Segmentação de Trajetórias Usando Aprendizagem Supervisionada com Janela Deslizante e Engenharia de Atributos

Wanusa Araújo de Pontes

CENTRO DE CIÊNCIAS EXATAS E DA NATUREZA
UNIVERSIDADE FEDERAL DA PARAÍBA

João Pessoa, PB

2022

Wanusa Araújo de Pontes

Segmentação de Trajetórias Usando Aprendizagem Supervisionada com Janela Deslizante e Engenharia de Atributos

Monografia apresentada ao curso de Estatística do Centro de Ciências Exatas da Natureza à Universidade Federal da Paraíba como parte dos requisitos para a obtenção do título de Graduada em Estatística.

Orientador: Dr. Telmo de Menezes e Silva Filho

Junho de 2022

Catalogação na publicação
Seção de Catalogação e Classificação

P814s Pontes, Wanusa Araújo de.
Segmentação de trajetórias usando aprendizagem supervisionada com janela deslizante e engenharia de atributos / Wanusa Araújo de Pontes. - João Pessoa, 2022.
39 p. : il.

Orientação: Telmo de Menezes e Silva Filho.
TCC (Curso de Bacharelado em Estatística) - UFPB/CCEN.

1. Segmentação de trajetórias. 2. Aprendizagem supervisionada. 3. Janelas deslizantes. 4. Engenharia de atributos. I. Silva Filho, Telmo de Menezes e Silva. II. Título.

UFPB/CCEN CDU 311(043.2)

Elaborado por Josélia Maria Oliveira da Silva - CRB-15/113

DEPARTAMENTO DE ESTATÍSTICA
UNIVERSIDADE FEDERAL DA PARAÍBA

Trabalho de Conclusão de Curso de Bacharelado em Estatística intitulado ***Segmentação de Trajetórias Usando Aprendizagem Supervisionada com Janela Deslizante e Engenharia de Atributos*** de autoria de Wanusa Araújo de Pontes, aprovada pela banca examinadora constituída pelos seguintes professores:

Prof. Dr. Amilcar Soares Junior
Universidade Federal da Paraíba

Profª. Drª. Thaís Gaudencio do Rêgo
Universidade Federal da Paraíba

Prof. Dr. Telmo de Menezes e Silva Filho
Universidade Federal da Paraíba

João Pessoa, 15 de junho de 2022

Departamento de Estatística, Universidade Federal da Paraíba
Conjunto Presidente Castelo Branco III, João Pessoa, Paraíba, Brasil CEP: 58050-585
Fone: +55 (83) 3216-7785

"Carrego todas as memórias,
todos os sabores que daqui provei e
levo comigo os abraços que ganhei."

Vocal Livre

AGRADECIMENTOS

Primeiramente, sou grata eternamente ao Senhor Deus, a Ele toda honra, glória e exaltação, por me sustentar e me fortalecer durante todo o curso. Lembro-me bem que na minha primeira semana de provas da faculdade tive uma crise de ansiedade na qual fiquei bastante doente, mas em nenhum momento Ele me desamparou. E, durante toda a pandemia, onde minha ansiedade se acentuou, mas principalmente no ano corrente, pois juntamente com a pandemia estava o TCC, que desde que entrei no curso é meu ponto fraco, e pra quem me conhece sabe que eu realmente sofro antecipadamente, e foi nessa que, infelizmente fui para a final de probabilidade IV. Sei que até aqui, me sustentou o Senhor, e eu só tenho a agradecer. Portanto, OBRIGADA, JESUS!

Minha gratidão a toda a minha família, meus pais, irmãos, cunhado, avós e avôs, tios e tias, primos e primas, somos muito unidos e sei que, no final, tudo isso é por vocês. Mas, em especial, quero citar aqui, além de meus pais, Jonas e Valquíria, e irmãos, Waleska e Winícius, pelo apoio e sustento, minha prima, Rafaelly com quem eu morei durante quase três anos de minha graduação, e foi quem aguentou minhas crises de ansiedade e noites sem dormir por conta de provas e trabalhos. Você é muito especial para mim, Rafa! Grata a Hellen e Ivanildo, pessoas com quem eu dividi apartamento e pude contar no meu primeiro ano da graduação, vocês foram importantes nessa trajetória.

Sou grata por minhas amizades de infância e ensino médio com quem posso contar para tudo, Filipe, Wallace, Rafael, Evelyn, Taianny, Maria Eduarda, Ilka, João Guilherme e Suênia. Acrescento ainda, minha gratidão a Thiago, meu amigo (em memória) de infância no qual perdi no ano de 2020 em um acidente rodoviário. Aos meus amigos que a estatística me presentou, eu não tenho palavras para agradecer a todos vocês pela amizade, por me apoiar, escutar, adotar, ajudar, e por compartilhar conhecimentos comigo. Obrigada, Nicolas, Letícia, Gislânia, Kleber, Francielle, Bianca e Manuel.

Por fim, mas não menos importante, gratidão a todos os professores do Departamento de Estatística, vocês foram essenciais em minha vida profissional, e serão, para sempre, meus eternos professores. Mas em especial, gostaria de citar Hemílio, aquele que quando na vida real sempre me auxilia em minhas demandas e dúvidas, Luís, Ana Hermínia, Maria Lídia, Everlane, Eufrásio, Tarciana, Tatiene, Marcelo, Rodrigo, e Cláudio Tablada.

Meu eterno agradecimento ao meu orientador Telmo, que tem sido paciente, apoiador e com quem eu tenho aprendido muito. Aos professores do centro de informação e da Memorial University of Newfoundland, Thaís, Yuri, Leonardo e Amilcar, os quais me acompanham nessa reta final e espero contar com vocês sempre, de todo meu coração é inexplicável a minha gratidão a vocês.

Obrigada, pessoal. Sou grata a Deus por ter cruzado minha história com vocês. Aqui é só o começo de uma grande e linda história.

RESUMO

Trajetórias são compostas por dados sequenciais de movimento que podem ser representados por animais, barcos, pessoas, automóveis, dentre outros, apresentando a sua geolocalização com base na latitude e longitude. Uma atividade comum ao analisar esse tipo de dados é a segmentação de trajetórias, que tem por objetivo dividir uma trajetória em segmentos disjuntos, que representam comportamentos diferentes. Este trabalho propõe um novo método de segmentação, chamado *Segmentation Based on Feature Engineering* (SBFE), o qual utiliza engenharia de atributos para representar diferentes variáveis, como velocidade, aceleração e *bearing*, na forma de boxplots, de forma a caracterizar o comportamento local da trajetória em janelas deslizantes. Os dados de boxplots são então usados para treinar um modelo de classificação, mais especificamente o *random forest*, para determinar quais pontos da trajetória representam pontos de particionamento. Assim, o SBFE permite que variáveis que representem características diversas do contexto da trajetória sejam consideradas. Em experimentos usando validação-cruzada com 10 *folds* e quatro conjuntos de dados reais (duas amostras cada dos conjuntos Geolife e Fishing), o SBFE foi avaliado junto ao método que é estado-da-arte na tarefa de segmentação, o *Wise Segmentation Sliding Window* (WS-II). Os métodos foram avaliados segundo a pureza, a cobertura e a média harmônica das duas anteriores. Para as amostras do conjunto Geolife, que possuem mais de duas classes possíveis para os segmentos, o SBFE obteve purezas altas (próximas a 100%) e coberturas menores do que as do WS-II. Já para as amostras do conjunto Fishing, cujos segmentos podem pertencer a apenas duas classes, os resultados dos dois métodos foram comparáveis, mostrando que o novo método tem desempenho competitivo com a literatura existente de segmentação de trajetórias.

Palavras-chave: <Segmentação de Trajetórias>, <Aprendizagem Supervisionada>, <Janelas Deslizantes>, <Engenharia de Atributos>.

ABSTRACT

Trajectories are composed of sequential movement data that can be represented by animals, boats, people, cars, among others, presenting their geolocation based on latitude and longitude. A common task when analyzing this type of data is the segmentation of trajectories, which aims to divide a trajectory into disjoint segments, which represent different behaviors. This work proposes a new segmentation method, called Segmentation Based on Feature Engineering (SBFE), which uses feature engineering to represent different variables, such as speed, acceleration and bearing, in the form of boxplots, in order to characterize the local trajectory behavior using sliding windows. Boxplot data are then used to train a classification model, more specifically random forest, to determine which trajectory points represent partitioning positions. Thus, SBFE allows variables that represent different characteristics of the trajectory context to be considered. In experiments using 10-fold cross-validation with four real datasets (two samples each from the Geolife and Fishing datasets), SBFE was evaluated along with the state-of-the-art method Wise Segmentation Sliding Window(WS-II). The methods were evaluated according to purity, coverage and their harmonic mean. For the samples from the Geolife dataset, which have more than two possible classes for the segments, SBFE obtained high purity (around to 100%) and lower coverage than WS-II. On the other hand, for the Fishing dataset samples, whose segments can belong to only two classes, the results of the two methods were comparable, showing that the new method has a competitive performance, when compared to the existing trajectory segmentation literature.

Key-words: <Trajectory Segmentation>, <Supervised Learning>, <Sliding Windows>, <Feature Engineering>.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS

AM	Aprendizagem de Máquina
CB-SMoT	Clustering Based Stops and Moves of Trajectories
GRASP-UTS	Greedy Randomized Adaptive Search Procedure for Unsupervised Trajectory Segmentation
OWS	Octal Window Segmentation
SBFE	Segmentation Based on Feature Engenering
SPD	Stay Point Detection
ST	Segmentação de Trajetórias
TRACLUS	Trajectory Clustering
WKMeans	Warped K-Means
WSII	Wise Sliding Window Segmentation

Sumário

1 INTRODUÇÃO

Uma trajetória é definida como sendo traços de objetos em movimento. E, os dados de trajetórias podem ser representados pelo movimento sequencial de objetos, animais, automóveis, pessoas, dentre outros. Esses dados apresentam propriedades como a heterogeneidade, pois os objetos em movimento experimentam diferentes circunstâncias, e autocorrelação, visto que a localização de um objeto em movimento está correlacionada com a sua próxima localização no tempo e espaço [3].

Na mineração de dados de trajetória quando o objetivo é classificar subcomportamentos na mesma, é necessário realizar a segmentação da trajetória, visto que um padrão de mobilidade não é considerado para todos os pontos dela, mas para as partes da subtrajetória. Portanto, esse é um dos passos mais delicados do pré processamento da mineração, onde particionam-se trajetórias em segmentos para identificação da mudança de contexto [3].

Para realizar o particionamento de trajetórias para a obtenção de segmentos, diferentes fundamentos podem ser propostos. Para isso, já existem algumas abordagens, como os trabalhos de Palma et al. (2008) [4], Zheng et al. (2011) [5], Leiva et al. (2013) [6], Soares et al. (2015) [7], Lee at al. (2007) [8] e Etemad (2019) [9]. Porém, diferente deste trabalho, esses algoritmos não usam algoritmos de aprendizagem supervisionada para auxiliar na segmentação.

Além dos trabalhos citados acima, a abordagem proposta por Etemad et al. (2020) descreve o método denominado por *Wise Sliding Window Segmentation* (WS-II), mais detalhado na Seção 2.1.2. O WS-II representa o estado-da-arte de segmentação de trajetórias, no entanto, ele se baseia apenas na informação de posição dos pontos da trajetória, não considerando outras possíveis características de contexto que possam contribuir para a segmentação, como velocidade, aceleração, etc.

Assim, este trabalho propõe um novo método de segmentação de trajetórias, capaz de lidar com atributos como a velocidade e diferença do tempo, que podem ser obtidos a partir das características já existentes nos dados. A partir desses atributos são geradas medidas estatísticas como as médias, medianas, mínimos e máximos a partir de janelas deslizantes para montar um conjunto de treinamento para um algoritmo de classificação (neste caso, o *random forest*), cujas predições são usadas para realizar a segmentação.

1.0.1 Objetivo geral

O objetivo deste trabalho é aplicar o processo de segmentação dos dados de trajetória, usando um algoritmo de janelas deslizantes, em conjunto com engenharia de atributos para treinar um classificador capaz de usar diferentes variáveis que representam o contexto de

uma trajetória para realizar a segmentação.

1.0.2 Objetivos específicos

Os passos específicos para atingir a solução, a partir do objetivo indicado, podem ser vistos abaixo:

- Realizar a engenharia de atributos durante a fase de pré-processamento dos dados.
- Utilizar o algoritmo de janelas deslizantes nos dados de trajetória a fim de detectar ou não a mudança de contexto de cada janela.
- Aplicar o processo de segmentação de trajetórias proposto.

1.1 Estrutura da monografia

No Capítulo 2 são abordados os temas fundamentais para a compreensão deste trabalho. No Capítulo 3, serão exibidos detalhes do processo de segmentação de trajetórias proposto, o *Segmentation Based on Feature Engineering* - SBFE. No Capítulo 4, é aplicado o método SBFE em dois bancos de dados e são apresentados seus resultados. Por fim, no Capítulo 5, são apresentadas as conclusões acerca do trabalho, levando em conta o problema abordado e analisando seus pontos fortes e fracos. Além disso, o tópico é finalizado ao comentar sobre trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão tratados conceitos teóricos necessários para o entendimento do desenvolvimento deste trabalho. Primeiramente, serão introduzidas definições sobre a segmentação de trajetórias, incluindo o algoritmo *Wise Sliding Window Segmentation*, proposto por Etemad et al. (2020), e métricas de avaliação de segmentação de trajetórias e Aprendizagem de Máquina (AM).

2.1 Segmentação de Trajetórias

Para compreender o estudo proposto, é necessário apresentar alguns conceitos básicos relacionados a pontos de trajetória e segmentação, assim como o algoritmo utilizado nomeado como janelas deslizantes.

Definição 2.1 (Ponto de Trajetória)**.** Um ponto de trajetória, l_j^o corresponde à localização geográfica do objeto o no tempo j, definido por

$$l_j^o = (x_j^o, y_j^o), \tag{1}$$

onde x_j^o e y_j^o referem-se a longitude e latitude do objeto o no tempo j, respectivamente Etemad et al. (2020) [1]. A latitude varia de 0° a ±90°, com valores positivos e negativos representando posições ao norte e ao sul da linha do Equador, respectivamente. Por sua vez, a longitude varia de de 0° a ±180°, com valores positivos e negativos representando posições ao leste e oeste do Meridiano de Greenwich, respectivamente.

Definição 2.2 (Trajetória Bruta)**.** Uma sequência $\mathcal{T}^o = (l_0^o, l_1^o, \ldots, l_n^o)$, de n pontos de trajetória de um objeto o, forma uma trajetória bruta. Os pontos de uma trajetória podem ser divididos em um ou mais subconjuntos, chamados segmentos.

Definição 2.3 (Segmento)**.** Um segmento é um conjunto de k pontos de trajetória subsequentes e pertencentes a uma mesma trajetória bruta, tal que $s^o = (l_j^o, \ldots, l_k^o)$, $j \geq 0$, $k \leq n$, e $s^o \in \mathcal{T}^o$.

O processo automático que tem como objetivo encontrar segmentos distintos e não-sobrepostos em uma trajetória bruta é chamado de Segmentação de Trajetórias (ST).

Definição 2.4 (Segmentação de Trajetórias)**.** Dada uma trajetória bruta $\mathcal{T}^o = (l_0^o, l_1^o, \ldots, l_n^o)$, define-se uma sequência de segmentos $S = (s_0^o, \ldots, s_p^o)$, tal que

$$\forall_{s_i^o, s_{i+1}^o \in S} s_i^o = (l_j^o, \ldots, l_{j+k}^o), s_{i+1}^o = (l_{j+k+1}^o, \ldots, l_{j+k+u}^o), \tag{2}$$

em que k é o tamanho de s_i^o e u é o tamanho de s_{i+1}^o.

Assim, o objetivo é encontrar uma função $ST : \mathcal{T} \rightarrow S$, em que $|\mathcal{T}| = n+1$, $|S| = p+1$. O ponto final de cada segmento é chamado de posição de partição, pois é onde ocorre a transição de um segmento para o próximo. Atualmente, o estado-da-arte de ST é o algoritmo *Wise Sliding Window Segmentation* (WS-II), proposto por Etemad et al. (2020) [1], mas diversos métodos foram propostos anteriormente na literatura, como exposto na Seção abaixo.

2.1.1 Métodos Existentes de Segmentação de Trajetórias

Existem diversas abordagens para realizar a segmentação de trajetórias. O método *Clustering Based Stops and Moves of Trajectories* (CB-SMoT) [4] busca por paradas, que são pontos de trajetórias em que os objetos apresentam uma velocidade menor, e movimentos, segmentando as trajetórias entre esses dois *clusters*, usando uma extensão do método *Density-Based Spatial Clustering of Applications with Noise* (DBSCAN). Outra técnicas, chamada *Stay Point Detection* (SPD) [5], supõe um ponto de permanência entre cada dois segmentos e obtém os segmentos com base nos parâmetros de limite de distância, θd e tempo, θt. A partir do ponto de permanência tem-se um novo segmento. Um objeto de trajetória é considerado como ponto de permanência caso passe mais de θt tempo nas proximidades de θd.

O *Warped K-Means* (WKMeans) [6] procura minimizar uma função de custo e utiliza o algoritmo *K-Means* para lidar com restrições temporais. O valor de K indica o número de segmentos a serem encontrados pelo WKMeans. Outro algoritmo não-supervisionado, chamado de GRASP-UTS [7], procura gerar segmentos homogêneos. Para iniciar, o algoritmo obtém os pontos de referência aleatoriamente. Posteriormente, gera segmentos mais homogêneos, movimentando os pontos da trajetória e ajustando os pontos de referência com base em um valor de função de custo.

O *Trajectory Clustering* (TRACLUS) [8] é mais um método não-supervisionado e é realizado em duas etapas, sendo a primeira responsável por particionar a trajetória em segmentos de linha e a segunda em agrupar tais linhas em grupos semelhantes. Por último, o *Octal Window Segmentation* (OWS) [9], é um método semelhante WS-II, detalhado na próxima Seção, porém para o OWS utilizou-se um tamanho fixo de janela deslizante.

2.1.2 *Wise Sliding Window Segmentation - WS-II*

O método de segmentação de trajetórias WS-II faz uso de janelas deslizantes para prever o ponto de trajetória esperado ao decorrer de cada janela. O termo janela deslizante indica que, a cada passo t, o algoritmo processa uma quantidade q de pontos de trajetória, formando uma janela $w_t^o = (l_j^o, \ldots, l_k^o)$, $j \geq 1$, $k \leq n$, $w^o \in \mathcal{T}^o$ e $|w_t^o| = q$, "deslizando" a janela para frente no passo $t + 1$, como mostra a Figura 1.

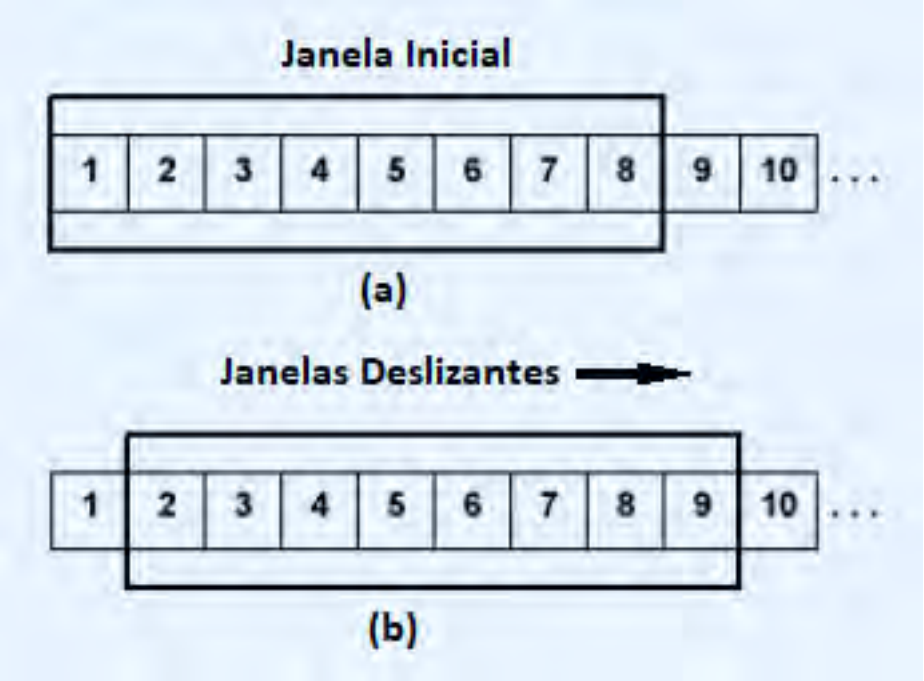


Figura 1: Sequência de janelas deslizantes. Se as janelas forem deslocadas um ponto a cada passo, janelas subsequentes compartilham $q-2$ pontos de trajetória. Fonte: Adaptada de `https://bityli.com/usJa3G`

O WS-II funciona em duas etapas: treinamento e predição. Na etapa de treinamento, o algoritmo realiza duas interpolações nos pontos de cada janela. Por exemplo, suponha que cada janela tenha tamanho $q = 7$, como na Figura 2, que apresenta uma janela $w_t^o = (l_1^o, \ldots, l_7^o)$. Em geral, o WS-II assume tamanhos ímpares de janela, de forma que os primeiros $(q-1)/2$ pontos (nesse caso, os primeiros 3 pontos) são extrapolados "para frente", obtendo uma posição estimada para o ponto seguinte (o quarto ponto), e os últimos $(q-1)/2$ pontos são extrapolados "para trás", obtendo uma posição estimada para o ponto do meio. As funções consideradas para a extrapolação foram: função cúbica, cinemática, linear e caminho aleatório.

O WS-II, então, obtém o ponto médio dos dois pontos extrapolados $\bar{l}_t^o$ e calcula o erro entre $\bar{l}_t^o$ e o ponto do meio real da janela (nesse caso, l_4^o), usando a distância de Haversine. A seguir, o WS-II realiza uma tarefa de classificação binária em que cada amostra de treinamento é um vetor de tamanho q, cujas posições correspondem aos valores de erro dos pontos da janela deslizante de tamanho q correspondente. Além disso, cada amostra de treinamento recebe o Rótulo 1, caso a janela contenha algum ponto de particionamento, caso contrário, a amostra recebe o Rótulo 0. A Figura 3a apresenta um exemplo de um conjunto de treinamento gerado por esse processo. Nota-se que nas janelas em que ocorrem erros maiores, foram observados pontos de particionamento, fazendo com que elas recebam o rótulo 1. Esses dados de erros e rótulos são, então usados para treinar um classificador *random forest* (explicado em detalhes na Seção 2.2.1).

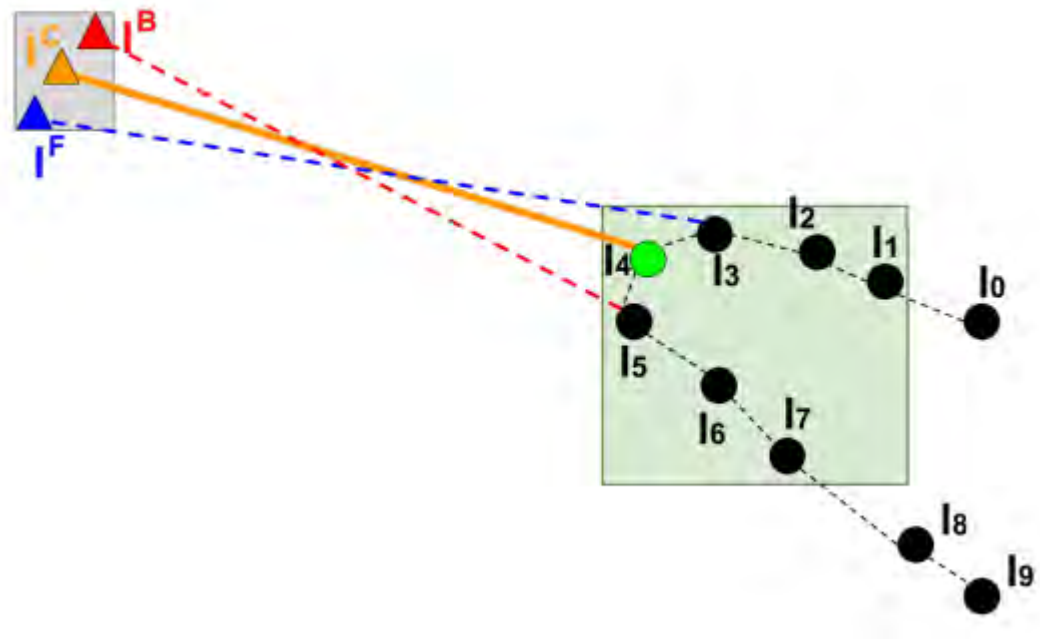


Figura 2: Interpolação para trás e frente dos pontos de trajetória. Fonte: Etemad et al. (2020) [1]

Na etapa de predição, para cada janela de tamanho q da trajetória de teste, o classificador treinado produz uma predição. Cada ponto de trajetória pode fazer parte de q janelas e, portanto, estará associado a q predições do classificador. Por exemplo, na Figura 3b, o classificador produziu uma predição b_{cls} para cada janela $w_m, w_{m+1}, \ldots, w_{m+9}$. Assim, para $q = 7$, o ponto central de w_{m+3} fará parte das janelas $w_m, w_{m+1}, \ldots, w_{m+6}$ e ficará associado às predições $(0, 1, 0, 0, 0, 1, 1)$. Dadas essas predições, o WS-II realiza um voto majoritário ($|\#0| = 3$ e $|\#1| = 4$), classificando o ponto central de w_{m+3} como ponto de particionamento.

A forma de obter o voto majoritário foi estendida no trabalho de Etemad (2020) [3], passando a utilizar o parâmetro μ para classificar o ponto central, onde o mesmo é considerado como ponto de particionamento se pelo menos $mu\%$ das predições classificarem o ponto de trajetória como ponto de particionamento.

	e_1	e_2	e_3	e_4	e_5	e_6	e_7	**label**
...	...	...	...	...	...	...	...	...
w_1	120	160	150	140	180	130	200	0
w_2	160	150	140	180	130	200	210	0
w_3	150	140	180	130	200	210	340	0
w_4	140	180	130	200	210	340	**560**	1
w_5	180	130	200	210	340	**560**	320	1
w_6	130	200	210	340	**560**	320	210	1
$w_{...}$	...	...	...	...	...	...	...	...
w_{10}	**560**	320	210	120	320	273	200	1
w_{11}	320	210	120	320	273	200	130	0

(a) Exemplo de um conjunto de treinamento gerado pelo método WS-II

	e_1	e_2	e_3	e_4	e_5	e_6	e_7	b_{cls}	m_{cls}
$w_{...}$	...	...	...	...	...	...	...	...	...
w_m	100	150	230	160	170	320	400	0	...
w_{m+1}	150	230	160	170	320	400	390	1	...
w_{m+2}	230	160	170	320	400	390	320	0	...
w_{m+3}	160	170	320	400	390	320	380	0	0
w_{m+4}	170	320	400	390	320	380	330	0	1
w_{m+5}	320	400	390	320	380	330	490	1	0
w_{m+6}	400	390	320	380	330	490	450	1	0
w_{m+7}	390	320	380	330	490	450	230	1	...
w_{m+8}	320	380	330	490	450	230	160	0	...
w_{m+9}	380	330	490	450	230	160	190	0	...
$w_{...}$	...	...	...	...	...	...	...	...	...

(b) Exemplo do voto majoritário

Figura 3: Exemplo de dados gerados pelo método WS-II nas etapas de treinamento (a) e de predição (b). Fonte: [1]

Essa predição por voto majoritário produz uma técnica robusta contra saltos espaciais devido a erros na coleta dos dados da trajetória. Valores maiores do tamanho de janela q tornam o algoritmo ainda mais robusto, porém fazem com que o algoritmo tenha dificuldade em detectar segmentos menores do que q.

2.1.3 Métricas de avaliação

Existem diversas métricas ara avaliar o desempenho de modelos de segmentação de trajetórias. Este trabalho utiliza as medidas de pureza e cobertura, pois não dependem de nenhum parâmetro de tolerância, além da sua média harmônica [7]. A pureza dos segmentos avalia a proporção de pontos pertence a um mesmo rótulo em cada segmento. A pureza é obtida pela Equação (3):

$$P(S, \Lambda_L) = \frac{1}{p} \sum_{i=1}^{p} \arg\max_{j \in [i,L]} \frac{N_{ij}}{N_i}, \tag{3}$$

onde S é o conjunto de segmentos estimados pelo método de segmentação, Λ_L é o conjunto de rótulos verdadeiros, p é o número de segmentos estimados pelo método de segmentação, L é o número de segmentos verdadeiros obtidos a partir de Λ_L, N_{ij} é o número de pontos de trajetória dentro do segmento s_i com rótulo λ_j, e N_i é o número total de pontos de trajetória encontrados para o segmento s_i.

Por sua vez, a cobertura avalia a interseção média entre cada segmento real e o maior segmento estimado com o qual o mesmo possui interseção, sendo calculada seguindo a Equação (4).

$$C(S, \Psi_v) = \frac{1}{v} \sum_{i=1}^{v} \arg\max_{j \in [i,L]} \frac{N_{\Psi_i} \cap s_j}{N_i}, \tag{4}$$

onde S é o conjunto de segmentos encontrados pelo método de segmentação, Ψ_v é o conjunto de segmentos baseados nos rótulos verdadeiros, $N_{\Psi_i} \cap s_j$ é o número de pontos de trajetória do segmento s_j que pertencem ao segmento Ψ_i, e N_i é o número de pontos do maior segmento detectado que possui interseção com Ψ_i.

Por fim, para obter uma medida geral de desempenho, toma-se a média harmônica de pureza e cobertura, dada pela Equação (5).

$$H(S, \Lambda_L, \Psi_v) = \frac{2 * P(S, \Lambda_L) * C(S, \Psi_v)}{P(S, \Lambda_L) + C(S, \Psi_v)}. \tag{5}$$

2.2 Aprendizagem de máquina

A aprendizagem de máquina (AM) é uma subárea da inteligência artificial que busca aprender comportamentos, tomar decisões e reconhecer padrões a partir de dados, geralmente representados na forma de uma matriz de atributos X, em que cada linha é chamada de exemplo, amostra ou instância. Existem quatro tipos principais de algoritmos de AM sendo eles: a aprendizagem supervisionada, utilizada neste trabalho e abordada em detalhes na Seção 2.2.1, aprendizagem não-supervisionada, aprendizagem semi-supervisionada e a aprendizagem por reforço.

2.2.1 Aprendizagem Supervisionada

A aprendizagem supervisionada é o paradigma da AM que busca estimar uma função $f : X \rightarrow Y$, onde X são os dados de entrada e Y os rótulos dos dados, que indicam a qual(is) classe(s) ou categoria(s) cada instância pertence, no caso de tarefas de classificação, ou os valores de uma ou mais variáveis quantitativas de interesse, no caso de tarefas de regressão. Este trabalho foca na tarefa supervisionada de classificação, que aparecerá com um dos passos no método proposto de segmentação de trajetórias.

A árvore de decisão é um dos algoritmos utilizados na AM para prever uma determinada decisão ou rótulo, a qual tem como entrada um vetor de valores de atributos, sendo eles discretos ou contínuos. As árvores de decisão são compostas por ramificações resultantes dos testes binários realizados nos nós das árvores. Para cada nó existe uma condição a ser avaliada, e por fim, no caso da árvore de decisão de classificação, obtém-se a classe atribuída à observação. Um exemplo de árvore de decisão é apresentado na Figura 4.

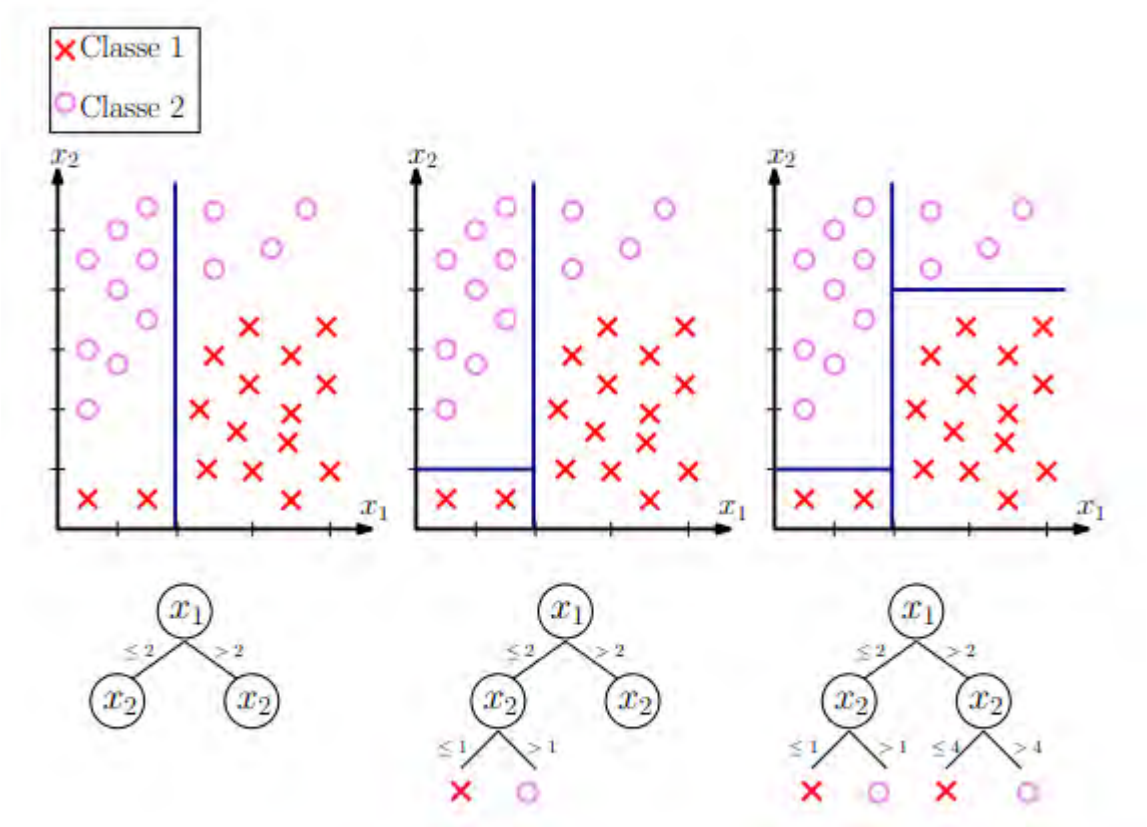


Figura 4: Três passos da construção de uma árvore de decisão. Fonte: *https* : *//bityli.com/2QOGj5W*

Então, como é possível observar na Figura 4, a construção de uma árvore de decisão é um processo iterativo. A cada passo, o algoritmo seleciona uma variável, neste caso x_1 ou x_2, para construir uma regra que separe o conjunto de dados. Para escolher uma das variáveis, o algoritmo verifica qual das variáveis é capaz de produzir uma regra que separe melhor os dados positivos (círculos na cor rosa) dos negativos (cruzes vermelhas). Cada regra vai progressivamente dividindo os dados em "caixas", até que ao final, cada caixa é completamente pura, ou seja, contém apenas elementos de uma mesma classe. Essas caixas finais correspondem às folhas da árvore [10].

Um dos classificadores mais utilizados na aprendizagem de máquina supervisionada atualmente é o *random forest*, cujo objetivo é treinar B árvores de decisão pouco correlacionadas e, posteriormente, para cada nova instância, predizer a classe mais votada entre as B árvores [10]. Para treinar árvores pouco correlacionadas, em cada passo de escolha de variável para construir uma nova regra, cada árvore pode escolher apenas de uma amostra aleatória de tamanho m das variáveis. Tipicamente $m = \sqrt{p}$, em que p é a quantidade original de variáveis do conjunto de dados de treinamento. Com isso, aumenta-se a probabilidade de as B árvores sejam independentes entre si e produzam predições complementares. A Figura 5 apresenta um exemplo de *random forest*.

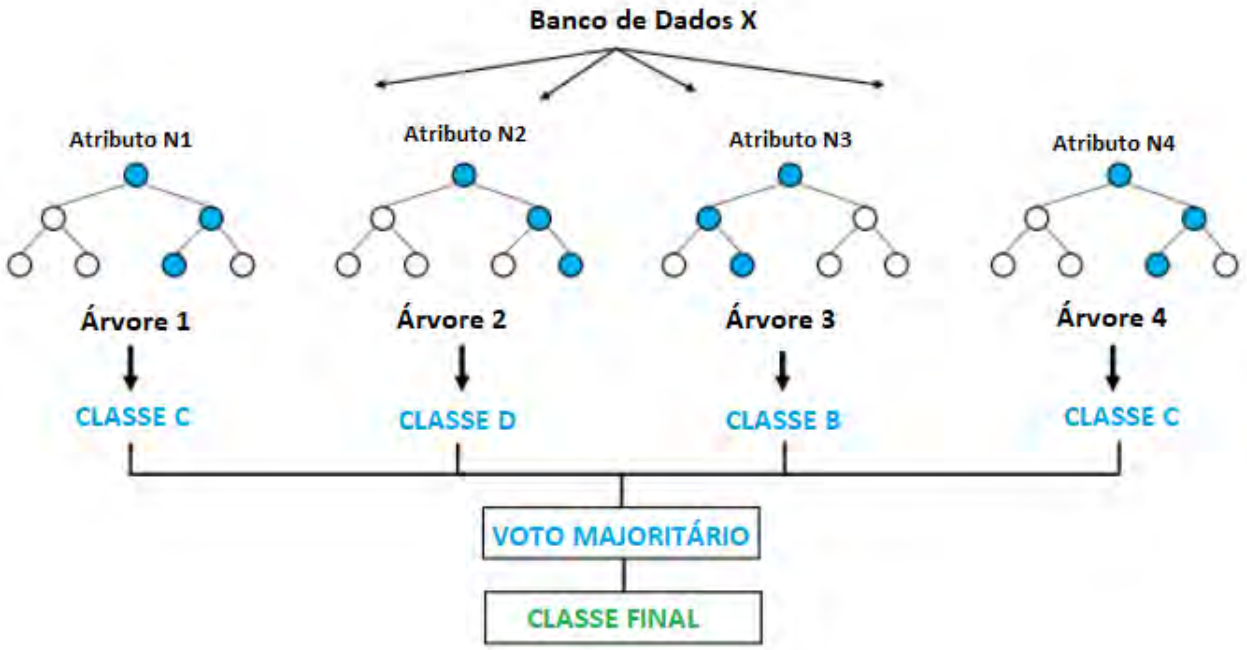


Figura 5: Exemplo de predição em um classificador *random forest*. Fonte: Adaptado de `https://bityli.com/lCdtMD`

Independente do classificador utilizado, em problemas reais é frequentemente necessário realizar algum tipo de tratamento nos dados, para torná-los úteis para servir de entrada em modelos de AM. A seção a seguir trata de ferramentas para realizar esse tratamento.

2.2.2 Pré-processamento e engenharia de atributos

A qualidade dos dados afeta diretamente os resultados gerados pela modelagem, portanto, faz-se necessário organizá-los e avaliá-los de forma que os mesmos não apresentem inconsistências, fase denominada de pré-processamento dos dados. Para isso, existem diversos caminhos para estruturar os dados da melhor forma, a fim de que os resultados preditivos sejam os mais reais possíveis.

Um dos caminhos para realizar a transformação dos dados, com o intuito de melhorar seu desempenho na fase de modelagem, é através da engenharia de atributos, que busca manipular atributos já existentes, ou extrair informações implícitas, para a criação ou modificação de novos atributos [11].

2.2.3 Validação cruzada

Por meio da validação cruzada é possível avaliar a capacidade de generalização do classificador utilizado. Para isso, faz-se necessário dividir os dados em treino e teste.

Na fase de treinamento, o classificador estima uma função $f : X \to Y$ para explicar os dados de treinamento. A função $\widehat{f}$ é aplicada nos dados de teste com a ausência de seu rótulo, com objetivo de realizar a classificação e testar se $\widehat{f}$ aprendeu o suficiente para classificar dados que não estavam na fase de treinamento. Podemos visualizar o esboço da validação cruzada por meio da Figura 6.

Figura 6: Validação Cruzada. Fonte: Schneider (2018) [2]

O método de validação cruzada apresentado acima, é o *k-fold*, que consiste em separar os dados em k subconjuntos disjuntos, a fim de treinar o modelo usando $k-1$ subconjuntos e testar nos dados do conjunto restante. Tal técnica é realizada k vezes e os dados são alternados de forma que todos os k subconjuntos sejam usados para teste.

3 METODOLOGIA

Nesta capítulo será abordada a fase do pré-processamento e o desenvolvimento do algoritmo proposto para realizar a segmentação dos dados de trajetória.

3.1 Pré-Processamento

Na fase do pré-processamento dos dados, são calculados cinco atributos, sendo eles: diferença do tempo, distância, velocidade, aceleração e o *rolamento*, que trata da mudança de direção entre dois pontos subsequentes dos dados da trajetória. Tais atributos são necessários para aplicar o algoritmo de janelas deslizantes, abordado ainda nesta seção.

A diferença do tempo Δt entre dois pontos é obtida por meio da subtração entre os instantes de tempo do ponto atual e do seu subsequente. A diferença de distância é calculada com a distância de Haversine, que é uma expressão de grande relevância na navegação, e fornece a distância entre dois pontos com base na latitude e longitude dos mesmos, neste caso, sendo o ponto atual com seu posterior. A distância de Haversine é obtida pela Equação (6).

$$d = 2r \arctan \sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos(\varphi_1)\cos(\varphi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}, \tag{6}$$

onde φ_1 corresponde à latitude do ponto em estudo, φ_2 corresponde à latitude do ponto subsequente ao de estudo, λ_1 corresponde à longitude do ponto em estudo, λ_2 corresponde à longitude do ponto subsequente ao de estudo e r refere-se ao raio da Terra.

A velocidade v é obtida realizando a razão entre a distância calculada e a diferença de tempo de dois pontos da trajetória, $v = d/\Delta t$. O cálculo da aceleração é realizado com a divisão entre a diferença da velocidade do ponto atual com seu posterior e o tempo, em segundos, do ponto atual $a = \Delta v/\Delta t$.

Por fim, são obtidos os valores do *rolamento*, que considera a mudança de direção entre dois pontos, com base nas informações geográficas dos mesmos, como a latitude e longitude, em graus, resultando nas Equações (7) e (8).

$$x = \sin(\Delta\lambda)\cos(\varphi_2) \tag{7}$$

$$y = \cos(\varphi_1)\sin(\varphi_2) - \sin(\varphi_1)\cos(\varphi_2)\cos(\Delta\lambda) \tag{8}$$

onde $\Delta\lambda$ diz respeito a diferença entre as longitudes de dois pontos e φ_1 e φ_2 referem-se

a latitude de dois pontos.

Portanto, ao calcular-se a arcotangente da divisão entre as Equações (7) e (8), têm-se o rolamento entre dois pontos da trajetória. Porém, como o valor obtido pelo rolamento inicial é retornado em um intervalo de -180 a 180, que não é o que é esperado para um bússola, portanto, faz-se necessário normalizar a medida adicionando e dividindo pelo valor 360.

Por meio dos atributos calculados, utiliza-se o método nomeado como janelas deslizantes, no qual é obtido o ponto do meio de cada janela e são geradas medidas estatísticas que compõem um boxplot (média, mediana, máximo, mínimo, primeiro e último quartil) para os pontos de trajetória antes e depois do ponto central, com base no classificador treinado, neste caso, o *random forest*, considerando os parâmetros padronizados da biblioteca sklearn do Python.

Por exemplo, considerando uma janela em que $ws = 15$, a Tabela 1 apresenta os atributos calculados para uma amostra dos dados de trajetória de tamanho 15. Essa amostra é comporta pelos dados de entrada para aplicar o algoritmo de janelas deslizantes. Os dados obtidos por esse algoritmo são demonstrados na Tabela 2, onde, como mencionado, são calculadas as medidas que constroem um boxplot para cada um dos atributos mencionados nessa seção, para antes e depois do ponto central da janela.

Tabela 1: Dados de Entrada Para as Janelas Deslizantes

	Distância	Aceleração	Rolamento	Velocidade
1	75,5649	0,1618	291,2834	6,6349
2	9588,42	-0,00011	286,402	5,9514
3	15199,8	0,052	332,954	12,562
4	209,368	0,0289	293,225	5,9039
5	31,15	-0,002	291,775	11,631
6	294,02	0,042	284,973	13,107
7	36,4	0,00024	199,204	11,204
8	268,83	0,0013	358,149	10,430
9	23,74	0,0002	350,118	4,770
10	337,66	-0,00024	86,846	12,407
11	71,73	0,006	91,281	9,9638
12	136,65	-0,075	253,05	5,951
13	109,9	-0,052	273,47	6,862
14	69,489	-0,429	272,557	5,433
15	299,69	-0,0164	176,09	11,727

Tabela 2: Dados de Saída Após Aplicar o Algoritmo de Janelas Deslizantes

Atributos	Mínimo	Média	Quartil 1	Mediana	Quartil 3	Máximo
Distância Prévia	10	2940	21	140	1883,5	17012
Distância Posterior	15,5	1143	10	17012	7244	106140
Aceleração Prévia	0,21	2,4	1,84	2,62	3,38	3,64
Aceleração Posterior	0,31	2,39	1,45	3,37	3,69	4,1
..				...		

3.2 Desenvolvimento do Algoritmo

O algoritmo de segmentação para dados de trajetória proposto nesse estudo é o *Segmentation Based on Feature Engineering* - SBFE. Primeiramente, para realizar a segmentação de trajetórias, temos como base o banco de dados gerado pelas janelas deslizantes, em que cada observação refere-se às informações de cada uma das janelas, com seus respectivos rótulos, indicando a existência ou não da mudança de contexto (do inglês, *concept drift*), em que o rótulo é igual a 1, quando há uma provável mudança na trajetória, e classificado como 0, quando tal mudança na trajetória não é identificada.

Para, de fato, realizar a segmentação de trajetória, um segmento é definido como sendo uma sequência de 1, seguido de uma sequência de 0, por exemplo, $1, 1, 1, 1, 1, 0, 0, 0, 0, 0$, consideramos como sendo um segmento. Sendo assim, o SBFE segmenta os dados de trajetória com base no segmento definido.

Porém, adiciona-se a essa estrutura de segmentação o parâmetro da variação de tempo, t , obtido pela diferença entre o primeiro ponto da trajetória do segmento e os pontos subsequentes do mesmo, em que caso a diferença de tempo entre eles seja maior que o parâmetro, os pontos de trajetória de todo o segmento pertence ao seu segmento anterior. Dessa forma, é classificado os segmentos assim como indicando a posição inicial e final pertencente a cada segmento.

3.3 Avaliação do Algoritmo

Para avaliar o método proposto calculou-se as métricas mencionadas na Seção 2.1.3, sendo elas: a pureza, cobertura e a média harmônica entre as duas primeiras. Utilizamos as mesmas medidas realizadas por Etemad et al. (2020), a fim de comparar os métodos.

3.4 Experimento

Primeiramente, obtém-se 10 amostras do banco de dados em estudo. Na construção do experimento para a avaliação dos dados utiliza-se o método da validação cruzada *k-fold*, em que $k = 10$. Para cada iteração, têm-se os dados de treino e teste. Para cada iteração,

uma amostra diferente é tida como dados de treino, enquanto as demais são consideradas como teste.

Para os dados de treino, é necessário estimar $f : X \to Y$, onde X são as medidas que compõem os boxplots de cada janela e Y são seus respectivos rótulos. Para isso obter a função estimada $\widehat{f}$ utiliza-se o classificador random forest.

A função $\widehat{f}$ é aplicada nos dados de cada amostra considerada como teste, obtendo assim seus respectivos rótulos. Com base em tais rótulos aplica-se o método SBFE, e com a finalidade de avaliar a segmentação realizada, calcula-se a pureza, cobertura e média harmônica da pureza e cobertura para cada iteração. Por fim, é possível gerar boxplots de tais métricas para visualizar a variabilidade de tais medidas.

3.5 Bancos de Dados

3.5.1 Geolife

O banco Geolife consiste em dados de mobilidade, coletados a partir do movimento de objetos identificados como: táxi, carro, trem, metrô, avião, barco, motocicleta, ônibus, pessoa correndo e andando, durante os meses de Abril de 2007 a outubro de 2011, pelo Microsoft Research Asia [1].

A partir desse banco de dados são obtidas 10 subamostras geradas aleatoriamente, com as quais foi construído o Banco de Dados 1. O Banco de Dados 2 foi construído com as amostras extraídas do Geolife, porém utilizadas por Etemad et al. (2020) para avaliação de seu método, comparado aos resultados obtidos no presente trabalho.

3.5.2 Fishing

O banco Fishing possui informações sobre a trajetória de barcos de pesca, com uma amostra de 16 navios operando em todas as principais bacias oceânicas durante o mês de Junho de 2012 a Dezembro de 2013, que corresponde a 573,748 pontos de trajetória dos barcos [1]. O banco apresenta também dados detalhados como o tempo coletado, latitude, longitude, a identificação do ponto de trajetória e do barco, e o rótulo que pode ser classificado como pesca e não pesca, 1 ou 0, respectivamente [1].

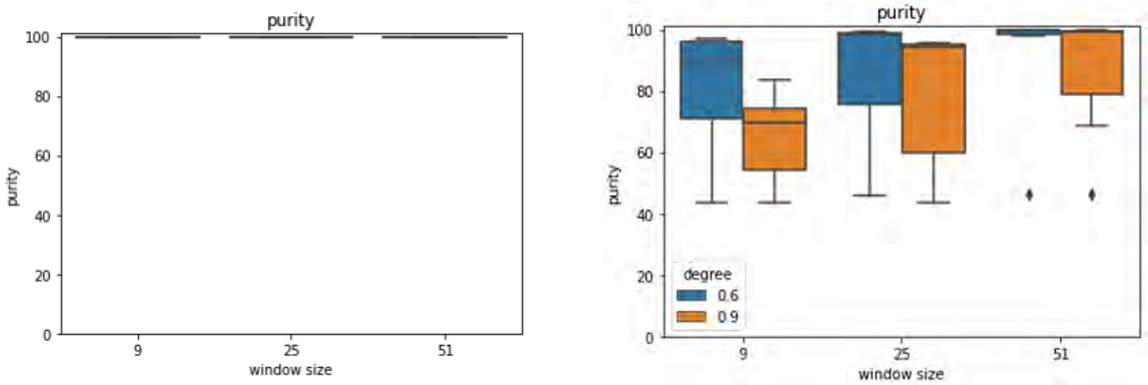
Foram extraídas do banco de dados 10 subamostras, considerando para cada uma delas um identificador de barco diferente. As subamostras foram obtidas através de seus rótulos verdadeiros, os quais representam a detecção de uma mudança na trajetória, nesse caso, de pesca para não pesca ou vice versa. Com isso, além dos dois pontos em que a mudança é identificada, separamos os 1000 pontos de trajetória antes e depois da mudança. Essas subamostras foram utilizadas na construção do Banco de Dados 3. As subamostras obtidas do banco Fishing por Etemad et al. (2020) chamamos de Banco de Dados 4.

4 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

Nesta seção serão apresentados os resultados decorrentes do uso da metodologia explicitada na seção anterior, ou seja, foi gerado um *boxplot* para os resultados de cada métrica obtida pela validação cruzada aplicada nos Bancos de Dados 1,2,3 e 4, tanto para o método SBFE e WS-II. Para isso, utilizou-se dois valores diferentes para parâmetro do voto majoritário utilizado no método WS-II, 0,6 e 0,9, e, para ambos os modelos, considerou-se três tamanhos de janelas deslizantes diferentes: 9, 25, e 51, a fim de avaliar a perfomance de ambos os métodos considerando tamanhos menores e maiores de janelas.

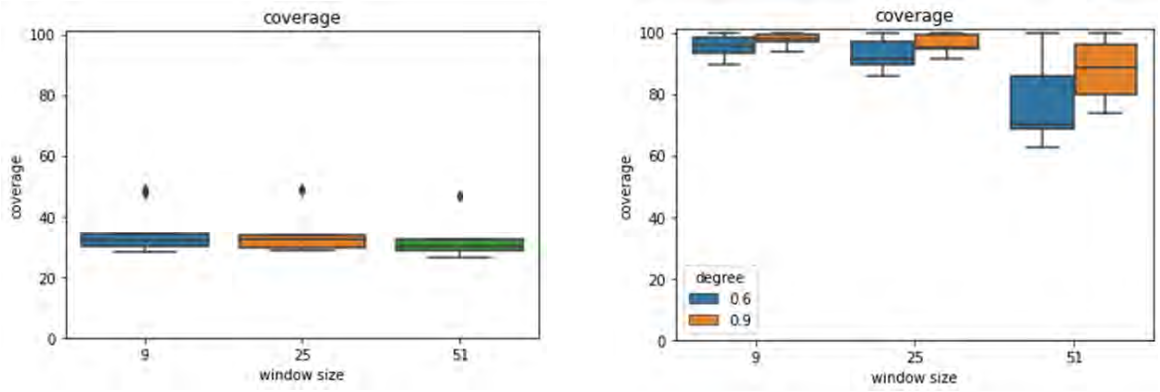
Para o Banco de Dados 1, pode-se observar nas Figuras 7, 8 e 9, os boxplots obtidos para a pureza, cobertura e média harmônica, respectivamente aplicados no Banco de Dados 1 para os métodos SBFE e WS-II. O primeiro está sempre representado pelos gráficos da esquerda de cada figura, enquanto que o segundo pelos gráficos à direita.

Na Figura 7, observa-se que a pureza obtida para o Banco de Dados 1 no método SBFE é bem mais precisa e maior do que a gerada pelo WS-II. Já na Figura 8, apresentam-se os boxplots representando a cobertura, onde é possível observar que o primeiro método tem uma menor variabilidade, considerando todos os tamanhos de janelas deslizantes. Porém, compreende uma mediana menor, quando comparado com o segundo método, representado pelo gráfico à direita. Esse mesmo comportamento pode ser observado na Figura 9, representando a média harmônica dos dados, apresentando apenas uma melhora na mediana do método SBFE e aumentando a variabilidade dos dados no método proposto por Etemad et al. (2020).



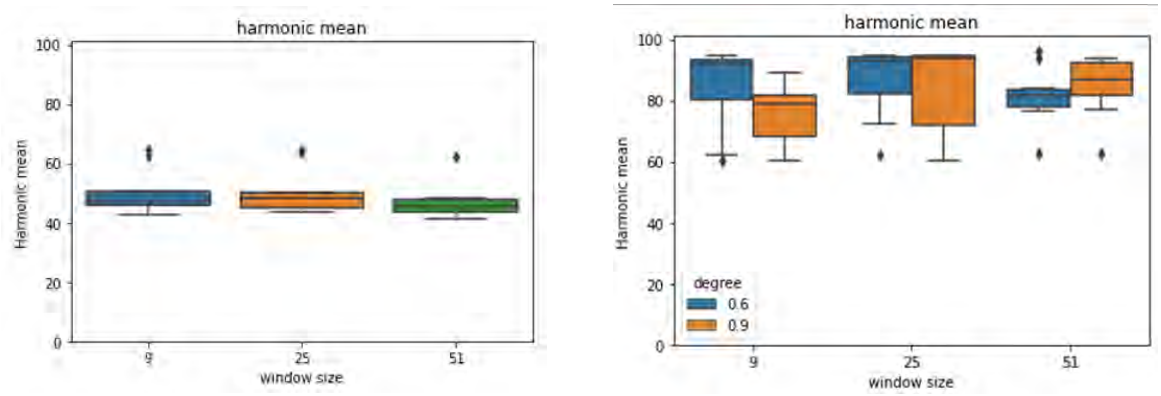
(a) Método SBFE **(b) Método WS-II**

Figura 7: Pureza do Banco de dados 1 - Geolife



(a) Método SBFE **(b) Método WS-II**

Figura 8: Cobertura do Banco de Dados 1 - Geolife

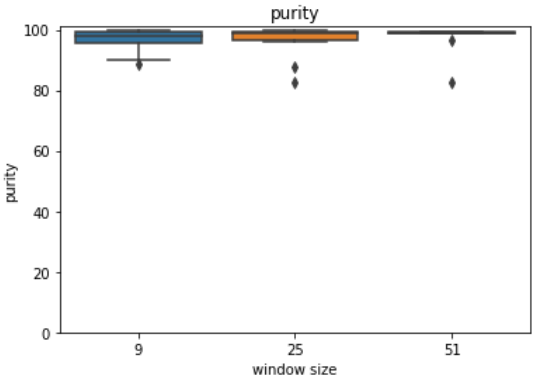


(a) Dados SBFE **(b) Dados WS-II**

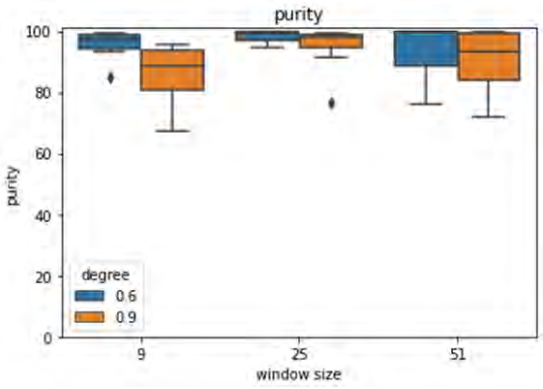
Figura 9: Média Harmônica do Banco de Dados 1 - Geolife

Ao aplicar os métodos SBFE e WS-II no Banco de Dados 2, é possível obter as purezas, coberturas, e médias harmônicas, representadas pelos boxplots apresentados nas Figuras 10, 11 e 12, respectivamente.

Na Figura 10, observa-se que a pureza do método SBFE é mais precisa e tem uma mediana maior, se comparada com os dados obtidos pelo método WS-II. Nas Figuras 11 e 12, representando as coberturas e médias harmônicas calculadas, respectivemente, o método WS-II apresenta uma mediana maior em relação ao SBFE.

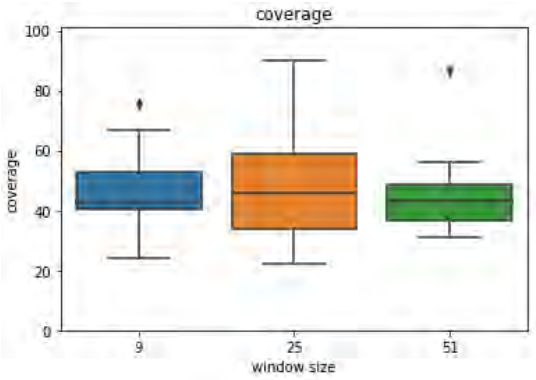


(a) Método SBFE

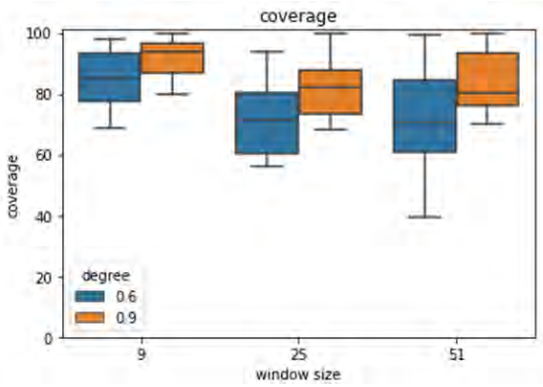


(b) Método WS-II

Figura 10: Pureza do Banco de Dados 2 - Geolife

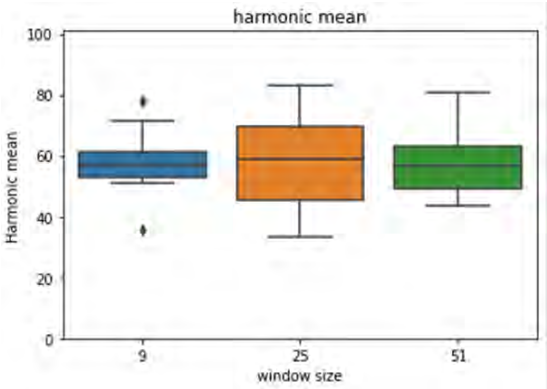


(a) Método SBFE

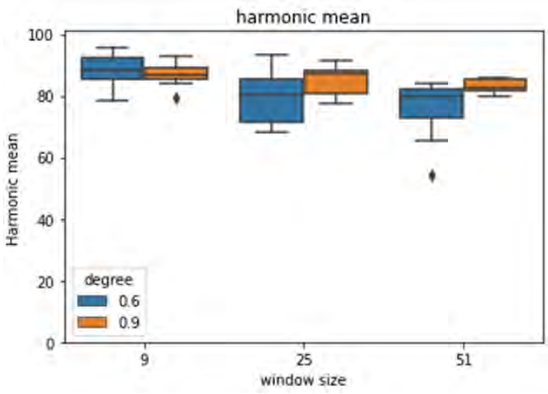


(b) Método WS-II

Figura 11: Cobertura do Banco de Dados 2 - Geolife



(a) Método SBFE



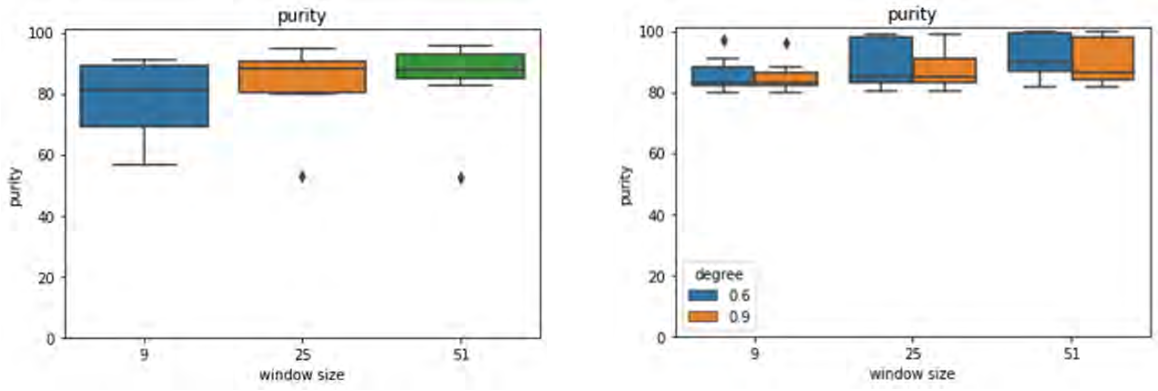
(b) Método WS-II

Figura 12: Média Harmônica do Banco de Dados 2 - Geolife

As Figuras 13, 14 e 15, são boxplots referentes as purezas, coberturas e médias harmônicas da pureza e cobertura calculadas ao aplicar-se a validação cruzada no Banco de Dados 3.

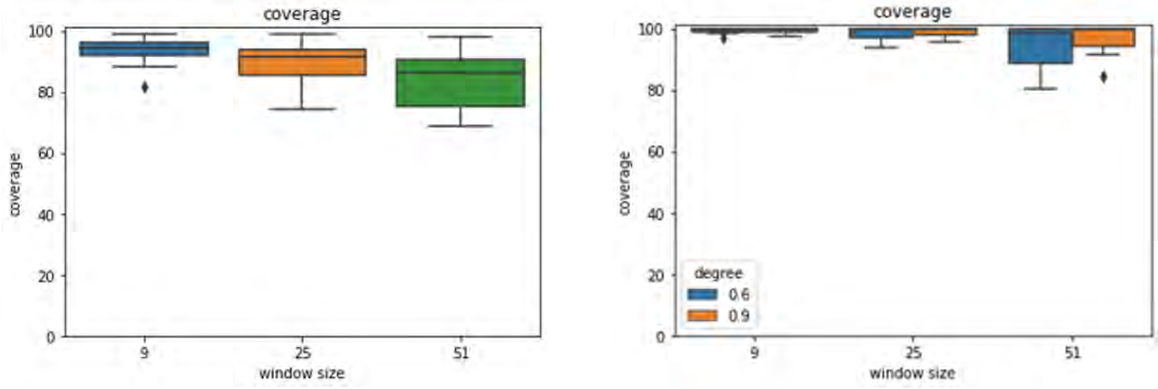
Na Figura 13, têm-se os purezas obtidas e é possível visualizar que a mediana apre-

sentada por ambos os métodos são próximas, diferenciando apenas a variabilidade das purezas calculadas, no qual para o método SBFE há uma menor variabilidade quando se considera um tamanho de janela maior. Porém, nas Figuras 14 e 15, que são os boxplots obtidos pelas coberturas e médias harmônicas, observa-se que o método WS-II apresenta menos variabilidade e uma mediana um pouco maior do que a do método SBFE.



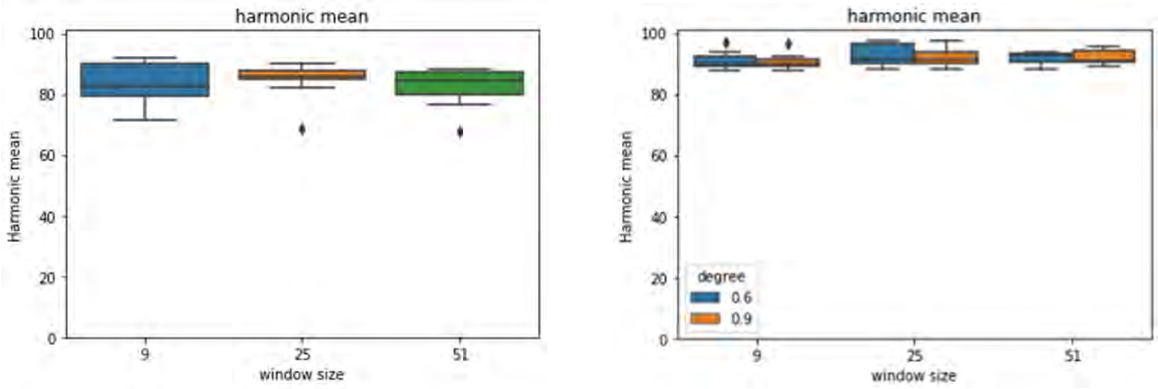
(a) Método SBFE **(b) Método WS-II**

Figura 13: Pureza do Banco de Dados 3 - Fishing



(a) Método SBFE **(b) Método SWS-II**

Figura 14: Cobertura do Banco de Dados 3 - Fishing

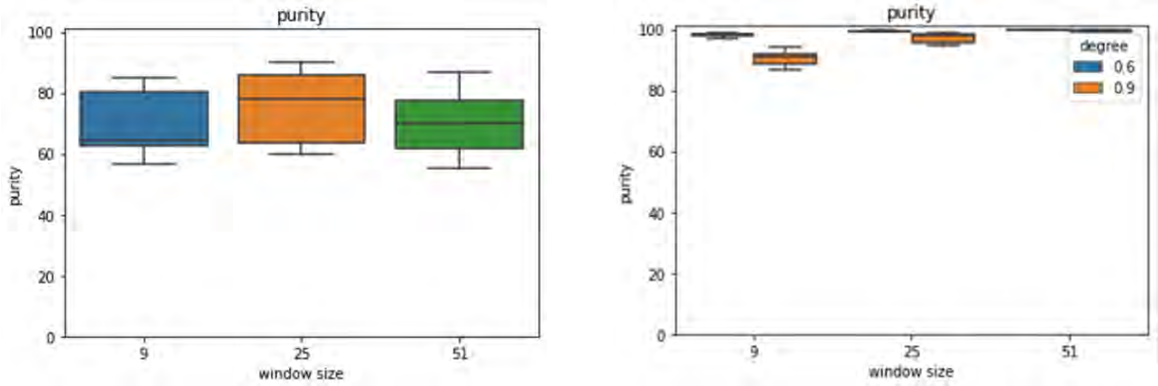


(a) Método SBFE **(b) Método WS-II**

Figura 15: Média Harmônica do Banco de Dados 3 - Fishing

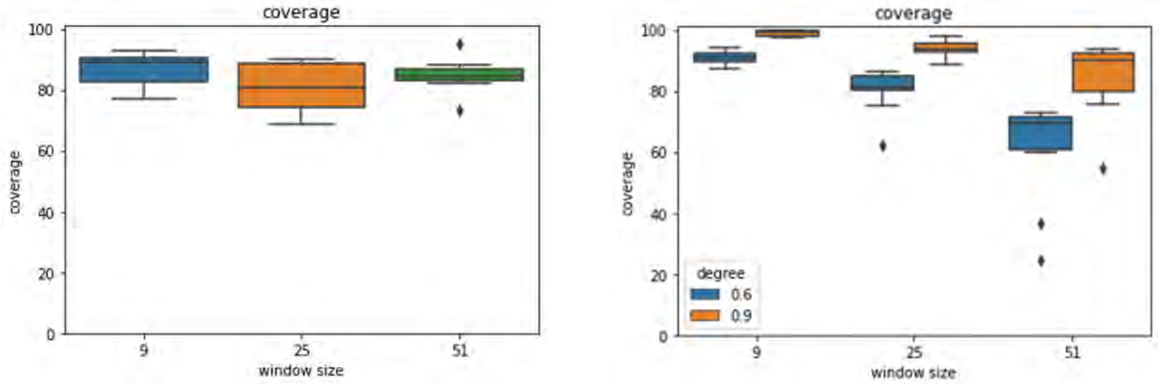
Nas Figuras 16, 17 e 18 apresentam-se as purezas, coberturas e médias harmônicas obtidas ao aplicar os métodos SBFE e WS-II no Banco de Dados 4, respectivamente.

A variabilidade das purezas obtidas pelo método WS-II é menor e apresenta uma mediana maior se comparada com o SBFE, como mostrado na Figura 16. A cobertura e média harmônica obtidas pelo SBFE é maior e consiste em uma menor variabilidade, quando o tamanho das janelas deslizantes é maior, nesse caso, 51. Nas coberturas e médias harmônicas calculadas para tamanho de janela deslizante menor, o WS-II, neste caso, apresenta uma menor variabilidade nas medidas e uma maior mediana.



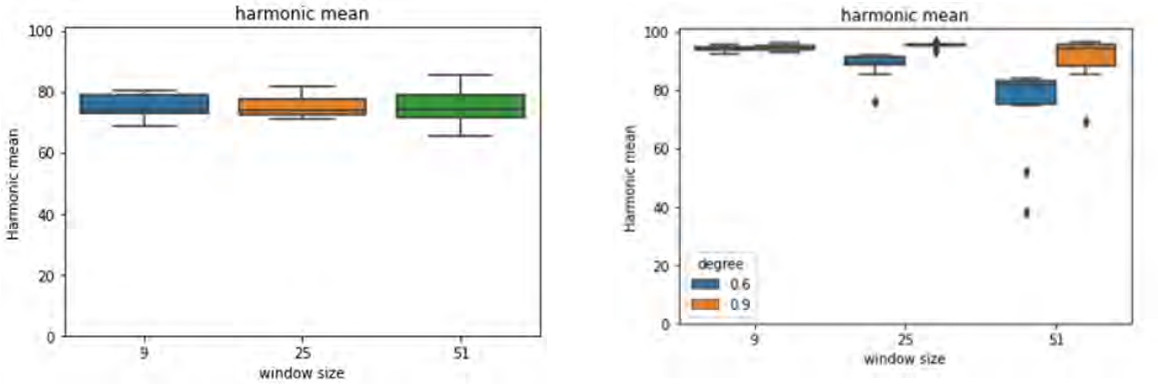
(a) Método SBFE **(b) Método WS-II**

Figura 16: Pureza do Banco de Dados 4 - Fishing



(a) Método SBFE **(b) Método WS-II**

Figura 17: Cobertura do Banco de Dados 4 - Fishing



(a) Método SBFE **(b) Método WS-II**

Figura 18: Média Harmônica do Banco de Dados 4 - Fishing

Ao analisar as purezas obtidas pelo método SBFE para os bancos de dados gerados a partir do Geolife, o qual contém mais de duas formas de trajetórias, são consideradas altas. Em contrapartida, as coberturas obtidas para esses bancos são relativamente baixas, que faz com que a média harmônica seja influenciada por por essas coberturas.

Por outro lado, para os bancos de dados extraídos do Fishing, o qual envolve apenas duas classes, pesca e não pesca, os valores das coberturas e purezas obtidas pelo modelo SBFE estão relativamente próximos.

5 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Com foco na tarefa de segmentação de pontos de trajetória, o método aqui proposto foi comparado com o modelo construído por Etemad et al. (2020) [1].

Com base nos resultados apresentados na Seção 4, observa-se que os valores das purezas obtidas pelo método SBFE, principalmente nos dados que apresentam mais de duas classes de trajetórias, são consideravelmente altas, sendo próximas de 100% e maiores do que os valores obtidos pelo WS-II. Por outro lado, as coberturas apresentadas pelo SBFE para os Bancos de Dados 1 e 2 foram bem menores do que os valores apresentados pelo WS-II. Nos Bancos de Dados 3 e 4, que possuem apenas duas classes possíveis de trajetórias, os dois métodos obtiveram resultados comparáveis, exceto a cobertura no Banco de Dados 3 e a pureza no Banco de Dados 4, com o WS-II tendo desempenhos melhores em ambos os casos.

Sendo assim, uma das limitações apresentadas pelo trabalho proposto é a possível super-segmentação de trajetórias indicada pelos valores altos de pureza, ou seja, o SBFE tende a dividir segmentos em vários sub-segmentos. Portanto, trabalhos futuros incluem avaliar os motivos dessa super-segmentação, buscando um maior equilíbrio entre pureza e cobertura na segmentação produzida pelo SBFE. Além disso o SBFE depende de dois hiperparâmetros: o tamanho da janela e a duração mínima de um segmento. Assim, será importante avaliar abordagens automáticas para encontrar valores ótimos para esses hiperparâmetros.

REFERÊNCIAS

[1] M. Etemad, Z. Etemad, A. Soares, V. Bogorny, S. Matwin, and L. Torgo, "Wise sliding window segmentation: A classification-aided approach for trajectory segmentation," in *Advances in Artificial Intelligence* (C. Goutte and X. Zhu, eds.), (Cham), pp. 208–219, Springer International Publishing, 2020.

[2] C. F. Schneider, "Machine learning aplicado na previsão de resultados de partida de futebol: um estudo de caso para comparação de diferentes classificadores.," 2018.

[3] M. Etemad, "Novel algorithms for trajectory segmentation based on interpolation-based change detection strategies," 2020.

[4] A. T. Palma, V. Bogorny, B. Kuijpers, and L. O. Alvares, "A clustering-based approach for discovering interesting places in trajectories," in *Proceedings of the 2008 ACM symposium on Applied computing*, pp. 863–868, 2008.

[5] Y. Zheng, L. Zhang, Z. Ma, X. Xie, and W.-Y. Ma, "Recommending friends and locations based on individual location history," *ACM Trans. Web*, vol. 5, feb 2011.

[6] L. A. Leiva and E. Vidal, "Warped k-means: An algorithm to cluster sequentially-distributed data," *Information Sciences*, vol. 237, pp. 196–210, 2013. Prediction, Control and Diagnosis using Advanced Neural Computations.

[7] A. Soares Júnior, B. N. Moreno, V. C. Times, S. Matwin, and L. d. A. F. Cabral, "Grasp-uts: an algorithm for unsupervised trajectory segmentation," *International Journal of Geographical Information Science*, vol. 29, no. 1, pp. 46–68, 2015.

[8] J.-G. Lee, J. Han, and K.-Y. Whang, "Trajectory clustering: a partition-and-group framework," in *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pp. 593–604, 2007.

[9] M. Etemad, A. S. Júnior, A. Hoseyni, J. Rose, and S. Matwin, "A trajectory segmentation algorithm based on interpolation-based change detection strategies.," in *EDBT/ICDT Workshops*, 2019.

[10] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 ed., 2009.

[11] M. L. D. da Silva, "Aprendizado de máquina e engenharia de atributos em predição de vendas," 2019.