Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Visualização de dados geoespaciais: Uma solução para a visualização de anomalias em redes móveis

George Geonardo de P. da Silva

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Orientador
Prof. Dr. Marcelo Marotta

Coorientador
Prof. Dr. Célia Ghedini Ralha

Brasília
2022

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Visualização de dados geoespaciais: Uma solução para a visualização de anomalias em redes móveis

George Geonardo de P. da Silva

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Prof. Dr. Marcelo Marotta (Orientador)
CIC/UnB

Prof. Dr. Aletéia Patricia Favacho de Araujo M.Sc. Jonathan Mendes de Almeida
Universidade de Brasília

Prof. Dr. Marcelo Grandi Mandelli
Coordenador do Bacharelado em Ciência da Computação

Brasília, 26 de Setembro de 2022

Dedicatória

Dedico o projeto aos meus pais, que sempre estiveram ao meu lado tanto nos momentos bons e ruins. Também agradeço a eles por me darem segurança e suporte na perseguição dos meus objetivos.
Dedico também a meus amigos e colegas de curso que me ajudaram de alguma forma, mas dedico especialmente ao Daniel Machado Faria, ao Jonathan de Almeida e a Brenda Barbosa. Cada um deles me ajudou de sua forma na minha vida, seja diretamente ou indiretamente, e que sem eles eu provavelmente não teria chegado tão longe.
Durante todo o percusso acadêmico eu aprendi e internalizei vários tipos de conhecimentos, porém o mais importante que talvez eu possa interpretar e expressar seria de que: "*Não é errado pedir a ajuda de alguém, mas principalmente, não é errado pedir ajuda a você mesmo*".

Agradecimentos

Agradeço ao meu orientador Marcelo Marotta pela atenção, generosidade e flexibilidade que foram sempre presentes durante o trabalho. Agradeço também ao Jonathan por me aconselhar bastante no projeto, e ao professor Marcelo Grandi Mandelli pelo auxílio com problemas pontuais no curso. A ajuda de todos me auxiliaram no término da proposta e na conclusão do ensino.

Agradeço por fim a Universidade de Brasília, pois a universidade além de instruir sobre o campo acadêmico, ensina sobre aspectos individuais e da vida, que anteriormente você ignorava ou não procurava entender.

Resumo

Atualmente, a tecnologia 5G está cada vez mais ganhando espaço na vida de todos, especialmente em países de primeiro mundo. Um dos principais recursos da tecnologia 5G é o aumento da vazão de download para os dispositivos conectados à rede, incluindo também, um canal de comunicação mais estável, confiável e seguro quando comparado com as tecnologias anteriores. No entanto, a medida em que essas células são requisitadas, principalmente em centros urbanos, tem-se como resultado instabilidades causadas pelo acúmulo de erros na rede. Estas instabilidades atrapalham a experiência do usuário, e por consequência causam perdas de faturamento para as provedoras. Nesse contexto, são propostas soluções para a detecção de anomalias com o uso de Inteligência Artificial e Aprendizado de Máquina, criando modelos capazes de detectar anomalias em dados de redes móveis, chamados de Call Detail Records (CDRs). Entretanto, apesar dos ganhos com uso de AM na identificação de anomalias na rede serem evidentes ao analisar publicações recentes, não é seguro afirmar que todos os problemas de redes estão precisamente identificados e resolvidos. Considerando o contexto de soluções baseadas em Inteligência Artificial (IA) e Aprendizado de Máquina (AM) para identificação de anomalias, o presente trabalho propõe uma solução que busca aprimorar a forma que os resultados obtidos de anomalias em redes móveis são visualizados. Para a detecção dessas anomalias e seus resultados, foi utilizada uma técnica estatística não supervisionada orientada a IA. O projeto implementado alcançou os objetivos propostos no manuscrito, sendo sua maior contribuição a sua capacidade de exibir os dados de CDRs e anomalias em um mapa. Por fim, outra contribuição é referente à geração dos gráficos e exportação de dados sobre as anomalias.

Palavras-chave: Visualização Geoespacial, Redes Móveis, Anomalias de Rede

Abstract

Currently, 5G technology is increasingly gaining space in everyone's lives, especially in first world countries. One of the main features of 5G technology is the increased download rate for devices connected to the network, also including a more stable, reliable and secure communication channel when compared to previous technologies. As these cells are requested, mainly in urban centers, we have as a result instabilities in the network that are caused by the accumulation of errors. These instabilities hinder the user experience, and consequently cause billing losses for network providers. In this context, solutions are being proposed for the detection of anomalies using Artificial Intelligence and Machine Learning, that creates models capable of detecting anomalies in mobile network data, labeled Call Detail Records (CDRs). However, although the gains from using AM in identifying network anomalies are evident when analyzing recent publications, it is not safe to say that all network problems are precisely and accurately identified and resolved by the models. Then, by considering the context of solutions based on AI and AM to identify anomalies, the present work proposes a solution that seeks to improve the manner which the results obtained from anomalies in mobile networks are visualized. For the detection of these anomalies and their results, an AI-oriented unsupervised statistical technique was used. The implemented project achieved the objectives proposed in the manuscript, its greatest contribution being its ability to display CDRs and anomalies data on a map. Finally, another contribution is related to the generation of graphs and data export about the anomalies.

Keywords: Geospatial Visualization, Mobile Networks, Network Anomalies

Sumário

Lista de Figuras

Lista de Abreviaturas e Siglas

AM Aprendizado de Máquina.

CDR Call Retail Record.

EFC Energy Flow Classifier.

IA Inteligência Artificial.

MB MapBox.

OPM OpenStreetMap.

RF Random Forest.

Capítulo 1

Introdução

Atualmente, a tecnologia 5G está cada vez mais ganhando espaço na vida de todos, especialmente, em países de primeiro mundo. No entanto, mesmo que em muitos lugares, inclusive no Brasil, a tecnologia 5G ainda não esteja disseminada para a maioria das pessoas, já é possível afirmar que a tecnologia 5G revolucionará várias áreas. Conforme a tecnologia consiga mais adeptos, não somente a telefonia móvel será transformada, mas também por exemplo a indústria de filmes, de redes sociais, a da internet incluindo a internet das coisas, a da medicina e a da agricultura.

Um dos principais recursos da tecnologia 5G é o aumento da velocidade de *download* para os dispositivos conectados à rede, incluindo também, um canal de comunicação mais estável, confiável e seguro quando comparado com as tecnologias anteriores. Por mais que exista melhorias na tecnologia, isso não significa que a tecnologia 5G não tenha problemas ainda não solucionados ou oportunidades de melhoria. A rede 5G, assim como a 4G, faz o uso de áreas geográficas para a prestação dos serviços de redes móveis por meio de áreas de serviços chamadas de células. A medida em que essas células são requisitadas, principalmente em centros urbanos, temos como resultado instabilidades na rede. As instabilidades são causadas pelo acúmulo de erros. Estas instabilidades atrapalham a experiência do usuário, e por consequência causam perdas de faturamento para as provedoras [1] [2].

O uso de outros recursos presentes na tecnologia 5G serão utilizados com maior frequência e com o passar dos anos se popularizará pelo mundo. O suporte ao gerenciamento e o controle de redes por meio de Inteligência Artificial (IA) é um dos recursos que fará

com que novas aplicações ajudem a identificar anomalias que surgem na rede [3]. Estas aplicações inteligentes serão imprescindíveis em identificar e tratar anomalias que causam erros nas redes, e para estas aplicações serem capazes de tomar as ações apropriadas o melhoramento da coleta dos dados é necessário. Assim o uso de soluções de IA ajudarão no aprimoramento da coleta e de soluções. Paralelamente o fato destes novos desafios existirem, naturalmente faz com que pesquisas relacionadas a detecção e tratamento de anomalias nas redes 5G sejam elaboradas [4].

1.1 Problema

A tecnologia 5G proporciona o aumento da largura de banda e confiabilidade de conexão e, por conseguinte, há um aumento de tráfego de rede em áreas específicas. Nesse contexto, pontos de acessos serão requisitados com maior frequência, e consequentemente usados por mais tempo nestas áreas. Assim, conforme mais dispositivos de diversos tipos são conectados à rede, tais como telefones, carros e notebooks, maior será o acúmulo de erros nas redes. Particularmente, em áreas urbanas, o acúmulo de erros causados por tráfego fora do padrão, isto é, anomalias na rede, podem ocasionar interrupções e/ou falhas nas células.

Atualmente, autores estão propondo soluções para a detecção de anomalias com o uso de IA e Aprendizado de Máquina (AM), criando modelos capazes de detectar anomalias em redes móveis. Dessa forma, os erros que ocorrem em redes móveis podem ser detectados, evitados ou corrigidos. Entretanto, apesar dos ganhos com uso de AM na identificação de anomalias na rede serem evidentes ao analisar publicações recentes, não é seguro afirmar que todos os problemas de redes estão precisamente identificados e resolvidos.

Existem vários fatores que afetam os modelos, por exemplo, a mobilidade do usuário, a qualidade e o comportamento do canal [5] [6]. Um grande desafio na criação de aplicações inteligentes baseadas em AM são os padrões de tráfegos. Na literatura, geralmente é assumido que os padrões são similares, e os dados estão ordenados e catalogados. No entanto, os dados podem não seguir um padrão para todos os casos e, muitas vezes, os dados não são devidamente documentados e tabelados. Infelizmente, na prática o que se encontra são conjunto de dados desordenados, mal classificados, com erros, duplicados

e até mesmo faltando campos importantes. Com isso, torna-se importante a tarefa de organizar, ordenar e documentar os dados para serem usados na modelagem de soluções de, por exemplo, detecção de anomalias.

Uma boa documentação textual com componentes visuais possui um potencial intrínseco de fomentar o desenvolvimento de novas soluções. Especificamente no contexto de soluções para detecção de anomalias, pois muitas vezes as soluções propostas são baseadas em uma classificação binária de anomalias. Isto implica que um tráfego só pode ser normal ou anormal, não existindo assim um meio termo. No entanto, os autores de [7] mostraram que anomalias podem variar em graus, isto é, pode haver anomalias que são "mais anormais"do que outras. Além disso, há um grande problema na forma em que os resultados obtidos, as anomalias detectadas, são apresentados.

Neste contexto de soluções baseadas em IA e AM para identificação de anomalias, o presente trabalho propõe uma solução que busca aprimorar a forma que os resultados obtidos de anomalias em redes móveis são visualizados. Para a detecção dessas anomalias e seus resultados, foi utilizada uma técnica estatística não supervisionada orientada a IA [7].

1.2 Objetivos

Este trabalho tem como objetivo a implementação de uma aplicação web que facilita a visualização dos dados obtidos. Essa implementação não só amplia a divulgação dos resultados obtidos em [7], mas que também possibilita a visualização de Call Detail Records (CDRs) de forma geral. Os dados de anomalias foram adquiridos por meio do uso de uma técnica estatística não supervisionada orientada a IA com o uso de AM. Além disso, esta aplicação possibilita a exportação de dados e arquivos gerados na visualização. Inicialmente, de acordo com o dia escolhido pelo usuário, os dados são apresentados em forma de quadriláteros. Assim, estes quadriláteros representam com suas respectivas cores um mapa de calor do tráfego de rede. Cada um destes quadros possuem a mesma área, e eles compõe um *grid* 100 x 100 quadros sobre a cidade de Milão, na Itália. Dessa forma, a visualização em forma de quadriláteros ajudam o usuário a distinguir as áreas urbanas e rurais. Com efeito, pode-se observar em quais áreas há uma maior ou menor

concentração de dispositivos na rede, que por meio das cores, representam o volume de tráfego da área. Além disso, a ferramenta disponibiliza a opção de transferir os arquivos para a máquina do cliente. Os dados estarão em formato *csv* e *geojson*.

Acresce que, além da visualização do mapa de calor, o usuário também será capaz de visualizar outros dados relacionados ao quadrilátero. Estes dados por quadro seriam a quantidade de anomalias da área escolhida, a posição geográfica e o identificador do quadrilátero, bem como os gráficos gerados pela aplicação.

O trabalho proposto faz o uso dos dados resultantes através dos CDRs da cidade de Milão, na Itália. Logo, a proposta possui um potencial de contribuição para qualquer área de pesquisa que tenha como objeto de estudo os CDRs e que os dados dos modelos estatísticos necessitam ser visualizados.

1.3 Metodologia

O propósito do projeto é a implementação de uma ferramenta capaz de exibir resultados de anomalias sobre um mapa. Os dados consistem de CDRs obtidos pelo sistema de gerenciamento da rede Telecom Italia em Milão, e atualmente acessível através do Harvard Dataverse. A Telecom Italia forneceu um arquivo *geojson* que especifica cada região coberta pela rede, além dos CDRs das áreas.

Estes dados CDRs foram colhidos por 62 dias, e geraram um conjunto de dados, o qual o conjunto teve elementos extraídos para a aplicação do algoritmo responsável por identificar anomalias. No entanto, o resultado do algoritmo aplicado depende da granularidade do tempo para detectar as anomalias de forma precisa. Assim, o uso de um intervalo de tempo menor pode ter como consequência um aumento na precisão, porém, o uso dos recursos computacionais também crescerá neste caso [7].

Os resultados obtidos, pelo algoritmo, são usados pela aplicação, para assim, exibir as anomalias e seus respectivos dados geoespaciais. Então, consequentemente, o algoritmo foi capaz de criar o agrupamento dos dados para cada dia, com comportamentos anormais na rede, considerando cada posição de célula e tráfego agregado. Acrescenta-se, que os dados quantitativos obtidos pelos autores [7], também são usados na geração de gráficos, que auxiliam na análise das anomalias.

Para o desenvolvimento deste trabalho foi necessário o pré-processamento dos dados resultantes do algoritmo, pois o arquivo gerado era único, e as anomalias identificadas eram prontamente inseridas no arquivo. Portanto, os dados destas anomalias não eram ordenados. Assim, primeiramente foi feito a ordenação do arquivo único, e depois foi feita a divisão deste arquivo único em outros 52 arquivos distintos. Cada arquivo, representa um dia, ou seja, representam os dados que começam a partir de 1 de novembro e terminam no dia 22 de dezembro.

Após a primeira etapa do pré-processamento do arquivo único para arquivos únicos menores, foi necessário um processamento adicional para converter estes arquivos do formato *csv* para o formato *geojson*. Isso foi necessário para atender as restrições de utilização de recursos pelo lado do servidor, e consequentemente, também melhorar a experiência do usuário.

1.4 Apresentação do Manuscrito

O trabalho possui, contando com este capítulo, 5 capítulos. No Capítulo 2, é apresentado quais os fundamentos teóricos envolvidos no trabalho, o qual este projeto é baseado, pendendo a conceitos da estatística, do AM e de modelos quânticos. Além disso, o capítulo 3 expõe uma síntese dos trabalhos relacionados na detecção de anomalias em redes móveis. Acrescenta-se que os trabalhos citados utilizam o conjunto de dados CDRs, e apresentam soluções que podem ser de modelos supervisionados ou não supervisionados. Já no Capítulo 4 a solução desenvolvida é descrita detalhando o planejamento, o problema, as funcionalidades, a implementação do protótipo, e por fim, quais resultados o projeto entregou. Finalmente, no Capítulo 5 são apresentadas as conclusões sobre a contribuição do trabalho realizado, e os possíveis trabalhos futuros para prosseguir com melhorias para a ferramenta.

Capítulo 2

Fundamentação Teórica

Este capítulo apresenta a fundamentação teórica do método utilizado para a detecção de anomalias em redes móveis proposto em [7], em que os autores propuseram o Energy Flow Classifier (EFC), um modelo estatístico capaz de detectar anomalias em dados CDRs. Por meio dessa solução baseada em IA, o EFC considera o padrão de tráfego presente de regiões geográficas diferentes e também os horários do dia. Nesse contexto, o uso não supervisionado de AM foi julgado oportuno para o algoritmo proposto pelo autor, ainda mais considerando que o algoritmo tem complexidade linear, sendo muito vantajoso quando comparado com outros algoritmos conhecidos e vastamente utilizados, como por exemplo o Random Forest (RF).

2.1 Energy-Based Flow Classifier para Detecção de Anomalias em Redes Móveis

Um sistema de gerenciamento de tráfego é capaz de coletar informações de redes móveis em regiões diferentes ao longo do tempo. Assim, cada tráfego avaliado pode ser caracterizado e modelado usando ideias encontradas na mecânica quântica, como mostrado em [7]. Diante disso, o conceito intuitivo do EFC é baseado no modelo Potts Figura 2.1 [8]. Cada dado de tráfego i é representado por uma configuração de grafo específico $G_k(\eta, \varepsilon)$.

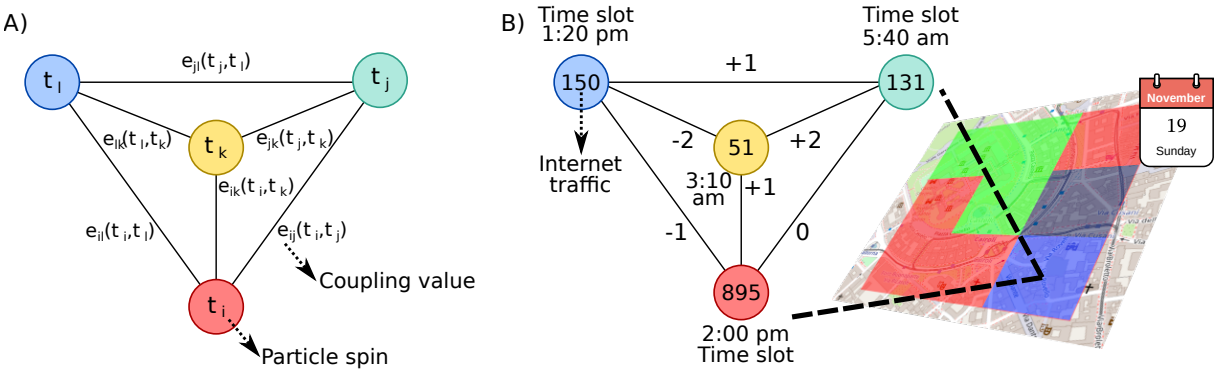

Figura 2.1: Intuição do funcionamento do EFC - A) Spins interativos em uma rede cristalina. B) Tráfego de rede mapeado em uma estrutura de grafos.

Seja $(T_1, ..., T_N)$ uma N-tupla de uma faixa de tempo durante o dia e, seja $s_{d,sq} = (t_1, ..., t_N)$ uma amostra dos volumes discretizados de tráfego coletados na data d, na região sq, tal que t_i é a quantidade discretizada de tráfego durante uma faixa de tempo T_i, $i = 1, ..., N$. Os valores contínuos do volume de tráfego presente no conjunto de dados são discretizados em quanta Q, codificados no alfabeto $\Omega = \{1, ..., Q\}$. Entretanto, $\varepsilon = \{(i, j)|i, j \in \eta; i \neq j\}$ é o conjunto de arestas determinado por todos os pares possíveis da faixa de tempo, criando assim uma malha de grafos que pode representar várias amostras de tráfego diferentes através de características em comum. Cada aresta possui um valor associado determinado pela função $e_{ij}(t_i, t_j)$.

Antes de calcular os acoplamentos, é necessário adquirir as frequências empíricas conjuntas e únicas $f_i(t_i)$ e $f_{ij}(t_i, t_j)$. Estas frequências são obtidas a partir de todas as amostras de treinamento $s_{d,sq} \in S_{training}$. Dessa maneira, as ocorrências das anomalias são avaliadas de um certo volume de tráfego t_i ou par de volume de tráfego (t_i, t_j). Então, ao contar as ocorrências das anomalias no volume de tráfego ou pares de tráfego, divide-se o valor obtido pelo número total de amostras em $S_{training}$.

Como o conjunto $S_{training}$ é finito e pequeno, se comparado ao universo de todas as distribuições possíveis do tráfego de volume durante um dia, faz com que interferências baseadas no $S_{training}$ sejam suscetíveis a subamostragem. Seguindo a abordagem teórica proposta por Morcos [9], e outros, houve a inserção de dados não coletados que possam existir nas frequências empíricas, para assim limitar os efeitos da subamostragem ao realizar as

seguintes operações:

$$f_i(t_i) \leftarrow (1-\alpha)f_i(t_i) + \frac{\alpha}{Q} \tag{2.1}$$

$$f_{ij}(t_i,t_j) \leftarrow (1-\alpha)f_{ij}(t_i,t_j) + \frac{\alpha}{Q^2} \tag{2.2}$$

tal que $(t_i,t_j) \in \Omega^2$, $0 \leq \alpha \leq 1$ é um parâmetro que define o peso dos dados e Q significa a cardinalidade de Ω. A inserção de dados é equivalente a assumir que $S_{training}$ é estendido com uma fração do tráfego das amostras de características uniformes.

Assim sendo, os pares diretos entre cada volume de tráfego em cada período de tempo são calculados da seguinte forma:

$$e_{ij}(t_i,t_j) = -(C^{-1})_{ij}(t_i,t_j), \tag{2.3}$$
$$\forall (i,j) \in \{1,...N\}^2, \forall (t_i,t_j) \in \Omega^2, t_i,t_j \neq Q$$

tal que

$$C_{ij}(t_i,t_j) = f_{ij}(t_i,t_j) - f_i(t_i)f_j(t_j) \tag{2.4}$$

é a matriz de covariância obtida por frequências únicas e conjuntas. Para remover o efeito de correlação indireta nos dados foi obtido a matriz inversa da matriz de covariância.

O número de restrições independentes considerados para a interferência desde modelo estatístico é $\frac{N(N-1)}{2}(Q-1)^2 + N(Q-1)$ ainda que o modelo tenha $\frac{N(N-1)}{2}Q^2 + NQ$ parâmetros. Sem a perda de generalidade, é definido que:

$$e_{ij}(t_i,Q) = e_{ij}(Q,t_j) = 0. \tag{2.5}$$

Assim, não é necessário calcular $e_{ij}(t_i,t_j)$ quando t_i or t_j é igual a Q [9] [10].

Finalmente, é possível usar estes acoplamentos diretos para calcular a energia Hamiltoniana de uma dada amostra $s_{d,sq} = (t_1,...,t_N) \in S_{testing}$ no conjunto de testes de uma certa faixa de tempo T_i: $s_{d,sq} = (t_1,...,t_N) \in S_{testing}$ in the testing set at a given timeslot T_i:

$$\mathcal{H}_i(s_{d,sq}) = -\sum_{j|i\neq j} e_{ij}(t_i,t_j), i = 1,...,N. \tag{2.6}$$

Apresentando uma anomalia na faixa de tempo T_i, tem-se que a energia será inversamente proporcional a probabilidade da amostra $s_{d,sq}$. Uma amostra de teste é considerada anormal na faixa de tempo T_i, se a energia está acima do limiar:

$$\tau_1 = \mathcal{H}_i(s_{d,sq}) > avg(\mathcal{H}_i(s_{d,sq})) + 4std(\mathcal{H}_i(s_{d,sq})) \tag{2.7}$$

para todo $s_{d,sq} \in S_{training}$ [11].

2.2 Complexidade do algoritmo

A complexidade do EFC é:

$$O(K'[N + M^3Q^3 + NM^2Q^2])$$

tal que N é o número de instâncias, ou seja, é a amostra do conjunto de treino, K' é o número de regiões usadas, M é o número de elementos e Q é o tamanho do alfabeto [7]. Como a complexidade da etapa de classificação é quadrática de acordo com o número de elementos usados M, e linear para o conjunto de dados de teste N mais o número das regiões utilizadas K':

$$O(NM^2).$$

visto que os elementos no contexto do trabalho significam o volume de tráfego e o carimbo de data. Em vista disso, os números de elementos, como também os tempos de classificação, podem ser mantidos pequenos.

O agrupamento *k-means*, que é um algoritmo amplamente utilizado na literatura para a detecção de anomalias em redes móveis, apresenta complexidade linear em número de dados de objetos:

$$O(KN)$$

onde K é o número de agrupamentos e N é o número de dados de objetos. Comparando a complexidade computacional EFC, com o agrupamento *k-means*, tem-se que a solução apresentada possui complexidade similar. Consequentemente, ao separar a cidade em diferentes regiões, e com cada região contendo quadriláteros de comportamentos simila-

res, um modelo estatístico é inferido para cada contexto. Portanto, com os resultados do modelo pode-se observar os níveis de energia de cada cenário, e com isso é possível diferenciar os tipos de anomalias decorrentes, que vão desde as quedas das células de rede até a subutilização destas.

A solução apresentada por [7] mostra uma baixa complexidade computacional e alto grau de flexibilidade em considerar os diferentes tipos de anomalias. Diante disso, as anomalias podem ser classificadas, e o *top-k* nos espectros de aumento e diminuição do volume de tráfego podem ser selecionados pelo gerenciador de rede. Apesar das soluções estatísticas baseadas em AM serem capazes de detectar anomalias em regiões inteiras, o modelo inferido pode ser melhorado com treinamento separado por regiões de comportamentos semelhantes. Igualmente, às atividades de tráfego variam consideravelmente entre zonas urbanas, rurais e periféricas. Assim, ao combinar estas características referentes à atividades de tráfego com a solução apresentada do algoritmo de agrupamento baseado em IA, é possível detectar o grau das anomalias considerando seus padrões temporais e também espaciais [7].

Capítulo 3

Trabalhos Correlatos

Neste capítulo, são discutidos os trabalhos relacionados que fundamentam este ensaio. Os autores e especialistas usados como inspiração possuem estudos e referências para a detecção de anomalias em redes móveis. Em particular, os pesquisadores citados geralmente focam no uso de soluções inteligentes e autônomas. No entanto, alguns autores também sugerem o uso de soluções heurísticas.

As técnicas apresentadas pelos autores de cada trabalho podem ser divididas em técnicas baseadas em agrupamento, incluindo a aprendizagem não supervisionada, e também dividida em técnicas baseadas em classificação, ou seja, baseadas em aprendizagem supervisionada. Além disso, a maioria dos trabalhos não consideram que há anomalias de diferentes graus com escopo em áreas com menos ou mais tráfego (por exemplo, áreas rurais e áreas urbanas). Entretanto, todos os trabalhos citados possuem algo em comum: a falta de uma ferramenta para a visualização e divulgação dos resultados obtidos.

3.1 Aprendizagem Supervisionada

Diferente trabalhos fizeram o uso de aprendizado profundo na detecção de anomalias em redes móveis, como os autores em [12], com o uso dos dados CDRs, aplicaram redes neurais de memória de curta e de longo prazo para a detecção de anomalias. O cenário usado pelos pesquisadores foi bem definido, pois os dados coletados foram classificados e rotulados de acordo com o *groundtruth* de cada região. No entanto, o trabalho não possui uma ferramenta que facilmente apresente os dados obtidos.

As soluções propostas por [13] e [1] indicam somente se há alguma anomalia ou não na área de rede escolhida, sendo que a área de rede é baseada no conjunto de CDRs da cidade de Milão, na Itália. Por consequência, não são analisados os graus das anomalias nos resultados obtidos por meio de redes neurais *feed-forward*, e novamente não há uma ferramenta simples para exibir estes dados.

3.2 Aprendizagem Não-Supervisionada

Na literatura, vários trabalhos propuseram soluções baseadas em aprendizagem não supervisionada para a detecção de anomalias, porém os autores de [14] apresentam uma solução que faz o uso do método de agrupamento *k-means* e agrupamento hierárquico. Ademais, o mesmo trabalho também faz o uso do conjunto de dados CDRs da cidade de Milão.

Diferentemente, os autores de [15], apesar de terem utilizado o método de agrupamento *k-means*, decidiram eliminar as anomalias dos conjuntos de dados para melhor preverem o tráfego de rede em vez de catalogar as anomalias. Os autores agruparam poucas regiões adjacentes com um intervalo de tempo de 1 hora durante uma semana. Com isso, houve a diminuição do custo do processamento computacional para utilizar soluções baseadas em AM não supervisionada, como o *k-means*.

Já os autores de [6] aplicaram uma abordagem interessante. Eles consideraram em suas estimações os diferentes padrões de tráfego presentes na cidade de Milão usando agrupamento *k-means*. No entanto, apesar dos autores aplicarem essa abordagem antes de detectar as anomalias, não são analisados os efeitos desta abordagem nos próprios padrões de tráfego.

Analogamente, foi apresentado pelos autores em [11] uma solução de aprendizado estatístico semi-supervisionado, com o uso de distribuição Gaussiana. O modelo foi treinado usando uma única região em Milão e esta região então foi comparada com as demais regiões. Decorrente do uso de modelos estatísticos, que requerem aplicação com distribuições similares, o custo computacional foi menor. Caso contrário, seria inviável computacionalmente comparar com outras áreas adjacentes para encontrar anomalias no tráfego de rede.

De forma diferente, os autores de [16], com o uso de dados simulados e não geográficos,

apresentam uma solução baseada em entropia para a detecção de anomalias em redes móveis.

Por fim, tendo todos esses trabalhos como base, os autores de [7] escolheram como base uma solução estatística, porém melhorando as capacidades de processamento. Este aprimoramento foi alcançado pela adaptação dos modelos da mecânica quântica, para a detecção de anomalias em tráfegos de rede, com localidades e tempos diversos. O resultado foi um algoritmo capaz de treinar um modelo estatístico, com adaptações do modelo quântico, que foi capaz de prever os padrões de atividades nas células de redes móveis. Inclusive, por meio deste modelo há como detectar e interpretar os graus das anomalias ao considerar os diferentes comportamentos de cada região.

Em suma, diante todos estes trabalhos e pesquisas, pode-se observar que, não há uma ferramenta capaz de apresentar facilmente os resultados obtidos. Então, o projeto aqui descrito neste manuscrito, e implementado, pretende oferecer uma solução para a melhor exibição dos dados obtidos pelas pesquisas citadas no capítulo, especialmente os resultados dos autores [7].

Capítulo 4

Proposta de Solução

Este capítulo apresenta a proposta deste trabalho, incluindo o seu planejamento, funcionalidades, a implementação do protótipo e os resultados obtidos. A proposta de solução tem como objetivo ampliar a visibilidade e a utilização dos resultados apresentados que foram obtidos baseados na solução proposta em [7]. A apresentação dos dados foi feita de forma visual, com interações geoespaciais além da exibição dos gráficos gerados pela aplicação. A interface gráfica auxilia o usuário em avaliar as anomalias de acordo com seus níveis de energia, com que frequência estas anomalias ocorrem e quais suas energias médias. Assim, atrelado ao local geoespacial, estes gráficos podem representar um local de alto volume de tráfego, como uma área urbana, ou um local de baixo volume de tráfego, como uma área rural.

4.1 Planejamento

Um calendário para o gerenciamento do tempo de implementação da aplicação e os casos de uso foram esboçados para a preparação da aplicação. Assim, por meio desses casos de uso alguns requisitos limitantes foram encontrados. Com isso foi possível ter uma base das tecnologias necessárias e quais limitações poderiam ser problemáticas como por exemplo, a limitação de que o servidor não faria nenhum cálculo, e sim somente as transferências dos dados necessários escolhidos pelo usuário. Então, caberia ao lado do cliente mostrar os dados corretamente, as suas posições geoespaciais, os gráficos, as anomalias, os volumes de tráfego de rede e as estatísticas necessárias.

4.1.1 Pré-processamento

Antes da aplicação ser implementada, foi necessário executar um pré-processamento dos dados obtidos, derivados dos resultados adquiridos no trabalho dos autores de [7]. Pelo fato da aplicação ter a necessidade de requisitar os dados de acordo com o dia escolhido pelo usuário, foi necessário primeiramente ordenar os dados.

A ordenação foi feita por meio de um *script* que procurava nos dados, em formato *csv*, por dias a partir de 1 de novembro de 2013 à 22 de dezembro do mesmo ano. No entanto, originalmente o conjunto de dados possuíam atividades de tráfego até o dia 1 de janeiro de 2014, porém esses dados não estavam corretos ou não tabulados de forma satisfatória e assim foram retirados do conjunto.

Ademais, após os dados serem ordenados por dia, a segunda prioridade de ordenação era com o momento em que ocorria a anomalia no tráfego de rede. Assim, estas atividades foram divididas em faixas de tempo de 10 minutos originalmente, e continuaram assim após o processamento. Por fim, acrescenta-se ainda uma terceira prioridade de ordenação, que era com o identificador do quadrilátero, ou célula.

4.1.2 Ferramentas

Como qualquer aplicação, a escolha das tecnologias e ferramentas ditam certos aspectos que podem tornar o desenvolvimento, a escalabilidade e a manutenção mais simples. Da mesma forma, inclui-se em conta a facilidade de implementar melhorias futuras para a aplicação. As ferramentas e tecnologias principais usadas no trabalho foram:

- Javascript
- Python
- HTML/CSS
- Bootstrap
- OpenStreetMap/Mapbox
- Leaflet
- Chart.js

Diante disso, pelo fato da aplicação ser web, o uso da linguagem Javascript era prudente, pois é uma das linguagens mais usadas atualmente para desenvolvimento de aplicações web. De fato, de acordo com pesquisas, o Javascript é a linguagem mais usada no ano de 2022, seguida de CSS/HTML e por fim Python [17]. Como resultado desta popularidade, a comunidade de Javascript possui vários frameworks e bibliotecas maduras para o desenvolvimento de aplicações web. Estes frameworks já possuem várias funcionalidades básicas e acessíveis, que assim evitam do desenvolvedor refazer todo um ecossistema para uma única aplicação.

De maneira idêntica, por ser uma das linguagens mais usadas, a linguagem Python foi utilizada no trabalho. Entretanto, de forma específica para o processamento dos dados, e não só isso, como também foi usada para transformar os dados *csv* em dados *geojson*. O motivo disso está na biblioteca Leaflet que interage com o mapa. Esta biblioteca necessita do formato *geojson* para realizar corretamente as cores, os formatos e as coordenadas no mapa. Como o Javascript, a popularidade de Python produz um ambiente propício para a criação e manutenção de várias bibliotecas úteis. Assim, estas bibliotecas atacam diversos problemas diferentes. Utilizando-se dessas ferramentas, o problema encontrado de ordenar e formatar os dados pôde ser resolvido de forma eficaz no projeto.

Salienta-se ainda que o uso de HTML e CSS para a estilização da página também foi necessária. Como o Javascript e Python, estas tecnologias são umas das mais usadas, e com isso evoluíram bastante desde suas criações. No início as páginas da web eram estáticas por meio do HTML, porém com o passar do tempo o CSS foi adotado para trabalhar em conjunto com o HTML. Em vista disso, foi possível que sítios fossem mais atraentes e estilizados. Em seguida o Javascript também foi incorporado, criando assim páginas interativas e dinâmicas para os usuários pelo lado cliente.

Com esta base do lado cliente definida, é decidido que a tarefa do lado servidor será de apenas de fornecer os dados pré-processados, os *scripts* e os módulos essenciais para a aplicação. Isso porque caso o servidor fosse responsável em calcular e gerar os gráficos necessários, acarretaria em um grande uso de recursos computacionais para cada nova instância criada por novos clientes.

Para ajudar na estilização da página, foi então escolhida a biblioteca Bootstrap, pois ela possui várias ferramentas que auxiliam na criação e personalização das páginas. O Bo-

otstap é baseado em tabelas ou matrizes para a ordenação da página, além de ser fácil de usar, mas ele também possui componentes que podem ser separados ou unidos numa mesma linha ou coluna. O fato desta biblioteca ser bem conhecida e testada ajudou na escolha dela para criar a aplicação.

A funcionalidade de apresentar coordenadas, informações e imagens geográficas são feitas pelos projetos OpenStreetMap (OPM) [18] e MapBox (MB) [19]. O OPM é um projeto colaborativo que tem como objetivo criar um acervo editável de dados geográficos. Estes dados são adquiridos por meio dos usuários e seus aparelhos celulares, ou qualquer outro dispositivo capaz de obter informações sobre o local desejado. O projeto feito pelo OPM foi de extrema importância para a realização deste trabalho, pois é usado tanto o OPM quanto o MB. Apesar do MB ser uma empresa privada, ela também ajuda no desenvolvimento colaborativo do OPM, com isso foi escolhida ela como fonte principal de imagens geográficas.

Outros projetos foram levados em consideração para apresentar as informações geográficas necessárias para o usuário, como o Google Maps. No entanto, o projeto da Google possui tecnologia proprietária, e que sem aviso ou sem solicitação da opinião da comunidade, poderiam mudar as suas APIs. Possivelmente o MB também pode mudar sua API, porém a aplicação também possui formas de usar o OPM padrão caso o MB não forneça mais as imagens.

Outra biblioteca JavaScript usada foi a Leaflet. Ela é capaz de criar mapas interativos, tanto para plataformas do tipo desktop, quanto para plataformas móveis. Além disso, esta biblioteca possui plugins que estendem as suas funcionalidades.

Por fim, a biblioteca Chart.js foi responsável por criar os gráficos necessários para a visualização dos dados. O uso dela, e seus plugins, foi indispensável para a interatividade dos gráficos, além de também fornecer vários tipos de gráficos com interfaces dinâmicas.

4.2 Funcionalidades

4.2.1 Projetar o Mapa e o Tráfego de Rede

A aplicação deve primeiramente exibir, com o uso do MB, a cidade de Milão na Itália e seus arredores. Para evitar que o usuário se perca, a aplicação tranca o usuário para somente explorar os locais que possuem dados pertinentes. Por agora a aplicação somente exibe os arredores da cidade de Milão, mas poderá ser estendida para outras localidades. Depois, a aplicação traça as áreas de tráfego representadas por quadriláteros de área de $0.055\,\mathrm{km}^2$. Estes quadriláteros também são preenchidos com cores que representam o volume de tráfego das áreas. A cor cinza representa uma área sem volume de tráfego suficiente, e de forma contrária o vermelho representa uma área com um grande volume de tráfego. Uma ampliação padrão é definida para o usuário já dispor de boa parte da área visível a ele como na Figura 4.1 abaixo.

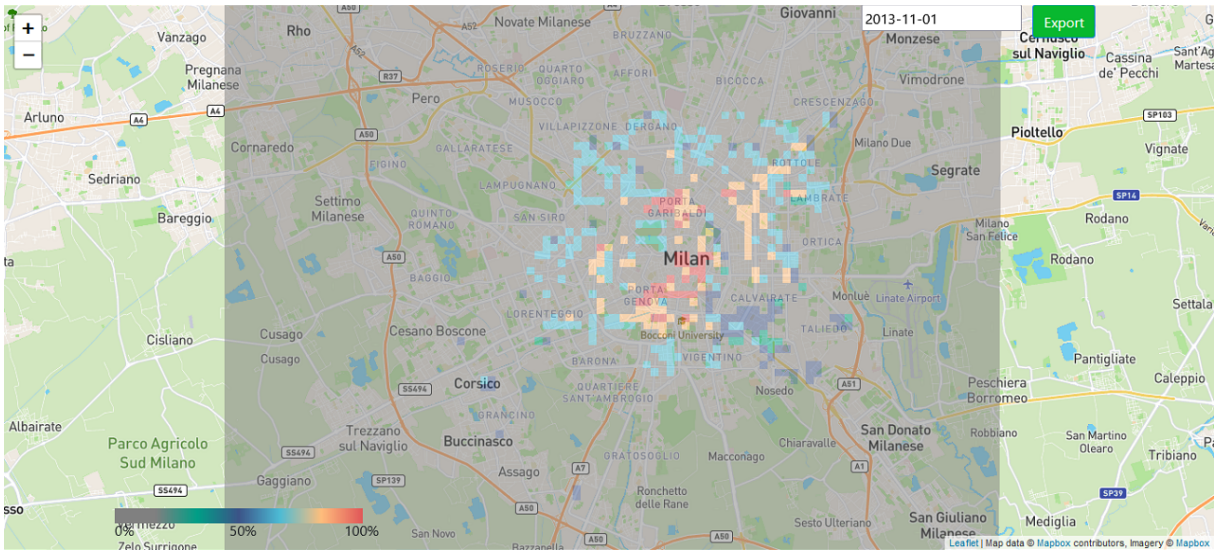

Figura 4.1: Projetando o mapa inicialmente.

Outras ampliações são possíveis com o uso do botão presente no lado superior esquerdo. No entanto, há um limite de quão longe o usuário pode observar e transladar o mapa para evitar do usuário se perder, sendo exemplificado na Figura 4.2. Mesmo que seja sem intenção, o usuário poderia deslocar-se para qualquer lugar do globo terrestre. Isso não é um comportamento interessante, pois não existem por agora outros dados obtidos

e tratados, além dos dados de Milão na Itália. Assim, isso só criaria confusão para o usuário da ferramenta.

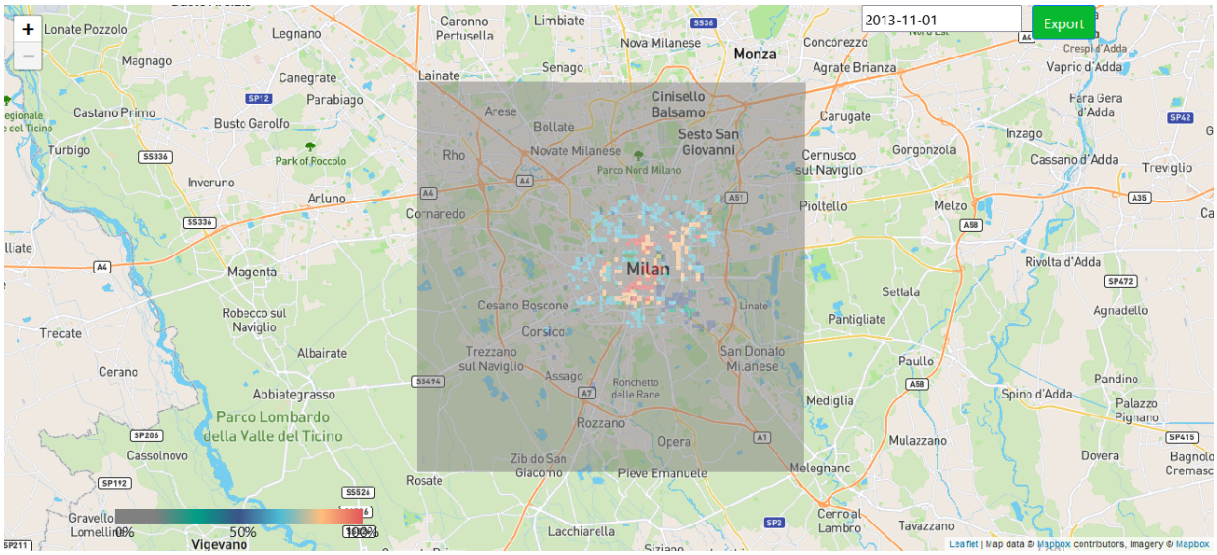

Figura 4.2: Projetando o mapa o mais ampliado possível.

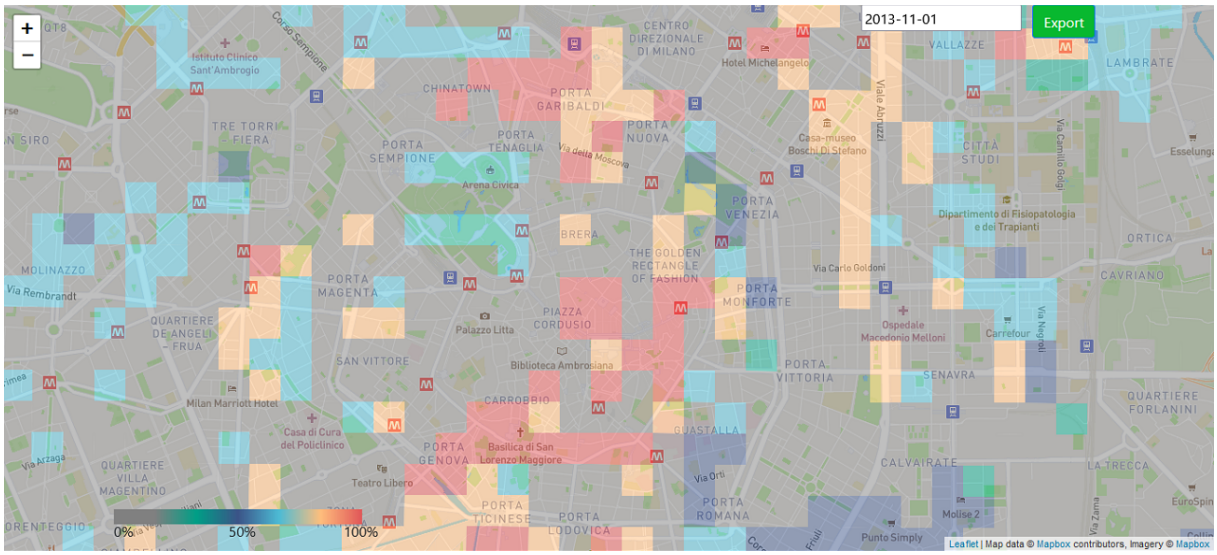

Figura 4.3: Projetando o mapa de forma mais aproximada.

Este comportamento de aproximação e distanciamento é muito interessante para o usuário. Isso porque fica mais evidente para o usuário os locais urbanos e rurais, e assim, cria-se

uma percepção maior das áreas apresentadas. Caso o usuário ache necessário se aproximar, ele também é capaz de executar essa ação como pode-se observar na Figura 4.3.

4.2.2 Interação com as Áreas

O usuário, ao pressionar uma das áreas, deverá receber informações sobre ela, como a quantidade de anomalias, suas coordenadas, seu número identificador e um botão que gera os gráficos sobre as anomalias. Caso não existam anomalias, não há como gerar gráficos. A Figura 4.4 apresenta um caso sem anomalias, e a Figura 4.5 exemplifica um caso com baixo volume de tráfego com muitas anomalias.

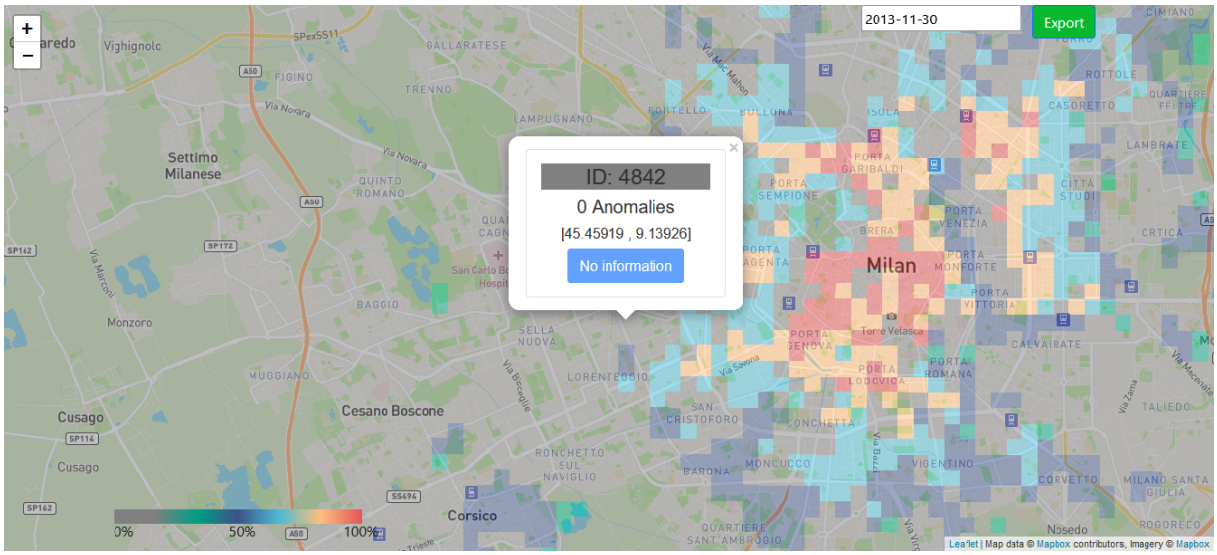

Figura 4.4: Volume de tráfego sem anomalias.

É possível observar na Figura 4.4 que, o botão *Charts* está acinzentado com o texto *No information*. Este comportamento é para o caso em que pode-se ter volume de tráfego, mas não há anomalias o suficiente para criar um gráfico, logo, os dados de volume de tráfego nessa áreas são desconsiderados. No entanto, no caso com um volume abaixo da média, representado pela cor azul-esverdeado Figura 4.5, é exibido o botão *Charts* normalmente com o número de anomalias mais acima.

Algo a salientar é a representação dos volumes de tráfego. Pode-se observar que na Figura 4.7 a cor da área do identificador é vermelha, ou seja, esta área possui um alto volume de tráfego. Para ajudar o usuário a compreender os volumes de tráfego, tem-se no

canto inferior esquerdo um gradiente de cores que mostra a saturação do volume. Sendo que cinza é um volume de tráfego desconsiderado, pois não possui anomalias, e a cor vermelha representa um alto volume de tráfego com anomalias. É interessante constatar que, o fato do volume estar abaixo da média, não significa que a área, tenha uma baixa quantidade de anomalias ou vice-versa como na Figura 4.6 e na Figura 4.7.

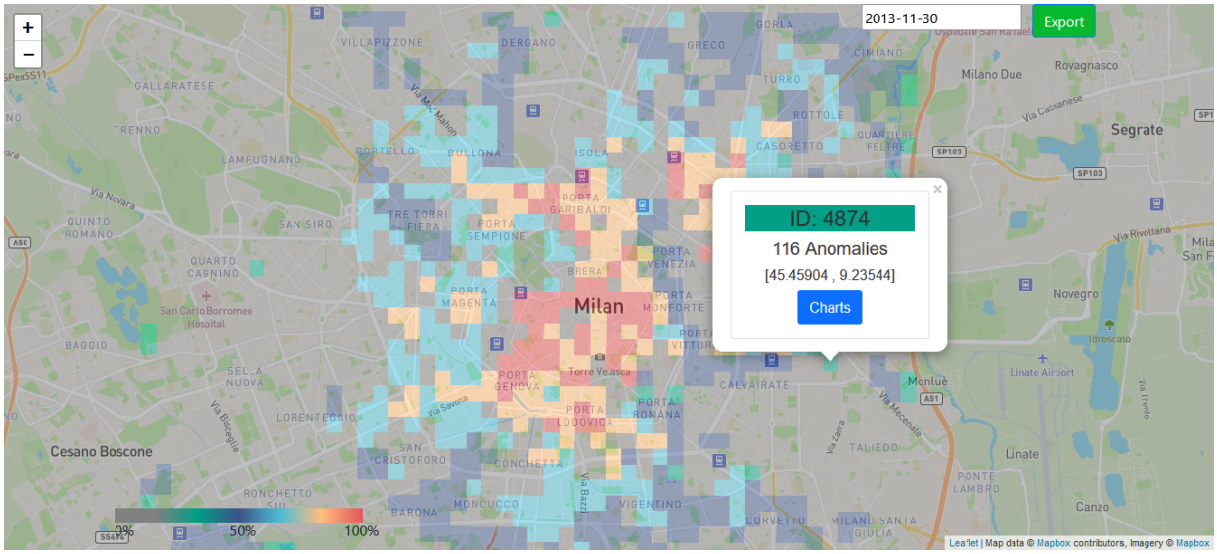

Figura 4.5: Volume de tráfego abaixo da média.

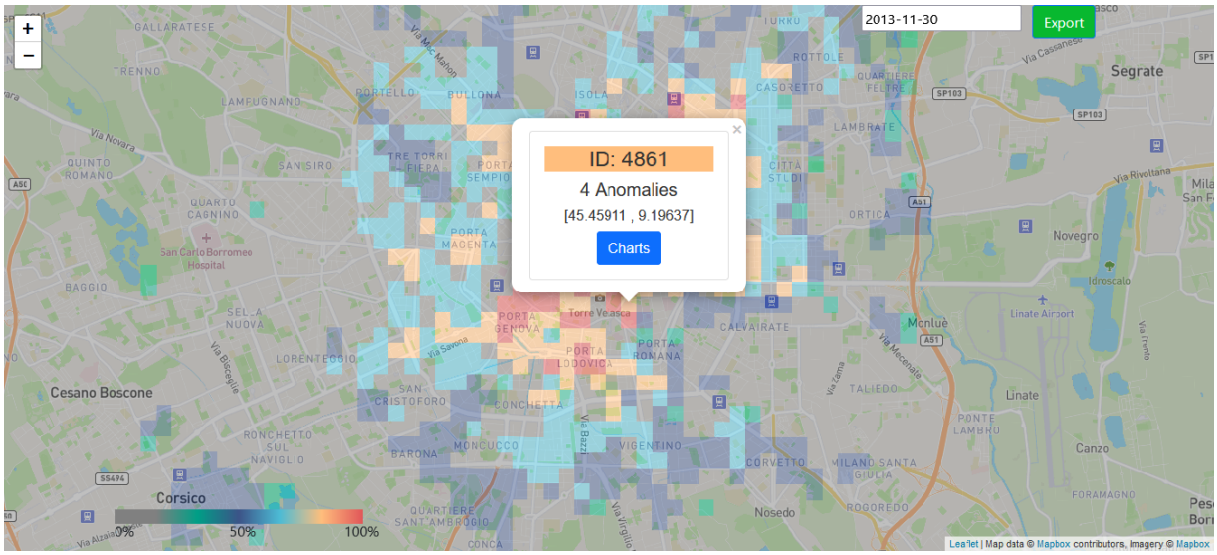

Figura 4.6: Volume de tráfego acima da média.

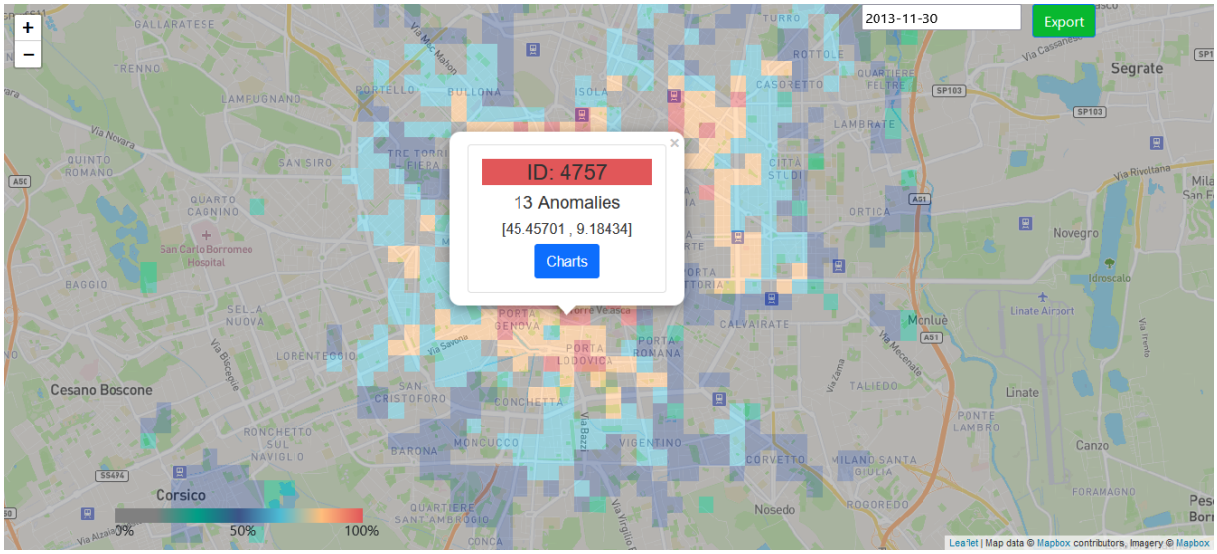

Figura 4.7: Volume de tráfego muito alto.

4.2.3 Geração dos Gráficos

A aplicação gera 3 gráficos, sendo dois do tipo barra e um gráfico do tipo dispersão. Acrescenta-se que nos gráficos do tipo barra é exibido no *eixo x* quatro momentos do dia, sendo estes momentos:

- Madrugada(00:00 - 05:59);
- Manhã(06:00 - 11:59);
- Tarde(12:00 - 17:59);
- Noite(18:00 - 23:59);

Estes mesmo momentos são divididos no gráfico tipo dispersão, porém em segundos. Pode-se perceber as divisões tanto na Figura 4.8 e na Figura 4.9. Sendo assim, um dos gráficos do tipo barra exibe a quantidade de anomalias encontradas em cada momento do dia, e o outro gráfico adicional, também do tipo barra, apresenta as médias de energia EFC durante os mesmos momentos do dia. Com efeito, o gráfico do tipo dispersão compara, para cada anomalia, a quantidade de energia EFC x Momento do dia em segundos.

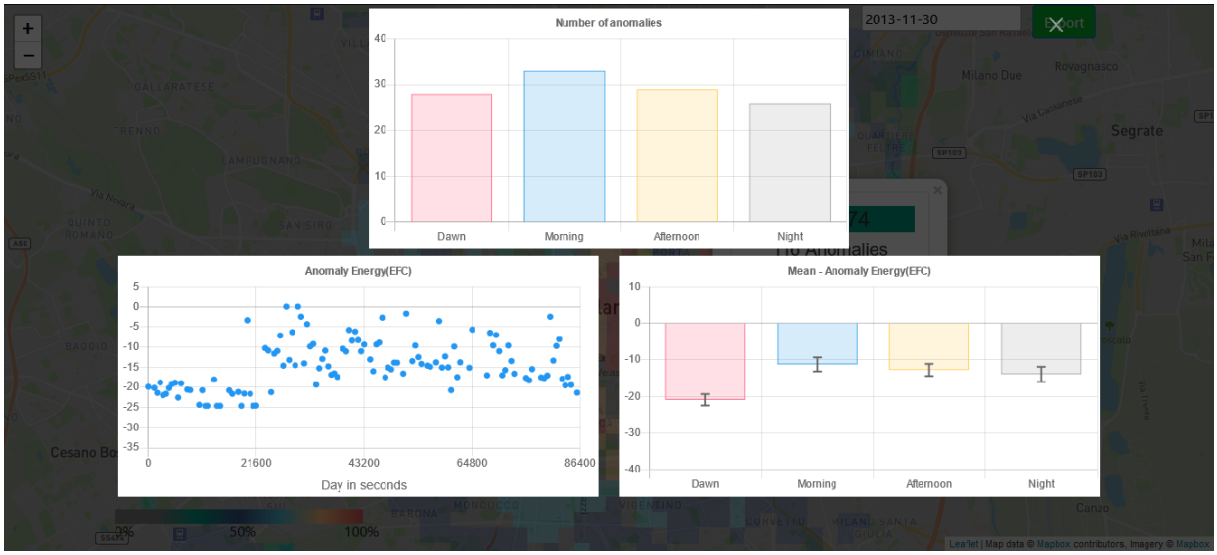

Figura 4.8: Gráficos gerados da Figura 4.5.

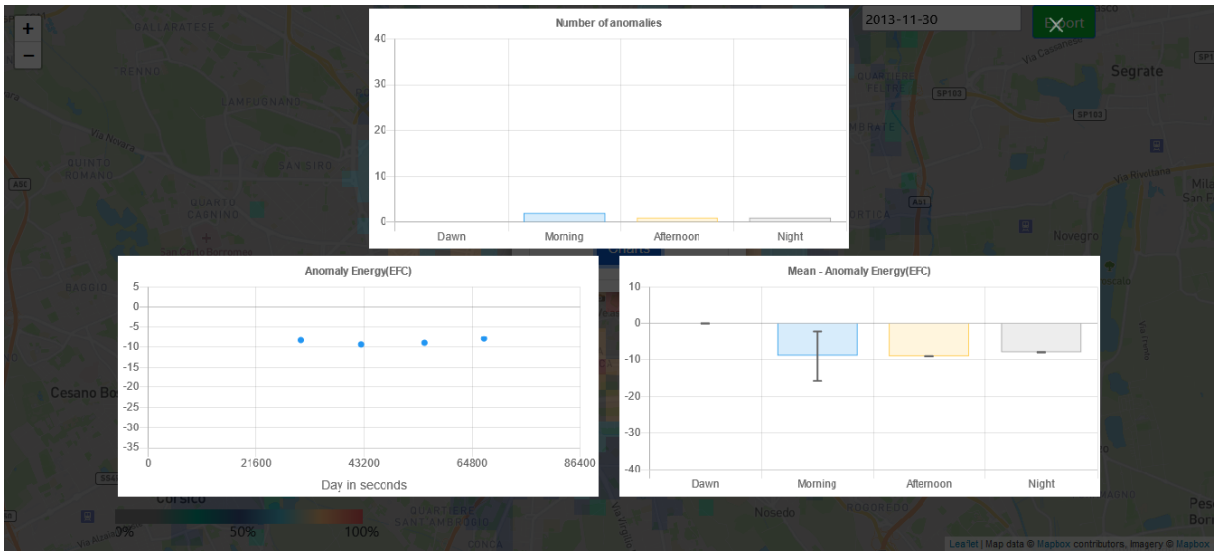

Figura 4.9: Gráficos gerados da Figura 4.6.

4.2.4 Escolha dos Dias e dos Meses

Os dados usados pelo cliente estão divididos por dia, pois assim é mais evidente a diferença de volume de tráfego de um dia para o outro. Salienta-se ainda que, a aplicação permite que o usuário escolha qualquer dia entre o 1 de novembro e 22 de dezembro do ano de 2013, como na Figura 4.10 e na Figura 4.11 de dias diferentes.

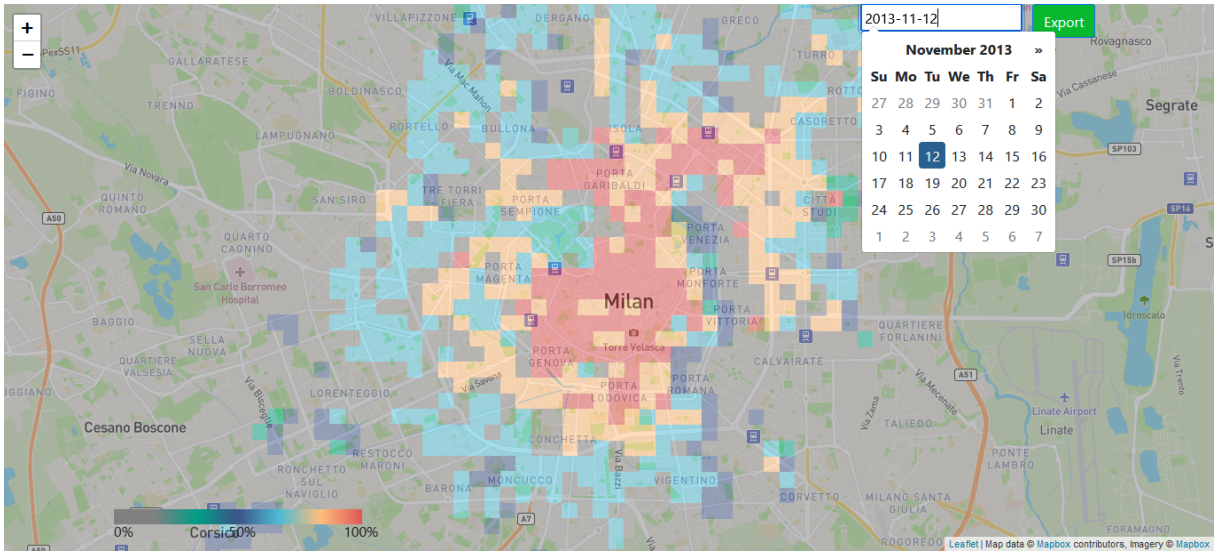

Figura 4.10: Volume de tráfego no dia 12 de novembro.

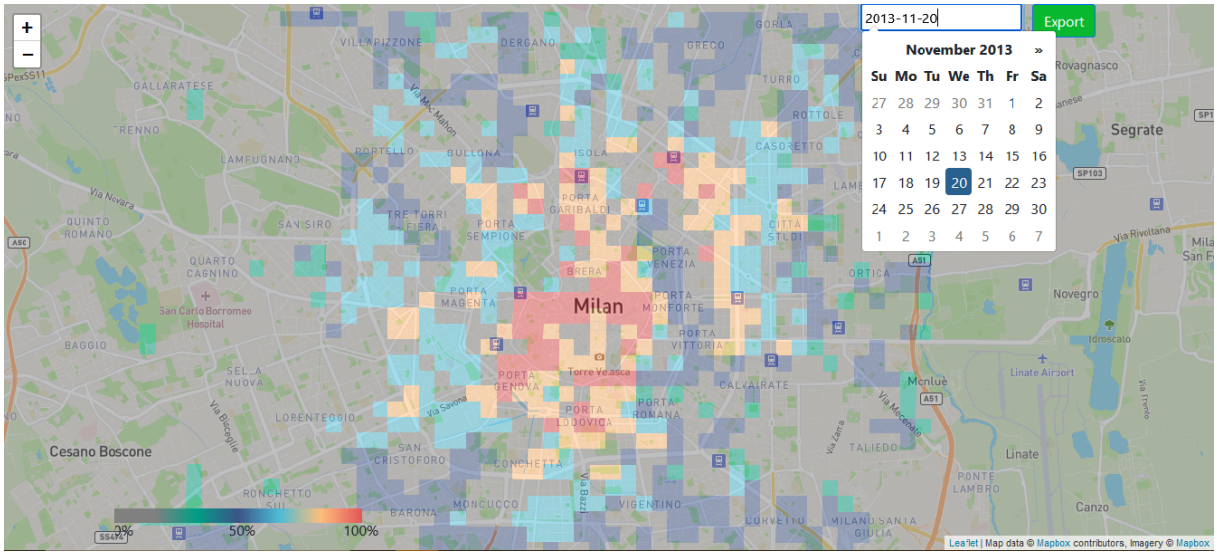

Figura 4.11: Volume de tráfego no dia 20 de novembro.

A escolha de dias diferentes, no mesmo mês, é feita pela interação do usuário com os dias do calendário. No entanto, para o usuário avançar ou retroceder um mês a ação é feita por meio dos símbolos «/» Figura 4.12 . Estes símbolos se encontram em cada canto superior do calendário. Ademais, o calendário também comunica que os dias com fonte mais escura são dias que possuem dados a serem exibidos. Porém, nem todas as fontes mais claras significam que não há dados a serem exibidos. Em alguns casos, estes dias de fonte clara apontam para os dias do próximo mês, ou do anterior Figura 4.13.

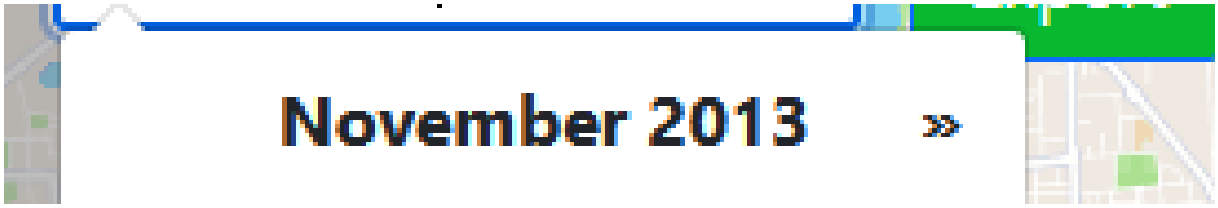

Figura 4.12: Como avançar o mês.

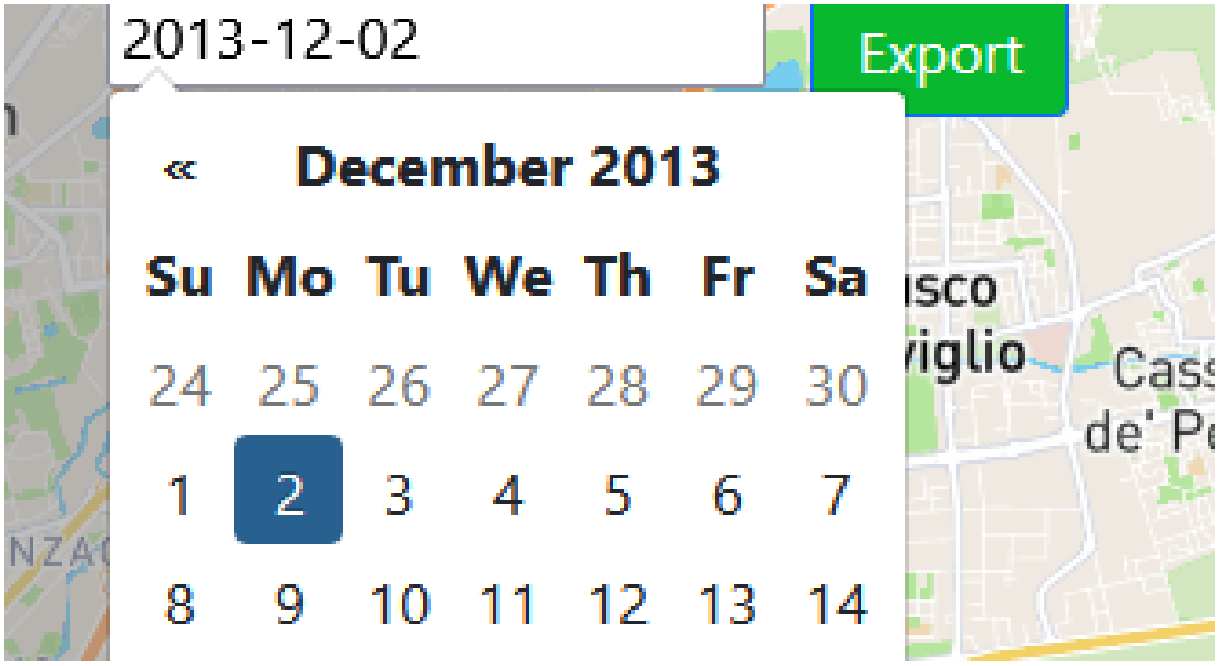

Figura 4.13: Como voltar o mês e exemplo de fontes claras.

4.2.5 Exportar os Dados

A próxima funcionalidade é feita por meio do botão *Export* do lado direito do calendário, exemplificado na Figura 4.13. De forma simples o usuário é capaz de fazer a descarga dos arquivos *csv* e *geojson* criados pela aplicação. A interface responsável possui cinco opções, como ilustrado também pela Figura 4.14:

- "Dia escolhido no calendário".geojson;
- "Dia escolhido no calendário".csv;
- geojsons-by-day.7z;
- csv-by-days.7z;

- All csv Data from Milano;

Os dois primeiros itens exportam somente os dados de um único dia. Os outros dois próximos itens exportam um arquivo comprimido que possui todos os dados divididos por dia. Já o último é um único arquivo *csv* com todos os dados.

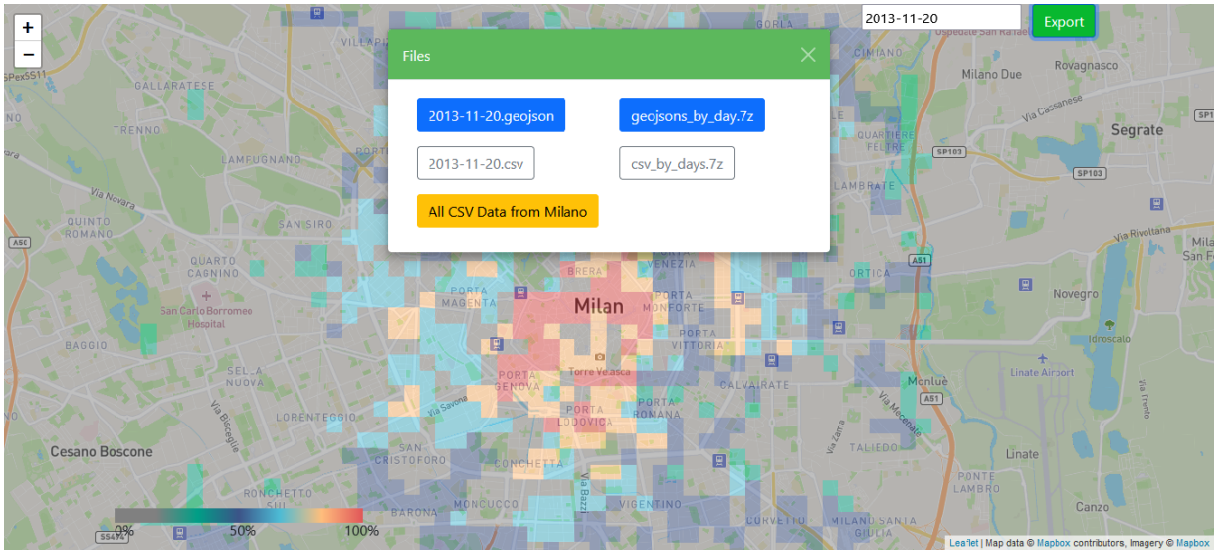

Figura 4.14: Interface para descarrega dos dados.

4.3 Protótipo

4.3.1 Instalação e Execução

O código fonte está disponível no seguinte repositório do Github. Para a instalação rápida na máquina, o usuário deve ter o Github CLI, o gerenciador de pacotes NPM e um navegador de rede atual. É altamente recomendado o uso de algum servidor HTTP no ambiente de desenvolvimento do usuário. Os dados *csv* e *geojson* se encontram aqui.

```
gh repo clone ggpsgeorge/Milan-Cluster && npm install
```

4.3.2 Implementação

Os elementos responsáveis pela interface da aplicação podem ser divididos em componentes. Esta forma de dividir os elementos é incentivada, pois evita reescrever código ao usar

as mesmas propriedades de estilo com padrão definido. Além disso, as próprias classes do Bootstrap e de outras ferramentas seguem uma linha parecida. É possível observar no Código 4.1 a que as classes *container-fluid*, *row* e *col* criam um tipo de tabela. Assim, esta tabela posiciona os gráficos gerados na ordem desejada como é apresentado na Figura 4.8.

Código 4.1: Elementos responsáveis por exibir os gráficos gerados.

```
<div id="overlay" class="container-fluid">
        <div class="row p-1">
            <div class="col"></div>
            <div class="col"></div>
            <div class="col-5">
                <div class="graph-container">
                    <canvas
                    id="number-anomalies-chart"></canvas>
                </div>
            </div>
            <div class="col"></div>
            <div class="col">
                <div id="close-container"
                class="position-relative">
                    <button id="close-button" type="button"

                    class="btn-close btn-close-white"

                    aria-label="Close"></button>
                </div>
            </div>
        </div>
        <div class="row p-1">
```

```
            <div class="col"></div>
            <div class="col-5">
                <div class="graph-container">
                    <canvas id="energy-time-scatter-graph"></canvas>
                </div>
            </div>
            <div class="col-5">
                <div class="graph-container">
                    <canvas id="energy-mean-graph"></canvas>
                </div>
            </div>
            <div class="col"></div>
        </div>
    </div>
```

Entretanto, os *scripts* Python não são usados diretamente pela aplicação. Como dito anteriormente, Python foi usado para ordenar e formatar os dados corretamente. Os dados *csv* foram ordenados com o uso da biblioteca Pandas. Pandas é uma biblioteca muito utilizada para a análise e manipulação de dados. Sendo assim, foi necessário apenas usar alguns métodos para obter os dados ordenados corretamente. No entanto, para transformar o *csv* em *geojson* foi criado um *script* mais robusto que também insere no *geojson* dados importantes, pois as ferramentas de interação com o mapa necessitam de propriedades específicas. Parte do *script* que contêm uma das funções mais importantes, que é a de gerar o *geojson* correto, é apresentado em Código 4.2.

Paralelamente, o código Javascript 4.3, que é responsável por toda a interface, geração dos gráficos e cálculos, começa definindo e atribuindo valores para cada variável. Então, é definido onde o mapa deve iniciar, ou seja, o globo terrestre centra em Milão na Itália. Outras variáveis de lista vazia que serão populadas na próxima etapa também são inicializadas. Primeiramente, é carregado a página, o calendário, o modal que é o botão do lado do calendário(Datepicker) e, finalmente, é carregado o *geojson* na página.

Código 4.2: Inserir dados csv no geojson.

```
def merge_csv_to_geojson(geojsonFilename, csvFilename,
destFilename):
    csv_dict = transform_csv_to_dict_records(csvFilename)

    with open(geojsonFilename, "r") as f:
        dataGEOJSON = json.load(f)
        f.close()

    geojson = {
        "type" : "FeatureCollection",
        "features" : []
    }

    i = 0
    for feature in dataGEOJSON['features']:
        default_properties = {
            "stroke": "white",
            "stroke-width": 0,
            "stroke-opacity": 0,
            "fill": "#808080",
            "fill-opacity": 0.5,
        }

        key = feature['id']
        geojson['features'].append(feature)
        geojson['features'][i]['properties'] =
        default_properties
        try:
            if(csv_dict[key]):
                color = get_cluster_color(csv_dict[key])
                geojson['features'][i]['properties']
                ['fill'] = color
                geojson['features'][i]['properties']
                ['activity'] = csv_dict[key]['activity']
        except KeyError:
            pass
        i+=1

    with open(destFilename, "w") as outfile:
        json.dump(geojson, outfile)
        outfile.close()
```

Código 4.3: Carregar página.

```
function loadPage(){

    loadMap(mymap);
    loadDatepicker(datepicker)
    loadModal(datepicker)

    if(datepicker.value == ""){datepicker.value =
    datepicker.placeholder}
    loadgeojson(datepicker.value, mymap, geojsonLayers);

}
```

Código 4.4: Cálculo da liberdade e da distribuiçao.

```
function get_degree_of_freedom(number_of_anomalies){
    let degree_of_freedom = number_of_anomalies - 1;
    if(degree_of_freedom < 0){degree_of_freedom = 0}
    return degree_of_freedom
};
function get_t_distribution(degree_of_freedom){
    if(degree_of_freedom > 0 && degree_of_freedom <= 30){
        return tDistribution[degree_of_freedom - 1];
    }else if(degree_of_freedom > 30){
        return z_value;
    }else{
        return 0;
    };
};
```

Finalmente, para o calculo da média das energias obtidas pelo EFC, foi utilizado o intervalo de confiança com a distribuição *t-student* como apontado pelo Código 4.4 e Código 4.5.

Código 4.5: Cálculo da média.

```
function calculate_tStudent(mean, deviation, process_data){
    let t_student_minus = {};
    let t_student_plus = {};
    Object.keys(process_data).forEach(function(key){
        let degree_of_freedom =
        get_degree_of_freedom(process_data[key]);
        let t_distribution =
        get_t_distribution(degree_of_freedom);
        let resp =
        deviation[key]/(Math.sqrt(process_data[key]));
        resp = resp*t_distribution;
        if(isNaN(resp)){resp = 0};
        t_student_minus[key] =
        (mean[key] - resp).toFixed(3);
        t_student_plus[key] = (mean[key] + resp).toFixed(3);
    });
    return [t_student_minus, t_student_plus];
```

Estas foram as funções e partes consideradas importantes no projeto. Existem outras bibliotecas, outras implementações de transformação que estão disponíveis *online* no Github.

4.4 Resultados

Em virtude das soluções apresentadas no trabalho e suas respectivas funcionalidades, tem-se por consequência certos comportamentos esperados de acordo com os requisitos definidos pelas funcionalidades necessárias na aplicação. Diante disso, em comparação com a conclusão dos autores [7], os resultados do projeto desenvolvido serão abordados em seguida.

4.4.1 Funcionalidades e Tempo de Resposta

A aplicação é capaz de fazer todas as funcionalidades listadas de forma satisfatória. O tempo de resposta da aplicação parece apropriado para exibir as áreas e gerar os gráficos mesmo sem nenhum tipo de otimização de código. A resposta para carregar toda a aplicação inicialmente é baixo com uma média de 20 ms. No entanto, ao requerer um novo dia a aplicação, sem precisar recarregar a página inteira novamente, o tempo de

resposta aumenta bastante, podendo ser entre 620 ms e 720 ms. Apesar desse aumento no tempo de resposta, a aplicação continua com o comportamento de uma aplicação de página única.

O aumento do tempo de resposta é causado por conta do comportamento do mapa, que deve sempre estar sendo exibido, o que torna a aplicação mais interessante e dinâmica para uso. Por esta razão, as áreas de volume de tráfego devem ser apagadas e repopuladas novamente, para assim o mapa continuar em exibição. Entretanto, a operação de apagar as áreas de volume de tráfego é custosa, embora sempre seja desejado um tempo de resposta melhor, o tempo atual é razoável.

4.4.2 Exportação e Formatação dos Arquivos

Os arquivos *csv* exportados e usados pela aplicação possuem um certo formato. Este formato é seguido por todos os arquivos separados *csv* para a leitura e geração dos *geojson*s. Os campos e alguns dos registros são apresentados pelo Arquivo 4.6.

Arquivo 4.6: Formato csv.

```
square_id,activity_date,cluster,activity_time,energy
2145,2013-11-01,4,14.0,-7.892002748896994
2145,2013-11-01,4,18.5,-9.545102632788172
2145,2013-11-01,4,18.67,-8.219112390958733
```

Com essa previsibilidade o *geojson* gerado com todas as suas áreas é preenchido com as informações de cada um. Estas informações são inseridas na chave *properties*. Esta chave só é criada para os quadriláteros que possuírem seus identificadores no arquivo *csv*, ou seja, se o campo square_id for o mesmo que a chave id no *geojson*. Um registro no *geojson*, do dia 1 de novembro de 2013, que possui a chave properties no Arquivo 4.7.

É possível perceber que certos valores são identificáveis facilmente. A chave *activity* possui os dados em uma lista dupla com os campos [activity_time, energy]. Já o campo *cluster* do *csv* é usado para decidir qual a cor a chave *fill* receberá. Em resumo, isso faz com que a aplicação já tenha os dados pré-processados para somente servir ao lado do cliente. Essa solução cria um resultado adequado para evitar o uso demasiado do servidor

para gerar um *geojson* a cada instância. É prudente já ter os dados processados para o servidor usar e para o usuário obter, pois essa geração seria demorada.

Arquivo 4.7: Formato geojson.

```
{"geometry": {"type": "Polygon", "coordinates":
[[[9.248933096034644, 45.451603935622785],
[9.251938424153582, 45.45159734657834],
[9.251929006916706, 45.44948208177911],
[9.248923791131448, 45.44948867034026],
[9.248933096034644, 45.451603935622785]]]}, "type":
"Feature", "id": 4479, "properties": {"stroke": "white",
"stroke-width": 0, "stroke-opacity": 0, "fill": "#3C5488",
"fill-opacity": 0.5, "activity": [[0.17,
-10.58471187375047], [0.33, -12.229258028700064], [0.5,
-12.704325330422964], [0.67, -13.77624234307391], [0.83,
-15.094357342993636], [1.0, -11.993481452386108], [1.17,
-8.725567246123683], [1.33, -12.404811813188427], [1.5,
-8.022421806015169], [1.67, -8.447871436143032], [1.83,
-8.447871436143027], ...
```

4.4.3 Observações Relevantes sobre as Anomalias

Vale ressaltar que, apesar de certas áreas possuírem um grande volume de tráfego, isto não significa que há a presença de muitas anomalias na rede como pode ser observado na Figura 4.5. Em [7], é apresentado um mapa de calor na Figura 4.15 que exemplifica alguns desses casos e, assim, permite-se um panorama geral dos dados.

O resultado é que existem casos de alto número de anomalias em áreas de pouco volume de tráfego, como as áreas em vermelho no mapa de calor. Intuitivamente, lugares com maior volume de tráfego teriam maior número anomalias, mas nem sempre é o caso. Além Figura 4.4, a Figura 4.5 e a Figura 4.6, também mostram que geralmente o lado inferior direito do centro da cidade possui pouco volume de tráfego, logo, pode-se ter um grande número de anomalias.

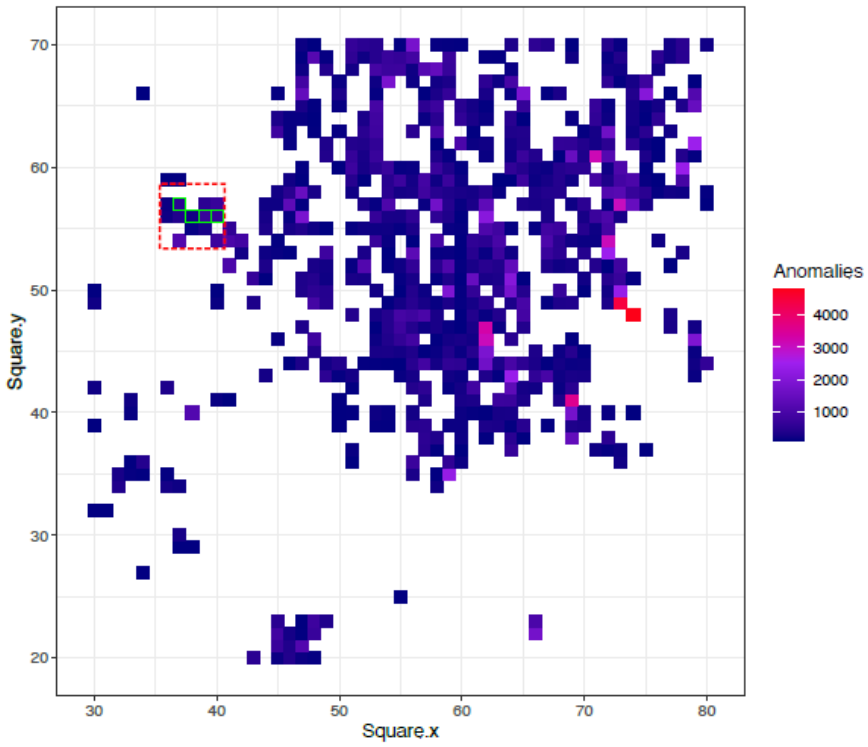

Figura 4.15: Agrupamento de todos os dias.

4.4.4 Gráficos

O resultado é satisfatório na geração e apresentação dos gráficos. No entanto, para casos que existem poucas anomalias os gráfico ficam muito vazios e com erros estimados muito grandes como na Figura 4.9. Ademais, o tempo para gerar os gráficos é muito pequeno, menor que 1ms, provavelmente porque a quantidade de anomalias não é grande, isto é, não passam da casa dos milhões.

Capítulo 5

Conclusões

O projeto implementado alcançou os objetivos propostos no manuscrito, sendo sua maior contribuição a capacidade de exibir dados geoespaciais de CDRs e dados de anomalias. Estes dados, ou resultados, apresentavam suas respectivas áreas com informações sobre a quantidade de anomalias e seu volume de tráfego corretamente sobre o mapa geoespacial. Com efeito, a lacuna que a ferramenta visa preencher, é importante para a área de pesquisa sobre anomalias em redes móveis, pois faltava uma ferramenta capaz de auxiliar na visualização dos resultados obtidos. Paralelamente, outra contribuição do projeto, é referente à geração dos gráficos e exportação de dados sobre as anomalias, que também era outro objetivo deste trabalho. Por consequência, as funcionalidades de gerar gráficos e exportar dados, ajudam a comunidade a explorar os resultados obtidos por outras pesquisas. De forma geral, o trabalho foi capaz de realizar todos os objetivos especificados. No entanto, certamente ainda há espaço para melhorias.

Como mencionado anteriormente, ainda há espaço para melhorias no trabalho apresentado. Assim, além das melhorias referentes à interface, refinamento e otimização de código, é interessante estender o escopo da aplicação para outros lugares do globo terrestre. Entretanto, não somente o escopo da ferramenta, mas também as funcionalidades podem ser estendidas. Como por exemplo, uma funcionalidade de comparação de duas áreas simultaneamente. Estas áreas, que podem ser de dias e meses diferentes, exibem ao usuário as diferenças de cada área. De certo, com as melhorias e as novas funcionalidades, uma aplicação notavelmente diferenciada será elaborada.

Referências

[1] Hussain, Bilal *et al.*: *Mobile edge computing-based data-driven deep learning framework for anomaly detection.* IEEE Access, 7:137656–137667, 2019. 1, 12

[2] De Almeida, Jonathan M *et al.*: *Optimal Allocation of vBBUs Considering Distance Between MDC and RRH in F-RANs.* IEEE International Conference on Communications (ICC), 2020. 1

[3] 3rd Generation Partnership Project (3GPP): *3GPP TS 29.520 version 16.4.0 Release 16 - 5g; 5g system; network data analytics services; stage 3.* Relatório Técnico 4, August 2020. 2

[4] Turner, Daniel *et al.*: *California Fault Lines: Understanding the Causes and Impact of Network Failures.* Em *Proceedings of the ACM SIGCOMM Conference*, página 315–326. Association for Computing Machinery, 2010. 2

[5] Zheng, Kedi *et al.*: *A Novel Combined Data-Driven Approach for Electricity Theft Detection.* IEEE Transactions on Industrial Informatics, 15(3):1809–1819, 2019. 2

[6] Zhu, Qiqi e Li Sun: *Big Data Driven Anomaly Detection for Cellular Networks.* IEEE Access, 8:31398–31408, 2020. 2, 12

[7] De Almeida, Jonathan M., Camila F. T. Pontes, Luiz A. Da Silva, Cristiano B. Both, Joao J. C. Gondim, Celia G. Ralha e Marcelo A. Marotta: *Abnormal behavior detection based on traffic pattern categorization in mobile cellular networks.* IEEE Transactions on Network and Service Management, páginas 1–1, 2021, ISSN 1932-4537. 3, 4, 6, 9, 10, 13, 14, 15, 31, 33

[8] Wu, F. Y.: *The Potts model.* Reviews of Modern Physics, 54(1):235–268, jan 1982, ISSN 0034-6861. 6

[9] Morcos, Faruck *et al.*: *Direct-coupling analysis of residue coevolution captures native contacts across many protein families.* Proceedings of the National Academy of Sciences of the United States of America, 108(49):E1293–E1301, dec 2011. 7, 8

[10] Giraud, BG, John M Heumann e Alan S Lapedes: *Superadditive correlation.* Physical Review E, 59(5):4983, 1999. 8

[11] Jaffry, Shan, Syed Tariq Shah e Syed Faraz Hasan: *Data-driven Semi-supervised Anomaly Detection using Real-World Call Data Record.* Em *IEEE Wireless Communications Networks Conference*, páginas 3–8, 2020. 9, 12

[12] Trinh, Hoang Duy *et al.*: *Detecting Mobile Traffic Anomalies through Physical Control Channel Fingerprinting: A Deep Semi-Supervised Approach*. IEEE Access, 7:152187–152201, 2019. 11

[13] Hussain, B., Q. Du e P. Ren: *Deep Learning-Based Big Data-Assisted Anomaly Detection in Cellular Networks*. Proceedings of IEEE Global Communications Conference, páginas 1–6, 2018. 12

[14] Parwez, Md Salik, Danda B. Rawat e Moses Garuba: *Big data analytics for user-activity analysis and user-anomaly detection in mobile wireless network*. IEEE Transactions on Industrial Informatics, 13(4):2058–2065, 2017. 12

[15] Sultan, Kashif, Hazrat Ali e Zhongshan Zhang: *Call Detail Records Driven Anomaly Detection and Traffic Prediction in Mobile Cellular Networks*. IEEE Access, 6:41728–41737, 2018. 12

[16] Papadopoulos, Stavros, Anastasios Drosou e Dimitrios Tzovaras: *A Novel Graph-Based Descriptor for the Detection of Billing-Related Anomalies in Cellular Mobile Networks*. IEEE Transactions on Mobile Computing, 15(11):2655–2668, 2016. 12

[17] Stackoverflow contributors: *Survey of the most used technologies*. `https://survey.stackoverflow.co/2022/`, 2022. 16

[18] OpenStreetMap contributors: *Planet dump retrieved from https://planet.osm.org* . `https://www.openstreetmap.org`, 2017. 17

[19] Mapbox contributors: *Planet dump retrived used from https://api.mapbox.com*. `https://www.mapbox.com/`, 2021. 17