Marco Aurélio Brunoro Thomé

Um Arcabouço para Detecção e Alerta de Anomalias de Mobilidade Urbana em Tempo Real

Vitória, ES

Novembro, 2020

Marco Aurélio Brunoro Thomé

Um Arcabouço para Detecção e Alerta de Anomalias de Mobilidade Urbana em Tempo Real

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Mestre em Informática.

Universidade Federal do Espírito Santo – UFES

Centro Tecnológico

Programa de Pós-Graduação em Informática

Orientador: Prof. Dr. Vinícius Fernandes Soares Mota

Co-orientadora: Profa. Dra. Roberta Lima Gomes

Vitória, ES

Novembro, 2020

Marco Aurélio Brunoro Thomé
Um Arcabouço para Detecção e Alerta de Anomalias de Mobilidade Urbana em Tempo Real/ Marco Aurélio Brunoro Thomé. – Vitória, ES, Novembro, 2020-
75 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Vinícius Fernandes Soares Mota

Co-orientadora: Profa. Dra. Roberta Lima Gomes

Dissertação de Mestrado – Universidade Federal do Espírito Santo – UFES
Centro Tecnológico
Programa de Pós-Graduação em Informática, Novembro, 2020.

1. Cidades inteligentes. 2. Ciência dos dados. I. Thomé, Marco Aurélio Brunoro. II. Universidade Federal do Espírito Santo. IV. Um Arcabouço para Detecção e Alerta de Anomalias de Mobilidade Urbana em Tempo Real

CDU 02:141:005.7

Marco Aurélio Brunoro Thomé

Um Arcabouço para Detecção e Alerta de Anomalias de Mobilidade Urbana em Tempo Real

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Mestre em Informática.

Prof. Dr. Vinícius Fernandes Soares Mota
Orientador

Profa. Dra. Roberta Lima Gomes
Co-orientadora

Prof. Dr. José Gonçalves Pereira Filho
Convidado 1

Prof. Dr. Thiago Henrique Silva
Convidado 2

Vitória, ES
Novembro, 2020

Dedico este trabalho à minha família e amigos, especialmente aos meus irmãos e irmã, que sempre me apoiaram e incentivaram a seguir meu próprio caminho e sempre estiveram ao meu lado para me apoiar, não importando o caminho.

Agradecimentos

Agradeço primeiramente aos orientadores Vinícius e Roberta pela dedicação e empenho no exercício do ensino, como poucos professores que tive em minha vida. Agradeço aos meus pais pelo constante zelo e estrutura, sem os quais dificilmente conseguiria esta e tantas outras vitórias. Agradeço aos meus três irmãos, meus mais queridos e amados amigos, por sempre me apoiarem em todas as minhas empreitadas sem hesitação, e nunca medirem esforços em meu auxílio, sempre verdadeiros comigo. Agradeço à Karine pelo incansável incentivo, apoio e cuidado em todos árduos dias, sempre a favor daquilo que me engrandece como pessoa. Agradeço ao meu amigo Gleibis, ao meu irmão Miguel e aos meus orientadores por terem atuado diretamente para melhoria desta dissertação. Agradeço aos meus amigos de igreja, especialmente ao Grupo de Teatro Santa Clara de Assis, um refúgio nas horas conturbadas e mais ainda um lugar de alegrias. Finalmente, agradeço a todos que fizeram parte de minha história, nos momentos bons e ruins, pois ajudaram a construir a pessoa que sou hoje e merecem todo meu carinho e respeito.

"O único que não possui chance alguma de conseguir
é aquele que desistiu de tentar."
(ditado familiar)

Resumo

Dados coletados por sensores, câmeras, redes sociais e aplicativos podem contribuir para detecção automática de eventos atípicos no trânsito, como acidentes e alagamentos. Além disso, a natureza heterogênea das diversas fontes de dados traz, como vantagem, a redundância de informações, o que permite aumentar o grau de confiabilidade de eventos detectados. Neste trabalho é proposta uma solução para detecção de anomalias em tempo real e envio de alertas, com uma interface que suporta fontes de dados heterogêneos. A abordagem é baseada em clusterização de séries temporais por faixa de horário, gerando uma série temporal padrão que é utilizada como parâmetro de classificação de observações em anomalias, por meio de uma técnica de detecção de outliers. Para validar a proposta, foi elaborado um protótipo, como prova de conceito, em que foram implementadas interfaces para dados disponibilizados pela prefeitura de Vitória-ES, provenientes da plataforma *Waze*, e *Twitter*. Então, o protótipo foi avaliado, variando os parâmetros, como algoritmo de clusterização e método de detecção de outlier, de forma a se obter uma boa configuração para uma possível implantação do sistema. A partir de dados reais da cidade, os resultados mostram que a solução proposta pode auxiliar os gestores e agentes das cidades nas tomadas de decisões. **Palavras-chaves**: Mobilidade urbana. Clusterização. Séries temporais. Detecção de anomalias. Tempo real.

Abstract

Data gathered by sensors, cameras, social networks, and applications can contribute on automatic detection of atypical traffic events like accidents and floodings. Moreover, the heterogeneous nature of a plethora of data sources has the advantage of information redundancy, which helps increasing the degree of reliability for detected events. In this work, it is proposed a solution for real-time anomaly detection and notifications, with an interface that allows heterogeneous data sources. The approach is based in splitted time series clustering by periods, generating a pattern that is used as a parameter for event classification, with an outlier detection technique. To evaluate the proposal, a prototype was developed, as a proof of concept, with deployment of data interfaces for Vitoria's Waze based API and Twitter information. Then, the prototype's performance was analyzed after varying some parameters, like the clustering algorithm and outlier detection method, in order to obtain a good configuration set for a definitive system use. By using real data collected in the city, the results show that the proposed solution can help managers and agents in their decision making. **Keywords**: Urban mobility. Clustering. Time series. Anomaly detection. Real-time.

Lista de ilustrações

Figura 1 – Visão geral do arcabouço 38
Figura 2 – *dados_via* para os dias 26 e 27 de Setembro de 2019 42
Figura 3 – Séries Temporais Diárias após *Function* nos dias 26 e 27 de Setembro de 2019 42
Figura 4 – Clusters em cada Período do Dia e Série Padrão Usando K-Means com 3 Clusters nos dias 26 e 27 de Setembro de 2019 43
Figura 5 – Observações de 26 e 27 de Setembro de 2019 Classificadas como Normais e Anomalias Usando K-Means com 3 Clusters, Janela de 30 Minutos e Histórico de 30 Dias 44
Figura 6 – Possíveis Fluxos de Informação na Avaliação de Anomalias 45
Figura 7 – Proposta de Fluxos de Informação 47
Figura 8 – Número de Alertas por Subtipo em Vitória de Outubro de 2017 a Outubro de 2018 52
Figura 9 – Número de Notificações de Cada Tipo por Cidade de Outubro de 2017 a Outubro de 2018 52
Figura 10 – Alertas Acumulados por Dia de Semana e Horário em Duas Avenidas, de Outubro de 2017 a Outubro de 2018 53
Figura 11 – Observações de Engarrafamento de Setembro de 2018 a Outubro de 2019 54
Figura 12 – Velocidades em Engarrafamentos em 4 Vias de Vitória em Um Mês . . 54
Figura 13 – Dados Reduzidos de 2 Dias da Av. Nossa Senhora da Penha 55
Figura 14 – Dois Clusters Encontrados com DBA na Av. Nossa Senhora da Penha em Dias da Semana 56
Figura 15 – Maiores Clusters Encontrados Usando DBA com 3 Clusters em Duas Vias 56
Figura 16 – Número de Anomalias por Dia de 1 de Julho a 15 de Outubro de 2019 60
Figura 17 – Taxas de Acerto Diárias com Janela de 30 Minutos e Histórico de 30 Dias, de 1 de Julho a 15 de Outubro de 2019 60
Figura 18 – Avaliação do Protótipo com Birch e DBSCAN 62
Figura 19 – Avaliação do Protótipo com OPTICS e K-Means com 3 Clusters 63
Figura 20 – Clusters Obtidos Usando K-Means com 3 Clusters, Histórico de 30 Dias e Janela de 30 Minutos, no Dia 1 de Outubro de 2019 64
Figura 21 – Anomalias Obtidas Usando K-Means com 3 Clusters, Histórico de 30 Dias, Janela de 30 Minutos e Z-Score, no Dia 1 de Outubro de 2019 . . 64
Figura 22 – Quantidade de Anomalias e Taxa de Acerto Obtidas Usando K-Means com 3 Clusters, Histórico de 30 Dias, Janela de 30 Minutos e Z-Score, de 1 de Julho a 15 de Outubro de 2019 65

Figura 23 – Quantidade Total de Anomalias Detectadas por Período do Dia, Usando K-Means com 3 Clusters, Histórico de 30 Dias, Janela de 30 Minutos e Z-Score, de 1 de Julho a 15 de Outubro de 2019 66

Lista de tabelas

Tabela 1 – Parâmetros do Algoritmo 1 41
Tabela 2 – Atributos dos Dados da Tabela de Alertas 51
Tabela 3 – Atributos dos Dados da Tabela de Engarrafamentos 51
Tabela 4 – Parâmetros de Entrada para Avaliar o Protótipo do Arcabouço 61

Lista de abreviaturas e siglas

ACO	Ants Colony Optimization
API	Application Programming Interface
CIOM	Central Integrada de Operações e Monitoramento
DBA	DTW Barycenter Averaging
DTW	Dynamic Time Warping
GA	Genetic Algorithms
GSA	Gravitational Search Algorithm
HTM	Hierarchical Temporal Memory
IoT	Internet of Things
JSON	JavaScript Object Notation
M2M	Machine to Machine
MQTT	Message Queuing Telemetry Transport
PSO	Particle Swarm Optimization
SDR	Sparse Distributed Representation
SVM	Support Vector Machine

Sumário

1 INTRODUÇÃO . . . 23
1.1 Definição do Problema . . . 24
1.2 Objetivos . . . 25
1.3 Contribuições . . . 25
1.4 Organização do Texto . . . 26

2 FUNDAMENTAÇÃO TEÓRICA . . . 27
2.1 Algoritmos de Clusterização . . . 27
2.1.1 DBSCAN . . . 27
2.1.2 OPTICS . . . 27
2.1.3 K-Means . . . 28
2.1.4 BIRCH . . . 28
2.1.5 K-Means com DBA . . . 29
2.2 Detecção de Anomalias . . . 29
2.2.1 Z-Score . . . 30
2.2.2 IQR . . . 30
2.3 Aprendizado de Máquina . . . 30
2.4 Séries Temporais . . . 31

3 REVISÃO DA LITERATURA . . . 33
3.1 Contextualização . . . 33
3.2 Pesquisa Sistemática . . . 34
3.3 Trabalhos Relacionados . . . 34
3.4 Discussão do Capítulo . . . 36

4 SOLUÇÃO PROPOSTA . . . 37
4.1 Visão Geral . . . 37
4.2 Coletor de Dados . . . 38
4.2.1 Dados Fornecidos pela Prefeitura . . . 38
4.2.2 Dados do Twitter . . . 39
4.3 Pré-Processamento . . . 39
4.4 Processamento . . . 40
4.4.1 Gerar Série Temporal Padrão . . . 41
4.4.2 Detectar Anomalias . . . 43
4.4.3 Avaliar Anomalias . . . 45
4.5 Sistema de Alertas . . . 46

4.6 Arquitetura de Implementação Proposta 47
4.7 Considerações Sobre o Capítulo 48

5 ESTUDO DE CASO 49
5.1 Coleta dos Dados 49
5.2 Caracterização dos Dados de Mobilidade de Vitória 50
5.2.1 Descrição dos Dados 50
5.2.2 Caracterização dos Dados de Alertas 51
5.2.3 Caracterização dos Dados de Engarrafamentos 53
5.3 Clusterização de Séries Temporais 54
5.4 Considerações Sobre o Capítulo 56

6 AVALIAÇÃO DO ARCABOUÇO 59
6.1 Métricas de Avaliação 59
6.2 Avaliação Paramétrica 61
6.3 Resultados 61
6.4 Considerações Sobre o Capítulo 65

7 CONCLUSÕES 67
7.1 Perspectivas de Trabalhos Futuros 68

REFERÊNCIAS 73

1 Introdução

O uso de diversas fontes de dados, como câmeras de vigilância, sensores, dispositivos móveis e redes sociais para coleta de dados faz parte da realidade de grandes cidades do mundo (ALBINO, 2015). De fato, elas têm se mostrado ferramentas úteis para a promoção de uma vivência mais segura, confortável e sustentável (PANAGIOTOU et al., 2016; MONTORI et al., 2016; PURNOMO et al., 2016). Há ainda as informações compartilhadas por meio de plataformas colaborativas, que representam sistemas de sensoriamento participativo (*crowdsensing* em inglês), em que os dispositivos móveis dos cidadãos se tornam sensores na cidade (SILVA et al., 2016). Um exemplo desse tipo de plataforma é o aplicativo Waze[1], que utiliza dados de tráfego coletados pelos aparelhos dos usuários do sistema para traçar rotas e informar a situação do trânsito. Essa abordagem de coleta de informações permite observar o espaço urbano pela perspectiva civil dos indivíduos, de forma diversificada, expandida e profunda na cidade, algo dificilmente alcançado por meio de infraestrutura municipal.

A adoção de plataformas colaborativas por parte dos cidadãos têm crescido de forma que agentes públicos buscam cada vez mais o estabelecimento de parcerias com estes provedores de informação, enriquecendo a massa de dados para monitoramento em uma cidade e diminuindo a dependência de infraestruturas públicas legadas. Um exemplo é a cidade de Vitória-ES, que ocupou o primeiro lugar do *Ranking Connected Smart Cities*[2] (Cidades Inteligentes e Conectadas) de 2018, entre os municípios brasileiros com até 500 mil habitantes. A prefeitura possui uma central de videomonitoramento, a CIOM, conectada a várias câmeras na cidade[3], onde agentes públicos supervisionam diversos locais para que ações possam ser tomadas quando necessário. No entanto, aumentar esse fluxo de dados requer maior investimento em crescimento e manutenção de infraestrutura, e mesmo assim dificilmente atingindo um grande nível de capilaridade. Em paralelo, por meio de uma parceria com o Waze, a central também recebe informações sobre a situação do trânsito em tempo real, providas pelos motoristas que circulam pela cidade com o aplicativo. Interpretar toda essa massa de informações se torna uma tarefa árdua, fazendo-se necessária a implantação de soluções para análise integrada e eficaz dos dados de forma a melhorar o monitoramento de eventos e incidentes nas cidades (MONTORI; BEDOGNI; BONONI, 2017).

A partir dessa problemática, no contexto deste trabalho foi elaborado um arcabouço para coleta e análise de dados de fontes heterogêneas, que emite alertas de anomalias

[1] <https://www.waze.com/pt-BR>
[2] <www.connectedsmartcities.com.br/resultados-downloads-connected-smart-cities/>
[3] <www.folhavitoria.com.br/geral/noticia/08/2019/vitoria-tera-centro-de-inteligencia-para-monitoramento-de-servico

detectadas em séries temporais. Séries temporais são conjuntos de dados referenciados no tempo (WOOLDRIDGE, 2012). Ainda, entende-se aqui como anomalia qualquer observação que se desvia das demais (HAWKINS, 1980). Assim, as autoridades podem receber avisos sempre que alguma via apresentar comportamento incomum, neste caso, uma velocidade detectada aquém da faixa de valores esperada, levando em conta comportamentos sazonais. Consequentemente, os responsáveis pela vigilância da cidade podem focar em pontos da cidade onde incidentes podem de fato estar ocorrendo e suas atenções sejam mais requisitadas.

O arcabouço processa os dados coletados para construir um histórico de dados de fontes heterogêneas da cidade e, a partir deste, identificar as anomalias. Para isso, foi definida uma interface de modelo de dados composta por um *timestamp* com ao menos um atributo descritivo do dado. Embora o arcabouço foi projetado para identificar anomalias em qualquer dado de séries temporais, neste trabalho é utilizado o atributo de velocidade média nas vias da cidade. O arcabouço armazena toda informação coletada, de onde se obtém uma janela deslizante de D dias. Para cada dia dentro da janela deslizante, os dados são agrupados em intervalos $Jmin$ por meio de uma função ($Funtion$), para cada via observada. Assim, cada dia se torna uma série temporal que representa o comportamento de determinada via sob o aspecto, por exemplo, da velocidade média dos veículos em engarrafamentos. Em seguida, é aplicada uma técnica de agrupamento (clusterização) sobre dados dentro do mesmo intervalo $Jmin$, das séries, resultando em uma série temporal padrão de grupos (clusters). Por fim, os dados em tempo real de uma via são comparados, dentro do intervalo diário correspondente, com a série padrão e, caso estas sejam classificadas como anomalia, podem ser emitidos alertas para as autoridades responsáveis.

A fim de validar a anomalia detectada, parte-se da premissa de que se múltiplas fontes afirmam algo a informação tende a ser mais confiável. Sendo assim, o arcabouço permite utilizar outras fontes de dados para validar as anomalias identificadas. Dessa forma, consegue-se aumentar a confiabilidade dos resultados, permitindo um melhor planejamento urbano por parte das autoridades (RATHORE et al., 2016). Como prova de conceito, foi desenvolvido um protótipo para analisar os resultados obtidos a partir de combinações paramétricas, com quatro algoritmos de clusterização e duas técnicas estatísticas de detecção de anomalias.

1.1 Definição do Problema

A problemática abordada nesta dissertação é uma generalização do caso dos agentes da central de videomonitoramento em Vitória. Busca-se solucionar a ausência de um sistema de detecção e alerta de anomalias em tempo real, baseado nos dados disponíveis. Uma

anomalia, de modo geral, é uma observação que foge do comportamento esperado, de maneira que pode ter sido causada por algum mecanismo diferente do usual. Em se tratando de dados urbanos, como velocidade de tráfego na via, as anomalias podem representar consequências de eventos incomuns no trânsito, como acidentes, alagamentos ou obras na via. Assim, tais anomalias podem ser utilizadas como provável indicativo de que a localidade monitorada deva ser observada.

1.2 Objetivos

O objetivo deste trabalho é a formulação de um arcabouço que auxilie autoridades no monitoramento das diversas localidades de um município ou região, utilizando, inclusive, informações provenientes de plataformas colaborativas. Este arcabouço deve, mais especificamente, ser capaz de identificar situações anormais em tempo real, de maneira que os funcionários recebam alertas dos locais que possivelmente demandem intervenção. A fim de tirar proveito de dados que apresentem comportamento sazonal, o método utilizado deve levar em consideração os valores esperados para cada período do dia ao classificar uma observação. Além disso, a solução deve prever um alto fluxo de entrada de dados, visto que uma cidade/região possui tamanho variável. Desta forma, almeja-se um sistema que:

- Identifica anomalias e envia alertas em tempo real;
- Permite utilizar múltiplas fontes de dados para aumentar a confiabilidade das anomalias detectadas;
- Leva em consideração a sazonalidade do comportamento urbano;
- Seja escalável quanto ao tamanho da cidade/região.

1.3 Contribuições

As principais contribuições desta dissertação são:

- Propõe um arcabouço para análise de dados de mobilidade urbana baseado em fontes de dados heterogêneas;
- Propõe um modelo de série temporal que resume os dados em intervalos de tempo ao longo das 24 horas do dia, reduzindo a dimensão dos dados;
- Propõe o uso de técnicas de clusterização para criar uma série temporal padrão, permitindo acelerar o processo de detecção de anomalias;

- Implementa um sistema de detecção e alerta de anomalias em tempo real, avaliado com dados reais, obtidos por meio de uma parceria com a Secretaria de Segurança Urbana de Vitória (ES).

Destaca-se também a publicação do artigo "Um Arcabouço para Detecção e Alerta de Anomalias de Mobilidade Urbana em Tempo Real" no Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos 2020.

1.4 Organização do Texto

O restante deste texto está organizado como segue. O Capítulo 2 apresenta os conceitos básicos para a compreensão da solução e as tecnologias utilizadas. O Capítulo 3 discute os trabalhos relacionados. O Capítulo 4 detalha a solução proposta. O Capítulo 5 explicita a forma como os dados foram obtidos e apresenta uma visão geral das informações coletadas. A avaliação da solução é apresentada no Capítulo 6. Por fim, o Capítulo 7 conclui e discute as limitações da solução e possíveis trabalhos futuros.

2 Fundamentação Teórica

Neste capítulo são apresentadas as definições de algoritmos e tecnologias utilizadas neste trabalho. São apresentados, inicialmente, os algoritmos de clusterização e as técnicas de detecção de outlier, por serem considerados conceitos fundamentais da proposta. Em seguida, são explicitados alguns conhecimentos que devem auxiliar no entendimento da abordagem adotada na solução proposta: aprendizado batch, mini-batch e online, e séries temporais.

2.1 Algoritmos de Clusterização

Os algoritmos de clusterização, uma área de aprendizado de máquinas, visam classificar os dados por semelhança (XU; TIAN, 2015). Neste trabalho, buscou-se avaliar o protótipo desenvolvido utilizando um conjunto desses algoritmos, escolhidos dentre os disponíveis na ferramenta Sci-Kit Learn[1] (utilizada para prototipação), os quais são apresentados em seguida.

2.1.1 DBSCAN

O *Density-Based Spatial Clustering of Applications with Noise* (ESTER et al., 1996), DBSCAN, é um algoritmo não paramétrico (não assume que os dados seguem uma função) para agrupar dados com ruído baseado em densidade. Ele agrupa pontos em regiões de alta densidade (pontos muito próximos) e marca pontos em regiões de baixa densidade como outliers. Os principais parâmetros deste algoritmo são a distância máxima entre pontos para que sejam considerados vizinhos e o número mínimo de pontos para que uma região seja considerada densa. Ele inicia marcando pontos como núcleos, aqueles que possuem um número mínimo de vizinhos, dentro de uma distância máxima. Os núcleos vizinhos são incluídos no mesmo cluster juntamente com todos os pontos vizinhos normais. Pontos que não se aproximam de nenhum núcleo são dados como ruídos. No caso de um conjunto de dados muito próximos em relação à distância máxima entre vizinhos, o algoritmo possui complexidade $O(n^2)$.

2.1.2 OPTICS

O *Ordering Points to Identify the Clustering Structure* (ANKERST et al., 1999), OPTICS, é um algoritmo baseado em densidade, similar ao DBSCAN, mas que corrige a dificuldade em detectar clusters em dados com densidade variante. Para resolver esse

[1] <https://scikit-learn.org/stable/modules/clustering.html>

problema, inicialmente os pontos são linearmente ordenados, de forma que pontos espacialmente próximos se tornam vizinhos na ordenação. Assim como o DBSCAN, o algoritmo OPTICS possui como parâmetro o número mínimo de pontos para que uma região seja considerada densa e a distância máxima para que dois pontos sejam considerados vizinhos. Esta distância máxima, no entanto, pode ser abstraída, de forma que qualquer ponto seja considerado vizinho. Neste caso, a complexidade do algoritmo é $O(n^2)$. Normalmente, é desejável fornecer um limite superior para essa distância, o que acelera a execução e pode inibir situações que retornam todo o conjunto como um cluster só.

2.1.3 K-Means

O algoritmo de clusterização K-Means (MACQUEEN, 1967) funciona separando as amostras em K clusters, agrupando os pontos de acordo com a menor distância entre estes e os centros espaciais dos clusters. Possui basicamente dois parâmetros: o número de clusters, K, a ser aplicado aos dados e a forma de cálculo de distância entre pontos, que normalmente é a distância euclidiana. O algoritmo inicia selecionando K indivíduos e atribui um dos K clusters a cada um. Em seguida, utiliza a distância mínima entre cada dado restante aos centros dos clusters para incluir o dado no cluster de menor distância. Para encontrar a solução definitiva utilizando K-Means, a complexidade do algoritmo é $O(n^{kd+1})$, onde d é o número de dimensões dos dados, mas geralmente as implementações do algoritmo utilizam uma versão heurística, de complexidade $O(nkdi)$, onde d é o número de dimensões dos dados e i o número de iterações necessários para convergência.

2.1.4 BIRCH

O *Balanced Iterative Reducing and Clustering using Hierarchies* (ZHANG; RAMAKRISHNAN; LIVNY, 1996), BIRCH, é um algoritmo de clusterização hierárquico, particularmente eficiente em grandes conjuntos de dados, sendo necessário, normalmente, executá-lo apenas uma vez sobre os dados. Este algoritmo possui 4 etapas:

1. Construção de uma *Clustering Feature Tree* (árvore de cluster característica) a partir dos dados;

2. Opcionalmente, verificação de cada folha da *CF Tree* inicial a fim de construir uma *CF Tree* menor, removendo outliers e agrupando subclusters em outros maiores;

3. Utilização de algum outro algoritmo de clusterização (geralmente K-Means) sobre os nós folhas, o que resulta em um conjunto de clusters representando os diversos padrões de distribuição dos dados;

4. Os centros dos clusters encontrados na etapa anterior são usados como sementes e

os dados são redistribuídos às sementes mais próximas para obter um novo conjunto de clusters (esta etapa também oferece a opção de descartar outliers).

O parâmetro principal do algoritmo é o número de clusters a ser encontrado. Além disso, o algoritmo encontra bons resultados ao passar uma vez sobre os dados, alcançando melhores qualidades com múltiplas iterações sobre os dados, de forma que a complexidade do algoritmo é $O(n)$.

2.1.5 K-Means com DBA

O *DTW Barycenter Averaging* (PETITJEAN; KETTERLIN; GANÇARSKI, 2011) é um método para cálculo de média de séries temporais que utiliza o *Dynamic Time Warping* no cálculo de distância entre séries temporais (Gupta et al., 1996) e que leva em consideração possíveis variações temporais de comportamento entre séries. O cálculo de distância com DTW é composto pela menor soma total das distâncias entre as duas séries temporais, obedecendo as seguintes regras:

- todo índice de cada série deve se relacionar a um ou mais índices da outra série
- os primeiros e os últimos índices de cada série devem ser respectivamente mapeados entre si, mas podem ser mapeados a outros índices também
- a função de mapeamento entre índices de uma série à outra deve ser monótona

O algoritmo de clusterização K-Means baseado em DBA[2] se utiliza desse método para criar clusters como na abordagem tradicional, mas cujo método para cálculo de distância entre indivíduos é DTW e o cálculo do centro do cluster via DBA.

2.2 Detecção de Anomalias

Aqui são apresentadas duas técnicas de detecção de anomalias, Z-Score e IQR. Há várias formas de classificar elementos como anômalos, incluindo o uso de clusterização (HODGE; AUSTIN, 2004), porém, como partiu-se da ideia de encontrar padrões no comportamento dos dados para então os classificar como anomalias, foi possível restringir as técnicas utilizadas àquelas que poderiam classificar um indivíduo tomando um conjunto como base. Ambos métodos utilizam a diferença entre uma amostra e o centro de um conjunto de dados para classificar a amostra como normal ou anomalia.

[2] <tslearn.readthedocs.io/en/stable/gen_modules/clustering/tslearn.clustering.TimeSeriesKMeans.html>

2.2.1 Z-Score

O Z-Score, de maneira resumida, é um valor que indica o quão longe uma amostra está da média do conjunto, utilizando a distância euclidiana dividida pelo desvio padrão dos dados. Este método permite comparar quão longe uma amostra está, levando em consideração a escala de distância do conjunto. O cálculo é feito da seguinte forma: $z = \frac{x-\mu}{\sigma}$, sendo x a amostra, μ a média e σ o desvio padrão do conjunto. As amostras cujo valor absoluto de z é abaixo de 3 são consideradas normais, o que em um conjunto com distribuição normal representa 99% dos dados. Assim, dados cujo valor absoluto de z é acima de 3 são considerados anormais.

2.2.2 IQR

A distância interquartil é uma medida de dispersão estatística, baseada nos 50% centrais dos dados de um conjunto, calculada pela diferença entre os percentis 75 e 25, ou seja, $IQR = Q_3 - Q_1$. Ela determina quão esparso os dados são, em relação aos dados do "meio" do total. Neste método, dados que estão abaixo de $Q_1 - 1,5IQR$ e acima de $Q_3 + 1,5IQR$ são considerados anomalias.

2.3 Aprendizado de Máquina

Aprendizado de máquina é um conceito que engloba métodos computacionais adaptativos, cujo comportamento pode mudar com o aumento de experiência. Existem três grandes categorias de aprendizado de máquina: supervisionados, não supervisionados e por reforço. Os métodos supervisionados são aqueles que se utilizam de dados previamente classificados na construção do conhecimento, de forma que é possível treinar o sistema para classificar novos dados. Métodos não supervisionados utilizam dados sem classificação, sendo o processo de aprendizado normalmente associado ao reconhecimento de padrões nos dados. Já a aprendizagem por reforço compreende a construção de conhecimento pelo erro, de maneira que o método possui um *feedback* acerca de seu desempenho ao longo do tempo.

Em aprendizado de máquina, um outro conceito interessante, normalmente associado a redes neurais, é em relação às diferentes abordagens na construção do conhecimento[3]: batch, mini-batch e online. No aprendizado batch, o sistema possui acesso a todo o histórico de dados, visitando-o uma única vez, gerando conhecimento a partir de todas as informações. No caso do método mini-batch, o conhecimento é construído em partes, sendo o conjunto de dados dividido em janelas e o aprendizado desenvolvido a partir de cada uma. Por último, no método online as janelas são reduzidas ao tamanho de uma observação,

[3] <https://www.kaggle.com/residentmario/full-batch-mini-batch-and-online-learning>

de forma que o conhecimento é gerado a cada informação nova. A divisão dos dados em janelas, do aprendizado mini-batch, se assemelha à abordagem utilizada nesta dissertação, especialmente em relação à janela de histórico deslizante.

2.4 Séries Temporais

Uma série temporal é um conjunto de dados referenciados no tempo (WOOLDRIDGE, 2012), representando informações de eventos coletados ao longo do tempo, com taxas de amostragem igualmente espaçadas ou não. Este tipo de dado apresenta uma relação de ordenação no tempo, de forma que a sequência do conjunto possivelmente guarda associações importantes. Exemplos de séries temporais são: velocidade de veículos na via, precipitação na cidade e preços de ações. Uma propriedade interessante que se pode notar em dados de séries temporais é a autocorrelação, que é observada como a similaridade cíclica entre dados, ou seja, valores que aproximadamente se repetem com o tempo. Em dados de trânsito, esta propriedade pode ser observada no comportamento recorrente de dias úteis, por exemplo, com engarrafamentos em horários de pico. Outra característica dos dados de séries temporais é a sazonalidade, que é a variação de comportamento inerente a determinadas épocas, como a redução de veículos na rua em períodos de férias escolares.

3 Revisão da Literatura

Neste capítulo são apresentados os trabalhos que foram objetos de estudo. Inicialmente, na Seção 3.1, é feita uma contextualização com trabalhos relacionados a detecção de anomalias, mobilidade urbana e uso de múltiplas fontes em análise de dados. Em seguida, na Seção 3.2, são ilustrados os parâmetros e método utilizados para busca de artigos relacionados, determinando o posicionamento desta dissertação na literatura. Por fim, na Seção 3.3, são listados os trabalhos relacionados, tratando da detecção de anomalias com uso de técnicas de aprendizado de máquinas.

3.1 Contextualização

A detecção de anomalias é uma ampla área de estudos, focados em diversos domínios como detecção de fraudes, processamento de imagens e monitoramento em séries temporais (HODGE; AUSTIN, 2004). As técnicas de detecção de anomalias são usadas para ajustar, remover ou mesmo monitorar observações anômalas. Existem diversas abordagens para realização desta tarefa, classificadas em (AHMED; MAHMOOD; HU, 2016) como: classificatória, estatística, por teoria da informação e por clusterização. Em (HODGE; AUSTIN, 2004) se encontram exemplos em diversos cenários, de abordagens estatísticas, com redes neurais, com outras técnicas de aprendizado de máquinas e com sistemas híbridos, que utilizam ao menos duas das outras três abordagens.

Um conjunto de dados bastante explorado em análise de dados e detecção de anomalias é o de dados urbanos, especialmente de mobilidade urbana. A análise de informações e detecção de anomalias desta área podem ajudar do planejamento de transporte público (BALOIAN et al., 2015) à indicação de rotas seguras e rápidas (LADEIRA et al., 2019; SOUZA et al., 2018). (Bawaneh; Simon, 2019) apresenta um algoritmo de detecção de anomalias em séries temporais de ocupação da via, medida por laços de indução (comumente utilizado em radares), para encontrar situações anormais como as causadas por acidentes, obras na via ou simplesmente por conta do horário. Já (FAIAL et al., 2019) apresenta um método para detecção de anomalias em tempos de viagem a partir de dados passados reais, diferenciando os dias da semana. Em ambos os casos foi utilizada técnicas estatísticas para diferenciar dados normais de anômalos.

Alguns trabalhos abordam o uso de múltiplas fontes de dados para detecção de anomalias em dados urbanos. Por exemplo, (PAN et al., 2013) utiliza trajetórias de táxis obtidas a partir de dispositivos GPS instalados nos veículos, criando padrões e detectando anomalias nos percursos, e dados coletados no Twitter para tentar descrever a anomalia. Já (Sidauruk; Ikmah, 2018) utiliza informações provenientes da plataforma Twitter e do

Waze, analisando a correlação entre a publicação de determinados termos nos *tweets* e a velocidade da via, usando redes neurais.

3.2 Pesquisa Sistemática

A busca por artigos de teor semelhante ao desta dissertação foi realizada sistematicamente, utilizando os seguintes parâmetros de pesquisa:

- contém "anomaly" ou "outlier", e "detection", por se tratar de uma solução para detecção de dados anormais
- contém "time series" ou "stream", indicando uma solução que recebe uma sequência de dados
- contém "unsupervised", indicando que não pode haver classificação prévia dos dados
- contém "pattern", visto que a classificação das observações é feita a partir de um padrão
- contém "real time" ou "online", visto que a detecção deve ocorrer em tempo real

Quando necessário filtrar os resultados, os dois primeiros itens da lista podem ser incluídos como termos obrigatórios para o título ou resumo. A seleção dos textos ocorreu, inicialmente, de acordo com o resumo dos trabalhos, que deveria se encaixar no tema de detecção de anomalias com uso de alguma técnica de geração de padrão. Em seguida, foram filtrados pela introdução, em uma leitura superficial e, quando enquadrado no tema, uma leitura completa era realizada. Assim, obteve-se os trabalhos apresentados na seção seguinte.

3.3 Trabalhos Relacionados

Nesta seção são apresentados trabalhos fortemente relacionados quanto à abordagem utilizada para detecção de anomalias. De modo geral, os métodos apresentam a geração de um padrão de normalidade ou de observações anômalas, a partir do agrupamento de dados, e da utilização deste para classificação de eventos.

No trabalho (IZAKIAN; PEDRYCZ, 2014) é apresentada uma técnica para detectar anomalias que envolve clusterização espaço-temporal fuzzy. O artigo generaliza o desenvolvimento do padrão de modo a abranger os casos em que as observações não necessariamente possuem distinta diferença espacial, para, então, detectar anomalias. Nesta dissertação, os dados utilizados podem ser claramente divididos espacialmente por vias, favorecendo a redução de escopo da solução para cada via da cidade.

As técnicas apresentadas por (AHMAD et al., 2017) e (WU; ZENG; YAN, 2018) possibilitam classificar observações como anomalias utilizando *Hierarchical Temporal Memory*, HTM, uma tecnologia de aprendizado baseado em memória. As redes neurais são utilizadas na identificação de padrões de ativação no decorrer do tempo, permitindo o reconhecimento de novos padrões sequenciais e, assim, gerando um modelo de predição. Enquanto o segundo artigo utiliza uma abordagem puramente baseada no HTM, o primeiro possui um passo seguinte, calculando a probabilidade de erro na predição e da classificação da observação em anomalia. Em ambos casos, não existe consideração do horário ao classificar novos dados, de forma que a classificação de uma informação pode ser influenciada por dados de outros horários, ao contrário desta dissertação.

Em (ALAM et al., 2019) é apresentado um método para detecção de anomalias baseado em clusterização e distância interquartil. Inicialmente, os dados obtidos são reduzidos, utilizando a mediana, em *slots* de 15 minutos, gerando séries temporais diárias com 96 pontos cada. Em seguida, as séries temporais de todas as fontes são clusterizadas por meio do K-Means, usando a distância DTW, resultando em séries temporais padrão. Por fim, classificam-se os dados por meio de distâncias entre percentis, permitindo que os limiares sejam definidos por um especialista ou automaticamente. Neste ponto, as observações não são diferenciadas quanto à localização, por exemplo, de modo que dados de locais distantes podem influenciar o resultado local. Além disso, o uso do DTW favorece o distanciamento de séries temporais por conta de mudanças de comportamento em algumas das faixas da série. Similarmente, (THUY; ANH; CHAU, 2018) utiliza uma distância entre séries temporais para clusterizar as sequências de observações, com a diferença que realiza uma re-amostragem das séries temporais originais, gerando subsequências que são homogeneizadas para, enfim, aplicar a clusterização e detecção de anomalias. Utilizando o DTW na clusterização, é possível que dados anômalos influenciem o padrão gerado, visto que as séries temporais ainda podem se aproximar de outras, formando o maior cluster. Nesta dissertação, ao utilizar clusterização unidimensional em dados de cada faixa de horário do histórico, dados que são possivelmente anômalos em uma faixa de horário não interferem no padrão de outra faixa de horário.

Em (BAEK et al., 2020) é proposto detectar anomalias em conjuntos de dados por meio de uma abordagem híbrida, um método não supervisionado para identificar grupos de dados, de forma que grupos pequenos e densos são classificados como anômalos, e um método supervisionado, que utiliza as classes obtidas pelo passo anterior para detectar anomalias. O primeiro passo é realizado por meio de clusterização, enquanto para o segundo alguns algoritmos supervisionados são utilizados, como SVM e Random Forest. Esta abordagem essencialmente se assemelha à desta dissertação no sentido que primeiro há uma etapa de classificação, realizada por um método não supervisionado, seguida de uma detecção de anomalia. A maior diferença se encontra no tipo de grupo procurado na primeira etapa, onde o artigo busca grupos de anomalias, enquanto neste texto o interesse

está nos dados normais, não anômalos, utilizando este padrão na detecção das anomalias.

Por fim, em (CALIKUS et al., 2020) é apresentado um framework generalista para detecção de anomalias, compatível com uma abstração das etapas tomadas nesta dissertação: adaptação da entrada, busca por um padrão de dados comuns, cálculo de distância de observações ao padrão e classificação das observações. O artigo prossegue com a escolha de algumas tecnologias para cada passo, avaliando diversas combinações. Na segunda etapa, no entanto, os métodos apresentados não consideram comportamentos cíclicos, de modo que não é possível reconhecer circunstâncias esperadas em função de cada período de um dia. Isso significa que o cálculo de anormalidade não necessariamente conta com a expectativa das observações naquele momento, ou seja, não considera os valores esperados em determinado período do dia ao classificar os dados recebidos. Nesta dissertação, ao utilizar um padrão de clusters por faixas de horário, é possível observar os valores normais para cada período do dia.

3.4 Discussão do Capítulo

Nesta dissertação é apresentado um método para detecção de anomalias baseado na geração de uma série temporal padrão, por meio de clusterização, e no uso desta para classificação de dados em tempo real. Diferentemente dos trabalhos apresentados, a técnica proposta utiliza uma superposição de dados diários, de acordo com o horário, para encontrar um padrão de eventos normais. Desta forma, a detecção de anomalias se beneficia da sazonalidade presente no comportamento dos dados de mobilidade na cidade. Além disso, é previsto o uso de múltiplas fontes no sistema visando o aumento de confiabilidade dos resultados. O próximo capítulo descreve o arcabouço desenvolvido para detecção de anomalias em tempo real.

4 Solução Proposta

Neste capítulo é apresentado o arcabouço proposto para detecção de anomalias em tempo real. O capítulo inicia com uma visão geral da proposta na Seção 4.1, em seguida são detalhados cada um dos módulos nas Seções 4.2 à 4.5. A Seção 4.6 aborda as tecnologias envolvidas no desenvolvimento do protótipo. O capítulo é finalizado com uma discussão acerca da proposta na Seção 4.7.

À medida que a solução é detalhada, são apresentados exemplos e resultados obtidos a partir da execução do protótipo, utilizando os dados obtidos na parceria com a prefeitura de Vitória, relativos à velocidade média nas vias quando ocorre congestionamento (não há registro de velocidade quando a via não possui engarrafamentos). Ressalta-se aqui que a base de dados é detalhada no capítulo seguinte.

4.1 Visão Geral

As anomalias são observações que fogem do comportamento esperado dos dados e que, neste trabalho, podem representar eventos atípicos de trânsito como acidentes e obras. A solução proposta tem como objetivo detectar anomalias em tempo real, baseando-se no histórico de eventos coletados de fontes heterogêneas. A Figura 1 ilustra a arquitetura modular do arcabouço, que permite uma implementação distribuída. Cada módulo é independente e possui funcionalidades bem definidas:

i) Coletor de Dados: define um modelo de interface de dados que permite obter dados de várias fontes e coleta estes.

ii) Pré-Processamento: disponibiliza os dados aptos a serem processados, removendo observações que não agregam valor ao resultado.

iii) Processamento: gera um padrão do comportamento dos dados usando clusterização e identifica anomalias.

iv) Sistema de Alertas: emite alertas de anomalias detectadas aos interessados.

As Seções 4.2, 4.3, 4.4 e 4.5, a seguir, detalham cada um dos respectivos módulos listados.

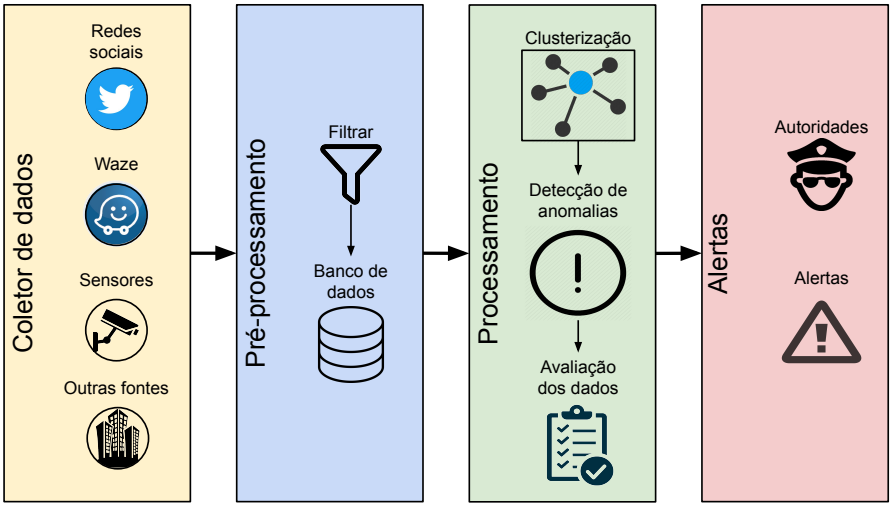


Figura 1 – Visão geral do arcabouço

4.2 Coletor de Dados

Este módulo é responsável por coletar dados de uma ou mais fontes em tempo real, implementando os mecanismos necessários a cada uma delas, e os entrega para a etapa de Pré-Processamento. Assim, cada instância se especializa em alguma fonte de dados, utilizando-se de informações públicas e/ou privadas, de forma que a implantação pode ser, ou envolver, um bloco de código compilado, uma caixa preta. De forma geral, foi definida a seguinte interface de modelo de dados, utilizada pelos módulos seguintes:

$$< timestamp,\ id_fonte,\ E = [E_1,\ E_2,\ ...,\ E_n] >$$

sendo *timestamp* a data e hora do evento, id_fonte a identificação da fonte e $E_{1..n}$ um conjunto de atributos descritivos do evento, por exemplo a localização e o tipo do evento. Desta forma, é possível implantar coletores específicos para cada fonte. Neste trabalho foram implementados coletores para duas fontes de dados, a API da prefeitura e a rede social Twitter. A coleta e os dados são detalhados no Capítulo 5.

4.2.1 Dados Fornecidos pela Prefeitura

A primeira fonte de dados é uma API de informações disponibilizados pela prefeitura de Vitória, relativos ao aplicativo Waze, onde se encontram três acessos: aos alertas de usuários acerca de incidentes, às velocidades médias de veículos em trechos engarrafados das vias e às avaliações de alertas enviados por usuários. No segundo tipo de dado, por exemplo, o conjunto E contém, entre outros, o nome da via e a velocidade média do trecho congestionado, de forma que a interface de modelo de dados é:

$$< eventDate,\ \text{api_vitoria_waze},\ id,\ city,\ level,\ length,\ turnType,\ type,\ speed,$$

$$roadType, delay, street >$$

Exemplos de observações são:

<2018-09-10T22:59:00, api_vitoria_waze, 1059610, Vitória, 3, 1087, NONE, NONE, 4.061111, 2, 144, R. Carlos Martins>

<2019-10-16T13:50:00, api_vitoria_waze, 3090013, Vitória, 3, 501, NONE, NONE, 2.822222, 2, 106, Av. Rio Branco>

4.2.2 Dados do Twitter

Para a segunda fonte de dados, o Twitter, foi implementado um *crawler* que busca publicações de contas oficiais de canais da imprensa e instituições públicas, com notícias do trânsito de Vitória. Esse *crawler* retorna todos os *tweets* de contas específicas, com determinados *hashtags* e dentro de um intervalo de tempo. Neste caso, o conjunto E contém o *tweet* completo:

$$< timestamp, \text{twitter_vitoria}, conta, hashtags, tweet >$$

Por exemplo:

<2019-10-14T20:26:15, twitter_vitoria, CBN Vitória, [CDT], "#CDT - Vitória: trânsito muito intenso nos acessos à Terceira Ponte pela Reta da Penha, Nossa Senhora dos Navegantes e Dukla de Aguiar.">

Outras fontes de dados, como metadados de imagens de câmeras de vídeo ou dados de laços indutivos, poderiam ser utilizadas para alimentar o sistema. No entanto, reconhecimento de padrões em imagens, análise de linguagem natural ou o tratamento de outras fontes de dados estão fora do escopo do projeto.

4.3 Pré-Processamento

Este é o módulo responsável por filtrar e armazenar os dados recebidos do Coletor de Dados. Esta funcionalidade foi concebida separadamente da obtenção de informações para permitir que um mesmo coletor possa servir a múltiplos filtros e para incluir a tarefa de armazenamento de dados, também intrínseca ao sistema.

Neste trabalho foram implementados tratamentos para dados obtidos a partir da API da prefeitura e do Twitter. Vale ressaltar que implementações para outras fontes de dados poderiam ser incluídas, como metadados de imagens de câmeras de videomonitoramento ou dados de sensores pluviométricos. No caso da API da prefeitura, dados que não

são de interesse são removidos, de forma que a interface de dados apresentada no módulo anterior é aqui reduzida a:

$$< eventDate, \text{api_vitoria_waze}, id, city\ speed, street >$$

Posteriormente, observações que não tratam de Vitória são retiradas, retirando também aquelas sem localização ou sem descrição dos eventos. Alguns exemplos de observações descartados são (na interface de dados reduzida):

<2018-09-11T09:18:00, api_vitoria_waze, 1060983, NULL, 6.372222, Segunda Ponte>

<2019-10-16T11:18:00, api_vitoria_waze, 3089229, Vila Velha, 1.430556, Av. Sen. Robert Kennedy>

<2018-09-10T23:19:00, api_vitoria_waze, 1059663, Vitória, 0.552778, NULL>

Os dados válidos são incluídos no banco de dados e enviados ao módulo Processamento.

Devido à natureza textual do Twitter, os dados (*tweets*) não possuem estrutura bem definida, isto é, as informações sobre qual via e o que ocorreu na mesma não seguem um padrão. A fim de obter informações precisas acerca de possíveis eventos de trânsito, o uso de processamento de linguagem natural seria necessário, fugindo do escopo deste trabalho. Por isso, no protótipo, os *tweets* são filtrados apenas pelo nome da via encontrado na publicação e armazenados para consulta sob demanda pelo módulo Processamento. Uma possível validação das anomalias detectadas utiliza estes dados, verificando se uma anomalia detectada a partir de dados provenientes do Waze também aparece no Twitter.

4.4 Processamento

O principal objetivo do módulo Processamento é classificar observações recebidas diretamente do Pré-Processamento e enviar notificações ao Sistema de Alertas em tempo real. O processo é dividido em três etapas:

- Gerar série temporal padrão: calcula o maior cluster por período do dia, baseado nas séries temporais de um período de D dias de observações, obtendo uma única série temporal;
- Detectar anomalias: compara observações em tempo real com a série padrão para as identificar como anomalia; e
- Avaliar anomalias: analisa o desempenho da detecção de forma automática, checando se existe alguma ocorrência sobre a anomalia em outras fontes de dados.

A implementação do protótipo considerou os dados da prefeitura, relativos ao aplicativo Waze, para calcular as séries temporais devido à sua integridade e completude. Por outro lado, os dados coletados no Twitter foram utilizados para avaliar as anomalias detectadas, ainda que por um método simples.

4.4.1 Gerar Série Temporal Padrão

Algoritmo 1: Algoritmo para agrupar e clusterizar dados

Input : $dados_via$, dia_atual, D, $Jmin$, `Function`, `Clustering`
Output : $pattern$

```
   // Separa os dias em dias de semana e fins de semana
1  Dados ← dados_via[dia_atual − D : dia_atual − 1]
   // Function retorna um vetor hora-dados
2  foreach dia in Dados do
3  |  Dados_reduzidos ← Function(dia, Jmin)
4  end
   // Clusteriza dados de todos os dias da mesma hora
5  foreach hora in Dados_reduzidos.horas do
6  |  pattern ← Clustering(Dados_reduzidos[hora])
7  end
```

Para detectar uma anomalia em tempo real, o Processamento precisa comparar um novo evento com um histórico de observações. Por exemplo, ao receber uma velocidade em uma via, o sistema deve ser capaz de identificar se ela está na faixa de valores esperados para o horário ou não. Assim, o objetivo desta etapa é obter uma série temporal padrão que identifique o comportamento de uma via em cada período do dia, usando uma técnica de clusterização, como descrito no Algoritmo 1.

Parâmetro	Descrição
$dados_via$	Dados de uma via
D	Quantidade de dias que serão considerados como histórico
dia_atual	Data do dia atual
$Jmin$	Tamanho da janela em minutos para divisão do dia em períodos
$Function$	Função para agregar as observações dentro de cada janela
$Clustering$	Método de clusterização a ser aplicado sobre as séries temporais diárias

Tabela 1 – Parâmetros do Algoritmo 1

Inicialmente, o algoritmo seleciona as informações de D dias, baseado em dia_atual, e as separa em dias de semana e fins de semana (Linha 1)[1]. A Figura 2 ilustra observações de velocidade média em dois dias úteis em uma via. Os eventos dentro de cada janela $Jmin$ são agregados por meio de uma função $Function$ (Linhas 2-4), como soma, média,

[1] O agrupamento de dados em meio de semana e fim de semana é justificado na Seção 5.2.2.

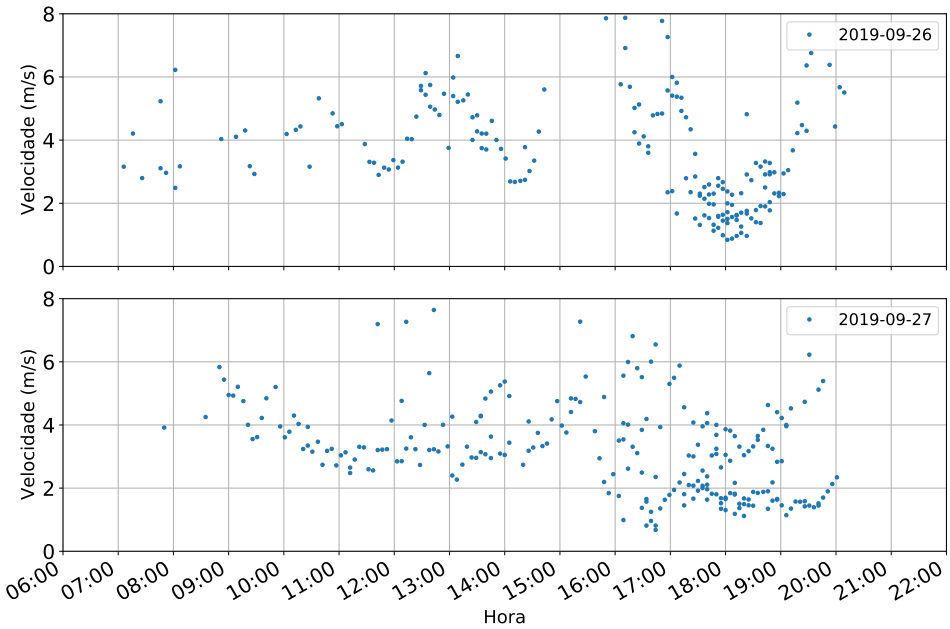


Figura 2 – *dados_via* para os dias 26 e 27 de Setembro de 2019

mediana e contagem de elementos enumeráveis dentro do intervalo. A Figura 3 ilustra o resultado da aplicação da função média dentro de cada janela sobre os dados numéricos dos mesmos dois dias, sendo a linha azul a série temporal do dia. Importante adiantar que, nos dados disponibilizados, quando não há engarrafamentos, não há registro de velocidade.

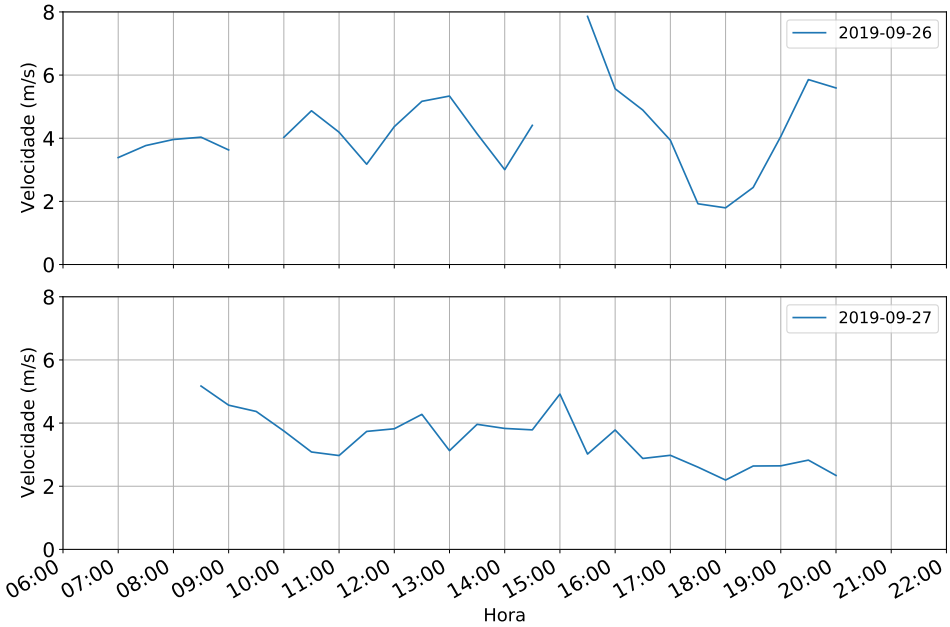


Figura 3 – Séries Temporais Diárias após *Function* nos dias 26 e 27 de Setembro de 2019

Os dados de cada série temporal em uma mesma faixa de horário são agrupados por um algoritmo de clusterização *Clustering* (Linhas 5-7), que retorna o maior *cluster* de cada período, resultando em uma série temporal padrão (*pattern*). Assim, evita-se que

dias atípicos, como feriados, influenciem a série padrão, dado que apenas o maior grupo de dados semelhantes são selecionados. Exemplificando, ao aplicar a clusterização sobre as velocidades médias entre 9:00 e 9:30 de cada dia de semana, serão obtidas as velocidades mais comuns de se encontrar naquela faixa de horário. A Figura 4 ilustra os dados das séries temporais diárias, clusterizadas unidimensionalmente por período do dia, e as séries temporais padrão obtidas com base nos dias 26 e 27 de Setembro de 2019, utilizando o protótipo com um histórico de 30 dias, K-Means com 3 clusters e janela de 30 minutos. Em cada faixa de horário (linha vertical), cada ponto corresponde ao dado de uma série padrão diária, e sua cor (aqui roxo, marrom e salmão claro) indica o cluster ao qual ele pertence. Apesar das cores se repetirem em múltiplas faixas de horário, elas não são relacionadas, ou seja, um ponto marrom na faixa de 08:00 não está relacionado a pontos marrons de outros horários. Além disso, as linhas, em laranja, indicam o centro do maior cluster em cada período, formando a série temporal padrão.

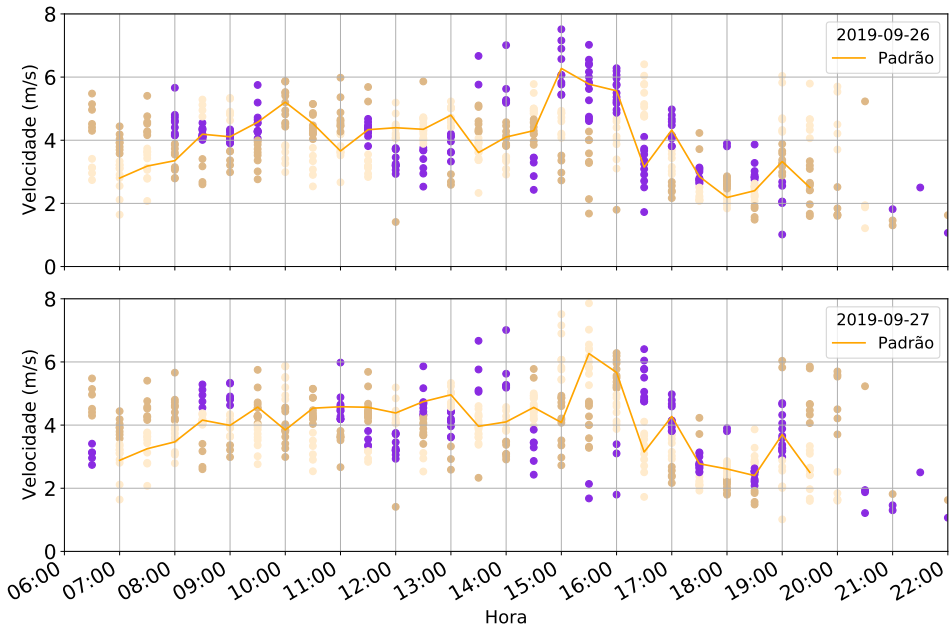


Figura 4 – Clusters em cada Período do Dia e Série Padrão Usando K-Means com 3 Clusters nos dias 26 e 27 de Setembro de 2019

4.4.2 Detectar Anomalias

Embora algoritmos de clusterização possam ser utilizados para detectar anomalias (o DBSCAN e o OPTICS, por exemplo, identificam pontos ruidosos), o custo computacional para a tarefa em tempo real seria alto. Por isso, a ideia principal é utilizar um grupo de dados com valores considerados comuns para a faixa de horário e avaliar cada novo evento a partir do conjunto de observações. Desta forma, classificam-se os dados de acordo com a distância até o centro do cluster de valores considerados normais.

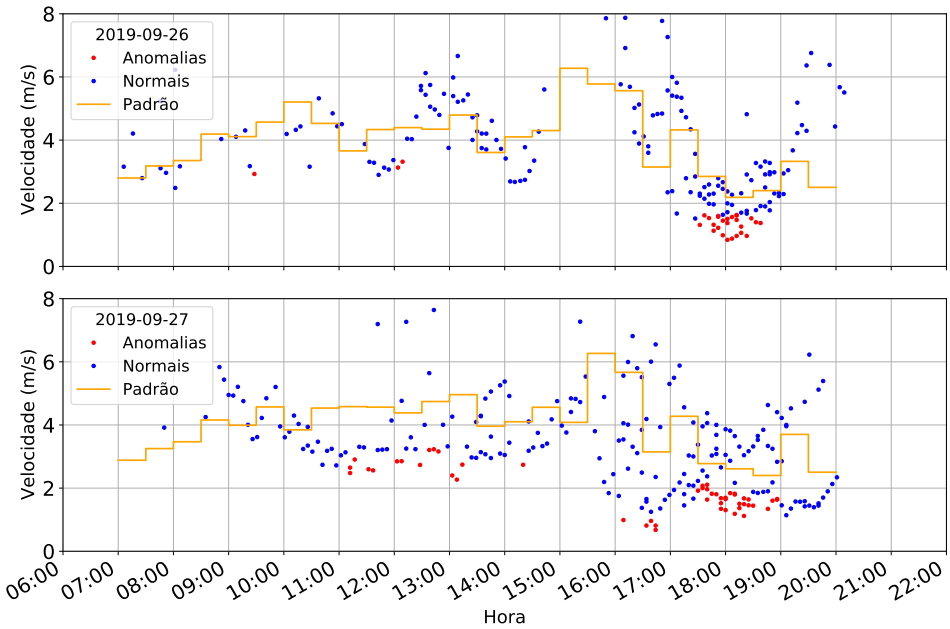


Figura 5 – Observações de 26 e 27 de Setembro de 2019 Classificadas como Normais e Anomalias Usando K-Means com 3 Clusters, Janela de 30 Minutos e Histórico de 30 Dias

No protótipo implementado, foram utilizadas duas técnicas para classificação de dados em anomalias: Z-Score e IQR. O Z-Score utiliza a distância da nova observação ao centro do cluster, normalizada pelo desvio padrão do cluster, para classificar a nova observação. O IQR também utiliza a distância entre a nova observação e o centro do cluster, mas o limiar se baseia na distância interquartil, entre os quartis 75 e 25.

A Figura 5 ilustra as anomalias detectadas com Z-Score nos dias 26 e 27 de Setembro de 2019, usando K-Means com 3 clusters, janela de 30 minutos e histórico de 30 dias. Nota-se que dados mais distantes da série padrão, representada pela linha laranja, são classificados como anomalias e que esta distância limite varia. Além disso, de um dia para o outro, percebe-se alteração tanto na série padrão como nos horários em que há anomalias detectadas. Isso ocorre pelo fato da alteração do histórico, descartando o dia mais antigo do histórico e adicionando o mais recente, gerando outra série padrão.

A sazonalidade de eventos urbanos possui certa variância, de forma que um incidente de trânsito, como engarrafamento, pode acontecer um pouco antes ou depois do horário usual. Por isso, é previsto nesta etapa a utilização opcional, também, das janelas imediatamente anterior e posterior à observação, para avaliar se uma observação será caracterizada como anomalia. Por exemplo: se um dado recebido se refere a uma observação realizada às 9:20, considerando uma janela de 30 minutos, pode-se avaliar este dado nas janelas de 9:00 a 9:30, e também de 8:30 a 9:00 e de 9:30 a 10:00. Se o resultado for positivo para anomalia nas três, o dado é considerado anômalo. Procura-se, assim, evitar que observações sazonais com algum desvio de tempo sejam erroneamente classificados

como anomalias.

Neste trabalho, os dados utilizados para análise e prototipação são velocidades médias nas vias da cidade (ressalta-se que o arcabouço permite utilizar outros dados), de forma que uma observação de valor maior que a série padrão, por mais longe que esteja, não é considerada anomalia, pelo simples fato de que uma via com tráfego melhor que o esperado não deve indicar uma situação que necessite de intervenção. Sendo assim, é possível traduzir a detecção de anomalias, ainda que de maneira simplória, para detecção de velocidades abaixo da faixa de valores normalmente encontrados na via. Ou seja, na Figura 5, pode-se dizer que, com base nos dados dos últimos 30 dias relativos aos dias 26 e 27 de Setembro de 2019, respectivamente, as observações recebidas marcadas em vermelho indicam velocidades muito abaixo do normal no respectivo horário, de forma que pode ser consequência de um evento não esperado, como um acidente.

4.4.3 Avaliar Anomalias

Anomalias encontradas na etapa anterior podem, de alguma forma, ser corroboradas com dados de outras fontes, como aplicativos, câmeras de videomonitoramento, sensores, redes sociais, etc. Estes dados podem ser utilizados como uma "segunda opinião", aumentando a confiabilidade do alerta que está para ser gerado. Por exemplo: a partir de uma detecção, é possível realizar uma busca (via API, banco de dados, etc) por algo que possa vir a corroborar a anomalia, como um metadado de visão computacional a partir de imagens de câmeras de videomonitoramento confirmando um acidente, um tweet informando uma obra, um alerta de enchente, etc. Outra das possibilidades de avaliação

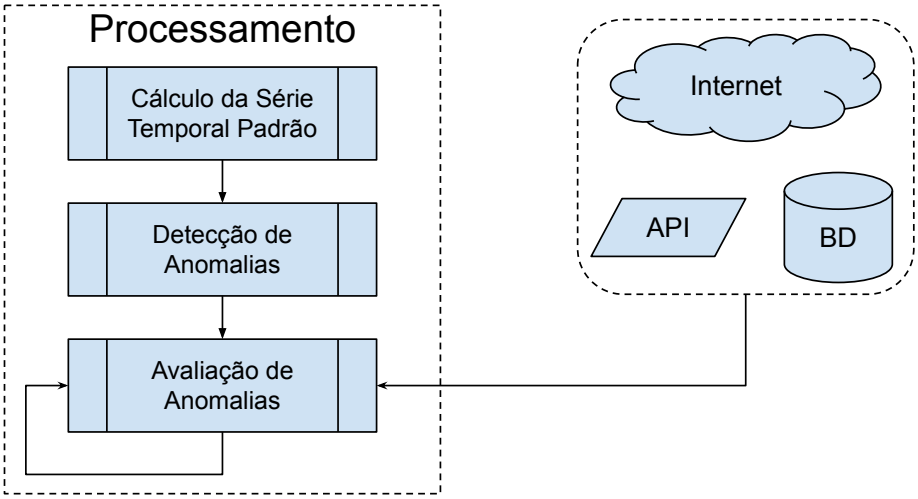


Figura 6 – Possíveis Fluxos de Informação na Avaliação de Anomalias

das anomalias detectadas, como ilustrado na Figura 6, é a utilização de detecções anteriores, de forma que a anomalia atual pode receber maior criticidade de acordo com o

número de detecções em sequência. Se, por exemplo, cinco anomalias forem detectadas consecutivamente, a última detecção poderia ter um grau de severidade maior, indicando um alerta de maior prioridade.

Este módulo, no protótipo, foi utilizado como um meio de avaliar a detecção de anomalias, como é apresentado no Capítulo 6, a partir de dados provenientes do Twitter. O Twitter é uma plataforma colaborativa de publicação de informações gerais, com participação de cidadãos e de autoridades, em que há canais especializados em determinados segmentos de noticiário, como mobilidade urbana. No caso específico de Vitória, é possível encontrar informações atuais acerca de eventos de trânsito na cidade no CBN Vitória[2], Eco101[3] e PRF ES[4]. A avaliação é feita da seguinte forma: a cada anomalia detectada, o sistema verifica se há um *tweet* publicado recentemente, de alguma das fontes citadas, que contém o nome da via referente à anomalia e, caso positivo, a detecção é dada como um acerto. Os próprios *tweets* poderiam ser utilizados, eventualmente, como fontes primárias de detecção de anomalias, mas, para isso, seria necessário realizar um processamento de linguagem natural, o que foge do escopo deste trabalho. Além disso, notou-se durante a avaliação do protótipo que, na cidade de Vitória, não há um constante engajamento nesta mídia social, o que dificulta o uso desta fonte para detecção de anomalias no método abordado neste trabalho.

4.5 Sistema de Alertas

O módulo Sistema de Alertas permite distribuir mensagens com avisos sobre as anomalias para os interessados. Este módulo possui uma conexão com o exterior, de forma a enviar alertas das anomalias detectadas, com base na implantação das particularidades de cada meio de comunicação, e recebendo inscrições de clientes que receberão as notificações dos alertas. Um exemplo de meio de envio de alertas é um simples concentrador (*broker*) MQTT[5]. O MQTT implementa um protocolo de comunicação M2M/IoT que segue o modelo *publish-subscribe*, no qual determinados nós publicam (*publishers*) informações em um tópico em um concentrador, que, por sua vez, encaminha a informação para todos os assinantes (*subcribers*) do tópico. Assim, os interessados em receber os alertas podem assinar o tópico de anomalias, recebendo alertas de eventos que estejam ocorrendo fora do esperado para uma via em um determinado horário. Outros exemplos de envio de notificações são email e SMS.

[2] <https://twitter.com/cbnvitoria>
[3] <https://twitter.com/_eco101>
[4] <https://twitter.com/PRF191ES>
[5] <http://mqtt.org>

4.6 Arquitetura de Implementação Proposta

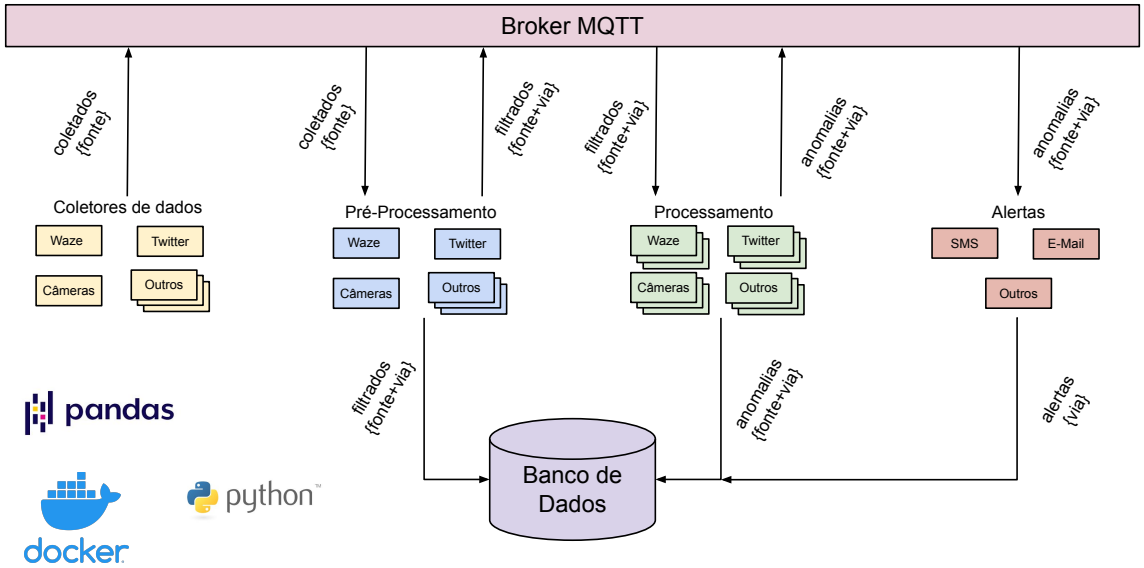


Figura 7 – Proposta de Fluxos de Informação

Um primeiro protótipo foi implementado utilizando a plataforma Colab do Google[6], de forma a facilitar a visualização dos dados e o desenvolvimento do software em partes. Foram utilizadas as bibliotecas NumPy[7] para importação dos dados, Pandas[8] para análise e processamento dos dados e Scikit-learn[9] para clusterização. A fim de simplificar a implementação, o protótipo foi construído como uma classe que aplica todos os métodos necessários sobre os dados: filtra, reduz em séries diárias, clusteriza e classifica as anomalias.

A partir do protótipo foi definida uma arquitetura modularizada, ilustrada na Figura 7. A intenção é permitir que cada módulo seja implantado separadamente, de maneira que a alocação de recursos seja facilitada. O interesse na divisão das funcionalidades surge diante da possibilidade de escalar o sistema usando algum parâmetro como base, como a via da cidade e a fonte do dado.

No trabalho desenvolvido, as vias foram utilizadas para especializar os containers de processamento. Assim, a quantidade (e consequentemente a abrangência) de containers ativos pode ser definida automaticamente, de acordo com o número de fontes de dados e com um ranking das vias mais importantes ou com base nos recursos disponíveis, ou manualmente, onde as autoridades escolheriam quais fontes serão observadas e quais avenidas devem ser monitoradas.

Para gerenciar o tráfego de dados entre os múltiplos containers, foi previsto o uso de um servidor MQTT, organizando o envio, recebimento e divisão das informações por meio

[6] <https://colab.research.google.com/notebooks/intro.ipynb>
[7] <https://numpy.org/>
[8] <https://pandas.pydata.org/>
[9] <https://scikit-learn.org/stable/>

de canais, que podem ser nomeados de acordo com algum parâmetro. Na Figura 7, as setas representam os canais previstos para envio e recebimento de informações, orquestrados pelo servidor MQTT. Neste exemplo, as fontes de dados e vias da cidade compõem o grupo de parâmetros para os containers.

O banco de dados nesta arquitetura é utilizado para armazenar todas as informações filtradas, de modo que o Processamento possa acessar as observações imediatamente quando necessário, sem a necessidade de implementar um método de persistência individual. Assim, o Processamento tem acesso às observações em tempo real, à medida que chegam ao sistema, classificando-as, e às anteriores, por meio do banco de dados, que permite o desenvolvimento da série temporal padrão. Mais ainda, o banco de dados pode ser utilizado para armazenar as anomalias detectadas, facilitando futuras análises pontuais e permitindo a criação de níveis de severidade de anomalias.

4.7 Considerações Sobre o Capítulo

Este capítulo apresentou o arcabouço do sistema proposto para detectar anomalias em tempo real e envio de alertas. A divisão da tarefa em módulos, como ilustrado na Figura 1, visa as vantagens da computação distribuída, de forma que para cada unidade é possível alocar recursos dinamicamente de acordo com a demanda, que pode variar com o tempo. Ainda, os módulos como funcionalidades isoladas permitem a especialização e a externalização de cada um, ou seja, sua substituição por um acesso externo. Por exemplo: a coleta de dados pode ser realizada por uma aplicação externa, privada, que envia as informações diretamente ao sistema para processamento.

Uma das limitações do sistema proposto é a dependência da sazonalidade das informações, ou seja, os eventos diários devem, em sua maioria, ser correspondentes. Isto significa que mudanças de comportamento não são absorvidas rapidamente, a depender do número de dias utilizados para construção do padrão. Além disso, a classificação das observações em anomalias não é aprofundada, não havendo distinção entre tipos de anomalias, tampouco suas causas. Ainda neste sentido, outra limitação é quanto à avaliação da relação entre anomalia e incidente de trânsito, visto a possibilidade de que um não esteja conectado ao outro na realidade. Por fim, para que o sistema funcione, é necessária uma certa quantidade de dados coletados previamente, o suficiente para o cálculo das séries temporais padrão.

5 Estudo de Caso

Este capítulo apresenta um estudo de caso para avaliar o arcabouço proposto, visando igualmente mostrar a aplicabilidade da solução no contexto do monitoramento da cidade de Vitória.

Além do acesso às imagens de diversas câmeras distribuídas pela cidade, a central de monitoramento de Vitória recebe dados obtidos pelo aplicativo colaborativo de trânsito, Waze, de forma anonimizada e agregada. Esses dados, provenientes dos usuários do aplicativo, indicam a situação das várias vias da cidade e foram utilizados no protótipo para avaliação. Apesar do uso *offline* na avaliação, ou seja, utilizando dados coletados previamente, o protótipo funciona de maneira *online*, classificando novas observações individualmente, à medida que são recebidas, e as inserindo no banco de dados para construção de conhecimento. Em outras palavras, mesmo que os dados estejam todos disponíveis no momento da execução, o fato de se ter uma janela deslizante acaba resultando em um comportamento similar ao do funcionamento em um ambiente real, com *streaming* de dados.

A primeira seção deste capítulo, 5.1, mostra como estas informações foram coletadas durante o desenvolvimento do trabalho. Em seguida, a Seção 5.2 descreve e ilustra os dados de alertas de usuários e de engarrafamentos disponibilizados, dando uma dimensão da quantidade e da qualidade das informações disponíveis, permitindo vislumbrar um caminho para a solução. Por fim, a Seção 5.3 discorre acerca das análises e métricas utilizadas sobre os dados ao passo que o protótipo foi sendo desenvolvido.

5.1 Coleta dos Dados

A obtenção dos dados deste trabalho ocorre por meio de uma parceria entre a Secretaria de Segurança Pública da Prefeitura Municipal de Vitória e a Universidade Federal do Espírito Santo, visando o desenvolvimento de soluções que possam auxiliar e melhorar os serviços de monitoramento municipais. Um destes é realizado pela CIOM[1], a central de videomonitoramento da Guarda Municipal de Vitória, onde pontos estratégicos no município são supervisionados. O trabalho dos agentes nessa central demanda alto foco, por isso possui grande potencial de fadiga (BOKSEM; MEIJMAN; LORIST, 2005). O trabalho foi concebido com o intuito de auxiliar essa função.

Foram implementados coletores de dados para duas fontes: mobilidade urbana, via API da prefeitura de Vitória, e rede social Twitter. Por meio do convênio, a prefeitura

[1] <https://www.vitoria.es.gov.br/prefeitura/cameras-de-videomonitoramento>

disponibilizou uma API para obtenção de dados a partir de Outubro de 2017. Os dados são divididos em páginas, do mais antigo nas primeiras páginas aos mais recentes nas páginas de número mais alto, em formato JSON. A coleta de dados foi realizada percorrendo todas as páginas, armazenando as informações de cada página em uma base de dados. Para o funcionamento online do sistema, ao invés de interromper a execução da coleta ao encontrar a última página, pode-se permanecer nesta página, até que seja preenchida, capturando cada novo dado recebido para processamento.

No caso do Twitter, um *crawler* coletou todos os *tweets* que continham certas menções (*hashtags*), de determinadas contas dentro do período analisado na dissertação. Deste modo, foi construída uma base de dados com *tweets* de canais de imprensa, voltados para notificação de trânsito. Estes *tweets* foram utilizados em buscas sob demanda pelo módulo Processamento, a fim de avaliar as anomalias detectadas. Durante a execução do protótipo, os dados da prefeitura são avaliados sequencialmente de acordo o tempo, como na stream de dados do funcionamento em ambiente real. A utilização dos dados do Twitter segue a mesma linha do tempo, de forma que no ambiente de implantação a requisição por dados pode ser feita por demanda, de acordo com o horário.

5.2 Caracterização dos Dados de Mobilidade de Vitória

Cronologicamente, a partir da parceria entre universidade e prefeitura, o primeiro passo foi caracterizar os dados disponíveis. Assim, um estudo inicial foi realizado a partir de dados da prefeitura, relativos aos anos de 2017, 2018 e 2019, para compreensão dos atributos disponíveis.

A partir dos dados recebidos da prefeitura de Vitória, referentes ao período de 2017 a 2018, foi realizada uma análise inicial de reconhecimento dos dados, apresentada na Seção 5.2.2, observando os atributos presentes e as ocorrências de cada tipo, para cada via e em diferentes períodos. Nessa análise foi revelado, ainda, um mapa das regiões mais ativas da cidade por faixa de horário. Em seguida, utilizando um conjunto de dados de outro tipo, relativos aos anos 2018 e 2019, foi realizada uma segunda análise, reportada na Seção 5.2.3.

5.2.1 Descrição dos Dados

A API da prefeitura disponibiliza acesso a três tipos de dados: notificações de usuários (alertas), avaliação de alertas de usuários de acordo com outros usuários (irregularidades) e velocidade média quando há congestionamento (engarrafamentos), em cada uma das vias. Os atributos dos dados de alertas e engarrafamentos são listados nas Tabelas 2 e 3, respectivamente. As descrições dos atributos nas tabelas foram inferidas em reuniões com a equipe responsável pelos dados na prefeitura e com base em (LENKEI,

Atributo	Curta descrição
uuid	Identificação única da observação
city	Cidade de onde o alerta foi enviado
confidence	Confiança da observação baseada na reação de outros usuários
reliability	Confiança da observação baseada na reação de outros usuários e no rank do usuário
type	Categoria da notificação
roadType	Tipo da via
magvar	Direção do usuário durante a notificação
subType	Tipo da notificação
street	Via de onde o alerta foi enviado
location	Localização do usuário durante a notificação
eventDate	Data e hora do alerta
speed	Velocidade média atual no segmento engarrafado da via
reportDescription	Descrição do incidente pelo usuário

Tabela 2 – Atributos dos Dados da Tabela de Alertas

2018). Neste trabalho foi dado foco para utilização de dados de velocidade média da via em engarrafamento.

Atributo	Curta descrição
id	Identificação única da observação
city	Cidade da notificação
level	Nível de congestionamento
length	Tamanho do congestionamento
turnType	Indica se o congestionamento ocorreu em uma curva
type	Tipo do engarrafamento
speed	Velocidade média no segmento engarrafado da via
roadType	Tipo da via onde o congestionamento ocorreu
delay	Atraso causado pelo congestionamento
street	Via de onde o alerta foi enviado
eventDate	Data e hora do alerta

Tabela 3 – Atributos dos Dados da Tabela de Engarrafamentos

5.2.2 Caracterização dos Dados de Alertas

Inicialmente, os dados recebidos para análise foram dados de alertas de usuários, vindos de um *backup* do banco de dados da própria prefeitura, coletados pela equipe responsável pela gerência das informações. As notificações de alerta de usuários possuem tipo e subtipo, e a quantidade de cada um, entre Outubro de 2017 e Outubro de 2018 na cidade de Vitória, é ilustrada na Figura 8. A Figura 9 mostra o número de notificações, de cada tipo, por cidade no período de Outubro de 2017 à Outubro de 2018, em escala logarítmica. Nota-se claramente que a cidade de Vitória nesse conjunto de dados possui o maior número de observações.

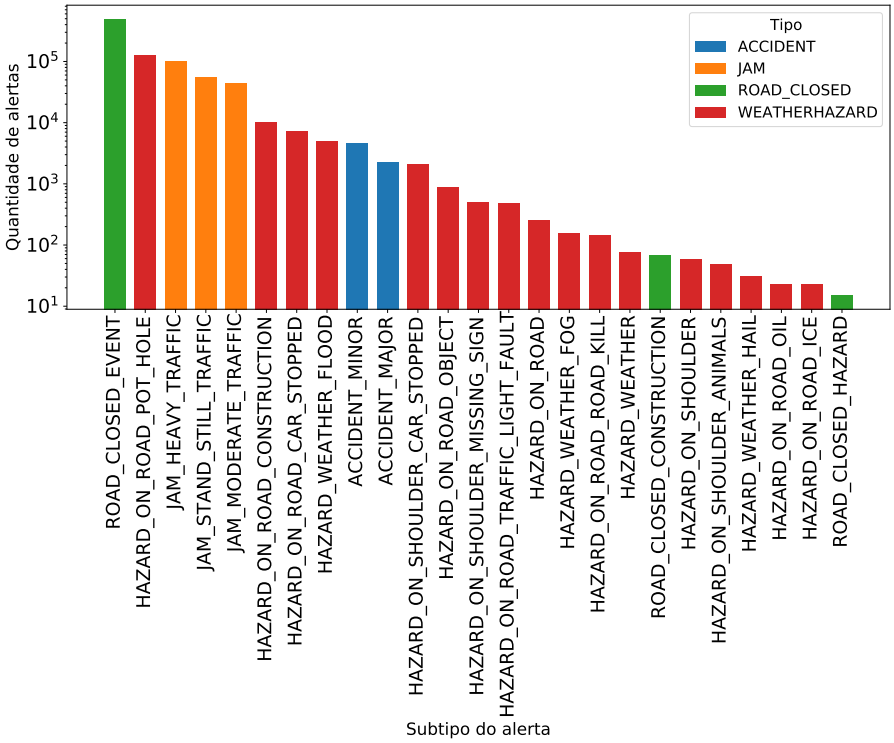


Figura 8 – Número de Alertas por Subtipo em Vitória de Outubro de 2017 a Outubro de 2018

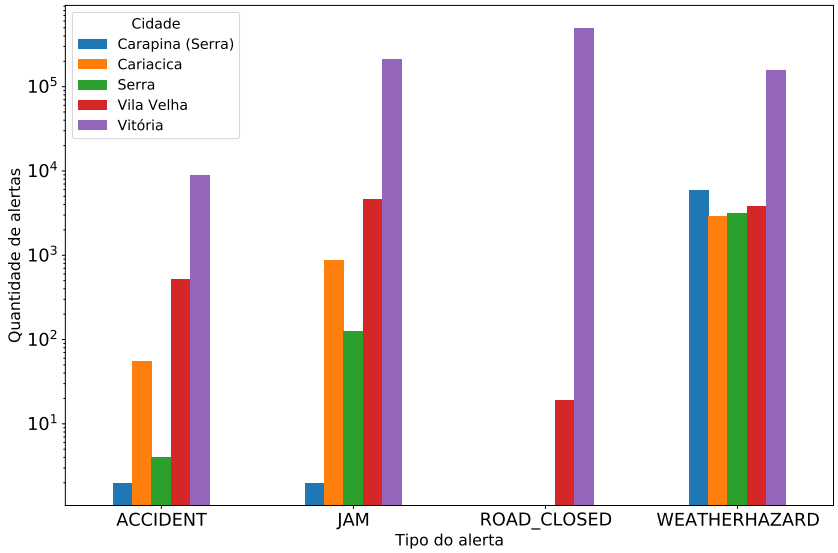
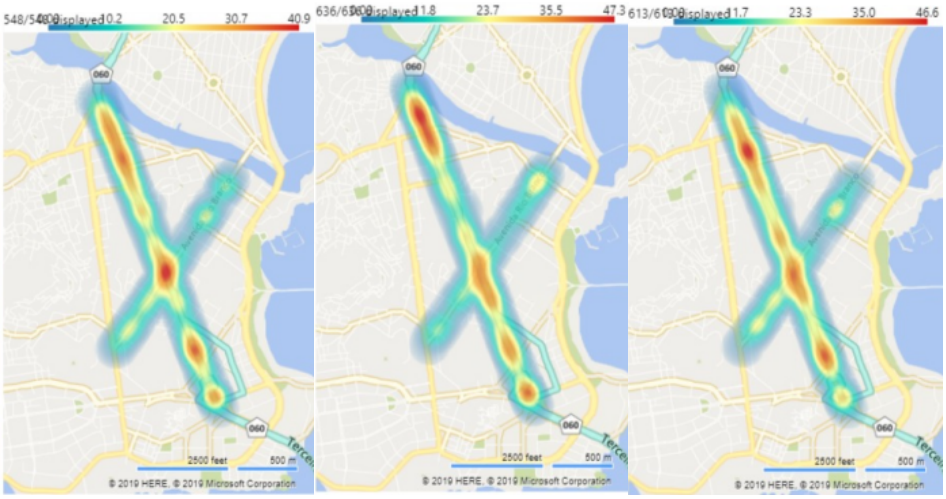


Figura 9 – Número de Notificações de Cada Tipo por Cidade de Outubro de 2017 a Outubro de 2018

A Figura 10 apresenta mapas de calor do número de alertas acumulados das avenidas Nossa Senhora da Penha, a mais extensa, e Rio Branco. Os resultados para os alertas enviados nas terças (Figura 10a), nas quintas (Figura 10b) e o acumulado dos dias úteis entre 17:00 e 18:00 (Figura 10c) indicam uma similaridade entre os dias úteis.



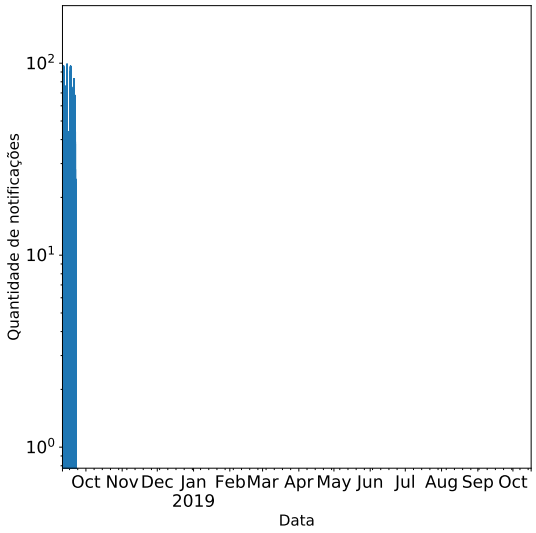
(a) Terças (b) Quintas (c) 17h-18h

Figura 10 – Alertas Acumulados por Dia de Semana e Horário em Duas Avenidas, de Outubro de 2017 a Outubro de 2018

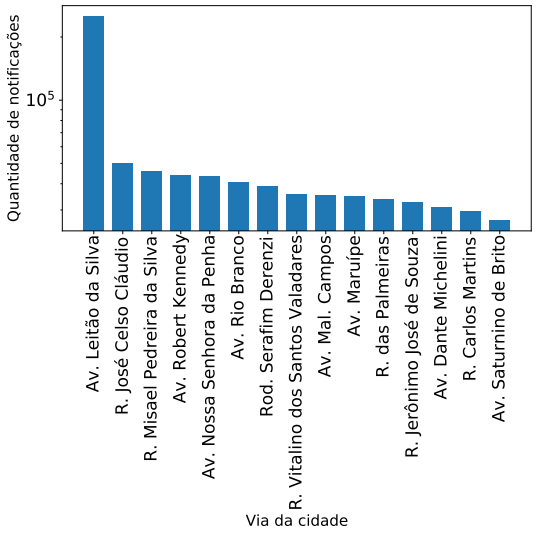
5.2.3 Caracterização dos Dados de Engarrafamentos

Após a primeira análise, realizada a partir dos dados de alertas de usuários, foram recebidos dados de engarrafamento coletados pela equipe da prefeitura, de Setembro de 2018 e Outubro de 2019. O gráfico da Figura 11a apresenta o número de eventos por dia, numa linha do tempo, dos dados de Vitória em um total de 2 milhões de observações. Nota-se a existência de lacunas de eventos em Novembro de 2018 e em Fevereiro de 2019, motivo que levou à utilização apenas dos dados de Março a Outubro de 2019 nas análises futuras e desenvolvimento, buscando continuidade e completude para a realização de testes. Em reunião, descobriu-se que as ausências de dados poderiam ocorrer diante das possíveis quedas de conexão e de energia, afetando o sistema de coleta da prefeitura.

A Figura 11b mostra um ranking das 15 vias da cidade com maior número de notificações, onde nota-se que poucas vias concentram a maior parte das ocorrências. Além disso, observa-se na Figura 12, a qual mostra as velocidades encontradas em 4 vias da cidade de Vitória durante congestionamento, um comportamento sazonal e local. Estes resultados, somados ao da Figura 10, sugerem não só a adoção de séries temporais por via, mas também que as séries possam ser categorizadas simplesmente em dias de semana e finais de semana. Neste ponto, a Avenida Nossa Senhora da Penha foi selecionada para as análises e testes posteriores, por ser uma das principais e mais ativas avenidas da cidade, além de uma das principais vias de acesso à universidade.



(a) Número de Eventos por Dia em Vitória



(b) Ranking das Vias de Vitória com Maior Quantidade de Observações

Figura 11 – Observações de Engarrafamento de Setembro de 2018 a Outubro de 2019

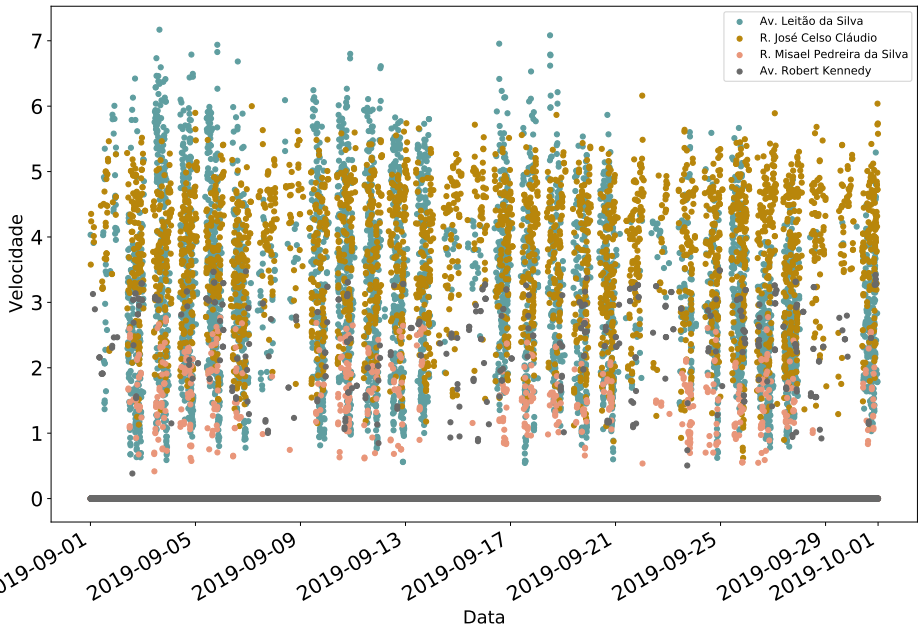


Figura 12 – Velocidades em Engarrafamentos em 4 Vias de Vitória em Um Mês

5.3 Clusterização de Séries Temporais

Após os resultados obtidos pelas Seções 5.2.2 e 5.2.3, observando o comportamento sazonal dos dados, optou-se por uma abordagem que obtenha um padrão, baseado em dados de dias anteriores ao atual, e que o utilize para classificar as observações recebidas em tempo real. Para isso, a fim de reduzir os dados de cada dia e evitar que dias com maior atividade pesassem demais no resultado, as observações diárias foram mescladas por faixa de horário, como ilustrado na Figura 13. Assim, cada dia é dividido em períodos,

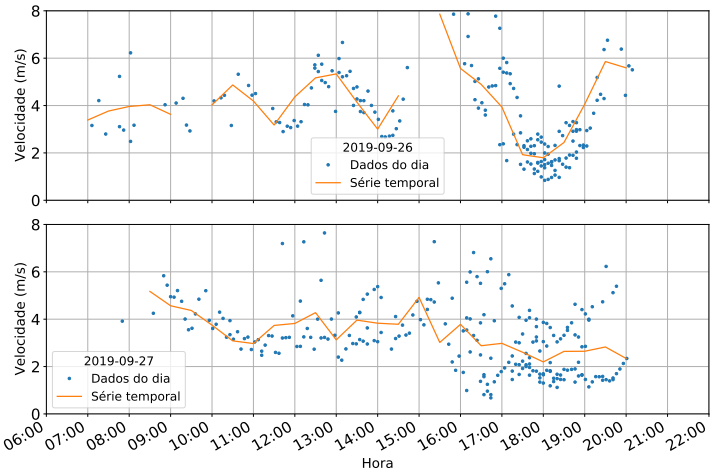


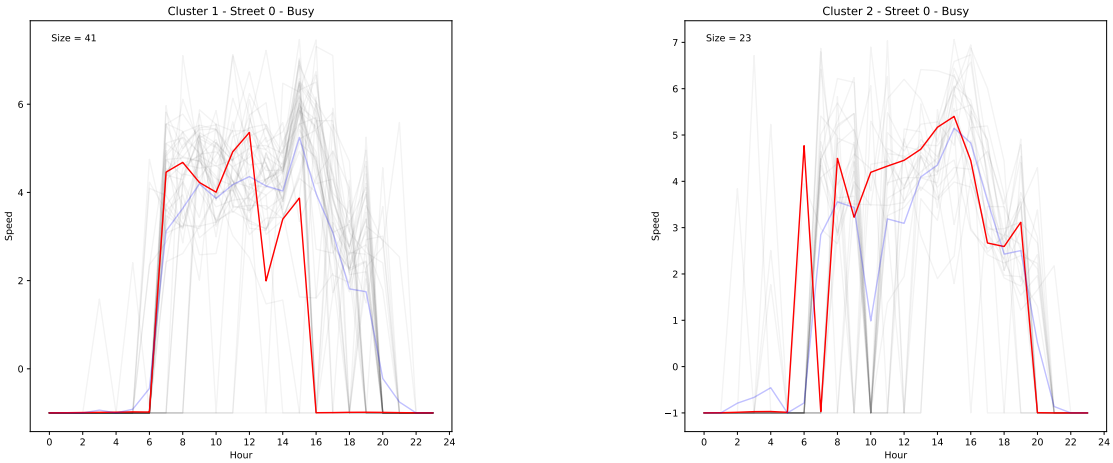
Figura 13 – Dados Reduzidos de 2 Dias da Av. Nossa Senhora da Penha

gerando séries temporais diárias, cujo valor de cada período é obtido aplicando uma função de agregação e/ou redução, como descrito no Algoritmo 1.

Dada a semelhança de comportamento das séries temporais diárias, a primeira abordagem utilizada para obter o padrão era o K-Means com DBA. A Figura 14 mostra os centros de dois clusters obtidos por este método, utilizando K-Means com 2 clusters. Já a Figura 15a mostra o maior cluster encontrado ao utilizar o mesmo método com 3 clusters. As linhas azuis nas imagens são as médias absolutas entre todas as séries dos respectivos clusters. Nota-se a diferença entre simplesmente calcular a média dos pontos das séries e utilizar o K-Means com DBA. A mesma técnica foi utilizada sobre os dados de alertas de congestionamentos na Avenida Rio Branco, cujo resultado é ilustrado na Figura 15b, para efeito de comparação.

Pelo K-Means com DBA era possível obter uma série temporal padrão, mas esta não necessariamente refletia o comportamento normal das vias em cada horário. Como esse método se utiliza da distância total entre séries para clusterizar as séries temporais, um dia, por exemplo, que apresentar um evento anormal em determinado horário pode acabar sendo incluído no maior cluster, influenciando a série padrão. Assim, um comportamento anormal poderia alterar o resultado final, nessa abordagem. Por esse motivo, esse método e a simples média das séries foram descartados.

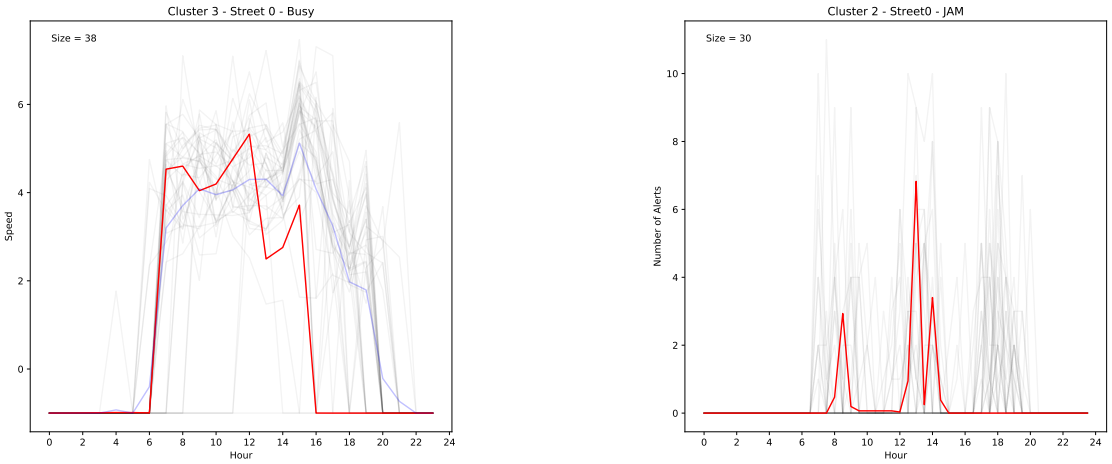
Finalmente, a fim de considerar apenas as observações mais comuns, a abordagem adotada se baseia em uma clusterização unidimensional, clusterizando observações de todos os dias, dentro da mesma faixa de horário. Assim, pontos de valores incomuns, como possíveis anomalias, ficam de fora do maior conjunto de dados, considerado o padrão.



(a) Cluster 1

(b) Cluster 2

Figura 14 – Dois Clusters Encontrados com DBA na Av. Nossa Senhora da Penha em Dias da Semana



(a) Padrão Obtido com DBA na Av. Nossa Senhora da Penha em Dias da Semana

(b) Padrão de Alertas Obtido com DBA na Av. Rio Branco com Todos os Dias

Figura 15 – Maiores Clusters Encontrados Usando DBA com 3 Clusters em Duas Vias

5.4 Considerações Sobre o Capítulo

Neste capítulo foram apresentados os dados utilizados no desenvolvimento do trabalho, demonstrando o comportamento sazonal nas vias da cidade. Também foi apresentada a distribuição de observações por localidade, de modo a analisar a quantidade de dados disponíveis por região.

Foram abordados principalmente três métodos para obter uma série temporal padrão de velocidades em engarrafamento, média, K-Means com DBA e clusterização

unidimensional. Essas técnicas permitem gerar uma série temporal que represente um histórico de dados, porém o uso da média e do DBA não impedem que observações possivelmente anômalas, com valores excepcionais, influenciem no resultado. Assim, optou-se por uma abordagem que obtivesse apenas as observações mais comuns de cada período do dia, a clusterização unidimensional por faixa de horário. As limitações dessa forma de obter um padrão, no entanto, derivam da esperança de que os dados possuam comportamento sazonal, de forma que essa solução não se encaixaria imediatamente em casos genéricos ou de alta variação, principalmente quando independente do tempo.

6 Avaliação do Arcabouço

O protótipo do arcabouço desenvolvido foi avaliado utilizando os dados de engarrafamento, disponibilizados na API da prefeitura de Vitória, e os dados coletados do Twitter. Essa análise foi realizada de modo *offline*, ou seja, todos os dados foram obtidos e armazenados para, em seguida, executar os algoritmos correspondentes aos módulos Pré-Processamento e Processamento, obtendo imediatamente todas as anomalias. Para a execução dos algoritmos, foi emulado um fluxo continuo de dados a partir dos arquivos estáticos a fim de reproduzir uma entrada de dados via *streaming* em tempo real.

Na Seção 6.1 são descritas as métricas para avaliação do protótipo. A Seção 6.2 apresenta os resultados obtidos a partir da variação de diversos parâmetros do arcabouço, de forma a trazer uma comparação de configurações do sistema. Por fim, a Seção 6.3 descreve os resultados da execução do protótipo utilizando o melhor conjunto de parâmetros encontrados a partir do estudo. Por simplicidade, todas as análise foram realizadas usando dados relativos à via Avenida Nossa Senhora da Penha, uma das vias com maior número de dados coletados e das mais importantes na cidade.

6.1 Métricas de Avaliação

A métrica de avaliação do protótipo foi elaborada de modo a gerar resultados concisos e menos subjetivos, para analisar o desempenho do protótipo a partir da variação dos parâmetros do arcabouço. Inicialmente, a avaliação era realizada levando-se em conta apenas o número total de anomalias detectadas no dia, como ilustrado na Figura 16. Assim, a configuração de parâmetros que resultasse no maior número de detecções, seria considerada ideal. Porém, além de desfavorecer a avaliação objetiva, esta abordagem desconsidera o número total de anomalias reais, ou seja, aqueles dados classificados como anomalia que realmente possam representar uma consequência de um evento anormal, como um acidente, por isso foi descartada.

A proposta para avaliar o protótipo utiliza uma métrica baseada em comparação entre anomalias detectadas e *tweets* relacionados: para cada anomalia, procura-se um *tweet*, no espaço próximo de tempo de 30 minutos, que mencione a rua relativa à anomalia. Caso um *tweet* seja encontrado, indica-se uma correspondência, classificando a anomalia como "certa". Assim, tomando k como um dia, a taxa de acerto R_k é calculada como T_k, a quantidade de anomalias "certas", dividida por A_k, o total de anomalias detectadas. Por fim, é obtida a média R desta taxa entre todos os dias. Isto é:

seja T_k o número de anomalias em k que estão relacionadas a um evento no Twitter

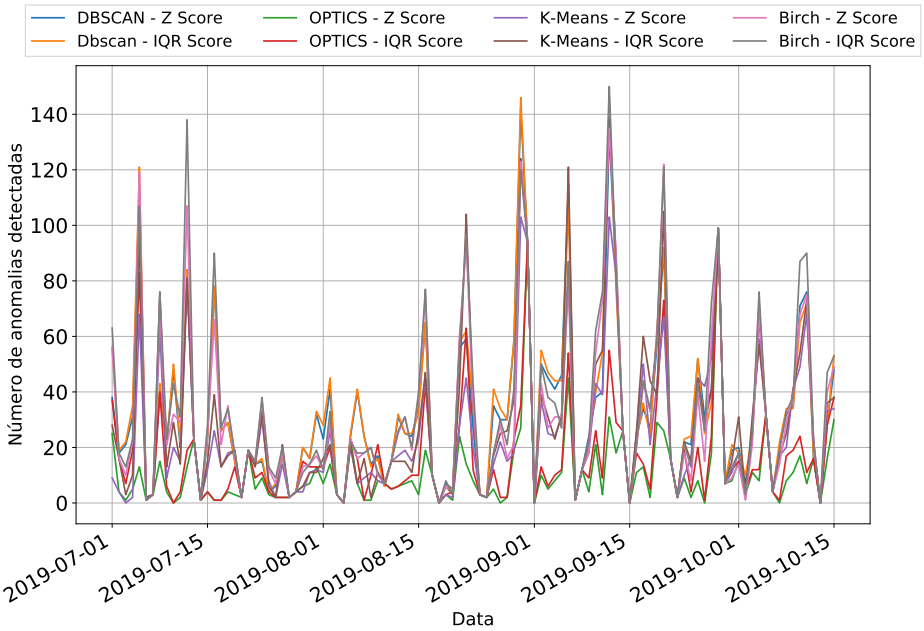


Figura 16 – Número de Anomalias por Dia de 1 de Julho a 15 de Outubro de 2019

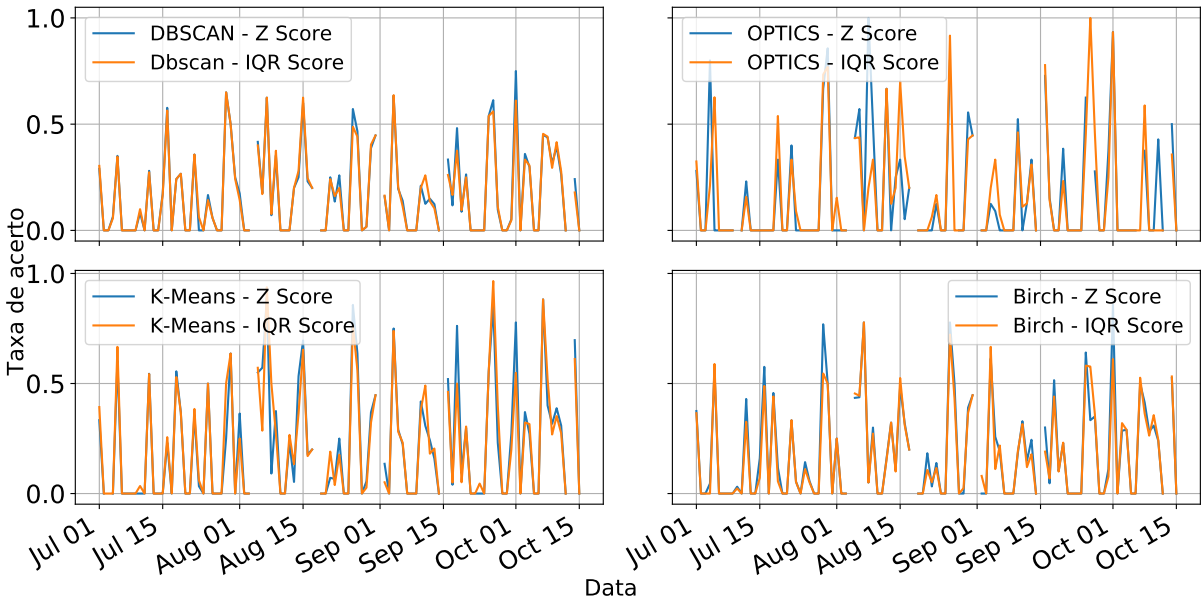


Figura 17 – Taxas de Acerto Diárias com Janela de 30 Minutos e Histórico de 30 Dias, de 1 de Julho a 15 de Outubro de 2019

$$\text{seja } A_k \text{ o total de anomalias detectadas em } k$$
$$\text{a taxa de acerto em } k \text{ é } R_k = \begin{cases} T_k/A_k, & \text{se } A_k > 0 \\ 0, & \text{se } A_k = 0 \end{cases}$$
$$\text{a taxa média é } R = \sum_k R_k/k$$

A Figura 17 ilustra R_k para k de 1 de Julho a 15 de Outubro de 2019, utilizando o protótipo com as quatro variações de algoritmos de clusterização e 2 técnicas de classificação de dados. A imagem evidencia a necessidade de utilizar um método que resuma os resultados,

dada a subjetividade na escolha da melhor configuração a partir do gráfico. Mais ainda, com o aumento do número de algoritmos de clusterização, técnicas de detecção de anomalia, tamanhos da janela e tamanho do histórico, haveria uma explosão na quantidade de curvas para análise, impossibilitando uma avaliação objetiva. Por isso, opta-se pela utilização do R como a métrica de avaliação do protótipo, de forma que o maior valor obtido em R indica a melhor opção.

6.2 Avaliação Paramétrica

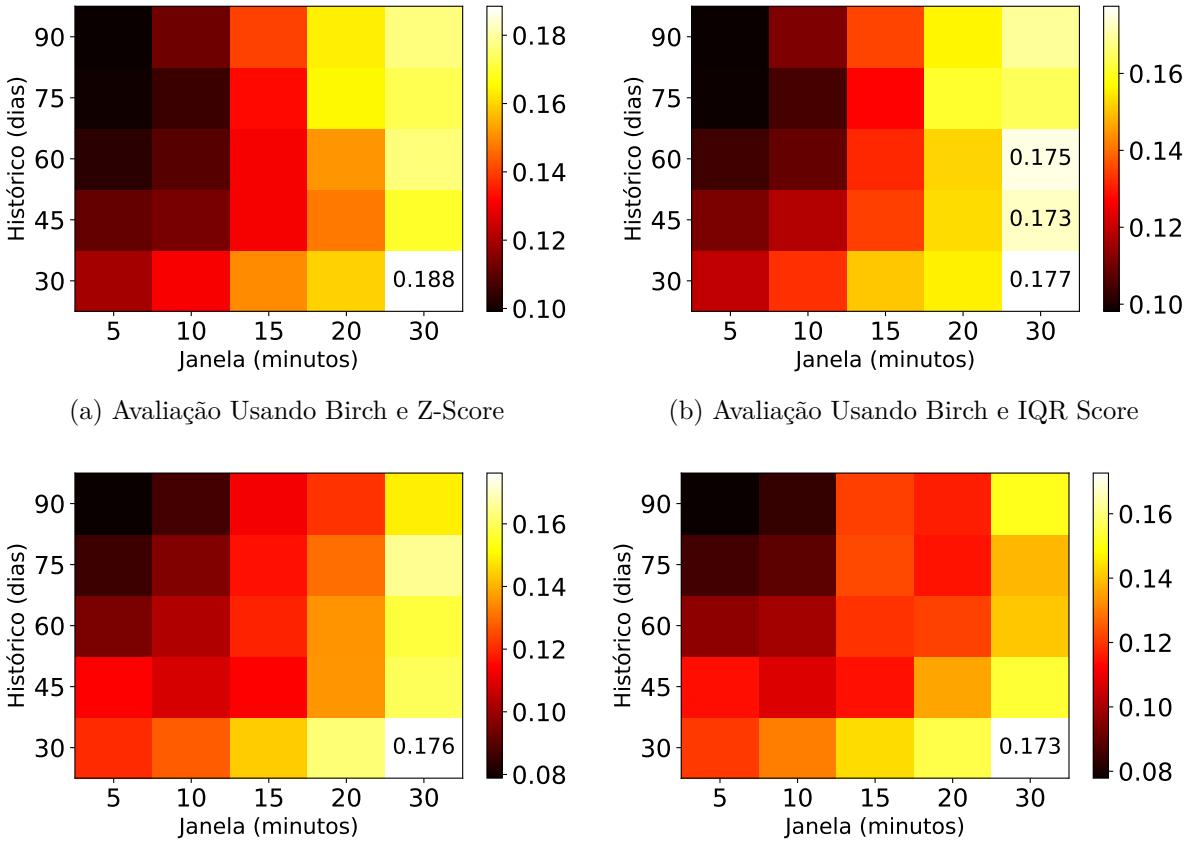
O arcabouço possui uma série de parâmetros dentre os quais destacam-se $Clustering$, D e $Jmin$, do Algoritmo 1, e o método para classificação de anomalia a partir de um grupo. No desenvolvimento da dissertação, foram utilizadas algumas variações de cada um desses parâmetros, como explicitado na Tabela 4, a fim de avaliar o desempenho do protótipo e encontrar a melhor configuração, utilizando os dados de engarrafamento mencionados na Seção 5.2.3. A partir desses valores, foram gerados mapas de calor para ilustrar os resultados de cada configuração, sendo que as células correspondentes aos 10% melhores resultados explicitam o valor R.

Parâmetro	Valores utilizados
Algoritmo de clusterização, $Clustering$	K-Means com 3 clusters, Birch, DBSCAN, OPTICS
Método de classificação de anomalia	Z-Score, IQR
Tamanho do histórico em dias, D	30, 45, 60, 75, 90
Tamanho da janela em minutos, $Jmin$	5, 10, 15, 20, 30

Tabela 4 – Parâmetros de Entrada para Avaliar o Protótipo do Arcabouço

6.3 Resultados

As Figuras 18 e 19 apresentam os resultados das avaliações do protótipo, R, por meio de mapas de calor. Primeiramente, nota-se que um histórico de 30 dias e uma janela de 30 minutos gerou, na maior parte dos casos, os melhores resultados. Essa informação sugere que eventos mais recentes devem possuir maior valor nas detecções de anomalias que os mais antigos, o que poderia ser justificado pelo fato do trânsito apresentar comportamento sazonal em diferentes períodos, semanalmente, por conta da movimentação característica de dias úteis, e mensalmente, por conta de períodos de férias escolares, feriados ou obras na infraestrutura. Além disso, de acordo com os resultados, os efeitos de eventos de mobilidade urbana aparentam possuir uma certa inércia, de forma que as consequências aparentam durar até uma hora. Isto pode refletir o fato de incidentes de trânsito possivelmente causarem um efeito dominó, reduzindo a velocidade dos veículos próximos, gerando engarrafamento, que somente retorna à normalidade algum tempo



(a) Avaliação Usando Birch e Z-Score

(b) Avaliação Usando Birch e IQR Score

(c) Avaliação Usando DBSCAN e Z-Score

(d) Avaliação Usando DBSCAN e IQR Score

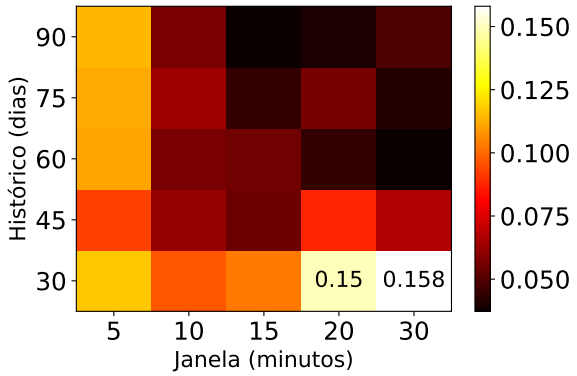
Figura 18 – Avaliação do Protótipo com Birch e DBSCAN

após a solução do incidente. Entre os algoritmos de clusterização e técnica de detecção de anomalias, a configuração K-Means, com 3 clusters, e Z-Score apresentou melhores valores da média das taxas diárias.

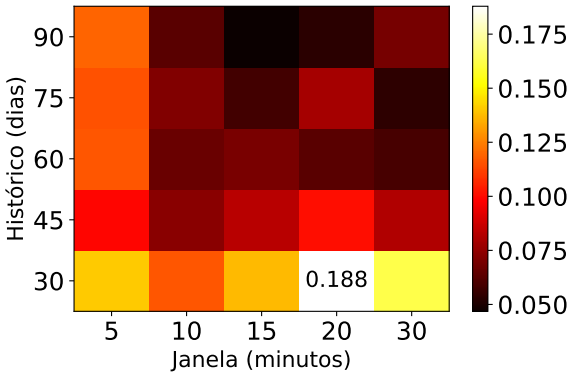
A partir da melhor configuração de parâmetros, os gráficos das Figuras 20, 21, 22 e 23 foram gerados, ilustrando os resultados da execução do protótipo a partir dos dados disponíveis de engarrafamento. A primeira figura apresenta os clusters obtidos para o dia 1 de Outubro de 2019, sendo a linha laranja a representante dos centros dos maiores clusters de cada período, a série temporal padrão. A segunda figura mostra as anomalias detectadas no mesmo dia. Notam-se dois grupos de anomalias, uma anomalia próxima das 13:00 e um aglomerado por volta de 18:00. Aproximadamente nestes dois horários, no Twitter, encontram-se postagens acerca de uma manutenção em uma ponte no final da Avenida Nossa Senhora da Penha e sobre um intenso congestionamento e retenções, respectivamente[1].

A Figura 22 apresenta a quantidade total de anomalias detectadas por dia, de 1 de Julho a 15 de Outubro de 2019, de forma a ilustrar a variação da quantidade de anomalias

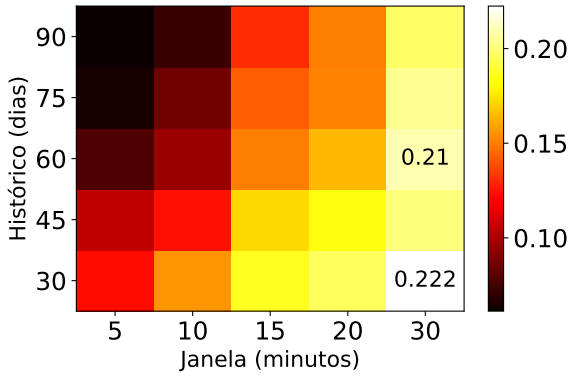
[1] "penha (reta OR avenida) (from:cbnvitoria OR from:prf191es OR from:_eco101) until:2019-10-02 since:2019-10-01"



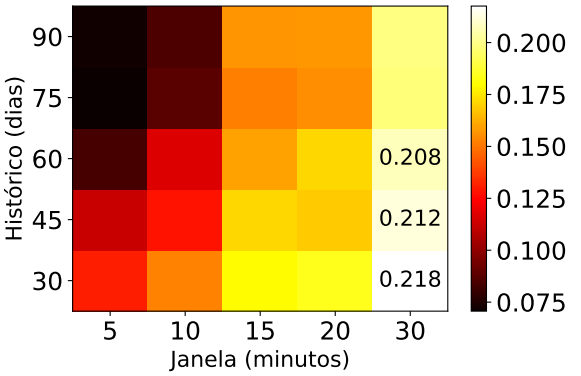
(a) Avaliação Usando OPTICS e Z-Score



(b) Avaliação Usando OPTICS e IQR Score



(c) Avaliação Usando K-Means com 3 Clusters e Z-Score



(d) Avaliação Usando K-Means com 3 Clusters e IQR Score

Figura 19 – Avaliação do Protótipo com OPTICS e K-Means com 3 Clusters

encontradas ao longo do tempo com o protótipo. Essa variação pode representar datas em que houve incidentes de trânsito. A segunda imagem da figura ilustra a taxa de acerto obtida, sendo que dias em que não há anomalias detectadas não são representados, gerando as descontinuidades do gráfico. Nota-se que o segundo gráfico não aparenta possuir relação de proporcionalidade em relação ao primeiro, de forma que dias com muitas anomalias detectadas não significam, na métrica utilizada nesta dissertação a partir do Twitter, altas taxas de acerto. Um exemplo é o dia 12 de Setembro de 2019, quando houve o maior número de anomalias detectadas, porém baixa taxa de acertos. Tal cenário pode ser consequência de situações como a baixa adesão da população e/ou de veículos de mídia na publicação de incidentes de trânsito no Twitter[2] e perturbações no trânsito que podem agravar ou gerar engarrafamentos (como engarrafamentos fantasmas).

A Figura 23 ilustra a quantidade de anomalias por período, somando as anomalias dos dias entre 1 de Julho de 2019 e 15 de Outubro de 2019. Nota-se que a maior parte das detecções ocorre no período vespertino dos dias, principalmente próximo de 17:00 e 13:00. Tais períodos são conhecidos por serem horários de final de expediente e de aulas

[2] "penha (reta OR avenida) (from:cbnvitoria OR from:prf191es OR from:_eco101) until:2019-09-13 since:2019-09-12"

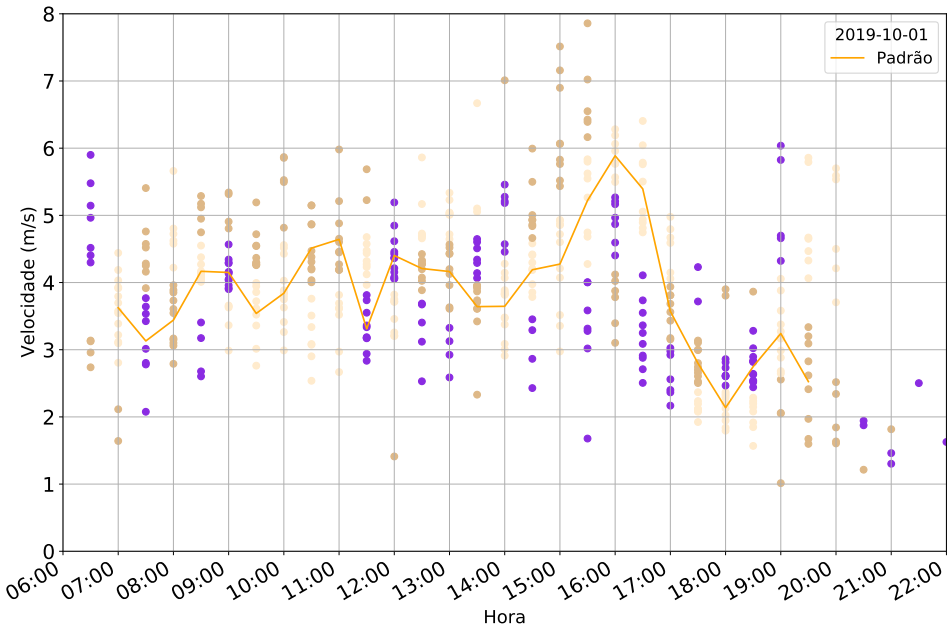


Figura 20 – Clusters Obtidos Usando K-Means com 3 Clusters, Histórico de 30 Dias e Janela de 30 Minutos, no Dia 1 de Outubro de 2019

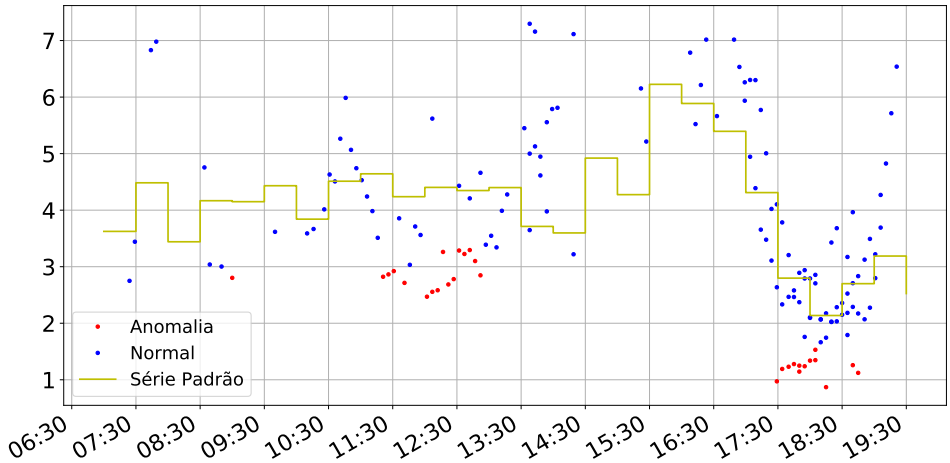


Figura 21 – Anomalias Obtidas Usando K-Means com 3 Clusters, Histórico de 30 Dias, Janela de 30 Minutos e Z-Score, no Dia 1 de Outubro de 2019

e de almoço, o que gera um aumento de densidade de veículos nas vias e possivelmente engarrafamentos. O fato do número de anomalias se concentrar no final do dia levanta suspeitas acerca das condições dos motoristas e das circunstâncias das vias. Sendo horários de almoço e de final de expediente, é natural que as pessoas, motoristas inclusos, estejam com alto nível de estresse, de forma a favorecer incidentes. Além disso, como as interferências para realização de obras nas vias são feitas durante o dia, pode-se afirmar que as chances de uma obra gerar engarrafamento nos horários de início de expediente são menores, de forma que os efeitos das interferências somente se dariam com o passar do dia, talvez de forma cumulativa.

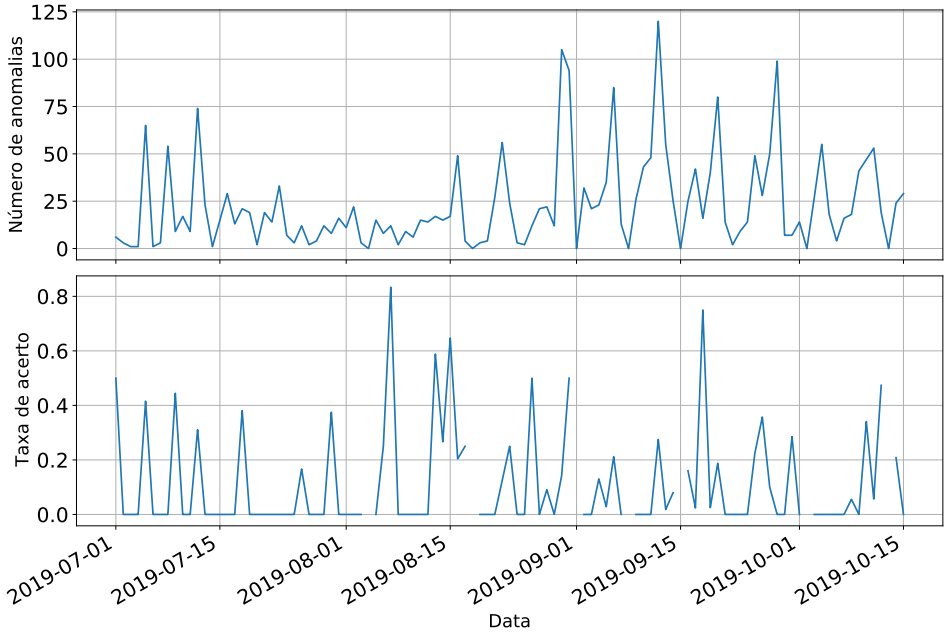


Figura 22 – Quantidade de Anomalias e Taxa de Acerto Obtidas Usando K-Means com 3 Clusters, Histórico de 30 Dias, Janela de 30 Minutos e Z-Score, de 1 de Julho a 15 de Outubro de 2019

6.4 Considerações Sobre o Capítulo

Neste capítulo foram abordadas algumas técnicas de clusterização e de detecção de outlier como parâmetros do sistema proposto, assim como uma métrica para avaliar o comportamento do protótipo desenvolvido. Os resultados sugerem o uso de K-Means com 3 clusters para encontrar a série padrão, representando as faixas de velocidades mais comuns em cada período do dia, e Z-Score para detectar as anomalias a partir dos maiores clusters obtidos. Essa configuração pode ser utilizada, então, para o desenvolvimento de um sistema real de detecção de anomalia. A partir dos resultados apresentados, nota-se os baixos valores para as taxas de acerto, possivelmente consequência do baixo engajamento social na rede utilizada na métrica de avaliação dentro da cidade. Apesar disso, percebe-se que as anomalias detectadas de fato podem levantar suspeitas de que algum incidente está ocorrendo, dado seus valores distantes da faixa daqueles normais, mais comuns. Por fim, pode-se afirmar que o arcabouço pode ser utilizado como forma de detectar anomalias de mobilidade urbana com o intuito de alertar as autoridades do município sobre possíveis eventos de trânsito.

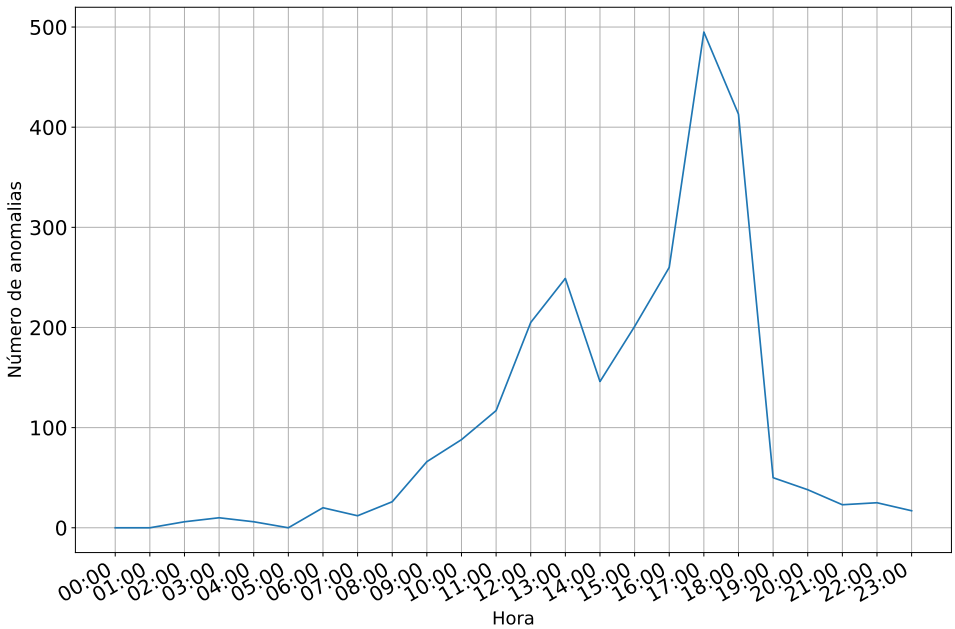


Figura 23 – Quantidade Total de Anomalias Detectadas por Período do Dia, Usando K-Means com 3 Clusters, Histórico de 30 Dias, Janela de 30 Minutos e Z-Score, de 1 de Julho a 15 de Outubro de 2019

7 Conclusões

Nesta dissertação foi apresentado o caso da, CIOM, a central de monitoramento da Guarda Municipal de Vitória, onde autoridades supervisionam diversos locais da cidade por meio de câmeras de videomonitoramento. Por se tratar de uma atividade que requer muita atenção, os agentes correm risco de fadiga, o que pode comprometer a eficiência da atividade. Além disso, a CIOM recebe informações de aplicativos colaborativos, como o Waze, coletadas ao longo da cidade, de forma que abre espaço para desenvolvimento de soluções que auxiliem o trabalho dos agentes. Nesta situação, um sistema para alertar os agentes, em tempo real, acerca de possíveis incidentes nas diversas localidades permitiria tomadas de decisão mais rápidas, acelerando o processo de resolução de problemas. Assim, foi proposto um arcabouço de detecção e alerta de anomalias em tempo real para notificar as autoridades de uma cidade inteligente, a fim de auxiliar no monitoramento das vias da cidade.

O arcabouço recebe informações de múltiplas fontes de dados para as utilizar no aumento da confiabilidade das detecções. Para isso, são geradas séries temporais padrão baseadas nos históricos de valores diários, de forma a considerar o comportamento sazonal urbano. Ainda, elaborado numa arquitetura distribuída e parametrizada, o arcabouço se torna escalável no sentido que a quantidade de módulos implantadas pode ser proporcional ao número de locais e de fontes de dados. Ele separa os dados de acordo com a via e reduz a quantidade de dados de cada dia, dividindo o dia em períodos e reduzindo os dados dentro de cada intervalo, gerando séries temporais diárias. Uma técnica de clusterização é aplicada unidimensionalmente sobre os valores relativos à mesma faixa de horário, de modo a obter o maior grupo de valores próximos e, assim, a faixa de valores mais comum, obtendo uma série temporal padrão para cada via. Ao receber um dado, o sistema utiliza um método de detecção de anomalias para classificar rapidamente a informação, tomando o padrão obtido como base. Finalmente, a arquitetura modular utilizada habilita um melhor dimensionamento e alocação de recursos, então, pode-se dizer que os objetivos em 1.2 são atendidos.

Um protótipo do arcabouço foi desenvolvido e avaliado utilizando dados reais provenientes da prefeitura de Vitória-ES, e os resultados mostram que o arcabouço é capaz de detectar anomalias utilizando dados de uma segunda fonte, como o Twitter, para as validar. Alguns algoritmos de clusterização e métodos de detecção de anomalias foram utilizados para analisar a taxa de anomalias detectadas que podem ser corroboradas com algum *tweet*, de forma que a melhor combinação possa ser configurada na implantação do sistema definitivo. Apesar das baixas taxas de "acerto" na métrica apresentada, o que pode ser justificado pela baixa adesão à plataforma Twitter na cidade e pelo reduzido número

de fontes utilizadas, é possível concluir, enfim, que o arcabouço pode auxiliar os agentes no monitoramento da cidade, reduzindo a fadiga no exercício da tarefa e direcionando a atenção aos locais onde é necessário.

7.1 Perspectivas de Trabalhos Futuros

Esta dissertação foi desenvolvida de maneira a abrir caminho a uma gama de outros trabalhos. Nesta seção são listados, de forma não exaustiva, possíveis estudos derivados desta dissertação. Vale destacar o desenvolvimento do protótipo como um sistema, visando a integração do mesmo com os sistemas da prefeitura para auxiliar os agentes municipais no supervisionamento das vias públicas. A partir da implementação do sistema, pode-se planejar futuras integrações, de maneira a utilizar outros canais de obtenção de dados. Além disso, o desenvolvimento de plataformas de visualização, como *dashboards*, podem permitir uma melhor assimilação dos resultados gerados pelo sistema, pré e pós implantação.

A redução de dados em cada dia, por faixas de horário, pode esconder comportamentos individuais interessantes, de forma que estudos acerca do impacto desta abordagem e de técnicas para atingir este objetivo, como clusterização bidimensional, podem determinar melhoras na abordagem da solução proposta. Uma outra possível maneira de garantir a diversificação dos dados, ainda que reduzidos, é com uso de clusterização em que a distância entre os indivíduos leve em consideração um peso. Assim, permite-se evitar que dados de dias mais ativos se sobressaiam aos demais, ao mesmo tempo que eventos muito similares, como múltiplos eventos de valores próximos e num curto espaço de tempo, podem ser mesclados em um único dado de peso maior.

A avaliação do protótipo realizada nesta dissertação carece de maior variação de métodos e de parâmetros, como uso de K-Means com 2, 4 ou mais clusters e de outras técnicas para classificação dos dados a partir do padrão. Este estudo pode revelar combinações mais eficientes para a tarefa de detecção de anomalias. Uma das possíveis formas de se determinar esta configuração é por meio de métodos de otimização não linear como algoritmos inspirados na natureza, PSO, GA, ACO e GSA, por exemplo. Vale ressaltar a possibilidade de um estudo a partir de dados públicos e conhecidos, para avaliar a eficácia deste método de detecção de anomalias, baseado em clusterização seguida de classificação. Adiante, a classificação binária, como o uso de um limiar, pode acarretar numa atribuição errônea, de forma que uma abordagem que utilize probabilidades ou graus de pertinência, como a lógica fuzzy, podem permitir uma classificação mais honesta. Além disso, o uso de lógica fuzzy também pode auxiliar ao considerar o desvio de tempo na classificação de uma anomalia, quando deve na verdade existir uma probabilidade de o comportamento sazonal ter ocorrido mais cedo ou mais tarde, o que porventura carece de um estudo.

Em relação às fontes suportadas pelo sistema, um possível estudo seria acerca da semântica dos dados, visando a integração de observações de fontes heterogêneas, que podem se referir à mesma informação, como laços de indução e câmeras de monitoramento de tráfego, aumentando a confiabilidade dos dados. Pode-se desenvolver, também, cálculos de níveis de severidade para as anomalias detectadas, utilizando dados de diversas outras fontes, número de anomalias detectadas em sequência, etc. Por fim, outras utilidades para o sistema poderiam ser discutidas, como o uso de metadados de câmeras de vigilância para detectar encontro de multidões, havendo implicações na segurança pública, e dados de áudio para detectar barulhos característicos ou incomuns, como sons de disparo de arma de fogo ou explosões.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior Brasil (CAPES) - Código de Financiamento 001, do CNPq e da Fundação de Amparo à Pesquisa do Espírito Santo (FAPES). Adicionalmente, este trabalho foi viabilizado por meio do termo de cooperação técnica 004/2018, entre a Secretaria Municipal de Segurança Pública de Vitória, Espírito Santo, e a UFES. O autor agradece o esforço da secretaria pela disponibilização dos dados em tempo real.

Referências

AHMAD, S. et al. Unsupervised real-time anomaly detection for streaming data. *Neurocomputing*, Elsevier B.V., v. 262, p. 134–147, 2017. ISSN 18728286. Citado na página 35.

AHMED, M.; MAHMOOD, A. N.; HU, J. A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, v. 60, p. 19 – 31, 2016. ISSN 1084-8045. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1084804515002891>. Citado na página 33.

ALAM, M. R. et al. Adaptable anomaly detection in traffic flow time series. *MT-ITS 2019 - 6th International Conference on Models and Technologies for Intelligent Transportation Systems*, IEEE, 2019. Citado na página 35.

ALBINO, V. U. B. R. M. D. Smart cities: definitions, dimensions, and performance. *Journal of Urban Technology*, p. 1723–1738, 2015. Citado na página 23.

ANKERST, M. et al. Optics: Ordering points to identify the clustering structure. In: *SIGMOD '99 Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*. [S.l.]: ACM Press, 1999. p. 49–60. Citado na página 27.

BAEK, S. et al. Clustering-based label estimation for network anomaly detection. *Digital Communications and Networks*, Elsevier Ltd, n. May, jun 2020. ISSN 23528648. Disponível em: <https://doi.org/10.1016/j.dcan.2020.06.001https://linkinghub.elsevier.com/retrieve/pii/S2352864818301779>. Citado na página 35.

BALOIAN, N. et al. Efficient planning of urban public transportation networks. In: *Ubiquitous Computing and Ambient Intelligence. Sensing, Processing, and Using Environmental Information*. Cham: Springer International Publishing, 2015. p. 439–448. ISBN 978-3-319-26401-1. Citado na página 33.

Bawaneh, M.; Simon, V. Anomaly detection in smart city traffic based on time series analysis. In: *2019 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. [S.l.: s.n.], 2019. p. 1–6. ISSN 1847-358X. Citado na página 33.

BOKSEM, M. A.; MEIJMAN, T. F.; LORIST, M. M. Effects of mental fatigue on attention: An erp study. *Cognitive Brain Research*, v. 25, n. 1, p. 107 – 116, 2005. ISSN 0926-6410. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0926641005001187>. Citado na página 49.

CALIKUS, E. et al. No free lunch but a cheaper supper: A general framework for streaming anomaly detection. *Expert Systems with Applications*, Elsevier Ltd, v. 155, p. 113453, 2020. ISSN 09574174. Disponível em: <https://doi.org/10.1016/j.eswa.2020.113453>. Citado na página 36.

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. AAAI Press, 1996. (KDD'96), p. 226–231. Disponível em: <http://dl.acm.org/citation.cfm?id=3001460.3001507>. Citado na página 27.

FAIAL, D. et al. Anomaly detection in vehicle traffic data using batch and stream supervised learning. In: *Progress in Artificial Intelligence*. Cham: Springer International Publishing, 2019. p. 675–684. ISBN 978-3-030-30241-2. Citado na página 33.

Gupta, L. et al. Nonlinear alignment and averaging for estimating the evoked potential. *IEEE Transactions on Biomedical Engineering*, v. 43, n. 4, p. 348–356, 1996. Citado na página 29.

HAWKINS, D. M. *Identification of outliers*. [S.l.]: Springer, 1980. v. 11. Citado na página 24.

HODGE, V. J.; AUSTIN, J. A Survey of Outlier Detection Methodologies. *Artificial Intelligence Review*, v. 22, n. 2, p. 85–126, 2004. ISSN 1573-7462. Disponível em: <https://doi.org/10.1007/s10462-004-4304-y>. Citado 2 vezes nas páginas 29 e 33.

IZAKIAN, H.; PEDRYCZ, W. Anomaly Detection and Characterization in Spatial Time Series Data: A Cluster-Centric Approach. *IEEE Transactions on Fuzzy Systems*, IEEE, v. 22, n. 6, p. 1612–1624, 2014. ISSN 10636706. Citado na página 34.

LADEIRA, L. et al. Serviço de sugestão de rotas seguras para veículos. In: *XXXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*. Porto Alegre, RS, Brasil: SBC, 2019. p. 608–621. ISSN 2177-9384. Disponível em: <https://sol.sbc.org.br/index.php/sbrc/article/view/7390>. Citado na página 33.

LENKEI, Z. *Crowdsourced traffic information in traffic management - Evaluation of traffic information from Waze*. Dissertação (Mestrado) — KTH Royal Institute of Technology, Stockholm, Sweden, 2018. Citado na página 51.

MACQUEEN, J. B. Some methods for classification and analysis of multivariate observations. In: *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*. [S.l.]: University of California Press, 1967. p. 281–297. Citado na página 28.

MONTORI, F.; BEDOGNI, L.; BONONI, L. A collaborative internet of things architecture for smart cities and environmental monitoring. *IEEE Internet of Things Journal*, IEEE, v. 5, n. 2, p. 592–605, 2017. Citado na página 23.

MONTORI, F. et al. Sensquare: A mobile crowdsensing architecture for smart cities. In: *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*. Reston, VA, USA: IEEE, 2016. p. 536–541. Citado na página 23.

PAN, B. et al. Crowd sensing of traffic anomalies based on human mobility and social media. *GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*, p. 334–343, 2013. Citado na página 33.

PANAGIOTOU, N. et al. Intelligent urban data monitoring for smart cities. In: *Machine Learning and Knowledge Discovery in Databases*. Cham: Springer International Publishing, 2016. p. 177–192. ISBN 978-3-319-46131-1. Citado na página 23.

PETITJEAN, F.; KETTERLIN, A.; GANÇARSKI, P. A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognition*, Elsevier, v. 44, n. 3, p. 678–693, 2011. Citado na página 29.

PURNOMO, F. et al. Smart city's context awareness using social media. *2016 International Conference on ICT for Smart Society, ICISS 2016*, p. 119–123, 2016. Citado na página 23.

RATHORE, M. M. et al. Urban planning and building smart cities based on the internet of things using big data analytics. *Computer Networks*, Elsevier, v. 101, p. 63–80, 2016. Citado na página 24.

Sidauruk, A.; Ikmah. Congestion correlation and classification from twitter and waze map using artificial neural network. In: *International Conference on Information Technology, Information System and Electrical Engineering*. [S.l.: s.n.], 2018. p. 224–229. ISSN null. Citado na página 33.

SILVA, T. H. et al. Users in the urban sensing process: Challenges and research opportunities. *Pervasive Computing: Next Generation Platforms for Intelligent Data Collection*, Academic Press, p. 45–95, 2016. Citado na página 23.

SOUZA, A. M. de et al. Por aqui é mais seguro: Melhorando a mobilidade e a segurança nas vias urbanas. In: *XXXVI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*. Porto Alegre, RS, Brasil: SBC, 2018. ISSN 2177-9384. Disponível em: <https://sol.sbc.org.br/index.php/sbrc/article/view/2475>. Citado na página 33.

THUY, H. T. T.; ANH, D. T.; CHAU, V. T. N. A Novel Method for Time Series Anomaly Detection based on Segmentation and Clustering. *Proceedings of 2018 10th International Conference on Knowledge and Systems Engineering, KSE 2018*, IEEE, p. 276–281, 2018. Citado na página 35.

WOOLDRIDGE, J. M. *Introductory Econometrics: A Modern Approach*. [S.l.]: South-Western, Cengage Learning, 2012. v. 5. Citado 2 vezes nas páginas 24 e 31.

WU, J.; ZENG, W.; YAN, F. Hierarchical Temporal Memory method for time-series-based anomaly detection. *Neurocomputing*, Elsevier B.V., v. 273, p. 535–546, 2018. ISSN 18728286. Disponível em: <https://doi.org/10.1016/j.neucom.2017.08.026>. Citado na página 35.

XU, D.; TIAN, Y. A Comprehensive Survey of Clustering Algorithms. *Annals of Data Science*, v. 2, n. 2, p. 165–193, 2015. ISSN 2198-5812. Disponível em: <https://doi.org/10.1007/s40745-015-0040-1>. Citado na página 27.

ZHANG, T.; RAMAKRISHNAN, R.; LIVNY, M. Birch: An efficient data clustering method for very large databases. *SIGMOD Rec.*, ACM, New York, NY, USA, v. 25, n. 2, p. 103–114, jun. 1996. ISSN 0163-5808. Disponível em: <http://doi.acm.org/10.1145/235968.233324>. Citado na página 28.