UNIVERSIDADE FEDERAL DO CEARÁ

CAMPUS QUIXADÁ

CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

PEDRO ALVES GONÇALVES JUNIOR

AVALIAÇÃO DE TÉCNICAS DE AUMENTO DE DADOS PARA TRAJETÓRIAS

QUIXADÁ

2021

PEDRO ALVES GONÇALVES JUNIOR

AVALIAÇÃO DE TÉCNICAS DE AUMENTO DE DADOS PARA TRAJETÓRIAS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação do Campus Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Ciência da Computação.

Orientadora: Profa. Ma. Lívia Almada Cruz

QUIXADÁ

2021

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

G627a Gonçalves Junior, Pedro Alves.
Avaliação de técnicas de aumento de dados para trajetórias / Pedro Alves Gonçalves Junior. – 2021.
43 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Ciência da Computação, Quixadá, 2021.
Orientação: Profa. Ma. Lívia Almada Cruz.

1. Análise de dados. 2. Trajetória. I. Título.

CDD 004

PEDRO ALVES GONÇALVES JUNIOR

AVALIAÇÃO DE TÉCNICAS DE AUMENTO DE DADOS PARA TRAJETÓRIAS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação do Campus Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Ciência da Computação.

Aprovada em: ____/____/____

BANCA EXAMINADORA

Profa. Ma. Lívia Almada Cruz (Orientadora)
Universidade Federal do Ceará (UFC)

Prof. Dr. Regis Pires Magalhães
Universidade Federal do Ceará (UFC)

Prof. Dr. Paulo de Tarso Guerra Oliveira
Universidade Federal do Ceará (UFC)

Prof. José Soares da Silva Neto
Instituto Federal de Educação, Ciência e Tecnologia
do Piauí (IFPI)

Aos meus pais, Ivanira e Pedro. A meu irmão, Wellington. A minha namorada, Bárbara. Vocês acreditaram em mim quando eu mesmo não acreditei.

AGRADECIMENTOS

Agradeço a Deus por me permitir chegar até aqui em meio a tantas dificuldades, por ser minha Fortaleza, meu Refúgio e por ter me dado a sabedoria necessária para realizar mais essa etapa em minha vida.

Agradeço aos meus pais, Ivanira e Pedro, pelo constante apoio, orientação, conselho e por todos os esforços feito para que eu pudesse chegar até aqui. Mãe, obrigado por ser meu maior exemplo de força, garra e dedicação. Pai, obrigado por todos os ensinamentos e toda proteção.

Agradeço ao meu irmão, Wellington, por sua amizade e por sempre me incentivar.

Agradeço à minha namorada, Bárbara, por seu grande apoio e compreensão. Obrigado por ser essa pessoa tão maravilhosa em minha vida.

Agradeço à minha família em geral, por sempre me incentivarem e acreditarem em mim.

Agradeço à todos os meus amigos, que de alguma forma me incentivaram. Em especial, Willamy e Iana Mary que me mostraram o real valor de uma amizade.

Agradeço à minha orientadora Lívia Almada, por sua orientação, ensinamentos e paciência comigo durante este trabalho. Obrigado Lívia!

Agradeço ao professor José Neto, pelas suas valiosas contribuições nesse trabalho.

Agradeço a todos os meus professores durante a graduação, em especial, Paulo de Tarso, Viviane Menezes, Lívia Almada, Regis Pires e Francicleber Martins pelo seu esforço constante de tornar seus alunos bons profissionais e boas pessoas.

Agradeço à todos que contribuíram positivamente para a minha formação.

"O sonho é que leva a gente para frente. Se a gente for seguir a razão, fica aquietado, acomodado!"

(Ariano Suassuna)

RESUMO

Atualmente, há uma grande quantidade de dados sendo geradas todos os dias, porém, esses dados não são obtidos de maneira uniforme, causando um problema de desbalanceamento. O mesmo ocorre com os dados de trajetória de veículos, nos quais os sensores de tráfego não capturam a passagem de objetos em movimento com a mesma frequência, gerando dados desbalanceados. Além do fato dos sensores estarem posicionados em locais fixos, o que não permite que o rastreamento completo dos objetos em movimento a serem capturados, gerando trajetórias esparsas. Esses problemas podem dificultar o desempenho dos modelos de predição de próxima localização. Com isso, o presente trabalho propõe técnicas de aumento de dados para trajetórias a fim de reduzir o problema da esparsidade de dados e do desbalanceamento com uso de técnicas de reamostragem. E ainda, criar diversos modelos preditivos para avaliar se há associação positiva entre a aplicação dessas técnicas e o desempenho de algoritmos de aprendizado de máquina.

Palavras-chave: Esparsidade. Desbalanceamento. Aumento de Dados. Predição de Próxima Localização. Trajetória

ABSTRACT

Currently, there is a large amount of data being generated every day, however, this data is not obtained uniformly, causing an imbalance problem. The same occurs with vehicle trajectory data, in which traffic sensors do not capture the passage of objects in motion with the same frequency, generating unbalanced data. In addition to the fact that the sensors are positioned at fixed locations, which does not allow the complete tracking of moving objects to be captured, generating sparse trajectories. These problems can hinder the performance of the prediction models of next location. With this, the present work proposes data augmentation techniques for trajectories in order to reduce the problem of data sparsity and imbalance with the use of resampling techniques. And yet, create several predictive models to assess whether there is a positive association between the application of these techniques and the performance of machine learning algorithms.

Keywords: Sparse. Unbalance. Inceased Data. Next Location Prediction. Trajectory

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

GPS	*Global Positioning System*
EST	*External Sensor Trajectory*
NLP	*Natural Language Processing*
XGBoost	*eXtreme Gradient Boosting*
VP	Verdadeiro Positivo
VN	Verdadeiro Negativo
FP	Falso Positivo
FN	False Negativo
EMDT	*Earth Mover's Distance on Trajectory*
EMD	*Earth Mover's Distance*
RNN	*Recurrent Neural Network*
UFC	Universidade Federal do Ceará

SUMÁRIO

1 INTRODUÇÃO

Recentemente, com o crescimento tecnológico em diversas áreas como computação, engenharias, dentre outras, é notável o aumento na disponibilidade de dados de diferentes tipos. Dentre esses dados, percebemos aqueles que vem permitindo o desenvolvimento de métodos modernos que ajudam a entender os padrões da mobilidade urbana, tanto no movimento de pessoas quanto de objetos. Esses dados são os de trajetórias, podendo esses ser do tipo bruto, quando contém somente informações espaço-temporais, ou do tipo semântico, onde são adicionados outros tipos de informações à trajetória, que podem ser textuais, de espaço, entre outros.

Com a popularização de dispositivos e aplicativos baseados em localização, houve um grande aumento de dados de trajetórias, fornecendo diversas outras informações além dos dados de *Global Positioning System* (GPS). Devido a essa explosão tecnológica, *smartphone* e outros dispositivos inteligentes que são capazes de rastrear constantemente a localização dos usuários conectados a internet, tornam possíveis recomendações de rotas, predição de destinos de viagens, além de auxiliar em tomadas de decisões no que diz respeito predições de tráfego, tudo isso graças a algoritmos de predição (BUCHER, 2017).

A descoberta de padrões e a extração de conhecimento das trajetórias contribuem para o desenvolvimento de modelos preditivos precisos, capacitando as cidades para uma melhor gestão, garantindo assim que a vida cotidiana da população melhore (MEHMOOD; PAPAGELIS, 2020). Como resultado, diversas aplicações do mundo real podem ser empregadas (*e.g.* transporte inteligente, planejamento urbano com otimizações de tráfego, dentre outras). A extração de conhecimento sobre o movimento dos indivíduos é de grande interesse para as organizações, sejam elas públicas (*e.g.* Governos Estaduais) ou privadas (*e.g.* aplicativos de recomendação de rotas).

Entretanto, esta vasta quantidade de dados gerada não é suficiente para treinar modelos de predição, pois, esses dados não são gerados de forma uniforme. Algumas classes (destinos) aparecem com uma frequência muito mais elevada do que as demais, prejudicando as previsões dos modelos.

A utilização de dados de sensores de rua para realizar predições de trajetórias é escasso na literatura. Além do trabalho de Cruz *et al.* (2019), são poucos os trabalhos conhecidos que envolvam esse mesmo tipo de dado. Segundo Cruz *et al.* (2019), dado um conjunto de sensores de rua localizados à margem de rodovias, onde cada sensor registra

a passagem de veículos, tendo por hipótese que seja possível identificar cada veículo com exclusividade, é possível construir uma *External Sensor Trajectory* (EST) (Trajetória de Sensores Externos). Como os sensores estão posicionados em localizações pré-definidas, é comum que o movimento completo do objeto não seja rastreado ou até mesmo nem seja capturado quando passar pelo sensor devido alguma falha. Com isso, é gerado um problema de desbalanceamento nas trajetórias.

Portanto, o objetivo deste trabalho consiste em implementar e avaliar técnicas de aumento de dados para reduzir o problema de desbalanceamento dos dados. Além de utilizar diversos modelos preditivos para prever a próxima localização em trajetórias de veículos.

O presente trabalho fez uso da técnica de aumento de dados baseado no método utilizado no trabalho de He *et al.* (2020), onde é construído um grafo de transição a partir das trajetórias conhecidas e então são recuperadas trajetórias não observadas dentro desse conjunto. Além de técnicas de janela deslizante que consiste em inserir partes de uma trajetória dentro de conjunto de dados de trajetórias. Vale ressaltar que a utilização dessa técnica é de fundamental importância para casos onde o conjunto de dados é pequeno, gerando então dados adicionais para o conjunto de treinamento.

As contribuições deste trabalho podem ser resumidas da seguinte forma:

- Proposta de técnicas de aumento de dados com intuito de reduzir o problema de desbalanceamento de dados.
- Desenvolvimento de um grafo de transição com base nos dados de trajetórias existentes para recuperação de trajetórias não observadas para um par origem-destino.
- Implementações voltadas para o processamento de dados de trajetórias e técnicas de aumento de dados para a biblioteca *PyMove*[1], uma biblioteca *Python* para processamento e visualização de trajetórias e outros dados espaço-temporais.

1.1 Objetivos

Nesta seção são apresentados os objetivos geral e específicos deste trabalho.

1.1.1 Geral

Implementar e avaliar técnicas de aumento de dados para trajetórias a fim de reduzir o problema de desbalanceamento dos dados. Além de criar modelos preditivos para avaliar

[1] https://github.com/InsightLab/PyMove

se há associação positiva entre a aplicação dessas técnicas e o desempenho de algoritmos de aprendizado de máquina.

1.1.2 Específicos

- Implementar e avaliar técnicas de aumento de dados;
- Comparar e avaliar diferentes modelos de classificação para trajetórias;
- Avaliar se há associação positiva entre a aplicação dessas técnicas e o desempenho de algoritmos de aprendizado de máquina na predição da próxima localização.

1.2 Organização

O presente trabalho tem a seguinte organização: No Capítulo 2, será apresentada a fundamentação teórica para a compreensão da proposta. Em seguida, no Capítulo 3 serão apresentados os trabalhos relacionados. No Capítulo 4, serão apresentados os procedimentos metodológicos, com a descrição das tarefas relacionadas ao desenvolvimento deste trabalho. No Capítulo 5, são exibidas as soluções propostas para o aumento de dados. No Capítulo 6 são mostrados os experimentos e resultados com base na metodologia adotada. E por fim, no Capítulo 7 são mostradas as conclusões deste trabalho, assim como os trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, são apresentados os conceitos fundamentais utilizados para compreensão desse trabalho.

2.1 Trajetórias de Objetos Móveis

De forma geral, uma trajetória pode ser definida como uma sequência de localizações espaço-temporais ordenadas pelo tempo. Onde cada registro da sequência é uma tupla formada por sua localização e o instante de tempo registrado no momento que a tupla foi gerada (YAO *et al.*, 2017).

Neste trabalho, são utilizados dados de sensores externos que capturam o movimento de objetos dentro de uma rede de ruas. Dessa forma, temos que os objetos em movimento são observados em posições fixas (localização dos sensores).

Segundo Cruz *et al.* (2019), dado um conjunto de sensores externos colocados a margem das rodovias, que registram a passagem de objetos em movimento onde seja possível identificar com exclusividade os objetos em movimento, podemos definir uma trajetória a partir desses sensores denominada **Trajetória de Sensores Externos** ou **EST** do inglês *External Sensor Trajectory*, definida a seguir.

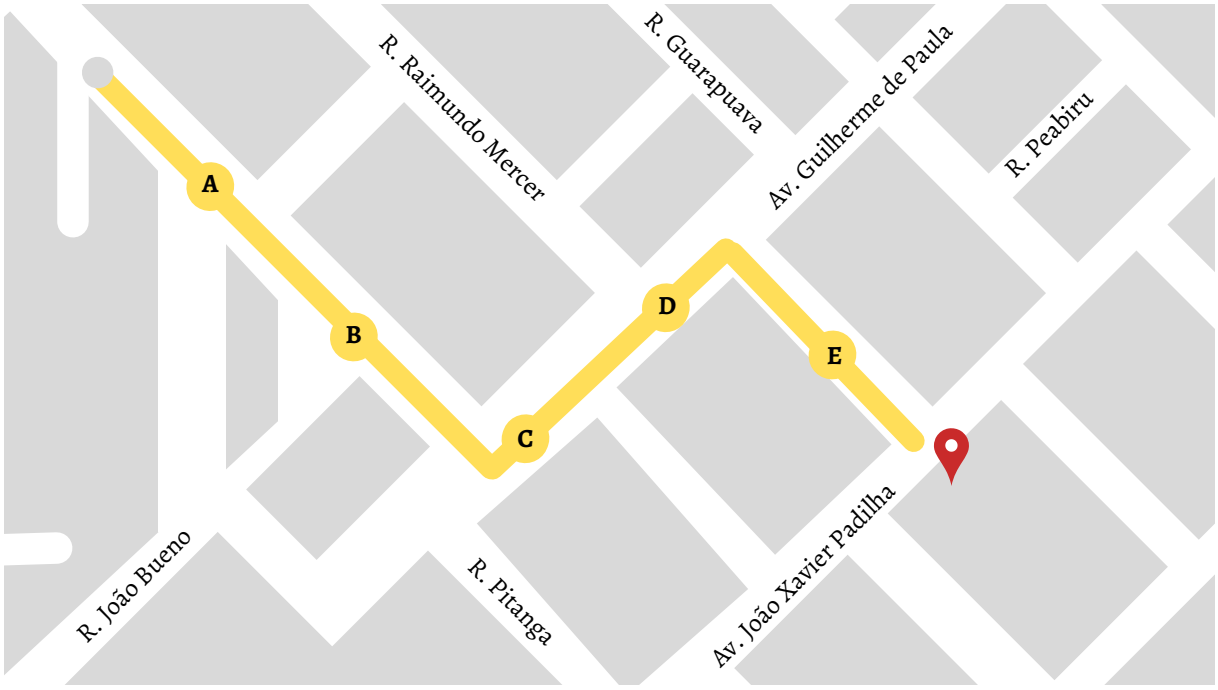
Definição 2.1.1 ***Trajetória de Sensores Externos:*** *Seja O o conjunto de observações geradas por um conjunto de sensores S. Seja $O[m] \subset O$ o conjunto de observações relacionadas ao objeto em movimento m. Uma trajetória pode ser extraída de $O[m]$ como uma sequência de observações $es\_traj_m = \langle o_1, o_2, ..., o_j \rangle$, tal que $\forall_i$, $1 \leq i \leq j$, $o_i = (m, s, t)$, $o_i \in O[m]$ e $t(o_i) \leq t(o_j)$, onde t é o instante de tempo em que o objeto em movimento m foi registrado pelo sensor s.*

Cruz *et al.* (2019) aponta em seu trabalho os seguintes problemas encontrados ao se trabalhar com dados dessa natureza: (i) quantidades de dados em grande escala, pois, a medida que o fluxo de veículos aumenta, também aumenta a complexidade de processar os dados das trajetórias e assim realizar as previsões online; (ii) trajetórias de inúmeros tipos, pois os sensores capturam diferentes tipos de veículos, e cada tipo apresenta um comportamento diferente; (iii) dados esparsos, pois o rastreamento completos dos objetos não é capturado devido os sensores estarem em localizações fixas e; (iv) trajetórias incompletas e incertas, neste caso pode haver falha nos sensores ao registrar o deslocamento do objeto em movimento.

Além dos problemas já mencionados, outra dificuldade encontrada é que os sensores não são visitados com a mesma frequência, apresentando uma quantidade de observações muito maior em alguns do que em outros. Desse modo, predizer a próxima localização a partir de dados de sensores externos torna-se uma tarefa difícil. Com isso, o propósito deste trabalho é propor e aplicar estratégias de aumento de dados em trajetórias EST com o intuito de minimizar o problema de desbalanceamento dos dados com o objetivo de obter um bom resultado para predição de próximo sensor.

A Figura 1 apresenta um exemplo de uma trajetória EST, onde os sensores de rua são representados pelos pontos de *A* até *E*. A trajetória tem início no sensor representado pela cor cinza, onde é realizada a primeira observação, em seguida o sensor *A* registra a passagem desse mesmo objeto, continuando o percurso até encerrar o rastreamento no ponto com o marcador em vermelho.

Figura 1 – Exemplo de uma Trajetória EST.



Fonte: Elaborado pelo autor (2021).

2.2 Desbalanceamento de Dados

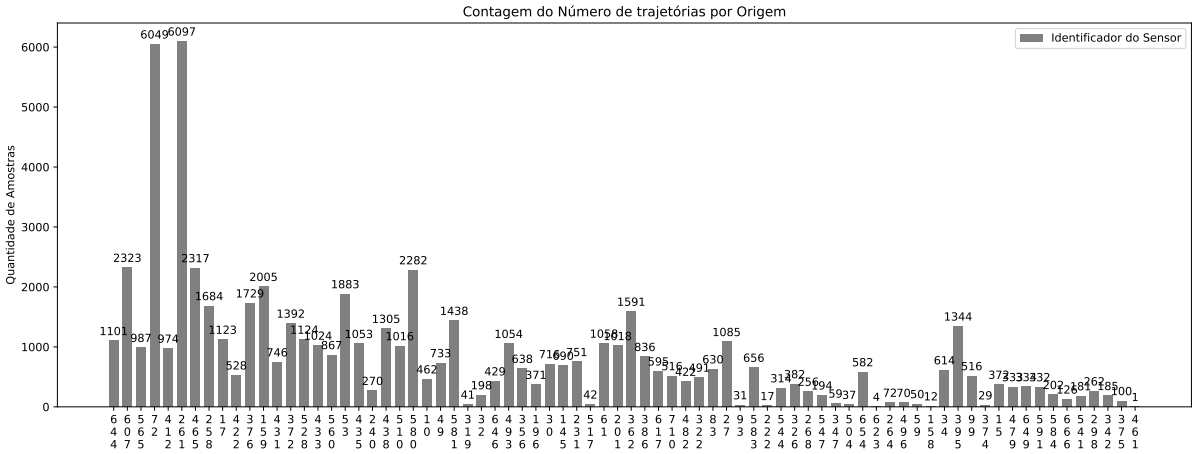
Atualmente, com o enorme aumento de dados devido o rápido processo de evolução tecnológica nas mais diversas áreas, é proposta a possibilidade de investigar aspectos da vida humana. Entretanto, esses dados não são gerados de maneira uniforme, com isso é comum

encontrar conjuntos de dados com distribuição de classes desbalanceadas.

Ao se deparar com essas situações de desbalanceamento, os modelos costumam fornecer resultados de classificação de forma enganosa, ou seja, a maioria dos exemplos tem boa cobertura, enquanto algumas amostras são distorcidas (LÓPEZ *et al.*, 2013). Haixiang *et al.* (2017) apontam estratégias básicas para lidar com a aprendizagem desbalanceada, uma delas é a estratégia de pré-processamento, no qual é executada com frequência a fim de se obter melhores dados de entrada para o modelo de aprendizagem. Nessa estratégia são incluídas técnicas de reamostragem, a fim de moderar o efeito da distribuição desigual das classes, equilibrando o espaço de amostra do conjunto de dados desbalanceado. A técnica de reamostragem pode ser executada de diferentes maneiras, sendo as mais conhecidas a sobreamostragem, que consiste em criar amostras não observadas para as classes minoritárias e a subamostragem, que consiste em reduzir o espaço amostral das classes majoritárias.

As Figuras 2 e 3 mostram as distribuições da frequência de origens e destinos nas trajetórias observadas, que totalizam 63.351 registros de trajetórias no período de 01 à 06 de setembro de 2017. Para o conjunto de dados utilizados nesse trabalho é possível observar que a frequência das origens e dos destinos, ou seja, os sensores inicial e final apresentam valores totalmente desbalanceados, tanto olhando para as origens, quanto olhando para os destinos.

Figura 2 – Contagem do número de trajetórias por origem.

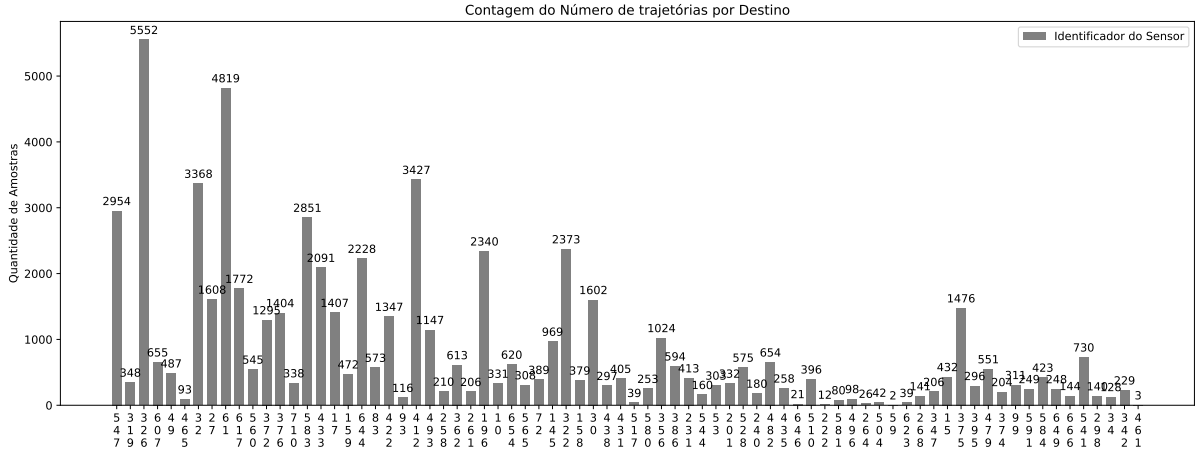


Fonte: Elaborado pelo autor (2021).

2.3 Aumento de Dados

O termo aumento de dados se refere a métodos para construção de otimização iterativa ou algoritmos de amostragem por meio da introdução de dados não observados ou

Figura 3 – Contagem do número de trajetórias por destino.



Fonte: Elaborado pelo autor (2021).

variáveis latentes (DYK; MENG, 2001). Esta é uma técnica usada para aumentar a quantidade de dados, gerando cópias parcialmente modificadas dos dados já existentes.

O aumento de dados pode ser aplicado em vários domínios, frequentemente usado em visão computacional para processamento de imagens, onde algumas das técnicas usadas são: redimensionamento, recorte, preenchimento, rotação, entre outros (SHORTEN; KHOSHGOFTAAR, 2019). Outra aplicação em que o aumento de dados vem sendo empregado é para *Natural Language Processing* (NLP) (SAG *et al.*, 2002), no qual são usadas técnicas de substituição lexical, retrotradução, transformação da superfície do texto, aumento de crossover de instância e manipulação da árvore de sintaxe são algumas das técnicas usadas em. Um exemplo da aplicação de aumento de dados em NLP é descrito no trabalho de Luque (2019), que utiliza a técnica de aumento de crossover de instância em *tweets*. Esta é uma técnica utilizada para análise de sentimento, no qual duas sentenças de mesma polaridade são divididas ao meio e tem suas metades trocadas.

Uma técnica de aumento de dados para trajetórias foi proposta por He *et al.* (2020), que consiste em gerar trajetórias não observadas a partir das transições de um grafo. Baseado nesta técnica, foi implementada uma versão similar que utiliza um grafo de transição como suporte para recuperação das trajetórias.

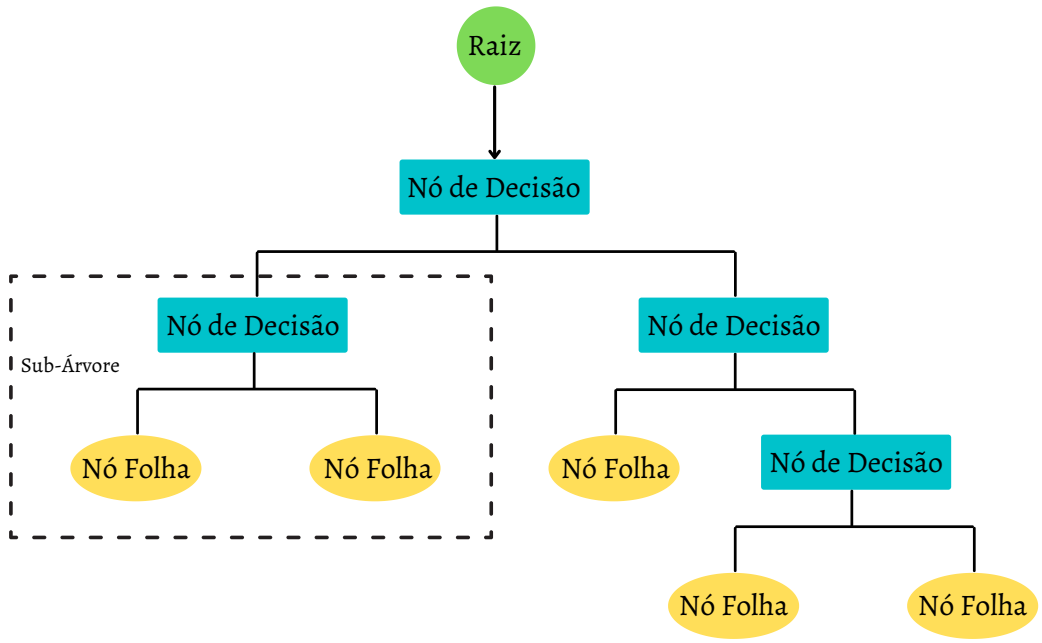
2.4 Modelos de Predição

Nesta seção são apresentados os modelos de predição utilizados para a realização deste trabalho.

2.4.1 Árvore de Decisão

O algoritmo Árvore de Decisão ou DT do inglês *Decision Tree* é uma técnica de predição que consiste em dividir uma decisão complexa na união de várias decisões mais simples, esperando que a solução final obtida dessa forma se pareça com a solução desejada pretendida (SAFAVIAN; LANDGREBE, 1991). A estrutura do algoritmo consiste em nós de decisão e nós folhas. Os nós de decisão, como o próprio nome já diz, são utilizados para tomar a decisão do próximo nó a ser verificado em função do valor de um atributo da amostra, enquanto que o nó folha é o último nó de um ramo, não apresentando ramificações a partir deles. Cada nó folha é a decisão final para a classe, com base em seus nós de decisão antecedentes. Para cada decisão tomada (SIM/NÃO), o algoritmo divide a árvore em sub-árvores, e essas decisões são com base nos atributos de dados fornecidos. A Figura 4 explica a estrutura geral do algoritmo *Decision Tree*.

Figura 4 – Estrutura geral do algoritmo *Decision Tree*



Fonte: Navlani (2018) (Adaptado).

Assim, como exibido na Figura 4, para que um modelo de árvore de decisão realize a predição da classe de forma eficiente, o mesmo utiliza de perguntas baseadas nos atributos do conjunto de dados para aprender, onde cada pergunta é baseada no valor de um único atributo distinto para cada nó de decisão.

2.4.2 Random Forest

O algoritmo *Random Forest* (RF) é construído sobre um algoritmo mais básico, a Árvore de Decisão (HO, 1995). O RF trabalha pela média dos resultados de muitos estimadores individuais que superajustam os dados (GOEL *et al.*, 2017), tal que cada estimador é uma árvore de decisão. Este método de conjunto minimiza a variação de predição que, por sua vez, melhora a precisão. Breiman (2001) descreve RF como uma combinação de classificadores de árvore, sendo que cada árvore depende dos valores de um vetor aleatório de amostra de forma independente e com a mesma distribuição para todas as árvores. Breiman (2001) define como mostrado na definição 2.4.1.

Definição 2.4.1 ***Random Forest:*** *O algoritmo Random Forest é um classificador que consiste em uma coleção de árvores estruturadas* $\{h(\boldsymbol{x}, \Theta_k), k = 1, ...\}$*, onde os* $\{\Theta_k\}$ *são vetores aleatórios independentes e identicamente distribuídos e cada árvore lança um voto de unidade para a classe mais popular na entrada* $\boldsymbol{x}$*.*

A Figura 5 mostra a estrutura do algoritmo de classificação, destacando o fluxo de cada árvore aleatória. O próprio modelo adiciona aleatoriedade extra quando está criando as árvores. Cada árvore gerada retorna um voto resultante para o modelo. Assim, o algoritmo seleciona o resultado da predição mais votado como o valor predito pelo classificador.

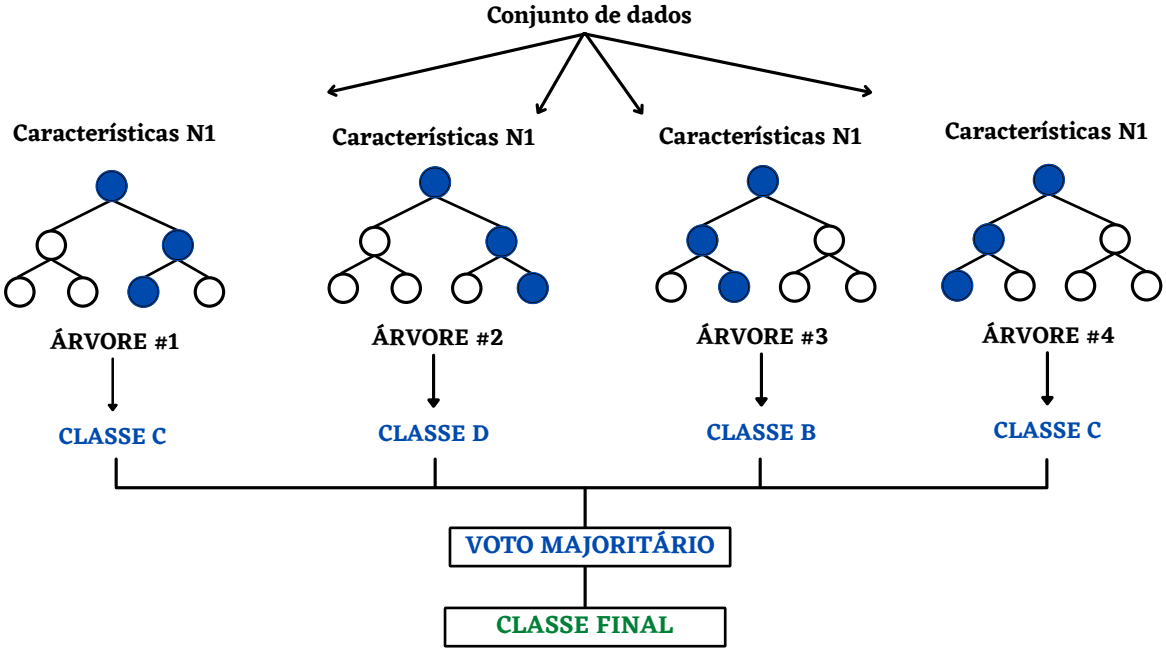
2.4.3 eXtreme Gradiente Boosting (XGBoost)

O algoritmo *eXtreme Gradient Boosting* (XGBoost) é uma técnica de *ensemble* que utiliza o método de *boosting*. O conceito de *ensemble* caracteriza a combinação de um conjunto de modelos de predição mais simples. O método de *boosting* significa utilizar o algoritmo Gradiente Descentente (BOTTOU, 2012) para minimizar a perda para modelos posteriomente adicionados. Segundo Géron (2017), a técnica de combinar um conjunto de modelos resulta em predições melhores do que o resultado do melhor modelo dentre o conjunto de modelos utilizados.

2.5 Métricas de Classificação

Para avaliar modelos de classificação, são utilizadas métricas, sendo as mais conhecidas Acurácia, Precisão, Revocação e *F1-Score* (SOKOLOVA *et al.*, 2006). Além das métricas já

Figura 5 – Estrutura do algoritmo de classificação *Random Forest*



Fonte: Abilash (2018) (Adaptado).

citadas, existe a Acurácia Balanceada que é melhor para usar com dados desbalanceados. Uma métrica de avaliação tem a tarefa de medir quão boas foram as previsões do modelo criado. Para calcular as métricas citadas, são usados quatro parâmetros, que são: Verdadeiro Positivo (VP); Verdadeiro Negativo (VN); Falso Positivo (FP) e False Negativo (FN). Cada termo é descrito a seguir.

- os termos VP e VN de uma classe significam que os valores previstos são os mesmos dos valores reais.
- os termos FP e FN de uma classe significam que os valores previstos são diferentes dos valores reais.

Por exemplo, em quadros médicos com dois grupos de pacientes: doentes e não doentes, um resultado é VP ao diagnosticar um paciente doente de forma positiva. Entretanto, se esse diagnóstico for negativo, significa que o resultado foi um FN. Já no caso de um diagnóstico negativo em um paciente não doente, o resultado foi um VN. Enquanto que se o diagnóstico for positivo, foi um FP.

Existem diferentes tipos de modelos de classificação, e para cada modelo uma métrica tem um melhor desempenho. A seguir é descrito com detalhes as métricas que serão utilizadas neste trabalho.

A métrica Acurácia representa a proporção entre a quantidade de valores preditos

de forma correta e a quantidade total de instâncias no conjunto de dados (CHICCO; JURMAN, 2020). O cálculo para saber seu valor é detalhado na equação 2.1.

$$acurcia = \frac{VP + VN}{VP + VN + FP + FN} \tag{2.1}$$

Para dados com classes desequilibradas, a pontuação de acerto pode enganar. Isso, devido o classificador ponderar de forma errada para esses dados, alcançando trivialmente uma pontuação de classificação igual à pontuação da classe majoritária (CARRILLO *et al.*, 2014). A Acurácia Balanceada é melhor empregada para avaliar o desempenho nesses casos, pois, calcula a média entre a taxa de verdadeiro positivo (sensibilidade) e verdadeiro negativo (especificidade). Na equação 2.2 pode ser observado o cálculo.

$$acurcia\_balanceada = \frac{sensibilidade + especificidade}{2} \tag{2.2}$$

$$sensibilidade = \frac{VP}{VP + FN} \tag{2.3}$$

$$especificidade = \frac{VN}{VN + FP} \tag{2.4}$$

No caso de dados de trajetórias, onde cada trajetória é uma sequência de pontos espaço-temporais ordenados pelo tempo, a próxima localização pode ser considerada como a classe a ser predita.

3 TRABALHOS RELACIONADOS

Este capítulo descreve os principais trabalhos relacionados para o desenvolvimento do presente trabalho.

3.1 Efficient and robust data augmentation for trajectory analytics: similarity-based approach

No trabalho de He *et al.* (2020) é proposta uma abordagem de aumento de dados de trajetória a partir dos dados existentes a fim de resolver o problema da esparsidade. Fundada sobre a ideia de concatenar as trajetórias existentes para reconstruir um número suficiente de trajetórias para representar aquelas que atravessam o par origem-destino diretamente. Para atingir esse objetivo, He *et al.* (2020) propuseram um grafo de transição para dar suporte à concatenação das sub-trajetórias de forma eficiente.

Como forma de validação das trajetórias concatenadas, uma medida de similaridade própria para avaliar o conjunto de trajetórias foi aplicado por He *et al.* (2020) em seu trabalho. A mesma foi comparada com outras duas estratégias existentes: *sweep-and-expand* (DAI *et al.*, 2015) e *popularity-based* (CHEN *et al.*, 2011).

He *et al.* (2020) desenvolveram a medida de similaridade *Earth Mover's Distance on Trajectory* (EMDT), uma versão aprimorada do *Earth Mover's Distance* (EMD), que mede a distância da distribuição espacial de impactos entre dois conjuntos de trajetórias, sendo estendidas para cobrir propriedades espaço-temporais e de sequência de conjuntos de trajetórias.

Semelhante ao trabalho de He *et al.* (2020), o presente trabalho visa reduzir o problema da esparsidade utilizando como suporte diferentes maneiras de aumento de dados, além da proposta técnicas de predição de próxima localização com base na sequencia antecedente.

3.2 Trajectory Prediction from a Mass of Sparse and Missing External Sensor Data

No trabalho de Cruz *et al.* (2019), foram propostas técnicas para a predição de próxima localização a partir de dados de sensores externos (EST). Dados estes muitas vezes incompletos e escassos, por decorrência de motivos como falha do sensor ao capturar a passagem de um veículo, produzindo assim, trajetórias incompletas. Além disso, este tipo de trajetória pode apresentar padrões de mobilidade muito diferentes, pois não se restringem a uma frota ou a uma comunidade de usuários.

Cruz *et al.* (2019) propõem uma abordagem para lidar com dados ausentes, utilizando dados do mundo real, escassos e incompletos. Por estas razões, os autores pensaram em quais modelos preditivos seriam melhor empregados, chegando a conclusão de que um conjunto de preditores baseados em *Recurrent Neural Network* (RNN).

Como os dados são escassos e incompletos, é aceitável quando o modelo não prevê corretamente o próximo local. Devido a isso, as estratégias de imputação foram avaliadas usando a abordagem de simulação livre. Também foi avaliada a qualidade do método medindo a proximidade do valor observado ao valor predito. Contudo, os resultados obtidos por Cruz *et al.* (2019) mostraram que o método utilizado poderia aumentar a precisão em cerca de 23%.

Assim como o trabalho de Cruz *et al.* (2019), o presente trabalho apresenta técnicas para predição de próxima localização a partir de dados de sensores externos, incluindo também técnicas de aumento de dados para reduzir o problema da esparsidade de dados. Além da utilização de diferentes algoritmos de aprendizado de máquina para predição.

3.3 SERM: *A Recurrent Model for Next Location Prediction in Semantic Trajectories*

No trabalho de Yao *et al.* (2017) é proposto um modelo recorrente de semântica enriquecida, conhecido por seu termo em inglês *Semantic Enriched Recurrent Model* (SERM). O mesmo utiliza vários fatores (usuário, local, horário e palavra-chave) que captura regularidades de transição espaço-temporal com reconhecimento de semântica para melhorar as precisões de previsão de localização. Os experimentos realizados foram baseados em dois conjuntos de dados de trajetória semântica em duas cidades: Nova Iorque e Los Angeles. O primeiro conjunto de dados, de Nova Iorque, consiste em 0,3 milhão de *check-in* do *Foursquare* de Janeiro de 2011 a Janeiro de 2012. O segundo conjunto de dados, de Los Angeles, consiste em 1,4 milhão de *tweets* de Agosto de 2014 a Novembro de 2014.

Com base nos resultados alçandos, Yao *et al.* (2017) concluiram que ao capturar as intenções das atividades de um usuário após incluir informações semânticas no processo de modelagem é evidenciado uma maior acurácia, mostrando valores com melhorias significativas em relação ao métodos comparados. Diferente do trabalho de Yao *et al.* (2017), o presente trabalho utiliza apenas a informação espacial para fazer as predições, incluindo técnicas de aumento de dados, a fim de melhorar a predição de próxima localização.

3.4 Comparação de trabalhos

O Quadro 1 traz as semelhanças e diferenças entre os trabalhos relacionados e este trabalho, de acordo com as suas principais características. Todos, com exceção do trabalho de He *et al.* (2020) possuem o objetivo de realizar predições de próxima localização, mas com diferentes abordagens, como o tipo de dado utilizado as técnicas de predição. Além disso, o presente trabalho faz uso de técnicas de aumento de dados utilizando grafo de transição, semelhante ao método utilizado por He *et al.* (2020), o que não ocorre nos demais trabalhos.

Quadro 1 – Comparação entre os trabalhos relacionados e o proposto

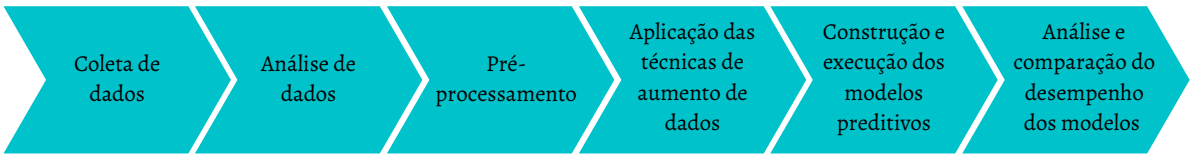
	He *et al.* (2020)	**Cruz *et al.* (2019)**	**Yao *et al.* (2017)**	**Trabalho Atual**
Tipo de Trajetória	GPS	EST	Semântica	EST
Técnica	EMDT	Modelo RNN	Modelo RNN	Diversas
Aumento de dados?	Sim	Não	Não	Sim
Objetivo	Imputação de trajetórias	Predição de Próximo Local	Predição de Próximo Local	Predição de Próximo Local

Fonte: Elaborado pelo autor (2021).

4 METODOLOGIA

Neste capítulo são descritos os procedimentos metodológicos para a realização deste trabalho. A Figura 6 apresenta a visão geral dos passos realizados no decorrer deste capítulo.

Figura 6 – Procedimentos metodológicos



Fonte: Elaborado pelo autor (2021).

4.1 Coleta dos dados

Os dados utilizados para a realização deste trabalho são de um dos projetos de pesquisa e desenvolvimento de uma parceria entre Universidade Federal do Ceará (UFC) e Senasp. Os dados consistem em registros de passagens de veículos anonimizados (sem identificação original do veículo) dentro da cidade de Fortaleza - CE.

4.2 Análise dos dados

Nesta etapa, são realizados procedimentos de análise a fim de compreender a estrutura e organização dos dados, visando identificar padrões nas trajetórias que possam ser utilizados. Mais adiante, será feita uma varredura buscando valores nulos para sejam tratados na etapa seguinte. Assim, será possível saber qual a melhor técnica empregada e qual algoritmo mais eficiente para a solução do problema de prever a próxima localização.

4.3 Pré-processamento dos dados

A etapa de pré-processamento se faz importante, pois, é nessa etapa que adequamos os dados para que possam ser passados como entrada para os modelos de predição. Também

é nesta etapa que fazemos o tratamento de dados nulos dentro do conjunto de dados, para que novos atributos possam ser gerados. Fazer o pré-processamento dos dados é fundamental e pode beneficiar positivamente o desempenho do modelo (TAN *et al.*, 2016).

4.4 Aplicação das técnicas de aumento de dados

Conjuntos de dados esparsos já mostrou-se um problema recorrente na atualidade. Tratar esse tipo de dado com técnicas que reduzem o problema da esparsidade e do desbalanceamento de classes tem ganhado cada vez mais espaço, com intuito de se elaborar modelos de predição melhores e mais robustos. Com isso, este trabalho propõe técnicas para o aumento de dados.

4.4.1 Aumento de dados com janela deslizante

Neste passo será aplicada a técnica de Janela Deslizante voltada para sequência de localizações. A técnica de janela deslizante é uma técnica já conhecida de outros contextos, como por exemplo em protocolos de redes de computadores. Para o contexto atual, o método abordado consiste em inserir no conjunto de dados, trechos das trajetórias reais, aumentando os dados com trajetórias válidas, pois as mesmas são sub-trajetórias das trajetórias existentes.

4.4.2 Aumento de dados com grafo de transição

Aqui, uma outra técnica de aumento de dados é aplicada. A técnica em questão é baseada na técnica proposta por He *et al.* (2020), que utiliza um grafo de transição para dar suporte a recuperação das trajetórias não observadas para um par origem-destino especificado, mas conservando a validade das transições por utilizar um grafo construído a partir dos dados reais.

Definição 4.4.1 ***Grafo de Transição:*** *Um Grafo de Transição consiste em um conjunto de vértices e arestas tal que* $TG = (V, E)$*. Os vértices do grafo são definidos pelos identificadores das labels de localizações, e as arestas são definidas pela transição entre um par de localizações* (l_i, l_j)*, tal que exista uma transição direta de* l_i *para* l_j *no conjunto de trajetórias recuperado a partir dos dados brutos.*

4.4.3 Balanceamento dos dados

Como são utilizados dados de sensores de rua, uma característica comum é que os sensores não são frequentados com a mesma frequência. Com classes desbalanceadas, modelos de predição treinados a partir desses dados podem apresentar uma acurácia enviesada para as classes majoritárias e mascarar a real acurácia obtida. Assim, balancear a quantidade de trajetórias para aqueles sensores que são origens e destinos traz uma possível melhoria na qualidade da predição.

4.5 Construção e execução dos modelos preditivos

Neste etapa, é feita a construção dos modelos de classificação utilizados neste trabalho. Diversos modelos preditivos serão construídos utilizando cada um dos conjuntos de dados obtidos a partir das técnicas aplicadas na etapa anterior.

4.6 Análise e comparação do desempenho dos modelos

Por fim, será avaliado o desempenho de classificação para cada um dos modelos utilizados com as diferentes técnicas de aumento de dados. Será realizado um comparativo entre os resultados dos dados balanceados e não balanceados a fim de levantar hipótese acerca da eficiência das técnicas propostas pra aumento de dados, e tentar entender qual a razão de alguns resultados se sobressaírem sobre outros.

5 TÉCNICAS DE AUMENTO DE DADOS

Neste capítulo, são apresentadas as soluções propostas para a redução do desbalanceamento de dados, explicando com detalhes a construção de cada técnica implementada.

5.1 Problema do desbalanceamento de dados

Devido o tipo de dado usado, foram identificadas algumas dificuldades já citadas anteriormente. Um deles é o desbalanceamento, causada por os sensores não capturarem com a mesma frequência os registros de passagem de veículos, pois em alguns pontos o fluxo é mais intenso. Com isso, o deslocamento completo de um objeto em movimento acaba não sendo rastreado, além do fato de que algum sensor pode não capturar a passagem do veículo, aumentando ainda mais o problema do desbalanceamento dos dados.

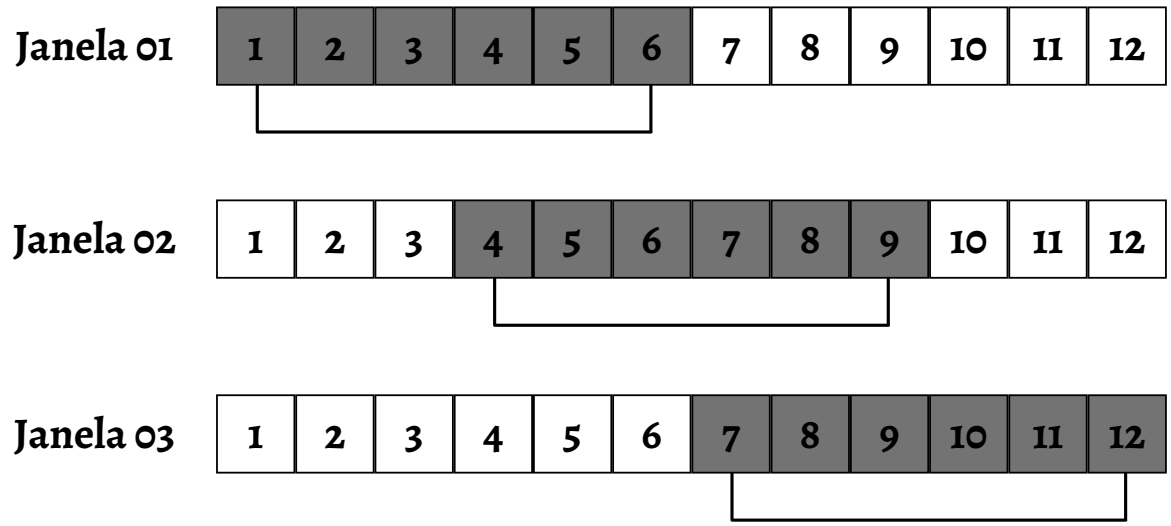
Para tal problema foram propostas duas abordagens: a primeira recebe o nome de Janela Deslizante, que como o próprio nome já diz, é definido um tamanho para a janela que desliza sobre a sequência de pontos, adicionando trechos da trajetória no conjunto de dados original. A segunda abordagem é uma técnica baseada em Grafos, onde um grafo de transição é construído a partir dos dados originais, assim é possível manter a veracidade dos dados, já que cada sensor vira um nó no grafo e as arestas são definidas pelas transições reais entre os sensores.

5.2 Aumento de dados com janela deslizante

Esta técnica consiste em deslizar sobre a sequência de localizações, pegando trechos da sequência e adicionando no conjunto de dados original. Para a aplicação desta técnica, são setados o tamanho da janela e o tamanho do salto. Onde o tamanho da janela representa a quantidade de pontos que será copiada da sequência e adicionada no conjunto, e o tamanho do salto é a quantidade de pontos que serão pulados até o início da próxima janela selecionada, pois definir um valor para o comprimento de salto é importante para evitar a geração exaustiva de dados. Neste trabalho, foram utilizados 6 e 3 como tamanho da janela e tamanho do salto, respectivamente. A escolha do valor 6 para o tamanho da janela foi devido a média do tamanho das trajetórias, e valor do salto foi o meio termo entre gerar o máximo de dados e não gerar dados intermediários.

A Figura 7 ilustra o funcionamento da aplicação da técnica de janela deslizante.

Figura 7 – Ilustração da aplicação da técnica de Janela Deslizante



Fonte: Elaborado pelo autor (2021).

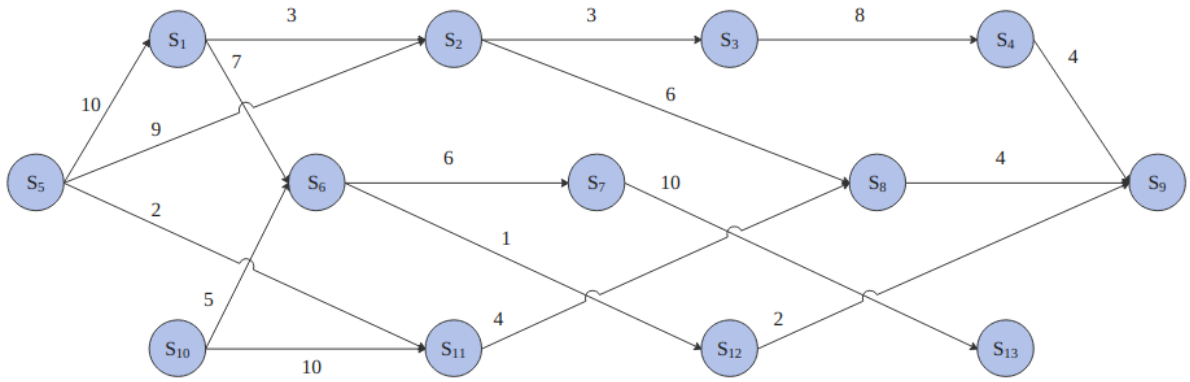
5.3 Aumento de dados com grafo de transição

Baseado na técnica usada no trabalho de He *et al.* (2020), este trabalho também faz uso de um grafo de transição para recuperar trajetórias não observadas. O grafo de transição em questão foi construído utilizando os dados das trajetórias reais, com isso, temos a garantia da validade das trajetórias recuperadas, pois, cada transição entre dois nós no grafo corresponde a uma transição real entre dois sensores de rua. Para cada transição foi atribuído um peso referente a quantidade de vezes que a transição foi percorrida no conjunto de dados de trajetórias.

A atribuição dos pesos nas transições proporcionou a construção de caminhos no grafo de transição de duas formas distintas. A primeira consistiu em recuperar as trajetórias priorizando as transições mais usadas, com os maiores pesos. Já a segunda, consistiu em priorizar as transições menos usadas, com os menores pesos. Cada uma dessas abordagens balanceia os conjunto de dados de forma diferente, assim, são gerados dois novos conjuntos de dados de trajetórias que podem impactar o desempenho dos modelos positivamente, ao reduzir o desbalanceamento dos dados.

De acordo com a Figura 8, podemos perceber que existe uma quantidade de transições de saída e de entrada em cada nó que varia de 0 à N, onde $N \in \mathbb{Z}_{+}^{*}$ (0 se não existir transições). Com isso, é possível que ao aplicar a técnica de sobreamostragem, não seja possível balancear a quantidade de trajetórias para cada *label*, seja ela de origem ou destino, pois, o número de

Figura 8 – Ilustração do grafo de transição construído



Fonte: Elaborado pelo autor (2021).

trajetórias recuperadas é inferior a quantidade desejada para balancear suas *labels*.

6 EXPERIMENTOS E RESULTADOS

Neste capítulo são apresentados os experimentos realizados ao longo deste trabalho, bem como os resultados obtidos. Para execução dos experimentos, foram utilizadas as bibliotecas: (i) *scikit-Learn*[1] para a utilização dos classificadores propostos e para as métricas de avaliação de desempenho; e (ii) *PyMove*[2] para algumas pré-análises e pré-processamento dos dados de trajetória. Também foram implementadas todas as técnicas propostas dentro do escopo deste trabalho, além integrar essas implementações a biblioteca *PyMove*.

6.1 Configuração do Ambiente

Todos os experimentos foram realizados em um computador com processador Intel Core i5 10a geração 1.6 GHz, 8GB de memória RAM e sistema operacional Ubuntu 20.04 LTS. As técnicas foram implementadas utilizando a linguagem Python.

6.2 Coleta dos dados

Os dados utilizados para a realização deste trabalho são do projeto `Senasp Big Data de Inteligência Artificial para Segurança Pública`, que consistem em dados de trajetórias de veículos anonimizados (ou seja, sem identificação original do veículo) dentro da cidade de Fortaleza - CE. Esses dados são originados dos sensores de rua, que capturam a passagem de vários tipos de veículos como carros de passeio, caminhões, motocicletas, dentre outros. Por se tratarem de sensores com posições pré-definidas (fixas), o movimento completo do objeto em movimento não é capturado, causando um problema de esparsidade. Além disso, algum sensor pode falhar ao capturar a passagem do objeto, ocasionando irregularidade das trajetórias, ou seja, uma mesma trajetória pode gerar diferentes registros (CRUZ *et al.*, 2019). Outro aspecto comum neste tipo de dado é que a frequência com que cada sensor captura a passagem de objeto em movimento não é uniforme, gerando dados desbalanceados, pois a distribuição do número de trajetória para as origens e os destinos não são iguais.

Os dados de trajetória são do período de 01 a 06 de setembro de 2017, onde foram registradas 705.920 leituras para um total de 76 sensores analisados e 124.149 veículos. As informações acopladas aos dados são (i) *datetime* - instante de tempo que o veículo passa

[1] https://scikit-learn.org/stable/
[2] https://github.com/InsightLab/PyMove

pelo sensor; (ii) *vehicle* - identificador do veículo; (iii) *equ_id* - identificador do equipamento de registro; (iv) *equ_lat* - latitude do equipamento de registro e; (v) *equ_lon* - longitude do equipamento de registro. A Tabela 1 apresenta a descrição das informações capturadas pelos equipamentos e salvas no conjunto de dados.

Tabela 1 – Informações dos dados obtidos pelos sensores de rua

Atributo	**Tipo**	**Descrição**
datetime	*Texto*	Data e hora exata em que o veículo passa pelo equipamento
vehicle	*Texto*	Identificador do veículo que passou pelo equipamento
equ_id	*Numérico*	Identificador do equipamento de registro
equ_latitude	*Numérico*	Localização latitudinal do equipamento de registro
equ_longitude	*Numérico*	Localização longitudinal do equipamento de registro

Fonte: Elaborado pelo autor (2020).

6.3 Análise e Pré-processamento dos dados

A primeira observação sobre os dados é que não havia um identificador que separasse as trajetórias umas das outras, com isso o primeiro passo realizado foi gerar um atributo com valor de *ID* para cada trajetória no conjunto de dados. Gerar um atributo para o conjunto de dados, consiste no processo de utilizar a informação conhecida em um domínio de aplicação para criar novos atributos. Com auxílio da biblioteca *PyMove* foi gerado uma coluna para identificar de modo único cada trajetória. A estratégia utilizada para definir um *id* para a trajetória foi a estratégia baseada em estatísticas, onde as trajetórias são divididas em segmentos com base nas estatísticas de tempo para cada segmento. Esse atributo recebe alguns parâmetros necessários para sua geração, são eles: (i) *mean_coef* - coeficiente de multiplicação do tempo médio para o segmento, por padrão 1.0 e o (ii) *std_coef* - coeficiente de multiplicação do tempo de desvio padrão para o segmento, por padrão 1.0. Com isso, é realizado o cálculo estatístico de tempo (média, desvio padrão, mínimo, máximo, soma e contagem) das *labels* locais de pares de uma trajetória. É importante observar que as trajetórias podem apresentar intervalos muito espaçados de tempo, por este motivo é calculado e criado um valor de limite (*threshold*) para cada segmento.

Realizado o passo inicial de geração do atributo *tid_stat*, é necessário fazer a extração dos dados referentes às sequências de sensores que representam as trajetórias dos objetos em movimento. Para extrair as trajetória, foi implementada uma função que gera o *DataFrame* das trajetórias a partir dos dados de registros dos sensores. A Tabela 2 ilustra como ficaram os dados após a aplicação da função desenvolvida.

Tabela 2 – Trajetórias extraídas com base no *tid* estatístico.

datetime	vehicle	trajectory	lat	lon	tid_stat
[2017-09-01 02:50:23, ...]	vehicle_01	[11, 32, 24, 25]	[-3.12345, ...]	[-38.12345, ...]	1
[2017-09-01 04:55:12, ...]	vehicle_02	[60, 17, 18, 29, 15]	[-3.61234, ...]	[-38.61234, ...]	2
[2017-09-01 08:25:52, ...]	vehicle_03	[10, 11, 12, 13, 14, 15, 20]	[-3.76123, ...]	[-38.76123, ...]	3
[2017-09-01 12:44:16, ...]	vehicle_02	[16, 17, 5, 19]	[-3.76123, ...]	[-38.76123, ...]	4
[2017-09-01 17:44:16, ...]	vehicle_04	[20, 21, 22, 23, 24]	[-3.87612, ...]	[-38.87612, ...]	5

Fonte: Elaborado pelo autor (2021).

Como mostrado na Tabela 2, os dados obtidos consistem na sequência de localizações, representadas pelo *id* do equipamento, no qual o veículo percorre de forma temporal. É importante ressaltar que, no momento da geração desse novo *DataFrame*, trajetórias com tamanho inferior a 3 localizações não entraram, pois acabaria prejudicando o treinamento do modelo de classificação por conter um tamanho tão pequeno. Dado que as sequências de localizações que representam as trajetórias foram geradas, o modelo de predição deve ser capaz de aprender com os pontos sequências que precedem o último ponto conhecido da trajetória, pois esse último ponto será o valor a ser predito pelo modelo.

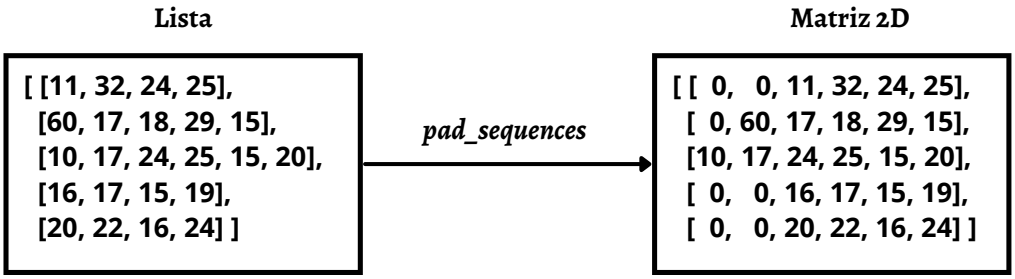
Como pode ser observado, as trajetórias possuem tamanhos diferentes. Com isso, foi necessário utilizar uma técnica de *pad_sequences* que transforma uma lista de listas em uma matriz 2D, preenchendo posições das listas menores que a maior lista com valores nulos no seu início ou fim (CHOLLET, 2018). Como a intenção é prever a próxima localização, não é interessante preencher as sequencias com valores nulos no fim, logo, o preenchimento deu-se no início das listas. Essa implementação está disponível na biblioteca *Keras*. Esta função foi utilizada porque todos os modelos de aprendizado de máquina obrigarem que as entradas tenham o mesmo tamanho.

Neste experimento foi fixado o tamanho das trajetórias em 6, onde o sexto ponto será considerado o *label* da mesma, com isso as trajetórias selecionadas terão tamanho variando de 3 à 6 da mesma maneira que foi escolhida para a técnica de janela deslizante, detalhada na seção 5.2. Na Figura 9 pode ser observado como ficam os dados após a aplicação do método de *pad_sequences*.

6.4 Aplicação das técnicas de aumento de dados

As técnicas de aumento de dados possibilitaram a realização de diferentes testes com combinações das mesmas. Foram gerados 7 diferentes conjuntos de dados a partir das técnicas implementadas ou de suas combinações. De início foram geradas as soluções com

Figura 9 – Ilustração da aplicação da técnica de *pad_sequences*



Fonte: Elaborado pelo autor (2021).

técnicas individuas, uma solução com janela deslizante e outra com grafo de transição, sendo que o tipo de transição (trechos menos frequentes e trechos mais frequentes) caracterizam duas combinações diferentes. Em seguida, foram geradas as soluções compostas pela união das técnicas. Vale ressaltar que a ordem com que as técnicas foram combinadas pesa na hora de treinar os modelos. Em seguida, foram geradas soluções a partir da combinação das técnicas utilizadas neste trabalho.

Desse modo, foi possível levantar hipótese sobre o impacto de cada técnica no conjunto de dados. Observando com mais atenção aquelas que geraram um resultado mais significativo, além daquelas que geraram ruídos na geração de dados.

6.5 Balanceamento dos dados

Partindo da premissa que temos um conjunto de dados desbalanceado, foi proposto implementar alguma técnica a fim de reduzir esse desbalanceamento ou até mesmo balancear por completo os dados. Das técnicas para balanceamento até então conhecidas (*e.g* SMOTE (CHAWLA *et al.*, 2002)), não são apropriadas para dados de trajetórias, pois as sequências geradas não apresentam nenhuma garantia da validade das trajetórias geradas. Para o tipo de dado em questão, de trajetórias, onde cada *id* de sensor representa uma localização real, é preciso conservar algumas características importantes, como as transições entre os pares de sensores, pois, cada sensor em questão possui transições válidas ou não possuem transições. Sabendo disso, foi implementado uma função de sobreamostragem que trate esses dados de forma mais eficiente.

A técnica de sobreamostragem implementada para reduzir o desbalanceamento dos destinos das trajetórias foi construída da seguinte forma: primeiro foi feita uma contagem

Algoritmo 1: Transition Graph Augmentation

```
Entrada: Conjunto de dados de trajetórias
Saída: Conjunto de dados balanceado
início
    Construa o grafo de transição;
    Obtenha todas as origens de trajetórias;
    Obtenha todos os destinos de trajetórias;
    para cada par origem-destino faça
        Encontre todos caminhos menores de origem para destino no grafo;
        para cada caminho encontrado faça
            se origem balanceada
                Finalize a iteração;
            fim
            se destino balanceado
                Finalize a iteração;
            fim
            se caminho não existe no conjunto de dados
                Adicione caminho aos dados;
            fim
        fim
    fim
fim
```

para obter a quantidade de trajetórias para cada classe (destino da trajetória) e outra para as origens, com isso, obtemos o valor máximo de amostras que serão geradas pelo algoritmo de sobreamostragem. Em seguida, essas informações foram passadas para a técnica de aumento de dados com grafo de transição, setando os valores máximos de amostras a serem geradas pela função *Transition Graph Augmentation*. Finalizado esse processo, um conjunto de dados balanceado ou semi-balanceado é obtido. A técnica de aumentar os dados a partir de grafo de transição mostrou-se robusta para recuperar trajetórias não observadas, entretanto, para alguns destinos acaba não sendo possível o balanceamento completo por não haver trajetórias suficientes encontradas no grafo. O principal motivo para isso, são que tanto os nós, quanto as arestas são construídos usando o próprio conjunto de dados, com isso, algum ou alguns destinos podem aparecer uma única vez em todo o conjunto de trajetórias, prejudicando a qualidade das predições.

6.6 Construção e execução dos modelos preditivos

Após a aplicação de todas as abordagens propostas, foram obtidos 7 novos conjuntos de dados distintos. Para cada conjunto gerado, foi realizado um treinamento usando os modelos *Decision Tree*, *Random Forest* e *XGBoost*.

Os modelos foram treinados com um dia inteiro de trajetórias e testado com o dia seguinte. Os dados de treino foram variados em 8 tipos diferentes: (i) utilizando os dados brutos das trajetórias, sem a utilização de técnicas especiais de aumentos de dados ou balanceamento; (ii) aplicando a técnica de janela deslizante; (iii) combinando as técnicas de janela deslizante e balanceando as trajetórias com os caminhos mais utilizados; (iv) combinando as técnicas de janela deslizante e balanceando as trajetórias com os caminhos menos utilizados; (v) balanceando as trajetórias com os caminhos mais utilizados; (vi) balanceando as trajetórias com os caminhos menos utilizados; (vii) combinando o balanceamento pelos caminhos mais utilizados com a janela deslizante e; (viii) combinando o balanceamento pelos caminhos menos utilizados com a janela deslizante. Perceba que nos casos onde foram aplicadas duas técnicas, a ordem com qual a técnica foi aplicada importa, pois, retornaram resultados diferentes para os modelos.

Com base na pesquisa realizada, trabalhos que envolvem o tipo de dado de trajetórias abordado são bem escassos. Assim, treinar diferentes modelos de aprendizado de máquina a fim de realizar comparações a cerca do comportamento desses modelos sobre esse o tipo de dado estudo, revela uma importante estratégia para obter informações valiosas, a exemplo, alcançar resultados similares sendo que foi utilizado desde algoritmos mais simples até algoritmos mais sofisticados.

6.7 Análise e comparação do desempenho dos modelos

Nesta sessão, são apresentados os resultados obtidos para cada modelo proposto com a aplicação de cada uma das técnicas implementadas no desenvolvimento deste trabalho.

A seguir são apresentados os resultados para cada modelo proposto com as respectivas combinações de técnicas de aumento de dados. Para cada modelo avaliado são exibidas duas tabelas: a primeira tabela mostrando os resultados da métrica Acurácia, onde a avaliação do modelo é dada pela quantidade de acertos sobre a quantidade total de predições; e a segunda mostrando os resultados da métrica Acurácia Balanceada, neste caso o desbalanceamento das classes é levado em consideração para o cálculo.

6.7.1 Modelo 1 - Árvore de Decisão

Este modelo classifica a amostra a partir de cada decisão tomada, seguindo um fluxo de decisões da raiz até o último nível da árvore. Para isso, foi utilizado o índice de gini como critério para particionar a árvore. Também foi fixado uma semente com valor 42 para que o resultado possa ser reproduzido novamente.

Tabela 3 – Acurácia

	Treino: Dia 01 Teste: Dia 02	**Treino: Dia 02 Teste: Dia 03**	**Treino: Dia 03 Teste: Dia 04**	**Treino: Dia 04 Teste: Dia 05**	**Treino: Dia 05 Teste: Dia 06**
Trajetória Bruta	0.60	0.45	0.48	**0.45**	0.53
Janela Deslizante	**0.64**	**0.53**	**0.49**	**0.45**	**0.55**
Janela Deslizante e Balanceamento (1)	0.62	0.43	0.40	0.33	0.48
Janela Deslizante e Balanceamento (2)	0.60	0.40	0.39	0.36	0.47
Balanceamento (1)	0.57	0.38	0.37	0.33	0.44
Balanceamento (1) e Janela Deslizante	**0.64**	0.48	0.46	0.40	0.50
Balanceamento (2)	0.54	0.32	0.35	0.31	0.43
Balanceamento (2) e Janela Deslizante	0.61	0.44	0.41	0.36	0.48

Fonte: Elaborado pelo autor (2021).

Tabela 4 – Acurácia Balanceada

	Treino: Dia 01 Teste: Dia 02	**Treino: Dia 02 Teste: Dia 03**	**Treino: Dia 03 Teste: Dia 04**	**Treino: Dia 04 Teste: Dia 05**	**Treino: Dia 05 Teste: Dia 06**
Trajetória Bruta	0.42	0.27	0.30	**0.31**	0.32
Janela Deslizante	**0.44**	**0.34**	**0.32**	**0.31**	**0.33**
Janela Deslizante e Balanceamento (1)	0.40	0.28	0.23	0.20	0.26
Janela Deslizante e Balanceamento (2)	0.42	0.26	0.23	0.24	0.26
Balanceamento (1)	0.39	0.24	0.23	0.21	0.24
Balanceamento (1) e Janela Deslizante	0.42	0.32	0.29	0.28	0.29
Balanceamento (2)	0.37	0.19	0.21	0.21	0.25
Balanceamento (2) e Janela Deslizante	0.39	0.28	0.26	0.25	0.28

Fonte: Elaborado pelo autor (2021).

6.7.2 Modelo 2 - Random Forest

Semelhante ao modelo de Árvore de Decisão, o modelo *Random Forest* também recebe como hiper-parâmetro o índice gini como critério para o particionamento de suas árvores e uma semente de valor 42. Além disso, foram setados 100 estimadores para a construção do modelo, valor definido com base no padrão do modelo.

Tabela 5 – Acurácia

	Treino: Dia 01 Teste: Dia 02	**Treino: Dia 02 Teste: Dia 03**	**Treino: Dia 03 Teste: Dia 04**	**Treino: Dia 04 Teste: Dia 05**	**Treino: Dia 05 Teste: Dia 06**
Trajetória Bruta	0.64	0.46	0.49	0.45	0.55
Janela Deslizante	**0.66**	**0.48**	**0.52**	**0.46**	**0.57**
Janela Deslizante e Balanceamento (1)	0.63	0.38	0.43	0.36	0.51
Janela Deslizante e Balanceamento (2)	0.62	0.37	0.41	0.35	0.48
Balanceamento (1)	0.60	0.34	0.39	0.34	0.48
Balanceamento (1) e Janela Deslizante	0.64	0.40	0.45	0.39	0.53
Balanceamento (2)	0.57	0.33	0.37	0.32	0.45
Balanceamento (2) e Janela Deslizante	0.63	0.41	0.42	0.38	0.52

Fonte: Elaborado pelo autor (2021).

Tabela 6 – Acurácia Balanceada

	Treino: Dia 01 Teste: Dia 02	**Treino: Dia 02 Teste: Dia 03**	**Treino: Dia 03 Teste: Dia 04**	**Treino: Dia 04 Teste: Dia 05**	**Treino: Dia 05 Teste: Dia 06**
Trajetória Bruta	0.43	0.27	0.31	**0.31**	0.32
Janela Deslizante	**0.46**	**0.28**	**0.32**	0.30	**0.34**
Janela Deslizante e Balanceamento (1)	0.42	0.24	0.24	0.22	0.28
Janela Deslizante e Balanceamento (2)	0.41	0.23	0.23	0.22	0.26
Balanceamento (1)	0.37	0.20	0.23	0.22	0.26
Balanceamento (1) e Janela Deslizante	0.43	0.24	0.26	0.26	0.31
Balanceamento (2)	0.35	0.20	0.21	0.22	0.25
Balanceamento (2) e Janela Deslizante	0.43	0.25	0.25	0.25	0.30

Fonte: Elaborado pelo autor (2021).

6.7.3 Modelo 3 - XGBoost

Este modelo também recebe um critério para particionamento, mas, diferente dos modelos Árvore de Decisão e *Random Forest*, o critério utilizado é o *FriedmanMSE* (FRIEDMAN, 2001). Além disso, uma taxa de aprendizagem também é requerida para que a perda tenha uma redução. O valor dessa taxa foi mantido como 0.1 de acordo com o padrão do modelo.

Tabela 7 – Acurácia

	Treino: Dia 01 Teste: Dia 02	**Treino: Dia 02 Teste: Dia 03**	**Treino: Dia 03 Teste: Dia 04**	**Treino: Dia 04 Teste: Dia 05**	**Treino: Dia 05 Teste: Dia 06**
Trajetória Bruta	0.69	0.44	0.53	0.45	0.60
Janela Deslizante	**0.72**	**0.51**	**0.55**	**0.49**	**0.62**
Janela Deslizante e Balanceamento (1)	0.64	0.46	0.45	0.34	0.52
Janela Deslizante e Balanceamento (2)	0.68	0.44	0.46	0.42	0.48
Balanceamento (1)	0.63	0.41	0.44	0.42	0.51
Balanceamento (1) e Janela Deslizante	0.69	**0.51**	0.49	0.44	0.56
Balanceamente (2)	0.67	0.42	0.46	0.43	0.55
Balanceamento (2) e Janela Deslizante	0.70	0.47	0.49	0.46	0.57

Fonte: Elaborado pelo autor (2021).

Tabela 8 – Acurácia Balanceada

	Treino: Dia 01 Teste: Dia 02	**Treino: Dia 02 Teste: Dia 03**	**Treino: Dia 03 Teste: Dia 04**	**Treino: Dia 04 Teste: Dia 05**	**Treino: Dia 05 Teste: Dia 06**
Trajetória Bruta	0.48	0.27	0.33	0.30	0.36
Janela Deslizante	**0.49**	0.32	**0.35**	**0.33**	**0.37**
Janela Deslizante e Balanceamento (1)	0.42	0.31	0.29	0.22	0.28
Janela Deslizante e Balanceamento (2)	0.45	0.28	0.29	0.28	0.11
Balanceamento (1)	0.41	0.27	0.27	0.27	0.29
Balanceamento (1) e Janela Deslizante	0.48	**0.35**	0.30	0.30	0.32
Balanceamento (2)	0.45	0.25	0.27	0.28	0.32
Balanceamento (2) e Janela Deslizante	**0.49**	0.31	0.30	0.30	0.34

Fonte: Elaborado pelo autor (2021).

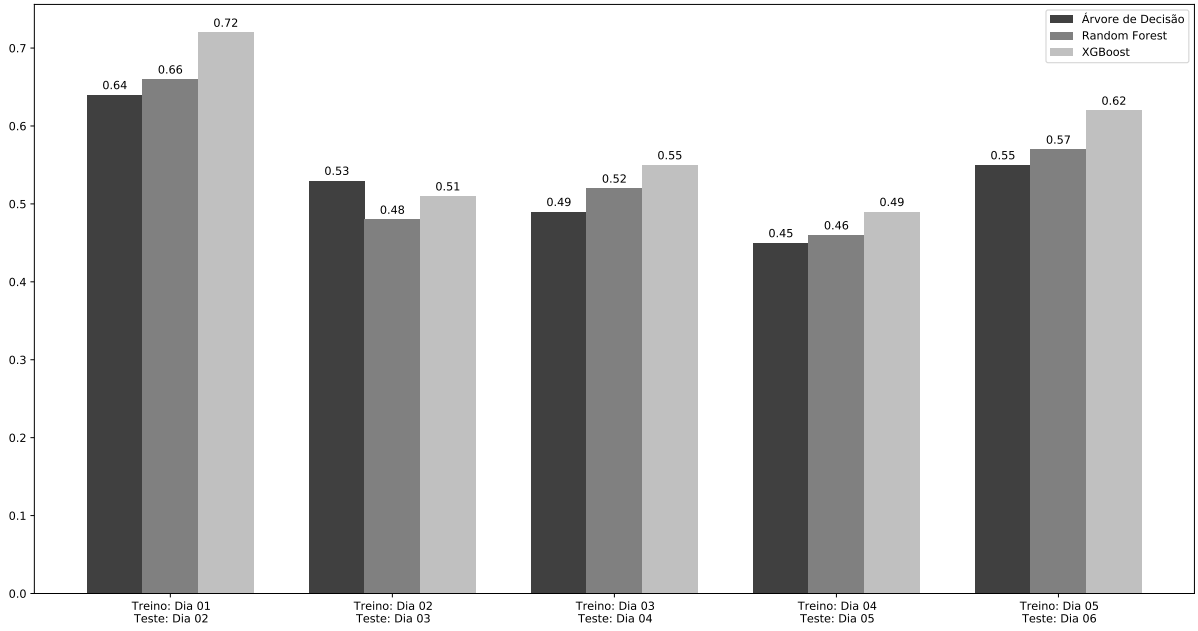
6.7.4 Análise dos resultados

Analisando os resultados das tabelas geradas com o resultado de cada modelo treinado, observamos que a técnica de janela deslizante sempre apresenta um resultado superior aos demais, enquanto que a técnica baseada em grafo gera mais ruído para os dados. A intuição é que como as trajetórias originais realizam percursos com certos padrões, dependendo da frota de veículos, gerar trajetórias com base em transições de um grafo recupera trajetórias misturadas, gerando ruído para o conjunto, pois, essas trajetórias podem apresentar padrões incomuns para os tipos já conhecidos. Enquanto isso, a janela deslizante utiliza sub-trajetórias das trajetórias existentes, com isso, a natureza da sequência não é violada, mantendo os mesmos padrões originais do conjunto.

Além do uso dessas diferentes técnicas, outro fator importante para a análise é modelo de predição proposto. Como observado, os resultados obtidos através das métricas são bem similares. Entretanto, é perceptível que o algoritmo XGBoost apresentou um desempenho superior aos demais, mostrando sua robustez para dados tabulares. O modelo XGBoost treinado neste trabalho alcançou 72% de acurácia e 49% se considerar o desbalanceamento das classes, com a redução do desbalanceamento de dados ao aplicar a técnica de janela deslizante.

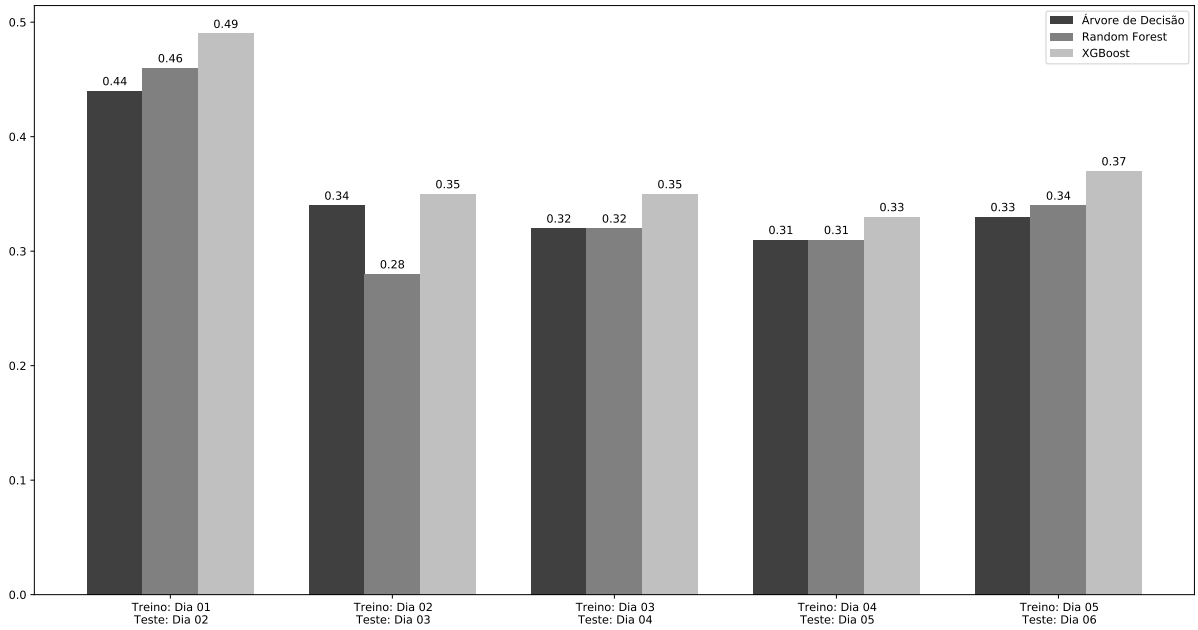
As Figuras 10 e 11 mostram os melhores resultados para cada modelo treinado, o que aponta principalmente os resultados obtidos com a aplicação da técnica de janela deslizante.

Figura 10 – Melhores resultados com acurácia



Fonte: Elaborado pelo autor (2021).

Figura 11 – Melhores resultados com acurácia balanceada.



Fonte: Elaborado pelo autor (2021).

7 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho, foram implementadas técnicas para o aumento de dados para tratar o problema da esparsidade de dados e desbalanceamento de conjunto de sensores de rua. Problemas comumente encontrados nos conjuntos de dados atuais, pela grande geração de informações não uniformes. Além disso, foi proposto um conjunto de modelos para averiguar se a utilização dessas técnicas têm impacto positivo para o desempenho dos mesmos na predição de próxima localização.

As técnicas implementadas apresentaram resultados opostos na qualidade dos dados gerados. A primeira técnica, de janela deslizante, apresentou um resultado positivo para o desempenho dos modelos. Já a segunda técnica, baseada em grafo, apresentou um resultado inferior aos de seus concorrentes.

Como analisado no Capítulo 6, os resultados dos experimentos em geral mostram que a técnica de janela deslizante proposta, impactou positivamente todos os modelos treinados com as diferentes combinações de técnicas. E em contra partida, a abordagem baseada em grafo apresentou os piores resultados para os modelos, chegando a conclusão de que esta estratégia é ruim para o tipo de dado em questão por conter o registro de diferentes frotas de veículos, no qual cada frota apresenta um padrão distinto. Além disso, ao comparar os resultados obtidos pelos modelos propostos, foi mostrado que o modelo XGBoost supera todos para cada técnica aplicada, provando que é um robusto algoritmo para predição.

Como trabalhos futuros, sugerimos a inserção da característica temporal aos dados para o treinamento dos modelos aqui apresentados. Também seria interessante observar o comportamento do aumento de dados baseado em grafo para uma única frota de veículos, e medir a similaridade das trajetórias geradas com alguma métrica de similaridade de trajetória, assim como utilizado por He *et al.* (2020).

REFERÊNCIAS

ABILASH, R. **Applying random forest (classification) — machine learning algorithm from scratch with real datasets**. 2018. Disponível em: https://medium.com/@ar.ingenious/applying-random-forest-classification-machine-learning-algorithm-from-scratch-with-real\-24ff198a1c57. Acesso em: 16 fev. 2021.

BOTTOU, L. Stochastic gradient descent tricks. In: **Neural networks: Tricks of the trade**. [*S. l.*]: Springer, 2012. p. 421–436.

BREIMAN, L. Random forests. **Machine learning**, Springer, v. 45, n. 1, p. 5–32, 2001.

BUCHER, D. Vision paper: Using volunteered geographic information to improve mobility prediction. In: **Proceedings of the 1st ACM SIGSPATIAL Workshop on Prediction of Human Mobility**. [*S. l.*: *s. n.*], 2017. p. 1–4.

CARRILLO, H.; BRODERSEN, K. H.; CASTELLANOS, J. A. Probabilistic performance evaluation for multiclass classification using the posterior balanced accuracy. In: SPRINGER. **ROBOT2013: First Iberian Robotics Conference**. [*S. l.*], 2014. p. 347–361.

CHAWLA, N. V.; BOWYER, K. W.; HALL, L. O.; KEGELMEYER, W. P. Smote: synthetic minority over-sampling technique. **Journal of artificial intelligence research**, v. 16, p. 321–357, 2002.

CHEN, Z.; SHEN, H. T.; ZHOU, X. Discovering popular routes from trajectories. In: IEEE. **2011 IEEE 27th International Conference on Data Engineering**. [*S. l.*], 2011. p. 900–911.

CHICCO, D.; JURMAN, G. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. **BMC genomics**, Springer, v. 21, n. 1, p. 6, 2020.

CHOLLET, F. **Deep Learning mit Python und Keras: Das Praxis-Handbuch vom Entwickler der Keras-Bibliothek**. [*S. l.*]: MITP-Verlags GmbH & Co. KG, 2018.

CRUZ, L. A.; ZEITOUNI, K.; MACEDO, J. A. F. de. Trajectory prediction from a mass of sparse and missing external sensor data. In: IEEE. **2019 20th IEEE International Conference on Mobile Data Management (MDM)**. [*S. l.*], 2019. p. 310–319.

DAI, J.; YANG, B.; GUO, C.; DING, Z. Personalized route recommendation using big trajectory data. In: IEEE. **2015 IEEE 31st international conference on data engineering**. [*S. l.*], 2015. p. 543–554.

DYK, D. A. V.; MENG, X.-L. The art of data augmentation. **Journal of Computational and Graphical Statistics**, Taylor & Francis, v. 10, n. 1, p. 1–50, 2001.

FRIEDMAN, J. H. Greedy function approximation: a gradient boosting machine. **Annals of statistics**, JSTOR, p. 1189–1232, 2001.

GÉRON, A. Hands-on machine learning with scikit-learn and tensorflow: Concepts. **Tools, and Techniques to build intelligent systems**, 2017.

GOEL, E.; ABHILASHA, E.; GOEL, E.; ABHILASHA, E. Random forest: A review. **Int. J. Adv. Res. Comput. Sci. Softw. Eng**, v. 7, n. 1, p. 251–257, 2017.

HAIXIANG, G.; YIJING, L.; SHANG, J.; MINGYUN, G.; YUANYUE, H.; BING, G. Learning from class-imbalanced data: Review of methods and applications. **Expert Systems with Applications**, Elsevier, v. 73, p. 220–239, 2017.

HE, D.; WANG, S.; RUAN, B.; ZHENG, B.; ZHOU, X. Efficient and robust data augmentation for trajectory analytics: a similarity-based approach. **World Wide Web**, Springer, v. 23, n. 1, p. 361–387, 2020.

HO, T. K. Random decision forests. In: IEEE. **Proceedings of 3rd international conference on document analysis and recognition**. [*S. l.*], 1995. v. 1, p. 278–282.

LÓPEZ, V.; FERNÁNDEZ, A.; GARCÍA, S.; PALADE, V.; HERRERA, F. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. **Information sciences**, Elsevier, v. 250, p. 113–141, 2013.

LUQUE, F. M. Atalaya at tass 2019: Data augmentation and robust embeddings for sentiment analysis. **arXiv preprint arXiv:1909.11241**, 2019.

MEHMOOD, S.; PAPAGELIS, M. Learning semantic relationships of geographical areas based on trajectories. In: IEEE. **2020 21st IEEE International Conference on Mobile Data Management (MDM)**. [*S. l.*], 2020. p. 109–118.

NAVLANI, A. **Decision Tree Classification in Python**. 2018. Disponível em: https://www.datacamp.com/community/tutorials/decision-tree-classification-python.

SAFAVIAN, S. R.; LANDGREBE, D. A survey of decision tree classifier methodology. **IEEE transactions on systems, man, and cybernetics**, IEEE, v. 21, n. 3, p. 660–674, 1991.

SAG, I. A.; BALDWIN, T.; BOND, F.; COPESTAKE, A.; FLICKINGER, D. Multiword expressions: A pain in the neck for nlp. In: SPRINGER. **International conference on intelligent text processing and computational linguistics**. [*S. l.*], 2002. p. 1–15.

SHORTEN, C.; KHOSHGOFTAAR, T. M. A survey on image data augmentation for deep learning. **Journal of Big Data**, Springer, v. 6, n. 1, p. 60, 2019.

SOKOLOVA, M.; JAPKOWICZ, N.; SZPAKOWICZ, S. Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. In: SPRINGER. **Australasian joint conference on artificial intelligence**. [*S. l.*], 2006. p. 1015–1021.

TAN, P.-N.; STEINBACH, M.; KUMAR, V. **Introduction to data mining**. [*S. l.*]: Pearson Education India, 2016.

YAO, D.; ZHANG, C.; HUANG, J.; BI, J. Serm: A recurrent model for next location prediction in semantic trajectories. In: **Proceedings of the 2017 ACM on Conference on Information and Knowledge Management**. [*S. l.*: *s. n.*], 2017. p. 2411–2414.