DESAFIO CIVITAS

EMD

DETECÇÃO DE PLACAS CLONADAS NO RIO DE JANEIRO

DESAFIO TÉCNICO PARA A VAGA NA PREFEITURA DO RIO DE JANEIRO

AMANDA AMARO

amandacsamaro@gmail.com

RIO DE JANEIRO, 01 DE JULHO 2024

1 Introdução

O estudo analisa dados de leitura de radares para identificar possíveis casos de clonagem de placas de veículos no Rio de Janeiro. A clonagem de placas é uma prática ilegal que resulta em multas injustas e fraudes.

Objetivo do Estudo

O objetivo é identificar possíveis placas clonadas, abordando inconsistências nos dados de leitura dos radares. O foco será em:

- Limpeza dos dados
- Verificar a consistência geográfica das leituras.
- Detectar anomalias nas velocidades capturadas.
- Identificar placas associadas a múltiplos tipos de veículos.
- Definição de um modelo para identificar placas clonadas

Foi utilizado SQL para processamento inicial dos dados no BigQuery e Python (Pandas) para análises e visualizações adicionais. A metodologia desenvolvida visa garantir a precisão dos resultados ao obter possíveis placas suspeitas e fornecer insights para melhorar a fiscalização.

2

Metodologia

Para a análise exploratória dos dados, utilizou-se Python com a biblioteca Pandas para carregar, inspecionar e manipular os dados. Os dados foram analisados em várias etapas para entender sua estrutura e identificar possíveis inconsistências que poderiam influenciar na análise de clonagem de placas.

Limpeza dos Dados

O banco contém 36.358.536 registros (Código 2.1), com a coluna datahora_captura apresentando 1.816.325 valores nulos (Código 2.2). Considerando que a integridade dos dados é crucial para a confiabilidade da análise e que se trata de 4,99% dos dados, para este trabalho eles foram desconsiderados (Código 2.3).

```
1  SELECT COUNT(*) AS total_records
2  FROM `rj-cetrio.desafio.readings_2024_06`;
```

Código 2.1: *Análise de quantidade de dados.*

```
1  SELECT
2    COUNTIF(datahora IS NULL) AS datahora_nulls,
3    COUNTIF(datahora_captura IS NULL) AS datahora_captura_nulls,
4    COUNTIF(placa IS NULL) AS placa_nulls,
5    COUNTIF(empresa IS NULL) AS empresa_nulls,
6    COUNTIF(tipoveiculo IS NULL) AS tipoveiculo_nulls,
7    COUNTIF(velocidade IS NULL) AS velocidade_nulls,
8    COUNTIF(camera_numero IS NULL) AS camera_numero_nulls,
9    COUNTIF(camera_latitude IS NULL) AS camera_latitude_nulls,
10   COUNTIF(camera_longitude IS NULL) AS camera_longitude_nulls
11 FROM `rj-cetrio.desafio.readings_2024_06`;
```

Código 2.2: *Análise de dados nulos.*

```sql
SELECT *
FROM `rj-cetrio.desafio.readings_2024_06`
WHERE
  datahora IS NOT NULL AND
  datahora_captura IS NOT NULL AND
  placa IS NOT NULL AND
  empresa IS NOT NULL AND
  tipoveiculo IS NOT NULL AND
  velocidade IS NOT NULL AND
  camera_numero IS NOT NULL AND
  camera_latitude IS NOT NULL AND
  camera_longitude IS NOT NULL;
```

Código 2.3: *Seleção de dados nulos distintos.*

Todas essas análises e mapas podem ser vistos com mais detalhes no notebook disponibilizado no repositório do GitHub ou direto pelo Collab.

Análise Exploratória

A análise exploratória é fundamental para entender a estrutura e as possíveis inconsistências nos dados de leitura dos radares. Foram realizadas as seguintes etapas:

- Análise de Outliers Geográficos
- Verificação de Câmeras com Coordenadas Duplicadas
- Análise de Velocidades Inconsistentes

Nos códigos a seguir a utilização do filtro WHERE (usado para excluir itens com nulos) foi omitida para melhorar a legibilidade.

Análise de Outliers Geográficos

Foram identificadas coordenadas de câmeras localizadas em áreas geograficamente improváveis para o escopo do projeto como Petrópolis, no oceano próximo ao Rio de Janeiro e no oceano ao redor da África. Esses pontos foram considerados erros de leitura portanto foram retirados, diminuindo a quantidade de cameras de 1421 para 1405 ao utilizar uma barreira geográfica (Código 2.4).

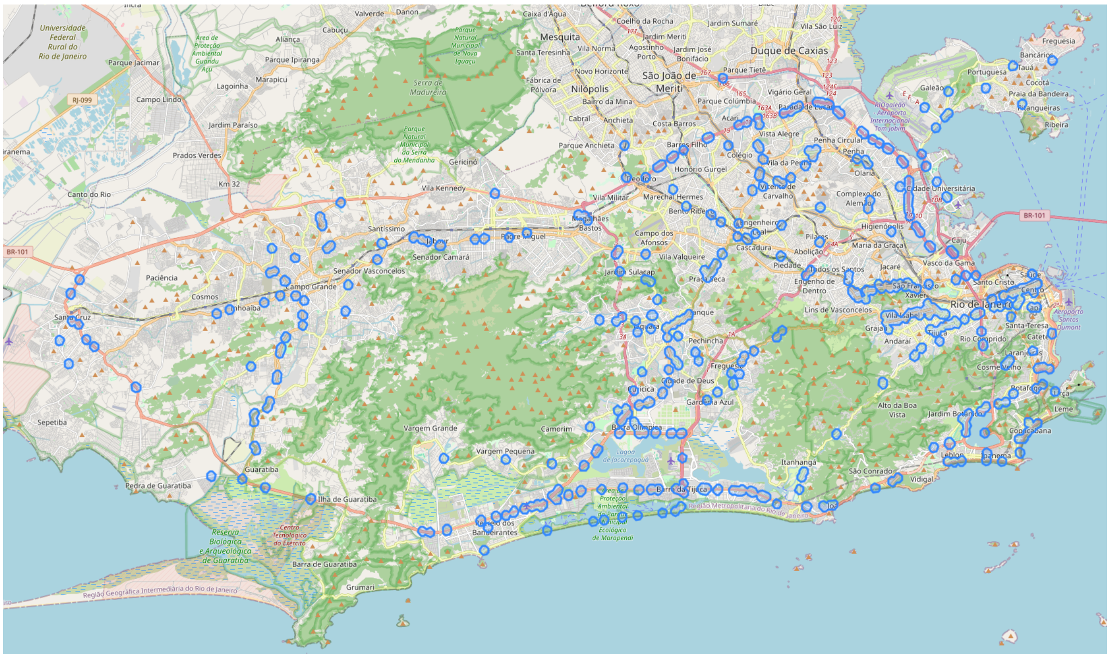

Figura 2.1: *Os pontos azuis representam as coordenadas dos radares dentro do limite do Rio de Janeiro*

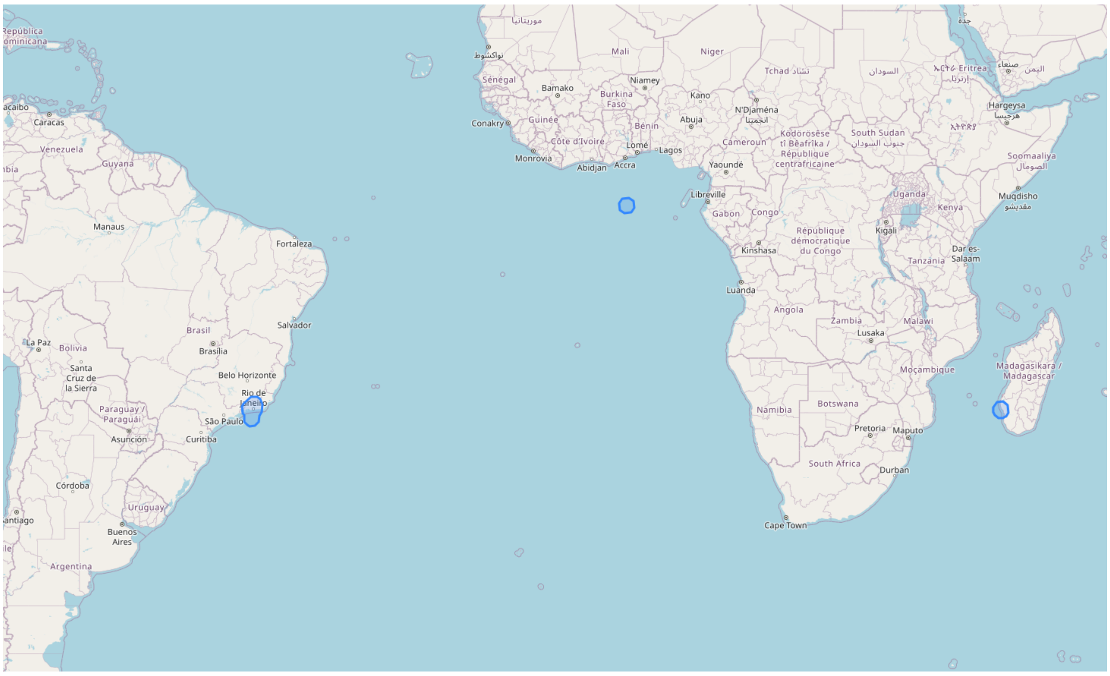

Figura 2.2: *Localização dos Outliers nas coordenadas dos radares.*

```sql
WITH
filtered_valid_data AS (
  SELECT DISTINCT
    TO_HEX(camera_numero) AS camera_numero,
    TO_HEX(placa) AS placa,
    TO_HEX(empresa) AS empresa,
    TO_HEX(tipoveiculo) AS tipoveiculo,
    camera_latitude,
    camera_longitude,
    datahora,
    datahora_captura,
    velocidade
  FROM
    `rj-cetrio.desafio.readings_2024_06`

cameras_out_of_boundary AS (
  SELECT
    camera_numero,
    camera_latitude,
    camera_longitude,
    ST_WITHIN(
      ST_GEOGPOINT(camera_longitude, camera_latitude),
      ST_GEOGFROMTEXT('POLYGON((-43.795 -23.082, -43.105 -23.082, -43.105
      ↪  -22.738, -43.795 -22.738, -43.795 -23.082))')
    ) AS within_boundary
  FROM
    filtered_valid_data
  GROUP BY
    camera_numero, camera_latitude, camera_longitude
  HAVING
    within_boundary = FALSE
)

SELECT
  camera_numero,
  camera_latitude,
  camera_longitude
FROM
  cameras_out_of_boundary;
```

Código 2.4: *Seleção de dados dentro dos limites do Rio de Janeiro.*

Análise de Velocidades Inconsistentes

A tabela abaixo resume as estatísticas descritivas das velocidades capturadas:

Tabela 2.1: *Descrição dos dados de velocidade (km/h) por empresa e pelo total.*

Empresa	Empresa A	Empresa B	Empresa C	Total
count	8.434.752	18.195.136	2.793.091	34.542.211
mean	34,30	38,52	31,70	37,01
std	8,53	16,46	14,39	15,01
min	1	0	1	0
25%	29	27	22	28
50%	34	38	30	36
75%	39	50	40	46
max	160	255	149	255

Sendo Empresa A = 0891967b413fa4, Empresa B = 1e2545af9d48c6 e Empresa C = 2ce01a80c7f3d0

Velocidades chegando a 255 km/h, o que não é plausível para uma área urbana. Além de que considerando que a velocidade no 75° percentil é de 46 km/h, essas leituras extremas indicam possíveis falhas de medição em alguns radares.

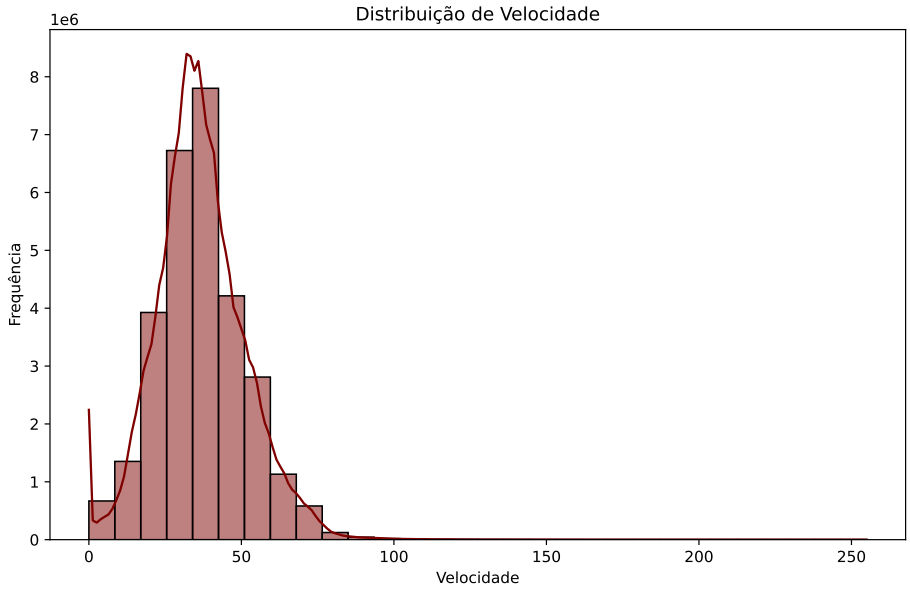

Figura 2.3: *Gráfico de distribuição da velocidade.*

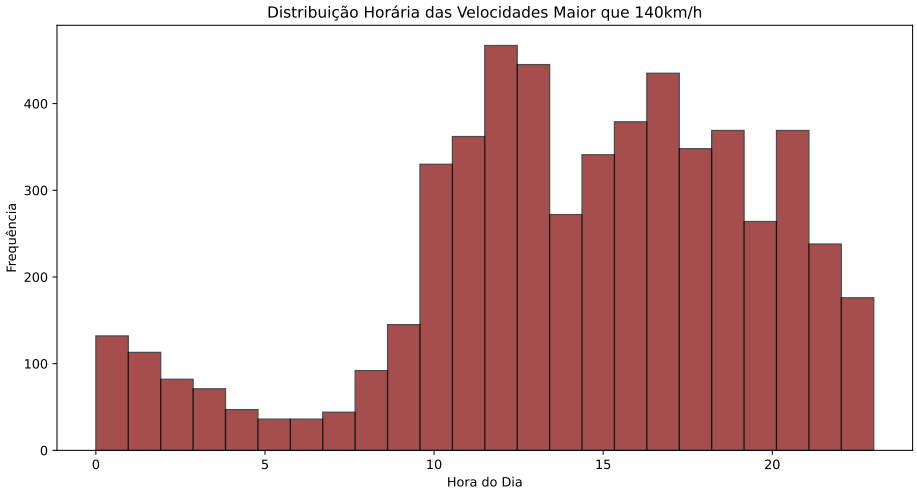

Figura 2.4: *Histograma com a frequência de dados com a velocidade acima de 140km/h com relação a hora.*

Os gráficos indicam velocidades mais altas durante manhãs e tardes, períodos típicos de maior trânsito. Isso sugere possíveis erros nos radares, uma vez que é improvável que tantas ocorrências de alta velocidade coincidam com horários de pico. Para lidar com esse problema qualquer dado acima de 140km/h entre 6 e 22 horas foi considerado um outlier. Para uma melhor análise seria importante uma auditoria nos equipamentos de medição e revisão dos dados para verificar e corrigir possíveis falhas.

Além disso, essa análise dos dados revelou um total de 306.884 registros com velocidade igual a zero. Este número de leituras não é muito significativo e com as informações dadas não é possível identificar se são problemas na coleta ou no processamento dos dados.

As causas podem incluir:

- **Falhas de Medição**: Problemas técnicos nos dispositivos de monitoramento que resultam em leituras incorretas de velocidade.
- **Paradas de Veículos**: Momentos em que os veículos estavam parados, como em semáforos ou congestionamentos.
- **Erros de Cadastro**: Problemas na configuração dos radares que poderiam levar a registros incorretos.

Detecção de Placas com Múltiplos Tipos de Veículos

Para identificar possíveis casos de clonagem de placas, a metodologia envolveu verificar as placas vinculadas a mais de um tipo de veículo. Inicialmente, foram filtrados os dados inválidos e placas que aparecem apenas uma vez, pois estas não fornecem informações relevantes sobre clonagem (Código 2.5). O número de placas que aparecem apenas uma vez é de 4.577.181, representando 12, 59% com relação ao total.

```sql
WITH
filtered_valid_data AS (
  SELECT DISTINCT
    TO_HEX(camera_numero) AS camera_numero,
    TO_HEX(placa) AS placa,
    TO_HEX(empresa) AS empresa,
    TO_HEX(tipoveiculo) AS tipoveiculo,
    camera_latitude,
    camera_longitude,
    datahora,
    datahora_captura,
    velocidade
  FROM
    `rj-cetrio.desafio.readings_2024_06`

-- Filtra placas que aparecem mais de uma vez
filtered_plates AS (
  SELECT
    placa
  FROM
    filtered_valid_data
  GROUP BY
    placa
  HAVING
    COUNT(*) > 1
),

-- Filtra dados válidos com placas que aparecem mais de uma vez
valid_data_with_multiple_plates AS (
  SELECT
    *
  FROM
    filtered_valid_data
  WHERE
    placa IN (SELECT placa FROM filtered_plates)
),

-- Identifica placas com múltiplos tipos de veículos
multiple_vehicle_types AS (
  SELECT
    placa,
    COUNT(DISTINCT tipoveiculo) AS vehicle_type_count
  FROM
    valid_data_with_multiple_plates
  GROUP BY
    placa
  HAVING
    vehicle_type_count > 1
)
SELECT
  placa,
  vehicle_type_count
FROM
  multiple_vehicle_types
ORDER BY
  placa;
```

Código 2.5: *Código para selecionar veículos com múltiplos tipos.*

```sql
    -- Calcula diferença de tempo entre registros consecutivos
    base_time_diff_data AS (
      SELECT
        *,
        LEAD(datahora) OVER (PARTITION BY placa ORDER BY datahora) AS
        ↪ next_datahora,
        LEAD(camera_numero) OVER (PARTITION BY placa ORDER BY datahora) AS
        ↪ next_camera_numero,
        LEAD(camera_latitude) OVER (PARTITION BY placa ORDER BY datahora) AS
        ↪ next_latitude,
        LEAD(camera_longitude) OVER (PARTITION BY placa ORDER BY datahora) AS
        ↪ next_longitude,
        TIMESTAMP_DIFF(LEAD(datahora) OVER (PARTITION BY placa ORDER BY datahora),
        ↪ datahora, SECOND) AS time_diff
      FROM
        valid_data_with_multiple_plates
    ),

    -- Calcula a distância geodésica entre pontos consecutivos
    geo_distance_calculation AS (
      SELECT
        *,
        ST_DISTANCE(
          ST_GEOGPOINT(camera_longitude, camera_latitude),
          ST_GEOGPOINT(next_longitude, next_latitude)
        ) / 1000 AS distance_km
      FROM
        base_time_diff_data
    ),

    -- Calcula velocidade baseada na distância e tempo
    velocity_calculation AS (
      SELECT
        *,
        CASE
          WHEN time_diff > 0 THEN distance_km / (time_diff / 3600)
          ELSE NULL
        END AS velocity_kmh
      FROM
        geo_distance_calculation
    ),
```

Código 2.6: *Parte do código final que foi utilizada para fazer os cálculos de distância, tempo e velocidade.*

Cálculo de Distância entre Pontos e Velocidade

Para detectar possíveis casos de clonagem de placas, uma parte crucial da metodologia envolveu o cálculo da distância entre os pontos de leitura e a velocidade dos veículos (Código 2.6). Esse processo foi realizado em várias etapas:

- **Agrupamento por Placa e Datahora:** Inicialmente, os dados foram agrupados por placa e ordenados por data e hora. Esse agrupamento permitiu analisar a sequência de leituras para cada placa de forma cronológica.
- **Cálculo da Diferença de Tempo entre Leituras Consecutivas:** Utilizando a função LEAD, foi possível calcular a diferença de tempo entre leituras consecutivas de uma mesma placa. A função LEAD gera uma nova coluna com o valor da próxima linha dentro do grupo, permitindo calcular o tempo transcorrido entre duas leituras consecutivas.
- **Cálculo da Distância Geodésica:** A distância geodésica entre as coordenadas das leituras consecutivas foi calculada utilizando a função ST_DISTANCE. Essa função computa a distância entre dois pontos geográficos (latitude e longitude), que foi posteriormente convertida de metros para quilômetros.
- **Cálculo da Velocidade:** Com a distância e a diferença de tempo calculadas, foi possível determinar a velocidade média do veículo entre duas leituras consecutivas.

3

Abordagem e Resultados

A abordagem utilizada no Código 3.1 para detectar possíveis casos de clonagem de placas resultou na identificação de 583.885 placas suspeitas, sendo 427.574 placas que se enquadraram no caso de ter diversos tipos de veículos.

As principais métricas e critérios empregados são descritos nessa abordagem foram:

- **Escolha da Velocidade Média Máxima**: Para determinar anomalias de velocidade, foi escolhido um valor conservador para a velocidade média máxima. A análise dos dados indicou uma média de aproximadamente 44 km/h. Considerando essa média, foi definido um limite de 70 km/h para identificar possíveis casos de clonagem. Este valor é significativamente mais alto que a média, proporcionando uma margem de segurança que minimiza falsos positivos.
- **Diferença de Tempo**: Foi aplicado um critério conservador ao definir a diferença de tempo entre leituras consecutivas de uma mesma placa. Estabeleceu-se que diferenças de tempo menores que 3600 segundos (1 hora) deveriam ser analisadas, garantindo que apenas deslocamentos plausíveis em períodos razoáveis fossem considerados.
- **Análise das Velocidades Elevadas**: Foi imposto um limite de 140 km/h para velocidades capturadas entre 6h e 22h, excluindo qualquer leitura acima desse valor. Esta escolha baseou-se no fato de que, na cidade do Rio de Janeiro, não há rodovias com limites de velocidade superiores a 120 km/h. Assim, qualquer registro acima de 140 km/h foi considerado uma anomalia e excluído da análise para garantir a integridade dos dados.

```sql
WITH
filtered_valid_data AS (
  SELECT DISTINCT
    TO_HEX(camera_numero) AS camera_numero,
    TO_HEX(placa) AS placa,
    TO_HEX(empresa) AS empresa,
    TO_HEX(tipoveiculo) AS tipoveiculo,
    camera_latitude,
    camera_longitude,
    datahora,
    datahora_captura,
    velocidade
  FROM
    `rj-cetrio.desafio.readings_2024_06`
  WHERE
    datahora IS NOT NULL
    AND datahora_captura IS NOT NULL
    AND placa IS NOT NULL
    AND empresa IS NOT NULL
    AND tipoveiculo IS NOT NULL
    AND velocidade IS NOT NULL
    AND camera_numero IS NOT NULL
    AND camera_latitude IS NOT NULL
    AND camera_longitude IS NOT NULL
    AND velocidade > 0
    AND NOT (velocidade > 140 AND EXTRACT(HOUR FROM datahora) BETWEEN 6 AND 22)
    AND ST_WITHIN(
        ST_GEOGPOINT(camera_longitude, camera_latitude),
        ST_GEOGFROMTEXT('POLYGON((-43.795 -23.082, -43.105 -23.082, -43.105
        ↪  -22.738, -43.795 -22.738, -43.795 -23.082))')
    )
),

-- Filtra placas que aparecem mais de uma vez
filtered_plates AS (
  SELECT
    placa
  FROM
    filtered_valid_data
  GROUP BY
    placa
  HAVING
    COUNT(*) > 1
),

-- Filtra dados válidos com placas que aparecem mais de uma vez
valid_data_with_multiple_plates AS (
  SELECT
    *
  FROM
    filtered_valid_data
  WHERE
    placa IN (SELECT placa FROM filtered_plates)
),
```

Código 3.1: *Abordagem final para os resultados de placas clonadas (pt.1).*

```sql
    -- Calcula diferença de tempo entre registros consecutivos
    base_time_diff_data AS (
      SELECT
        *,
        LEAD(datahora) OVER (PARTITION BY placa ORDER BY datahora) AS next_datahora,
        LEAD(camera_numero) OVER (PARTITION BY placa ORDER BY datahora) AS next_camera_numero,
        LEAD(camera_latitude) OVER (PARTITION BY placa ORDER BY datahora) AS next_latitude,
        LEAD(camera_longitude) OVER (PARTITION BY placa ORDER BY datahora) AS next_longitude,
        TIMESTAMP_DIFF(LEAD(datahora) OVER (PARTITION BY placa ORDER BY datahora), datahora, SECOND) AS time_diff
      FROM
        valid_data_with_multiple_plates
    ),

    -- Calcula a distância geodésica entre pontos consecutivos
    geo_distance_calculation AS (
      SELECT
        *,
        ST_DISTANCE(
          ST_GEOGPOINT(camera_longitude, camera_latitude),
          ST_GEOGPOINT(next_longitude, next_latitude)
        ) / 1000 AS distance_km
      FROM
        base_time_diff_data
    ),

    -- Calcula velocidade baseada na distância e tempo
    velocity_calculation AS (
      SELECT
        *,
        CASE
          WHEN time_diff > 0 THEN distance_km / (time_diff / 3600)
          ELSE NULL
        END AS velocity_kmh
      FROM
        geo_distance_calculation
    ),
```

Código 3.2: *Continuação da abordagem final para os resultados de placas clonadas (pt.2).*

```sql
-- Identifica placas com múltiplos tipos de veículos
multiple_vehicle_types AS (
  SELECT
    placa,
    COUNT(DISTINCT tipoveiculo) AS vehicle_type_count
  FROM
    valid_data_with_multiple_plates
  GROUP BY
    placa
  HAVING
    vehicle_type_count > 1
),

-- Identifica possíveis placas clonadas
possible_clones AS (
  SELECT
    v.*,
    mv.placa IS NOT NULL AS multiple_vehicle_types
  FROM
    velocity_calculation v
  LEFT JOIN
    multiple_vehicle_types mv
  ON
    v.placa = mv.placa
  WHERE
    (v.velocity_kmh > 70  AND v.time_diff< 3600)
    OR mv.placa IS NOT NULL
)

SELECT
  *
FROM
  possible_clones
ORDER BY
  placa, datahora;
```

Código 3.3: *Continuação da abordagem final para os resultados de placas clonadas (pt.3).*

4

Discussão e Conclusão

Considerações Conservadoras

Adotou-se uma abordagem conservadora ao selecionar os valores para a velocidade média máxima e a diferença de tempo entre leituras. Este cuidado minimiza a possibilidade de falsos positivos, assegurando que apenas os casos mais prováveis de clonagem sejam destacados.

Análise dos Locais de Maior Ocorrência

Durante a análise dos dados de leitura dos radares, identificou-se que havia várias câmeras registradas com as mesmas coordenadas geográficas. Para a análise no Rio de Janeiro, todas as câmeras foram agrupadas de acordo com a localização (Código 4.1).

```
1        SELECT
2            camera_latitude,
3            camera_longitude,
4            COUNT(DISTINCT TO_HEX(camera_numero)) AS num_cameras
5        FROM `rj-cetrio.desafio.readings_2024_06`
6        GROUP BY camera_latitude, camera_longitude
7        HAVING num_cameras > 1;
```

Código 4.1: *Agrupamento de cameras de acordo com a localizacao.*

Dessa foram identificados 10 locais com maior número de aparições de casos suspeitos, ajudando a identificar pontos críticos na cidade onde há maior incidência de possíveis veículos com placas clonadas passarem. A identificação desses hotspots é crucial para orientar ações de fiscalização e investigação.

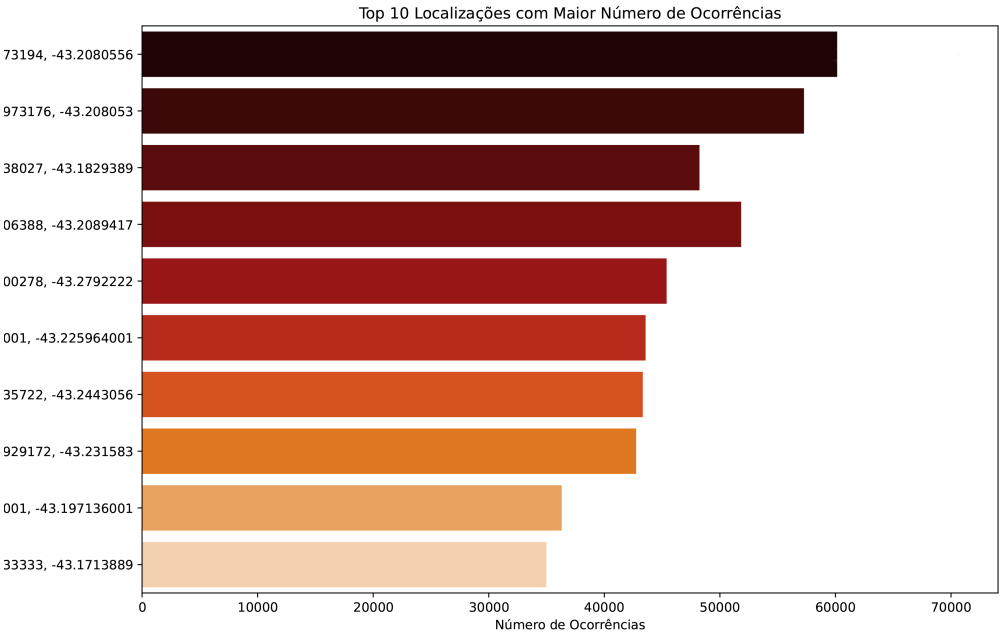

Figura 4.1: *Número de ocorrências por coordenada*

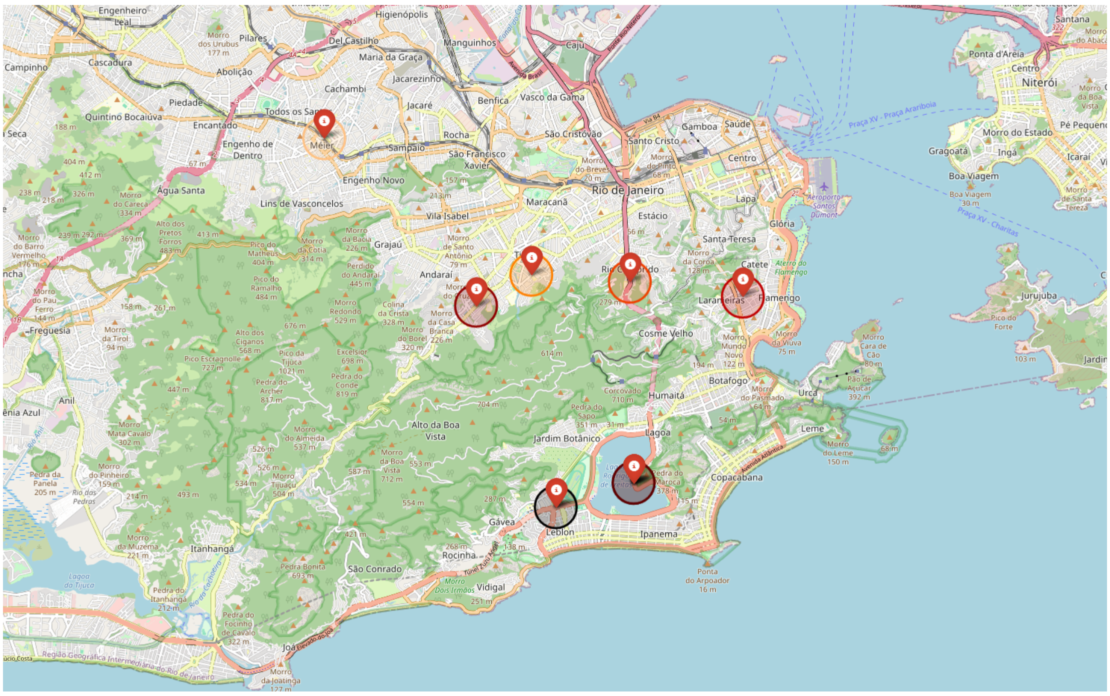

Figura 4.2: *Localização das coordenadas com maior número de ocorrência de suspeita de clonagem. Cada ponto está envolto em um círculo de cor referente à Figura 4.1. O número de tags parece menor por conta de radares muito próximos uns dos outros.*

Tipos de Veículos

A análise também revelou a distribuição dos tipos de veículos envolvidos nas leituras de placas suspeitas. A Tabela 4.1 apresenta a contagem dos diferentes tipos de veículos detectados e percebe-se que o tipo de veículo identificado como Tipo D representa a maioria das leituras suspeitas, com 8.721.661 ocorrências, seguido pelos outros tipos com menor frequência.

Tabela 4.1: *Contagem de cada tipo de veículo dentre o resultado de placas suspeitas.*

Tipo Veículo	Count
Tipo A	465.113
Tipo B	785.076
Tipo C	309.443
Tipo D	8.721.661
Total	10.281.293

Sendo Tipo A = 031cc0037e816d, Tipo B = 7a6376f47ca915, Tipo C = b88652111099ed e Tipo D = e2e0029fc0d3e5

Conclusão

A análise realizada foi conservadora ao identificar possíveis casos de clonagem de placas de veículos no Rio de Janeiro, mas atende bem ao propósito de ser um estudo inicial. Utilizando técnicas de análise de dados e SQL, foi possível filtrar, processar e detectar anomalias nos dados de leitura dos radares.

A abordagem apresentada pode ser utilizada como uma base para futuras análises e aprimoramentos. Além disso, a análise dos locais com maior incidência de casos suspeitos e a distribuição dos tipos de veículos fornecem informações valiosas para direcionar os esforços de combate à clonagem de placas.

Para aprimorar ainda mais a detecção de placas clonadas, recomenda-se integrar outras fontes de dados e utilizar técnicas de machine learning para identificar padrões mais complexos.

Trabalhos Futuros

A duplicidade de registros em uma mesma localidade pode introduzir redundâncias nos dados, afetando a precisão das análises de velocidade e de detecção de anomalias. Como melhoria futura, recomenda-se a implementação de um mecanismo para identificar e consolidar os dados provenientes de câmeras sobrepostas, garantindo assim uma base de dados mais limpa e precisa para estudos subsequentes.

Desafio Civitas
EMD

Detecção de Placas Clonadas no Rio de Janeiro

Amanda Amaro
amandacsamaro@gmail.com

Rio de Janeiro, 01 de Julho 2024