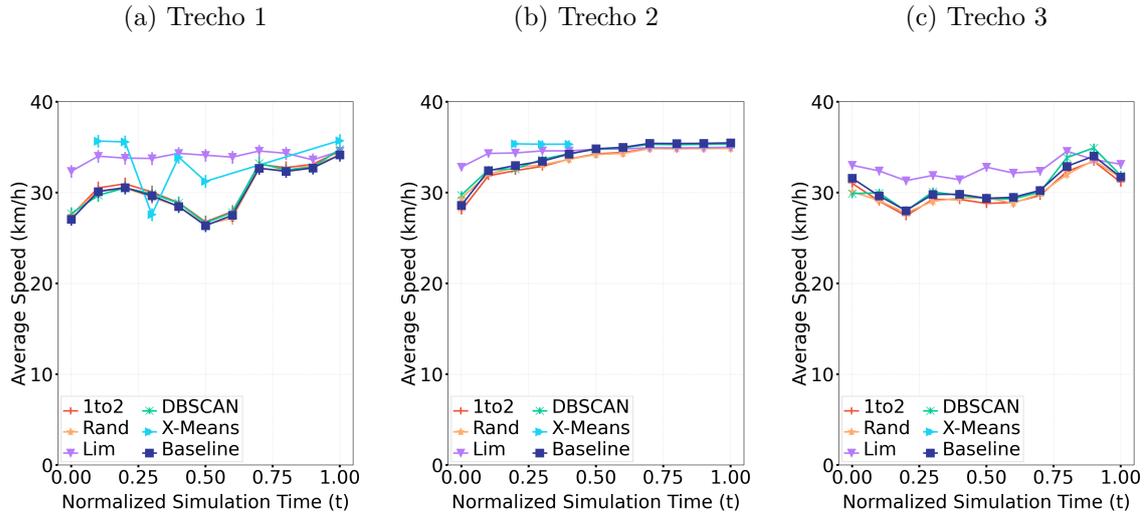


Figura 3.13: Análise da Velocidade com predição do Arima para 50 veículos.

contínuos e no modelo ARIMA para a perspectiva de classificação de LOS, as linhas se distanciaram consideravelmente.

Esse fenômeno acontece possivelmente por conta da quantidade de elementos que compõe a série temporal, entendemos que para melhores resultados, se faz necessários mais dados na série para aprimorar o treinamento e a melhoria dos valores de predição.

Cabe reiterar, que o processo de predição ocorre após a conclusão dos experimentos e replicação de dados contínuos, para isso utilizando uma série histórica gerada a partir de dados passados. O ARIMA não considera variações em tempo real durante a simulação do modelo através de janela temporal.

A escolha de um modelo de previsão de séries temporais nesta pesquisa foi motivado pela necessidade de verificar se um modelo com um certo nível de desacoplamento das camadas pode abranger uma variedade de recursos e técnicas. Seja adição de algoritmos, expansão de rede veicular ou técnicas de previsão de dados. Além disso, a utilização da previsão de dados pode reduzir ainda mais o tráfego de dados pelo *framework*.

3.0.7 Análise de rede

Nesta subseção, serão analisadas questões relacionadas ao desempenho das estratégias de redução de dados em um contexto de fluxo contínuo, avaliando o consumo de recursos de rede durante os processos de monitoramento e detecção de congestionamento de tráfego.

Um dos principais objetivos deste trabalho é estabelecer um modelo eficiente de redução de dados na computação em névoa em fluxos contínuos, visando evitar congestionamento de rede, perda de pacotes, altos custos de comunicação, desperdício de largura de banda e aumento da latência causados pela geração de grandes volumes de dados pela camada de mobilidade.

A largura de banda da rede (NB) é dada por S_B/T , onde S_B representa os dados de tráfego e T é a largura de banda do canal. S_B também é dado por $(L_B + \lambda_B)$, onde

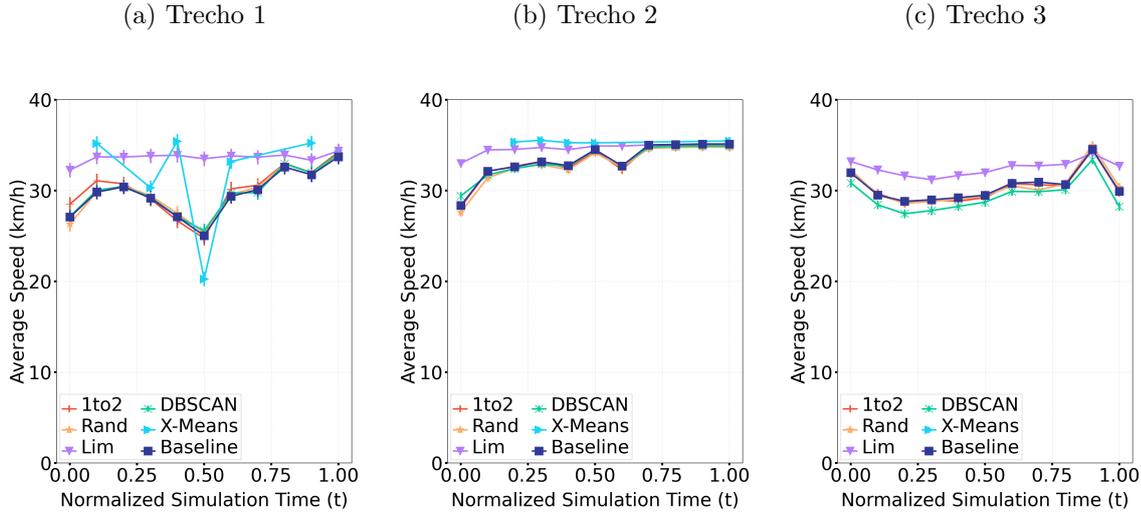


Figura 3.14: Análise da Velocidade com previsão do Arima para 70 veículos.

L_B é a soma de todos os tamanhos de pacotes e λ_B é a taxa média de chegada. Quanto maior o número de veículos, maior é o NB consumido. Uma maneira de abordar este problema é minimizar L_B sujeito à restrição $a_o \geq a_{min}$, onde a_o é a precisão obtida e a_{min} é a precisão mínima aceitável para detectar congestionamento de tráfego. A Figura 3.18 ilustra os resultados do consumo da rede entre a névoa e a nuvem, para cada um dos métodos utilizados ao longo dos experimentos. As análises consideraram as medidas de densidade variando entre 30, 50 e 70 veículos/ km^2 .

Os resultados da Figura 3.18 mostram que a eficiência da rede não depende apenas da precisão na identificação de congestionamentos em comparação com a estratégia do algoritmo **Baseline**, mas também do comportamento desses algoritmos em relação ao uso dos recursos de rede. Uma estratégia eficiente de redução de dados deve ser capaz de reduzir significativamente os dados, mantendo ao mesmo tempo níveis adequados de precisão para detectar pontos de retenção nas rodovias. Além disso, esses conjuntos de dados resultantes devem fornecer informações para apoiar coleta, análises e previsões de retenções de tráfego na camada de mobilidade.

A Figura 3.18 ilustra os resultados de uso dos recursos de rede para os algoritmos: **Baseline**, **1to2**, **Limite**, **Rand**, **XMEANS** e **DBSCAN**. Em primeira análise, pode-se observar que os algoritmos de agrupamento são menos onerosos no que se refere ao consumo da rede.

O algoritmo **Baseline** apresenta resultados de uso de rede ligeiramente mais elevados, uma vez que não há redução de dados nesta estratégia. Por outro lado, os algoritmos **DBSCAN** e **XMEANS** consumiram menos de 25% do uso da rede para todos os cenários.

Os algoritmos **1to2** e **Rand** consumiram menos de 50% do uso da rede para todos os cenários e ainda mantiveram uma boa precisão de detecção de LOS. Em análise adicional pode-se perceber que os valores médios ficaram próximos aos identificados no algoritmo **Baseline**. Este comportamento pode estar relacionado ao fato de que nestas estratégias,

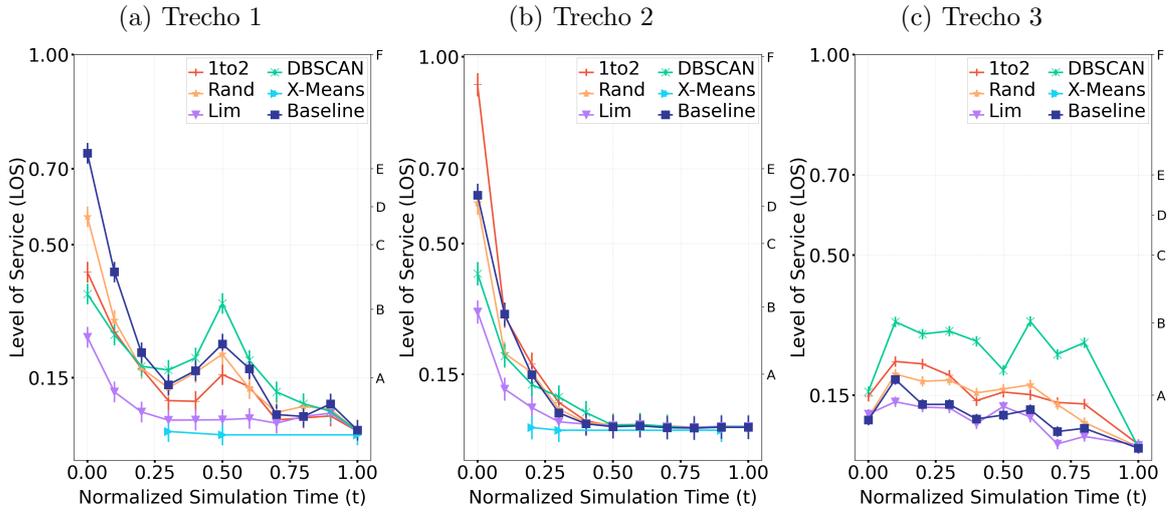


Figura 3.15: Análise do LOS com previsão do Arima para 30 veículos.

o mecanismo de envio de dados permanece similar ao adotado pelo algoritmo *Baseline*.

Podemos observar também que quando há um número maior de veículos no cenário, há mais possibilidades de congestionamento e mais redução na utilização da rede. Os algoritmos **Limite**, **1to2** e **Rand** estão dentro do intervalo de confiança do uso da rede em relação ao **Baseline**, mas com um custo maior do que os algoritmos de agrupamento.

Tanto o X-Means quanto o DBSCAN reduzem o tráfego de dados em 60,85% e 75,61% em relação ao **Baseline**, respectivamente. Além disso, o **DBSCAN** reduz 28,90% mais dados de tráfego do que o **1to2**, que é a melhor abordagem para redução de dados sem a utilização de agrupamento, utilizada nesse trabalho. O algoritmo **DBSCAN** também apresentaram alto grau de acurácia quando foram avaliados os níveis de representatividade, tanto no LOS, quanto na velocidade. A Tabela 3.3 demonstra, de forma mais detalhada, os dados apresentados na Figura 3.18.

Tabela 3.3: Network Usage

Density km^2	30		50		70	
Algorithm	NB (KB/sec)	%	NB (KB/sec)	%	NB (KB/sec)	%
Baseline	928.0	41.87	7.606	31.93	607.0	28.37
1to2	247.0	39.2	1.06	31.42	544.0	31.28
Rand	350.39	49.0	7.580	43.7	300.3	32.7
Lim	505.66	71.51	2.318	32.4	610.33	37.1
CDS-X-Means	183.34	0.94	8.934	2.20	816.30	1.88
CDS-DBSCAN	183.59	5.39	8.931	3.39	815.75	3.09

Analisando a Tabela 3.3, pode-se observar que o algoritmo **DBSCAN** e **XMEANS**

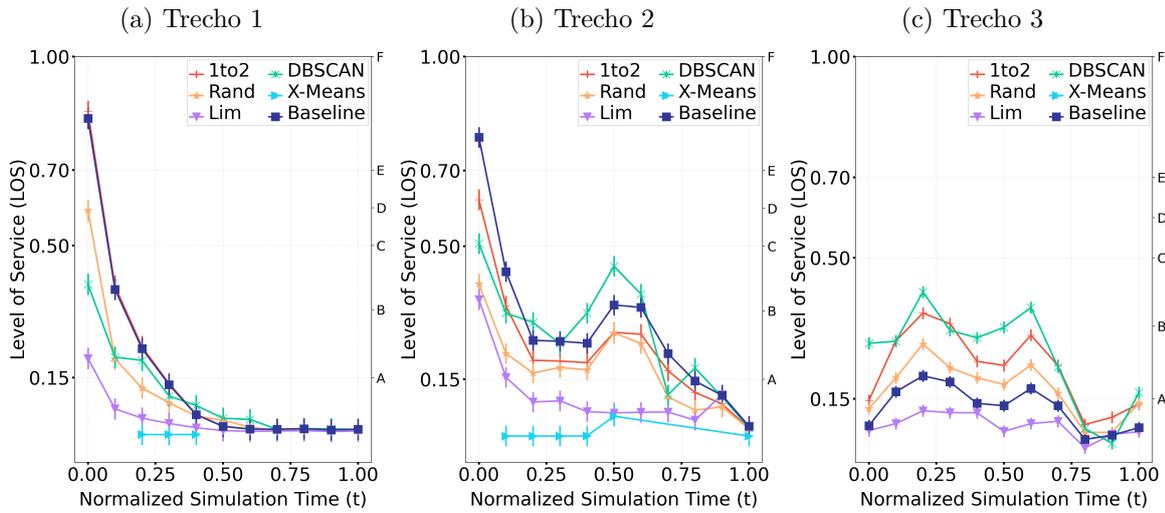


Figura 3.16: Análise do LOS com previsão do Arima para 50 veículos.

utilizam a menor quantidade relativa de recursos de rede, porém conforme mostrado nas Figuras 3.9, 3.10 e 3.11, apenas o **DBSCAN** manteve os melhores níveis de acurácia representados na identificação e classificação dos principais pontos de congestionamento em todas as densidades analisadas. Adicionalmente, o algoritmo manteve um comportamento constante frente ao algoritmo *Baseline*. Entre os algoritmos baseados em amostragem simples, podemos observar o equilíbrio do resultados de uso de rede e que, com o aumento da densidade veicular, existe uma tendência de diminuição do consumo da rede.

De acordo com os resultados da simulação, alguns métodos baseados em agrupamento fornecem uma classificação assertiva de fluxo de tráfego quando analisados sob a métrica LOS, com menos uso de recursos de rede do que outras abordagens, como **1to2**, **Rand** e **Limite**. Esse comportamento é corroborado pelos resultados do **DBSCAN** e por um trabalho anterior (PEIXOTO et al., 2020). Neste trabalho, os autores analisam a relação entre o uso da rede e o número de grupos realizados pelo **DBSCAN**.

Os autores constataram em seus estudos que o aumento do número de veículos e, conseqüentemente, do número de aglomerados, indica um menor número de pontos de ruído. Além disso, com o aumento do número de veículos, o número de veículos por *cluster* também aumenta, reduzindo proporcionalmente a quantidade de dados enviados por grupo. Portanto, pode-se mostrar com base nos experimentos apresentados que o número de *clusters*, densidade veicular e LOS são significativamente interdependentes.

Na Figura 3.18, os dados do algoritmo **DBSCAN** mostram uma tendência inversa em relação ao algoritmo **Baseline**, indicando que à medida que a densidade aumenta, menos recursos de rede são necessários. Isso ocorre devido ao congestionamento causado pelo maior número de veículos, gerando *clusters* com mais elementos, gerando mais representatividade e um menor consumo da rede.

Adicionalmente, modelos de previsão de dados como ARIMA podem contribuir no contexto de redução de dados com um certo nível de acurácia, conforme verificado nas análises de LOS e velocidade do modelo de previsão demonstradas anteriormente.

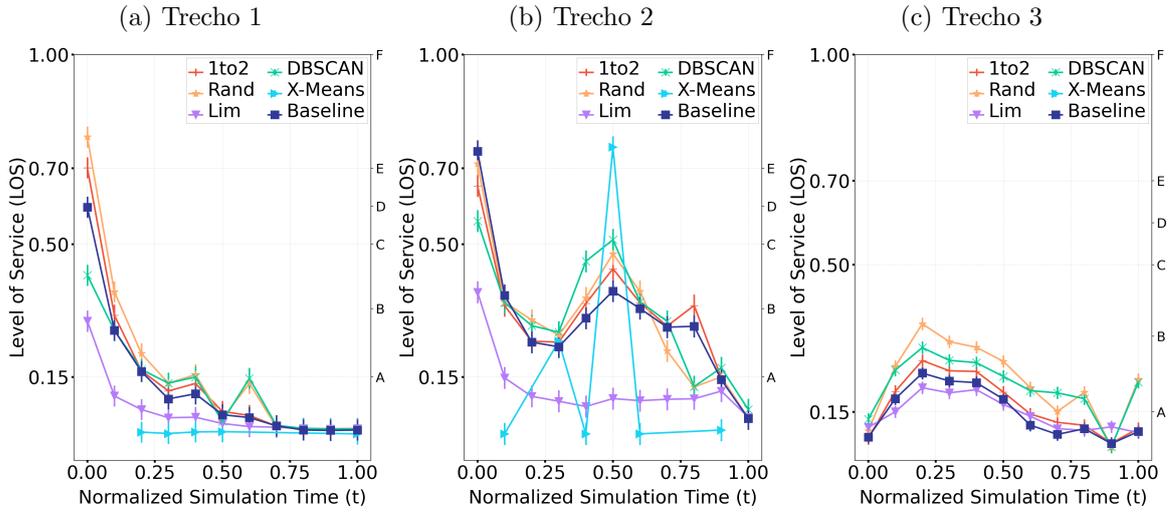


Figura 3.17: Análise do LOS com previsão do Arima para 70 veículos.

Na presente pesquisa o ARIMA está sendo utilizado com base na composição de uma série histórica dos dados armazenados na nuvem, portanto não contempla eventuais mudanças no ambiente veicular no momento do experimento, ou seja, esse modelo está condicionado a eventos ocorridos no passado para definir a previsão do futuro. Entretanto, se for utilizada como única técnica do *framework* proposto, o consumo de rede entre a névoa e nuvem que é o objeto de nosso estudo, será nulo no contexto de sua utilização, tendo em vista que os dados já estão armazenados na nuvem.

Podemos analisar as taxas de redução apresentadas sob a perspectiva dos custos de comunicação relativos a *Operational Expenditure* (OPEX). Assim podemos trabalhar os efeitos práticos dessa redução, em dois pontos importantes:

- Armazenamento de dados, que está relacionado à quantidade de dados necessários para que um sistema de transporte inteligente possa realizar análises e identificar eventuais mudanças no comportamento do tráfego;
- O custo de comunicação, que está relacionado ao comportamento do tráfego de dados pela rede durante o monitoramento e gerenciamento dos espaços urbanos.

As taxas de redução apresentadas têm um impacto relevante no armazenamento de dados. Estudos demonstram que um veículo conectado por um sistema de transporte inteligente pode exigir cerca de 1 TB de dados por dia (XU et al., 2018). Correlacionando esses valores com o preço médio do armazenamento em nuvem, que é de cerca de \$24,00/mês cada 1 TB, é possível supor que o custo máximo será de $1 \text{ TB} * 30 \text{ dias} = 30 \text{ TB/mês}$ por veículo. Em um cenário de 70 veículos, esse valor pode chegar a 10.800TB, gerando um total anual de cerca de US\$ 756.000 apenas para armazenamento de dados. Estes números são significativamente maiores em um cenário de veículos autônomos.

Outro aspecto a ser observado consiste no aumento do número de veículos autônomos conectados (CAV). O CAV é uma tecnologia transformadora com potencial para reduzir

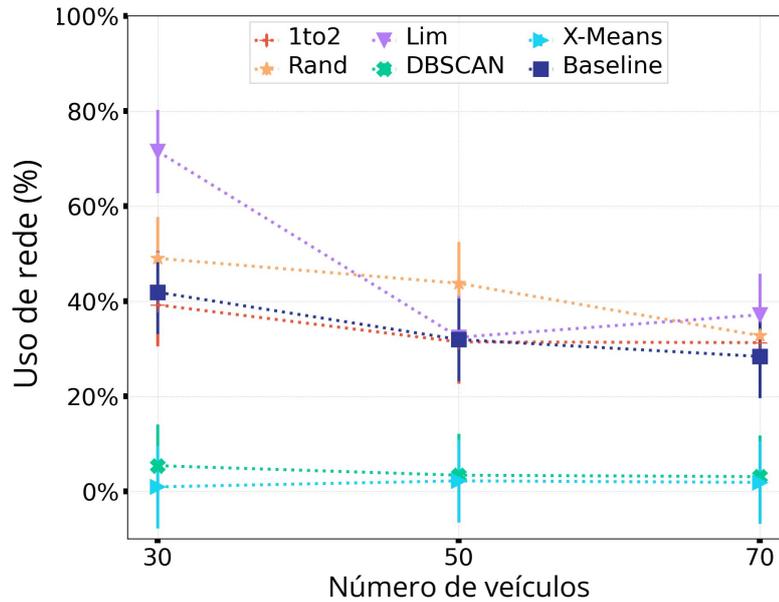


Figura 3.18: Comparativo da Largura de Banda e a Quantidade de Veículos.

acidentes de trânsito, melhorar a qualidade de vida e aumentar a eficiência dos sistemas de transporte. Estima-se que cerca de 70% dos novos veículos em todo o mundo terão uma interface de conectividade integrada a uma rede celular para fornecer dados veiculares de gateways, softwares ou sensores (GINO, 2006). Por se tratar de um modelo a presente proposta pode suportar a adoção de CAVs em seu contexto para trabalhos futuros.

Considerando que os dados gerados por CAVs podem ultrapassar 2 GB por segundo, esse crescimento exponencial de dados é inevitável, aumentando o custo de comunicação da rede e a complexidade computacional para orquestração de dados (por exemplo, armazenamento, processamento e análise) e extração de conhecimento. Assim, ao analisar a métrica de redução de dados proposta neste experimento frente aos valores cobrados por este serviço em 2022, pode-se notar uma redução relevante nos valores. A figura 3.19 abaixo ilustra esta análise.

Além disso, há também os custos relacionados ao tráfego de dados pela rede. Nesta proposta, destacamos os custos no enlace entre a Névoa e a Nuvem. Nesta conjuntura, conceitos como taxa de transição, largura de banda ou taxa de uso do *spectro* são de fundamental importância. Cada célula 5G é capaz de suportar pelo menos um milhão de dispositivos por Km². Assim, a velocidade de download e upload é dividida igualmente entre todos os dispositivos conectados na mesma célula (SHAW; BELTRÁN; SOWERBY, 2017).

No contexto da computação urbana, centenas de veículos podem compartilhar este mesmo canal de comunicação e cada um deles é capaz de emitir dados a uma taxa de transição de 1 Mhz. Esses dados são recebidos pela RSU e enviados para o processamento na infraestrutura de Névoa ou Nuvem. Essa carga de trabalho aliada ao uso dos diferentes dispositivos nesta mesma rede podem levar atraso de rede, sobrecarga e a consequente

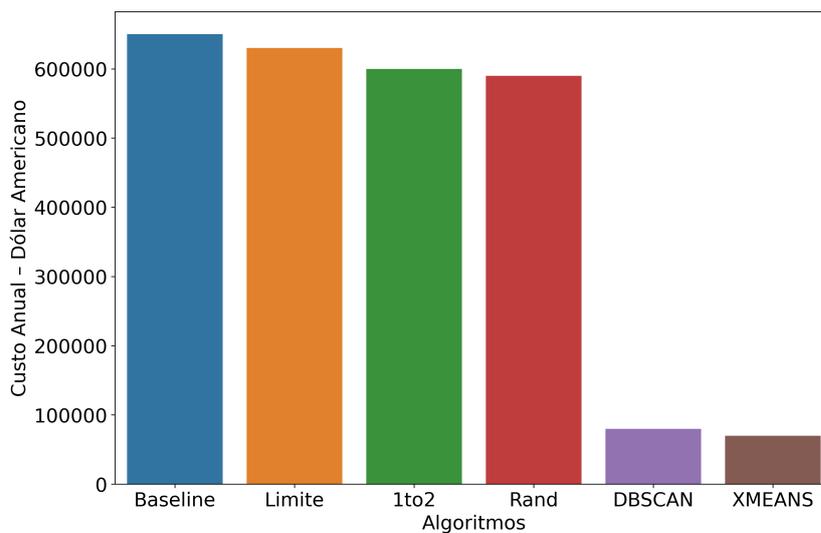


Figura 3.19: Estimativa de custo de armazenamento (anual).

perda de qualidade em função da desigualdade na taxa de uso do *spectro*. Portanto, técnicas de redução de dados como **DBSCAN** podem contribuir para combater esse aumento de custo associado, reduzindo em cerca de 70% o uso dos recursos endereçados à rede veicular. Essa ação pode contribuir para o aumento da qualidade geral do serviço e gerar um melhor aproveitamento mais adequado dessa infraestrutura.

CONCLUSÕES

O congestionamento do tráfego é um problema global que afeta as grandes cidades, causando impactos econômicos e sociais significativos. Os sistemas de transporte inteligente enfrentam o desafio de lidar com uma grande quantidade de dados para medir e gerenciar o fluxo de tráfego, o que pode aumentar os custos da rede e sobrecarregar a infraestrutura existente, especialmente na conexão entre a borda da rede e a nuvem. Para lidar com esse panorama, esse trabalho propõe um *framework* de redução de dados na névoa criado para tratar o problema de redução do custo de comunicação em um sistema de gerenciamento de tráfego inteligente.

Neste trabalho concluímos que algoritmos baseados em agrupamento têm um desempenho superior na redução de dados em comparação com abordagens baseadas em amostragem simples. Isso acontece por que em um cenário de maior densidade veicular, mais veículos são incluídos no grupos representativos e menos informação sobre o contexto veicular é trafegada da névoa para a nuvem. Mesmo com menos informação a acurácia dos algoritmos de agrupamento segue bem próxima da tendência de resultado do algoritmo **Baseline**.

Dos algoritmos de agrupamento utilizados neste trabalho, foi possível observar que o algoritmo **DBSCAN** manteve níveis de redução e representatividade mais significativos, se apresentado com promissor quando aplicado a um contexto de fluxo de dados contínuo.

Esse trabalho busca promover o uso da computação em névoa na computação urbana, principalmente em relação aos Sistemas de Transporte Inteligente.

O *framework* proposto nessa pesquisa, trabalha com diferentes abordagens para compor um modelo arquitetural capaz de atuar em diferentes cenários. Essa arquitetura foi desenvolvida para manter um fraco acoplamento entre as camadas, pois essa dinâmica possibilita a inclusão eventual de outros cenários urbanos a esse modelo. Essa ação tem como objetivo não apenas garantir a independência do *framework*, mas também abrir oportunidades para novos experimentos em ambientes simulados ou reais.

O principal objetivo desta pesquisa foi desenvolver um *framework* de redução de dados baseado em computação em névoa para lidar com fluxos contínuos de dados, permitindo o

monitoramento e detecção de congestionamento. Para isso, além do modelo de camadas, foram utilizados um conjunto de 6 algoritmos divididos em amostragem e agrupamento. Os de amostragem simples são **Baseline**, **1TO2**, **Rand** e **Limite** e os algoritmos baseados em agrupamento temos **XMEANS** e **DBSCAN**.

Além do objetivo geral, buscamos alcançar alguns objetivos específicos relacionados abaixo:

- Definição e modelagem de um ambiente de simulação que lidasse com rede e cenários urbanos;
- Definição e implementação de um padrão de conectividade para recebimento de dados em fluxo contínuo;
- Definição de estratégias de redução de dados na névoa utilizando o conceito de janelas temporais;
- Implementação dos modelos de classificação do fluxo de tráfego na nuvem;
- Realização de experimentos utilizando replicações para indicar a influência dos algoritmos na acurácia e tomada de decisão em ambiente de simulação de tráfego;
- Realização de experimentos utilizando replicações para avaliar a redução do uso da rede no modelo proposto;
- Definição de estratégias de predição de dados com base nas séries temporais;
- Submissão de artigos científicos para avaliação da proposta pela comunidade acadêmica.

4.1 TRABALHOS FUTUROS

Essa proposta de pesquisa visa desenvolver um *Framework* de redução de dados na computação em névoa para monitorar e detectar congestionamento em um cenário urbano específico. Para isso, além do modelo de camadas, foi utilizado um conjunto de 6 algoritmos divididos em amostragem e agrupamento. Os de amostragem simples são **Baseline**, **1TO2**, **Rand** e **Limite** e os algoritmos baseados em agrupamento temos **XMEANS** e **DBSCAN**.

Como trabalho futuro, sugere-se implementar mecanismos para minimizar os efeitos dos algoritmos de agrupamento na acurácia da classificação de nível de serviço (LOS) em áreas de menor densidade nas rodovias. Quanto ao uso de algoritmo baseado em densidade (**DBSCAN**), sugere-se avaliar o impacto do raio de vizinhança nas abordagens e como ele influencia nos níveis de acurácia ao aplicar a classificação do nível de serviço (LOS). Essa sugestão tem como base trabalhos que demonstraram que quando o valor ε escolhido é muito pequeno, uma grande parte dos dados de tráfego não será agrupada gerando ruído. Porém, se o valor ε for muito alto, os *clusters* se unem perdendo a qualidade em termos de representatividade dos dados.

Um aspecto relevante é que a proposta de pesquisa se concentra na análise do comportamento dos algoritmos de agrupamento em termos de eficiência na redução de dados e

capacidade de manter subconjuntos representativos em relação ao conjunto de dados original. Uma importante extensão dessa pesquisa seria analisar o comportamento de mais algoritmos de agrupamento e de predição em um contexto de fluxo de dados contínuo para a identificação de congestionamentos, além do DBSCAN, XMEANS e ARIMA que foram utilizados nessa proposta.

Como sugestão final de extensão, temos a adaptação dos conceitos desta proposta para o contexto de veículos autônomos conectados (CAV), considerando que o *framework* apresentado é estruturado em camadas com um certo grau de desacoplamento, permitindo a customização para diferentes cenários. No entanto, os mecanismos de comunicação precisarão de ajustes e adaptações para funcionarem no contexto dos carros autônomos.

4.2 PUBLICAÇÕES

Com base no desenvolvimento do projeto de pesquisa e nos resultados obtidos, foi necessário submeter esses dados à apreciação da academia por meio da submissão de artigos que abordam diferentes aspectos da proposta. Entre os artigos aceitos para publicação destacam-se os *journals* submetidos à *Ad Hoc Networks*, *Vehicular Communications*, com **qualis A1**, além da conferência *Vehicular Technology Conference* (VTC) também com **qualis A1**. As publicações e seus respectivos status constam na lista abaixo.

- Peixoto M, Mota E, Maia A, Santos M, Junior W, Villas L **Exploiting Fog Computing with an Adapted DBSCAN for Traffic Congestion Detection System**, Vehicular Technology Conference: VTC2020-Fall, 2020 (Aceito)
- Peixoto M; Maia A; Mota E; Rangel E; Costa D; Turgut D; Villas L **A Traffic Data Clustering Framework based on Fog Computing for VANETs**, Vehicular Communications, 2021 (Aceito)
- Peixoto M; Mota E; Maia A; Lobato W; Salahuddin M; Boutaba R; Villas L **Fog-Jam: A Fog Service for Detecting Traffic Congestion in a Continuous Data Stream VANET**, Ad Hoc Networks, 2022 (Aceito)

REFERÊNCIAS BIBLIOGRÁFICAS

- AL-SULTAN, S. et al. A comprehensive survey on vehicular ad hoc network. *Journal of Network and Computer Applications*, v. 37, p. 380 – 392, 2014. ISSN 1084-8045. Disponível em: <http://www.sciencedirect.com/science/article/pii/S108480451300074X>.
- ATZORI, L.; IERA, A.; MORABITO, G. The internet of things: A survey. *Computer Networks*, v. 54, n. 15, p. 2787 – 2805, 2010. ISSN 1389-1286. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1389128610001568>.
- BATISTA, B. G. et al. Architecture for internet of things environment management with quality of service assurance. In: *IEEE HPCS - International Conference on High Performance Computing and Simulation, 2018, Orléans, France*. [S.l.: s.n.], 2018.
- BILAL, K. et al. Potentials, trends, and prospects in edge technologies: Fog, cloudlet, mobile edge, and micro data centers. *Computer Networks*, v. 130, p. 94 – 120, 2018. ISSN 1389-1286.
- BOARD, T. R. Highway capacity manual. *National Research Council*, 2010.
- CERQUEIRA, V. et al. On evaluating floating car data quality for knowledge discovery. *IEEE Transactions on Intelligent Transportation Systems*, p. 1–12, 2018. ISSN 1524-9050.
- CHIANG, M. Fog networking: An overview on research opportunities. *arXiv preprint arXiv:1601.00835*, 2016.
- DESAI, P. V. A survey on big data applications and challenges. In: *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*. [S.l.: s.n.], 2018. p. 737–740.
- ESTER, M. et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. [S.l.]: AAAI Press, 1996. (KDD'96), p. 226–231.
- GINO, A. *Highway Traffic Act Investigations*. [S.l.]: Emond Publishing, 2006.
- GUIDO, G. et al. Big data for public transportation: A dss framework. In: *IEEE. 2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*. [S.l.], 2017. p. 872–877.
- HOU, X. et al. Vehicular Fog Computing: A Viewpoint of Vehicles as the Infrastructures. *IEEE Transactions on Vehicular Technology*, v. 65, n. 6, p. 3860–3873, 2016.

IDC. *Worldwide Internet of Things Forecast Update 2022–2029, document #US45213219*. Framingham, MA, USA: [s.n.], 2023.

LEITE, D. M. et al. The influence of resource allocation on cloud computing performance. In: *Proceedings of the Symposium on Applied Computing*. New York, NY, USA: ACM, 2017. (SAC '17), p. 1516–1521. ISBN 978-1-4503-4486-9.

MUKHERJEE, M.; SHU, L.; WANG, D. Survey of fog computing: Fundamental, network applications, and research challenges. *IEEE Communications Surveys Tutorials*, v. 20, n. 3, p. 1826–1857, thirdquarter 2018. ISSN 1553-877X.

PATROUMPAS, K.; SELLIS, T. Window specification over data streams. In: SPRINGER. *International Conference on Extending Database Technology*. [S.l.], 2006. p. 445–464.

PEIXOTO, M. L. et al. Exploiting fog computing with an adapted dbscan for traffic congestion detection system. In: IEEE. *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*. [S.l.], 2020. p. 1–5.

PELLEG, D.; MOORE, A. W. X-means: Extending k-means with efficient estimation of the number of clusters. In: *Proceedings of the Seventeenth International Conference on Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000. (ICML '00), p. 727–734. ISBN 1558607072.

Schmidt, R. K. et al. Exploration of adaptive beaconing for efficient intervehicle safety communication. *IEEE Network*, v. 24, n. 1, p. 14–19, Jan 2010.

SCHWARZ, G. Estimating the dimension of a model. *The Annals of Statistics*, Institute of Mathematical Statistics, v. 6, n. 2, p. 461–464, 1978. ISSN 00905364.

SHAW, B. A.; BELTRÁN, H. F.; SOWERBY, K. W. *Valuing spectrum at mm wavelengths for cellular networks*. [S.l.], 2017. Disponível em: <https://ideas.repec.org/p/zbw/itsp17/168539.html>.

SILVA, M. et al. A crowdsensing platform for monitoring of vehicular emissions: A smart city perspective. *Future Internet*, v. 11, n. 1, 2019. ISSN 1999-5903.

SOUZA, A. M. de et al. Itssafe: An intelligent transportation system for improving safety and traffic efficiency. In: *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*. [S.l.: s.n.], 2018. p. 1–7. ISSN 2577-2465.

SOUZA, A. M. de; VILLAS, L. A. A fully-distributed traffic management system to improve the overall traffic efficiency. In: *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. New York, NY, USA: ACM, 2016. (MSWiM '16), p. 19–26. ISBN 978-1-4503-4502-6.

SOUZA, A. M. de et al. Icarus: Improvement of traffic condition through an alerting and re-routing system. *Computer Networks*, Elsevier, v. 110, p. 118–132, 2016.

STEFANELLI, V. F. A questão da mobilidade urbana nas metrópoles brasileiras. Universidade Federal do Rio de Janeiro, 2015.

TANIZAKI, M.; WOLFSON, O. Randomization in traffic information sharing systems. In: *Proceedings of the 15th Annual ACM International Symposium on Advances in Geographic Information Systems*. New York, NY, USA: Association for Computing Machinery, 2007. (GIS '07). ISBN 9781595939142.

TOLPIN, D. Progressive temporal window widening. *arXiv preprint arXiv:1604.00997*, 2016.

UPPOOR, S. et al. Generation and analysis of a large-scale urban vehicular mobility dataset. *IEEE Transactions on Mobile Computing*, IEEE, v. 13, n. 5, p. 1061–1075, 2013.

WAHID, A.; RAO, A. C. S.; GOEL, D. Server communication reduction for gps-based floating car data traffic congestion detection method. In: KRISHNA, A.; SRIKANTIAH, K.; NAVEENA, C. (Ed.). *Integrated Intelligent Computing, Communication and Security*. Singapore: Springer Singapore, 2019. p. 415–425. ISBN 978-981-10-8797-4.

Xinzhou Wu et al. Vehicular Communications Using DSRC: Challenges, Enhancements, and Evolution. *IEEE Journal on Selected Areas in Communications*, v. 31, n. 9, p. 399–408, sep 2013.

XU, W. et al. Vifi: Vehicle-to-vehicle assisted traffic offloading via roadside wifi networks. In: *2018 IEEE Global Communications Conference (GLOBECOM)*. [S.l.: s.n.], 2018. p. 1–6. ISSN 2576-6813.

YI, S.; LI, C.; LI, Q. A survey of fog computing: Concepts, applications and issues. In: *Proceedings of the 2015 Workshop on Mobile Big Data*. New York, NY, USA: ACM, 2015. (Mobidata '15), p. 37–42. ISBN 978-1-4503-3524-9. Disponível em: <http://doi.acm.org/10.1145/2757384.2757397>.

ZHENG, Y. et al. Urban computing: Concepts, methodologies, and applications. *ACM Trans. Intell. Syst. Technol.*, ACM, New York, NY, USA, v. 5, n. 3, p. 38:1–38:55, set. 2014. ISSN 2157-6904.

Zhu, L. et al. Big data analytics in intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, v. 20, n. 1, p. 383–398, 2019. ISSN 1558-0016.