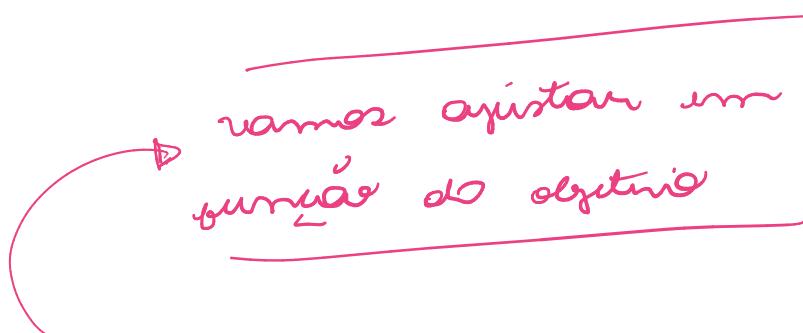

Pontifícia Universidade Católica de Minas Gerais
Departamento de Ciência da Computação



Análise e Comparação Comportamental de Traces de Redes Veiculares(VANETs)

Túlio Nunes Polido Lopes

Orientador: Prof. Felipe Domingos da Cunha

Belo Horizonte, MG, agosto de 2021

Conteúdo

1 Introdução	2
2 Trabalhos Relacionados	2
3 Metodologia	4
3.1 Etapas do trabalho	4
3.2 Base de dados	5
3.3 Ferramentas	5
3.4 Etapas do processo de estudo da base	5
3.4.1 Limpeza da base	6
3.4.2 Algoritmo para secção do trace de Roma	6
3.4.3 Cálculo dos limites utilizados no algoritmo de secção de traces	11
3.4.4 Matriz Origem/Destino(OD)	13
3.5 Métricas avaliadas	13
4 Resultados Parciais	14
4.1 Análise inicial	14
4.2 Avaliação das métricas	16
4.2.1 Viagens por veículo	16
4.2.2 Comprimento das trajetórias	17
4.2.3 Duração das trajetórias	18
4.2.4 Velocidade média por trajetória	19
4.2.5 Pontos de concentração de origem/destino das viagens	21
5 Conclusão	24
6 Trabalhos futuros	25

1 Introdução

→ você fala muito pouco sobre o problema. É importante relacionar os parágrafos.

Desde a popularização de veículos automotores no século XX, diversas cidades, em especial as grandes metrópoles, vêm sofrendo com a crescente demanda por controle de trânsito. No Brasil, a cidade de São Paulo é o ápice desse problema, e, mesmo utilizando frequentes rodízios de veículos e alterações nas vias, não há mudança significativa, como pode ser visto em Cintra (2013). Já existem trabalhos conjuntos de diversas áreas da ciência, como geografia, urbanismo e engenharias, cujo objetivo é a solução a longo prazo desse obstáculo e, com o advento de veículos autônomos, é possível inferir que poder-se-á utilizar redes veiculares complexas para automação e controle do trânsito, o que levaria a uma melhora no tempo de resposta dos veículos, gerando uma melhora na sua circulação. *muito conciso aqui.*

Para a implementação de tal sistema, seria ideal avaliar a melhor forma de desenvolvimento dos protocolos dessa rede. Sabendo dessa meta, é de grande valia estudos comportamentais da movimentação veicular. Esses possibilitariam uma visão mais assertiva de como construir um mecanismo de comando dos automóveis que permita um fluxo mais eficiente.

O objetivo deste trabalho é fazer uma análise de comportamento de redes veiculares, utilizando-se uma base referente à cidade de Roma (Itália) e outra à cidade de Dublin (Irlanda), cujos conjuntos de veículos são compostos por táxis e ônibus, respectivamente. Buscou-se desenvolver uma metodologia para exploração de trases veiculares, empregando dados de GPS para estabelecer quais padrões de comportamento podem ser identificados para cada base considerando as métricas selecionadas.

O presente trabalho se encontra dividido da seguinte forma: na seção 2, os trabalhos relacionados são descritos de acordo com sua importância e influência para o artigo desenvolvido. Em seguida, na seção 3, é descrita a metodologia utilizada para analisar as bases, onde é retratado a forma de disposição dos dados de cada base, as etapas desenvolvidas da análise, a descrição de algoritmos para secção de trases e as métricas avaliadas. Na seção 4 são descritos e analisados os resultados obtidos das análises primária e individuais de cada métrica escolhida. Na seção 5 são tiradas conclusões com base nos dados obtidos e, por fim, na seção 6 são descritos possíveis caminhos a serem tomados em trabalhos futuros.

2 Trabalhos Relacionados

Nessa seção são descritos os trabalhos que serviram de base para o desenvolvimento deste estudo, bem como sua importância para o mesmo.

Dentre os artigos utilizados para obtenção de informações de parâmetros e métricas de análises de trases estão: Basta et al. (2016), que desenvolve uma proposta de um modelo de mobilidade social para ser utilizados em simulações de redes veiculares. Ele se baseia em esferas sociais, onde cada proprietário de veículo se conecta a outros proprietários por uma rede familiar, laboral ou de amizades; Celes et al. (2019), que estuda a mobilidade de ônibus

qual o assunto
dos trabalhos que
serão desenvolvidos ??

intramunicipais e seu impacto na construção de redes veiculares em uma cidade. Os dados foram calibrados e usados para construção de uma rede baseada em grafos temporais, assim cada grafo contém a interação dos veículos naquele instante de tempo; Gainaru et al. (2009), onde é feita uma proposta de um modelo de mobilidade social, considerando a relevância de simulação microscópica, isto é, simulação de informações como perfil de agressividade do motorista, que permite testar diferentes cenários de combinações de personalidades.

O artigo Gong et al. (2015), foi utilizado como referência para o algoritmo de secção de viagens individuais de trases contínuos. Ele estabelece parâmetros e métodos para seccionar dados de GPS de smartphones que não estejam divididos em viagens. Utiliza-se a ideia de clusterização do algoritmo DBSCAN somado ao uso de um Suport Vector Machine(SVM) para detectar áreas onde o usuário se manteve por um determinado tempo, definindo os pontos de interesse durante sua trajetória. Também importante para o desenvolvimento do algoritmo de secção de trases é o artigo Li et al. (2008) que busca analisar o movimento de pessoas utilizando dados de GPS de celulares com a finalidade de definir quais os pontos de parada durante o trajeto, por exemplo, shoppings, lojas, edifícios etc. Para isso foi desenvolvido um algoritmo que busca quando uma pessoa ficou dentro de um raio específico por muito tempo, definindo um novo ponto de parada. Algoritmo, este, que serviu de base para um desenvolvimento de uma heurística de obtenção de trajetórias individuais a partir de um conjunto contínuo de dados.

Kong et al. (2018) possui informações importantes para a implementação da matriz de origem/destino desenvolvida na atual pesquisa. Nele é proposto um método de simulação de dados de veículos particulares a fim de facilitar a obtenção de *datasets* para estudos futuros. Essa simulação é feita em três passos: *Demand description*, onde é construída uma matriz de origem e destino(OD) e corrigida de acordo com a proporção de táxis para veículos particulares; *Network description*, onde ocorre a preparação do ambiente de simulação utilizando SUMO e OpenStreetMaps; e *Simulation*, que é a própria simulação.

Os seguintes artigos serviram de introdução ao tema investigado: Ning et al. (2017) dá uma breve explanação sobre o que são as *Vehicular Social Networks*(VSN) comparando-as com redes sociais e com redes ad-hoc veiculares(VANET). Também é feito um estudo de caso a fim de descobrir possíveis desafios em sua construção no mundo real; Cintra (2013) descreve dados sobre o problema do trânsito de São Paulo ainda em 2013, onde há quilômetros de engarrafamentos que leva condutores a perderem até 2 horas por dia parados. Os principais motivos apresentados são a grande quantidade de veículos e a má gestão das vias.

Com as informações obtidas nos estudo citados, o trabalho corrente busca fazer uma análise do comportamento veicular em uma rede, com o objetivo de definir e entender padrões de comportamento dos veículos em uma grande cidade. O conhecimento alcançado pode ser utilizado futuramente de forma auxiliar na construção de uma rede veicular inteligente, que precisará estabelecer o comportamento padrão dos veículos autônomos atentando antecipadamente aos movimentos comuns para o trânsito de uma metrópole.

*para fazer a análise
do trace . . . ~*

3 Metodologia

Esta seção busca descrever a metodologia utilizada na ~~pesquisa~~. Ela é dividida em 5 partes, sendo: Etapas do trabalho, onde é explicado como se deu a divisão da pesquisa; Base de dados, que descreve os dois traces utilizados; Ferramentas, que cita e explica quais ferramentas foram utilizadas para o desenvolvimento do estudo; Etapas do processo de estudo da base, que explica como foi feita a limpeza da base, o desenvolvimento do algoritmo de divisão de traces e o desenvolvimento da Matriz OD; e Métricas avaliadas, que cita e descreve as métricas escolhidas para investigação das bases de dados.

3.1 Etapas do trabalho

O trabalho proposto se divide em 5 etapas descritas na Figura 1, sendo elas: A limpeza da Base, Análise Inicial, Secção de Traces, Matriz O/D, Análise individual das métricas.

O primeiro passo é realizar uma limpeza da base, onde são excluídos os atributos não utilizados e as instâncias problemáticas, como as duplicadas ou com atributos incompletos. O objetivo dessa limpeza é acelerar o processamento dos dados e tornar improvável que os resultados estejam enviesados por instâncias problemáticas. A segunda etapa consiste de uma análise inicial onde são utilizadas métricas básicas de estatística como gráficos de densidade e mapas de calor para avaliação do comportamento do trânsito de Roma. Com essa análise é possível identificar indícios de padrões comportamentais, bem como identificar possíveis métricas a serem melhor analisadas nas etapas seguintes.

Na terceira parte, foi desenvolvido um algoritmo para secção do trace de Roma, por não estar dividido em viagens individuais. Esse algoritmo foi discutido tendo como base as referências citadas na seção II. Seu desenvolvimento é importante para uma melhor avaliação do trace de Roma por possibilitar implementar uma matriz de origem e destino (Matriz OD). Após a construção e otimização desse algoritmo, a matriz de origem e destino foi implementada, na quarta etapa. Com a matriz OD é possível fazer uma análise mais profunda dos *traces* de Roma e Dublin. Ela foi construída com a finalidade de dividir as cidades de Roma e Dublin em áreas quadradas que representassem quarteirões, possibilitando uma exploração quantitativa da base através de diversas métricas.

As métricas foram investigadas na última etapa, onde há uma avaliação estatística dos resultados obtidos e interpretação dos mesmos.



Figura 1: Fluxograma das Etapas do Trabalho .

3.2 Base de dados

Dois traces foram usados durante as análises. Um trace de Roma que contém dados de táxis e um trace de Dublin que contém dados de ônibus. Ambas as bases são descritas a seguir.

O trace de Roma está contido em um arquivo .csv. Neste arquivo, cada linha representa um ponto de GPS gravado, possuindo, necessariamente nesta ordem: Um inteiro para representar o ID do veículo; Um String contendo a data e hora em que o ponto foi gravado; Um float para a longitude; Um float para a latitude; Um booleano que define se o ponto sofreu calibração ou não. Os veículos do trace de Roma são todos táxis da região.

Quadro 1 - Exemplo dos Dados de GPS de Roma

ID	time	long-x	lat-y	is-calibrated
101	2014-02-04 05:00:01	12.48884	41.90304	1
101	2014-02-04 05:00:03	12.48875	41.90301	0

O trace de Dublin contém dados de GPS de ônibus municipais, também em .csv. Diferente do de Roma, esse não é um trace calibrado, porém possui uma maior quantidade de atributos a disposição. Dentre esses novos atributos temos at stop, que define se o ônibus está em um ponto de parada, stop ID, que identifica a parada atual do veículo, journey ID, que identifica o ID da viagem atual, e line ID, que identifica a linha do veículo.

Quadro 2 - Exemplo dos Dados de GPS de Dublin

Timestamp	line ID	journey ID	long	lat	stop ID	vehicle ID	at stop
1356998403	747	3493	-6236852	53425327	7411	40040	0
1356998405	27	3883	-6233417	53342232	395	33521	0

3.3 Ferramentas

Nesse trabalho a linguagem utilizada no processamento dos dados foi Python devido a suas bibliotecas científicas otimizadas para processamento de grande volume de dados como Pandas e NumPy. Todos os dados foram ordenados por duas chaves, a primeira é o ID do veículo, já a segunda, o horário em que o ponto de GPS foi gravado.

3.4 Etapas do processo de estudo da base

Para uma análise mais bem fundamentada das informações, o ideal é a criação de uma matriz de origem/destino(OD), onde as linhas representam as posições iniciais das viagens, enquanto as colunas, as posições finais. Com ela é possível fazer a análise mais aprofundada dos dados, onde poderemos definir as áreas com maior saída e maior entrada de veículos em determinado intervalo de tempo. Como explicado na seção 3.2, os dados das viagens da base de Roma não possuem um parâmetro que indica o início e o fim de uma corrida fazendo-se necessário a utilização de metodologias, como as descritas em [Gong et al. \(2015\)](#), para a separação desses dados brutos em cada corrida dos respectivos táxis.

3.4.1 Limpeza da base

A limpeza da base de dados é feita utilizando basicamente a biblioteca pandas, do Python, citada na seção "Ferramentas". Essa biblioteca permite a carga completa da base de dados na memória, já possuindo algumas funções para automatizar a limpeza da base. Primeiro foram excluídas todas instâncias duplicadas, para evitar que os cálculos feitos posteriormente possuíssem um enviesamento. Também foi utilizado uma função que exclui todas as instâncias com algum atributo nulo, evitando qualquer problema de cálculo durante o processamento dos dados. Esses erros nas bases de dados normalmente ocorrem durante a leitura dos dados, como quando um sensor falha e gera uma instância problemática que deve ser removida.

3.4.2 Algoritmo para secção do trace de Roma

O passo seguinte deste projeto engloba a divisão do trace de Roma que possui uma gravação contínua de dados, sem que haja separação das trajetórias individuais. Foram utilizados dois passos principais, o primeiro é a criação de uma lista de todos intervalos de tempo e distância entre dois pontos consecutivos. O segundo passo é a utilização desses valores para encontrar de forma aproximada os momentos onde provavelmente houve o fim/início de uma trajetória de táxi.

No Algoritmo 1, os dados passam por um loop onde a distância e o tempo de gravação entre dois pontos consecutivos são gravados em duas listas. O *for* externo cria uma nova lista para cada ID de veículos, contido no atributo *key*, enquanto o atributo *value* contém a lista de pontos específica daquele veículo. O ciclo interno percorre a lista de pontos do ID atual, onde grava a diferença entre os dois horários, em segundos, e a diferença entre as duas coordenadas, em metros, respectivamente nas listas *timeGaps* e *coordGaps*. O algoritmo de Haversine utilizado abaixo é um método desenvolvido para calcular a distância em metros entre duas coordenadas geográficas.

Algoritmo 1 - Geração das listas de intervalos de tempo e distância.
↓ quitar!

Algorithm 1

```
1: Entrada: Pontos ordenados por horário, onde cada ponto possui uma coordenada e  
um horário de gravação.  
2: Saída: Uma lista contendo a diferença de tempo entre cada ponto e outra lista  
contendo a diferença de distância.  
3: for key,value in Entrada do  
4:   timeGaps[key] = nova lista  
5:   coordGaps[key] = nova lista  
6:   for i=0 to value.length -1 do  
7:     point1 = value[i]  
8:     point2 = value[i+1]  
9:     pointCoord1 = point1.coordinate  
10:    pointCoord2 = point2.coordinate  
11:    distance = haversine(pointCoord1,pointCoord2)  
12:    coordGaps[key].append(distance)  
13:    pointTime1 = point1.hour  
14:    pointTime2 = point2.hour  
15:    timeGaps[key].append(pointTime2 - pointTime1)  
16:   end for  
17: end for  
18: Retorna timeGaps,coordGaps; FIM
```

O passo seguinte é a separação de viagens, que é feito por outro algoritmo. Ele recebe as duas listas de intervalos, de tempo e de distância, e retorna uma nova lista contendo os índices dos pontos onde devem ocorrer o início de novas *trips*. É importante ressaltar que esse algoritmo é uma heurística que busca aproximar os dados de uma realidade plausível, não sendo de nenhum modo testado de forma empírica. O primeiro algoritmo para detecção de viagens individuais é o Algoritmo 2, descrito abaixo. O mesmo teve como base o algoritmo descrito em [Li et al. \(2008\)](#), cuja ideia original era a detecção de pontos de parada de pedestres em regiões de interesse como praças, shoppings, etc.

(Algoritmo 2 - Criação das viagens - Li et al. (2008)) .

Do juntar com o algoritmo.

Algorithm 2

Entrada: Uma lista contendo a diferença de tempo entre cada ponto, outra lista contendo a diferença de distância e um vetor dos IDs.

2: **Saída:** Uma lista contendo os índices onde devem ocorrer as separações das viagens.

```
for key in Keys do
    4:    flag = FALSE
    tempo = 0
    6:    separator[key] = nova lista
    for i=0 to timeGaps.length -1 do
        8:        if coordGaps[key][i] / timeGaps[key][i] < 0.1 then
            tempo += timeGaps[key][i]
        10:       else
            if flag == TRUE then
                12:           separator[key].append(i)
                flag = FALSE
            14:           tempo = 0
            else
                16:               tempo = 0
                flag = FALSE
            18:           end if
            end if
        20:           if tempo > 30.0 then
            flag = TRUE
        22:           end if
        end for
    24: end for
    Retorna separator; FIM
```

O ciclo externo percorre um vetor contendo as chaves, ou IDs, dos veículos. Para cada ID presente é criada uma lista onde serão gravadas os índices de fim de viagens. Também são criadas duas variáveis de controle, tempo, que grava quantos segundos o carro ficou parado, e *flag*, que define se o tempo parado é suficiente para definir uma nova viagem.

O ciclo interno caminha entre os dois vetores de entrada simultaneamente. A velocidade de cada posição é analisada, e, caso seja menor que 0.1 metros por segundo, é considerado que naquele momento o carro estava possivelmente parado, logo soma-se à variável tempo o valor em segundos daquele intervalo. Caso a velocidade seja maior que 0.1 metros por segundo, é verificado se a *flag* é verdadeira, caso seja, aquele ponto é considerado um ponto de ruptura e é salvo na saída, enquanto as variáveis de controle são zeradas. Caso a *flag* seja falsa as variáveis de controla são zeradas e a possível parada, desconsiderada. No fim do ciclo uma condição *if* confere se o tempo acumulado em velocidade baixa é maior que 30 segundos, e, caso verdadeiro, a *flag* assume valor *TRUE*, significando que assim que houver um sinal de aceleração do veículo um ponto de ruptura deve ser criado.

Esse método de partição dos trajes se mostrou pouco eficiente para a base utilizada. Numa análise feita acompanhando os pontos de ruptura selecionados durante a trajetória dos

veículos, os resultados não foram tão satisfatórios. Isso foi constatado pelo fato das trajetórias construídas pelo algoritmo apresentarem claramente uma inconsistência, onde muitas viagens não parecem naturais partindo do princípio que uma viagem de táxi deve ir do ponto A ao ponto B da cidade na menor distância possível, o que não ocorre, de forma explícita, em diversas trajetórias. Esse problema é evidente na Figura 2, onde há um exemplo de trajetória individual de um veículo gerada pelo Algoritmo 2.

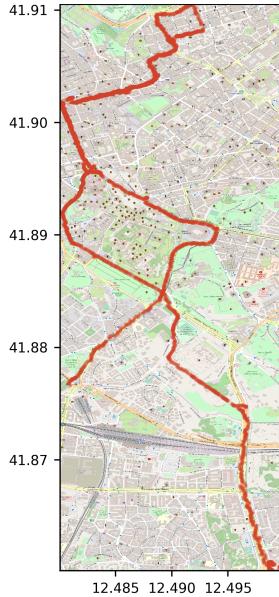


Figura 2: Exemplo de Trajetória Gerada de um Veículo

É perceptível que essa trajetória selecionada pelo algoritmo não é viável, tendo em vista que o veículo dá uma volta claramente muito maior que necessária pra uma viagem. Assim, é provável que o algoritmo não tenha reconhecido outros pontos começo e fim de viagem nesse trecho. Somado a esse problema, o algoritmo também gerou muitas fragmentações nas viagens, como pode ser observado no Figura 3, onde cada troca de cores representa o fim de uma *trip* e início da próxima.

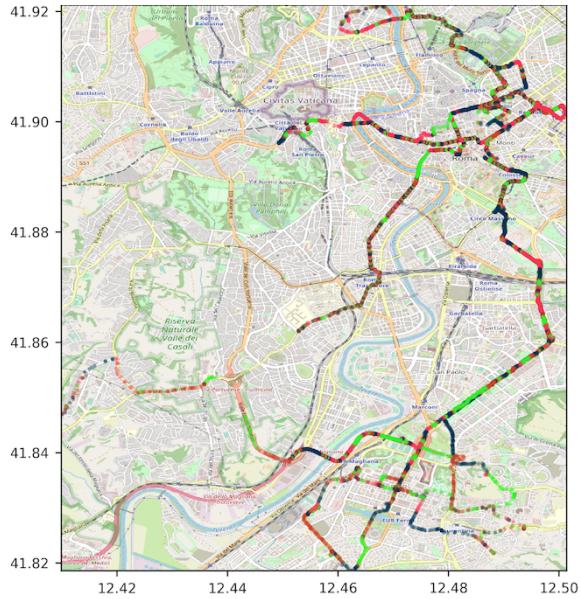


Figura 3: Trajetórias Consecutivas

A segunda versão desse algoritmo foi desenvolvida pensando em evitar fragmentações excessivas das trajetórias. Para isso, ele faz uma análise direta entre quaisquer dois pontos de um ID, verificando se os limites de distância e tempo entre os pontos foram ultrapassados.

No algoritmo 3, o *for* externo percorre a lista de IDs, criando uma nova lista de paradas para cada veículo na lista e um novo limite para o *for* interno. O ciclo interno percorre a lista de pontos do ID representado pelo atributo *key*. Primeiro é calculado a distância entre o ponto i e o ponto $i+1$. Caso essa distância ultrapasse o limite, cuja escolha é explicada na subseção 3.4.3, o próximo passo é o cálculo do tempo entre os dois pontos. Caso o limite de tempo também seja ultrapassado, adicionamos o índice do ponto i na lista de separações(*separator*). Ao fim de ambos os ciclos, a lista é retornada contendo os índices respectivos a cada ID.

Algoritmo 3 - Criação das viagens - versão 2

Algorithm 3

Entrada: Um vetor dos IDs($Keys$) e um dictionary contendo cada ID e seus respectivos pontos(P).

2: **Saída:** Uma lista contendo os índices onde devem ocorrer as separações das viagens.

```

for key in Keys do
    separator[key] = nova lista
    numPoints = |P[key]|
    for i=0 to numPoints -1 do
        dist = distancia(Pi,Pi + 1)
        if dist > limiteDist then
            temp = tempo(Pi,Pi + 1)
            if tempo > limiteTempo then
                separator[key].append(i)
        end if
    end if
end for
end for
16: Retorna separator; FIM

```

Os resultados da nova forma de fragmentação são visíveis na Figura 4 onde é observável uma diminuição na quantidade de fragmentação de trajetórias, representada pelas mudanças de cores.

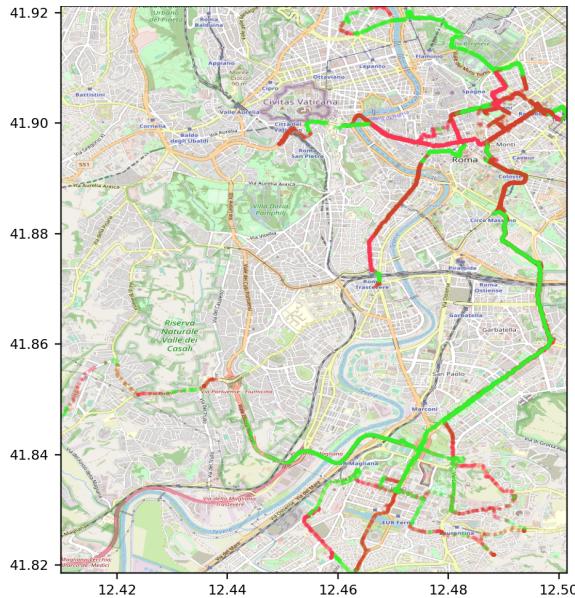


Figura 4: Trajetórias Consecutivas - versão 2

3.4.3 Cálculo dos limites utilizados no algoritmo de secção de traces

Os limites utilizados no algoritmo de separação de viagens foram escolhidos com base nas Funções de Densidade Cumulativa(CDF) presentes nas figura 5 e 6, retiradas da base de

dados de Roma.

Quatro combinações de valores foram selecionados para os testes levando em conta os dados apresentados nas CDFs. No caso da distância, foram escolhidos os limites de 25 e 50 metros, representando 95% e 99% dos valores. Já em relação ao tempo entre dois pontos, foram escolhidos 15 e 18 segundos, representando 95% e 98% dos dados, respectivamente.

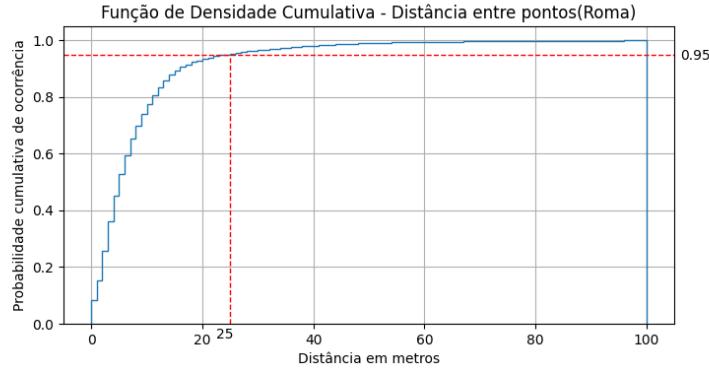


Figura 5: Distância entre pontos (Roma)

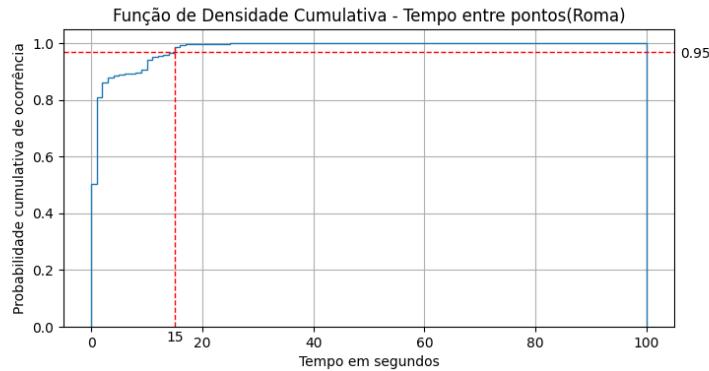


Figura 6: Tempo entre pontos (Roma)

A base de dados de Dublin, por não ser calibrada e por se tratar de ônibus, resultou em valores muito diferentes tanto para tempo quanto para distância, demonstrados nas figura 7 e 8. Nesse caso, os valores de distância selecionados foram 300 e 350 metros, representando aproximadamente 90% e 92% dos dados, respectivamente. Da mesma forma, os valores de tempo selecionados foram 30 e 38 segundos, respectivamente.

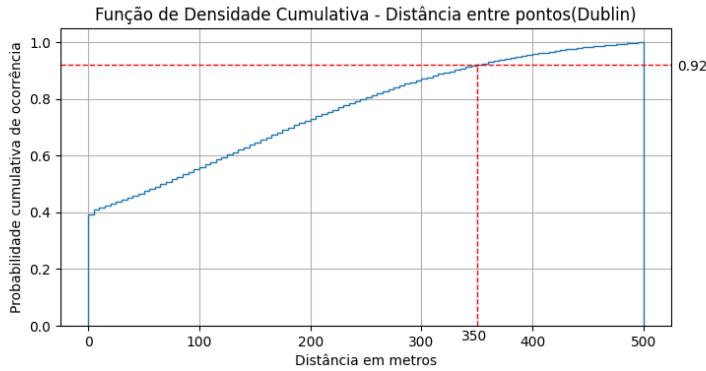


Figura 7: Distância entre pontos (Dublin)

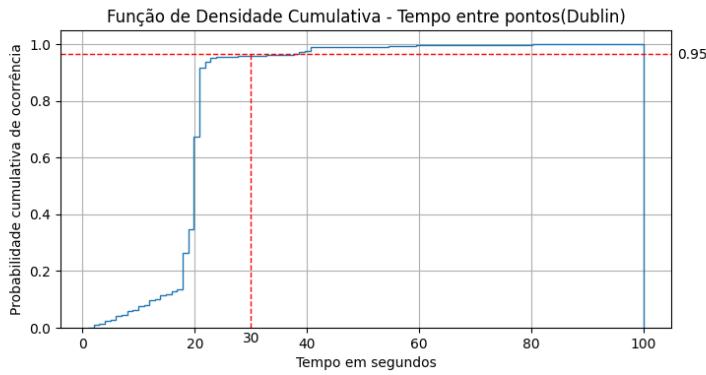


Figura 8: Tempo entre pontos (Dublin)

3.4.4 Matriz Origem/Destino(OD)

O quarto passo das cinco etapas em que o trabalho está dividido é o desenvolvimento de uma matriz OD para cada base. Essa matriz possui os valores de altura e largura de cada célula fixos em 100m, formando um quadrado de 100mx100m. Por conta disso, as matrizes formadas não possuem o mesmo número de linhas e colunas, já que as áreas de cada base de dados não formam necessariamente um quadrado.

O uso da matriz permite observar os pontos de maior concentração de início de viagens, bem como os pontos de maior concentração de fim de viagens. Essas informações são visualizadas através de um histograma 2D, e empiricamente através da busca manual da localização exata das células em destaque.

3.5 Métricas avaliadas

A quinta parte, e final, da avaliação das bases exige a escolha de algumas métricas. Neste trabalho foram selecionadas quatro métricas de estudo para ambas as bases, sendo: viagens por veículo, tamanho das trajetórias, tempo das trajetórias e velocidade média das trajetórias.

Além disso, com a implementação de uma matriz OD para cada base, na qual cada posição representava um quarteirão de 100m x 100m, foi analisado o padrão de movimentação dos veículos nas cidades. COMPLETAR****

4 Resultados Parciais

4.1 Análise inicial

Na análise inicial, foi feito um estudo básico para identificar qualquer tipo de indício de padrão de comportamento da mobilidade dos veículos. Foram analisados mapas de calor, que demonstram visualmente a quantidade de veículos em cada região em determinado momento, além do gráfico de densidade de veículos por hora do dia.

É de conhecimento geral que durante um dia o fluxo de veículos é heterogêneo. A Figura 9 demonstra claramente uma concentração bem maior de veículos durante alguns horários específicos do dia. Entre os horários com maior densidade de veículos, destacam-se três picos: O de 07:00 às 08:00, o de 11:00 às 12:00 e o de 19:00 às 20:00.

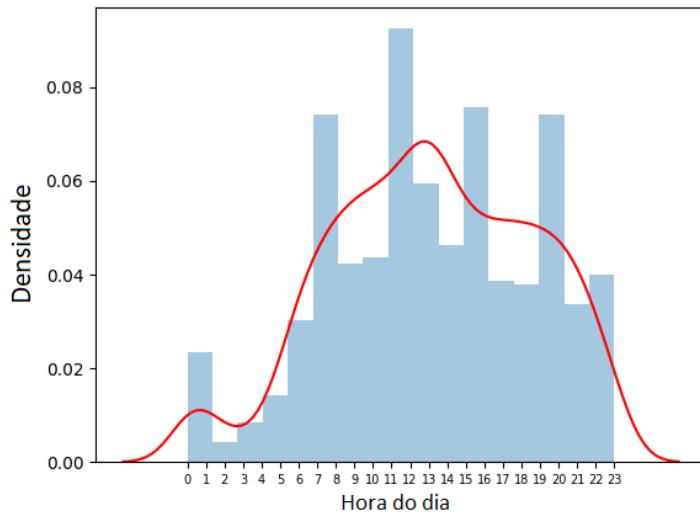


Figura 9: PDF de densidade de veículos por hora do dia (Roma)

Esses picos coincidem com os horários de uma jornada de trabalho padrão, onde o primeiro pico se daria no momento em que as pessoas da cidade se deslocam ao trabalho ou escola, o segundo ocorreria no momento de intervalo, onde há também uma tendência maior de jovens saindo e chegando nas escolas, e o terceiro seria o movimento de retorno.

A ideia de movimento pendular é reforçada pelos mapas de calor da cidade, onde pode ser visualizado a diferença de movimento de veículos de acordo com o horário, como pode ser visto na comparação das Figuras 10 e 11.

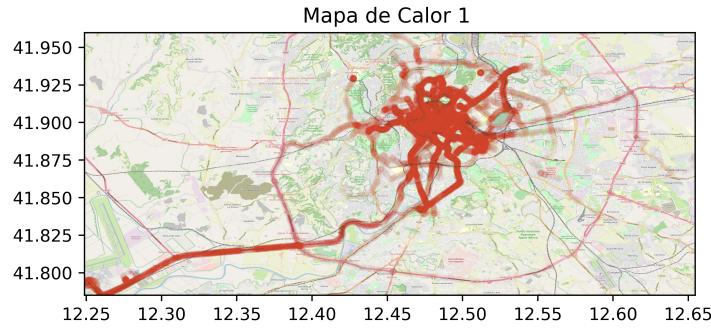


Figura 10: Mapa de Calor de densidade veicular de 00h-01h (Roma)

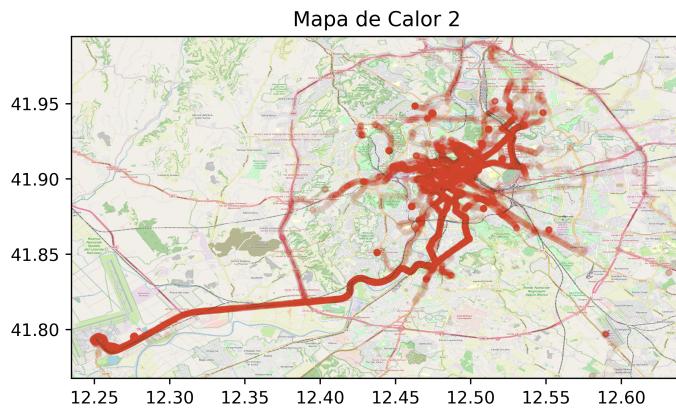


Figura 11: Mapa de Calor de densidade veicular de 07h-08h (Roma)

Na Figura 10, temos os dados de movimentações de táxis referentes ao horário de 00h à 01h. Já na Figura 11, temos os mesmos dados porém referentes ao horário de 07h às 08h. Comparando-se ambos os mapas, é visível, no Mapa de Calor 2(Figura 11), um maior fluxo de carros em direção ao aeroporto da cidade, localizado na região sudoeste do mapa. Há também um fluxo maior nas rodovias que contornam Roma, indicando ser um horário com maior entrada e saída de veículos da cidade. A região central se mantém concentrada nos dois mapas, porém entre 00:00 e 01:00 há uma dispersão de veículos menor.

Os gráficos mostrados evidenciam o que já é conhecido no ramo da Geografia como Migração Pendular, isto é, um movimento diário de pessoas que se deslocam a regiões da cidade

onde há maior concentração de comércio, indústrias e instituições de ensino durante o dia e, durante o início do período noturno, há o retorno dessas pessoas para suas residências, em geral mais afastadas dos centros das cidades. A causa desse fluxo é a horizontalização das cidades, em que a população com seu próprio crescimento tende a estabelecer novas moradias ao redor da cidade, cada vez mais afastadas do centro. Levando a um aumento cada vez maior no fluxo de veículos durante os horários de pico.

4.2 Avaliação das métricas

Foram estudadas 4 métricas de descrição das bases para melhor entendimento das diferenças e semelhanças entre os dois conjuntos de dados examinados. A primeira delas é a quantidade de viagens que cada veículo faz num período de 24 horas. A segunda métrica é o tamanho das trajetórias e a terceira, sua duração. A última métrica é a velocidade média dos veículos durante uma trip. Elas permitem uma comparação comportamental dos dois tipos de veículos, táxis, no caso de Roma, e ônibus, no caso de Dublin.

Também foram investigados histogramas 2D dos pontos de origem e dos pontos de destino das viagens feitas pelos veículos. O objetivo dessa análise é entender melhor o padrão comportamental da movimentação desses veículos no interior da cidade. Esses histogramas permitem o entendimento de como a cidade se movimenta ao longo do dia, quais pontos são críticos, quais são as tendências de movimentação de acordo com o horário, etc.

4.2.1 Viagens por veículo

Os gráficos de viagens por veículo mostram quantos veículos possuem determinada quantidade de viagens. Nesse caso, as viagens dos ônibus são consideradas apenas pontos iniciais aos finais, desconsiderando paradas. As figuras 12 e 13 ilustram um histograma com a quantidade de viagens por veículo para Roma e Dublin, respectivamente.

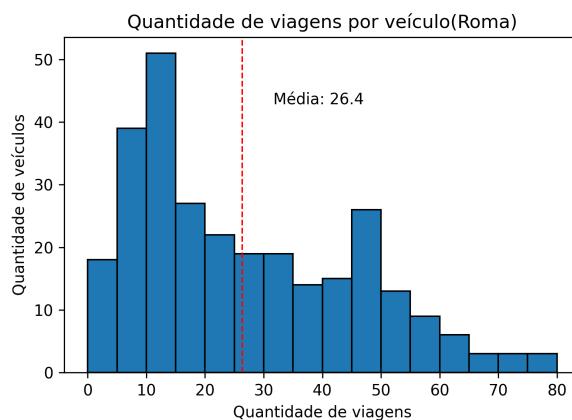


Figura 12: Quantidade de viagens por veículo (Roma)

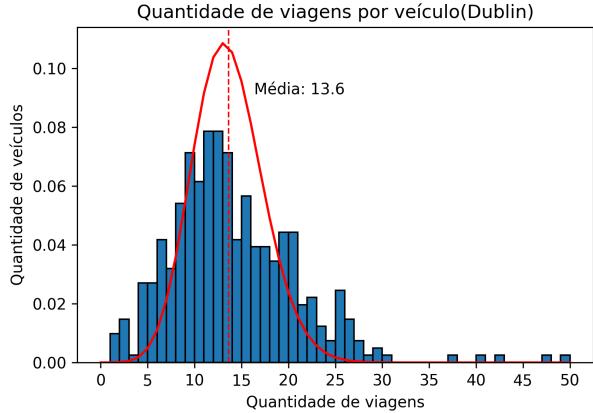


Figura 13: Quantidade de viagens por veículo (Dublin)

Vemos uma quantidade média menor de viagens para os veículos de Dublin, com uma concentração grande de veículos com até 30 viagens por dia. Roma mostra uma distribuição mais espaçada, contendo veículos com até 80 viagens e uma média de 26 viagens por veículo, praticamente o dobro da média de Dublin. Como ônibus são veículos com trajetórias normalmente longas, isso explicaria a quantidade menor de viagens diárias. Outro detalhe importante é que a distribuição de Dublin pode ser identificada como uma distribuição poisson, com a formação de um sino seguido de uma cauda.

4.2.2 Comprimento das trajetórias

A segunda métrica analisada foi o tamanho das trajetórias presentes nas bases de Roma e Dublin. As figuras 14 e 15 ilustram histogramas do tamanho das trajetórias para Roma e Dublin, respectivamente.

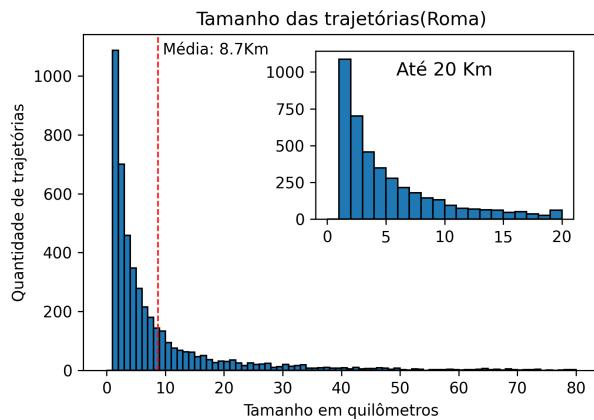


Figura 14: Tamanho das trajetórias (Roma)

No caso de Roma, percebemos um comportamento de cauda longa, com grande quantidade de trajetórias pequenas, a maioria menor que 10Km, e uma média de 8.7Km. Já Dublin se

aproxima mais de uma distribuição F com média de 23.3Km e poucas trajetórias abaixo de 10Km.

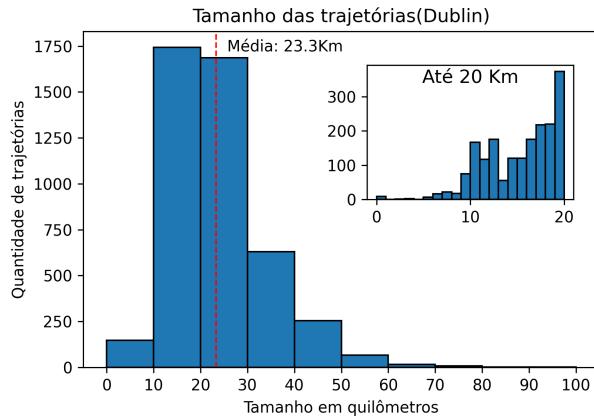


Figura 15: Tamanho das trajetórias (Dublin)

Isolando-se as trajetórias até 20Km, é percebido que o conjunto de dados de Dublin possui um crescimento do tamanho, possuindo trajetórias entre 10Km e 20Km. De forma diferente, Roma continua com o comportamento de cauda longa, com poucas instâncias maiores que 10Km.

Isso pode ser explicado pela diferença comportamental dos veículos. Ônibus possuem trajetórias já traçadas e planejadas, sendo incomum viagens pequenas pois seriam ineficientes. Táxis são comumente usados por pessoas que precisam de deslocamento rápido, o que seria mais apto a ser escolhido para trajetórias curtas ou longas que necessitem de deslocamento rápido.

4.2.3 Duração das trajetórias

Os histogramas de tempo das trajetórias representam quantas trajetórias estão incluídas em determinado intervalo de tempo. As figuras 16 e 17 ilustram esses histogramas para as cidades de Roma e Dublin, respectivamente. A diferença de resultados das duas bases é muito alta. Portanto, no caso de Roma a escala escolhida vai de 0 a 140 minutos, sendo o foco do gráfico de 0 a 20 minutos. Em Dublin a escala vai de 0 a 800 minutos, sendo o foco de 0 a 200 minutos.

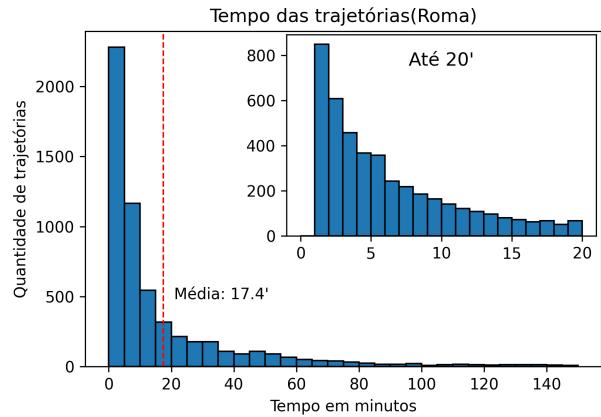


Figura 16: Duração das trajetórias (Roma)

Roma mostra possuir trajetórias muito curtas com média de 17.4 minutos, e, novamente, é observável o comportamento de cauda longa quando estabelecemos o limite do gráfico de até 140 minutos. Dublin possui uma média de tempo muito maior, sendo 110.4 minutos, e mantém o comportamento de distribuição F. É um padrão de comportamento esperado para ônibus que tendem a ter trajetórias longas.

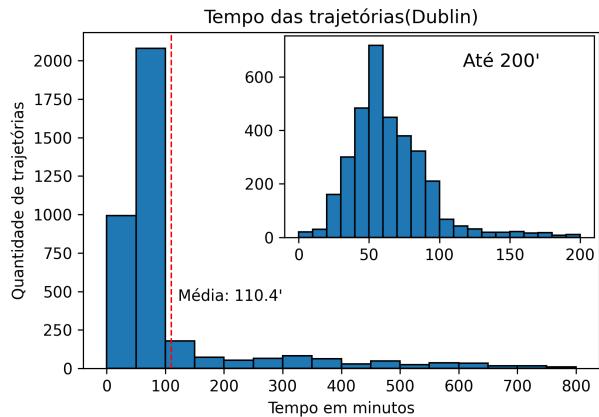


Figura 17: Duração das trajetórias (Dublin)

Nos focos dos gráficos, ou insets, percebe-se um comportamento de distribuição F mais clara para os valores de Dublin. Já Roma demonstra o mesmo comportamento de cauda longa, onde a quantidade de trajetórias existentes diminui conforme a duração cresce.

Os gráficos das Figuras 16 e 17 reforçam a ideia obtida pelos histogramas de tamanho das trajetórias, confirmando que nos táxis há uma tendência das trajetórias serem curtas.

4.2.4 Velocidade média por trajetória

Os histogramas de velocidade média das trajetórias expõem quantas trajetórias possuem uma velocidade média dentro de um intervalo específico e são representadas nas Figuras 18, no caso

de Roma, e 19, no caso de Dublin. Da mesma forma que a métrica avaliada anteriormente, os resultados destoaram, portanto a escala dos insets foi diferenciada, onde Roma vai de 0 a 250 Km/H e Dublin, 0 a 125 Km/H.

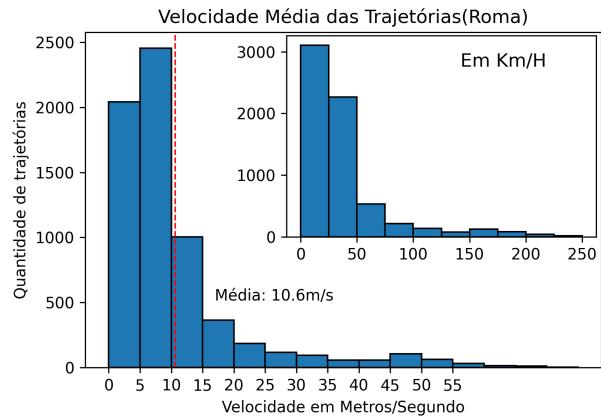


Figura 18: Velocidade média por trajetória (Roma)

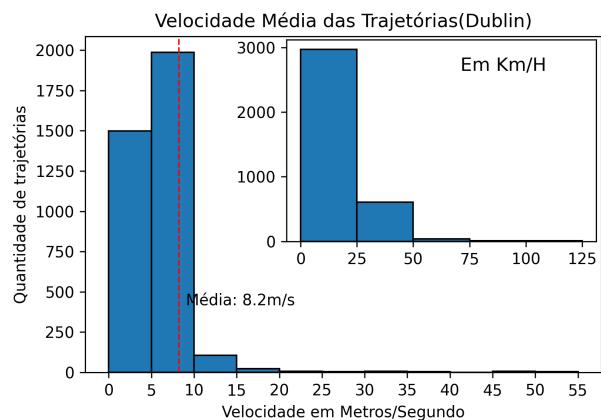


Figura 19: Velocidade média por trajetória (Dublin)

A diferença entre as médias não é muito grande entre as duas bases, sendo a de Roma pouco maior. Como ônibus são veículos de rotação interna na cidade, é esperável que a velocidade média seja mais baixa, além de controlada, o que pode ser confirmado pelo gráfico com a maior parte das trajetórias de velocidade média até 20Km/H. Os táxis por terem mais liberdade de movimentação, tendem a se mover mais rápido, o que também é demonstrado nos gráficos com uma quantidade um pouco maior de trajetórias cuja média de velocidade é acima dos 20Km/H. É visível também, que no caso de Roma há uma dispersão maior dos dados, já Dublin possui uma grande concentração entre 0 e 20Km/H, o que pode indicar um controle de velocidade nos ônibus.

4.2.5 Pontos de concentração de origem/destino das viagens

Heatmaps contendo informações dos pontos de origem/destino foram plotados com o intuito de melhorar nosso entendimento da movimentação dos veículos em ambas cidades. Eles podem ser observados nas figuras 20 e 21, para Dublin.

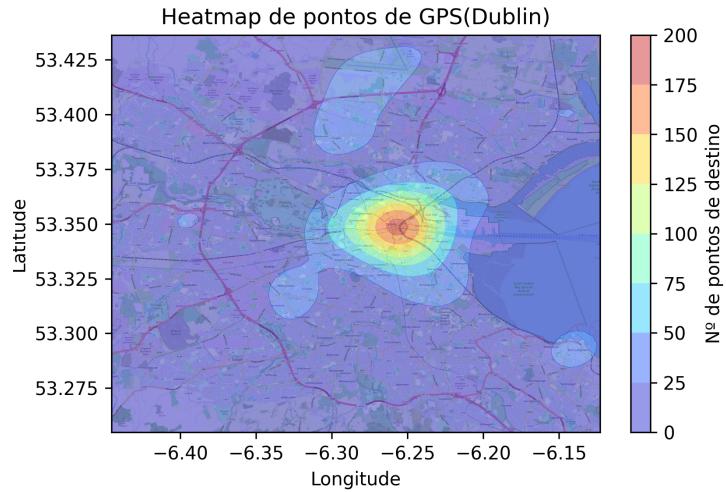


Figura 20: Heatmap dos pontos de origem/destino (Dublin)

No caso de Dublin, há uma grande concentração de início/fim de viagens na região central da cidade. Também ocorrem duas bolhas visíveis fora da região central. A primeira, ao norte, fica localizada na entrada do aeroporto. Já a segunda, na região sudeste, está localizada no porto da cidade. Há também algumas pequenas bolhas espalhadas em regiões comerciais da cidade.

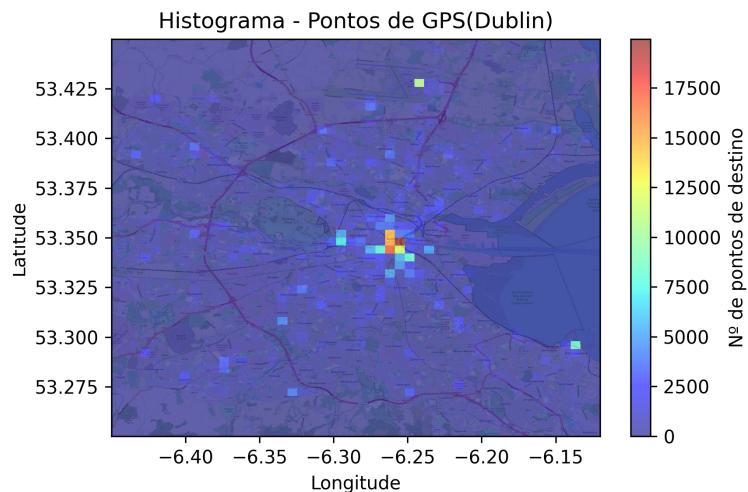


Figura 21: Histograma2D dos pontos de origem/destino (Dublin)

A partir dessa informações é concluído que a tendência é de haver grande concentração de

viagens em regiões de embarque/desembarque para outros municípios e, principalmente, em regiões comerciais da própria cidade.

Nas Figuras 22 e 23, vemos também Heatmap de Dublin. Esses, porém, constam apenas pontos de Origem de trips de GPS.

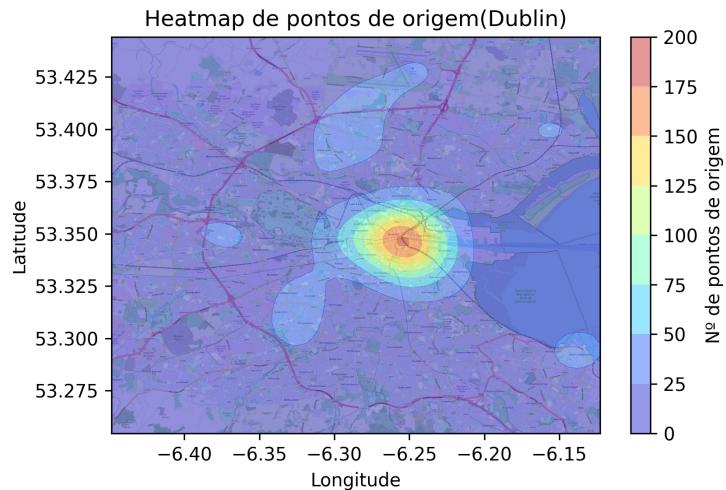


Figura 22: Heatmap dos pontos de origem (Dublin)

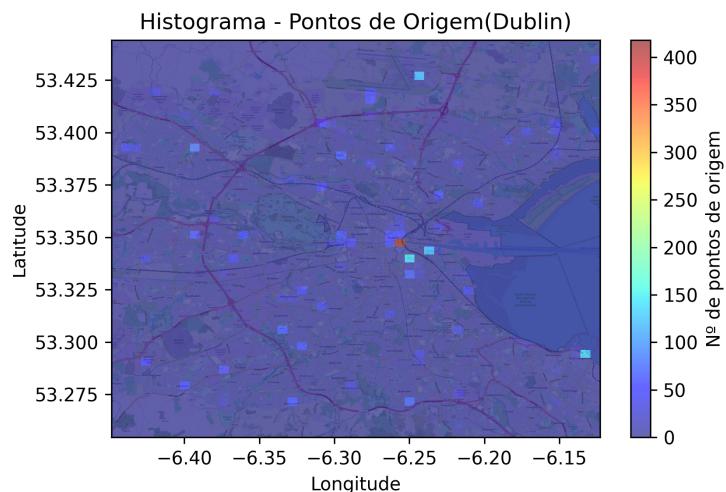


Figura 23: Histograma2D dos pontos de origem (Dublin)

No caso do Heatmap de origem de Dublin, comparado aos Heatmaps gerais(Figuras 20 e 21), há ligeiramente uma menor concentração de pontos de origem no centro da cidade, e uma maior concentração um pouco maior nas bolhas periféricas localizadas nas regiões comerciais.

As Figuras 24 e 25 possuem a mesma ideia das 22 e 23, mas para os pontos de destino das viagens.

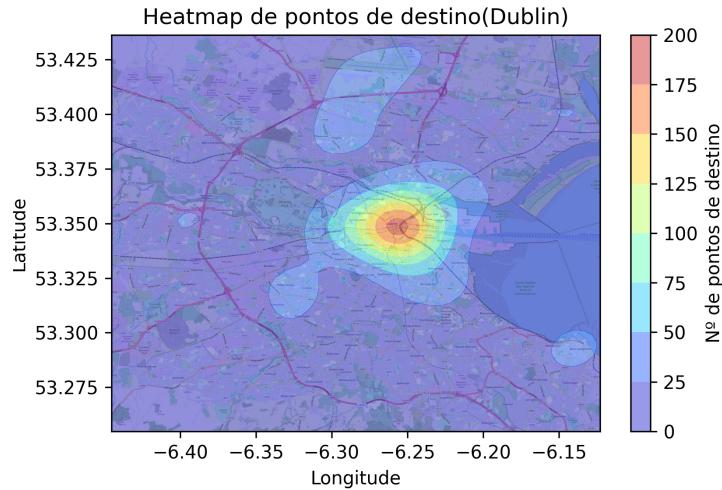


Figura 24: Heatmap dos pontos de destino (Dublin)

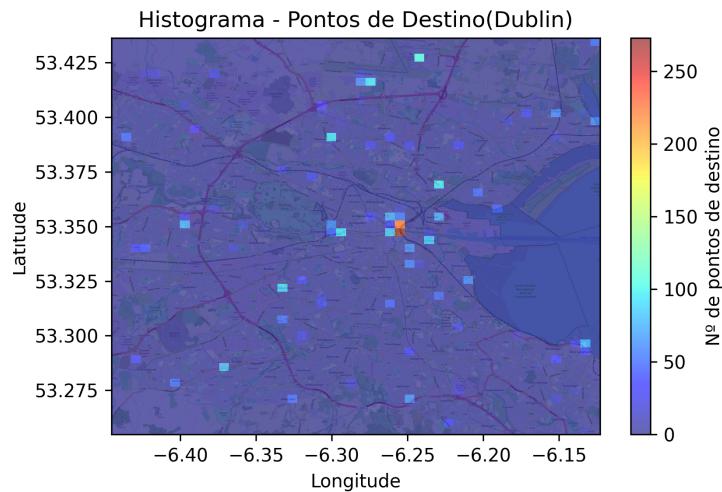


Figura 25: Histograma2D dos pontos de destino (Dublin)

Comparados aos heatmaps anteriores, os pontos de destino possuem distribuição muito parecida com os das Figuras 20 e 21.

Há pouca diferença entre a concentração de pontos de origem e destino. É provável que haja uma diferença muito maior se isolarmos esses pontos por cada hora do dia, já que, como foi mencionado anteriormente, existe um movimento natural de pêndulo em grandes cidades. No caso de Dublin, por se tratar de ônibus, essa diferença pode não ser muito aparente, tendo em vista que rotas de ônibus costumam ser fixas.

5 Conclusão

6 Trabalhos futuros

Referências

- Basta, N., ElNahas, A., Grossmann, H. P. & Abdennadher, S. (2016), Generic geo-social mobility model for vanet, *in* '2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)', IEEE, pp. 1–5.
- Celes, C., Boukerche, A. & Loureiro, A. A. (2019), Towards understanding of bus mobility for intelligent vehicular networks using real-world data, *in* '2019 IEEE Global Communications Conference (GLOBECOM)', IEEE, pp. 1–6.
- Cintra, M. (2013), 'A crise do trânsito em são paulo e seus custos', *GV EXECUTIVO* 12(2), 58–61.
- Gainaru, A., Dobre, C. & Cristea, V. (2009), A realistic mobility model based on social networks for the simulation of vanets, *in* 'VTC Spring 2009-IEEE 69th Vehicular Technology Conference', IEEE, pp. 1–5.
- Gong, L., Sato, H., Yamamoto, T., Miwa, T. & Morikawa, T. (2015), 'Identification of activity stop locations in gps trajectories by density-based clustering method combined with support vector machines', *Journal of Modern Transportation* 23(3), 202–213.
- Kong, X., Xia, F., Ning, Z., Rahim, A., Cai, Y., Gao, Z. & Ma, J. (2018), 'Mobility dataset generation for vehicular social networks based on floating car data', *IEEE Transactions on Vehicular Technology* 67(5), 3874–3886.
- Li, Q., Zheng, Y., Xie, X., Chen, Y., Liu, W. & Ma, W.-Y. (2008), Mining user similarity based on location history, *in* 'Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems', pp. 1–10.
- Ning, Z., Xia, F., Ullah, N., Kong, X. & Hu, X. (2017), 'Vehicular social networks: Enabling smart mobility', *IEEE Communications Magazine* 55(5), 16–55.