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

Marco A. B. Thomé[1], **André N. Prestes**[1], **Roberta L. Gomes** [1], **Vinícius F. S. Mota**[1]

[1]Departamento de Informática – Universidade Federal do Espírito Santo (UFES)
LPRM – 29075-910 – Espírito Santo – ES – Brazil

{mabthome,vinicius.mota,rgomes}@inf.ufes.br, andren.p@hotmail.com

Abstract. *Data gathered by sensors, cameras, social networks, and applications can contribute to atypical traffic events automatic detection. The heterogeneous nature of data sources has the advantage of information redundancy, increasing the degree of reliability for the detected events. This paper proposes a framework for detection and notification of anomalous events in real-time, with an interface that allows heterogeneous data sources. For this, after receiving events concerning traffic conditions, the framework groups them as time series. The time series are clustered to create a pattern, allowing real-time anomaly detection and alert. To validate the framework, Waze based and Twitter data interfaces were deployed and we compared several clustering algorithms and outlier detection techniques. By using real data from the city of Vitória-ES, the results showed that the proposed framework is scalable, and alerts can help authorities in their decision making.*

Resumo. *Os dados coletados por sensores, câmeras, redes sociais e aplicativos podem contribuir para detecção automática de eventos atípicos no trânsito. A natureza heterogênea das fontes de dados traz como vantagem a redundância de informação, aumentando o grau de confiabilidade de eventos detectados. Este trabalho propõe um arcabouço de detecção de eventos anômalos e alertas em tempo real, com uma interface que suporta fontes de dados heterogêneos. Para isto, ao receber os eventos de uma via, o arcabouço os agrupa como séries temporais diárias. É aplicada clusterização nestas séries temporais para criar um histórico padrão, que permite detectar anomalias e emitir alertas em tempo real. Para validar o arcabouço, foram implementadas interfaces para dados disponibilizados pela prefeitura de Vitória-ES, provenientes da plataforma Waze, e Twitter, e um conjunto de algoritmos de clusterização e de detecção de anomalias. Utilizando dados reais da cidade, os resultados mostraram que o arcabouço proposto é escalável e os alertas podem auxiliar os gestores nas tomadas de decisões.*

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 das grandes cidades do mundo [Albino 2015]. De fato, elas têm se mostrado como ferramentas úteis para a promoção de uma vivência mais segura, confortável e sustentável nas cidades [Panagiotou et al. 2016, Montori et al. 2016, Purnomo et al. 2016]. Somado aos dados

obtidos pela infraestrutura de monitoramento das cidades, há também 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].

A adoção de plataformas colaborativas por parte dos cidadãos vem crescendo de tal forma que agentes públicos têm buscado mais o estabelecimento de parcerias com empresas. Com esta tecnologia, é possível enriquecer a massa de dados para monitoramento de eventos em uma cidade sem a necessidade da construção de uma infraestrutura pública legada. Um exemplo é a cidade de Vitória-ES, que ocupa o primeiro lugar do *Ranking Connected Smart Cities*[1] (Cidades Inteligentes e Conectadas) de 2018, entre os municípios brasileiros com até 500 mil habitantes. A prefeitura possui uma Central Integrada de Operações e Monitoramento (CIOM) conectada a várias câmeras na cidade[2], onde agentes públicos vigiam os locais para que ações possam ser tomadas. 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. No entanto, interpretar essa massa de informações se torna uma tarefa árdua, tornando-se necessária a implantação de soluções para análise integrada e eficaz dos dados, melhorando o monitoramento de eventos e incidentes nas cidades [Montori et al. 2017].

Este trabalho propõe um arcabouço para coletar e analisar dados de fontes heterogêneas, e emitir alertas de anomalias detectadas. A solução desenvolvida utilizou dados disponibilizados pela prefeitura de Vitória, como a velocidade média em cada via nos períodos de congestionamento. Assim, entendendo-se como anomalia qualquer observação que desvia das demais [Hawkins 1980], as autoridades podem receber avisos sempre que alguma via apresentar congestionamento na qual a velocidade média detectada for aquém da faixa de valores esperados (padrão), levando em consideração comportamentos sazonais, como horários de pico. Desta forma, os responsáveis podem focar em pontos da cidade onde incidentes podem de fato estar ocorrendo. Além disso, a solução utiliza dados do *Twitter* para validar cada anomalia detectada por meio de uma segunda fonte.

O arcabouço processa os dados coletados para construir um histórico de observações da cidade e, a partir deste, detectar anomalias de trânsito em tempo real. Para isto, define-se uma interface de modelo de dados composto por um *timestamp* com, no mínimo, um atributo descritivo do evento. Este atributo pode ser, por exemplo, a velocidade média da via, descrição textual de um evento ou a localização de um acidente. Dos dados, obtém-se uma janela deslizante de '*D*' dias de histórico, transformada em um conjunto de séries temporais diárias. Cada série é composta por eventos ocorridos dentro de intervalos ao longo do dia. As séries são então agrupadas por similaridade, resultando em uma série padrão. Por fim, as observações em tempo real são comparadas dentro do intervalo com a série padrão e, caso estas sejam classificadas como anomalia (*outlier*) para o horário, são emitidos alertas para as autoridades responsáveis. Partindo da premissa que

[1] www.connectedsmartcities.com.br/resultados-downloads-connected-smart-cities/

[2] www.folhavitoria.com.br/geral/noticia/08/2019/vitoria-tera-centro-de-inteligencia-para-monitoramento-de-servicos

se mais de uma fonte afirma algo, a informação tende a ser mais confiável, o arcabouço utiliza eventos de múltiplas fontes para validar as detecções. Desta forma, consegue-se aumentar a confiabilidade das informações, permitindo um melhor planejamento urbano por parte das autoridades [Rathore et al. 2016].

Para o desenvolvimento do arcabouço, foram comparados quatro algoritmos de clusterização e duas técnicas estatísticas de detecção de outliers. Como resultado, a escolha entre esses algoritmos define a sensibilidade do arcabouço para emitir alertas de anomalias. Por fim, as principais contribuições deste artigo são:

- Propõe um arcabouço para gerenciamento 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; e
- Implementa um sistema de detecção e alerta de anomalias em tempo real, avaliado com dados reais, obtidos em parceria com a prefeitura de Vitória.

O restante deste artigo está organizado como segue: a Seção 2 apresenta os trabalhos relacionados; a Seção 3 detalha o funcionamento da solução proposta; a avaliação da solução é apresentada na Seção 4; e, por fim, conclusões e trabalhos futuros são apresentados na Seção 5.

2. Trabalhos Relacionados

A detecção de anomalias em um conjunto de dados é um assunto estudado em diversos domínios, tais como detecção de fraudes, processamento de imagens e monitoramento em séries temporais [Hodge and Austin 2004]. As técnicas de detecção de anomalias são usadas para ajustar, remover ou mesmo monitorar eventos anômalos. Por isso, há diversas abordagens exploradas para realizar essa tarefa de detecção: classificatória, estatística, por teoria da informação e por clusterização. Em [Ahmed et al. 2016] são encontrados exemplos de cada uma dessas abordagens no cenário de redes. Já [Hodge and Austin 2004] apresenta exemplos em diversos cenários, de abordagens estatísticas, com redes neurais, com 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, de Souza et al. 2018]. [Bawaneh and 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 pelo 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 casos foi utilizada estatística para diferenciar dados normais de dados anômalos.

No caso de múltiplas fontes de dados para detecção de anomalias em dados urbanos, [Pan et al. 2013] utiliza dados de GPS instalados em táxis e dados coletados no

Twitter para verificar o evento. Já [Sidauruk and Ikmah 2018] utiliza informações provenientes da plataforma *Twitter* e do *Waze* para analisar a correlação entre a publicação de determinados termos e a velocidade da via, usando redes neurais.

Neste trabalho, o objetivo é comparar eventos de fontes heterogêneas em tempo real com um padrão gerado pela clusterização do histórico de eventos. Dessa forma, o arcabouço é capaz de gerar alertas ao detectar eventos anômalos na mobilidade urbana.

3. Solução Proposta

A solução proposta tem como objetivo detectar eventos anômalos em tempo real, baseado no histórico de eventos coletados de fontes heterogêneas. Tendo isso em vista, o arcabouço foi definido com uma arquitetura modular, como ilustrado pela Figura 1, favorecendo a computação distribuída. Cada módulo é independente e possui responsabilidades bem definidas: **i) coletor de dados**: define um modelo de interface de dados que permite obter dados de várias fontes; **ii) pré-processamento**: filtra e armazena os dados; **iii) processamento**: define os parâmetros para gerar as séries temporais e detecta anomalias em tempo real. **iv) sistema de alertas**: emite os alertas para as autoridades envolvidas. As subseções seguintes detalham cada módulo e suas implementações.

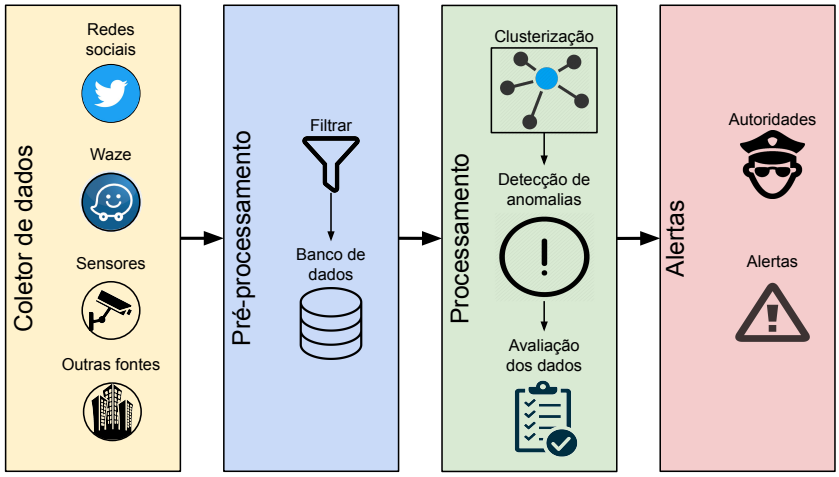


Figura 1. Visão geral do arcabouço

3.1. 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. Para isto, foi definida a seguinte interface de modelo de dados:

$$< timestamp, \textit{id\_fonte}, E = [E_1, E_2, ..., E_n] >$$

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.

O coletor foi implementado para duas fontes de dados distintas. A primeira delas é uma API de acesso aos dados disponibilizados pela prefeitura de Vitória, relativos ao *Waze*. Neste caso, o conjunto E contém nome da via e velocidade média da via congestionada. No caso da segunda fonte de dados, o *Twitter*, foi implementado um *crawler* para

buscar publicações de contas oficiais de canais de imprensa, com notícias do trânsito de Vitória. Este *crawler* retorna todos os *tweets* de uma conta específica, com um determinado *hashtag* e dentro de um intervalo de tempo. Para esta fonte, o conjunto E contém o *tweet* completo, sendo o nome da via e o evento extraídos no pré-processamento. Outras fontes de dados, como imagens de câmeras de vídeo ou dados de laços indutivos, bem como uma maior extração de dados dos *tweets* poderiam ser utilizadas para alimentar o sistema, auxiliando na detecção de anomalias. No entanto, reconhecimento de padrões em imagens, análise de linguagem natural ou o tratamento de outras fontes de dados estão fora do escopo deste trabalho.

3.2. Pré-Processamento

Este é o módulo responsável por filtrar e armazenar os dados recebidos do Coletor de Dados. Considerando que cada fonte tem sua especifidade, o filtro para cada uma deve ser criado neste módulo.

Foram implementados filtros para os dados da prefeitura e para o *Twitter*. No primeiro, remove-se dados que tratem de outras cidades além de Vitória, sem localização ou sem descrição dos eventos. Os dados válidos são incluídos no banco de dados e enviados ao módulo de processamento. Devido à natureza textual do *Twitter*, os dados não são estruturados, isto é, as informações sobre qual via e o que ocorreu na mesma não seguem um padrão. Por isso, os *tweets* são filtrados pelo nome da via encontrado na publicação e armazenados para consulta sob demanda pelo módulo de processamento, para validar se um evento detectado a partir de dados provenientes do *Waze* também aparece no *Twitter*.

3.3. Processamento

O principal objetivo do módulo de processamento é detectar anomalias de trânsito e as transmitir ao sistema de alertas em tempo real. Para isso, são realizadas três etapas:

- Calcular o maior cluster baseado nas séries temporais de um período de D dias de eventos, obtendo uma série temporal padrão;
- Comparar novos eventos com a série padrão e identificar se é uma anomalia; e
- Avaliar se existe ocorrência sobre uma anomalia em outras fontes de dados.

A implementação do arcabouço considerou os dados mantidos pela prefeitura para calcular as séries temporais devido à sua integridade e completude. Por outro lado, os dados do *Twitter* foram utilizados para verificar as anomalias detectadas.

3.3.1. Calculando a Série Temporal Padrão

Para detectar uma anomalia em tempo real, o processamento precisa comparar um novo evento com o histórico de eventos. Por exemplo, ao receber uma velocidade média de uma via, o sistema deve ser capaz de identificar se aquela velocidade está na faixa de valores esperados para o horário.

Embora a mobilidade também seja influenciada pelo dia da semana, observamos, em testes preliminares, que os resultados ao utilizar séries multidimensionais, que consideram separadamente o dia da semana (segunda a domingo) e faixa de horário, são semelhantes aos resultados obtidos ao separar os dias apenas em meio de semana e fim de semana. Ao agrupar os dados em séries temporais com apenas o intervalo de horário de dias

úteis, ganha-se em escalabilidade. Ainda, dado o custo computacional e a comparação por faixa de horário, a série temporal padrão só precisa ser calculada uma vez, totalmente ou por faixas de horário.

O objetivo desta etapa é definir uma série temporal padrão que melhor identifique as características de uma via em cada janela de horário. O Algoritmo 1 descreve como essa série padrão é obtida. Ele recebe como entrada os dados de uma via ($dados\_via$), a quantidade de dias (D) que serão considerados como histórico, o dia atual (dia_atual), o tamanho da janela em minutos ($Jmin$) que será utilizada para dividir cada dia, a função que irá agregar as observações dentro de cada janela ($Function$) e o método de clusterização ($Clustering$) que será aplicado para agrupar as séries temporais diárias.

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). A Figura 2a ilustra dois dias úteis que contêm eventos de velocidade média (m/s) em uma via. Os eventos dentro de cada janela $Jmin = 30min$ são agregados por meio de uma função $Function$ (Linhas 2-4). A função de agregação pode ser, por exemplo, soma, média, mediana ou contagem dos elementos dentro do intervalo. A Figura 2b ilustra a aplicação da função média sobre dados numéricos dentro de um intervalo, sendo que a linha laranja representa a série temporal do dia. Importante ressaltar que, nos dados disponibilizados, quando não há engarrafamentos, não há registro para a via.

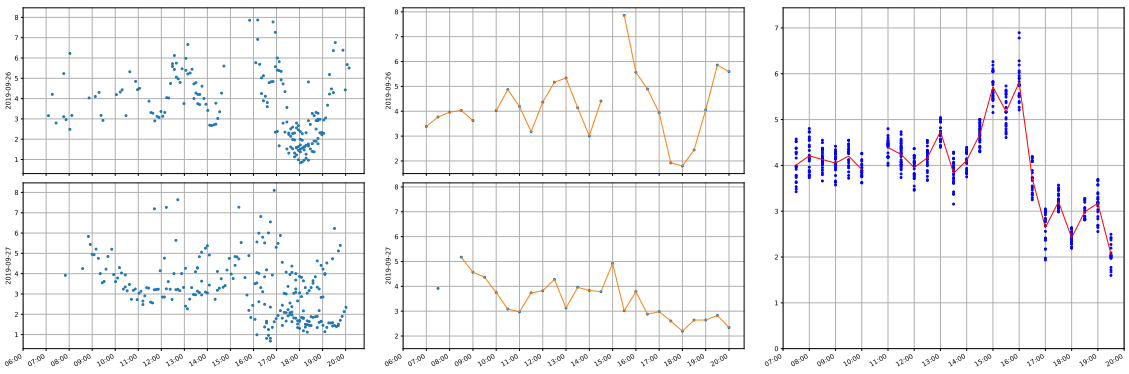
Algorithm 1: Algoritmo para agrupar e clusterizar dados

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
```

Os dados de cada série temporal em uma mesma faixa de horário são agrupadas por um algoritmo de clusterização $Clustering$ (Linhas 5-7), que retorna os maiores *clusters* encontrados, resultando em uma série temporal padrão (*pattern*). Desta forma, apenas as janelas de horários semelhantes são agrupadas. Assim, evita-se que feriados influencie a série padrão, justamente devido a falta de eventos. Por exemplo, ao aplicar a clusterização sobre as velocidades médias entre 9:00 às 9:30 de cada dia de semana, será obtido os valores mais comuns de se encontrar naquela faixa de horário. Existem vários algoritmos de clusterização na literatura [Fahad et al. 2014], dentre os quais, para efeitos de comparação, foram utilizados no arcabouço: um algoritmo hierárquico – BIRCH, complexidade *O(n)*; um baseado em particionamento – K-Means, complexidade *O(nkd)*; e dois baseados em densidade – DBSCAN e OPTICS, ambos complexidade *O(nlogn)*.

A Figura 2c ilustra a série temporal padrão a partir de um histórico de 90 dias utilizando o K-Means. Os pontos em azul pertencem ao maior cluster, enquanto a linha vermelha representa a série temporal padrão, passando pelos centros dos clusters.



(a) $dados\_via$ **(b)** Séries Temporais após Function **(c)** $pattern$

Figura 2. Exemplo de execução do Algoritmo 1

3.3.2. Detecção de Anomalias

Embora algoritmos de clusterização sejam utilizados também para detecção de anomalias (o DBSCAN e o OPTICS, por exemplo, identificam pontos ruidosos), o custo computacional para detecção de anomalia em tempo real seria alto. Por isso, cada novo evento é comparado à série temporal padrão gerada pela etapa anterior. Para isso, foram implementadas duas técnicas de detecção de *outliers*:

Z-Score : Distância normalizada entre uma amostra e a média dos dados. Calculado como $z = (x - \mu)/\sigma$, sendo x a amostra, μ a média e σ o desvio padrão. As amostras cujo $|z| > 3$ são consideradas anomalias.

IQR : Distância interquartil é uma medida de dispersão calculada pela diferença entre os percentis 75 e 25, ou seja, $IQR = Q_3 - Q_1$. Neste método, amostras que estiverem fora do intervalo $[Q_1 - 1.5, Q_3 + 1.5]$ são consideradas anomalias [Tukey 1977].

A sazonalidade de eventos urbanos possui certa variância, de forma que um evento, como engarrafamento, pode acontecer um pouco antes ou depois do horário usual. Por isso, a detecção de anomalias utiliza também as janelas imediatamente anterior e posterior ao evento para avaliar se o caracteriza um evento anômalo. Por exemplo: se um evento ocorrer às 9:20, considerando uma janela de 30 minutos, aplicamos Z-Score, ou IQR, sobre este nas janelas de 8:30 a 9:00, de 9:00 a 9:30 e de 9:30 a 10:00. Se o resultado for positivo para anomalia nas três, o evento é considerado anômalo.

3.3.3. Validação de Anomalias

Consideramos que as anomalias encontradas na etapa anterior podem ser corroboradas com dados de outras fontes, como outros aplicativos, câmeras de videomonitoramento,

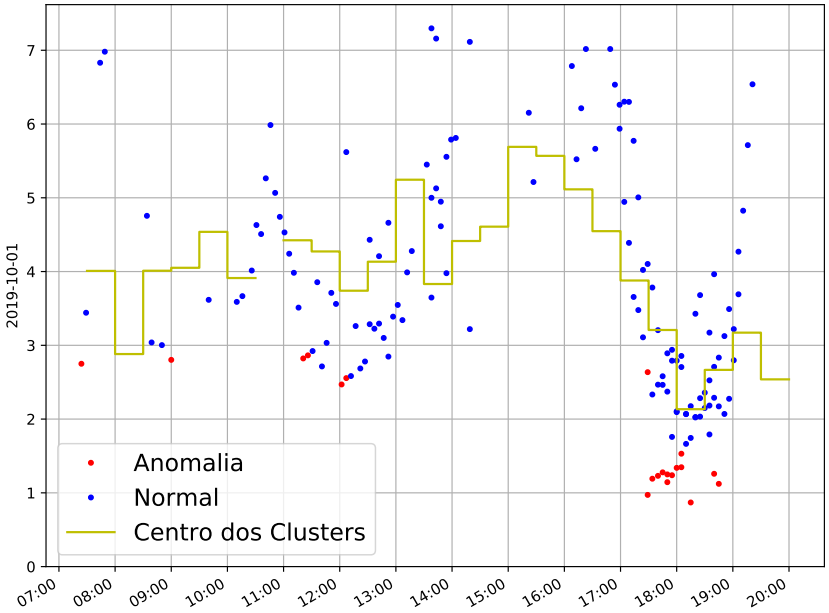


Figura 3. Exemplo de detecção de anomalias com Z-Score

sensores, redes sociais, etc. Estes dados podem ser utilizados como uma "segunda opinião", para aumentar a confiabilidade do alerta que está para ser gerado.

Neste trabalho, a coleta de dados do *Twitter* foi implementada como um "observador", responsável por coletar *tweets* em canais especializados de trânsito. 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, podemos encontrar informações atuais acerca de eventos na cidade no CBN Vitória[3], Eco101[4] e PRF ES[5].

Sempre que uma anomalia é detectada a partir de dados da prefeitura, ela é comparada com *tweets* dentro de uma janela de tempo. Caso haja a ocorrência de pelo menos um *tweet* relacionado à mesma via dentro dessa janela de tempo, aciona-se uma *flag* para indicar que a anomalia detectada tem outras fontes, gerando um alerta prioritário.

3.4. Sistema de Alertas

O módulo de alertas permite distribuir mensagens com alertas para as autoridades responsáveis. Este módulo foi implementado como um concentrador (*broker*) MQTTT[6]. 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, as autoridades interessadas em receber os alertas podem, então, assinar o tópico de anomalias, recebendo os eventos, como congestionamentos, que estejam ocorrendo fora do esperado para uma via em um determinado horário.

[3] `https://twitter.com/cbnvitoria`
[4] `https://twitter.com/_eco101`
[5] `https://twitter.com/PRF191ES`
[6] `http://mqtt.org`

4. Avaliação do Arcabouço

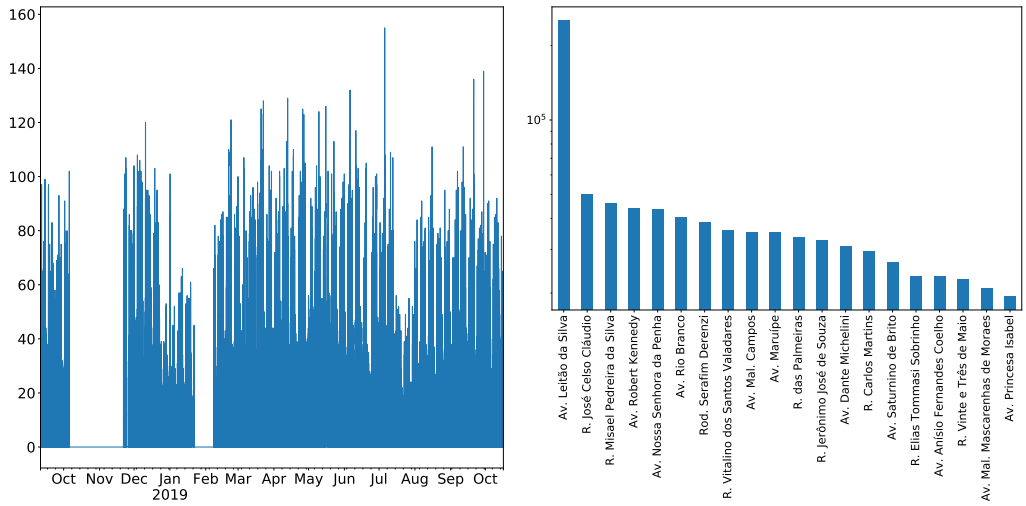
O arcabouço foi avaliado por meio de um estudo de caso da cidade de Vitória-ES, cujos dados reais de trânsito dentro do período de um ano foram disponibilizados. Primeiramente, é apresentado um resumo dos dados e justificado o intervalo utilizado para avaliar o arcabouço. Em seguida, são apresentadas as métricas de avaliação dos métodos de clusterização e de detecção de anomalias para, então, analisar os resultados obtidos.

4.1. Sumário dos Dados

A coleta dos dados foi realizada por meio de uma API fornecida pela prefeitura de Vitória, em parceria com a UFES. Esta API publica eventos de congestionamento em formato *JSON* a cada 5 minutos. Os dados disponibilizados são: alertas dos usuários, pontuação dos alertas pela avaliação dos outros usuários e velocidades em engarrafamentos. Apenas a última foi utilizada pelo arcabouço, cujos atributos são descritos na Tabela 1. Vale ressaltar que nos períodos em que não há congestionamento em uma via, a velocidade desta não estará registrada na API.

Atributo	Curta Descrição
id	Identificação única da observação
eventDate	Data e hora do alerta
city	Cidade de onde o alerta foi enviado
street	Via de onde o alerta foi enviado
speed	Velocidade média atual no segmento engarrafado da via

Tabela 1. Atributos dos eventos de engarrafamentos



(a) Eventos por minuto **(b)** Eventos nas vias

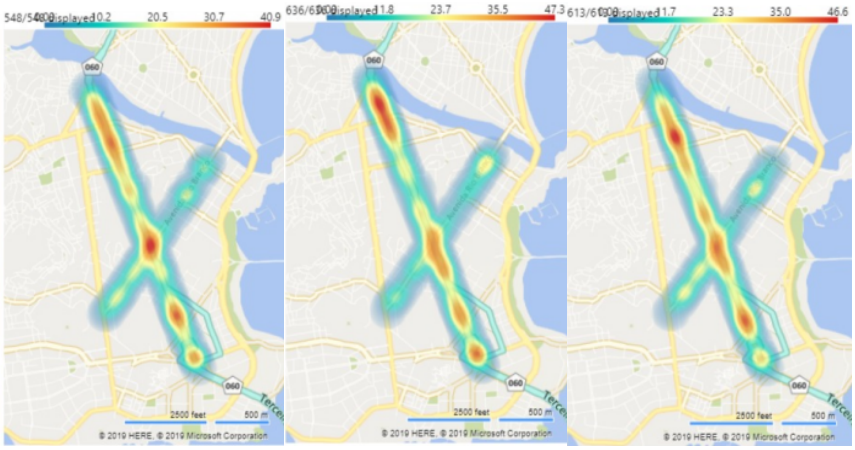
Figura 4. Número de notificações em Vitória

Foram coletados 1.916.760 eventos de trânsito das 4 cidades da Região Metropolitana entre Outubro/2018 e Outubro/2019, sendo que somente os 1.903.229 de eventos filtrados de Vitória foram considerados. O gráfico da Figura 4a apresenta o número de

eventos por minuto coletados e filtrados. Como há lacunas de eventos em Novembro/2018 e em Fevereiro/2019, foram utilizados os dados de Março a Outubro de 2019.

A Figura 4b apresenta a quantidade de eventos nas 20 vias da cidade com maior número de notificações. Nota-se que poucas vias concentram a maioria das ocorrências de evento. Com base nisso, para o desenvolvimento do arcabouço, optou-se por criar uma série temporal padrão por via.

A Avenida Nossa Senhora da Penha foi selecionada para os testes, por ser uma das principais avenidas da cidade. A Figura 5 apresenta o mapa de calor do número de alertas acumulados, onde a avenida em questão é a mais extensa no gráfico. Como exemplo, apresentamos os resultados para os alertas enviados nas terças (Figura 5a), nas quintas (Figura 5b) e o acumulado dos dias úteis entre 17:00 e 18:00 (Figura 5c). Observou-se uma similaridade entre os dias úteis, em cada faixa de horário, o que foi fundamental para definir que a série temporal padrão fosse baseada em janelas de horários, separando meios de semana de finais de semana, ao invés de uma série para cada dia da semana.



(a) Terças **(b)** Quintas **(c)** 17h-18h

Figura 5. Alertas acumulados por dia de semana e horário em duas avenidas

4.2. Métricas de Avaliação

Para avaliar e comparar os algoritmos de clusterização incluídos nesta versão do arcabouço, foram utilizadas as seguintes métricas: **i) número de anomalias detectadas**; e **ii) taxa de detecções de anomalia com eventos correlacionados** em relação ao total de detecções, neste caso, com algum evento reportado no *Twitter*. Esta taxa é calculada como $T = N_{tw}/N_{an}$, sendo N_{tw} o número de anomalias detectadas que estão relacionadas a um evento no *Twitter* e N_{an} o número total de anomalias detectadas. Baseado em testes preliminares, definiu-se a janela $tw = 90$ minutos, isto é, um evento no *Twitter* está relacionado a outro do *Waze* se ocorreram na mesma via em até 90 minutos de diferença.

O número de anomalias considera um histórico de D dias, isto é, uma série temporal para cada dia útil nos últimos D dias. Ressalta-se que o histórico é deslizante e a clusterização considera sempre os D dias anteriores ao dia analisado.

4.3. Resultados

Para obter os resultados apresentados nesta seção, com base em avaliações preliminares, foi utilizado um período $D = 90$ dias e uma janela $Jmin = 30$ minutos, mantendo um compromisso entre recursos, granularidade e variação dos resultados. A Figura 6 apresenta a quantidade de anomalias detectadas nos 15 primeiros dias de Outubro/2019 com a solução proposta. Pode-se observar que os algoritmos DBSCAN e OPTICS geraram menos detecções que o K-Means e o BIRCH. Isso demonstra que a depender do algoritmo utilizado, as autoridades poderiam receber, por exemplo, mais ou menos alertas.

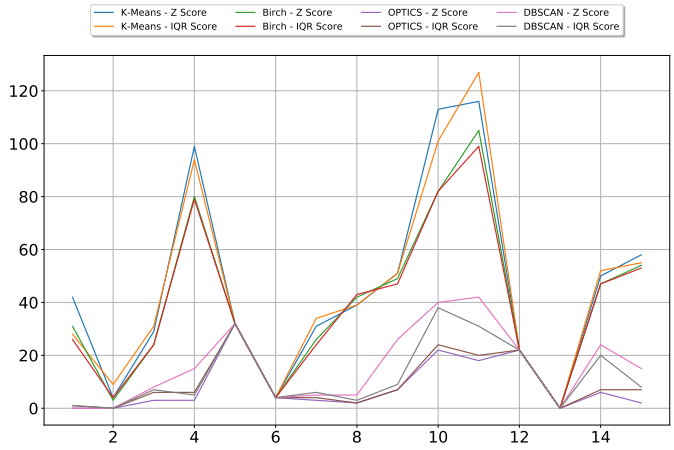
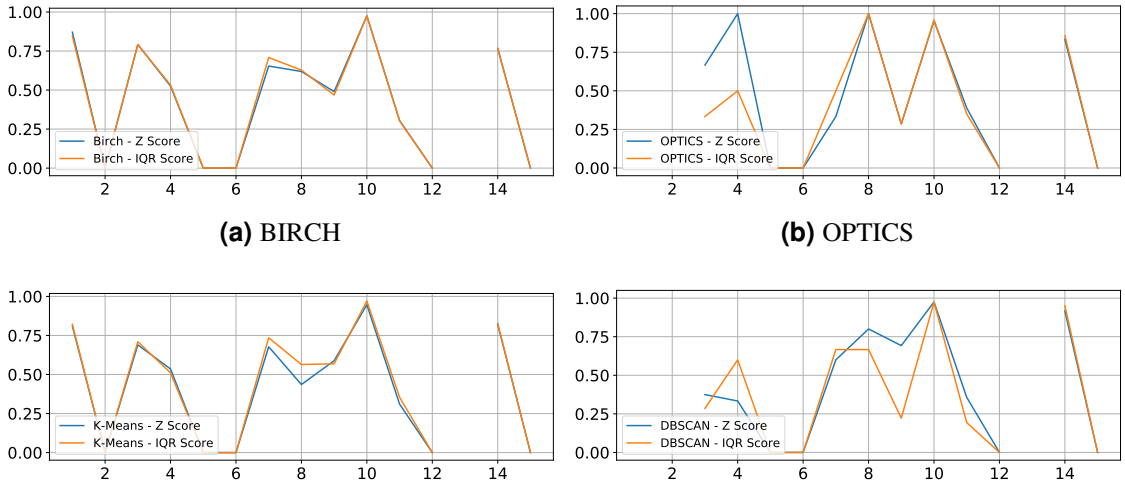


Figura 6. Número de anomalias (eixo-y) por dia (eixo-x) em Outubro/2019



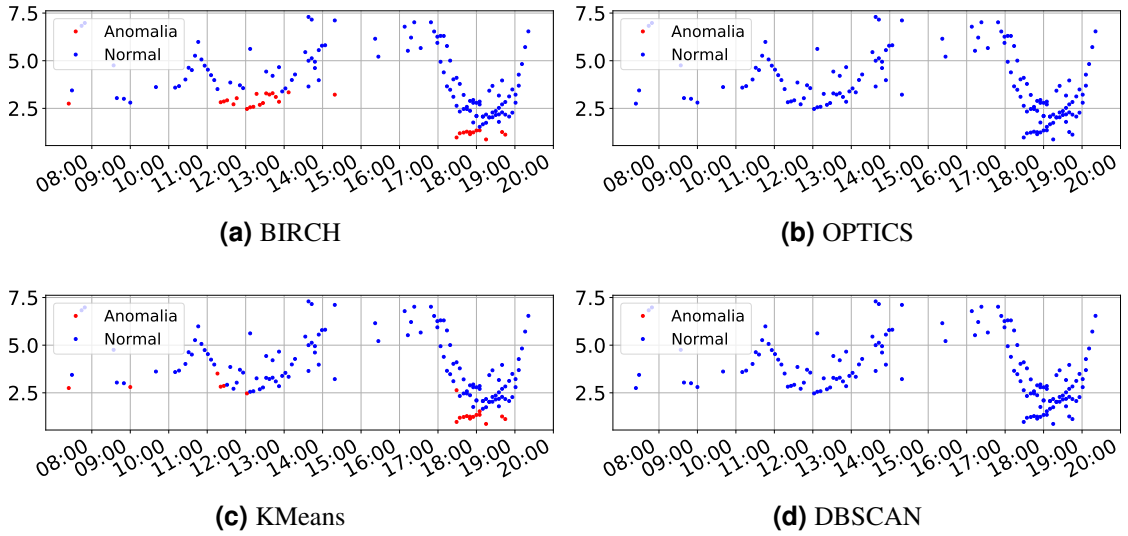
(a) BIRCH **(b)** OPTICS

(c) KMeans **(d)** DBSCAN

Figura 7. Taxa de anomalias detectadas com *tweets* relacionados (eixo-y) por dia (eixo-x) em Outubro/2019

A taxa de anomalias detectadas em que havia algum *tweet* relacionado em horário próximo nos mesmos 15 dias é apresentada na Figura 7. Os dias sem dados representam a ausência de anomalias detectadas no dia. Apesar dos algoritmos OPTICS e DBSCAN mostrarem que não houve detecções nos dois primeiros dias, observa-se que as taxas de

todos os algoritmos possuem comportamento semelhante, ou seja, apesar destes detectarem menos anomalias, eles possuem uma taxa de correspondência com o *Twitter* próxima ao K-Means e BIRCH.



(a) BIRCH **(b)** OPTICS

(c) KMeans **(d)** DBSCAN

Figura 8. Detecção de anomalias utilizando Z-Score por hora do dia

Para ilustrar o funcionamento do arcabouço em tempo real, a Figura 8 mostra as anomalias obtidas com cada algoritmo de clusterização ao longo do dia 01/10/2019. Nota-se que os métodos DBSCAN e OPTICS não apresentam grupos de anomalias detectadas, com o método Z-Score, ao passo que com os algoritmos K-Means ($k = 5$) e BIRCH há ao menos 2 grupos de anomalias detectadas, de 11:00 às 13:00 e de 17:00 às 19:00. Com o método K-Means houve outro grupo detectado às 9:00.

Entre os *tweets* identificados no dia, havia o aviso de manutenção em uma ponte no final da avenida em questão, publicada próximo às 13:00. Tendo em vista a proximidade dos dois sentidos da avenida, a obra pode ser considerada uma possível causa da redução de velocidade e, assim, do grupo de anomalias detectadas próximo do horário. Há ainda avisos às 17:30, 17:50 e 18:20 acerca de trânsito intenso e retenções, corroborando as detecções do final da tarde.

Assim, o algoritmo BIRCH agrupou as séries de forma que o Z-Score identificasse as anomalias nas mesmas faixas de horários em que surgiram *tweets* relatando problemas no trânsito, como pode ser visto na Figura 8. Embora o algoritmo K-Means tenha tido resultado similar, o Z-Score detectou um conjunto de anomalias antes das 10h que não foi possível corroborar com outras fontes.

5. Conclusão

Este trabalho apresentou um arcabouço para detecção e alerta de anomalias em tempo real em uma cidade inteligente. Para isso, foi proposta uma solução utilizando uma série temporal padrão baseada no agrupamento do histórico de valores na mesma faixa de horário em dias úteis. Desta forma, cada ocorrência em tempo real é comparada com a série para verificar se é uma anomalia. A finalidade é prover um meio de alertar autoridades sobre possíveis eventos anômalos ocorrendo nas vias, para que tomem ações mais rapidamente.

O arcabouço foi avaliado utilizando dados reais provenientes da prefeitura de Vitória-ES, e os resultados mostram um comportamento sazonal nos dias de semana, como esperado. O arcabouço mostrou-se capaz de detectar anomalias em eventos de trânsito em tempo real, considerando uma janela de tempo, e utilizou dados de uma segunda fonte, *Twitter*, para validar uma anomalia. Por fim, a solução proposta oferece um meio de detecção de anomalias para gerar alertas às autoridades.

Como trabalhos futuros, pretende-se avaliar a utilização de outras fontes de dados, como metadados de câmeras de videomonitoramento e sensores pluviométricos. Com o uso de mais fontes de dados, pode-se definir uma confiabilidade dos dados, de modo que informações semelhantes de fontes distintas sejam agregadas, o que permitiria, também, determinar quais fontes são mais confiáveis. Além disso, pretende-se adicionar diferentes níveis de severidade, ou pesos, para as anomalias detectadas, que levem em consideração, por exemplo, a distância das anomalias aos dados normais, o número de anomalias detectadas em sequência e o número de fontes que corroboram a detecção. Outro possível caminho é o uso de análise de linguagem natural para tentar obter a causa da anomalia detectada, a partir de *tweets*.

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. Os autores agradecem o esforço da secretaria pela disponibilização dos dados em tempo real.

Referências

Ahmed, M., Mahmood, A. N., and Hu, J. (2016). A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, 60:19 – 31.

Albino, V. U. B. R. M. D. (2015). Smart cities: definitions, dimensions, and performance. *Journal of Urban Technology*, pages 1723–1738.

Baloian, N., Frez, J., Pino, J. A., and Zurita, G. (2015). Efficient planning of urban public transportation networks. In *Ubiquitous Computing and Ambient Intelligence. Sensing, Processing, and Using Environmental Information*, pages 439–448, Cham. Springer International Publishing.

Bawaneh, M. and Simon, V. (2019). Anomaly detection in smart city traffic based on time series analysis. In *2019 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pages 1–6.

de Souza, A. M., Botega, L. C., Garcia, I. C., and Villas, L. A. (2018). Por aqui é mais seguro: Melhorando a mobilidade e a segurança nas vias urbanas. In *XXXVI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, Porto Alegre, RS, Brasil. SBC.

Fahad, A., Alshatri, N., Tari, Z., Alamri, A., Khalil, I., Zomaya, A. Y., Foufou, S., and Bouras, A. (2014). A survey of clustering algorithms for big data: Taxonomy and empirical analysis. *IEEE transactions on emerging topics in computing*, 2(3):267–279.

Faial, D., Bernardini, F., Miranda, L., and Viterbo, J. (2019). Anomaly detection in vehicle traffic data using batch and stream supervised learning. In *Progress in Artificial Intelligence*, pages 675–684, Cham. Springer International Publishing.

Hawkins, D. M. (1980). *Identification of outliers*, volume 11. Springer.

Hodge, V. and Austin, J. (2004). A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2):85–126.

Ladeira, L., Souza, A., Pereira, G., Silva, T. H., and Villas, L. (2019). Serviço de sugestão de rotas seguras para veículos. In *XXXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 608–621, Porto Alegre, RS, Brasil. SBC.

Montori, F., Bedogni, L., and Bononi, L. (2017). A collaborative internet of things architecture for smart cities and environmental monitoring. *IEEE Internet of Things Journal*, 5(2):592–605.

Montori, F., Bedogni, L., Di Chiappari, A., and Bononi, L. (2016). Sensquare: A mobile crowdsensing architecture for smart cities. In *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, pages 536–541, Reston, VA, USA. IEEE.

Pan, B., Zheng, Y., Wilkie, D., and Shahabi, C. (2013). 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*, pages 334–343.

Panagiotou, N., Zygouras, N., Katakis, I., Gunopulos, D., Zacheilas, N., Boutsis, I., Kalogeraki, V., Lynch, S., and O'Brien, B. (2016). Intelligent urban data monitoring for smart cities. In *Machine Learning and Knowledge Discovery in Databases*, pages 177–192, Cham. Springer International Publishing.

Purnomo, F., Heryadi, Y., Gaol, F. L., and Ricky, M. Y. (2016). Smart city's context awareness using social media. *2016 International Conference on ICT for Smart Society, ICISS 2016*, pages 119–123.

Rathore, M. M., Ahmad, A., Paul, A., and Rho, S. (2016). Urban planning and building smart cities based on the internet of things using big data analytics. *Computer Networks*, 101:63–80.

Sidauruk, A. and Ikmah (2018). Congestion correlation and classification from twitter and waze map using artificial neural network. In *International Conference on Information Technology, Information System and Electrical Engineering*, pages 224–229.

Silva, T. H., Celes, C., Neto, J., Mota, V., Cunha, F., Ferreira, A., Ribeiro, A., Vaz de Melo, P., Almeida, J., and Loureiro, A. (2016). Users in the urban sensing process: Challenges and research opportunities. *Pervasive Computing: Next Generation Platforms for Intelligent Data Collection*, pages 45–95.

Tukey, J. W. (1977). *Exploratory data analysis*. Reading, Mass. : Addison-Wesley Pub. Co.