Pós-Graduação em Ciência da Computação

ROBSON DIAS ALVES TIMOTEO

Combinação Dinâmica de Regressores Aplicada à Localização de Usuários Móveis em Redes sem fio

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
http://cin.ufpe.br/~posgraduacao

Recife
2020

ROBSON DIAS ALVES TIMOTEO

Combinação Dinâmica de Regressores Aplicada à Localização de Usuários Móveis em Redes sem fio

Tese apresentada ao Programa de Pós-Graduação em Ciências da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Doutor em Ciências da Computação.

Área de Concentração: Ciência da Computação.

Orientador:Daniel Carvalho da Cunha

Recife
2020

Catalogação na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

T585c Timoteo, Robson Dias Alves
Combinação dinâmica de regressores aplicada à localização de usuários móveis em redes sem fios / Robson Dias Alves Timoteo. – 2020.
138 f.: il., fig., tab.

Orientador: Daniel Carvalho da Cunha.
Tese (Doutorado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2020.
Inclui referências e apêndice.

1. Ciência da computação. 2. Aprendizagem de máquina. I. Cunha, Daniel Carvalho da (orientador). II. Título.

004 CDD (23. ed.) UFPE - CCEN 2020 - 156

Robson Dias Alves Timoteo

“Combinação Dinâmica de Regressores Aplicada à Localização de Usuários Móveis em Redes sem Fio”

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Doutor em Ciência da Computação.

Aprovado em: 11/02/2020.

_____

Orientador: Prof. Dr. Daniel Carvalho da Cunha

BANCA EXAMINADORA

_____

Prof. Dr. George Darmiton da Cunha Cavalcanti
Centro de Informática / UFPE

_____

Prof. Dr. Abel Guilhermino da Silva Filho
Centro de Informática / UFPE

_____

Prof. Dr. Francisco Madeiro Bernardino Junior
Centro de Ciências e Tecnologia/UNICAP

_____

Prof. Dr. Lisandro Lovisolo
Departamento de Engenharia Eletrônica e de Telecomunicações/UERJ

_____

Prof. Dr. Charles Casimiro Cavalcante
Departamento de Engenharia de Teleinformática/UFC

Dedico essa dissertação à minha esposa, Gabriela.

AGRADECIMENTOS

A Deus, por tudo que tem me proporcionado.

À minha esposa Gabriela, pelo Amor e pelo grande apoio em tudo que faço. Sem o suporte dela, essa Tese não existiria.

Aos Meus pais, pela educação que me proporcionaram.

À minha Sogra, que apoiou esse projeto e tem sido uma segunda mãe para mim.

Ao meu orientador e amigo, Prof. Daniel Cunha, pela dedicação e conselhos durante essa longa jornada.

Ao Engenheiro Paulo Novais pelas medições realizadas em campo.

Ao amigo Lizando Nunes da Silva, pela grande colaboração ao longo do trabalho.

Aos meus professores de Inglês Anthony Lindeen e Vickie Crockett pela ajuda na revisão dos artigos e preparação para as apresentações.

RESUMO

Nos últimos anos, a disseminação de dispositivos computacionais móveis tais como *smartphones* e *tablets* tem estimulado a demanda por serviços baseados em localização. Em geral, técnicas de localização do tipo *fingerprinting* são amplamente utilizadas para implementação desses serviços. No entanto, tais técnicas enfrentam sérias limitações devido a variações abruptas dos níveis de sinais, quando o usuário muda de um ambiente *indoor* para um *outdoor* ou vice-versa, ou mesmo quando estações móveis de diferentes fabricantes são usadas. Nesta tese, é proposta uma nova técnica de localização do tipo *fingerprinting* baseada em aprendizado de máquina, que emprega diferenças de níveis de sinais e atrasos de onda, em vez de usar valores absolutos de níveis de sinais. A ideia principal da proposta é que as diferenças de sinais das estações móveis posicionadas geograficamente no mesmo local são afetadas de forma semelhante pela atenuação de edifícios e outros obstáculos. A técnica proposta utiliza dados da rede celular para treinar um algoritmo supervisionado de aprendizado de máquina com o intuito de estimar o ângulo horizontal de chegada do sinal usando regressão. Para aumentar a acurácia das regressões, é proposto o método DySEREK (*dynamic selection of regressors ensemble using k-nearest-oracles*), que consiste basicamente na seleção dinâmica de subconjuntos de regressores e é baseado na adaptação do método KNORA (*k-nearest oracles*) aplicado à regressão. Por fim, a técnica de localização proposta e baseada no método DySEREK é empregada em uma aplicação real a partir de medições realizadas em uma rede celular de terceira geração (3G). Os resultados obtidos indicam que a técnica proposta é menos sensível à mudança de ambiente (*outdoor* para *indoor*) e mais estável quando aplicada em ambientes *indoor* ou *outdoor*, considerando cenários com menos estações-base (redes celulares em regiões suburbanas ou rurais) e menos dados para treinamento (*drive-tests* menos onerosos realizados pelas operadoras de telecomunicações).

Palavras-chaves:Sistemas baseados em múltiplos regressores. Aprendizagem de máquina. Redes celulares. Localização de usuários. *Fingerprinting*.

ABSTRACT

In recent years, the dissemination of mobile computing devices, such as smartphones and tablets, has motivated the demand for location-based services. In general, fingerprint-based localization techniques are widely used to implement these services. However, such techniques face serious limitations due to abrupt variations in signal levels when the user changes from an indoor to an outdoor environment or vice-versa, or even when mobile stations from different manufacturers are used. In this thesis, we propose a new machine learning fingerprint-based localization technique that uses signal level differences and wave delays, instead of using absolute values of signal levels. The main idea of the proposal is that the signal differences of mobile stations geographically positioned at the same location are similarly affected by attenuation of buildings and other obstacles. The proposed technique uses data from the cellular network to train a supervised machine learning algorithm to estimate the horizontal angle-of-arrival using regression. To increase the accuracy of regressions, the DySEREK (*dynamic selection of regressors ensemble using k-nearest-oracles*) method is proposed, which basically consists of the dynamic selection of regressors subsets and is based on the adaptation of the KNORA (*k-nearest oracles*) method applied to the regression. Finally, the proposed localization technique based on the DySEREK method is employed in a real application from measurements obtained in a third generation (3G) cellular network. Numerical results indicate that the proposed technique is less sensitive to environment change (from outdoor to indoor) and more stable when applied in indoor or outdoor environments, considering scenarios with fewer base stations (cellular networks in suburban or rural regions) and less data for training (less costly drive-tests performed by telecom operators).

Keywords: Multiple regressor systems. Machine learning. Cellular networks. Mobile user location. Fingerprinting..

LISTA DE FIGURAS

Figura 1 – Passos de geração, seleção e integração para criação do conjunto de regressores. 26
Figura 2 – Métodos de geração de diversidade por manipulação dos dados: (a) *Bagging*. (b) *Random Subspace*. 27
Figura 3 – Topologia dos elementos de um sistema baseado em múltiplos regressores: (a) Topologia paralela. (b) Topologia em cascata. (c) Topologia hierárquica. 32
Figura 4 – Tipos de seleção de regressores: (a) Seleção estática. (b) Seleção dinâmica. 34
Figura 5 – Ganho de desempenho (%) *versus* variação de custo computacional (%) entre a técnica CBAS e o melhor *benchmark* (BAGG, RAPT ou BOOS): (a) Coeficiente de correlação. (b) Medida de desacordo. 51
Figura 6 – Diagrama do funcionamento básico do método DySEREK. 55
Figura 7 – Tempo médio normalizado de treinamento e predição, considerando as variantes do DySEREK, o método DESIP e o *pool* CBAS. 67
Figura 8 – Mapa de calor ilustrando o ganho de RMSE entre a pior e a melhor variante do DySEREK. 68
Figura 9 – RMSE médio normalizado, considerando os grupos de variantes *Elimination* (Dy-E e Dy-EW) e *Union* (Dy-U e Dy-UW), do método DySEREK por *dataset* 70
Figura 10 – RMSE médio normalizado, considerando todos os *datasets*, valores de h_K e conjuntos de validação, para cada variante do método DySEREK e tamanho de *pool* de regressores. 71
Figura 11 – (a) Área de localização para a implementação de técnicas do tipo *fingerprinting* (FP) e medições feitas em campo. (b) *Grid* de localização com respectiva resolução. 80
Figura 12 – Relação entre o banco de dados de correlação e o *grid* de localização para a implementação de técnicas de localização baseada em FP. . . . 81
Figura 13 – Diagrama da fase *on-line* da técnica de radiolocalização baseada em FP. 83
Figura 14 – Área de localização com indicação da rota do *drive-test* realizado para a obtenção das medições *outdoor* 85
Figura 15 – Diagrama ilustrativo do processo de construção do conjunto de treinamento do algoritmo de ML. 87
Figura 16 – Geração do CDB: Azimute $\theta_{g,i}$ da g-ésima quadrícula do *grid* em relação ao grupo G_i. 90
Figura 17 – Ambiente urbano na cidade de Recife-PE com a indicação das medições do melhor servidor (*best server*) e da localização das ERBs. 92

Figura 18 – Planta baixa do prédio onde foram realizadas as medições *indoor*. . . . 93
Figura 19 – ERBs consideradas para cada cenário: (a) SCE-4, (b) SCE-6 e (c) SCE-9. 94
Figura 20 – Aumento do erro médio de predição (em %) para as técnicas FP-AoA e FP-RSSI quando a EM procurada se desloca do ambiente *outdoor* para o *indoor*. 98
Figura 21 – Distribuição do erro de predição de posição η para as técnicas FP-AoA e FP-RSSI para os ambientes *indoor*-*outdoor*, ruidoso e não-ruidoso, em todos os cenários considerados. 99
Figura 22 – Erro médio de predição de posição para ambas as técnicas de localização, considerando todos os ambientes e cenários para cada *fold* da validação cruzada. 101
Figura 23 – Função de distribuição cumulativa do erro de predição de posição, considerando todos ambientes e cenários. 102
Figura 24 – Tempo médio de treinamento normalizado para técnicas de localização FP-AoA e FP-RSSI em todos ambientes e cenários. 103
Figura 25 – Diagrama ilustrativo da formação dos *datasets* DS-25, DS-50, DS-75 e DS-100, considerando os *folds* da validação cruzada ($K = 4$) nas medições *outdoor*. 107
Figura 26 – Variação da quantidade de instâncias de treino, teste-*indoor* e teste-*outdoor* para os *datasets* DS-25, DS-50, DS-75 e DS-100. 109
Figura 27 – Erro médio de predição de posição para as técnicas FP-AoA e FP-AoA-Dy, considerando todos os cenários, ambientes e *datasets*. 110
Figura 28 – Mapa de calor com a variação percentual do erro médio de predição de posição da técnica FP-AoA-Dy em relação à técnica FP-AoA, para todos os cenários, ambientes e *datasets* considerados. 111
Figura 29 – Ganho médio (em %) da técnica FP-AoA-Dy em relação à técnica FP-AoA, para os diferentes *datasets* e tipos de ambiente. 113

LISTA DE TABELAS

Tabela 2 – *Datasets* públicos usados nos experimentos com a técnica CBAS. . . . 47
Tabela 3 – Coeficiente de correlação obtido para os 10 *datasets* considerando as técnicas BAGG, RAPT, BOOS e CBAS. 48
Tabela 4 – Medidas de descordo obtidas para os 10 *datasets* considerando as técnicas BAGG, RAPT, BOOS e CBAS. 49
Tabela 5 – Desvio do erro médio quadrático obtida para os 10 *datasets*, considerando a geração de três *pools* e a técnica CBAS. 50
Tabela 6 – Custo computacional médio normalizado para as técnicas BAGG, RAPT, BOOS e CBAS aplicadas aos 10 *datasets* considerados. 50
Tabela 7 – *Datasets* públicos usados nos experimentos com o método DySEREK. . 63
Tabela 8 – Desvio do erro médio quadrático obtido para os 14 *datasets* (32 regressores). 64
Tabela 9 – Desvio do erro médio quadrático obtido para os 14 *datasets* (64 regressores). 65
Tabela 10 – Desvio do erro médio quadrático obtido para os 14 *datasets* (128 regressores). 66
Tabela 11 – Parâmetros utilizados para ajuste (*tuning*) do método DySEREK . . . 69
Tabela 12 – Trabalhos relacionados à aplicação de ML a técnicas de localização de usuários móveis em redes sem fio usando FP. 78
Tabela 13 – Erro médio de predição de posição $\bar{\eta}$, medido em metros, para as técnicas de localização FP-AoA e FP-RSSI nos ambientes *outdoor*, *indoor* e *indoor-outdoor*, ruidosos (R) e não-ruidosos (N-R). 96
Tabela 14 – Parâmetros utilizados para ajuste (*tuning*) do método DySEREK aplicado à localização de usuários móveis. 106
Tabela 15 – *Dataset* Abalone 128
Tabela 16 – *Dataset* AirQuality 129
Tabela 17 – *Dataset* Auto93 130
Tabela 18 – *Dataset* Autompg 131
Tabela 19 – *Dataset* Bodyfat 131
Tabela 20 – *Dataset* BostonHousing 132
Tabela 21 – *Dataset* Carseats 133
Tabela 22 – *Dataset* Concrete 133
Tabela 23 – *Dataset* Cpus 134
Tabela 24 – *Dataset* ForestFire 135
Tabela 25 – *Dataset* Hitters 136
Tabela 26 – *Dataset* Servo 137

Tabela 27 – *Dataset* Slump 137
Tabela 28 – *Dataset* Wage 138

LISTA DE ABREVIATURAS E SIGLAS

ANN	*artificial neural network*
AoA	*angle of arrival*
Auto-ML	*auto machine learning*
BAGG	Bagging
BOOS	Boosting
CBAS	*Combining-Based Technique*
CDB	*fingerprint correlation database*
CFR	conjunto final de regressores
CIR	conjunto inicial de regressores
DESIP	*Dynamic Ensemble Selection with Instantaneous Pruning*
DNN	*deep neural network*
Dy-UW	DySEREK-UNION-W
Dy-U	DySEREK-UNION
Dy-EW	DySEREK-Elimination-W
Dy-E	DySEREK-Elimination
DySEREK	dynamic selection of regressors ensemble using k-nearest-oracles
EM	estação móvel
ERB	estação rádio base
FCC	*Federal Communications Commission*
FP	*fingerprinting*
FP-AoA	*Fingerprinting* based on *angle of arrival*
FP-AoA-Dy	Fingerprinting based on *angle of arrival with DySEREK*
FP-RSSI	*Fingerprinting* based on *received signal strength indicator*
GPS	*global positioning system*
GSM	*global system for mobile communications*
IoT	Internet of Things
ISL	*An Introduction to Statistical Learning*
k-NN	*k-nearest neighbors*
KNORA	*k-Nearest Oracle*
MARS	*multivariate adaptive regression splines*

MKL	*multiple kernel learning*
ML	*machine learning*
MSE	*mean square error*
TDMA	*time division multiple access*
PD	*propagation delay*
PNN	*probabilistic neural network*
RF	rádio frequência
RAPT	Random Patches
RBM	*restricted Boltzmann machine*
RMSE	*root mean square error*
RSSI	*received signal strength indicator*
RSSIS	*received signal strength indicators*
SVM	*support vector machine*
SVR	*support vector regression*
TDoA	*time difference of arrival*
ToA	*time of arrival*
UCI	*University of California, Irvine*
W-CDMA	*wideband code-division multiple access*

SUMÁRIO

1 INTRODUÇÃO 16
1.1 MOTIVAÇÃO E OBJETIVOS DO TRABALHO 16
1.2 ORGANIZAÇÃO DA TESE 19
1.3 RESUMO DAS PRINCIPAIS CONTRIBUIÇÕES 20

2 SISTEMAS BASEADOS EM MÚLTIPLOS PREDITORES 23
2.1 REGRESSOR E CONJUNTO DE REGRESSORES 23
2.2 CRIAÇÃO DO CONJUNTO DE REGRESSORES 26
2.3 TOPOLOGIA DE SISTEMAS BASEADOS EM MÚLTIPLOS REGRESSORES 31
2.4 SELEÇÃO DE SUBCONJUNTOS DE REGRESSORES 33
2.4.1 Método KNORA 34
2.5 RESUMO DO CAPÍTULO 37

3 COMBINAÇÃO PARA GERAÇÃO DO CONJUNTO INICIAL DE REGRESSORES 39
3.1 INTRODUÇÃO 39
3.2 MEDIDAS DE DIVERSIDADE 41
3.3 TÉCNICA PROPOSTA 43
3.4 RESULTADOS NUMÉRICOS 46
3.5 RESUMO DO CAPÍTULO 52

4 SELEÇÃO DINÂMICA DE REGRESSORES 53
4.1 CLASSIFICAÇÃO E REGRESSÃO 53
4.2 MÉTODO DYSEREK 54
4.2.1 Descrição 54
4.2.2 Variantes do Método DySEREK 57
4.3 RESULTADOS NUMÉRICOS 61
4.4 RESUMO DO CAPÍTULO 73

5 GEOLOCALIZAÇÃO DE USUÁRIOS MÓVEIS 75
5.1 INTRODUÇÃO 75
5.2 TÉCNICAS DE RADIOLOCALIZAÇÃO BASEADAS EM FP 79
5.3 PROPOSTA DE RADIOLOCALIZAÇÃO BASEADA EM FP ANGULAR . . 83
5.3.1 Resultados Numéricos 92
5.4 APLICAÇÃO DO MÉTODO DYSEREK EM FP ANGULAR 105
5.5 RESUMO DO CAPÍTULO 114

6 CONCLUSÕES 116

REFERÊNCIAS 119

APÊNDICE A - DATASETS 128

1 INTRODUÇÃO

1.1 MOTIVAÇÃO E OBJETIVOS DO TRABALHO

A disseminação de dispositivos computacionais móveis tem permitido que as pessoas ampliem o acesso aos mais diversos tipos de informações por meio da Internet, bem como pelo uso de aplicativos e redes sociais. No final de 2019, havia cerca de 5,6 bilhões de assinaturas móveis associadas a *smartphones*, com a expectativa de esse valor atingir algo em torno de 7,4 bilhões em 2025, correspondendo a aproximadamente 83% do total de assinaturas móveis (ERICSSON, 2019).

O aumento expressivo do número de conexões móveis, não apenas por conta dos *smartphones*, mas também em virtude de dispositivos móveis de uma maneira geral, serve como um incentivo relevante para o crescimento do interesse em serviços baseados em localização (HUANG et al., 2018). Serviços baseados em localização são aplicações que fornecem informações respaldadas na localização do dispositivo e, consequentemente, do usuário. A acurácia da localização depende, entre outras coisas, do *hardware* e do *software* utilizados no sistema de comunicação móvel. Dessa forma, as técnicas ou algoritmos de localização são diretamente responsáveis pelo valor agregado que os serviços baseados em localização são capazes de fornecer.

Dentre as opções de técnicas de localização, o sistema de posicionamento global (GPS, *global positioning system*) é um dos sistemas mais utilizados no mundo, com aplicações em defesa e segurança nacionais, bem como em uso comercial, civil e científico (NI et al., 2004; RIZK; SHOKRY; YOUSSEF, 2018). Apesar de bem consolidado, o GPS apresenta algumas limitações significativas, seja em ambientes *outdoor*[1] ou *indoor*, uma vez que necessita do estabelecimento de linha de visada entre transmissor e receptor para que sua acurácia não seja comprometida. Além disso, o consumo de energia do *hardware* do GPS costuma ser

[1] Devido a sua ampla utilização na área de localização, iremos utilizar os termos da língua inglesa *indoor* e *outdoor* para indicar ambientes internos e externos, respectivamente.

alto, o que reduz a autonomia da bateria do dispositivo móvel (LI; LEI, 2019; IBRAHIM; YOUSSEF, 2011).

Diante das limitações do GPS, outras alternativas devem ser exploradas, tais como a localização por meio de sinais de rádio frequência (RF), também conhecida por radiolocalização (PERAL-ROSADO et al., 2018). Uma das técnicas bem sedimentadas na literatura é a radiolocalização baseada em *fingerprinting* (FP), a qual utiliza a similaridade entre indicadores de intensidade do nível de sinal (RSSIs, *received signal strength indicators*) para estimar a posição do usuário (BORKOWSKI; LEMPIAINEN, 2006),(WANG et al., 2014),(ZHANG et al., 2019). Não obstante a sua ampla utilização, a técnica de radiolocalização baseada em FP possui alguns desafios relacionados à natureza estocástica do sinal de RF a serem superados. Primeiramente, sistemas de radiolocalização FP baseados em indicador de intensidade do nível de sinal (RSSI, *received signal strength indicator*) perdem a localização do usuário devido à variação abrupta do RSSI quando o usuário sai de ambientes *indoor* para *outdoor* (GOLDONI et al., 2019; LUI et al., 2011).

Outra questão relevante se refere à variação das medições de RSSI quando estações móveis (EMs) de diferentes fabricantes são usadas. Essa variação de RSSI pode ser encontrada até mesmo com EMs diferentes do mesmo fabricante (HUANG; LIN, 2020). Por fim, a taxa de crescimento do número de usuários móveis dos últimos anos tem impactado no aumento do número de estações rádio base (ERBs) nas redes de próxima geração, o que compromete a escalabilidade da técnica de radiolocalização baseada em FP (KIM et al., 2013).

Diante das questões apresentadas, o primeiro objetivo desta tese de Doutorado é a proposição de uma técnica inovadora de radiolocalização baseada em FP, denotada por FP-AoA, que utiliza diferenças de RSSI e parâmetros de atraso de onda da rede celular para prever o ângulo de chegada (AoA, *angle of arrival*) do usuário móvel e com isso, obter uma boa acurácia, diminuindo os efeitos dos problemas anteriormente mencionados.

Com o advento da Internet das Coisas (IoT, *Internet of Things*) e das redes de próxima geração, o número de dispositivos sem fio, que já é maior do que a população mundial desde 2014 (CISCO, 2019), tem crescido consideravelmente. Uma vez que grande parte desses dispositivos conectados são sensores de vários tipos e funcionalidades, uma quantidade massiva de dados é gerada e, para que possa ser transformada em informação útil para as organizações, precisa ser tratada por técnicas apropriadas, tais como processamento de *Big Data* (MOROCHO-CAYAMCELA; LEE; LIM, 2019) e aprendizado de máquina (SULTAN; ALI; ZHANG, 2018).

A necessidade de processar dados não apenas em maior quantidade, mas também em maior velocidade, ressalta a importância da ampliação da capacidade computacional dos sistemas, o que tem ocorrido nas últimas décadas. Este cenário tem possibilitado e incentivado o desenvolvimento de sistemas com múltiplos algoritmos de aprendizado de máquina (ML, *machine learning*), os quais têm apresentado desempenho superior quando comparados a sistemas com algoritmos de ML simples (OLSON; MOORE, 2019; ZHANG; MA, 2012; MENDES-MOREIRA et al., 2012). Nesse aspecto, a tarefa de classificação tem recebido grande parte dos esforços no que se refere ao desenvolvimento de abordagens baseadas em conjunto de preditores (KRAWCZYK et al., 2017; MENDES-MOREIRA et al., 2012). Em particular, podemos destacar o método k oráculos mais próximos (KNORA, *k-nearest oracle*), abordagem baseada em seleção dinâmica de conjunto de classificadores, que tem mostrado excelente desempenho em classificação (VRIESMANN et al., 2012). Considerando o destaque do método KNORA em tarefas de classificação e a pouca quantidade de algoritmos de seleção dinâmica para tarefas de regressão, o segundo objetivo desta Tese é a proposição de um novo método de seleção dinâmica de regressores, baseado em uma adaptação do método KNORA, denominado seleção dinâmica de subconjunto de regressores baseado no k oráculos mais próximos *(DySEREK, dynamic selection of regressors ensemble using k-nearest-oracles)*.

Por fim, uma vez que a aplicação de algoritmos de ML em técnicas de localização de usuários em redes sem fio tem se consolidado como uma importante área de pesquisa nos últimos anos (LI et al., 2019b), o terceiro e último objetivo desta Tese de Doutorado é empregar o método seleção dinâmica de subconjunto de regressores baseado no k oráculos mais próximos *(DySEREK, dynamic selection of regressors ensemble using k-nearest-oracles)* como parte da técnica de localização FP-AoA e verificar como tal abordagem se comporta frente ao sistema de localização com regressor simples.

1.2 ORGANIZAÇÃO DA TESE

No Capítulo 2, são apresentados alguns conceitos básicos relacionados à utilização de sistemas com múltiplos preditores e um detalhamento sobre o método KNORA, algoritmo de combinação dinâmica aplicado a sistemas de múltiplos classificadores que será adaptado para uso em regressão.

No Capítulo 3, uma nova técnica de geração de conjunto de regressores é proposta. A técnica é baseada na combinação dos algoritmos *Bagging*, *Boosting* e *Random Subspace*. O objetivo de combinar os métodos é gerar um conjunto de regressores com maior diversidade. O desempenho da técnica proposta é comparada com outros algoritmos de geração de conjunto de regressores (*pool*). Nesta comparação, a correlação linear e a medida de desacordo (*disagreement measure*) são utilizadas como métrica para diversidade.

O método DySEREK, uma adaptação do método KNORA para regressão, é exposto no Capítulo 4. Inicialmente, são descritas as peculiaridades que devem ser observadas na adaptação de modelos de classificação para regressão. Em seguida, é apresentada uma descrição detalhada do método proposto e suas quatro variantes. O desempenho da nova proposta é comparado com o de outros sistemas de múltiplos regressores e de um regressor simples. Finalmente, é realizada uma investigação de como os parâmetros do método proposto influenciam seu desempenho.

No Capítulo 5, são expostos inicialmente os conceitos básicos sobre localização de

usuários em redes celulares por meio da técnica baseada em FP. Em seguida, uma proposta inovadora, que utiliza FP angular, é apresentada e seu desempenho é comparado ao da técnica FP tradicional (baseada em RSSI) em diversas situações quanto ao número de estações rádio base (ERBs) e tipos de ambientes. Para isso, foram utilizados dados obtidos a partir de medições de campo. Logo após, o método DySEREK é utilizado como regressor no método de localização proposto, em um contexto no qual há um número menor de instâncias para treinamento.

Finalmente, no Capítulo 6, são apresentadas as conclusões que foram obtidas com o estudo e as perspectivas de trabalhos futuros.

1.3 RESUMO DAS PRINCIPAIS CONTRIBUIÇÕES

As principais contribuições desta tese estão listadas a seguir:

- Uma nova técnica de geração de *pool* baseada em combinação é proposta para aumentar a diversidade de sistemas de aprendizagem de múltiplos preditores em problemas de regressão. Mostra-se que a técnica proposta vence as abordagens individuais de geração de *pool* na maioria dos *datasets* considerados, proporcionando um ganho de desempenho com custos computacionais maiores ou equivalentes.

- Uma nova técnica de localização baseada em *fingerprinting*, usando ângulos horizontais obtidos a partir dos sinais de rádio frequência, foi proposta para estimar a posição de usuários móveis em redes celulares. A proposta utiliza diferenças de intensidade de nível de sinal e atrasos de propagação para obter o ângulo horizontal de chegada do usuário. Em comparação com a abordagem tradicional de *fingerprinting*, nossa proposta apresenta um erro médio de predição de posição menor em grande parte dos ambientes e cenários investigados, é menos sensível à mudança de ambiente e mais estável quando aplicado em cenários com menos estações rádio base, tais como as redes rurais e suburbanas.

- Proposta de novos atributos (estimativa dos ângulos horizontais) para serem utilizados no método *fingerprinting*, os quais possibilitam a paralelização do treinamento dos algoritmos de aprendizagem de máquina e da construção do *grid* de localização, o que pode ser interessante para promover a escalabilidade em redes ultra-densas, como as redes celulares 5G.

- Novo método de combinação dinâmico de combinação de regressores, denominado DySEREK, o qual consiste basicamente na seleção dinâmica de subconjuntos de regressores e é baseado na adaptação do método KNORA. Em testes realizados, utilizando-se 14 *datasets*, constatou-se que o DySEREK apresentou desempenho superior quando comparado com o método DESIP (*Dynamic Ensemble Selection with Instantaneous Pruning*).

- O método de combinação dinâmica de regressores proposto nesta tese é utilizado como regressor para a técnica de localização baseada em *fingerprint* angular. Nesse caso, verifica-se que a utilização do método proposto diminui significativamente o erro médio de posição para os cenários com quatro e seis estações rádio base. Nesses cenários, os maiores ganhos com a aplicação do método são obtidos quando conjuntos de treinamento menores são utilizados para treinar os regressores, situação que procura simular o caso no qual poucas medições de campo são realizadas.

- Publicação do artigo "*An approach using support vector regression for mobile location in cellular networks*" no periódico Computer Networks (Springer) (Qualis A1 - Computação).

- Publicação do artigo "*An RSS-based regression model for user equipment location in cellular networks using machine learning*" no periódico Wireless Networks (Qualis A3 - Computação).

- Publicação do artigo "*A Novel Combining-Based Method of Pool Generation for Ensemble Regression Problems*" na conferência The Thirty-Second International Flairs Conference (Qualis B1 - Computação).

- Publicação do artigo "*A scalable fingerprint-based angle-of-arrival machine learning approach for cellular mobile radio localization*" no periódico Computer Communications (Qualis A2 - Computação).

2 SISTEMAS BASEADOS EM MÚLTIPLOS PREDITORES

Nos últimos anos, algoritmos de ML têm sido amplamente utilizados em diversas áreas do conhecimento, desde diagnósticos médicos a sistemas de recomendações (THOMAS et al., 2019; AHMED; IMTIAZ; KHAN, 2018). O aumento da capacidade computacional possibilitou a utilização de sistemas com múltiplos algoritmos de ML, os quais têm demonstrado ótimo desempenho quando comparados a preditores individuais (REN; ZHANG; SUGANTHAN, 2016). Neste Capítulo, inicialmente são apresentados os fundamentos teóricos que servem como base para justificar a utilização de sistemas com múltiplos regressores. Em seguida, são apresentadas as técnicas utilizadas para criação de um conjunto de regressores com alta diversidade. Após isso, são elencadas as topologias utilizadas em sistemas com múltiplos regressores. Por último, são expostas as diferenças entre as seleções estática e dinâmica de regressores e classificadores, além de um detalhamento acerca do método KNORA.

2.1 REGRESSOR E CONJUNTO DE REGRESSORES

Nos dias atuais, a regressão é uma das tarefas mais importantes e amplamente utilizada na área de ML (WATT; BORHANI; KATSAGGELOS, 2020). Sua função é, a partir de dados conhecidos, construir uma função hipótese. A função construída pode ser utilizada para atribuir um valor contínuo a um padrão de teste desconhecido, ou seja, a utilização da função hipótese permite que o regressor possa fazer uma generalização baseada em dados já conhecidos.

A capacidade de generalização de um modelo é uma das principais características desejadas no projeto de sistemas na área de ML (KUHN; JOHNSON, 2013). Segundo (ROONEY et al., 2004), dada uma base de dados específica, a utilização de um conjunto de regressores visa a obtenção de um modelo com maior capacidade de generalização, quando comparado a qualquer modelo gerado a partir de um regressor individual.

Em (DIETTERICH, 2000), são apresentadas três razões fundamentais que justificam o melhor desempenho quando há agregação de algoritmos de ML. A primeira razão é estatística, pois com a utilização de um conjunto de regressores, há uma diminuição do risco de se escolher o pior modelo, principalmente quando há poucos dados para treinar o algoritmo. A segunda razão é computacional, pois diversos algoritmos de ML, tais como redes neurais artificiais (ANNs, *artificial neural networks*), podem ficar presos em um ponto de mínimo local. Essa parada no mínimo local pode ser evitada quando um conjunto de regressores é utilizado. Por exemplo, no caso das ANNs, pesos iniciais distintos poderiam ser utilizados para cada regressor do conjunto. Por fim, a terceira razão é representacional, pois pode não ser possível representar a função hipótese procurada com um só regressor. No entanto, pode ser possível obter uma melhor aproximação com a utilização de um conjunto de regressores. Por exemplo, em (BALSHI et al., 2009), foi utilizado o método *splines* de regressão adaptativa multivariada (MARS, *multivariate adaptive regression splines)*, que utiliza vários regressores lineares para representar uma função não linear, para realizar predições de áreas queimadas em função das alterações no clima.

A utilização de conjunto de modelos (*ensemble methods*) foi inicialmente investigada no contexto da regressão (REN; ZHANG; SUGANTHAN, 2016). Nesse contexto, existe uma prova teórica de que uma seleção apropriada de regressores pode garantir uma melhor precisão em relação a qualquer regressor do conjunto. Essa prova é baseada em decomposição de ambiguidade (KROGH, 1994) e é valida para um único *dataset* (BROWN et al., 2005). Para múltiplos *datasets*, a prova, apresentada em (UEDA; NAKANO, 1996), é obtida com a decomposição do erro em três termos, quais sejam, tendência (*bias*), variância e covariância. Tal decomposição é a base teórica principal para explicar a razão pela qual um conjunto de regressores pode ter um desempenho superior quando comparado a um único regressor (REN; ZHANG; SUGANTHAN, 2016).

Considere um conjunto de M regressores, denotados por $\{f_i\}_{i=1}^{M}$, em que cada regressor

f_i é treinado separadamente com um conjunto de treino de tamanho N, denotado por $\mathbf{z}_i^N$. De posse de uma instância [1] de teste (entrada) $\mathbf{x}$, a saída do i-ésimo regressor é representada por $f_i(\mathbf{x}; \mathbf{z}_i^N)$, ou simplesmente f_i. Tendo como indicador o erro quadrático médio (MSE, *mean square error*), a equação de decomposição do erro é dada por (UEDA; NAKANO, 1996)

$$E[(\overline{f} - t)^2] = \overline{bias}^2 + \frac{1}{M}\overline{VAR} + \left(1 - \frac{1}{M}\right)\overline{COV}, \tag{2.1}$$

em que $E[\cdot]$ é o operador valor esperado e $\overline{f}$ é a média das saídas dos M regressores. O primeiro termo de (2.1) depende do valor médio da tendência, que é definida como o valor médio da diferença entre a saída dos regressores f_i e a saída real t, dada por

$$\overline{bias} = \frac{1}{M}\sum_{i=1}^{M}(E[f_i] - t)\,. \tag{2.2}$$

Um sistema com alto *bias* indica que os preditores não estão conseguindo gerar um função de hipótese adequada (*underfitting*) (KUHN; JOHNSON, 2013).

Ainda em (2.1), temos a média das variâncias e covariâncias dos regressores, definidas respectivamente por

$$\overline{VAR} = \frac{1}{M}\sum_{i=1}^{M} E[(f_i - E[f_i])^2] \tag{2.3}$$

e

$$\overline{COV} = \frac{1}{M(M-1)}\sum_{i=1}^{M}\sum_{j=1,j\neq i}^{M} E[(f_i - E[f_i])(f_j - E[f_j])]\ . \tag{2.4}$$

A variância $\overline{VAR}$ mede a variabilidade média dos regressores utilizados. Sistemas com alta variância não conseguem obter uma boa generalização, pois a função hipótese está com uma alta adequação ao conjunto de treinamento (*overfitting*) (KUHN; JOHNSON, 2013). Por fim, a covariância $\overline{COV}$ representa a correlação média entre os erros cometidos pelo preditores f_i e f_j. De acordo com (2.4), nota-se que a covariância pode ser negativa, o que aumentaria o desempenho do conjunto de regressores. Em (BROWN; WYATT; TINO, 2005), é demonstrado que a seleção apropriada de regressores está ligada à diversidade

[1] amostra de um conjunto

entre eles, ou seja, o melhor conjunto será aquele no qual os erros obtidos pelos regressores possuem uma baixa correlação. Visto que a diversidade é uma importante característica do conjunto de regressores, na próxima Seção serão apresentadas as principais técnicas de indução de diversidade.

2.2 CRIAÇÃO DO CONJUNTO DE REGRESSORES

Um sistema que utiliza múltiplos regressores (*ensemble learning*) pode ser dividido em três passos: geração, poda e integração (ROLI; GIACINTO; GIANNI, 2001). Segundo (BRITTO; SABOURIN; OLIVEIRA, 2014), a poda pode ser vista como uma seleção. Logo, nesta tese adotaremos os passos de geração, seleção e integração, ilustrados na Fig. 1. Vale ressaltar que essa descrição se aplica de forma generalizada, no entanto, em alguns casos, podemos não ter uma separação clara entre os passos (MENDES-MOREIRA et al., 2012).

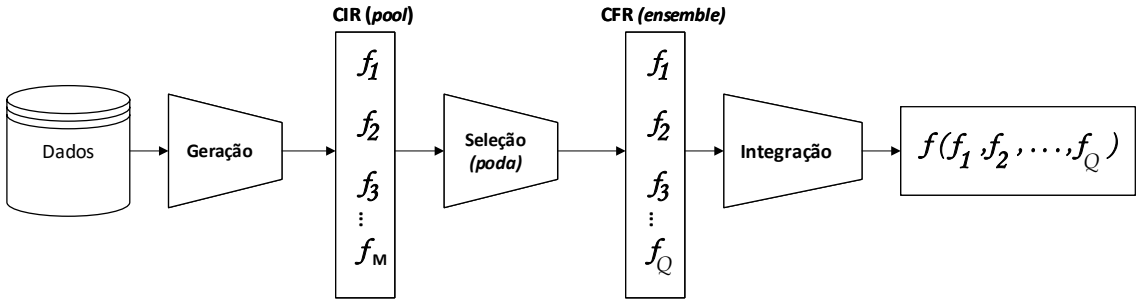

Figura 1 – Passos de geração, seleção e integração para criação do conjunto de regressores.

O primeiro passo para criação do conjunto é a geração dos regressores que farão parte do conjunto inicial de regressores (CIR)[2]. Nessa fase, são treinados M regressores que, para garantir uma melhor capacidade de generalização, devem ter uma alta diversidade entre si (BROWN; WYATT; TINO, 2005).

A indução de diversidade pode ser conseguida por meio de diversas técnicas. Segundo (MENDES-MOREIRA et al., 2012), podemos induzir diversidade manipulando os dados, os parâmetros ou a estrutura do sistema.

[2] Na língua inglesa, esse conjunto é chamado de *pool*.

Em relação aos métodos de manipulação de dados, destacamos as técnicas *Bootstrap Aggregation (Bagging)* (BREIMAN, 1996), *Random Subspace* (HO, 1998) e *Adaptive Boosting (AdaBoost)* (FREUND; SCHAPIRE, 1996). A Fig. 2 apresenta o funcionamento das técnicas *Bagging* e *Random Subspace*, na qual temos uma representação tabular para os dados de treinamento. Neste tipo de representação, cada linha representa uma instância de treinamento e cada coluna representa um atributo (*feature*).

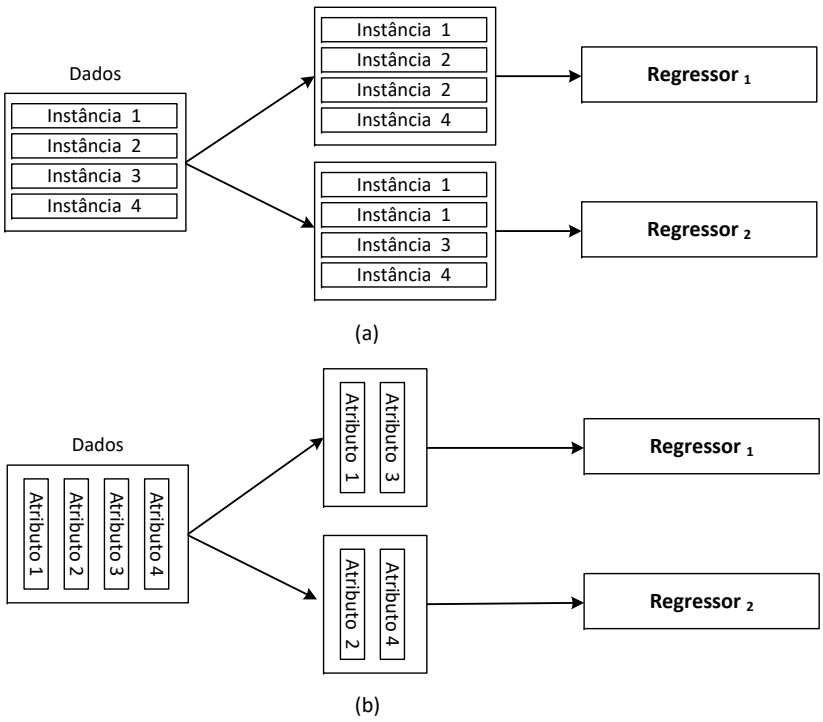

Figura 2 – Métodos de geração de diversidade por manipulação dos dados: (a) *Bagging*. (b) *Random Subspace*.

Na técnica *Bagging*, um novo subconjunto de dados é gerado a partir da amostragem, com repetição, das instâncias de treinamento (ou linhas da tabela). A Fig. 2(a) apresenta o funcionamento do *Bagging* de forma simplificada. Podemos notar que o Regressor 1 é treinado com as instâncias $\{1, 2, 2, 4\}$, enquanto as instâncias $\{1, 1, 3, 4\}$ são utilizadas para treinar o Regressor 2. Os Regressores 1 e 2 serão diferentes, pois foram gerados a partir de diferentes subconjuntos de dados.

Na técnica *Random Subspace*, um novo subconjunto de dados é gerado a partir da amostragem, sem repetição, dos atributos (colunas ou *features*) do *dataset*. O funcionamento da técnica *Random Subspace* é mostrado na Fig. 2(b), na qual os atributos que estão nas colunas 1 e 3 são utilizados para o geração do Regressor 1, enquanto que os atributos das colunas 2 e 4 são utilizados para geração do Regressor 2. Os regressores gerados serão diferentes, pois os subconjuntos de dados gerados diferem em relação aos atributos.

As técnicas de *Bagging* e *Random Subspace* podem ser utilizadas simultaneamente. Por exemplo, em (WANG; MA, 2012) ambas as técnicas são aplicadas para o treinamento de um conjunto de máquinas de vetores de suporte (SVMs, *support vector machines*) para predição de risco de crédito.

A terceira técnica, *AdaBoost*, assim como a técnica *Bagging*, é baseada em reamostragem. Todavia, na técnica *AdaBoost*, cada instância tem uma probabilidade diferente de ser selecionada. A cada iteração, as instâncias com maior erro de predição tem um incremento na probabilidade de serem selecionadas. A técnica *AdaBoost* foi originalmente desenvolvida para tarefas de classificação. Na tarefa de classificação, o objetivo é atribuir uma classe, dentre um número predeterminado de classes (categorias), assim não foi possível a utilização da técnica diretamente na regressão, pois a função que atualiza as probabilidades de seleção tem um erro de generalização restrito ao intervalo $]0,5; 1,0]$ (MENDES-MOREIRA et al., 2012). Os algoritmos *AdaBoost.R* (FREUND; SCHAPIRE, 1995) e *AdaBoost.RT* (SHRESTHA; SOLOMATINE, 2006) são adaptações da técnica *AdaBoost* para tarefas de regressão.

Além dos dados, os parâmetros do regressor podem ser utilizados para indução de diversidade. Nesse caso, cada regressor utiliza um conjunto diferente de parâmetros. Um tipo de técnica amplamente utilizada é a validação cruzada com *K-fold*[3] , uma técnica de reamostragem na qual as amostras são divididas aleatoriamente em K subconjuntos de

[3] O termo da língua inglesa *fold* será usado para fazer referência a um subconjunto da validação cruzada.

tamanhos aproximadamente iguais (ABU-MOSTAFA; MAGDON-ISMAIL; LIN, 2012). Esses subconjuntos são denominados *folds* e são divididos em dois grupos: o conjunto de testes com apenas um *fold* e o conjunto de treinamento com $(K-1)$ *folds*. O Algoritmo 1 mostra o pseudocódigo da validação cruzada com K *folds*. Em (GÖNEN; ALPAYDIN, 2011), é apresentada a aprendizagem com múltiplos *kernels* (MKL, *multiple kernel learning*), na qual um conjunto de *kernels* com parâmetros diferentes é utilizado em tarefas de regressão e classificação.

Algoritmo 1: Descrição da técnica de validação cruzada *K-fold*.

```
1 Particione os dados em K conjuntos disjuntos Z_1 ... Z_K
2 para j = 1 até K faça
3 |   Use Z_j para validação e os outros conjuntos para treino. Calcule o erro no
  |   conjunto de validação Z_j.
4 fim
5 Retorne o erro médio dos conjuntos de validação.
```

A diversidade estrutural pode ser obtida utilizando diferentes algoritmos ou alterando a arquitetura do regressor (REN; ZHANG; SUGANTHAN, 2016). No primeiro caso, o sistema é dito heterogêneo, como será visto na Seção 2.4, pois não utiliza apenas um algoritmo base. No segundo caso, um exemplo seria utilizar um conjunto de ANNs com número diferente de camadas.

Após o treinamento dos regressores, temos a fase de seleção ou poda. Nesta fase, são selecionados os regressores que farão parte do conjunto final de regressores (CFR)[4]. Segundo (HERN et al., 2006), o objetivo da seleção é diminuir o número de regressores, sem reduzir significativamente a precisão das previsões do conjunto. A diminuição do custo computacional é outro fator que justifica a seleção (BAKKER; HESKES, 2003). Além disso, em alguns casos uma seleção apropriada de regressores pode superar consideravelmente o desempenho do conjunto completo (MENDES-MOREIRA et al., 2012). Na Fig. 1, após a geração, é realizada a seleção, na qual são selecionados Q regressores do CIR para compor o CFR. Vários algoritmos de seleção podem ser utilizados nessa fase, tais como os

[4] Na língua inglesa, esse conjunto é chamado de *ensemble*.

algoritmos de seleção do tipo ordenação, *cluster* ou otimização (ZHOU; WU; TANG, 2010).

Nos algoritmos do tipo ordenação, os regressores são ordenados em relação a algum critério e os primeiros são selecionados. Por exemplo, em (HERN et al., 2006), os regressores são selecionados em função do desvio do erro quadrático médio (RMSE, *root mean square error*) obtido pelo subconjunto após a adição do regressor, sendo o RMSE dado por

$$\mu_e = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_i - f_r\left(\mathbf{x}_i\right)\right)^2}\ , \tag{2.5}$$

em que N é o número de instâncias utilizadas para verificar o desempenho do regressor, y_i é o valor real da variável alvo (*label*) para instância $\mathbf{x}_i$ e $f_r(\mathbf{x}_i)$ é o valor predito de y_i obtido pelo r-ésimo regressor que está sendo testado. Nos algoritmos do tipo *cluster*, os regressores são agrupados de modo que sejam selecionados indivíduos que pertençam a diferentes grupos. Em (BAKKER; HESKES, 2003), ANNs são separadas em *clusters* e um representante de cada grupo é selecionado para fazer parte do CFR. Por fim, nos algoritmos do tipo otimização, a seleção do conjunto é obtida por meio da maximização ou minimização de uma função objetivo. Por exemplo, em (ZHOU; WU; TANG, 2010), algoritmos genéticos são utilizados para a escolha do melhor subconjunto de ANNs.

Após a seleção, as saídas dos regressores são integradas para gerar a predição final do sistema. A saída f do CFR é representada como (ROONEY et al., 2004; MENDES-MOREIRA et al., 2012)

$$f(\mathbf{x}_t) = \sum_{r=1}^{Q} w_r f_r(\mathbf{x}_t)\ , \tag{2.6}$$

em que Q é o número total de regressores no CFR, $\mathbf{x}_t$ é um vetor que representa a instância de teste, f_r é a saída do r-ésimo regressor e w_r é o peso do r-ésimo regressor. Assim, conforme (2.6), a integração pode ser realizada por meio de combinação (fusão) das saídas. Alguns pesquisadores apontam que a integração também pode ser vista como um segundo estágio de seleção (ROONEY et al., 2004), mas como pode ser observado em (2.6),

a seleção seria um caso específico da integração, no qual os regressores não selecionados teriam seus pesos ajustados para zero (MENDES-MOREIRA et al., 2012). Apresentados os conceitos referentes à criação do conjunto de regressores, na próxima Seção serão expostos os possíveis tipos de organização de seus elementos (regressores).

2.3 TOPOLOGIA DE SISTEMAS BASEADOS EM MÚLTIPLOS REGRESSORES

A organização (topologia) dos elementos de um sistema baseado em múltiplos regressores é uma característica importante a ser considerada pelo projetista. A topologia é a forma como os elementos do sistema estão interligados. Em (LU, 1996), temos a divisão da topologia em três categorias: paralela, cascata e hierárquica. Vale ressaltar que essa divisão foi sugerida inicialmente para conjuntos de múltiplos classificadores, mas é totalmente aplicável a conjunto de regressores.

A Fig. 3(a) ilustra a representação da topologia em paralelo. Nessa topologia, os regressores geram resultados independentes e um processo de integração é utilizado para gerar a resposta final. Como os regressores estão em paralelo, é possível treinar todos simultaneamente. A disposição em paralelo pode gerar um altíssimo ganho de desempenho em relação ao tempo, pois dependendo do tamanho da base de dados, o tempo para treinar um único regressor pode ser longo.

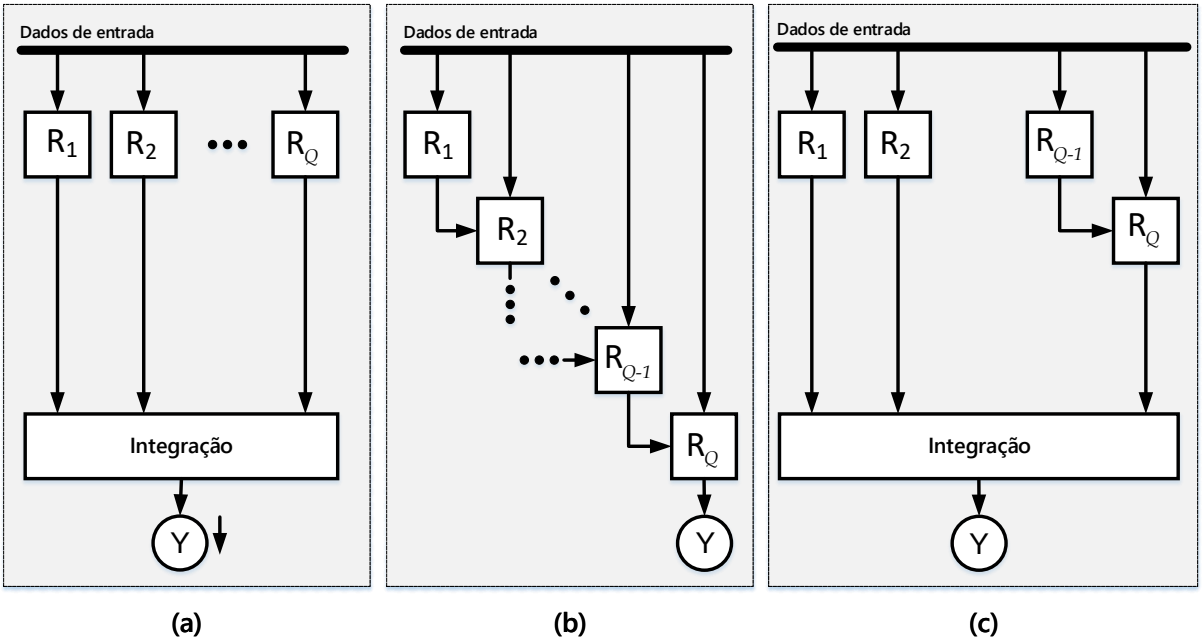

Figura 3 – Topologia dos elementos de um sistema baseado em múltiplos regressores (Adaptação de (LU, 1996)): (a) Topologia paralela. (b) Topologia em cascata. (c) Topologia hierárquica.

A topologia em cascata é representada na Fig. 3(b). Nessa topologia, o primeiro regressor recebe os dados de entrada, enquanto os demais regressores recebem os dados de entrada e a saída de seu antecessor. Por exemplo, o n-ésimo regressor pode tentar prever o erro que será gerado pelo $(n-1)$-ésimo regressor. Esse tipo de arquitetura não é recomendado para sistemas que necessitam de alto desempenho em tempo real, pois tanto o processo de treinamento quanto o processo de predição são realizados de forma serial (LU, 1996). Por último, na Fig. 3(c) temos a topologia hierárquica, a qual pode ser vista como uma combinação das duas topologias mencionadas anteriormente.

Além da topologia, outro importante tipo de categorização diz respeito ao tipo de cada elemento do conjunto. Caso todos os regressores sejam treinados com o mesmo algoritmo de ML, o conjunto é dito homogêneo e o algoritmo utilizado é denominado de regressor base. No caso de utilização de mais de um algoritmo, o conjunto é caracterizado como heterogêneo (ROONEY et al., 2004). Os sistemas implementados com conjuntos homogê-

neos são mais simples, quando comparados aos sistemas implementados com conjuntos heterogêneos (MENDES-MOREIRA et al., 2012). No entanto, a utilização de mais de um algoritmo de ML gera uma maior diversidade entre os elementos, o que se caracteriza como um fator chave para o desempenho do conjunto de regressores (MENDES-MOREIRA et al., 2012).

2.4 SELEÇÃO DE SUBCONJUNTOS DE REGRESSORES

A seleção de subconjuntos de regressores que irão participar do conjunto final pode ocorrer de duas formas, quais sejam, estática ou dinâmica. Na seleção estática, os regressores são selecionados uma única vez e o subconjunto formado é utilizado para todas as predições. Na abordagem dinâmica, um novo subconjunto de regressores é formado para cada instância de teste.

A Fig. 4 apresenta os esquemas de seleção estática e dinâmica de regressores. Em ambos os esquemas, temos seis regressores (R_1 a R_6) no conjunto inicial. Na Fig. 4(a), que representa a seleção estática, os regressores R_3, R_4 e R_5 são selecionados e esse conjunto é utilizado para realizar a predição de todas as instâncias de teste. Na Fig. 4(b), que mostra a seleção dinâmica, um novo subconjunto de três regressores é selecionado toda vez que é necessário realizar a predição de uma instância de teste. Por exemplo, na Fig. 4(b), temos a indicação do subconjunto formado pelos regressores R_x, R_y e R_z para a predição da primeira instância de teste.

Na seleção dinâmica, o subconjunto selecionado está diretamente relacionado à instância que será predita (instância de teste). Assim, teremos uma maior especialização dos regressores e, por consequência, uma expectativa de melhor desempenho (KO; SABOURIN; BRITTO, 2008). Algoritmos baseados na seleção dinâmica têm tido um melhor desempenho quando comparados a algoritmos que utilizam a seleção estática. No entanto, a utilização da seleção dinâmica requer um maior custo computacional, o que nos leva a considerar a complexidade da tarefa a ser executada como um guia para a escolha da estratégia de

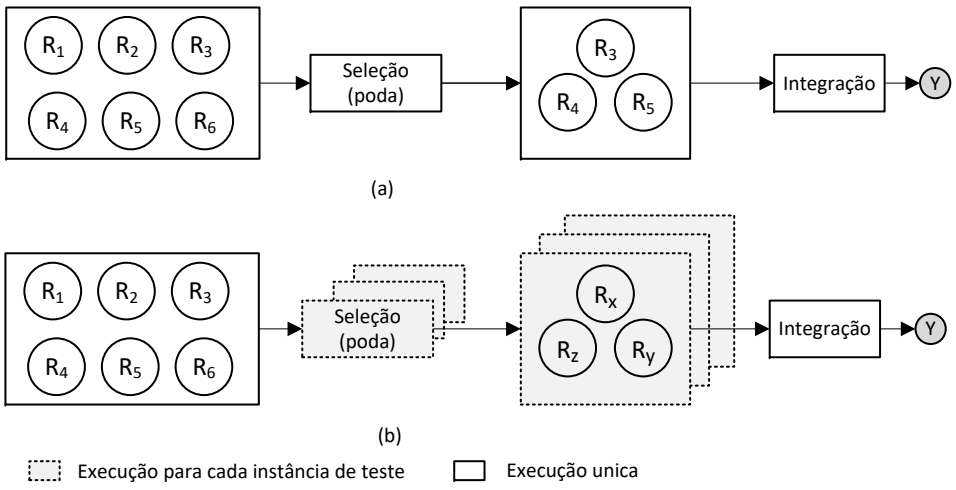

Figura 4 – Tipos de seleção de regressores: (a) Seleção estática. (b) Seleção dinâmica.

seleção (BRITTO; SABOURIN; OLIVEIRA, 2014).

A ideia base da seleção dinâmica é utilizar regressores que conseguem um alto desempenho nas regiões próximas à instância de teste (MENDES-MOREIRA et al., 2012), ou seja, o objetivo é selecionar especialistas em instâncias com características similares à instância que será predita. Nesse contexto, admite-se a presença de um oráculo (*Oracle*), que é responsável por fazer a seleção do subconjunto de regressores (KUNCHEVA; RODRIGUEZ, 2007).

2.4.1 Método KNORA

As pesquisas mais recentes em seleção dinâmica de modelos têm focado no problema de classificação (KRAWCZYK et al., 2017). Em (KO; SABOURIN; BRITTO, 2008), é proposto o método KNORA, cujo desempenho tem se mostrado superior ao de outros métodos de classificação (VRIESMANN et al., 2012).

O método KNORA utiliza a topologia em paralelo, ilustrada na Fig. 3(a), ou seja, todos os classificadores estão aptos a executar a mesma tarefa de classificação. O método KNORA utiliza somente um algoritmo de ML como base e, por essa razão, é classificado

Algoritmo 2: KNORA-ELIMINATE

Entrada:
C = conjunto inicial de classificadores;
h_K = número de instâncias vizinhas mais próximas de $\mathbf{x}_t$;
S_t = conjunto de teste;
S_v = conjunto de validação;
Saída:
$E_c[\,]$, conjuntos finais de classificadores (um para cada instância $\mathbf{x}_t$)

```
início
    k_v = h_K
    para cada instância de teste x_t^(i) ∈ S_t faça
        enquanto k_v > 0 faça
            S_d ← k_v instâncias vizinhas de x_t^(i) ∈ S_v
            para cada classificador c_j em C faça
                se c_j classificou corretamente todas as instâncias em S_d então
                    E_c[i] = E_c[i] ∪ c_j
                fim
            fim
            se E_c[i] == então
                k_v = k_v − 1
            senão
                quebre o laço
            fim
        fim
        se E_c[i] == então
            n_a ← número máximo de classificações corretas obtido em S_d, ∀c_i ∈ C
            E_c[i] ← todos classificadores c_j ∈ C com n_a acertos
        fim
    fim
fim
retorna E_c[ ]
```

como homogêneo. Em (KO; SABOURIN; BRITTO, 2008), são apresentadas quatro variantes do método KNORA: KNORA ELIMINATE, KNORA ELIMINATE-W, KNORA UNION e KNORA UNION-W.

Em todas as variantes do método KNORA, com o objetivo de induzir diversidade, os classificadores podem ser treinados com as técnicas de *Bagging* (BREIMAN, 1996), *Random Subspace* (HO, 1998) ou *AdaBoost* (FREUND; SCHAPIRE, 1996). Após o treinamento dos classificadores, o método KNORA seleciona um subconjunto de classificadores sempre que faz a predição de uma instância de teste. A seleção é baseada nas h_K instâncias vizinhas

Algoritmo 3: KNORA-UNION

Entrada:
C = conjunto inicial de classificadores;
h_K = número de instâncias vizinhas mais próximas de $\mathbf{x}_t$;
S_e = conjunto de teste;
S_v = conjunto de validação

Saída:
$E_c[\,]$, conjuntos finais de classificadores (uma para cada instância $\mathbf{x}_t$)

```
1  início
2      para cada instância de teste x_t^(i) ∈ S_t faça
3          S_d ← k_v instâncias vizinhas de x_t^(i) em S_v
4          para cada instância x_d ∈ S_d faça
5              para cada classificador c_j em C faça
6                  se c_j classificou corretamente x_d então
7                      E_c[i] = E_c[i] ∪ c_j
8                  fim
9              fim
10         fim
11     fim
12 fim
13 retorna E_c[ ]
```

mais próximas da instância de teste, considerando a distância euclidiana como métrica de distância. Para facilitar o entendimento dos algoritmos descritos a seguir, definimos as porções do *dataset* como conjuntos de treino, de validação e de teste, denotados por S_{tn} , S_v e S_t, respectivamente. De maneira análoga, as instâncias pertencentes a cada conjunto anteriormente definido são denotadas por $\mathbf{x}_{tn}$, $\mathbf{x}_v$ e $\mathbf{x}_t$. Adicionalmente, o conjunto de instâncias vizinhas mais próximas formado durante a seleção dinâmica será denotado por S_d e suas instâncias, por $\mathbf{x}_d$.

No Algoritmo 2, estão descritos os passos da variante KNORA ELIMINATE para a predição de um conjunto de teste S_t. Para a i-ésima instância de teste $\mathbf{x}_t^{(i)}$ de S_t , busca-se em S_v as h_K instâncias vizinhas mais próximas de $\mathbf{x}_t^{(i)}$ para formar o conjunto S_d. Os classificadores que conseguem acertar todos os padrões de S_d são adicionados ao conjunto final de classificadores, denotado por $E_c[i]$, selecionados para a i-ésima instância de teste $\mathbf{x}_t^{(i)}$. Caso não haja nenhum classificador que acerte todos os padrões, o índice k_v é decrementado. Caso k_v chegue a zero, o algoritmo verifica qual o classificador que

obteve o maior número de acertos n_a para as h_K instâncias vizinhas iniciais. Feito isso, todos os classificadores com n_a acertos são incluídos no conjunto $E_c[i]$.

O Algoritmo 3 descreve a variante KNORA UNION. De forma semelhante à variante KNORA ELIMINATE, para cada instância de teste $\mathbf{x}_t^{(i)}$ em S_t, temos a seleção de um conjunto S_d que contém as h_K instâncias vizinhas mais próximas de $\mathbf{x}_t^{(i)}$. Obtido o conjunto S_d, serão incluídos em $E_c[i]$ todos os classificadores que acertarem pelo menos um padrão $\mathbf{x}_d$. Além disso, o número de acertos de cada classificador determina a quantidade de votos que ele terá. Logo, quanto maior o número de classificações corretas de um determinado classificador, maior peso ele terá.

Finalmente, temos as variantes KNORA UNION-W e KNORA ELIMINATE-W, modificações das variantes KNORA UNION e KNORA ELIMINATE, respectivamente. A diferença é que as variantes KNORA UNION-W e KNORA ELIMINATE-W levam em consideração a distância Euclidiana do padrão de validação acertado em relação à instância de teste $\mathbf{x}_t^{(i)}$ que está sendo predita. Assim, classificadores que acertarem padrões em S_v mais próximos à instância de teste $\mathbf{x}_t^{(i)}$ terão maior peso. Maiores detalhes sobre o método KNORA e suas variantes podem ser encontrados em (KO; SABOURIN; BRITTO, 2008; VRIESMANN et al., 2012).

2.5 RESUMO DO CAPÍTULO

Nesse Capítulo, inicialmente foram apresentados os fundamentos teóricos que servem como base para justificar a utilização de sistemas com múltiplos regressores, com destaque aos aspectos computacionais, estatísticos e representativos. Em relação ao conjunto de regressores, foi utilizada a decomposição do erro para mostrar que uma maior diversidade entre os regressores do conjunto pode aumentar o desempenho do *pool*. Em seguida, foram detalhadas as técnicas *Bagging*, *Random Subspace* e *AdaBoost*, as quais são utilizadas para criação de conjunto de regressores com alta diversidade. Após isso, foram expostas as vantagens e desvantagens dos tipos de topologias utilizadas em sistemas com múltiplos

preditores, bem como as diferenças entre seleção estática e dinâmica para a escolha final dos regressores. Por último, foi descrito sucintamente o método KNORA, algoritmo que servirá como base para o sistema de regressão proposto neste trabalho.

3 COMBINAÇÃO PARA GERAÇÃO DO CONJUNTO INICIAL DE REGRESSORES

O desempenho de sistemas baseados em múltiplos regressores depende, entre outras coisas, de uma geração adequada do CIR, também chamado de *pool* de regressores. Embora as técnicas de geração apresentadas no Capítulo 2 tenham benefícios, elas possuem alguns pontos fracos. Por exemplo, a técnica de *Boosting* é muito sensível ao ruído, enquanto o algoritmo de *Bagging* reduz a variância, mas tem pouca efetividade na redução da tendência (*bias*) (KOTSIANTIS; KANELLOPOULOS, 2012). Por fim, a técnica *Random Subspace* pode ser difícil de ser aplicada individualmente em problemas com um baixa dimensionalidade (poucas *features*) (KOTSIANTIS; KANELLOPOULOS, 2012). Neste Capítulo, uma nova técnica baseada em combinação para a geração do *pool* de regressores é apresentada. Essa técnica constitui a base para o método de seleção dinâmica de regressores que será apresentado no Capítulo 4. Inicialmente, são expostos os motivos que levaram a concepção da proposta. Em seguida, são descritas as medidas de diversidade utilizadas para comparar a técnica proposta com outras técnicas geradoras de *pool*. Após isso, a implementação da proposta é detalhada. Por fim, são mostrados resultados numéricos da técnica proposta relacionados à diversidade e ao RMSE e, em seguida, comparados ao desempenho de outros algoritmos de geração de *pool* de regressores.

3.1 INTRODUÇÃO

Um ponto crucial para os sistemas baseados em múltiplos preditores é que erros diferentes sejam cometidos em qualquer amostra do conjunto de dados. A combinação de preditores será inútil se todos os componentes do conjunto fornecerem a mesma saída. Portanto, precisamos de diversidade nas decisões dos membros do grupo, especificamente quando eles estão cometendo erros.

A importância da diversidade para sistemas baseados em múltiplos preditores está bem

estabelecida na literatura (BREIMAN, 1996; HO, 1998; AVNIMELECH; INTRATOR, 1999; BROWN, 2004; BROWN; WYATT; TINO, 2005; MAO et al., 2019). Uma maneira de aumentar a diversidade em sistemas baseados em conjuntos é combinar técnicas tradicionais. Nesse contexto, várias iniciativas podem ser encontradas na literatura. Por exemplo, em relação à classificação, uma combinação de *Bagging* e *Random Subspace*, duas abordagens individuais para a geração do CIR, é proposta em (PANOV; DŽEROSKI, 2007). Neste trabalho, os autores mostram que a combinação tem desempenho semelhante ao algoritmo *Random Forest* (BREIMAN, 2001), com a vantagem de ser aplicável a qualquer algoritmo, ou seja, além da arvore de decisão, outros preditores podem ser utilizados como preditor base.

Em relação a problemas de regressão, uma técnica obtida pela combinação de *Bagging*, *Random Subspace* e *Boosting* é proposta em (KOTSIANTIS; KANELLOPOULOS, 2012). Neste caso, oito regressores são usados para cada técnica componente. Assim, temos três conjuntos separados, cada um gerado com uma técnica específica. A predição final é realizada calculando-se a média das predições de todos os componentes do conjunto. No artigo, os autores mostram que há um significativo ganho na diversidade do conjunto, pois a técnica diminui a correlação entre os erros realizados nas predições. Em (HADAVANDI; SHAHRABI; SHAMSHIRBAND, 2015), as técnicas *Boosting* e projeção do subespaço (*subspace projection*) são usadas para modelar problemas com múltiplos rótulos (*targets*) e alta dimensionalidade. É demonstrado que a técnica combinada é capaz de aumentar a diversidade e assim, melhorar a acurácia do conjunto de redes neurais.

A técnica proposta nesta tese, denotada por CBAS (*Combining-Based Technique*), utiliza uma combinação dos métodos *Bagging*, *Random Subspace* e *Boosting* com o propósito de aumento da diversidade. Para isso, a ideia consiste em criar preditores fracos, mas que são especialistas em diferentes subespaços do conjunto de treinamento.

3.2 MEDIDAS DE DIVERSIDADE

A diversidade pode ser vista como uma medida de desacordo, a qual está diretamente relacionada à capacidade de generalização (KUNCHEVA, 2004b). Cometer erros diferentes é essencial para a capacidade de generalização do conjunto, porque se todos os membros do conjunto fornecerem a mesma saída, sua combinação será inútil. Portanto, diversidade nas saídas dos regressores é uma característica desejável. Idealmente, os resultados dos regressores devem ser independentes ou correlacionados negativamente (ZHANG; MA, 2012).

Apesar da diversidade ser uma medida importante para a combinação de preditores, não existe uma definição formal para medir a diversidade. Em relação a conjunto de regressores, diferentes tipos de medidas de diversidade pareadas são apresentadas em (DUTTA, 2009), quais sejam, coeficiente de correlação, covariância, medida de desacordo (*disagreement measure*) e informação mútua. Neste trabalho, iremos utilizar o coeficiente de correlação e a medida de desacordo como medidas de diversidade. Enquanto a primeira é uma abordagem mais tradicional, a segunda é uma adaptação para regressão de uma medida usada na tarefa de classificação (DUTTA, 2009).

Com o objetivo de definir as medidas de diversidade de um conjunto de regressores, assuma que D é um conjunto de dados com N instâncias, tal que a i-ésima instância está associada a um valor contínuo y_i. Além disso, considere um regressor R_a, o qual possui uma saída Y^a que pode assumir N valores no conjunto $\mathbf{Y}^a = [y_1^a, \ldots, y_i^a, \ldots, y_N^a]$. Da mesma forma, considere também o regressor R_b com saída Y^b, que pode assumir valores no conjunto $\mathbf{Y}^b = \left[y_1^b, \ldots, y_i^b, \ldots, y_N^b\right]$. Isso posto, será apresentado o cálculo das medidas de diversidade anteriormente mencionadas, considerando os regressores R_a e R_b.

A primeira medida de diversidade abordada é o coeficiente de correlação linear ρ_{ab}

entre as saídas de dois regressores Y^a e Y^b, que é dado por

$$\rho_{ab} = \sum_{i=1}^{N} \left(\frac{(y_i^a - \mu_a)(y_i^b - \mu_b)}{\sqrt{\sum_{i=1}^{N}(y_i^a - \mu_a)^2 \sum_{i=1}^{N}(y_i^b - \mu_b)^2}} \right) , \tag{3.1}$$

em que μ_a e μ_b são as médias dos valores de $\mathbf{Y}^a$ e $\mathbf{Y}^b$, respectivamente. A diversidade entre Y^a e Y^b é inversamente proporcional a ρ_{ab}. Assim, um par de regressores com um baixo coeficiente de correlação é melhor para o conjunto, pois dessa maneira teremos uma maior diversidade (DUTTA, 2009).

A segunda medida de diversidade considerada é a medida de desacordo (*disagreement measure*). Essa medida foi originalmente usada para representar a diversidade entre classificadores básicos e complementares (SKALAK et al., 1996). Posteriormente, foi utilizada para medir a diversidade na técnica *Random Subspace* (HO, 1998). No contexto de classificação, define-se a medida de desacordo como a razão entre o número de observações em que um classificador está correto e o outro está incorreto em relação ao número total de observações (KUNCHEVA, 2004b).

Considere dois classificadores denotados por C_a e C_b. Além disso, assuma que "0" representa uma classificação correta e que "1" representa uma classificação incorreta. O número total de instâncias preditas por C_a e C_b, denotada por N_{ab}, é dado por

$$N_{ab} = N_{00} + N_{01} + N_{10} + N_{11} , \tag{3.2}$$

em que N_{00} é o número de instâncias classificadas corretamente por C_a e C_b, enquanto N_{01} (N_{10}) representa o número de previsões em que apenas C_a (C_b) acertou. Finalmente, N_{11} é o número de instâncias previstas incorretamente pelos dois classificadores.

Para problemas de regressão, a medida de desacordo é estendida como segue (DUTTA, 2009). Para cada instância, é obtido o desvio padrão σ dadas as predições feitas por todos os regressores. Assumindo que α representa o valor verdadeiro do rótulo (*target*), um valor predito β é considerado correto, se $\alpha - \sigma < \beta < \alpha + \sigma$. Caso o valor predito não esteja neste intervalo, a predição será considerada incorreta. Assim, a medida de desacordo d_{ab}

é dada por

$$d_{ab} = \frac{N_{01} + N_{10}}{N_{ab}} \,. \tag{3.3}$$

Por último, vale ressaltar que as medidas de diversidade mencionadas anteriormente são medidas pareadas, ou seja, são medidas de diversidade entre dois regressores. Neste trabalho, vamos considerar a média de todos os pares possíveis de regressores para calcular a diversidade do conjunto.

3.3 TÉCNICA PROPOSTA

Nesta Seção, será apresentada detalhadamente uma nova proposta de técnica baseada em combinação para a geração do *pool*, denotada por CBAS (*Combining-Based Technique*). Para aumentar a diversidade, novos elementos do CIR são criados com base na combinação dos algoritmos *Bagging*, *Boosting* e *Random Patches*, que serão denotados, deste ponto em diante, por BAGG, BOOS e RAPT, respectivamente. O algoritmo RAPT é uma derivação do *Random Subspace*. Sua principal diferença em relação ao *Random Subspace* é que no RAPT há uma amostragem aleatória das instâncias e das *features* para o treinamento de cada preditor, enquanto no algoritmo *Random Subspace*, só as *features* são amostradas.

A técnica proposta é representada no Algoritmo 4 e possui as seguintes variáveis de entrada: M, o número de regressores no *pool*; D, o conjunto de dados de treinamento; S_b, o número de amostras de instância para a técnica BAGG; F_{min} e F_{max}, números mínimo e máximo de *features* para a técnica RAPT, respectivamente. A ideia principal da proposta é aumentar a diversidade usando estratégias empregadas nas técnicas BAGG, RAPT e BOOS. A técnica CBAS retorna um conjunto Ψ com M regressores, cada um treinado com um conjunto de treinamento diferente.

Considerando R^i e D^i como o i-ésimo regressor treinado e seu conjunto de treinamento, respectivamente, diferentes estratégias de geração de *pool* são usadas, dependendo da quantidade de regressores gerados anteriormente. A cada dois regressores gerados usando

Algoritmo 4: CBAS: Técnica de combinação para geração do CIR (*pool*).

Entrada:
D, Conjunto de dados;
M, Número de regressores;
S_b, Número de amostras BAGG;
F_{min}, Número minimo de *features* RAPT
F_{max}, Número máximo de *features* RAPT
Saída:
1 Ψ = CIR (*pool* de regressores)
2 **início**
3 $\Psi \leftarrow$ Lista vazia de regressores
4 $\mathbf{w}_u \leftarrow$ Vetor de pesos de probabilidade uniforme
5 **para** $i = 1$ ***até*** M **faça**
6 **se** *i não é múltiplo de 3* **então**
7 $D^i \leftarrow$SAMPLE (D, S_b, F_{min}, F_{max}, $\mathbf{w}_u$);
8 **senão**
9 $\Gamma \leftarrow$ instâncias que não foram usadas no treinamento dos regressores anteriores R^{i-1} e R^{i-2};
$\mathbf{w}_e \leftarrow$ Soma dos erros de predição obtidos por R^{i-1} e R^{i-2} para cada instância de Γ;
$D^i \leftarrow$SAMPLE (Γ, S_b, F_{min}, F_{max}, $\mathbf{w}_e$);
10 **fim**
11 Treinar o regressor R^i usando D^i;
12 Incluir R^i in Ψ;
13 **fim**
14 **retorna** Ψ
15 **fim**

as técnicas BAGG e RAPT, um terceiro regressor é produzido por meio de uma combinação do algoritmo BOOS com as duas técnicas anteriores. Em outras palavras, quando o índice i não é múltiplo de 3, as técnicas BAGG e RAPT são usadas para construir o i-ésimo regressor. Caso contrário (quando i é múltiplo de 3), o algoritmo BOOS é usado em combinação com as técnicas BAGG e RAPT para produzir o i-ésimo regressor. O objetivo da combinação é criar preditores fracos, especialistas em diferentes subespaços do conjunto de dados de treinamento (instâncias e *features*). O ponto-chave da ideia baseada em combinação proposta neste trabalho é indicado pelas linhas $4 - 12$ do Algoritmo 4.

As técnicas BAGG e RAPT são aplicadas usando a função `SAMPLE`, que é descrita no Algoritmo 5. Os parâmetros de entrada da função `SAMPLE` são D_s, a fonte de amostras; $\mathbf{w}$,

Algoritmo 5: Função de amostra usando pesos de probabilidade e combinação das técnicas BAGG e RAPT.

Entrada:
D_s, Conjunto fonte de amostras;
S_b, Número de amostras BAGG;
F_{min}, Número mínimo de *features* RAPT
F_{max}, Número máximo de *features* RAPT
w, vetor de pesos de probabilidade das amostras
Saída:
Δ, Novo conjunto de treino

```
1 Função SAMPLE (D_s, S_b, F_min, F_max, w)
2 |  χ ← Obter S_b amostras de D_s com reposição, usando o vetor w;
3 |  Φ ← Obter entre F_min e F_max amostras das features de D_s sem reposição
4 |  Δ ← Novo conjunto com instâncias χ e features Φ
5 |  retorna Δ
6 fim
```

o vetor de pesos de probabilidade das amostras; e, finalmente, S_b, F_{min} e F_{max}, os mesmos parâmetros adotados no Algoritmo 4. A função `SAMPLE` seleciona instâncias χ e *features* Φ de D_s para criar um novo conjunto de dados Δ. Como pode ser visto no Algoritmo 5, as instâncias χ são tomadas aleatoriamente com substituição e usando o vetor de peso de probabilidade **w**, enquanto as *features* Φ são selecionadas sem substituição.

Quando o regressor R^i é treinado sem o algoritmo BOOS (quando i não é múltiplo de 3), o vetor **w** é assumido como $\mathbf{w}_u$, o qual representa uma distribuição uniforme. Em outras palavras, todas as instâncias em D_s terão a mesma probabilidade de serem selecionadas. Além disso, o conjunto completo de treinamento D é usado como fonte das amostras linha 7 do Algoritmo 4). Diferentemente, quando o algoritmo BOOS é usado para treinamento (i é múltiplo de 3), um conjunto de dados Γ é criado com instâncias que não foram usadas para treinar os dois regressores anteriores (R^{i-1} e R^{i-2}), ou seja, $\Gamma = D - (D^{i-1} \cup D^{i-2})$. O conjunto de dados Γ é usado como entrada para a função `SAMPLE` , um primeiro aspecto que difere da abordagem sem a técnica BOOS. Um segundo aspecto diz respeito aos pesos de probabilidade, calculados a partir dos erros cometidos pelos regressores R^{i-1} e R^{i-2} ao prever as instâncias contidas em Γ.

Seja N o número de instâncias do conjunto Γ, tal que $\Gamma = [\gamma_1, \ldots, \gamma_j, \ldots, \gamma_N]$. Define-

se um vetor de peso de probabilidades $\mathbf{w} = [w_1, \ldots, w_j, \ldots, w_N]$, no qual o j-ésimo peso de probabilidade w_j , associado com a j-ésima instância de Γ, é dado por

$$w_j = \varepsilon_j^{i-1} + \varepsilon_j^{i-2}, \tag{3.4}$$

em que ε_j^{i-1} e ε_j^{i-2} são os erros de predição dos regressores R^{i-1} e R^{i-2} , respectivamente, para a instância γ_j. Em outras palavras, a probabilidade de selecionar γ_j é proporcional à soma dos erros de previsão relacionados aos dois regressores anteriores. Portanto, o novo regressor R^i se concentrará em instâncias que foram pouco aprendidas. Finalmente, o conjunto de dados de treinamento D^i é construído a partir do conjunto Γ e, após o treinamento, o regressor R^i é incluído no *pool* de regressores Ψ.

3.4 RESULTADOS NUMÉRICOS

A técnica proposta foi implementada com o uso da linguagem R e do pacote *Caret* (KUHN; JOHNSON, 2013). Para todos os experimentos, foi usado um conjunto com 36 regressores ($M = 36$) e o algoritmo k-vizinhos mais próximos (k-NN, *k-nearest neighbors*) como regressor base. Para todos os regressores, a técnica de reamostragem *10-cross-validation* (10-validação-cruzada) foi usada para o ajuste do modelo *(tuning)*, enquanto o melhor k para o k-NN foi pesquisado no intervalo [1; 11]. O algoritmo k-NN foi escolhido porque seu custo computacional tem um relacionamento direto com as propriedades do conjunto de dados (número de instâncias e *features*), o que nos permite analisar a relação custo-benefício associada ao custo computacional. Ademais, a proposta pode ser facilmente estendida a outros regressores, tais como ANNs ou regressão com vetores de suporte (SVRs, *support vector regression*).

Na Tab. 2, temos os dez *datasets* públicos usados nos experimentos. Esses conjuntos de dados abrangem diferentes tipos de problemas com dados contínuos, discretos e categóricos. Além disso, eles diferem no número de instâncias e *features*.

Tabela 2 – *Datasets* públicos usados nos experimentos com a técnica CBAS.

Dataset	Número de *features*	Número de instâncias
Airquality	5	153
CPUs	6	209
Autompg	7	392
Concrete	8	1030
Abalone	8	4177
Carseats	10	400
Wage	11	3000
Forestfires	12	513
Boston Housing	13	506
Hitters	19	322

As medidas de diversidade definidas na Seção 3.2 são usadas para avaliar a diversidade. Para cada conjunto de dados listado na Tab. 2, 90% das instâncias são usadas para geração do *pool* (conjunto de treinamento), enquanto 10% são usados para avaliar a diversidade. *Pools* de regressores, usando as técnicas BAGG, RAPT, BOOS e a proposta CBAS são criados para cada *dataset* com o objetivo de comparar a diversidade. As técnicas BAGG, RAPT e BOOS são referidas como nossos *benchmarks*. A melhor técnica para cada *dataset* e cada métrica de diversidade é destacada em negrito na apresentação dos resultados. Para verificar se as diferenças entre os desempenhos das técnicas de geração de *pool* são estatisticamente relevantes, o teste de Friedman é aplicado juntamente com o teste *post-hoc* de Nemenyi (DEMSAR, 2006). Quando a diferença entre as técnicas não é estatisticamente relevante, temos um empate. Nesse caso, mais de uma técnica é indicada em negrito.

A Tab. 3 mostra os coeficientes de correlação obtidos para todos os *datasets*, considerando os *benchmarks* e a técnica CBAS. Também na Tab. 3, temos uma comparação de vitórias, empates e derrotas (*win-draw-lose*, W-D-L) entre as técnicas de geração de *pool* para cada *dataset*. Considerando que quanto menor o coeficiente de correlação, maior a diversidade, a técnica CBAS vence em oito dos dez *datasets*. É possível perceber que, para

Tabela 3 – Coeficiente de correlação obtido para os 10 *datasets* considerando as técnicas *Bagging* (BAGG), *Random Patches* (RAPT), *Boosting* (BOOS) e a técnica proposta (CBAS).

Dataset	BAGG	RAPT	BOOS	CBAS
Airquality	0,91	0,76	0,84	**0,66**
CPUs	0,90	0,87	0,86	**0,80**
Autompg	0,91	0,80	0,84	**0,78**
Concrete	0,84	0,68	0,76	**0,59**
Abalone	0,74	0,53	0,63	**0,48**
Carseats	0,52	0,26	0,38	**0,20**
Wage	0,91	**0,50**	0,82	0,61
Forestfires	0,38	0,15	0,23	**0,08**
Boston Housing	0,89	***0,74**	0,85	***0,73**
Hitters	0,61	0,57	0,49	**0,46**
W-D-L	0-0-10	1-1-8	0-1-9	8-1-1

(*) Diferença estatisticamente irrelevante

o *dataset Boston Housing*, a diferença entre as técnicas RAPT e CBAS não é estatisticamente relevante, o que significa que as duas abordagens são equivalentes.

A Tab. 4 ilustra as medidas de desacordo obtidas para todos os *datasets*, considerando as técnicas BAGG, RAPT, BOOS e CBAS. Nesse caso, quanto maior a discordância, maior a diversidade. Assim, os resultados se comportam de maneira oposta quando comparados aos obtidos pelo coeficiente de correlação. Similar ao caso anterior, a técnica CBAS é a vencedora nos mesmos oito *datasets*. Além disso, nossa proposta é equivalente à técnica RAPT para o *dataset Boston Housing*, uma vez que a diferença entre eles não é estatisticamente relevante, como foi observado para o coeficiente de correlação.

A Tab. 5 exibe o RMSE alcançado em todos os *datasets*, considerando a técnica CBAS e a abordagem proposta em (KOTSIANTIS; KANELLOPOULOS, 2012), na qual são gerados três conjuntos de regressores, um para cada técnica de reamostragem (BAGG, RAPT e BOOS). Para a predição final, é utilizada uma metodologia de média, ou seja, o valor predito é a média das saídas dos regressores (segunda coluna da Tab. 5). Na terceira coluna da Tab. 5, temos o RMSE para a mesma metodologia, no entanto, usando a abordagem

Tabela 4 – Medidas de descordo obtidas para os 10 *datasets*, considerando as técnicas *Bagging* (BAGG), *Random Patches* (RAPT), *Boosting* (BOOS) e a técnica proposta (CBAS).

Dataset	BAGG	RAPT	BOOS	CBAS
Airquality	0,06	0,13	0,10	**0,18**
CPUs	0,04	0,06	0,07	**0,10**
Autompg	0,04	0,11	0,08	**0,12**
Concrete	0,09	0,21	0,14	**0,25**
Abalone	0,12	0,23	0,16	**0,24**
Carseats	0,26	0,40	0,34	**0,43**
Wage	0,04	**0,22**	0,08	0,18
Forestfires	0,34	0,41	0,40	**0,45**
Boston Housing	0,05	***0,11**	0,06	***0,12**
Hitters	0,23	0,22	0,25	**0,28**
W-D-L	0-0-10	1-1-8	0-0-10	8-1-1
(*) Diferença estatisticamente irrelevante				

CBAS para gerar um único conjunto de regressores. Semelhante ao cálculo das medidas de diversidade, o RMSE é obtido em relação a 10% das instâncias (conjunto de teste). Os testes de Friedman e *post-hoc* de Nemenyi mostram que a diferença entre as técnicas é estatisticamente irrelevante, o que significa que as técnicas são equivalentes [1]. Apesar disso, a nossa proposta reduz o número de regressores utilizados sem comprometer a precisão das previsões.

Outro aspecto importante para comparar os métodos de geração de *pools* é o custo computacional. A Tab. 6 exibe o custo computacional médio normalizado para cada técnica de geração e cada *dataset*. Para todas as técnicas, é assumido o uso do mesmo *hardware*. Para cada técnica e *dataset*, o menor tempo consumido é igual a $1,0$ (valor de referência destacado em negrito) e os outros tempos são comparados a ele. Por exemplo, considerando o *dataset Airquality*, o custo computacional das técnicas RAPT, BOOS e CBAS é 5% maior que o custo da técnica BAGG. Em relação à comparação W-D-L, ob-

[1] As medidas de diversidade e os valores de RMSE utilizados para os testes estatísticos estão disponíveis em <https://github.com/timotrob/CBASDiversity>.

Tabela 5 – Desvio do erro médio quadrático obtida para os 10 *datasets*, considerando a geração de três *pools* (KOTSIANTIS; KANELLOPOULOS, 2012) e a técnica CBAS.

Dataset	3 *pools* de regressores	CBAS
Airquality	***12,17**	***10,07**
CPUs	***37,15**	***26,53**
Autompg	***2,96**	***3,15**
Concrete	***7,58**	***7,86**
Abalone	***2,58**	***2,40**
Carseats	***2,22**	***1,94**
Wage	***12,54**	***8,30**
Forestfires	***1,38**	***1,34**
Boston Housing	***3,22**	***3,34**
Hitters	***550,12**	***541,44**
W-D-L	0-10-0	0-10-0
(*) Não é estatisticamente relevante		

Tabela 6 – Custo computacional médio normalizado para as técnicas BAGG, RAPT, BOOS e CBAS aplicadas aos 10 *datasets* considerados.

Dataset	BAGG (*Bagging*)	RAPT (*Random Patches*)	BOOS (*Boosting*)	CBAS
Airquality	**1,00**	1,05	1,05	1,05
CPUs	1,07	**1,00**	1,06	1,07
Autompg	1,00	**1,00**	1,06	1,06
Concrete	1,09	1,08	**1,00**	1,08
Abalone	**1,00**	1,16	1,22	1,16
Carseats	1,08	**1,00**	1,08	1,08
Wage	**1,00**	1,17	1,13	1,20
Forestfires	**1,00**	1,08	1,08	1,09
Boston Housing	1,10	1,11	**1,00**	1,10
Hitters	**1,00**	1,10	1,11	1,10
W-D-L	5-0-5	3-0-7	2-0-8	0-0-10

servamos que a técnica BAGG possui o menor custo computacional entre as estratégias, ganhando em 50% dos *datasets*. Para os *datasets* restantes, a técnica RAPT vence em três deles, enquanto a técnica BOOS é a menos complexa para os *datasets Concrete* e *Boston Housing*. Embora a técnica CBAS não tenha o menor custo computacional para nenhum dos *datasets*, ela nem sempre apresenta o maior custo.

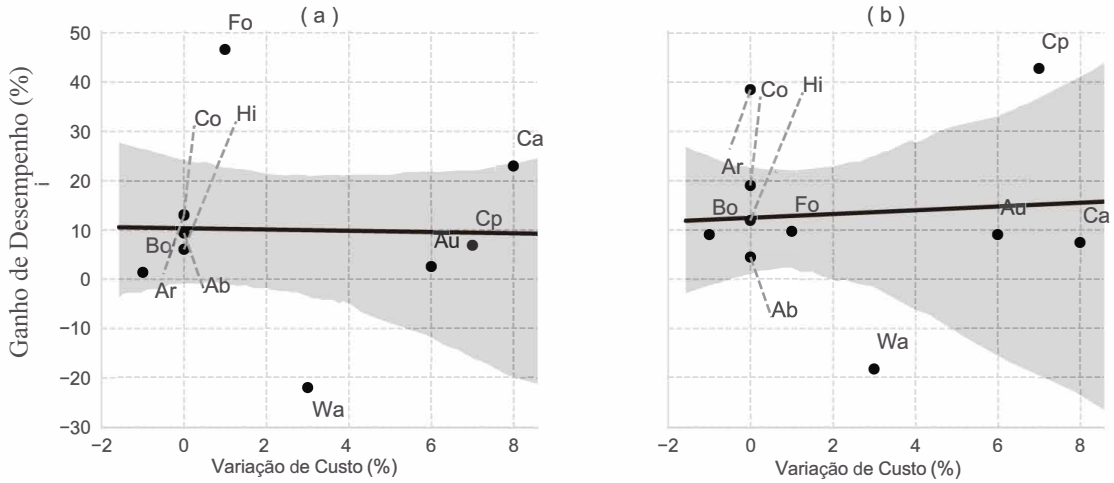

Figura 5 – Ganho de desempenho (%) *versus* variação de custo computacional (%) entre a técnica CBAS e o melhor *benchmark* (BAGG, RAPT ou BOOS): (a) Coeficiente de correlação. (b) Medida de desacordo. *Datasets*: `Ar` - *Airquality*; `Cp` - *Cpus*; `Au` - *Autompg*; `Co` - *Concrete*; `Ab` - *Abalone*; `Ca` - *Carseats*; `Wa` - *Wage*; `Fo` - *Forestfires*; `Bo` - *Boston Housing*; `Hi` - *Hitters*.

A relação entre o ganho de desempenho e a variação do custo computacional entre a técnica CBAS e o melhor *benchmark* é exibida nas Figs. 5(a) e 5(b) para o coeficiente de correlação e a medida de desacordo, respectivamente. Para cada *dataset* (representado por um ponto identificado), a variação do custo computacional e o ganho de desempenho, ambos em %, da técnica CBAS são comparados com os mesmos parâmetros do melhor *benchmark* (BAGG, RAPT ou BOOS). Ainda na Fig. 5, a parte cinza representa o intervalo de confiança (95%) considerando uma regressão linear.

Em geral, observa-se que, dependendo do *dataset* e da medida de diversidade, a relação complexidade-desempenho apresenta um comportamento diferente. Por exemplo, para o coeficiente de correlação como medida de diversidade e assumindo o *dataset Forestfires*, o melhor *benchmark* é a técnica RAPT ($0,15$ na Tab. 3). Nesse caso, podemos ver na Fig. 5(a), que a técnica CBAS oferece aproximadamente $46,7\%$ de ganho de desempenho associado a 1% de variação de custo, ou seja, praticamente sem aumentar a complexidade. Considerando novamente o coeficiente de correlação, mas agora o *dataset Carseats* (RAPT é o melhor *benchmark* de acordo com a Tab. 3), vemos que a técnica CBAS fornece

aproximadamente $23,1\%$ de ganho de desempenho ao preço de um aumento de 8% no custo computacional. Finalmente, para ambas as medidas de diversidade e assumindo o *dataset Wage*, o melhor *benchmark* é a técnica RAPT ($0,50$ na Tab. 3 e $0,22$ na Tab. 4), ainda melhor que a técnica CBAS. Nesse caso, temos uma diminuição do ganho de desempenho em torno de 20%, com um aumento de 3% em termos de custo computacional. Face ao exposto, é possível afirmar que a relação complexidade-desempenho da técnica CBAS depende do *dataset* e da medida da diversidade. Na maioria dos casos, a técnica CBAS forneceu um ganho de desempenho com custos computacionais maiores ou equivalentes.

3.5 RESUMO DO CAPÍTULO

Nesse Capítulo, uma nova técnica de geração de *pool* baseada em combinação foi proposta para aumentar a diversidade de sistemas de aprendizagem de múltiplos preditores em problemas de regressão. Algoritmos tradicionais baseados em conjuntos, como *Bagging*, *Random Patches* e *Boosting*, foram usados como *benchmark* para comparação. Além disso, no que diz respeito ao RMSE, nossa proposta é comparada a uma técnica baseada em combinação de regressores apresentada em (KOTSIANTIS; KANELLOPOULOS, 2012). Para todos os experimentos, o algoritmo k-NN foi assumido como regressor base e foram considerados dez *datasets* públicos que contemplavam diferentes tipos de problemas. Para avaliar a diversidade da técnica proposta, foram utilizados o coeficiente de correlação e a medida de desacordo. Os resultados numéricos mostraram que a técnica proposta (CBAS) venceu as abordagens individuais de geração de *pool* em 80% dos *datasets* considerados. Nesses casos, a técnica CBAS proporcionou um ganho de desempenho com custos computacionais maiores ou equivalentes. Em relação ao RMSE, a técnica CBAS alcançou um desempenho equivalente, sem comprometer a precisão das previsões, quando comparada à abordagem combinada usando três *pools* de regressores, proposta em (KOTSIANTIS; KANELLOPOULOS, 2012).

4 SELEÇÃO DINÂMICA DE REGRESSORES

Nos últimos anos, a seleção dinâmica de preditores tem demonstrado ótimo desempenho na tarefa de classificação quando comparados com outros métodos de classificação (LI et al., 2019a; BALLARD; WANG, 2016). Nesse contexto, destaca-se o método KNORA, o qual foi detalhado na Subseção 2.4.1. Neste Capítulo, é apresentado o método DySEREK, uma adaptação do método KNORA para regressão. Inicialmente, são descritas as peculiaridades que devem ser observadas na adaptação de modelos de classificação para regressão. Em seguida, é apresentada uma descrição detalhada do método proposto e suas quatro variantes. Por último, uma comparação do método DySEREK com outros algoritmos de ML é indicada, além de uma análise do desempenho do método proposto em função do ajuste de seus parâmetros.

4.1 CLASSIFICAÇÃO E REGRESSÃO

As tarefas de classificação e regressão podem ser consideradas dentre as mais utilizadas em ML. Ambas as tarefas são consideradas supervisionadas, pois utilizam um conjunto adequado de entrada-saída como dados de treino (HAYKIN, 1999). A tarefa de classificação tem como objetivo atribuir uma classe, dentre um número predeterminado de classes (categorias), a um padrão de teste apresentado (HAYKIN, 1999). Na regressão, temos a criação de uma função hipótese, a qual é uma aproximação da função real que relaciona entrada e saída (HAYKIN, 1999). Assim, na classificação temos uma saída discreta (categoria ou rótulo), enquanto na regressão temos um valor real contínuo como saída. Devido a essas diferenças, não é possível aplicar diretamente um método de classificação em regressão e vice-versa.

Na área de ML, quando um algoritmo se destaca no contexto de classificação, é comum que surjam várias adaptações desse algoritmo para tarefa de regressão. Por exemplo, o

algoritmo SVM, utilizado nos problemas de classificação, foi adaptado para a tarefa de regressão por meio da SVR (VAPNIK., 1995). A SVR pode ser considerada um caso de sucesso, considerando a adaptação de um classificador para regressor, pois apresentou excelentes resultados na tarefa de regressão (SMOLA; SCHÖLKOPF, 2004). No entanto, existem adaptações que não foram bem sucedidas. Por exemplo, o *AdaBoost.R* (FREUND; SCHAPIRE, 1995) foi a primeira adaptação do *AdaBoost* para regressão, mas apresentou sérios problemas devido ao alto custo computacional (MENDES-MOREIRA et al., 2012).

A adaptação realizada deve levar em conta similaridades e diferenças nas tarefas de classificação e regressão. Nos sistemas que utilizam múltiplos algoritmos de ML, essa diferença é latente na fase de integração das saídas. Por exemplo, não é possível utilizar a média ou a mediana para combinar as saídas de classificadores. Da mesma maneira, não há como fazer uso da técnica do voto majoritário para combinar as saídas de regressores.

Uma vez apresentadas as principais diferenças entre as tarefas de regressão e classificação, podemos passar às métricas de distância, as quais são utilizadas na seleção dinâmica dos regressores no método proposto.

4.2 MÉTODO DYSEREK

4.2.1 Descrição

O método proposto aqui nesta tese é denominado de seleção dinâmica de subconjunto de regressores baseado no k oráculos mais próximos *(DySEREK, dynamic selection of regressors ensemble using k-nearest-oracles)* e basicamente consiste em uma adaptação do método KNORA, detalhado na Subseção 2.4.1. Além dos bons resultados reportados em problemas de classificação, a escolha do método KNORA ocorreu devido à facilidade de adaptação do algoritmo para tarefa de regressão quando comparado a outros algoritmos de seleção dinâmicas detalhados em (CRUZ; SABOURIN; CAVALCANTI, 2018). No método KNORA, a figura do oráculo é o ponto central do método, pois os classificadores são selecionados a partir do número de acertos obtidos na predição das h_K instâncias vizinhas

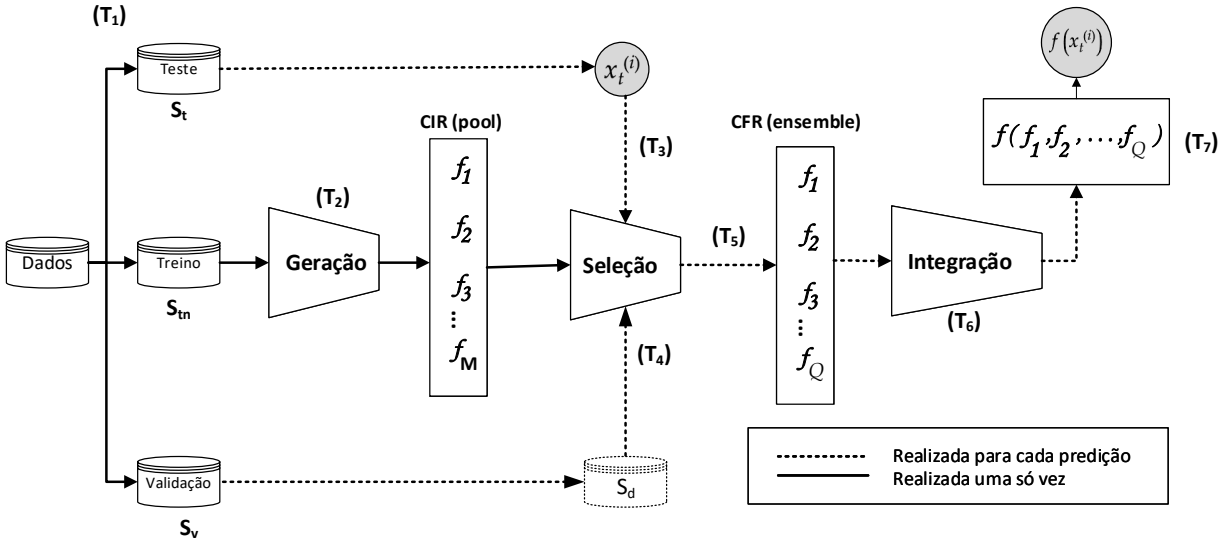

Figura 6 – Diagrama do funcionamento básico do método DySEREK.

mais próximas da instância de teste, encontradas a partir do conjunto de validação.

No método DySEREK, para possibilitar a seleção dos regressores pelo oráculo, as predições dos regressores precisam ser avaliadas como boas ou ruins. Na tarefa de regressão, temos uma saída numérica contínua, que geralmente é representada em notação de ponto flutuante. Uma vez que não é recomendável verificar a igualdade entre o valor predito e valor real (alvo) (OLIVEIRA; STEWART, 2006), será utilizado um limiar de erro para classificar a predição realizada como boa ou ruim. O cálculo do limite de erro é baseado na mediana dos erros das predições realizadas, conforme será detalhado posteriormente.

A Fig. 6 ilustra o funcionamento básico do método proposto. A descrição é baseada em sete passos (T_1 a T_7) que ocorrem de forma sequencial. Dessa forma, os passos podem ser interpretados como marcos temporais. No primeiro instante (T_1), o *dataset* é dividido em três conjuntos, quais sejam, conjuntos de treino, de validação e de teste, denotados por S_{tn}, S_v e S_t, respectivamente.

A geração do CIR (*pool*) é realizada no instante T_2 a partir do conjunto de treinamento S_{tn}. Para induzir diversidade, o método DySEREK emprega a técnica CBAS, descrita no Capítulo 3, que utiliza uma combinação dos algoritmos BAGG, RAPT e BOOS, com o

objetivo de aumentar a diversidade do *pool* de regressores. Além da combinação dessas técnicas de manipulação de dados, a indução de diversidade via parâmetros também é utilizada, pois cada regressor utiliza um conjunto diferente de parâmetros.

Voltando à Fig. 6, o instante T_3 se refere à introdução da instância $\mathbf{x}_t^{(i)}$, i-ésima instância de teste do conjunto S_t que será predita. No instante T_4, busca-se as h_K instâncias vizinhas mais próximas de $\mathbf{x}_t^{(i)}$ no conjunto S_v. Tais instâncias irão compor o subconjunto S_d. Para a seleção das instâncias de S_d, o método DySEREK utiliza a distância Euclidiana como métrica de distância. A seleção das h_k instâncias vizinhas é realizada para cada nova predição, conforme pode ser visto pelo fluxo representado pela linha tracejada na Fig. 6.

No instante T_5, é selecionado o subconjunto de regressores que fará parte do CFR (*ensemble*) usado para a predição de $\mathbf{x}_t^{(i)}$. Esse subconjunto é escolhido com base na qualidade das predições realizadas pelos regressores em relação às instâncias contidas em S_d. São consideradas boas e ruins, as predições situadas acima e abaixo do limite de erro, respectivamente. De maneira semelhante ao método KNORA, o DySEREK possui duas variantes em relação ao tipo de seleção dos regressores, quais sejam, as variantes DySEREK-*Elimination* e DySEREK-*Union*. Tais variantes serão devidamente explicadas na Subseção 4.2.2. Assim, a seleção dos regressores depende do limite de erro e da variante DySEREK adotada.

Ainda na Fig. 6, temos a combinação dos regressores selecionados no instante T_6 . A integração das saídas dos regressores pode ocorrer nas formas ponderada ou simples (sem pesos). A utilização de pesos permite a criação de duas variantes adicionais do método DySEREK: a DySEREK-*Elimination*-W e a DySEREK-*Union*-W, que também serão descritas na Subseção 4.2.2. Finalmente, no instante T_7, é calculada a saída predita da i-ésima instância de teste $\mathbf{x}_t^{(i)}$, denotada por $f(\mathbf{x}_t^{(i)})$. O algoritmo retorna ao instante T_3, obtendo a próxima instância de teste e uma nova execução dos passos T_4 a T_7 é realizada.

Explicado o funcionamento geral do método DySEREK, podemos passar ao cálculo

dos limites de erro, que servem como critério para a seleção dos regressores. No método DySEREK, o limite de erro tem importância fundamental na seleção dos regressores, uma vez que estabelece um critério a partir do qual uma predição é considerada boa ou ruim. Antes de estabelecermos o limite de erro, precisamos definir o vetor de erros $\mathbf{E}^{(i)} = [e_r^{(i)}], r = 1, 2, \ldots, M$, no qual cada elemento $e_r^{(i)}$ é o valor absoluto do erro de predição do r-ésimo regressor para a i-ésima instância de validação, tal que

$$e_r^{(i)} = |t_v^{(i)} - f_r(\mathbf{x}_v^{(i)})|\,,\ r = 1, 2, \ldots, M;\ i = 1, 2, \ldots, V\ , \tag{4.1}$$

em que $t_v^{(i)}$ é o valor real (*target*) da i-ésima instância de validação $\mathbf{x}_v^{(i)}$, $f_r(\mathbf{x}_v^{(i)})$ é o valor predito pelo r-ésimo regressor para a instância $\mathbf{x}_v^{(i)}$, M é o número de regressores do *pool*, V é o número de instâncias de validação e $|\cdot|$, o operador módulo. Dado o vetor de erros $\mathbf{E}^{(i)}$, podemos então definir o limite de erro para instância $\mathbf{x}_v^{(i)}$, denotado por γ_i, como

$$\gamma_i = M_D(\mathbf{E}^{(i)})\,, \tag{4.2}$$

em que $M_D(\mathbf{E}^{(i)})$ é a mediana dos elementos do vetor $\mathbf{E}^{(i)}$. De posse do limite de erro, as variantes do método DySEREK podem ser apresentadas, uma vez que a formação do CFR em cada uma delas depende diretamente da qualidade das predições realizadas em relação aos limites de erro.

4.2.2 Variantes do Método DySEREK

No método DySEREK, o esquema utilizado para a seleção dos regressores e a maneira como são combinadas suas saídas dão origem a quatro variantes do método. Todas as variantes têm como base as variações do método KNORA descritas em (KO; SABOURIN; BRITTO, 2008). As quatro variantes do método KNORA foram adaptadas para a tarefa de regressão, gerando quatro diferentes métodos DySEREK (variantes), conforme descritos a seguir.

Variante DySEREK-*Elimination*

No Algoritmo 6, estão descritos os passos da variante DySEREK-*Elimination* para a

Algoritmo 6: DySEREK-*Elimination*

Entrada:
R, *pool* de regressores;
h_K, número de instâncias vizinhas obtidas a partir do conjunto S_v;
S_t, conjunto de teste;
S_v, conjunto de validação;

Saída:
$E_R[\,]$, conjuntos finais de regressores (um para cada instância de teste $\mathbf{x}_t$)

```
1  início
2      k_v = h_K
3      para cada instância de teste x_t^(i) ∈ S_t faça
4          enquanto k_v > 0 faça
5              S_d ← k_v instâncias vizinhas de x_t^(i) ∈ S_v
6              para cada regressor r_j em R faça
7                  se r_j fez predição boa para todas as instâncias em S_d então
8                      E_R[i] = E_R[i] ∪ r_j
9                  fim
10             fim
11             se E_R[i] == então
12                 k_v = k_v − 1
13             senão
14                 quebre o laço
15             fim
16         fim
17         se E_R[i] == então
18             g ← número máximo de predições boas obtidas em S_d, ∀r_i ∈ R
19             E_R[i] ← todos regressores r_j ∈ R com g predições boas
20         fim
21     fim
22 fim
23 retorna E_R[ ]
```

predição de um conjunto de teste S_t. Para cada instância $\mathbf{x}_t^{(i)}$, busca-se, no conjunto S_v, as h_K instâncias vizinhas mais próximas de $\mathbf{x}_t^{(i)}$ que formarão o conjunto S_d (linha 5). Sempre que um regressor consegue fazer boas predições para todos os padrões em S_d (linhas 7 a 9), ele é adicionado ao conjunto $E_R[i]$, conjunto de regressores selecionados *(ensemble of regressors)* para realizar a predição da instância de teste $\mathbf{x}_t^{(i)}$. Com base na definição apresentada na Subseção 4.2.1, temos que $e_r^{(h)}$ representa o erro absoluto do r-ésimo regressor na h-ésima instância de S_d, denotada por $\mathbf{x}_d^{(h)}$,

dado por

$$e_r^{(h)} = |t_d^{(h)} - f_r(\mathbf{x}_d^{(h)})| \tag{4.3}$$

em que $t_d^{(h)}$ é o valor real (alvo) relacionado à instância $\mathbf{x}_d^{(h)}$ e $f_r(\mathbf{x}_d^{(h)})$ é o valor predito pelo regressor. A predição será categorizada como boa, se $e_r^{(h)} \leq \gamma_h$, na qual γ_h é o limite de erro para a instância $\mathbf{x}_d^{(h)}$, calculado conforme (4.2).

O número de instâncias vizinhas tem início em h_K, e é verificado se algum regressor fez boas predições para todas as h_k instâncias de S_d. Caso nenhum regressor tenha conseguido acertar todos os padrões em S_d, então é realizada uma nova tentativa com um número menor de vizinhos (h_k é decrementado). Caso h_k chegue a zero e o conjunto $E_R[i]$ continue vazio, o algoritmo verifica qual foi o regressor que obteve o maior número de predições boas, denotado por g (considerando h_K vizinhos). Após isso, todos os regressores que tiveram um número de boas predições igual a g são incluídos em $E_R[i]$ (linhas 18 e 19). Na variante DySEREK-*Elimination*, a saída final da predição é dada por

$$f(\mathbf{x}_t) = \frac{1}{Q} \sum_{r=1}^{Q} f_r(\mathbf{x}_t) \ , \tag{4.4}$$

em que Q é o número total de regressores do CFR, $\mathbf{x}_t$ é um vetor que representa a instância de teste que será predita e $f_r(\cdot)$ é a saída do r-ésimo regressor pertencente ao CFR.

Variante DySEREK-*Union*

O Algoritmo 7 descreve a variante DySEREK-*Union*. Assim como na variante DySEREK-*Elimination*, temos uma seleção de um subconjunto de regressores para cada instância de teste $\mathbf{x}_t^{(i)}$ em S_t (linhas 2 a 11). Inicialmente, é definido o conjunto S_d, que representa o conjunto de k_v vizinhos mais próximos da instância de teste

Algoritmo 7: DySEREK-*Union*

Entrada:
R, *pool* de regressores;
S_t, conjunto de teste;
S_v, conjunto de validação;
h_K, número de instâncias vizinhas obtidas a partir do conjunto S_v;
Saída:
$E_R[\,]$, conjuntos finais de regressores (um para cada instância de teste $\mathbf{x}_t$)

```
1  início
2      para cada instância de teste x_t^(i) ∈ S_t faça
3          S_d ← k_v vizinhos de x_t^(i) em S_v
4          para cada instância x_d ∈ S_d faça
5              para cada regressor r_j em R faça
6                  se r_j fez boa predição de x_d então
7                      E_R[i] = E_R[i] ∪ r_j
8                  fim
9              fim
10         fim
11     fim
12 fim
13 retorna E_R[ ]
```

$\mathbf{x}_t^{(i)}$ (linha 3). Serão incluídos no conjunto $E_R[i]$ todos os regressores que tiverem pelo menos uma predição boa de qualquer padrão $\mathbf{x}_d^{(h)}$ do conjunto S_d. Além disso, o número de predições boas de um regressor também determina o peso que este regressor terá na média ponderada das saídas. Na variante DySEREK-*Union*, a saída final da predição é dada por

$$f(\mathbf{x}_t) = \frac{1}{Q}\left(\frac{\sum_{r=1}^{Q} f_r(\mathbf{x}_t)\cdot g_r}{\sum_{r=1}^{Q} g_r}\right), \tag{4.5}$$

em que Q é o numero total de regressores no CFR, $\mathbf{x}_t$ é um vetor que representa a instância de teste, f_r é a saída do r-ésimo regressor pertencente ao CFR e g_r é o numero de predições boas do r-ésimo regressor.

Variante DySEREK-*Elimination*-W

Em geral, a variante DySEREK-*Elimination*-W é similar à variante DySEREK-*Elimination*. A diferença está no fato de que cada regressor do CFR possui um peso

w_r, definido por

$$w_r = \sum_{h=1}^{h_K} \left(\frac{\varphi(e_{rh})\varphi(d_h)}{m_e m_d} \right) , \tag{4.6}$$

em que $\varphi(x) = e^{-x}$, e_{rh} é o erro absoluto do r-ésimo regressor na h-ésima instância de S_d, denotada por $\mathbf{x}_d^{(h)}$, e d_h é a distância entre a instância $\mathbf{x}_d^{(h)}$ e a instância de teste $\mathbf{x}_t$. Os parâmetros m_e e m_d representam, respectivamente, os valores máximos de $\varphi(e_{rh})$ e $\varphi(d_h)$ obtidos para a instância $\mathbf{x}_d^{(h)}$. Uma vez que d_h e e_{rh} podem ser iguais a zero para alguma instância $\mathbf{x}_d^{(h)}$, a função $\varphi(\cdot)$ evita que tenhamos uma indeterminação em algum termo de (4.6). Assim, a saída final da variante DySEREK-*Elimination*-W é dada por

$$f(\mathbf{x}_t) = \frac{1}{Q} \left(\frac{\sum_{r=1}^{Q} f_r(\mathbf{x}_t) \cdot w_r}{\sum_{r=1}^{Q} w_r} \right) . \tag{4.7}$$

Variante DySEREK-*Union*-W

Na variante DySEREK-*Union*-W, assim como no caso anterior, cada regressor do CFR possui um peso w_r. A diferença é que o número de predições boas obtidas por cada regressor, denotado por g_r, é considerado no cálculo de w_r, tal que

$$w_r = \sum_{h=1}^{h_K} \left(\frac{\varphi(e_{rh})\varphi(d_h)}{m_e m_d} \right) g_r \ . \tag{4.8}$$

Portanto, a saída final da variante DySEREK-UNION-W é dada por (4.7), mas com os pesos w_r calculados conforme (4.8).

Descrito o funcionamento do método DySEREK e suas variantes, os resultados numéricos relativos ao seu desempenho serão apresentados a seguir e comparados com o desempenho de outros algoritmos de regressão.

4.3 RESULTADOS NUMÉRICOS

Em geral, algoritmos que utilizam conjunto de preditores, tais como o método DySEREK, necessitam de mais recursos computacionais. Assim, antes de o modelo de ML ser colocado

em produção, considera-se uma boa prática comparar o desempenho desses algoritmos com outros que utilizam menos recursos computacionais. Além disso, outro fator importante é o ajuste (*tuning*) de parâmetros, pois dependendo do conjunto de parâmetros utilizados, o desempenho do algoritmo pode ser consideravelmente diferente.

Face ao exposto, a presente Seção possui dois propósitos. O primeiro é comparar o desempenho do método DySEREK com outros algoritmos de ML e o segundo é analisar como o ajuste (*tuning*) dos parâmetros do método DySEREK influencia seu desempenho. Para atingir os dois objetivos, serão usados os 14 *datasets* públicos listados na Tab. 7 para a obtenção dos resultados.

Inicialmente, o desempenho do método DySEREK será comparado com o de outros três algoritmos, a saber: o regressor único k-NN, o *pool* de regressores CBAS, apresentado no Capítulo 3, e o método de seleção dinâmica DESIP (DIAS; WINDEATT, 2014a; DIAS; WINDEATT, 2014b). O algoritmo k-NN foi escolhido para permitir a comparação dos resultados do método DySEREK com um regressor treinado e ajustado com os dados completos de treinamento. No segundo caso, a comparação do método DySEREK será realizada com a média do conjunto (*pool* CBAS). Por último, temos como referência de comparação o método DESIP, um segundo método dinâmico de seleção de conjuntos de regressores (*ensemble*), semelhante ao método DySEREK, no qual um novo conjunto de regressores é escolhido para cada predição realizada.

Na implementação de qualquer sistema de múltiplos regressores, o número de regressores que compõem o *pool* é um fator que pode influenciar diretamente no desempenho do conjunto. Dessa forma, foram gerados três *pools* com 32, 64 e 128 regressores para todas as variantes do DySEREK, assim como para o *pool* CBAS e o método DESIP. Além disso, para cada tamanho de *pool*, foi gerada uma tabela comparativa de vitórias, empates e derrotas (*win-draw-lose*, W-D-L), considerando todos os *datasets* listados na Tab. 7. Para verificar se as diferenças entre os algoritmos são estatisticamente relevantes,

Tabela 7 – *Datasets* públicos usados nos experimentos com o método DySEREK.

Dataset	Número de *features*	Número de Instâncias
Auto93	23	93
Autompg	7	392
Boston Housing	13	506
Carseats	10	400
Concrete	8	1030
Hitters	19	322
Servo	5	167
Wage	11	3000
Abalone	8	4177
Airquality	5	9358
BodyFat	13	252
CPUs	6	209
Forestfires	12	513
Slump Test	9	103

os testes de Friedman e *post-hoc* de Nemenyi foram aplicados (DEMSAR, 2006). Nos casos em que não houve diferenças estatisticamente relevantes, os algoritmos foram marcados com um asterisco (*) e nesse caso, um empate foi registrado.

Em relação à configuração do método DySEREK, todas as suas variantes foram ajustadas com um conjunto de validação igual a 10% do conjunto de treinamento. O valor do parâmetro h_K, definido como o número de instâncias vizinhas à instância de teste utilizadas pelos regressores do método DySEREK, foi ajustado conforme o tamanho do *pool* de regressores. Assim, foram utilizados os valores de h_K iguais a 5, 7 e 9 para os *pools* com 32, 64 e 128 regressores, respectivamente. Em todos os experimentos realizados, 90% do *dataset* foi assumido como conjunto de treino e 10% como conjunto de teste para todos os algoritmos.

A Tab. 8 indica o RMSE obtido para cada algoritmo em todos os *datasets*, considerando a utilização de um *pool* com 32 regressores. Para identificar as variantes do método DySEREK, são utilizados os acrônimos Dy-E (DySEREK-*Elimination*), Dy-EW (DySEREK-*Elimination*-W), Dy-U (DySEREK-*Union*) e Dy-UW (DySEREK-*Union*-W). Observando

Tabela 8 – Desvio do erro médio quadrático obtido para os 14 *datasets* considerados e um *pool* com 32 regressores.

Dataset	CBAS	DESIP	Dy-E	Dy-EW	Dy-U	Dy-UW	k-NN
Auto93	2,37	10,74	4,32	3,93	**2,09**	3,89	2,88
Autompg	2,87	2,73	3,64	3,81	2,59	**2,57**	3,44
Boston Housing	3,24	5,37	5,50	5,57	3,87	4,42	**2,88**
Carseats	1,93	2,10	2,14	2,14	2,26	**1,65**	2,21
Concrete	8,18	8,63	9,05	9,99	8,59	**7,30**	8,32
Hitters	349,76	277,14	267,88	360,18	**250,10**	369,28	298,80
Servo	9,45	**4,88**	7,73	6,41	5,77	6,28	5,81
Wage	12,45	17,99	7,94	9,92	8,02	**6,45**	16,38
Abalone	2,37	2,46	2,62	2,80	2,40	**1,95**	2,36
Airquality	7,98	8,23	8,73	9,48	7,48	**7,30**	7,85
BodyFat	4,57	3,22	3,23	3,64	3,14	3,44	**2,66**
CPUs	46,31	35,95	119,94	83,57	67,62	147,84	**30,78**
Forestfires	30,90	150,05	30,80	31,90	28,77	**14,01**	52,49
Slump Test	5,07	4,85	5,15	7,61	**2,59**	4,33	4,04
W-D-L	0-0-14	1-0-13	0-0-14	0-0-14	3-0-11	7-0-7	3-0-11

a Tab. 8, é possível notar que a variante Dy-UW (DySEREK-*Union*-W) foi a que teve melhor desempenho, com 7 vitórias de 14 possíveis. Na segunda posição, temos a variante Dy-U (DySEREK-*Union*) e o algoritmo k-NN, cada um com 3 vitórias. Quanto ao método DESIP, este venceu somente em um *dataset*. Dentre as variantes do método DySEREK, ficou evidente que o desempenho das variantes do grupo *Union* foi melhor do que o desempenho das variantes do grupo *Elimination*, uma vez que as primeiras venceram em aproximadamente 70% dos *datasets*. Por fim, vale destacar o desempenho do algoritmo k-NN, mostrando que, em alguns casos, bons resultados podem ser alcançados apenas com um regressor simples.

A Tab. 9 indica o RMSE obtido para cada algoritmo em todos os *datasets*, considerando a utilização de um *pool* com 64 regressores. Neste ponto, cabe uma ressalva ao fato de que os resultados para o algoritmo k-NN estão diferentes do caso anterior (32 regressores). A razão é que, para o experimento com 64 regressores, novos conjuntos de treino e teste, na

Tabela 9 – Desvio do erro médio quadrático obtido para os 14 *datasets* considerados e um *pool* com 64 regressores.

Dataset	CBAS	DESIP	Dy-E	Dy-EW	Dy-U	Dy-UW	k-NN
Auto93	2,08	3,99	10,95	4,80	8,21	**1,72**	2,88
Autompg	3,16	2,80	4,34	3,70	**2,15**	4,12	3,44
Boston Housing	**2,90**	5,15	5,85	5,29	3,52	3,37	5,51
Carseats	2,37	2,21	2,19	2,62	***2,08**	***2,09**	2,29
Concrete	**7,19**	9,00	8,84	10,48	7,81	7,74	7,35
Hitters	227,11	270,40	237,40	482,61	**161,55**	444,24	459,74
Servo	7,17	5,26	9,53	**4,30**	6,82	6,73	4,60
Wage	13,50	19,04	**4,49**	5,28	8,87	8,36	14,32
Abalone	**2,14**	2,42	2,73	2,81	2,18	2,29	2,53
Airquality	7,87	8,69	8,97	9,21	7,29	**6,72**	8,06
BodyFat	3,16	2,86	3,01	**2,62**	2,78	3,85	3,87
CPUs	65,34	89,66	34,02	147,06	21,65	**11,81**	13,61
Forestfires	**24,64**	110,82	63,13	30,16	50,42	152,97	37,88
Slump Test	6,01	3,46	5,01	4,42	4,38	**2,66**	3,37
W-D-L	4-0-10	0-0-14	1-0-13	2-0-12	2-1-11	4-1-9	0-0-14

* - Valores estatisticamente irrelevantes.

mesma proporção anterior, foram gerados. Este fato se repetiu no caso do *pool* com 128 regressores.

Quando o *pool* de 64 regressores é utilizado, conforme pode ser visto na Tab. 9, a variante Dy-UW e o *pool* CBAS tiveram o melhor resultado com 4 vitórias cada. Outro detalhe é que, com o aumento do tamanho do *pool*, as variantes Dy-E (DySEREK-*Elimination*) e Dy-EW (DySEREK-*Elimination*-W) se tornaram vencedoras em 1 e 2 *datasets*, respectivamente (não haviam vencido em nenhum *dataset* no *pool* com 32 regressores). Dessa forma, apesar de as variantes do grupo *Union* continuarem vencendo na maioria dos *datasets* (em torno de 43%), as variantes do grupo *Elimination* melhoraram de desempenho, vencendo nos *datasets Wage*, *Servo* e *BodyFat*. O algoritmo k-NN e o método DESIP não obtiveram vitórias quando 64 regressores são utilizados.

Os resultados para o *pool* de 128 regressores são mostrados na Tab. 10. Nesse caso, a variante Dy-U teve o melhor desempenho com 5 vitórias, passando a superar a variante

Tabela 10 – Desvio do erro médio quadrático obtido para os 14 *datasets* considerados e um *pool* com 128 regressores.

Dataset	CBAS	DESIP	Dy-E	Dy-EW	Dy-U	Dy-UW	k-NN
Auto93	4,41	3,90	3,54	**1,58**	3,47	1,71	2,88
Autompg	3,60	**2,40**	2,97	3,40	2,69	2,61	3,26
Boston Housing	2,88	3,62	3,09	4,51	**2,42**	3,13	3,35
Carseats	2,21	2,56	2,13	2,22	**2,05**	2,39	2,44
Concrete	7,09	7,16	10,76	8,57	8,05	6,88	**6,56**
Hitters	314,55	421,94	571,21	**278,03**	352,46	298,93	273,09
Servo	8,87	7,75	5,92	**5,65**	5,92	6,76	9,20
Wage	12,30	17,13	**3,98**	6,39	8,74	8,00	12,04
Abalone	2,22	2,24	2,55	2,71	**2,09**	2,48	2,33
Airquality	7,90	8,52	8,43	9,41	***7,29**	***7,31**	8,06
BodyFat	3,59	3,23	**2,75**	3,38	3,20	3,28	3,09
CPUs	14,67	20,22	10,08	70,61	**8,53**	267,86	38,15
Forestfires	27,49	46,10	37,88	67,62	**15,45**	31,14	21,23
Slump Test	3,51	5,71	**2,69**	3,88	2,35	4,21	3,65
W-D-L	0-0-14	1-0-13	3-0-11	3-0-11	5-1-8	0-1-13	1-0-13

* - Valores estatisticamente irrelevantes.

Dy-UW, algo que não aconteceu nos experimentos com os *pools* de 32 e 64 regressores. Por outro lado, a variante Dy-UW, maior vencedora das variantes do DySEREK para os *pools* anteriores, não obteve nenhuma vitória. Em relação ao grupo *Elimination*, é importante ressaltar a evolução no desempenho de suas variantes (Dy-E e Dy-EW), cada uma com 3 vitórias. Outro ponto importante é que, mesmo com 128 regressores, as variantes do DySEREK não conseguiram superar o algoritmo k-NN no *dataset* Concrete. Por fim, o método DESIP foi o algoritmo vitorioso no *dataset Autompg*.

Os resultados indicados nas Tabs. 8, 9 e 10 nos permitem afirmar que o método DySEREK (todas as variantes) apresentou melhor desempenho (menor RMSE) em aproximadamente 70% dos *datasets* avaliados para os *pools* com 32 e 64 regressores. Para o *pool* com 128 regressores, o método DySEREK teve melhor desempenho em cerca de 86% dos *datasets*. Os empates registrados nos *datasets Carseats* e *Airquality* nos *pools* com 64 e 128 regressores, na devida ordem, são contabilizados como vitórias do método

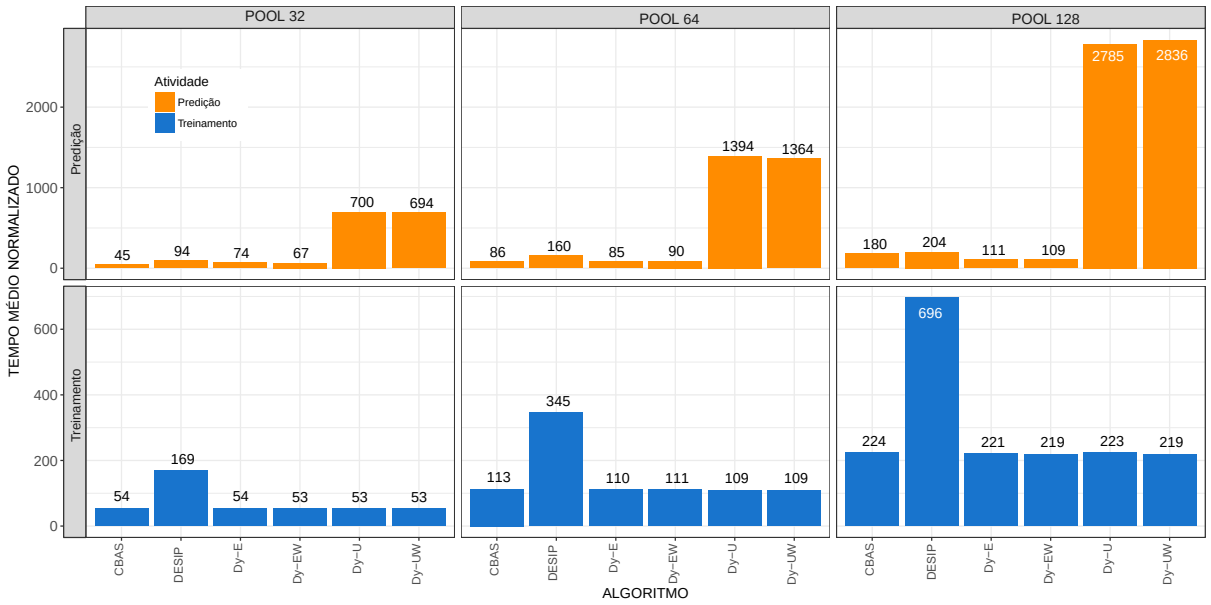

Figura 7 – Tempo médio normalizado de treinamento e predição para os *pools* com 32, 64 e 128 regressores, considerando todas as variantes do DySEREK, o método DESIP e o *pool* CBAS. A normalização tem como referência os tempos de treinamento t_T e de predição t_P do regressor k-NN (regressor único), cuja razão $t_T/t_P \approx 56$.

DySEREK, uma vez que envolve duas de suas variantes. Considerando que os *datasets* utilizados nos experimentos são bem diversificados em termos de número de instâncias e *features*, é possível afirmar que o método DySEREK obteve êxito no quesito desempenho quando comparado a outras técnicas de regressão.

Além do desempenho dos algoritmos de ML, os tempos consumidos nas etapas de treinamento e de predição são importantes, pois definem em que tipos de aplicação os algoritmos podem ser utilizados. A Fig. 7 exibe os tempos médios normalizados de treinamento e predição para os *pools* de 32, 64 e 128 regressores, considerando todas as variantes do DySEREK, o método DESIP e o *pool* CBAS. O algoritmo k-NN não é incluído, uma vez que é usado como unidade de referência. Na realidade, para cada *pool* de regressores, uma execução diferente do algoritmo k-NN é realizada para a tomada do tempo de referência. No entanto, como o número de instâncias em cada execução é idêntico, podemos considerar que o tempo de referência adotado para a normalização é praticamente o

	Pool		
Dataset	32	64	128
Auto93	51,69%	84,23%	55,38%
AutoMpg	32,49%	50,46%	*23,16%
BostonHousing	*30,50%	*42,44%	46,20%
Carseats	26,67%	20,63%	14,52%
Concrete	26,85%	*26,14%	*36,02%
Hitters	32,27%	66,52%	51,33%
Servo	*25,32%	54,89%	16,31%
Wage	34,99%	49,37%	54,42%
AirQuality	22,96%	27,09%	22,50%
Abalone	30,43%	*22,45%	22,93%
Bodyfat	*13,68%	31,90%	18,73%
Cpus	*54,26%	91,97%	96,78%
Forestfires	56,08%	*80,28%	77,15%
SlumpTest	65,88%	46,83%	36,00%

Ganho RMSE 13,68% 96,78%

(*) = Variante DySEREK não foi vencedora

Figura 8 – Mapa de calor para o ganho do RMSE calculado entre a pior e a melhor variante do método DySEREK (quando uma das variantes foi vencedora), considerando todos os *datasets* e *pools* de regressores.

mesmo para todos os *pools* em questão. Dito isso, os tempos de treinamento e de predição do algoritmo k-NN, denotados por t_T e t_P, respectivamente, são adotados como referência para os valores normalizados indicados na Fig. 7. Além disso, para que seja possível a comparação dos tempos de cada atividade (treinamento ou predição), a razão entre t_T e t_P vale aproximadamente 56.

É possível observar que, conforme esperado, o tempo de treinamento é diretamente proporcional ao número de regressores do *pool*. Além disso, o tempo de treinamento do *pool* CBAS e das variantes do DySEREK são similares, visto que o *pool* CBAS é utilizado para treinar os regressores do método DySEREK. Ainda na Fig. 7, podemos verificar que o tempo de treinamento do método DESIP é o maior de todos, independente do tamanho do *pool*. Uma possível razão para esse fato é que, no método DESIP, a composição do CFR é realizada durante a fase de treinamento, sendo atribuído um CFR diferente a cada

Tabela 11 – Parâmetros utilizados para ajuste (*tuning*) do método DySEREK.

Parâmetros DySEREK	Valores testados
Tamanho do *pool* (regressores)	32, 64, 128
Variantes do método	Dy-E, Dy-EW, Dy-UW, Dy-UW
Hiper-k (h_K)	5, 7, 9
Tamanho do conjunto de validação	10%, 20%, 30%

instância de validação. No que diz respeito ao tempo de predição, o método DESIP não supera as variantes Dy-E e Dy-EW, todavia possui tempo médio de predição menor do que os tempos das variantes Dy-U e Dy-UW. De fato, em todos os tamanhos de *pool* analisados, os tempos de predição das variantes Dy-U e Dy-UW se destacam por serem os maiores. Esse fato está relacionado ao número de regressores usados para a geração da predição final, pois basta o regressor ter uma das predições das h_K instâncias vizinhas mais próximas da instância de teste, tida como predição boa, para participar da predição final.

A Fig. 8 ilustra o mapa de calor para o ganho do RMSE calculado entre a pior e a melhor variante do método DySEREK, considerando todos os *datasets* e *pools* de regressores. As células marcadas com asterisco indicam cenários nos quais nenhuma variante do DySEREK foi vitoriosa. O mapa de calor em questão nos mostra que o desempenho do método DySEREK, segundo suas variantes, pode ser bem diversificado conforme sua parametrização. Considerando apenas os casos em que alguma variante do método DySEREK foi vencedora, nota-se que, para o *pool* de 32 regressores, o ganho de RMSE variou aproximadamente entre 23% (*dataset AirQuality*) e 66% (*dataset Slump Test*). No caso do *pool* com 64 regressores, a faixa de variação do ganho de RMSE foi de $20,6\%$ (*dataset Carseats*) a 92% (*dataset CPUs*). Finalmente, no *pool* com 128 regressores, a variação do ganho de RMSE se situou entre $14,5\%$ (*dataset Carseats*) e 97% (*dataset CPUs*). As variações percentuais apresentadas para os três *pools* evidenciam que o ajuste (*tuning*) é muito importante para a obtenção de bons resultados na aplicação do método DySEREK.

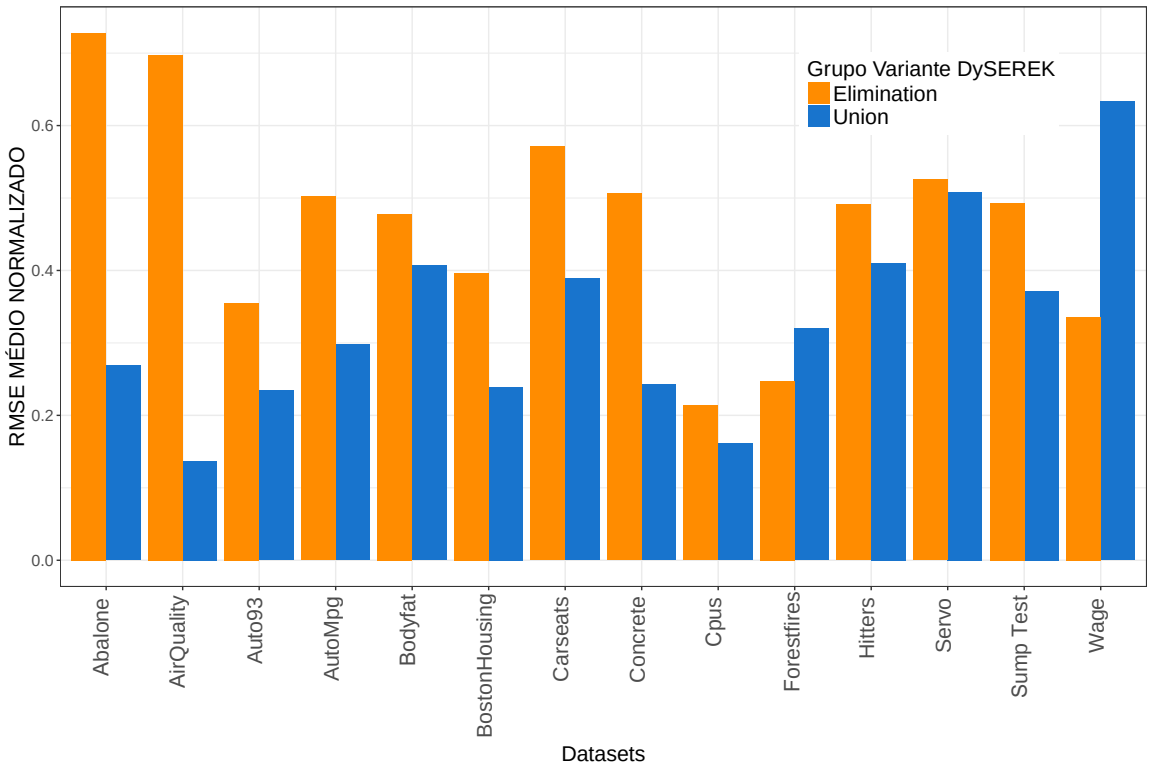

Figura 9 – RMSE médio normalizado, considerando os grupos de variantes *Elimination* (Dy-E e Dy-EW) e *Union* (Dy-U e Dy-UW), do método DySEREK por *dataset*.

Face ao exposto, vamos analisar a influência do conjunto de parâmetros utilizados no desempenho do método DySEREK. Para isso, faz-se necessário um melhor entendimento de como a escolha da variante (Dy-E, Dy-EW, Dy-U ou Dy-UW), o tamanho do *pool*, o valor de h_K e o tamanho do conjunto de validação afetam o desempenho do DySEREK. Com isso em mente, o método DySEREK foi avaliado para as 108 combinações diferentes dos parâmetros listados na Tab. 11 aplicadas aos 14 *datasets* indicados na Tab. 7.

Para que a comparação das variantes do método DySEREK seja justa, é necessário que os valores de RMSE obtidos sejam normalizados, visto que, dependendo do *dataset* considerado, a escala do RMSE é diferente, como pode ser observado nas Tabs. 8, 9 e 10. A normalização foi realizada por *dataset* e considerando o intervalo $[0, 1]$. Assim, 0 e 1 são atribuídos, respectivamente, ao menor e maior RMSE do *dataset*.

A Fig. 9 ilustra o RMSE médio normalizado dos grupos de variantes *Elimination* (Dy-E e Dy-EW) e *Union* (Dy-U e Dy-UW) para todos os 14 *datasets*. Os resultados do

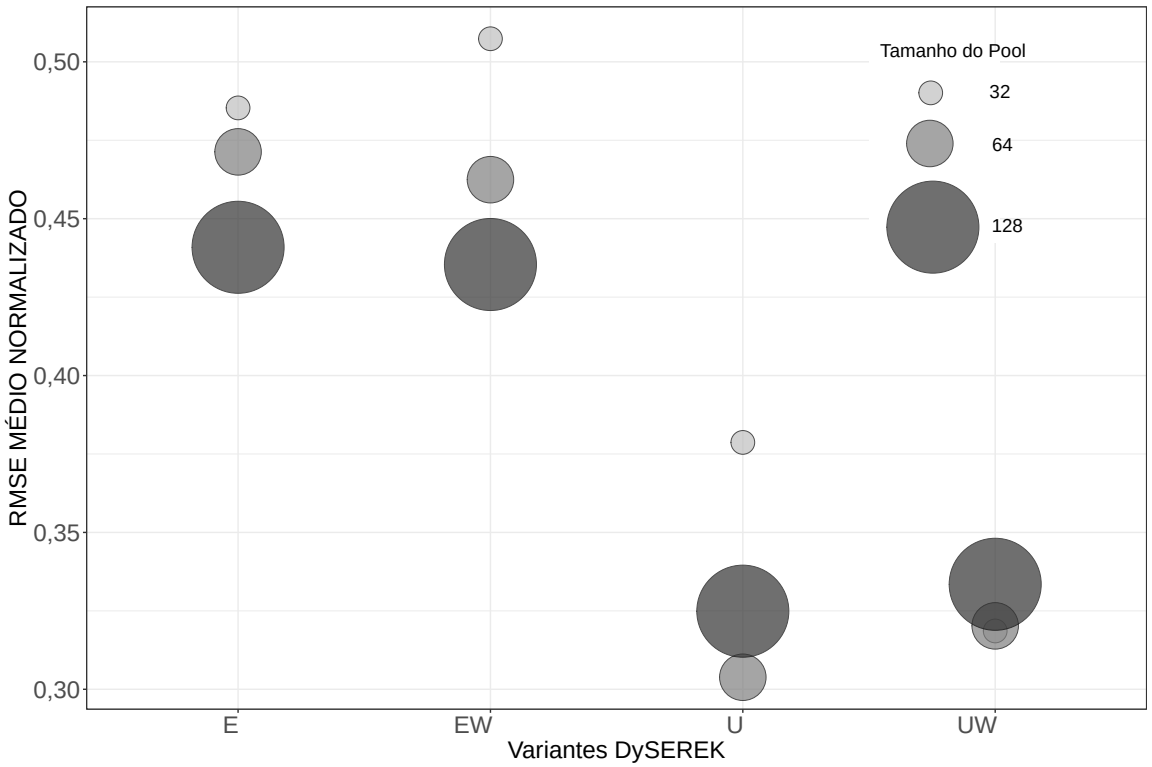

Figura 10 – RMSE médio normalizado, considerando todos os *datasets*, valores de h_K e conjuntos de validação, para cada variante do método DySEREK e tamanho de *pool* de regressores.

grupo *Union* são representados pelas barras de cor azul, enquanto os resultados relativos ao grupo *Elimination* são representados pelas barras de cor laranja. Uma vez que todas as combinações possíveis dos parâmetros listados na Tab. 11 foram avaliadas, o valor de RMSE médio normalizado indicado para cada grupo de variantes (por *dataset*) considera 54 modelos. É possível notar que, de uma maneira geral, o grupo *Union* apresentou o melhor desempenho, sendo derrotado apenas em dois *datasets*: *Forestfires* e *Wage*. Vale ressaltar que, conforme pode ser visto na Fig. 9, dados os *datasets* nos quais o grupo *Union* teve o melhor desempenho, as maiores diferenças em relação ao grupo *Elimination* foram alcançadas nos dois maiores *datasets* em número de instâncias (*Abalone* e *AirQuality*). No entanto, no terceiro maior *dataset* (*Wage*), o grupo *Elimination* supera o grupo *Union*.

Além do tipo de estratégia de combinação (*Union* ou *Elimination*), outro fator importante para a utilização do método DySEREK é o tamanho do *pool* de regressores. Este é um parâmetro importante, pois a utilização de um número grande de regressores pode

aumentar o custo computacional do DySEREK sem trazer melhoria de desempenho. A Fig. 10 mostra o RMSE médio normalizado, considerando todos os *datasets*, valores de h_K e conjuntos de validação, para cada variante do método DySEREK e tamanho de *pool* (32, 64 e 128 regressores). Cada variante do DySEREK é indicada no eixo das abscissas e o tamanho dos círculos (o centro do círculo representa o valor do RMSE médio normalizado) indica o tamanho do *pool* utilizado . Podemos perceber que a variante Dy-E teve seu desempenho melhorado (diminuição do RMSE) com o aumento do tamanho do *pool*. O mesmo comportamento foi observado para a variante Dy-EW, no entanto os valores obtidos sugerem uma possível saturação do ganho com o aumento do *pool*. Diante dos resultados apresentados para o grupo *Elimination*, concluímos que, para os *datasets* avaliados nesta tese, é possível obter ganhos adicionais com o aumento do número de regressores do *pool*, com maior probabilidade para a variante Dy-E.

Quanto ao grupo *Union*, um comportamento diferente foi observado. Na variante Dy-U, a mudança de 32 para 64 regressores promoveu ganho (inclusive, o maior dos ganhos representados na Fig. 10). Contudo, o aumento do *pool* de 64 para 128 regressores trouxe uma piora de desempenho, o que sugere um número intermediário de regressores que maximiza o ganho em RMSE para a variante Dy-U. Por fim, foi percebido que a variante Dy-UW foi a única que não conseguiu, em nenhum momento, uma melhoria de desempenho com o incremento do número de regressores do *pool*.

Cabe enfatizar que os resultados apresentados na Fig. 10 consideram as médias dos RMSEs normalizados obtidos em todos os *datasets* abordados neste estudo. Logo, os valores podem ser diferentes para cada *dataset* especificamente. Apesar disso, os resultados apresentados nos permitem extrair algumas conclusões. Em primeiro lugar, fixado o tamanho do *pool*, as variantes do grupo *Union* apresentam melhor desempenho que as variantes do grupo *Elimination*. Em segundo lugar, o autor conjectura que as variantes do grupo *Elimination* necessitam de *pools* maiores que 128 para obterem desempenhos equivalentes

às variantes do grupo *Union*. No entanto, tal suposição pode não se confirmar em virtude, principalmente, do comportamento apresentado pela variante Dy-EW. Finalmente, a busca por melhor desempenho das variantes do grupo *Union* em função do aumento do tamanho do *pool* não se mostra uma opção eficiente para valores maiores do que 128 ou mesmo maiores do que 64 regressores. Em particular para a variante Dy-U, suspeita-se que o ganho máximo possível, em média, possa ser obtido com um *pool* que tenha entre 64 e 128 regressores.

4.4 RESUMO DO CAPÍTULO

No presente Capítulo, foi apresentado o DySEREK, um novo método de seleção dinâmica de conjunto de regressores baseado no método KNORA. Seus bons resultados em tarefas de classificação motivaram a proposição do DySEREK. Inicialmente, foram expostas as principais diferenças entre as tarefas de classificação e regressão, bem como as peculiaridades que devem ser consideradas na adaptação de algoritmos de classificação para regressão e vice-versa. Em seguida, foram apresentados detalhes das implementações das quatro variantes propostas para o DySEREK, quais sejam, Dy-E (DySEREK-*Elimination*), Dy-EW (DySEREK-*Elimination*-W), Dy-U (DySEREK-*Union*) e Dy-UW (DySEREK-*Union*-W). Após isso, experimentos com 14 *datasets* públicos e diferentes configurações de *pools* foram realizados com o objetivo de comparar os desempenhos dos métodos DySEREK, *pool* CBAS, regressor único (k-NN) e o método DESIP, outro algoritmo de seleção dinâmica. Verificou-se que, dependendo do número de regressores do *pool*, o DySEREK (todas as variantes) teve melhor desempenho em uma faixa de 70 a 86% dos *datasets*. Finalmente, foram executados vários experimentos com o objetivo de verificar a influência dos parâmetros do DySEREK, tais como hiper-k, tamanho do conjunto de validação e variante adotada, em seu desempenho. Observou-se que, em comparação com as variantes do grupo *Elimination* (Dy-E e Dy-EW), as variantes do grupo *Union* (Dy-U e Dy-UW) tiveram melhores resultados em cerca de 83% dos *datasets*, usando *pools* menores. Por

outro lado, as variantes *Elimination* apresentaram tempos de predição menores, além de uma tendência de melhores desempenhos com o crescimento do *pool*, o que carece de uma investigação.

5 GEOLOCALIZAÇÃO DE USUÁRIOS MÓVEIS

Nos últimos anos, os avanços nos sistemas de comunicação contribuíram para um grande crescimento na utilização de telefones celulares para acessar serviços de Internet (DEVILLE et al., 2014). Assim, a demanda por aplicativos para dispositivos móveis aumentou em todo o mundo, em especial por aplicativos focados em serviços de uso massificado, tais como segurança nas transações com cartão de crédito (HAMMAD; FAITH, 2017), publicidade baseada em localização móvel (GUTIERREZ et al., 2019), método de pagamento com base em informações de localização (SALOMON; MAHAFFEY, 2019), assistência na estrada (TROGH et al., 2019) e rastreamento de doenças (NYHAN et al., 2019). Nesse contexto, as informações sobre a localização de usuários tornaram-se essenciais para melhorar a qualidade de tais aplicativos. Neste Capítulo, é apresentado um novo método de localização do tipo *fingerprinting* (FP), o qual utiliza ML para prever o ângulo de chegada (AoA, *angle of arrival*) da EM utilizando as diferenças de RSSI e de atraso de propagação (PD, *propagation delay)*. Inicialmente, são expostas as motivações pela alta demanda de serviços de localização utilizando redes celulares. Em seguida, a descrição do funcionamento básico de técnicas de localização do tipo FP é apresentada. Posteriormente, a proposta de nova técnica de localização baseada em FP é apresentada e uma comparação dela com a técnica FP tradicional (baseada em RSSI) é realizada. Finalmente, a técnica de localização proposta é implementada utilizando o método de regressão DySEREK e seus resultados são comparados com aqueles obtidos por meio de um único regressor.

5.1 INTRODUÇÃO

Nos dias atuais, a utilização intensa de dispositivos móveis tem aumentado a demanda por sistemas de localização, especialmente para serviços baseados em localização (SUN et al., 2017). Existe uma variedade de técnicas de localização que podem ser classificadas

em relação ao ambiente de destino (interno - *indoor* ou externo - *outdoor*) e que apresentam diferentes desafios e restrições para atingir seu objetivo principal, que é encontrar o usuário.

No caso da localização *outdoor*, o GPS é um dos sistemas de localização mais usados no mundo (NI et al., 2004; RIZK; SHOKRY; YOUSSEF, 2018), mas que possui limitações relevantes. No caso de posicionamento ao ar livre, uma vez que uma linha de visada desobstruída é necessária para o desempenho eficiente do GPS, sua acurácia é comprometida em áreas urbanas altamente densas e sob condições climáticas adversas. Além disso, o *hardware* do GPS é um módulo que consome muita energia em todos os dispositivos móveis, diminuindo a autonomia do aparelho celular no que diz respeito à duração de sua bateria (LI; LEI, 2019). Ademais, alguns fatores como tamanho, custo e limitação de potência não permitem a utilização do GPS em redes de pequenos objetos, tais como redes de sensores e redes no contexto de IoT.

Para a localização *indoor*, os principais desafios relacionados ao GPS geralmente são a ineficiência do sistema no interior de construções por conta dos níveis de atenuação sofridos pelos sinais de rádio em virtude do uso de diferentes materiais (vidro, madeira, concreto etc) na construção de prédios (NI et al., 2004). Outro desafio importante em ambientes *indoor* é o combate ao desvanecimento por propagação multipercurso, fenômeno pelo qual o sinal de rádio observado na antena do receptor resulta de uma combinação de múltiplas cópias do sinal original, causadas por reflexões em obstáculos presentes no ambiente de propagação.

Diante das limitações do GPS, outras alternativas devem ser exploradas, tais como a localização por meio de sinais de RF, também conhecida por radiolocalização (PERAL-ROSADO et al., 2018). As técnicas de radiolocalização podem ser divididas basicamente em dois grupos, quais sejam, as técnicas baseadas em FP, que utilizam a similaridade entre RSSIs para estimar a posição do usuário; e as técnicas baseadas em lateração de potência,

que utilizam os RSSIs de duas ou mais ERBs para obter um sistema de equações, cuja solução indica a estimativa da localização do usuário. Neste trabalho, o método de localização proposto se enquadra na categoria de técnicas baseadas em FP, logo não iremos abordar os conceitos relacionados ao método de lateração de potência neste documento. Mais informações sobre sistemas de localização baseados em lateração de potência podem ser encontradas em (TIMOTEO et al., 2017) e (KULKARNI; DESAI, 2019). Vale ressaltar que algumas técnicas de localização não se enquadram em nenhum dos dois grupos mencionados anteriormente. Por exemplo, em (OLIVEIRA et al., 2019), a posição do usuário é encontrada por meio de duas regressões diretas que têm como alvo a latitude e longitude.

As técnicas baseadas em FP têm sido amplamente utilizadas em conjunto com algoritmos de ML para localização de usuários em redes sem fio, seja para ambientes *indoor* ou *outdoor*. Tab. 12 lista alguns trabalhos relacionados à aplicação de ML a técnicas de localização baseadas em FP. Dentre as técnicas de localização *outdoor* com utilização de ML, podemos destacar (TIMOTEO et al., 2016), (ZOU et al., 2016(a))) e (ABDULLAH; ABDEL-QADER; BAZUIN, 2016). Em (TIMOTEO et al., 2016), SVR é utilizada em conjunto com a técnica FP para localizar usuários em uma rede celular GSM (*global system for mobile communications*). Em (ZOU et al., 2016(a))), propõe-se um método de localização *outdoor* baseado em k-NN com FP para corrigir posições estimadas pelo GPS. Em (ABDULLAH; ABDEL-QADER; BAZUIN, 2016), um modelo de localização *indoor* baseado em FP com rede neural probabilística (PNN, *probabilistic neural network*) é proposto. Resultados experimentais mostraram que uma melhor acurácia foi conseguida com a utilização de PNN no lugar do k-NN.

Em relação às técnicas de localização *indoor* com utilização de ML, destacamos (KIM et al., 2016), (ZOU et al., 2016(b))), (FÉLIX; SILLER; ÁLVAREZ, 2016) e (WANG; GAO; MAO, 2016). Em (KIM et al., 2016), é descrita a implementação de um sistema de localização *indoor* com k-NN e FP. Neste caso, foram utilizados sinais da rede Wi-Fi para localização

Tabela 12 – Trabalhos relacionados à aplicação de ML a técnicas de localização de usuários móveis em redes sem fio usando FP.

Artigo	Método	Ambiente
(TIMOTEO et al., 2016)	SVR	*outdoor*
(ZOU et al., 2016(a)))	k-NN	*outdoor*
(ZOU et al., 2016(b)))	k-NN com pesos	*indoor*
(YIU; YANG, 2016)	*FireFly*	*outdoor*/*indoor*
(KIM et al., 2016)	k-NN	*indoor*
(FÉLIX; SILLER; ÁLVAREZ, 2016)	DNN	*indoor*
(ABDULLAH; ABDEL-QADER; BAZUIN, 2016)	PNN	*outdoor*
(WANG; GAO; MAO, 2016)	DNN	*indoor*

de usuários em estações de metrô na cidade de Seul. Em (ZOU et al., 2016(b))), propõe-se um método de localização *indoor* que utiliza o algoritmo k-NN com pesos para normalizar as medições de RSSI, que, por sua vez, são utilizadas com FP. A normalização é aplicada para evitar discrepância nas medições devido ao modelo ou fabricante do dispositivo móvel.

Em (FÉLIX; SILLER; ÁLVAREZ, 2016), uma rede neural de aprendizagem profunda (DNN, *deep neural network*) é usada para implementar localização *indoor* usando FP. Resultados experimentais mostraram que o método proposto obteve bom desempenho ao preço de um elevado custo computacional de treinamento das DNNs. Em (WANG; GAO; MAO, 2016), é proposto um método de localização *indoor* que utiliza DNN em conjunto com máquinas restritas de Boltzmann (RBMs, *restricted Boltzmann machines*). A RBM é utilizada para ajudar no treinamento das DNNs, diminuindo o consumo de recursos computacionais, problema abordado em (FÉLIX; SILLER; ÁLVAREZ, 2016). Resultados experimentais mostraram que abordagens FP baseadas nas informações do estado do canal tiveram um desempenho superior àquelas baseadas no RSSI.

Por último, ressaltamos que algumas técnicas de localização podem ser utilizadas tanto para localização *outdoor* como para *indoor*. Por exemplo, em (YIU; YANG, 2016), os autores

propõem a utilização do algoritmo *FireFly* (YANG, 2009) em conjunto com um modelo probabilístico baseado em FP para localização nos dois tipos de ambiente.

Os trabalhos anteriormente mencionados não esgotam as inúmeras alternativas de sistemas de localização baseados em FP, mas ressaltam a importância dos fundamentos das técnicas de localização que utilizam FP. Dessa forma, os conceitos básicos sobre radiolocalização baseada em FP serão apresentados na próxima Seção.

5.2 TÉCNICAS DE RADIOLOCALIZAÇÃO BASEADAS EM FP

As técnicas de radiolocalização baseadas em FP constituem um grupo de técnicas de localização de EMs para ambientes sem linha de visada e podem ser aplicadas em qualquer rede sem fio (CHENG; LIU; FENG, 2019; DIAZ; AHMED; KAISER, 2018). Em geral, as técnicas baseadas em FP são divididas em duas fases, quais sejam, fase de treinamento (*off-line*) e fase de predição (*on-line*). Na primeira, são realizadas medições de campo para treinamento dos modelos de predição. Essa fase é dita *off-line*, porque não há contato ou qualquer tipo de coleta de dados da EM procurada. Na segunda fase (*on-line*), são realizadas medições relativas à EM. Essa medições servem como dados de entrada para o modelo de busca construído na fase *off-line*. Sendo assim, considere uma região geográfica na qual desejamos localizar uma EM. Inicialmente, será definida uma área de interesse na qual será aplicada a técnica de localização baseada em FP. De uma maneira geral, a área de interesse mencionada é definida como um quadrilátero cujos vértices possuem coordenadas geográficas conhecidas, conforme mostrado na Fig. 11(a). A ideia central é construir um banco de dados de correlação (CDB, *fingerprint correlation database*) com informações que associem algum atributo medido na rede celular com a posição geográfica da EM. Dentre os parâmetros de rede mais utilizados como atributos, podemos destacar o RSSI, o AoA, o tempo de chegada (ToA, *time of arrival*) e a diferença de tempo de chegada (TDoA, *time difference of arrival*) (DIAZ; AHMED; KAISER, 2018). Assumindo que o parâmetro de rede utilizado é o RSSI, os pontos azuis ilustrados na Fig. 11(a) re-

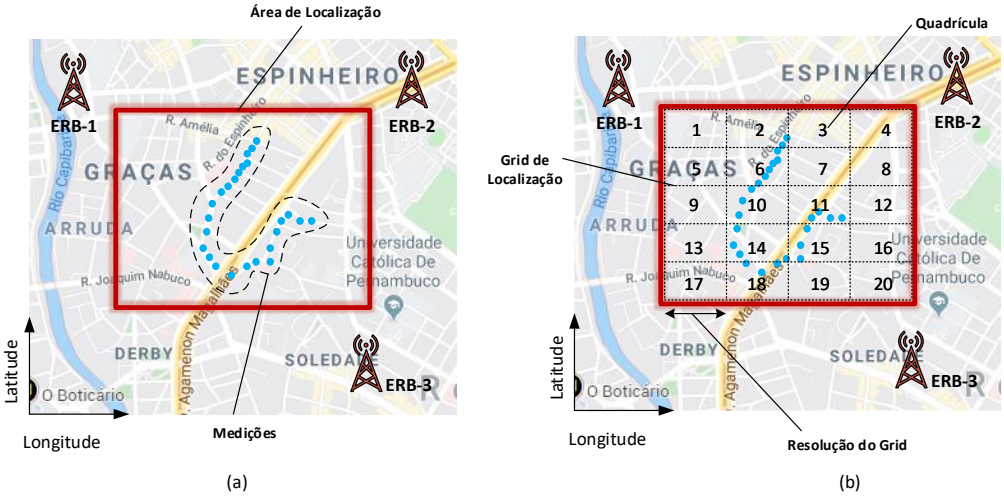

Figura 11 – (a) Área de localização para a implementação de técnicas do tipo FP e medições feitas em campo. (b) *Grid* de localização com respectiva resolução.

presentam as medições feitas em campo, por meio de um *drive-test*, que serão utilizadas na construção do banco de dados de correlação (CDB, *fingerprint correlation database*), também conhecido como mapa de rádio.

Com o objetivo de fazer a associação entre as possíveis posições geográficas da EM a ser localizada e os RSSIs coletados no *drive-test*, o quadrilátero que define a área de localização é dividido em pequenas porções quadradas denominadas quadrículas, formando um reticulado, que usualmente é denominado de *grid* de localização, o qual pode ser visto na Fig. 11(b). Vale ressaltar esse tipo de *grid* é denominado *grid* regular e a sua resolução é definida como o tamanho do lado da quadrícula, como pode ser visto na Fig. 11(b). Nos *grids* do tipo irregular, a geometria do *grid* é alterada de modo a se ter uma melhor adaptação ao método de localização. Não faz parte do escopo deste trabalho avaliar como a geometria do *grid* influencia o desempenho da técnica de localização baseada em FP. Estudos nesse contexto são descritos em (BITTENCOURT; URBANO; CUNHA, 2018). Sendo assim, o *grid* regular que caracteriza a área de localização será denominado simplesmente de *grid*.

Face ao exposto, podemos perceber na Fig. 11(b) que nem todas as quadrículas do *grid* são contempladas por medições de campo. Isto acontece pelo fato de a malha viária não

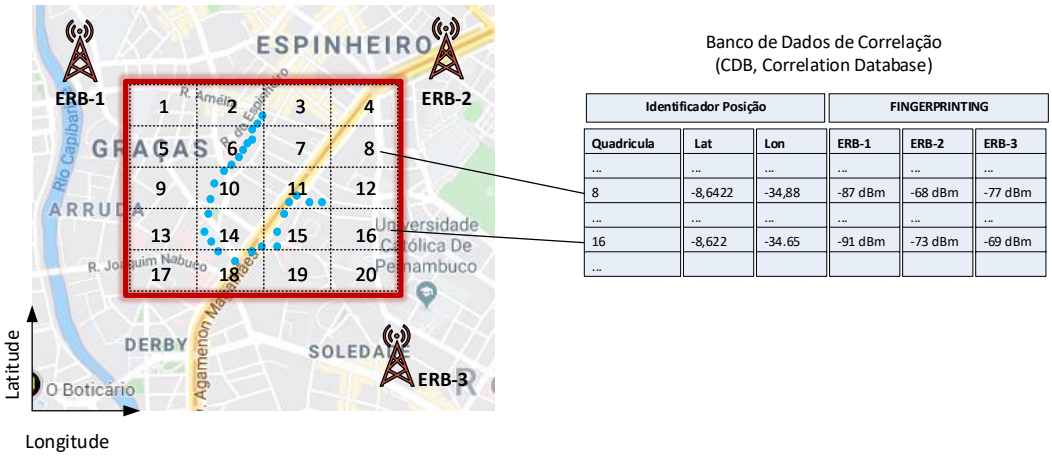

Figura 12 – Relação entre o banco de dados de correlação e o *grid* de localização para a implementação de técnicas de localização baseada em FP.

percorrer todas as quadrículas da área de localização ou simplesmente, por haver propriedades privadas na região que dificultam à obtenção de medidas de campo. Para solucionar o problema de termos o CDB incompleto, modelos de propagação eletromagnética ou algoritmos de ML podem ser utilizados para estimar os RSSIs associados às quadrículas não contempladas com medições de campo realizadas no *drive-test* (TIMOTEO; CUNHA; CAVALCANTI, 2014).

Existe uma grande variedade de técnicas baseadas em FP, como abordado em (CAMPOS; LOVISOLO, 2011), e todas possuem os mesmos elementos básicos, quais sejam, vetor FP, CDB, servidor de localização, espaço de busca e padrão de combinação. Inicialmente, define-se como vetor FP, o vetor com todos os atributos (valores observados) que são usados para verificação de similaridade. Conforme mencionado anteriormente, estamos assumindo que o atributo utilizado é apenas o RSSI. Na Fig. 12, podemos ver uma representação básica de como seria um CDB com os vetores FP contendo os valores de RSSI preditos para as quadrículas 8 e 16, considerando três ERBs na região. Ainda na Fig. 12, podemos visualizar as coordenadas geográficas (latitude e longitude) associadas às respectivas quadrículas.

A ideia básica da técnica de localização baseada em FP, como pode ser vista na Fig. 12, é relacionar cada registro (vetor) do CDB a uma posição na área de localização

(coordenadas geográficas do centro da quadrícula do *grid*). Esta atividade é realizada na fase *off-line*. Na fase *on-line*, quando se busca a EM, é fornecido ao CDB um vetor FP associado à EM, tal que a sua posição estimada será dada pelas coordenadas geográficas (latitude e longitude) do registro do CDB com a maior similaridade ao vetor FP fornecido. Vale ressaltar que uma busca em todos os registros do CDB pode consumir muito tempo e fazer com que a localização se torne inviável. Nesse cenário, existem várias alternativas que podem ser utilizadas para reduzir o espaço de busca, tais como algoritmos genéticos (CAMPOS; LOVISOLO, 2019) e filtros (BENIKOVSKY; BRIDA; MACHAJ, 2011). Em (TIMOTEO et al., 2016), parâmetros de atraso de onda são utilizados para reduzir o espaço de busca em uma implementação de um método de localização do tipo FP em uma rede GSM.

A Fig. 13 ilustra o diagrama simplificado do funcionamento da fase *on-line* da técnica de localização baseada em FP conforme descrito em (CAMPOS; LOVISOLO, 2011), no qual as setas numeradas representam as etapas do processo. Na etapa 1, a aplicação cliente faz uma requisição de localização da EM ao servidor de localização. Em seguida, na etapa 2, o servidor de localização solicita à rede de acesso medições do usuário móvel procurado. Na etapa 3, a rede de acesso retorna as medições realizadas e o servidor de localização faz a composição do vetor FP do usuário móvel procurado. Na etapa 4, o servidor de localização solicita ao CDB a área de busca reduzida, segundo alguma técnica de redução de espaço de busca. Na quinta etapa, o servidor recebe o resultado da busca e usa uma função de comparação para estimar a posição da EM. Finalmente, na última etapa, o servidor envia a localização estimada à aplicação cliente.

Vale ressaltar que, mesmo utilizando algoritmos de ML, os atuais métodos de localização do tipo FP enfrentam alguns desafios devido à natureza estocástica do sinal de RF. Por exemplo, geralmente sistemas FP baseados em RSSI perdem a localização do usuário, devido à variação abrupta do RSSI quando o usuário sai de um ambiente *indoor* para um

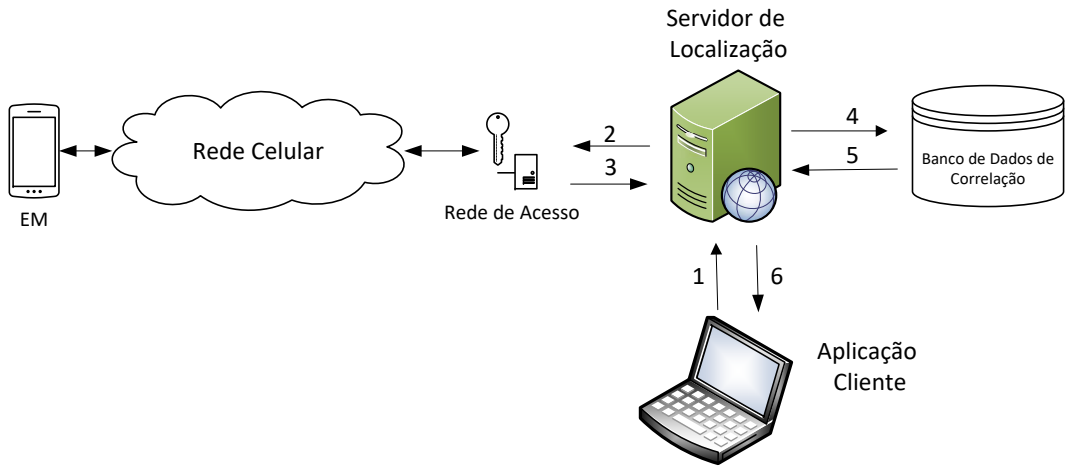

Figura 13 – Diagrama da fase *on-line* da técnica de radiolocalização baseada em FP (Adaptação de (CAMPOS; LOVISOLO, 2011)).

outdoor (GOLDONI et al., 2019) . Outro problema diz respeito à variação das medições de RSSI quando EMs de diferentes fabricantes são usadas. Essa variação de RSSI pode ser encontrada até mesmo com EMs diferentes do mesmo fabricante (HUANG; LIN, 2020; LUI et al., 2011).

Por último, a escalabilidade do sistema de localização é outro fator importante, pois *smartphones* estão cada vez mais populares e o aumento da quantidade de usuários exige um crescimento no número de ERBs da rede celular (KIM et al., 2013). Assim, temos um maior número de *grids* de localização e isso traz um sério problema para os modelos de localização atuais, pois a criação do *grid* depende do modelo preditor de RSSI, o que compromete a escalabilidade desses sistemas. Assim, face ao exposto, na próxima Seção será proposta uma nova técnica de radiolocalização que utiliza diferenças de RSSI e PD com o objetivo de diminuir os efeitos dos problemas anteriormente mencionados.

5.3 PROPOSTA DE RADIOLOCALIZAÇÃO BASEADA EM FP ANGULAR

Nesta Seção, é apresentada uma técnica de radiolocalização do tipo FP escalável, que utiliza algoritmos de ML para realizar a predição dos ângulos horizontais de chegada obtidos

a partir dos sinais de RF com o intuito de estimar a posição da EM. Em vez de empregar os valores absolutos de RSSI, a técnica proposta utiliza as diferenças de RSSI e atraso de propagação (PD, *propagation delay)* para obter o ângulo horizontal da EM. A ideia principal da proposta é que as diferenças de RSSI das ERBs posicionadas geograficamente no mesmo local são afetadas de forma semelhante pela atenuação do edifício e de outros obstáculos. Outra vantagem é que, utilizando as diferenças de RSSI, a influência do tipo de dispositivo móvel ou do fabricante é diminuída, pois alguns dispositivos medem os RSSIs com mais acurácia e podem alterar o comportamento de modelos de localização baseados no valor absoluto do RSSI.

Antes de detalharmos as etapas da proposta de radiolocalização baseada em FP angular, é necessário definirmos o conceito de grupo de ERBs, que será denotado neste trabalho por G. Um grupo G de ERBs é definido como o conjunto de N_e ERBs situadas nas mesmas coordenadas geográficas (latitude e longitude). Conforme será explicado adiante, há restrições para o número mínimo de grupos de ERBs e a quantidade mínima de ERBs em cada grupo.

A técnica de radiolocalização baseada em FP angular pode ser implementada em oito etapas, descritas no Algoritmo 8. Tais etapas são organizadas em duas fases, denominadas *off-line* e *on-line*, tal qual a técnica FP tradicional. A fase *off-line* contém as etapas de 1 a 4, enquanto a fase *on-line* inclui as etapas de 5 a 8. Na fase *off-line*, não é realizada coleta de medições de RSSI e PD relacionadas à EM procurada.

A primeira etapa da técnica proposta é definir a área de localização. A Fig. 14 ilustra a área de localização definida como um *grid* de q quadrículas. Assim como na técnica FP tradicional, cada quadrícula do *grid* está associada à posição geográfica do seu centro. A resolução do *grid* é um limitador da acurácia da técnica, pois assumimos que a EM está no centro da quadrÍcula.

Definida a área de localização, a próxima etapa consiste na coleta de medições de RSSI

Algoritmo 8: Descrição da técnica proposta baseada em FP angular.

1: Definir a área (*grid*) de localização;
2: Coletar medições do *scanner* (RSSI e PD) na área de localização para construir a base de treinamento;
3: Treinar os algoritmos de ML para a obtenção das funções hipótese para os N_p grupos de ERBs ($f_i(\cdot)$ é a função hipótese do i-ésimo grupo G_i , $i = 1, 2, \ldots, N_p$);
4: Gerar o CDB com os valores dos ângulos horizontais previstos e PDs para cada quadrícula do *grid* de localização;
5: Coletar os níveis de RSSI e de PD do usuário móvel procurado em relação a todas as ERBs;
6: Dados os níveis de RSSI e PD, fazer a predição do ângulo horizontal (θ_i) para cada G_i usando $f_i(\cdot)$;
7: Gerar um espaço de busca reduzida S_r no CDB aplicando filtragem em função dos PDs;
8: Buscar em S_r a quadrícula com menor distância Euclidiana considerando os ângulos horizontais preditos e medidos.
=0

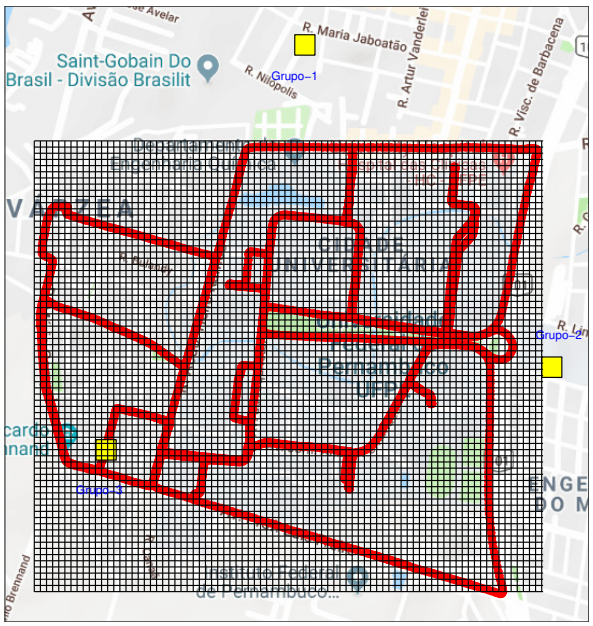

Figura 14 – Área de localização (definida por seu *grid*) com indicação da rota do *drive-test* realizado para a obtenção das medições *outdoor* (trajetória em pontos vermelhos). Os quadrados amarelos indicam a localização dos grupos de ERBs.

e PD nessa área. O PD é um parâmetro de atraso de onda (ida e volta) da rede celular e seu valor consiste em um número inteiro que varia no intervalo de 0 a 56. Tais números representam um intervalo de distâncias da EM a ERB em passos de 234 m. Assim, o valor PD=0 é mapeado para o intervalo $[0, 234[\, m$, enquanto o valor PD=1 representa

o intervalo $[234, 468[\, m$ e assim por diante. O valor máximo de PD é 56, que representa distâncias maiores que $13,1\, km$.

Assumindo que L amostras de campo são coletadas na área de localização, cada uma é composta por RSSIs e PDs que são obtidos para cada ERB e grupo de ERBs, respectivamente. Considere que $\mathbf{P}_\ell$ é o vetor que representa o conjunto de RSSIs (em dBm) para ℓ-ésima amostra coletada, tal que

$$\mathbf{P}_\ell = \{P_{\ell,i}^{(u)}\}\,, i = 1, \ldots, N_p; u = 1, \ldots, N_e\,, \tag{5.1}$$

em que $P_{\ell,i}^{u}$ é o RSSI medido em relação à u-ésima ERB pertencente ao i-ésimo grupo. Ademais, definimos Γ_ℓ como o conjunto de PD medidos tal que

$$\Gamma_\ell = \{\gamma_{\ell,i}\}\,, \quad i = 1, \ldots, N_p\,, \tag{5.2}$$

em que $\gamma_{\ell,i}$ é a medida de PD para o i-ésimo grupo de ERBs. Por exemplo, assuma que há $N_p = 2$ grupos, cada um com $N_e = 3$ ERBs. Nesse cenário, cada amostra coletada no *drive-test* é composta por suas coordenadas geográficas, um vetor de RSSIs medidos $\mathbf{P}_\ell$, dado por

$$\mathbf{P}_\ell = \{P_{\ell,1}^{(1)}; P_{\ell,1}^{(2)}; P_{\ell,1}^{(3)}; P_{\ell,2}^{(1)}; P_{\ell,2}^{(2)}; P_{\ell,2}^{(3)}\}$$

e um conjunto de PD medidos $\Gamma_\ell = \{\gamma_{\ell,1}; \gamma_{\ell,2}\}$.

Após a realização das medições, o próximo passo é treinar os algoritmos de ML, um para cada grupo G_i, para prever o ângulo horizontal da EM (alvo), denotado por $\widehat{\theta}_i$. Vale ressaltar que, devido à grande variação no AoA, é necessária uma função de transformação para suavizar a variável alvo. Uma função exponencial decrescente pode ser usada para esta tarefa (JR, 1985; KUNCHEVA, 2004a). Assim, para a suavização do AoA, a função de transformação adotada, denotada por $t_a(x)$, é dada por

$$t_a(x) = \frac{1}{1 + e^x}\,. \tag{5.3}$$

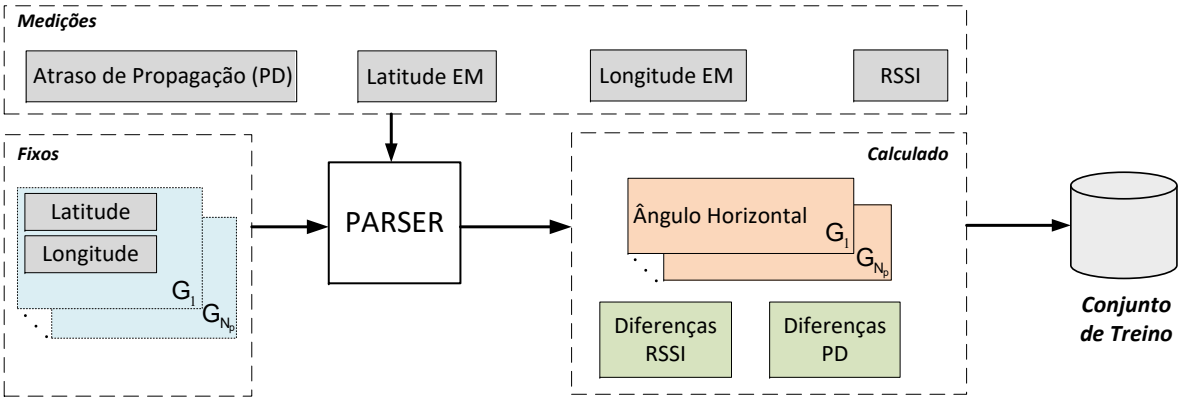

Figura 15 – Diagrama ilustrativo do processo de construção do conjunto de treinamento (referente a cada grupo G_i de ERBs) do algoritmo de ML.

A Fig. 15 mostra o diagrama do processo de formação de conjuntos de treinamento considerando N_p grupos de ERBs. Qualquer algoritmo de regressão pode ser treinado, mas o algoritmo k-NN foi escolhido devido ao seu bom desempenho em problemas de localização (WANG et al., 2016; GUO et al., 2018). O conjunto de treinamento é criado a partir de medições de dados (RSSI, PD, latitude e longitude) coletadas no campo e dados fixos (coordenadas geográficas de cada grupo de ERBs). Para gerar esse conjunto, foi desenvolvido um programa de *parser* usando a linguagem Python, para extrair e processar os dados.

Ainda na Fig. 15, podemos ver que os atributos do conjunto de treinamento são compostos pelas diferenças de RSSI e PD. Considerando a ℓ-ésima amostra coletada no campo, o vetor de diferenças de RSSI $\Delta_\ell = \{\Delta_{\ell,i}^{(uj)}\}$ é calculado, sendo seus elementos dados por

$$\Delta_{\ell,i}^{(uj)} = P_{\ell,i}^{(u)} - P_{\ell,i}^{(j)},\ u, j = 1, 2, \ldots, N_e, u \neq j, u < j\,, \tag{5.4}$$

em que $P_{\ell,i}^{(u)}$ e $P_{\ell,i}^{(j)}$ são os RSSIs das u-ésima e j-ésima ERBs, respectivamente, as quais pertencem ao grupo $G_i\,, i = 1, \ldots, N_p$. Assim, as diferenças de RSSI são coletadas para ERBs que estão no mesmo grupo. Ademais, temos o vetor de diferenças de PD $\mathbf{T}_\ell$, que

também é usado no conjunto de treinamento, e seus elementos $\tau_\ell^{(mn)}$ são definidos como

$$\tau_\ell^{(mn)} = \gamma_\ell^{(m)} - \gamma_\ell^{(n)},\ m, n = 1, 2, \ldots, N_p; m \neq n, m < n, \tag{5.5}$$

em que $\gamma_\ell^{(m)}$ e $\gamma_\ell^{(n)}$ são os menores PDs relacionados aos grupos G_m and G_n, respectivamente. Vale ressaltar que as diferenças de PDs referem-se apenas a grupos distintos de ERBs. O menor PD é considerado, pois é muito provável que a ERB com o menor atraso possua a melhor linha de visada em relação ao ponto de medição.

Considere agora o exemplo no qual temos $N_p = 3$ grupos de $N_e = 3$ ERBs. Nesse cenário, temos o vetor de diferenças de RSSI

$$\Delta_\ell = \{\Delta_{\ell,1}^{(12)}; \Delta_{\ell,1}^{(13)}; \Delta_{\ell,1}^{(23)}; \Delta_{\ell,2}^{(12)}; \Delta_{\ell,2}^{(13)}; \Delta_{\ell,2}^{(23)}; \Delta_{\ell,3}^{(12)}; \Delta_{\ell,3}^{(13)}; \Delta_{\ell,3}^{(23)}\},$$

ou seja, temos o número de combinações de tamanho 2 a partir de N_e elementos distintos para cada grupo de ERBs. De maneira semelhante, temos um conjunto de diferenças de PD

$$\mathbf{T}_\ell = \{\tau_\ell^{(12)}; \tau_\ell^{(13)}; \tau_\ell^{(23)}\}$$

com diferentes combinações de tamanho 2 utilizando os N_p elementos.

Ao término da construção do conjunto de treinamento, os dados de treinamento consistem em L vetores compostos pelas diferenças de RSSI $\{\Delta_\ell\}_{\ell=1}^{L}$ e diferenças de PD $\{\mathbf{T}_\ell\}_{\ell=1}^{L}$. É importante mencionar que as diferenças de RSSI são usadas para permitir localizações *indoor* e *outdoor*. A ideia básica de usar a diferença de RSSI, e não o valor absoluto diretamente, é que as atenuações sofridas pelos sinais devido às barreiras físicas do ambiente serão aproximadamente iguais, pois os sinais de RF têm o mesmo ponto de origem e aproximadamente as mesmas frequências (ANDERSEN; RAPPAPORT, 1995).

Outro aspecto relevante é que o valor absoluto dos níveis de RSSI pode variar dependendo do fabricante do dispositivo móvel ou até mesmo do modelo utilizado. Essa variação pode limitar a acurácia dos sistemas de posicionamento baseados em FP (HUANG; LIN,

2020; LUI et al., 2011). No entanto, a influência dessa variação no nível do sinal diminui quando as diferenças de RSSI são assumidas, uma vez que estas permanecem praticamente inalteradas, apesar da variação do nível absoluto do sinal (HOSSAIN et al., 2011).

O quarto passo da técnica proposta é gerar o CDB associado à área de localização. O CDB é definido como o espaço de busca geral, representado pelo conjunto S de 3-tuplas, dado por

$$S = \{(\mathbf{p}_g, \Theta_g, \mathbf{t}_g) \in R^2 \times R^{N_p} \times R^W,\ g = 1, 2, \ldots, q\}\,, \tag{5.6}$$

em que o vetor $\mathbf{p}_g = [p_g^{(a)}, p_g^{(b)}]$ representa as coordenadas geográficas da g-ésima quadrícula do *grid* de localização, sendo $p_g^{(a)}$ a longitude e $p_g^{(b)}$, a latitude do centro da quadrícula. Além disso, o vetor

$$\Theta_g = \{\theta_{g,i}\},\ i = 1, 2, \ldots, N_p\,,$$

representa o ângulo horizontal da g-ésima quadrícula em relação a cada um dos N_p grupos. A Fig. 16 ilustra como o ângulo horizontal $\theta_{g,i}$ é medido no sentido horário, tendo o Norte Verdadeiro como referência. Finalmente, o vetor

$$\mathbf{t}_g = \{\tau_g^{(mn)}\},\ m, n = 1, 2, \ldots, N_p; m \neq n, m < n$$

representa as diferenças de PD para cada par de grupos de ERBs, em que a dimensão de $\mathbf{t}_g$, denotada por W em (5.6), é a combinação dois a dois de N_p elementos distintos (grupos de ERBs).

A fase *on-line* tem início com o quinto passo, no qual o RSSI e o PD da EM procurada, denotados por

$$\mathbf{P}_{MU} = \{P_{MU,i}^{(u)}\}\,, i = 1, \ldots, N_p; u = 1, \ldots, N_e$$

e

$$\Gamma_{MU} = \{\gamma_{MU,i}\}\,, i = 1, \ldots, N_p\,,$$

respectivamente, são medidos. O RSSI $P_{MU,i}^{(u)}$ é a medição da intensidade do nível de sinal de RF que a EM recebe em relação à u-ésima ERB, pertencente ao grupo G_i, enquanto

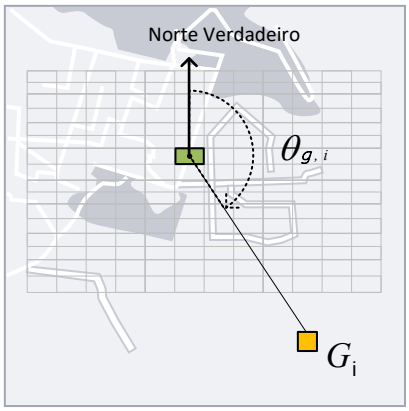

Figura 16 – Geração do CDB: Azimute $\theta_{g,i}$ da g-ésima quadrícula do *grid* em relação ao grupo G_i.

que $\gamma_{MU,i}$ é o PD medido em relação ao i-ésimo grupo de ERBs. Ambas as medidas são obtidas por meio da rede de acesso, como detalhado em (CAMPOS; LOVISOLO, 2011).

Dadas as medições em relação à EM procurada, o vetor de diferenças de RSSI $\Delta_{MU} = \{\Delta_{MU,i}^{(uj)}\}$ é obtido, cujos elementos são dados por

$$\Delta_{MU,i}^{(uj)} = P_{MU,i}^{(u)} - P_{MU,i}^{(j)}\,,\; u,j = 1,2,\ldots,N_e, u \neq j, u < j\,, \tag{5.7}$$

enquanto que o vetor de diferenças de PD $\mathbf{T}_{MU}$ tem seus elementos $\tau_{MU}^{(mn)}$ dados por

$$\tau_{MU}^{(mn)} = \gamma_{MU}^{(m)} - \gamma_{MU}^{(n)},\; m,n = 1,2,\ldots,N_p, m \neq n, m < n\,. \tag{5.8}$$

Após o cálculo de Δ_{MU} e $\mathbf{T}_{MU}$, esses vetores são usados como entradas para as funções hipótese $f_i(\cdot)$ com o objetivo de realizar a predição dos N ângulos horizontais, um para cada grupo G_i, denotados por

$$\widehat{\Theta}_{MU} = \{\widehat{\theta}_{MU,i}\},\; i = 1,2,\ldots,N_p\,.$$

Depois da previsão dos ângulos horizontais da EM, o próximo passo é aplicar o Algoritmo 9 para reduzir a área de busca S usando o filtro de CDB (CAMPOS; LOVISOLO, 2011). A ideia central do Algoritmo 9 é maximizar o número de correspondências entre os PDs esperados e medidos. Assim, considerando todas as quadrículas do *grid*, o algoritmo tem início pela obtenção do número máximo de correspondências n_{max} de modo que

Algoritmo 9: Filtro CDB para gerar a área de busca reduzida S_r

Entrada:
$\mathbf{T}_{MU}$, vetor de diferenças de PD referente à EM
S, área de localização com q quadrículas
Saída:
S_r, Área de busca reduzida

```
1  início
2  |  n_max ← número máximo de acertos (todas as quadrículas)
3  |  m_g ← NrMatches(T_MU, T_g),   g = 1, 2, ..., q
4  |  se n_max > 0 então
5  |  |  S_r ← quadrículas com m_g = n_max
6  |  senão
7  |  |  S_r ← S
8  |  fim
9  fim
10 retorna S_r
```

$\tau_{MU}^{(mn)} = \tau_g^{(mn)}$, em que $\tau_{MU}^{(mn)}$ é a diferença de PDs medidos entre os grupos G_m e G_n e $\tau_g^{(mn)}$ é a diferença esperada de PD entre G_m e G_n para a g-ésima quadrícula. Em seguida, todas as quadrículas em que o número de correspondências entre os vetores $\mathbf{T}_{MU}$ e $\mathbf{T}_g$ é igual a n_{max}, são incluídas no subconjunto S_r (linha 5). Se $n_{max} = 0$, não há espaço reduzido e o espaço completo S é atribuído a S_r (linha 7). Nesse caso, a área de localização original S é obtida como saída do Algoritmo 9.

O último passo da técnica proposta, conforme Algoritmo 8, é estimar a posição da EM usando distância Euclidiana como medida de similaridade. A distância Euclidiana d_g entre os vetores $\widehat{\Theta}_{MU}$ e Θ_g pode ser expressa como

$$d_g = \sqrt{\sum_{i=1}^{N_p} \left(\widehat{\theta}_{MU,i} - \theta_{g,i}\right)^2} \ , \ g = 1, 2, \ldots, q \ , \tag{5.9}$$

em que $\widehat{\theta}_{MU,i}$ é o ângulo horizontal previsto da EM em relação ao grupo G_i e $\theta_{g,i}$ é o azimute da g-ésima quadrícula em relação ao grupo G_i. Finalmente, a melhor posição estimada é a quadrícula que possui a menor distância Euclidiana d_g, considerando apenas as células no espaço reduzido S_r.

Por fim, detalhados os passos para a implementação da técnica de localização proposta, serão apresentados a seguir os resultados numéricos de uma implementação baseada em

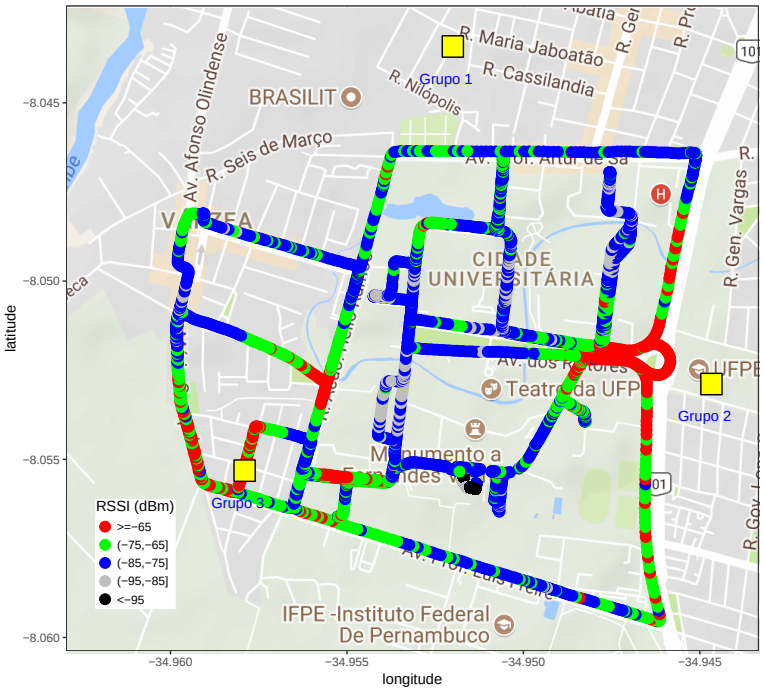

Figura 17 – Ambiente urbano na cidade de Recife-PE com a indicação das medições do melhor servidor (*best server*) e da localização das ERBs.

medições de campo reais obtidas em uma região urbana da cidade de Recife-PE, considerando ambientes *indoor* e *outdoor*, tanto ruidosos como não-ruidosos.

5.3.1 Resultados Numéricos

Neste trabalho, o desempenho da técnica de localização proposta é avaliado por meio de simulações computacionais usando a linguagem Python, com ênfase no pacote *scikit-learn* (PEDREGOSA et al., 2011). Os algoritmos utilizados neste trabalho são supervisionados, o que implica necessidade de dados para treinamento. Com isso em mente, foram realizadas medições *indoor* e *outdoor* de RSSIs e PDs do enlace de descida (enlace ERB-EM) em uma portadora na faixa de 1,8 GHz de uma rede de telefonia celular de terceira geração (3G) que utiliza a interface aérea por acesso múltiplo por divisão de código de banda larga (W-CDMA, *wideband code-division multiple access)*. A área de localização considerada foi uma região urbana de $1,6\,km^2$ na cidade de Recife-PE.

As medições *outdoor* foram obtidas por meio de um *drive test*. A Fig. 17 ilustra a região da cidade onde foram realizadas as medições. Os quadrados amarelos indicam as

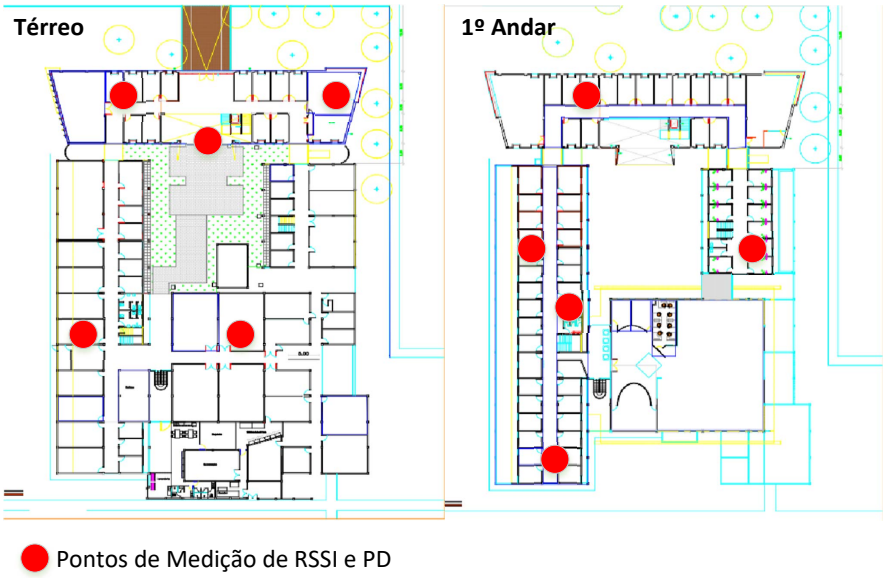

Figura 18 – Planta baixa do local onde foram realizadas as medições *indoor*. As medições estão representadas por círculos vermelhos.

posições dos grupos de ERBs. Os níveis de RSSI, medidos em dBm, do melhor servidor (*best server*) são indicados pela legenda de cores. No total, foram realizadas 6.615 medições externas.

Em relação às medições *indoor*, foram realizadas 3.064 medições internas no Centro de Informática da UFPE, as quais estão distribuídas em dez pontos diferentes. A Fig. 18 apresenta a planta baixa do prédio no qual foram realizadas as medições. Os pontos de medições de RSSI e PD são representados por círculos vermelhos. Todas as medições *indoor* e *outdoor* foram realizadas com um equipamento *scanner* DRT4301A fabricado pela DRT (*Digital Receiver Technology*)[1].

Para comparar a acurácia da técnica proposta com a técnica de FP baseada em níveis absolutos de RSSI, ambas as técnicas foram implementadas. A primeira implementação é a nossa proposta, a qual é descrita no Algoritmo 8 e é denominada como *Fingerprinting* baseado no ângulo de chegada (FP-AoA). A segunda implementação é a abordagem clássica de FP mostrada em (CAMPOS; LOVISOLO, 2011), que é denominada como *Fingerprinting*

[1] <http://www.drti.com/>

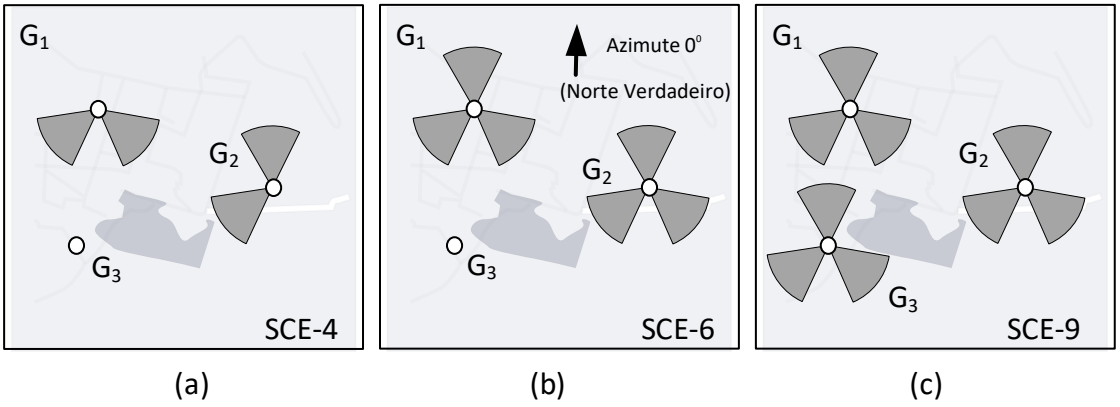

Figura 19 – ERBs consideradas para cada cenário: (a) SCE-4, (b) SCE-6 e (c) SCE-9.

baseado no indicador de intensidade do nível de sinal (FP-RSSI). Em relação a ambos os métodos, vale mencionar dois detalhes sobre a implementação: (*i*) o algoritmo k-NN é usado para construir o CDB; (*ii*) o procedimento descrito no Algoritmo 9 é empregado para reduzir o espaço de busca.

As duas técnicas mencionadas anteriormente foram implantadas em três cenários diferentes relacionados ao número de ERBs na área urbana mostrada na Fig. 17. Esses cenários são nomeados como SCE-N_b, em que N_b representa o número total de ERBs na área de localização. A Fig. 19 ilustra todos os cenários considerados neste trabalho. Por exemplo, a Fig. 19(a) indica que quatro ERBs são usadas (duas do grupo G_1 e duas do grupo G_2). Os setores circulares de cor cinza representam as antenas, uma por ERB, com um ângulo horizontal de 63°. Além disso, para todos os grupos, os azimutes das antenas são 0°, 120° e 240°. O azimute 0° é indicado na Fig. 19(b). Os outros azimutes (120° e 240°) são medidos no sentido horário a partir do azimute 0°. Os cenários SCE-4 e SCE-9 seguem a mesma orientação de azimute que o cenário SCE-6.

Ainda na Fig. 19, podemos ver que várias combinações de ERBs para os cenários SCE-4 e SCE-6 são viáveis. Vamos assumir a notação $G_i(A_u, A_j)$, em que A_u e A_j indicam, respectivamente, os azimutes da u-ésima e j-ésima ERB do grupo G_i. Por exemplo, no cenário SCE-4, uma combinação possível de ERBs é $[G_1(0°, 120°); G_3(0°, 120°)]$. No entanto, a combinação de ERBs selecionadas para o cenário SCE-4 em nosso experimento

foi [$G_1(120°, 240°)$;

$G_2(0°, 240°)$]. Para o cenário SCE-6, a melhor combinação considera todas as ERBs dos grupos G_1 e G_2, enquanto que para o cenário SCE-9, todas as ERBs de todos os grupos representam a melhor configuração. A razão para escolher as configurações mencionadas anteriormente (e mostradas na Fig. 19) é o fato de a técnica FP-RSSI ter tido o melhor ajuste (*tuning*) com essas configurações. Devido ao alto consumo de tempo em testar as configurações, a seleção das ERBs foi feita usando apenas 50% das amostras, as quais foram escolhidas aleatoriamente.

Para treinar os modelos de ML, 6.615 medições *outdoor* foram realizadas experimentalmente e divididas em dois subconjuntos: um conjunto de dados de treinamento com 90 % das medições e um conjunto de dados de teste com os 10 % restantes . Além disso, 3.064 medições *indoor* foram realizadas para verificar a acurácia dos métodos de localização para usuários em um ambiente *indoor*. Vale ressaltar que as medições *indoor* são usadas apenas para o estágio de teste.

Para comparar as técnicas de localização FP-AoA e FP-RSSI, definimos o erro de predição de posição η como a diferença (em metros) entre os pontos reais e os preditos. A técnica de validação denominada validação cruzada (*cross-validation*) é aplicada para verificar a capacidade de generalização das técnicas de localização. Neste trabalho, consideramos $K = 10$ e o erro médio de predição de posição $\bar{\eta}$ definido como

$$\bar{\eta} = \sqrt{\frac{1}{K}\sum_{j=1}^{K}\eta_j}\,, \tag{5.10}$$

em que η_j é o erro de predição de posição para o j-ésimo *fold* de teste.

Além da capacidade de generalização, outro aspecto importante é o comportamento da técnica de localização na presença de *outliers*[2]. Esses valores discrepantes podem ser gerados pelos efeitos de difração, reflexão ou absorção das ondas devido à existência de edifícios e outros obstáculos no ambiente. Portanto, para verificar o impacto dos *outliers*,

[2] Valor atípico (*outlier*) é uma observação que apresenta um grande afastamento em relação aos demais pontos da série.

Tabela 13 – Erro médio de predição de posição $\bar{\eta}$, medido em metros, para as técnicas de localização FP-AoA e FP-RSSI nos ambientes *outdoor*, *indoor* e *indoor-outdoor*, ruidosos (R) e não-ruidosos (N-R). A média é obtida por meio de validação cruzada com $k = 10$ *folds* e considerando os cenários SCE-4, SCE-6 e SCE-9.

		SCE-4		SCE-6		SCE-9	
Ambiente	Tec. Loc.	(N-R)	(R)	(N-R)	(R)	(N-R)	(R)
Indoor-Outdoor	FP-AoA	107,02	127,51	102,51	140,22	71,91	71,91
	FP-RSSI	127,94	150,22	129,58	150,41	84,74	84,74
Outdoor	FP-AoA	85,17	115,78	82,94	132,75	57,78	57,78
	FP-RSSI	96,45	162,41	91,45	165,14	47,65	47,65
Indoor	FP-AoA	117,12	126,08	111,55	143,68	78,45	78,45
	FP-RSSI	142,50	144,58	147,08	143,60	101,88	101,88

assumimos essas variações como um ruído que segue uma distribuição de Rayleigh com média de 12 dB e desvio padrão igual a 7 dB, para simular diferentes níveis de ruído nas medições da EM (SUZUKI, 1977). Consideramos que essa distribuição de Rayleigh simula uma mudança abrupta no ambiente do usuário, pois esses parâmetros podem ser vistos como a atenuação média devido à construção de paredes (YARKONY; BLAUNSTEIN, 2006; SEIDEL; RAPPAPORT, 1992). Logo, por uma questão de simplicidade, usaremos apenas ruído para nos referir à distribuição de Rayleigh mencionada anteriormente.

A Tab. 13 mostra o erro médio de predição de posição $\bar{\eta}$ (em metros), definido em (5.10), para as técnicas de localização FP-AoA e FP-RSSI em relação aos 10 *folds* da validação cruzada. É importante destacar que quanto menor o erro de predição, melhor a acurácia da técnica de localização. Assumimos três cenários quanto ao número de ERB consideradas, conforme ilustrado na Fig. 19. Para cada cenário, há três ambientes onde a EM procurada se encontra: (*i*) *outdoor*, quando a EM não está dentro de alguma edificação; (*ii*) *indoor*, quando a EM está no interior de alguma edificação; e (*iii*) *indoor-outdoor*, quando a EM pode estar ao ar livre ou dentro de uma edificação. Por fim, para cada cenário e ambiente, podemos ter ou não a presença de ruído. Conforme explicado anteriormente, a presença de ruído nas medições é modelada por uma distribuição de Rayleigh com média

de 12 dB e desvio padrão de 7 dB. Dessa forma, os ambientes ruidosos e não-ruidosos são identificados por (R) e (N-R), respectivamente.

Inicialmente, podemos ver que, na maioria dos casos, a técnica FP-AoA supera a técnica FP-RSSI, com exceção dos cenários SCE-6 *indoor* e SCE-9 *outdoor*. Em seguida, podemos observar que, em geral, as acurácias das técnicas FP-AoA e FP-RSSI se tornam melhores com o aumento do número de ERBs. Por exemplo, considerando apenas o ambiente *outdoor* sem ruído, a técnica FP-AoA tem um melhor desempenho nos cenários SCE-4 e SCE-6, para os quais a técnica FP-AoA fornece erros médios de predição de $85,17\,m$ e $82,94\,m$, respectivamente. Para os mesmos cenários, a técnica FP-RSSI atinge $\bar{\eta} = 96,45\,m$ para o SCE-4 e $\bar{\eta} = 91,45\,m$ para o SCE-6. Dessa maneira, podemos ver que a mudança de cenário (de SCE-4 para SCE-6) gera, no máximo, uma diferença de erro de predição de $5,0\,m$ quando a técnica FP-RSSI é aplicada. Por outro lado, quando são consideradas nove ERBs (SCE-9), a técnica FP-RSSI supera a técnica FP-AoA, com erros médios de predição de $\bar{\eta} = 47,65\,m$ para o primeiro e $\bar{\eta} = 57,78\,m$ para o segundo.

Ainda na Tab. 13, podemos notar que as duas técnicas de localização tem sua acurácia comprometida nos cenários SCE-4 e SCE-6, ambos *outdoor*, quando a presença do ruído é considerada. Para a técnica FP-AoA, o erro médio de predição aumenta 35,9% ($85,17\,m$ para $115,78\,m$) para o cenário SCE-4 e 60% ($82,94\,m$ para $132,75\,m$) para o cenário SCE-6. Por outro lado, o aumento de $\bar{\eta}$ é ainda maior para a técnica FP-RSSI, onde temos um incremento de $68,4\%$ ($96,45\,m$ para $162,41\,m$) e $80,6\%$ ($91,4\,m$ para $165,14\,m$) para os cenários SCE-4 e SCE-6, respectivamente. Portanto, nos cenários SCE-4 e SCE-6, ambos *outdoor*, a técnica de localização FP-RSSI é mais sensível ao ruído que a técnica FP-AoA. No entanto, quando o cenário SCE-9 é assumido, ambas as técnicas não são sensíveis ao ruído. Essa característica pode ser explicada por meio da análise do comportamento do Algoritmo 9. Esse algoritmo é responsável por reduzir a área de busca usando diferenças de PD. Assim, um alto número de ERBs implica um filtro mais

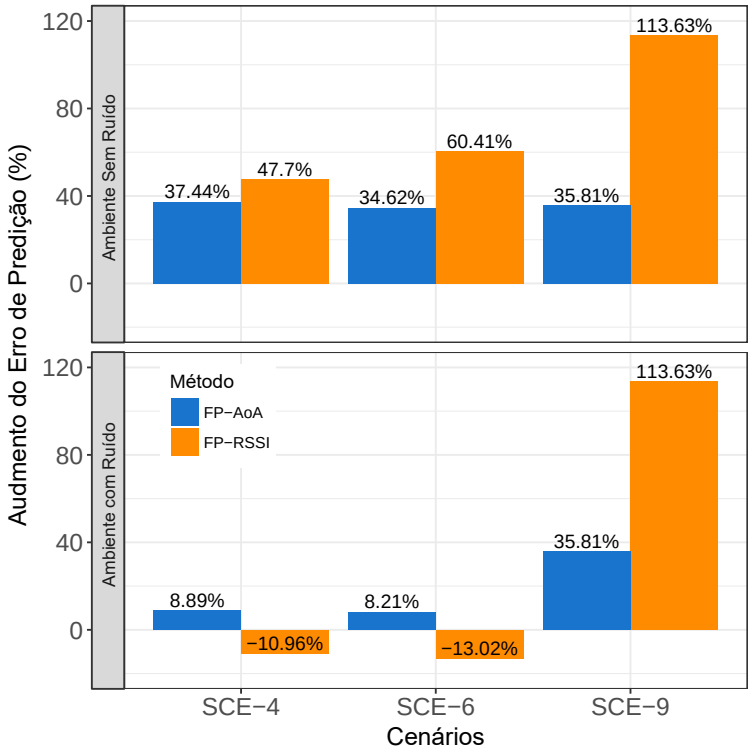

Figura 20 – Aumento do erro médio de predição (em %) para as técnicas FP-AoA e FP-RSSI em todos os cenários (ruidosos e não-ruidosos), quando a EM procurada se desloca do ambiente *outdoor* para o *indoor*.

restritivo e, consequentemente, uma área reduzida menor, minimizando o efeito causado pelo ruído.

Considerando apenas o ambiente *indoor*, podemos ver que, na maioria dos casos, o erro de predição da técnica FP-AoA é menor que o erro da técnica FP-RSSI, no intervalo de $12,8\%$ a $24,6\%$, conforme pode ser calculado a partir dos dados da Tab. 13. Nesse caso, vale ressaltar que a FP-AoA é uma técnica de localização *outdoor* que pode ser usada mesmo se a EM estiver em um ambiente *indoor*. Em outras palavras, apesar de a técnica FP-AoA ter sido originalmente projetada para localização *outdoor*, podemos aplicá-la para predizer a posição da EM dentro de um ambiente *indoor* (não de maneira exata, pois não é possível identificar o andar no caso de edifícios).

Como esperávamos, o desempenho de ambas as técnicas de localização piora quando a EM procurada passa de um ambiente *outdoor* para um *indoor*. Em vista disso, a Fig. 20 mostra o aumento do erro de predição (em %) para as técnicas de localização FP-AoA e

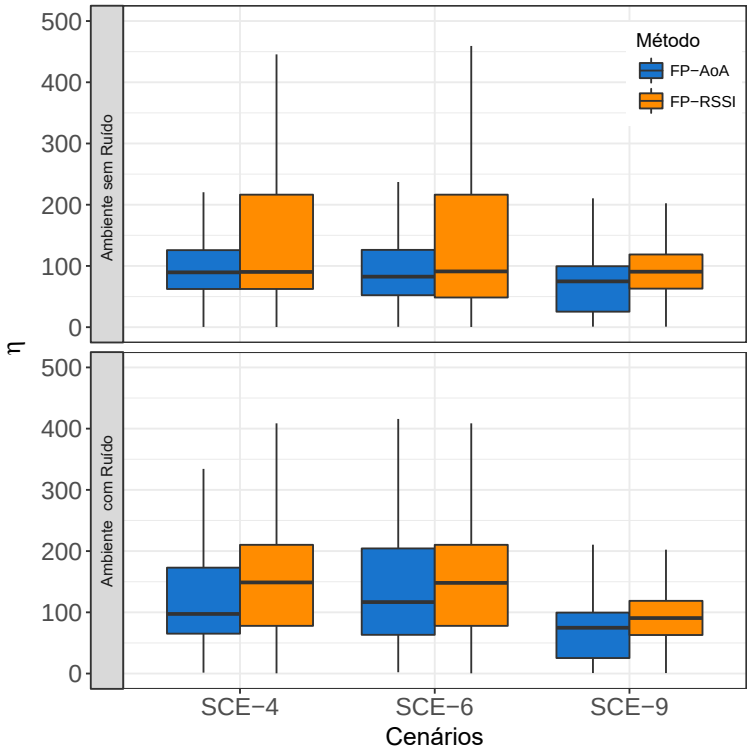

Figura 21 – Distribuição do erro de predição de posição η (em metros) para as técnicas FP-AoA e FP-RSSI para os ambientes *indoor-outdoor*, ruidoso e não-ruidoso, em todos os cenários.

FP-RSSI em todos os cenários ruidosos e não-ruidosos, quando a EM se move de um ambiente *outdoor* para um ambiente *indoor*. Podemos ver que o erro de predição da técnica FP-RSSI tem aumentos percentuais maiores do que os da técnica FP-AoA para todos os cenários, onde o aumento mais significativo ocorre no cenário SCE-6 não-ruidoso. Nesse caso, a técnica FP-RSSI tem um aumento de $60,83\%$ (de $91,45\,m$ para $147,08\,m$). Também na Fig. 20, é possível observar que os aumentos do erro são menores em ambientes ruidosos, exceto no cenário SCE-9. Isso não significa que as técnicas de localização funcionam melhor em um ambiente ruidoso, mas que a redução do espaço de busca se torna mais relevante para as técnicas FP-AoA e FP-RSSI. Portanto, o algoritmo de redução de espaço de busca desempenha um papel essencial em ambientes ruidosos e o tipo de mudança de ambiente tem um efeito menor sobre o algoritmo, uma vez que ele depende de PD do grupo.

Em relação aos algoritmos de predição (técnicas de localização incluídas), é muito im-

portante analisar a variação do erro. A Fig. 21 mostra a distribuição do erro de predição de posição η para as técnicas FP-AoA e FP-RSSI em ambientes *indoor* e *outdoor* ruidosos e não ruidosos para todos os cenários usando *box plots*. Devido à técnica de validação cruzada, todas as instâncias (amostras *indoor* e *outdoor*) são usadas como instâncias de teste em um dos dez *folds*. No *box plot*, um aspecto importante é o intervalo de distribuição (altura da caixa) dos erros de predição. Quanto menor a largura da caixa, maior a estabilidade da técnica de localização. Na Fig. 21, vemos que, para ambas as técnicas de localização, há um intervalo mais restrito para o cenário SCE-9. Portanto, as predições para ambos os algoritmos são mais estáveis quando comparamos o cenário SCE-9 com os outros cenários. Além disso, ainda no cenário SCE-9, observamos que as predições de FP-RSSI são menos instáveis do que as da técnica FP-AoA. Por outro lado, para os cenários SCE-4 e SCE-6, os intervalos de predições da técnica FP-AoA são mais restritos que os da técnica FP-RSSI em ambientes não ruidosos. Para os cenários SCE-4 e SCE-6 não-ruidosos, podemos assumir que as técnicas de localização são equivalentes em termos de estabilidade.

Outro aspecto importante a ser analisado é a capacidade de generalização das técnicas de localização. Neste trabalho, como explicado anteriormente, a técnica de validação cruzada com $K = 10$ é usada para verificar a capacidade de generalização das técnicas de localização FP-AoA e FP-RSSI. Assim, podemos verificar o desempenho de cada abordagem de localização para diferentes conjuntos de dados. A Fig. 22 ilustra, em forma de uma matriz de gráficos, o erro de predição de posição η_j para ambas as técnicas de localização, considerando todos os ambientes ruidosos e não-ruidosos (*indoor*, *outdoor* e *indoor-outdoor*) e cenários (SCE-4, SCE -6 e SCE-9) para cada um dos dez *folds* da validação cruzada. A técnica FP-AoA é representada pelas linhas em cor azul, enquanto a técnica FP-RSSI pelas linhas em cor laranja. Em relação à presença/ausência de ruído, as linhas tracejadas representam ambientes ruidosos, enquanto as linhas sólidas representam

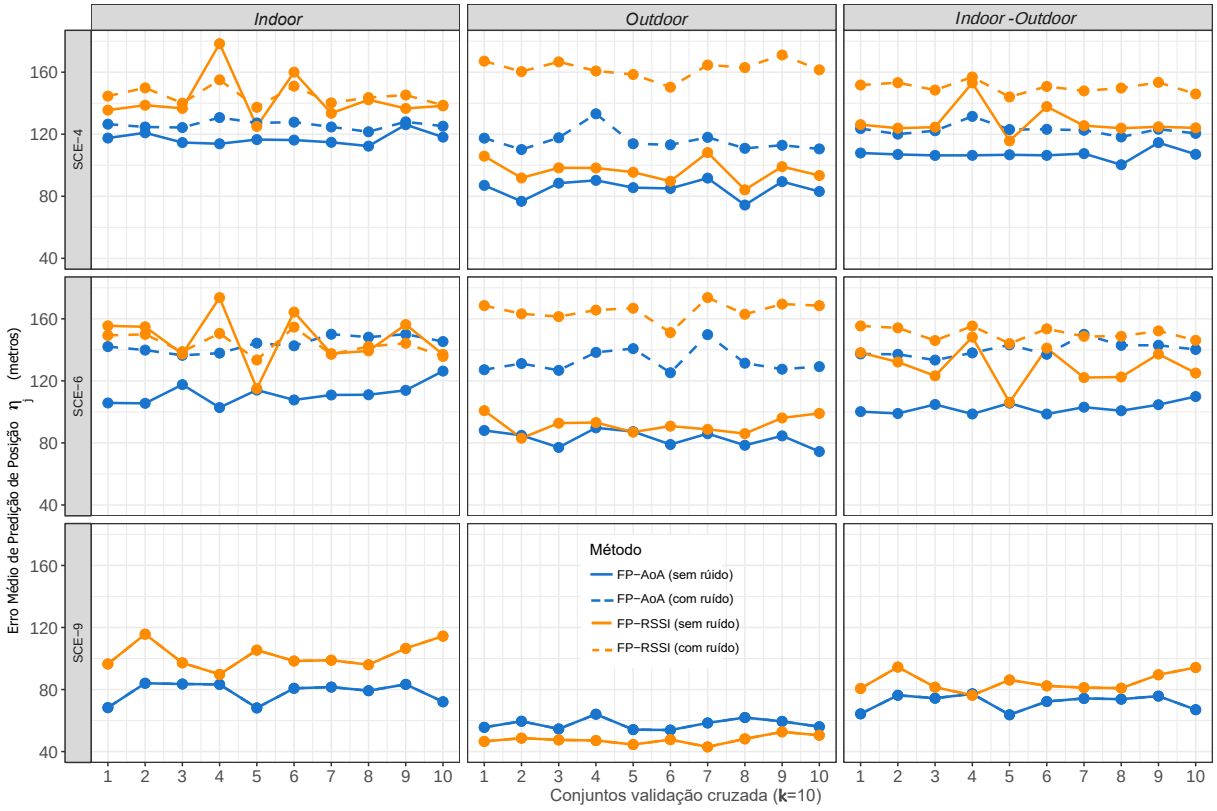

Figura 22 – Erro médio de predição de posição (em metros) para ambas as técnicas de localização, considerando todos os ambientes ruidosos e não-ruidosos (*indoor*, *outdoor* e *indoor-outdoor*) e cenários (SCE-4, SCE-6 e SCE-9) para cada *fold* da validação cruzada.

ambientes não-ruidosos. Dado um cenário, um ambiente e uma técnica de localização específica, a sensibilidade ao ruído representa a variação da acurácia devido à presença do ruído. Na Fig. 22, a sensibilidade ao ruído pode ser vista como a distância entre linhas sólidas e tracejadas correspondentes à mesma técnica. Quanto menor a distância entre as curvas, menos sensível ao ruído é a técnica de localização. Portanto, ambas as técnicas de localização são insensíveis ao ruído em todos os ambientes (*indoor*, *outdoor* e *indoor-outdoor*), como podemos observar na Fig. 22, para o cenário SCE-9 (terceira linha da matriz de gráficos), uma vez que as linhas sólidas e tracejadas estão sobrepostas.

Para o ambiente *outdoor* no cenário SCE-9, vemos que o erro de predição é menor para a técnica FP-RSSI tanto na presença quanto na ausência de ruído. Quando mudamos do cenário SCE-9 *outdoor* para outros cenários, percebemos que a técnica FP-AoA apresenta

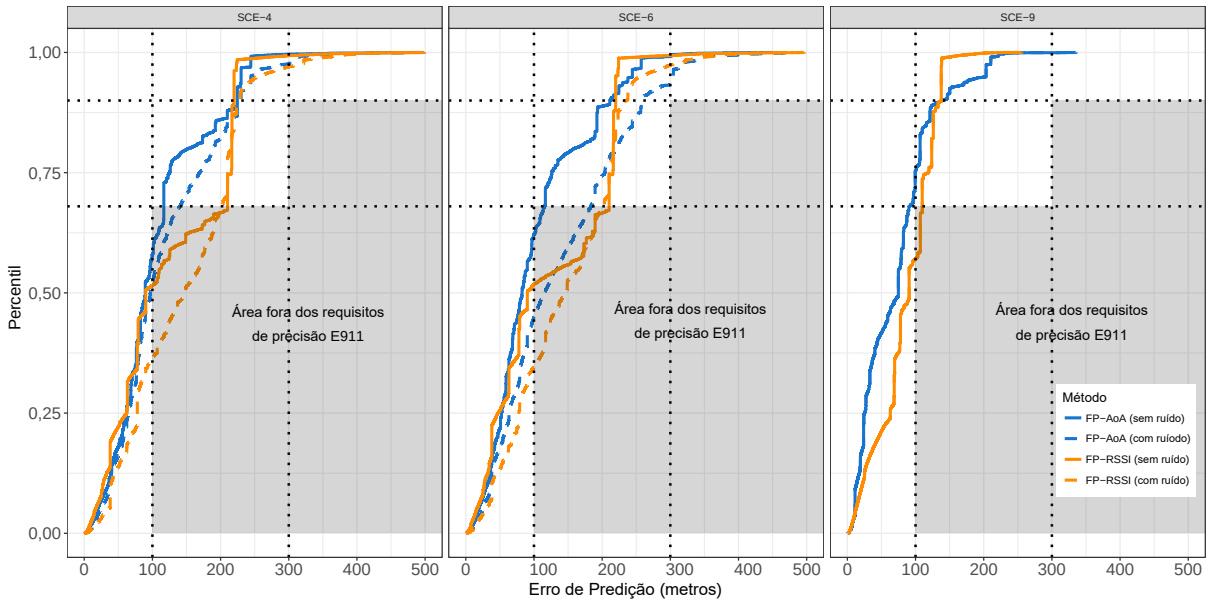

Figura 23 – Função de distribuição cumulativa do erro de predição de posição para ambas as técnicas de localização, considerando os ambientes *indoor-outdoor* ruidosos e não-ruidosos, para todos os cenários (SCE-4, SCE-6 e SCE-9).

uma acurácia maior, especialmente no cenário SCE-4 para ambientes ruidosos e não-ruidosos. Por sua vez, quando o usuário muda de ambiente (de *outdoor* para *indoor*), em geral, a técnica FP-AoA exibe um erro de predição menor que a técnica FP-RSSI, tanto para condições ruidosas como para não-ruidosas.

De acordo com a FCC (*Federal Communications Commission*), para sistemas de localização baseados em rede celular, qualquer EM que faça chamadas de emergência 911 (E-911) deve ser localizada com uma acurácia até $100\,m$ (67% dos casos) e até $300\,m$ para 90% dos casos (FCC, 2010). Dado que a Fig. 23 representa a função de distribuição cumulativa do erro de predição de posição para ambas as técnicas de localização, considerando os ambientes *indoor-outdoor* ruidosos e não-ruidosos para os cenários SCE-4 (lado esquerdo), SCE- 6 (centro) e SCE-9 (lado direito). As duas linhas tracejadas horizontais da Fig. 23 representam os percentis 67% e 90%, enquanto a região cinza em cada gráfico representa a área fora dos requisitos de acurácia de localização do E-911. É possível observar que apenas a técnica FP-AoA (linha em cor azul no caso do cenário SCE-9) supera

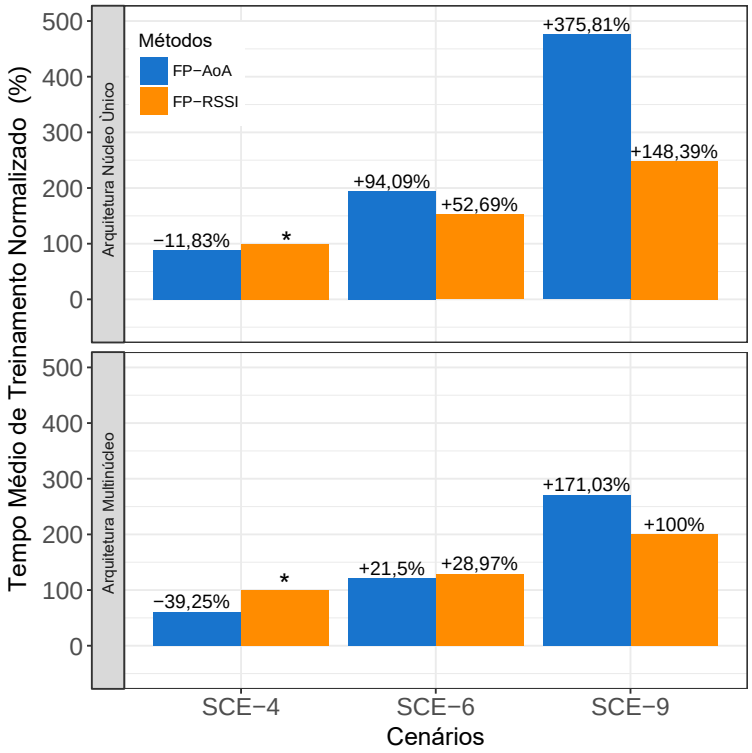

Figura 24 – Tempo médio de treinamento normalizado para técnicas de localização FP-AoA e FP-RSSI em ambientes *indoor-outdoor* não-ruidosos para todos os cenários. O método FP-RSSI para o cenário SCE-4 é assumido como referência.

as especificações de acurácia do E-911. Apesar disso, acreditamos que outros algoritmos de ML podem ser usados para melhorar o desempenho das duas técnicas de localização, ascendendo as especificações da (FCC, 2010).

Outro aspecto importante de qualquer sistema de localização é sua escalabilidade, especialmente com o avanço da tecnologia *Big Data* em redes celulares de próxima geração (LI et al., 2019b; SULTAN; ALI; ZHANG, 2018). Dado esse contexto, usamos o tempo de processamento gasto nos estágios de treinamento e predição do modelo para analisar a escalabilidade das técnicas. A Fig. 24 mostra o tempo médio de treinamento normalizado para as técnicas FP-AoA e FP-RSSI em todos os cenários *indoor-outdoor* não-ruidosos, usando arquiteturas computacionais com processamento de núcleo único (*single-core*) e multinúcleo (*multi-core*). Tais arquiteturas são examinadas para verificar a capacidade de paralelismo de cada técnica de localização. Assumimos a técnica FP-RSSI para o cenário

SCE-4 como *benchmark*, indicado na Fig. 24 pelas barras com asterisco ($*$). Como os tempos gastos na predição foram quase os mesmos para as duas técnicas de localização (uma variação menor que 2%), concentramos nossa análise apenas nos tempos de treinamento.

Para o cenário SCE-4, observa-se que a técnica FP-AoA possui um tempo de treinamento 11,83% menor que o valor de referência para uma arquitetura de núcleo único. Quando uma arquitetura multinúcleo é assumida, a redução do tempo de treinamento atinge 39,25%. Esse fato combinado à melhoria de desempenho (menor previsão de erro) oferece vantagens relevantes à técnica FP-AoA, quando aplicado em redes celulares menos densas, tais como as redes rurais e suburbanas.

Com relação à arquitetura de núcleo único, podemos ver que o tempo de treinamento para a técnica FP-AoA aumenta exponencialmente quando passamos do cenário SCE-4 para o SCE-9. Esse comportamento é explicado pelo aumento no número de atributos (*features*) que afetam o tempo de execução do regressor k-NN. Ao mesmo tempo, quando o cenário SCE-9 é assumido, um novo grupo de ERBs é incluído, o que faz com que o método FP-AoA calcule um ângulo horizontal adicional para realizar a previsão da localização. Em relação à técnica FP-RSSI, o aumento no tempo de treinamento é menor do que o aumento na técnica FP-AoA. Nesse caso, diferente da técnica FP-AoA, a taxa de crescimento do tempo de treinamento está ligada somente ao aumento do número de atributos.

Na arquitetura multinúcleo (parte inferior da Fig. 24) e considerando a técnica FP-AoA, percebe-se que a taxa de crescimento do tempo de treinamento devido à adição de ERBs é mais suave quando comparada à arquitetura com um único núcleo de processamento. A principal razão para esse comportamento é que o *grid* de localização, mostrado na Fig. 14, pode ser construído independentemente dos modelos de regressores para a técnica FP-AoA. Dessa forma, é possível paralelizar as tarefas de treinamento do modelo de ML e a construção do *grid* de localização, o que favorece a escalabilidade para aplicação

em redes de próxima geração fortemente dependentes de uma grande quantidade de dados. Além disso, o paralelismo pode ser uma grande vantagem para a aplicação de aprendizado automático de máquina (Auto-ML, *auto machine learning*) (QUANMING et al., 2018). Os *frameworks* de Auto-ML podem procurar os melhores modelos de ML automaticamente, sem recriar o *grid* de localização para cada modelo testado.

Finalmente, para verificar se as diferenças entre a acurácia das técnicas de localização são estatisticamente relevantes, é aplicado o teste de *rank* não paramétrico de Kruskal-Wallis, que estende o teste de Wilcoxon para mais de duas séries (VERZANI, 2018). A hipótese nula H_0 é aquela em que todos os métodos são equivalentes e a hipótese alternativa H_1 é aquela em que há alguma diferença entre as técnicas. Para o teste de Kruskal-Wallis realizado em todos os conjuntos de dados com um nível de confiança $\alpha = 0,05$, o valor-p obtido foi de $2,2 \cdot 10^{-16}$. Assim, como $p << \alpha$, a hipótese H_0 pode ser rejeitada e pelo menos duas abordagens diferem.

A partir dos resultados do teste de Kruskal-Wallis, sabemos que existem diferenças entre as técnicas de localização, mas não é possível afirmar quais pares de técnicas são diferentes. Nesse caso, podemos usar o teste da soma de *rank* de Wilcoxon com correções para testes múltiplos com $\alpha = 0,05$ para fazer comparações pareadas (BENJAMINI; HOCHBERG, 1995). Foi obtido um valor-p ajustado menor que α para todos os pares, exceto para o par FP-RSSI SCE-4 e FP-RSSI SCE-6, com um valor-p ajustado igual a $0,535$. Esse valor-p indica que as técnicas FP-RSSI mencionadas anteriormente são estatisticamente equivalentes.

5.4 APLICAÇÃO DO MÉTODO DYSEREK EM FP ANGULAR

Nesta Seção, o método DySEREK, apresentado no Capítulo 4, é utilizado como regressor na implementação da técnica de localização baseada em FP angular. O algoritmo k-NN é utilizado como regressor base do método DySEREK e os regressores são ajustados (*tuning*) por meio de validação cruzada com $K = 10$ (*10-cross-validation*) e o melhor k

do algoritmo k-NN sendo procurado no intervalo $[1, 7]$.

Além do ajuste do algoritmo k-NN, é necessário também realizar uma busca para encontrar os melhores parâmetros para o método DySEREK. Como visto no Capítulo 4, há um grande número de combinações possíveis. Assim, os resultados apresentados na Seção 4.3 serão utilizados para restringir a escolha dos parâmetros do método DySEREK. Em particular, o *dataset AirQuality* pode ser utilizado como paradigma, pois, considerando os tipos de atributos e a quantidade de instâncias, é o *dataset* que possui maior similaridade com o conjunto de dados utilizado para predição do ângulo horizontal do usuário móvel em relação às ERBs. Nesse caso, o *pool* será formado por 128 regressores, pois foi a configuração que apresentou melhor desempenho no *dataset AirQuality*. Além disso, em média, os melhores resultados foram obtidos nas variantes do método DySEREK que utilizaram pesos (Dy-EW e Dy-UW). Assim, nesse contexto, a Tab. 14 mostra os parâmetros e respectivos valores utilizados no ajuste do método DySEREK para o problema de localização de usuários em redes sem fio.

Para diferenciar as duas técnicas de localização baseadas em FP angular, iremos denotar a que utiliza um conjunto de regressores baseado no método DySEREK de FP-AoA-Dy. A segunda técnica é aquela que foi apresentada na Seção 5.3, a qual utiliza um único regressor k-NN (FP-AoA). Em relação aos cenários, são utilizados os cenários SCE-4, SCE-6 e SCE-9, mesmos cenários apresentados na Seção 5.3, com quatro, seis e nove ERBs, respectivamente. No que diz respeito ao *dataset* de medições de RSSI e PD,

Tabela 14 – Parâmetros utilizados para ajuste (*tuning*) do método DySEREK aplicado à localização de usuários móveis.

Parâmetro DySEREK	Valores utilizados
Tamanho do *pool* (regressores)	128
Variantes do método	Dy-EW, Dy-UW
Hiper-k	5, 7, 9
Tamanho do conjunto de validação	10%, 20%, 30%

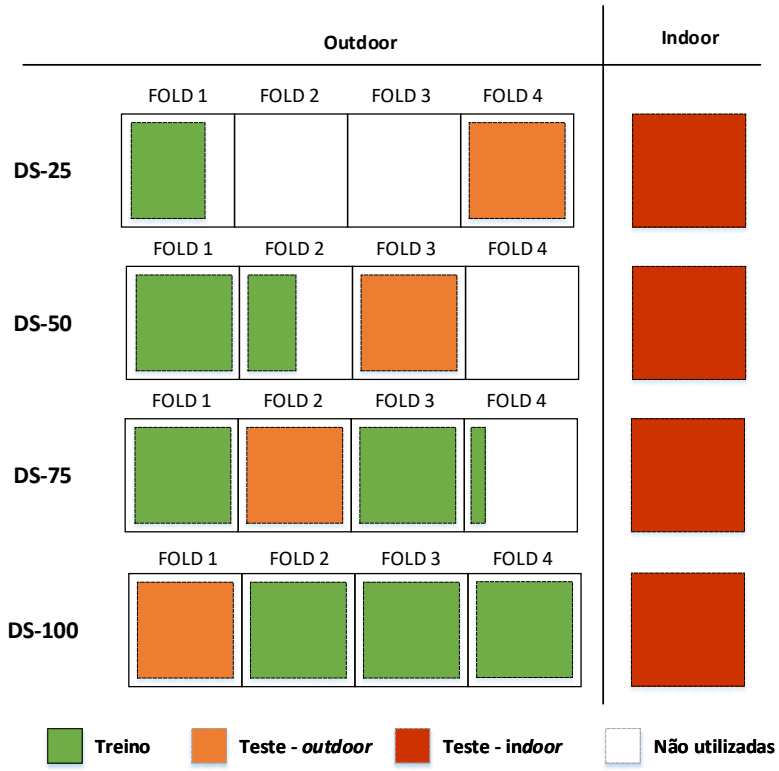

Figura 25 – Diagrama ilustrativo da formação dos *datasets* DS-25, DS-50, DS-75 e DS-100, considerando os *folds* da validação cruzada ($K = 4$) nas medições *outdoor*.

semelhante à técnica FP-AoA, temos 6.015 medições *outdoor* e 3.064 medições *indoor*. Ainda semelhante à técnica FP-AoA, as medições *indoor* são utilizadas somente como instâncias de teste, ou seja, não são utilizadas para o treinamento dos regressores.

Para testar a capacidade de generalização das técnicas, foram gerados quatro *folds* utilizando validação cruzada com $K = 4$ a partir das medições *outdoor*. A partir desses *folds*, foram criados quatro *datasets*. A Fig. 25 ilustra a formação dos *datasets* em função dos *folds* gerados. Os *datasets* são nomeados como DS-X, em que X indica o percentual das instâncias do conjunto de treino utilizadas para treinar os regressores. Por exemplo, no DS-25, o *fold* 4 é utilizado como conjunto de teste-*outdoor* (cor laranja) e 25% das instâncias dos *folds* 1, 2 e 3 (cor verde) são utilizadas como instâncias de treino. Para facilitar o entendimento, os 25% das instâncias de treino são representados dentro do *fold* 1, mas na implementação essas instâncias foram escolhidas de forma aleatória, considerando os *folds* 1, 2 e 3. Vale ressaltar que as medições (instâncias) *indoor* são sempre utilizadas

como instâncias de teste em todos os *datasets* e formam o conjunto teste-*indoor* representado pela cor vermelha na Fig. 25. Para o DS-50, o *fold* 3 é utilizado como teste-*outdoor* e 50% dos *folds* 1, 2 e 4 são usados para treinar os regressores. O processo de geração dos conjuntos de treino continua para os *datasets* DS-75 e DS-100, conforme pode ser visto na Fig. 25. O objetivo dessa variação no tamanho do conjunto de treino é verificar o desempenho das técnicas quando poucas medições estiverem disponíveis. Dessa forma, com exceção do DS-100, nem todas as instâncias reservadas para treino serão utilizadas efetivamente. Para exemplificar um caso prático que caracteriza a obtenção de poucas medições de campo, considere a situação de uma operadora de telecomunicações que necessita realizar o mapeamento de uma região urbana muito densa, na qual há muitas áreas de difícil acesso. Uma vez que o levantamento completo de medições de sinais referente a todas as ERB da região é extremamente custoso (tempo e recursos humanos), a operadora acaba por obter uma quantidade menor de medições para a criação de seu modelo de predição.

A Fig. 26 complementa o entendimento do processo de formação dos *datasets*, ilustrando a variação das quantidades de instâncias (medições) utilizadas para treino e teste em cada *fold*. No eixo das abscissas, temos o *dataset*, enquanto no eixo das ordenadas, temos a quantidade de instâncias divididas em três grupos, a saber: treino, teste-*indoor* e teste-*outdoor*. Ainda na Fig. 26, é possível visualizar a variação da proporção entre os conjuntos de treino (cor verde), teste-*indoor* (cor vermelha) e teste-*outdoor* (cor laranja). Por exemplo, no *dataset* DS-25, o menor conjunto é o de treino ($1,3$ mil instâncias), mas acompanhando o fluxo representado pela cor verde, percebe-se que o conjunto de treino aumenta até se tornar o maior dos três conjuntos no *dataset* DS-100, com $5,2$ mil instâncias.

A Fig. 27 ilustra uma matriz de gráficos com o erro médio de predição de posição, dado em metros, para cada *dataset*, em todos os cenários (SCE-4, SCE-6 e SCE-9) e consi-

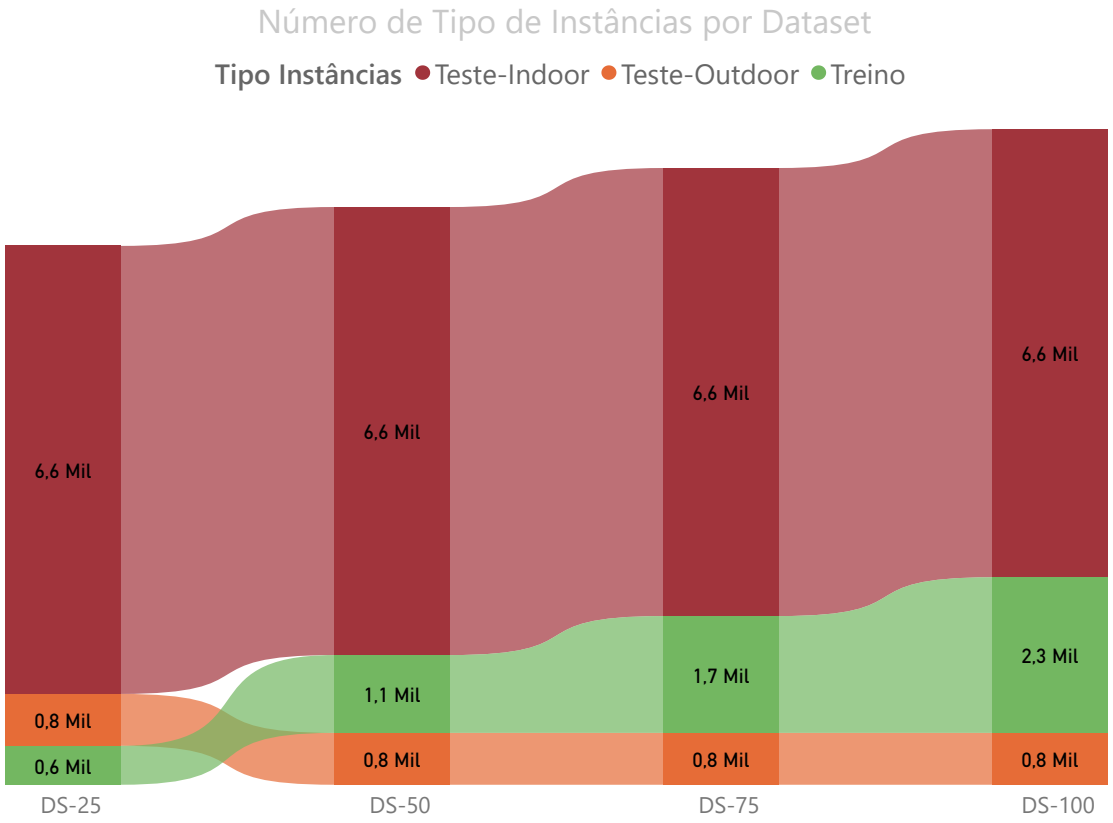

Figura 26 – Variação da quantidade de instâncias de treino, teste-*indoor* e teste-*outdoor* para os *datasets* DS-25, DS-50, DS-75 e DS-100.

derando os ambientes *indoor*, *outdoor* e *indoor-outdoor*. Todos ao ambientes são ruidosos, com ruído tendo a mesma distribuição definida na Seção 5.3.1. Cada linha da matriz representa um cenário, enquanto cada coluna está relacionada a um ambiente. Logo, cada posição da matriz representa um determinado cenário e um ambiente específico, com o *dataset* indicado no eixo das abscissas e o erro médio de predição no eixo das ordenadas. Por último, os resultados da técnica FP-AoA estão representados pelas linhas em cor azul, enquanto os relativos à técnica FP-AoA-Dy são representados pelas linhas em cor vermelha.

Observando a Fig. 27, podemos ver que para o ambiente *indoor* no cenário SCE-4, a técnica FP-AoA-Dy supera a técnica FP-AoA em todos os *datasets*, com exceção do DS-100. Nesse contexto (SCE-4 *indoor*), a maior diferença ocorreu no DS-50, onde a técnica FP-AoA-Dy teve um erro médio de posição de $131,99\,m$, enquanto a técnica FP-AoA apresentou um erro médio de $146,15\,m$. É interessante notar que quando todo conjunto

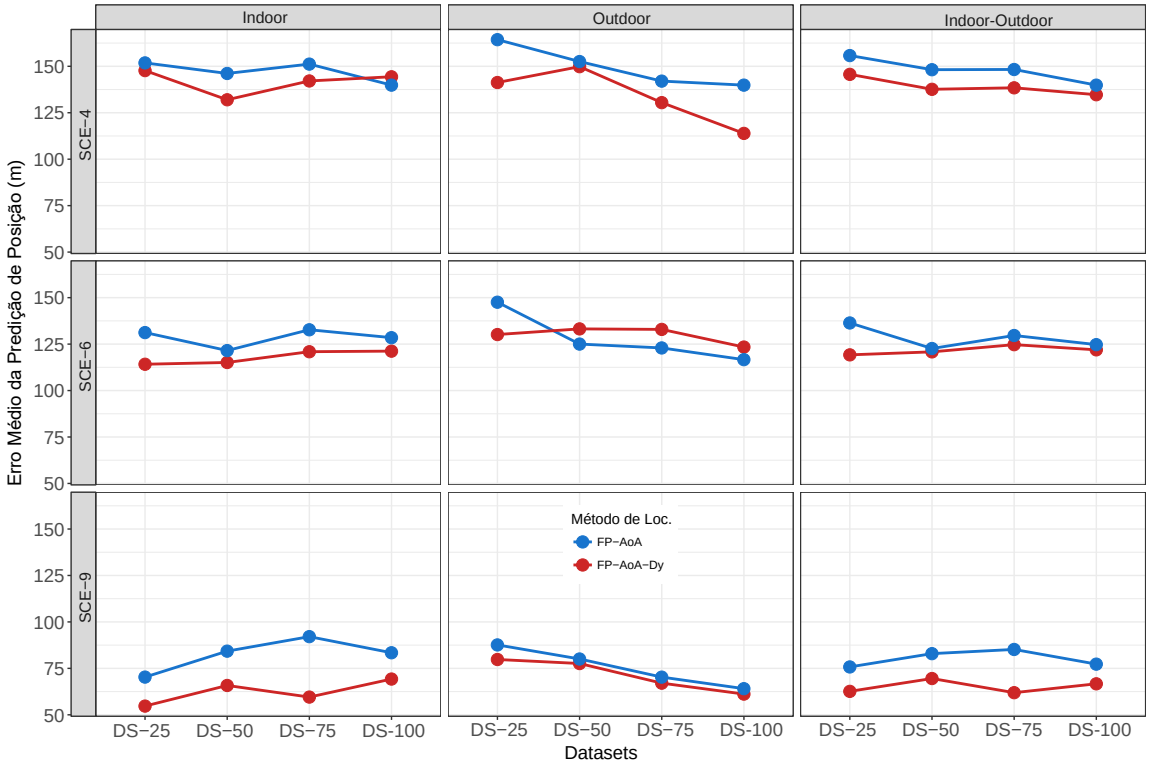

Figura 27 – Erro médio de predição de posição (em metros) para as técnicas FP-AoA e FP-AoA-Dy, considerando todos os cenários, ambientes e os *datasets* DS-25, DS-50, DS-75 e DS-100.

de treino é utilizado (DS-100), a técnica FP-AoA supera a técnica FP-AoA-Dy. Nesse caso, o primeiro teve um erro médio de $139,88\,m$, enquanto o último teve um erro de $144,36\,m$.

Ainda no ambiente *indoor*, mas agora considerando o cenário SCE-6, como pode ser visto na Fig. 27, a técnica FP-AoA-Dy teve melhor desempenho que a técnica FP-AoA em todos os *datasets*, sendo que a maior diferença ocorreu no DS-25. Nesse caso, o erro médio de posição do FP-AoA-Dy foi de $114,17\,m$, sendo $17,06\,m$ a menos que o erro médio obtido pelo FP-AoA, que foi de $131,23\,m$.

Quando nove ERBs foram utilizadas no ambiente *indoor*, a diferença de desempenho entre as técnicas FP-AoA-Dy e FP-AoA aumentou quando comparada aos cenários SCE-4 e SCE-6, como pode ser visto pelo maior espaçamento entre as linhas na Fig. 27 (terceira linha e primeira coluna da matriz de gráficos). Nesse contexto, a técnica FP-AoA-Dy obteve o menor erro médio de posição, no valor de $54,71\,m$, no *dataset* DS-25, enquanto o

Cenário	Dataset	Ambiente: Indoor	Outdoor	Indoor-Outdoor
SCE-4	DS-25	2,72%	14,03%	6,50%
	DS-50	9,69%	1,80%	7,12%
	DS-75	6,00%	8,15%	6,65%
	DS-100	-3,21%	18,57%	3,68%
SCE-6	DS-25	13,00%	11,80%	12,59%
	DS-50	5,29%	-6,54%	1,48%
	DS-75	8,90%	-8,12%	3,80%
	DS-100	5,64%	-5,81%	2,25%
SCE-9	DS-25	22,19%	8,95%	17,35%
	DS-50	21,91%	3,05%	16,16%
	DS-75	35,33%	4,62%	27,32%
	DS-100	16,96%	4,56%	13,71%

Ganho FP-AoA-Dy(%) -8,83% 35,33%

Figura 28 – Mapa de calor com a variação percentual do erro médio de predição de posição da técnica FP-AoA-Dy em relação à técnica FP-AoA, para todos os cenários (SCE-4, SCE-6 e SCE-9) e ambientes (*indoor*, *outdoor* e *indoor-outdoor*) estudados, bem como para os *datasets* considerados (DS-25, DS-50, DS-75 e DS-100).

FP-AoA teve um erro médio de $70,30\,m$. No entanto, vale destacar que a maior diferença entres os erros médios de posição ocorreu no *dataset* DS-75, com erros médios de posição de $59,55\,m$ e $92,08\,m$ obtidos respectivamente pelas técnicas FP-AoA-Dy e FP-AoA.

Considerando o ambiente *outdoor*, em geral a diferença de desempenho entre as técnicas diminuiu, como pode ser visto pelo menor espaçamento entre as linhas na Fig. 27. No cenário SCE-4, a técnica FP-AoA-Dy teve um menor erro médio de posição em todos os *datasets*. O mesmo resultado se repetiu no cenário com nove ERBs. No entanto, quando seis ERBs (SCE-6) são utilizadas, a técnica FP-AoA-Dy só teve melhor desempenho no *dataset* DS-25, sendo superada pela técnica FP-AoA nos outros três *datasets* (DS-25, DS-75 e DS-100).

Por último, na Fig. 27 temos o erro médio de posição considerando o ambiente *indoor-outdoor*. Nesse caso, a técnica FP-AoA-Dy superou a técnica FP-AoA em todos os cenários (SCE-4, SCE-6 e SCE-9) e em todos os *datasets* (DS-25, DS-50, DS-75 e DS-100). No entanto, vale ressaltar que no cenário SCE-9, a diferença de desempenho entre as técnicas

é maior para todos os *datasets* quando comparada às diferenças obtidas nos cenários SCE-4 e SCE-6.

A Fig. 28 ilustra um mapa de calor referente à variação percentual do erro médio de predição de posição da técnica FP-AoA-Dy em relação à técnica FP-AoA, obtida para cada *dataset* em todos os cenários e ambientes. Valores percentuais positivos indicam que a técnica FP-AoA-Dy apresentou melhor acurácia, ou seja, um erro de predição menor. Por outro lado, valores percentuais negativos mostram que a técnica FP-AoA obteve melhor desempenho. É possível verificar que os maiores ganhos percentuais da técnica FP-AoA-Dy foram obtidos no cenário SCE-9 e ambiente *indoor* (posições do mapa de calor em cor azul mais escura). Nesse contexto (SCE-9 *indoor*), o maior ganho, no valor de $35,33\%$, ocorreu para o *dataset* DS-75. Os ganhos percentuais menores, inclusive com valores negativos (perdas), foram observados no ambiente *outdoor* e cenário SCE-6 (posições do mapa de calor em cor laranja). Nesse caso, a maior perda foi de $8,12\%$ no *dataset* DS-75. Vale destacar que, fora do cenário SCE-6, como pode ser visto na Fig. 28, só houve outra perda, ocorrida no cenário SCE-4 (*indoor*) no valor de $3,21\%$.

Outro importante aspecto é analisar o desempenho das técnicas considerando as medições disponíveis para treinamento dos regressores utilizados. Assim, a Fig. 29 ilustra o ganho percentual médio da técnica FP-AoA-Dy em relação à técnica FP-AoA, por configuração de *dataset* e tipo de ambiente. O ambiente *indoor* é representado pela cor laranja, enquanto os ambientes *outdoor* e *indoor-outdoor* são representados, respectivamente, pela cores verde e azul. O eixo das abscissas contém as possíveis configurações de *datasets*, enquanto o eixo das ordenadas representam o ganho percentual médio. Vale destacar que no eixo das abscissas, os *datasets* estão dispostos, da esquerda para a direita, em ordem decrescente de ganho total (altura total da barra vertical) nos três ambientes. Logo, o *dataset* DS-25 está posicionado mais à esquerda, uma vez que apresentou a maior soma dos ganhos. Em seguida, tivemos o DS-75, seguido do DS-50 e, por último, o DS-100.

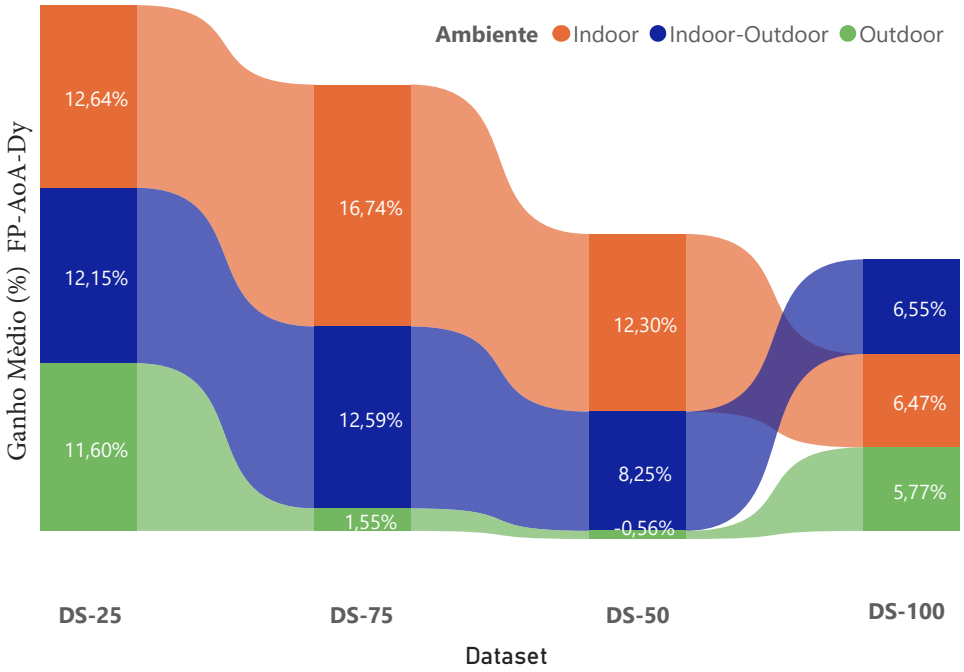

Figura 29 – Ganho médio (em %) da técnica FP-AoA-Dy em relação à técnica FP-AoA, para os diferentes *datasets* e tipos de ambiente (*indoor, outdoor e indoor-outdoor*).

Na Fig. 29, é possível notar que os maiores ganhos da técnica FP-AoA-Dy ocorrem no DS-25, *dataset* com menor número de instâncias de treino. Acompanhando o fluxo à direita, observa-se que há uma diminuição do ganho médio quando o tamanho do conjunto de treinamento cresce, embora haja uma inversão das posições dos *datasets* DS-50 e DS-75. Essa troca de posição ocorreu em virtude do alto ganho obtido pela técnica FP-AoA-Dy no cenário SCE-9 *indoor* e *dataset* DS-75, cujo valor foi de $35,22\%$, conforme consta no mapa de calor (*vide* Fig. 28). Percebe-se que os menores ganhos ocorreram no ambiente *outdoor* (região em cor verde), inclusive com perda de $-0,56\%$ para o *dataset* DS-50. Este fato é justificado pelas variações percentuais apresentadas na Fig. 28, na qual a técnica FP-AoA-Dy apresentou os menores ganhos nos cenários SCE-4 e SCE-9 ($1,80\%$ e $3,05\%$, respectivamente), além de ter tido a segunda pior perda no cenário SCE-6 ($-6,54\%$). Por outro lado, no ambiente *indoor* (região em cor laranja), a técnica FP-AoA-Dy apresentou, em geral, ganhos médios mais elevados em relação à técnica FP-AoA, chegando a $16,74\%$ no DS-75. De fato, a análise conjunta das Figs. 28 e 29 permite

identificar que a técnica FP-AoA-Dy, para o DS-75, obteve ganhos expressivos em todos os cenários considerados (SCE-4: 6%, SCE-6: $8,9\%$ e SCE-9: $35,33\%$), justificando o ganho médio de $16,74\%$. No entanto, o ganho médio da técnica FP-AoA-Dy se mostrou mais uniforme nos três ambientes quando os dados para treinamento foram menores (DS-25), reforçando a condição de que o método DySEREK, baseado em combinação dinâmica de regressores, se mostra mais eficiente quando o conjunto de treinamento possui tamanho menor.

Finalmente, para verificar se as diferenças entre os resultados obtidos pelas técnicas são estatisticamente relevantes podemos utilizar o teste pareado para duas amostras de Wilcoxon (DEMSAR, 2006). O teste foi aplicado considerando nível de confiança $\alpha = 0,05$ e de forma pareada para cada cenário (SCE-4, SCE-6 e SCE-9) e *dataset*. Nos testes, foram obtido um valor-p abaixo de $0,05$ para todos os *datasets* nos cenários SCE-4 e SCE-9. No cenário SCE-6 tivemos um valor-p maior que $0,05$ nos *datasets* DS-25, DS-50 e DS-75, o que indica que não há diferenças estatisticamente relevantes para o cenário SCE-6 para esses *datasets*. No cenário SCE-6, os valores-p obtidos foram $0,062$, $0,5106$, 0.9913 para os *datasets* DS-25, DS-50 e DS-75.

5.5 RESUMO DO CAPÍTULO

Nesse capítulo, uma técnica inovadora de localização baseada em *fingerprinting*, usando ângulos horizontais obtidos a partir dos sinais de RF, foi proposta para estimar a posição da EM. A principal motivação para essa proposição foi a introdução de um método de localização projetado para ambientes *outdoor*, mas que pode ser aplicado para prever a posição do usuário em um ambiente *indoor*. Diferente da abordagem tradicional de *fingerprinting*, que é baseada nos valores absolutos RSSI, nossa proposta utiliza diferenças de RSSI e PD para obter o ângulo horizontal do usuário móvel. A abordagem tradicional de *fingerprinting* também foi implementada para comparação com a proposta apresentada. Em ambos os métodos, medições de propagação de ondas de RF em uma frequência

de $1,8$ GHz W-CDMA foram obtidas em uma região urbana na cidade de Recife-PE, Brasil. Resultados numéricos mostraram que nossa proposta apresentou um erro médio de predição de posição menor para todos os ambientes e cenários investigados, exceto em ambientes *outdoor* ruidosos e não-ruidosos para um cenário com nove ERBs. Além disso, o método proposto foi menos sensível à mudança de ambiente (de *outdoor* para *indoor*) e mais estável quando aplicado em ambientes *indoor* ou *outdoor* em cenários com menos ERBs. Constatou-se que o método de *fingerprinting* baseado em AoA proposto permitiu a paralelização do treinamento dos algoritmos de ML e da construção do *grid* de localização, o que pode ser interessante para promover a escalabilidade em redes ultra-densas, como sistemas celulares 5G. Finalmente, o método DySEREK foi utilizado como regressor para a técnica FP-AoA proposta nesse capítulo. Verificou-se que a utilização do método DySEREK diminuiu significativamente o erro médio de posição para os cenários com quatro e seis ERBs. Nesses cenários, os maiores ganhos com a aplicação do DySEREK foram obtidos quando conjuntos de treinamento menores foram utilizados para treinar os regressores, situação que representa o caso em que poucas medições de campo foram realizadas.

6 CONCLUSÕES

O estudo realizado nesta Tese de Doutorado teve caráter multidisciplinar, abrangendo as áreas de Aprendizado de Máquina e Telecomunicações, e consistiu de três partes. Na primeira parte, visando fornecer contribuições na área de Aprendizado de Máquina, foi proposto um novo método de seleção dinâmica de regressores, denominado DySEREK e baseado no método KNORA. Na segunda parte, com o intuito de contribuir na área de Telecomunicações, em particular no desenvolvimento de técnicas de localização de usuários em redes sem fio, foi sugerida uma nova técnica de localização baseada em *fingerprinting* usando ângulos horizontais preditos a partir de diferenças de intensidade de nível de sinal e parâmetros de atraso de onda das redes celulares. Por fim, na terceira e última parte desta Tese, foi promovida a multidisciplinaridade das duas áreas do conhecimento previamente citadas, com a aplicação do método DySEREK na técnica de localização proposta em busca de aprimoramentos quanto à substituição do regressor simples por um sistema de múltiplos regressores.

Na primeira parte do trabalho, a motivação para a adaptação do método KNORA para regressão foi o seu bom desempenho na tarefa de classificação. Nesse contexto, como a diversidade é ponto-chave no desempenho de sistemas com múltiplos preditores, foi proposta uma técnica de criação do *pool* de regressores, denominada CBAS, baseada na combinação dos algoritmos *Bagging*, *Random Subspace* e *Boosting*, para ser incorporada à seleção dinâmica realizada pelo método DySEREK. No que diz respeito à técnica CBAS, os resultados mostraram que a proposta venceu as abordagens individuais de geração de *pool* em 80% dos *datasets* considerados no estudo. Ademais, em relação ao RMSE, a técnica CBAS teve desempenho equivalente quando comparada à abordagem com múltiplos *pools* proposta em (KOTSIANTIS; KANELLOPOULOS, 2012).

Ainda na primeira parte desta Tese, a técnica CBAS foi utilizada como gerador de

pool de regressores na concepção do método DySEREK. Na etapa de seleção, o método DySEREK empregou o oráculo para a escolha dos melhores regressores, porém, diferentemente do método KNORA, o DySEREK utilizou a mediana dos erros para caracterizar as predições como boas ou ruins. Semelhante ao KNORA, foram propostas quatro variantes do método DySEREK segundo a estratégia de seleção dos regressores, quais sejam, Dy-E, Dy-EW, Dy-U e Dy-UW. Experimentos realizados em 14 *datasets* públicos com diferentes tamanhos de *pool* mostraram que o DySEREK teve melhor desempenho em 70 a 86% dos *datasets* quando comparado com três algoritmos de referência, a saber: regressor único, *pool* e DESIP. Também foi verificada a influência dos parâmetros do método DySEREK em seu desempenho. As variantes do grupo *Union* (Dy-U e Dy-UW) tiveram melhores resultados em cerca de 83% dos *datasets*, usando *pools* menores. No entanto, as variantes do grupo *Elimination* (Dy-E e Dy-EW) apresentaram tempos de predição menores, além de uma tendência de melhores desempenhos com o crescimento do *pool*, o que precisa ser melhor explorado.

Na segunda parte da Tese, foi sugerida uma nova técnica de localização baseada em *fingerprinting* usando ângulos horizontais preditos a partir de diferenças de RSSI e PD. A principal motivação para a proposta foi a busca por uma solução projetada para ambientes *outdoor*, mas que pudesse ser utilizada para prever a posição do usuário em um ambiente *indoor*. Os resultados obtidos mostraram que a técnica de localização proposta teve um erro médio de predição de posição menor em cerca de 88% dos cenários investigados, quando comparada à abordagem tradicional de *fingerprinting* que utiliza o valor absoluto de RSSI. Além disso, a técnica elucidada foi menos sensível à mudança de ambiente (de *outdoor* para *indoor*) e mais estável quando aplicada em ambientes *indoor* ou *outdoor*, considerando cenários com um menor número de ERBs. Outra importante constatação foi que a nova abordagem permitiu a paralelização do treinamento dos algoritmos de ML e da construção do *grid* de localização, o que é interessante para promover a escalabilidade

em redes com um alto volume de dados.

Na terceira e última parte desta Tese, o método DySEREK foi utilizado como regressor integrante da técnica de localização proposta na segunda parte do trabalho. Verificou-se que a utilização do DySEREK diminuiu significativamente o erro médio de posição para os cenários com quatro e seis ERBs. Nesses cenários, os maiores ganhos com a aplicação do método DySEREK foram obtidos quando menores conjuntos de dados (poucas medições de campo) foram utilizados para treinar os regressores, situação típica à qual operadoras de telecomunicações estão sujeitas na ocasião do mapeamento de sinais de RF em regiões densamente urbanas.

Por fim, sugerimos como perspectiva de trabalhos futuros, a investigação de como novas técnicas de Auto-ML (QUANMING et al., 2018) podem ser utilizadas com o objetivo de encontrar os melhores parâmetros para o método DySEREK. Adicionalmente, em relação à técnica de localização proposta, recomendamos a utilização de simuladores de redes celulares, tais como o OMNeT++ (VIRDIS, 2019), para verificar como a técnica de localização se comporta em relação a mudanças realizadas no ambiente urbano, com o propósito de verificar a necessidade e a periodicidade de calibração do sistema de localização (retreinamento dos modelos).

REFERÊNCIAS

ABDULLAH, O.; ABDEL-QADER, I.; BAZUIN, B. A probability neural network-jensen-shannon divergence for a fingerprint based localization. In: IEEE. *Proc. of the 2016 Annual Conference on Information Science and Systems (CISS)*. [S.l.], 2016. p. 286–291.

ABU-MOSTAFA, Y. S.; MAGDON-ISMAIL, M.; LIN, H.-T. *Learning from data*. [S.l.]: AMLBook New York, NY-USA, 2012.

AHMED, M.; IMTIAZ, M. T.; KHAN, R. Movie recommendation system using clustering and pattern recognition network. In: IEEE. *Proc. of the 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*. [S.l.], 2018. p. 143–147.

ANDERSEN, S. Y. Y. B.; RAPPAPORT, T. S. Propagation measurements and models for wireless communications channels. *IEEE Communications Magazine*, v. 33, n. 1, p. 42–49, 1995.

AVNIMELECH, R.; INTRATOR, N. Boosting regression estimators. *Neural Computation*, v. 11, n. 2, p. 499–520, 1999.

BAKKER, B.; HESKES, T. Clustering ensembles of neural network models. *Neural Networks*, v. 16, n. 2, p. 261–269, 2003. ISSN 08936080.

BALLARD, C.; WANG, W. Dynamic ensemble selection methods for heterogeneous data mining. In: IEEE. *2016 12th World Congress on Intelligent Control and Automation (WCICA)*. [S.l.], 2016. p. 1021–1026.

BALSHI, M. S.; MCGUIRE, A. D.; DUFFY, P.; FLANNIGAN, M.; WALSH, J.; MELILLO, J. Assessing the response of area burned to changing climate in western boreal north america using a multivariate adaptive regression splines (mars) approach. *Global Change Biology*, Wiley Online Library, v. 15, n. 3, p. 578–600, 2009.

BENIKOVSKY, J.; BRIDA, P.; MACHAJ, J. RF fingerprinting location techniques. In: ***Handbook of Position Location: Theory, Practice, and Advances***. [S.l.: s.n.], 2011. p. 487–520.

BENJAMINI, Y.; HOCHBERG, Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, Wiley Online Library, v. 57, n. 1, p. 289–300, 1995.

BITTENCOURT, G. P.; URBANO, A. A.; CUNHA, D. C. A proposal of an rf fingerprint-based outdoor localization technique using irregular grid maps. In: IEEE. *Proc. of the 2018 IEEE Wireless Communications and Networking Conference (WCNC)*. [S.l.], 2018. p. 1–6.

BORKOWSKI, J.; LEMPIAINEN, J. Practical network-based techniques for mobile positioning in UMTS. *EURASIP J. Adv. Signal Processing*, EURASIP, v. 2006, n. 012930, 2006.

BREIMAN, L. Bagging predictors. *Machine Learning*, v. 24, n. 2, p. 123–140, 1996. ISSN 0885-6125. Disponível em: <http://link.springer.com/10.1007/BF00058655>.

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

BRITTO, A. S.; SABOURIN, R.; OLIVEIRA, L. E. S. Dynamic selection of classifiers - a comprehensive review. *Pattern Recognition*, v. 47, n. 11, p. 3665–3680, 2014. ISSN 00313203.

BROWN, G. *Diversity in neural network ensembles*. Tese (Doutorado) — University of Birmingham, UK, 2004.

BROWN, G.; WYATT, J.; HARRIS, R.; YAO, X. Diversity creation methods: a survey and categorisation. *Information Fusion*, v. 6, n. 1, p. 5–20, 2005. ISSN 15662535.

BROWN, G.; WYATT, J.; TINO, P. Managing diversity in regression ensembles. *Journal of Machine Learning Research*, v. 6, p. 1621–1650, 2005. ISSN 1532-4435. Disponível em: <http://eprints.pascal-network.org/archive/00004768/>.

CAMPOS, R. S.; LOVISOLO, L. RF fingerprinting location techniques. *Handbook of Position Location: Theory, Practice, and Advances*, Wiley Online Library, p. 487–520, 2011.

CAMPOS, R. S.; LOVISOLO, L. Genetic algorithm-based cellular network optimisation considering positioning applications. *IET Communications*, IET, v. 13, n. 7, p. 879–891, 2019.

CHENG, L.; LIU, Z.; FENG, L. An improved fingerprint algorithm based on wireless sensor networks. In: IOP PUBLISHING. *Journal of Physics: Conference Series*. [S.l.], 2019. v. 1302, n. 2, p. 022093.

CISCO. *Global Mobile Data Traffic Forecast Update 2015-2020*. [S.l.], 2019. Disponível *on-line* em <www.cisco.com>. Acesso em Dez/2019.

CRUZ, R. M.; SABOURIN, R.; CAVALCANTI, G. D. Dynamic classifier selection: Recent advances and perspectives. *Information Fusion*, Elsevier, v. 41, p. 195–216, 2018.

DEMSAR, J. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, v. 7, p. 1–30, 2006.

DEVILLE, P.; LINARD, C.; MARTIN, S.; GILBERT, M.; STEVENS, F. R.; GAUGHAN, A. E.; BLONDEL, V. D.; TATEM, A. J. Dynamic population mapping using mobile phone data. *Proceedings of the National Academy of Sciences*, National Acad Sciences, v. 111, n. 45, p. 15888–15893, 2014.

DIAS, K.; WINDEATT, T. Dynamic ensemble selection and instantaneous pruning for regression. In: CITESEER. *ESANN*. [S.l.], 2014.

DIAS, K.; WINDEATT, T. Dynamic ensemble selection and instantaneous pruning for regression used in signal calibration. In: *Wermter S. et al. (eds) Artificial Neural Networks and Machine Learning – ICANN 2014. Lecture Notes in Computer Science*. [S.l.: s.n.], 2014. v. 8681, p. 487–520.

DIAZ, E. M.; AHMED, D. B.; KAISER, S. Geographical and fingerprinting data for positioning and navigation systems. *Elsevier*, Elsevier Academic Press, p. 311–334, 2018.

DIETTERICH, T. G. Ensemble methods in machine learning. In: *Proceedings of the First International Workshop on Multiple Classifier Systems (MCS 2000)*. [s.n.], 2000. p. 1–15. ISBN 3-540-67704-6. ISSN 03029743. Disponível em: <http://www.cs.orst.edu/{~}t>.

DUTTA, H. Measuring diversity in regression ensembles. In: *Proc. of the* 4th Indian Int. Conf. on Artif. Intelligence (IICAI 2009). [S.l.: s.n.], 2009. v. 9, p. 1–17.

ERICSSON. *Ericsson mobility report*. [S.l.], 2019. Disponível *on-line* em <https://www.ericsson.com/en/mobility-report>. Acesso em Dez/2019.

FCC. *Second report and order on wireless E911 location Accuracy Requirements*. [S.l.], 2010.

FÉLIX, G.; SILLER, M.; ÁLVAREZ, E. N. A fingerprinting indoor localization algorithm based deep learning. In: IEEE. *Proc. of the 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN)*. [S.l.], 2016. p. 1006–1011.

FREUND, Y.; SCHAPIRE, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. In: *Proc. of the European Conf. on Computational Learning Theory (EuroCOLT 1995)*. [S.l.: s.n.], 1995. v. 55, n. 1, p. 5417–5421.

FREUND, Y.; SCHAPIRE, R. R. E. Experiments with a new boosting algorithm. *International Conference on Machine Learning*, p. 148–156, 1996. ISSN 0706-652X, 1205-7533. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.51.6252>.

GOLDONI, E.; PRANDO, L.; VIZZIELLO, A.; SAVAZZI, P.; GAMBA, P. Experimental data set analysis of rssi-based indoor and outdoor localization in lora networks. *Internet Technology Letters*, Wiley Online Library, v. 2, n. 1, p. e75, 2019.

GÖNEN, M.; ALPAYDIN, E. Multiple kernel learning algorithms. *Journal of Machine Learning Research*, v. 12, p. 2211–2268, 2011. ISSN 15324435. Disponível em: <files/1477/JMLR-Gonen-Alpaydin-2011-Multiple{_}Kernel{_}Learning{_}Algor>.

GUO, X.; LI, L.; ANSARI, N.; LIAO, B. Accurate wifi localization by fusing a group of fingerprints via a global fusion profile. *IEEE Transactions on Vehicular Technology*, IEEE, v. 67, n. 8, p. 7314–7325, 2018.

GUTIERREZ, A.; O'LEARY, S.; RANA, N.; DWIVEDI, Y.; CALLE, T. Using privacy calculus theory to explore entrepreneurial directions in mobile location-based advertising: Identifying intrusiveness as the critical risk factor. *Computers in Human Behavior*, Elsevier, v. 95, p. 295–306, 2019.

HADAVANDI, E.; SHAHRABI, J.; SHAMSHIRBAND, S. A novel boosted-neural network ensemble for modeling multi-target regression problems. *Engineering Applications of Artificial Intelligence*, v. 45, p. 204–219, 2015.

HAMMAD, A.; FAITH, P. *Location based authentication*. [S.l.]: Google Patents, 2017. US Patent 9,721,250 B2.

HAYKIN, S. *Neural Networks: A Comprehensive Foundation*. [S.l.]: Pearson Education, 1999.

HERN, D.; MART, G.; HERNANDEZ-LOBATO, D.; MARTINEZ-MUNOZ, G.; SUAREZ, A. Pruning in ordered regression bagging ensembles. In: *Proc. of the 2006 IEEE Int. Joint Conference on Neural Network (IJCNN 2006)*. [S.l.: s.n.], 2006. p. 1266–1273.

HO, T. K. The random subspace method for constructing decision forest. *IEEE Trans. Pattern Analysis and Machine Intelligence*, v. 20, n. 8, p. 832–844, 1998. ISSN 01628828. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=709601>.

HOSSAIN, A. M.; JIN, Y.; SOH, W.-S.; VAN, H. N. Ssd: A robust RF location fingerprint addressing mobile devices' heterogeneity. *IEEE Trans. Mobile Computing*, IEEE, v. 12, n. 1, p. 65–77, 2011.

HUANG, H.; GARTNER, G.; KRISP, J. M.; RAUBAL, M.; WEGHE, N. V. de. Location based services: ongoing evolution and research agenda. *Journal of Location Based Services*, Taylor Francis, v. 12, n. 2, p. 63–93, 2018. Disponível em: <https://doi.org/10.1080/17489725.2018.1508763>.

HUANG, H.; LIN, S. Widet: Wi-Fi based device-free passive person detection with deep convolutional neural networks. *Computer Communications*, Elsevier, v. 150, p. 357–366, 2020.

IBRAHIM, M.; YOUSSEF, M. A hidden markov model for localization using low-end gsm cell phone'. In: *Proc. of the IEEE Int. Conf. on Communications (ICC 2011)*. [S.l.: s.n.], 2011. p. 1550–3607.

JR, E. S. G. Exponential smoothing: The state of the art. *Journal of Forecasting*, Wiley Online Library, v. 4, n. 1, p. 1–28, 1985.

KIM, J.; JI, M.; JEON, J.-i.; PARK, S.; CHO, Y. k-nn based positioning performance estimation for fingerprinting localization. In: IEEE. *Proc. of the Eighth International Conference onUbiquitous and Future Networks (ICUFN 2016)*. [S.l.], 2016. p. 468–470.

KIM, Y.; CHON, Y.; JI, M.; PARK, S.; CHA, H. Scalable and consistent radio map management scheme for participatory sensing-based wi-fi fingerprinting. In: *Proc. of the Third Int. Workshop Mobile Sensing (IWMS 2013)*. [S.l.: s.n.], 2013. p. 14–18.

KO, A. H.; SABOURIN, R.; BRITTO, A. S. From dynamic classifier selection to dynamic ensemble selection. *Pattern Recognition*, v. 41, n. 5, p. 1735–1748, 2008. ISSN 00313203.

KOTSIANTIS, S.; KANELLOPOULOS, D. Combining bagging, boosting and random subspace ensembles for regression problems. *International Journal of Innovative Computing, Information and Control*, v. 8, n. 6, p. 3953–3961, 2012.

KRAWCZYK, B.; MINKU, L. L.; GAMA, J.; STEFANOWSKI, J.; WOŹNIAK, M. Ensemble learning for data stream analysis: A survey. *Information Fusion*, Elsevier, v. 37, p. 132–156, 2017.

KROGH, A. Neural network ensembles, cross validation, and active learning. In: *Proc. of the 7th International Conference on Neural Information Processing Systems (NIPS 1994)*. [S.l.: s.n.], 1994. p. 231–238.

KUHN, M.; JOHNSON, K. *Applied predictive modeling*. New York: Springer, 2013.

KULKARNI, V. R.; DESAI, V. Computational intelligence for localization of mobile wireless sensor networks. In: *Computational Intelligence:* ***Theories, Applications and Future Directions - Volume II***. [S.l.]: Springer, 2019. p. 449–463.

KUNCHEVA, L. I. Classifier ensembles for changing environments. In: SPRINGER. *Proc. of the International Workshop on Multiple Classifier Systems*. [S.l.], 2004. p. 1–15.

KUNCHEVA, L. I. *Combining pattern classifiers: Methods and algorithms*. [S.l.: s.n.], 2004.

KUNCHEVA, L. I.; RODRIGUEZ, J. J. Classifier ensembles with a random linear oracle. *IEEE Trans. Knowledge and Data Engineering*, v. 19, n. 4, p. 500–508, 2007. ISSN 10414347.

LI, D.; LEI, Y. Deep learning for fingerprint-based outdoor positioning via lte networks. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 19, n. 23, p. 5180, 2019.

LI, D.; WEN, G.; LI, X.; CAI, X. Graph-based dynamic ensemble pruning for facial expression recognition. *Applied Intelligence*, Springer, p. 1–19, 2019.

LI, Z.; XU, K.; WANG, H.; ZHAO, Y.; WANG, X.; SHEN, M. Machine-learning-based positioning: a survey and future directions. *IEEE Network*, IEEE, v. 33, n. 3, p. 96–101, 2019.

LU, Y. Knowledge integration in a multiple classifier system. *Applied Intelligence*, v. 6, n. 2, p. 75–86, 1996. ISSN 0924-669X. Disponível em: <http://link.springer.com/10.1007/BF00117809>.

LUI, G.; GALLAGHER, T.; LI, B.; DEMPSTER, A. G.; RIZOS, C. Differences in rssi readings made by different wi-fi chipsets: A limitation of wlan localization. In: IEEE. *Proc. of the Int. Conference on Localization and GNSS (ICL-GNSS 2011)*. [S.l.], 2011. p. 53–57.

MAO, S.; CHEN, J.-W.; JIAO, L.; GOU, S.; WANG, R. Maximizing diversity by transformed ensemble learning. *Applied Soft Computing*, Elsevier, v. 82, p. 105580, 2019.

MENDES-MOREIRA, J.; SOARES, C.; JORGE, A. M.; SOUSA, J. F. D. Ensemble approaches for regression. *ACM Computing Surveys*, v. 45, n. 1, p. 1–40, 2012. ISSN 03600300. Disponível em: <http://dl.acm.org/citation.cfm?doid=2379776.2379786>.

MOROCHO-CAYAMCELA, M. E.; LEE, H.; LIM, W. Machine learning for 5G/B5G mobile and wireless communications: potential, limitations, and future directions. *IEEE Access*, IEEE, v. 7, p. 137184 – 137206, 2019.

NI, L. M.; LIU, Y.; LAU, Y. C.; PATIL, A. P. LANDMARC: Indoor location sensing using active RFID. *Wireless Networks*, v. 10, n. 6, p. 701–710, 2004. ISSN 1022-0038.

NYHAN, M.; KLOOG, I.; BRITTER, R.; RATTI, C.; KOUTRAKIS, P. Quantifying population exposure to air pollution using individual mobility patterns inferred from mobile phone data. *Journal of Exposure Science & Environmental Epidemiology*, Nature Publishing Group, v. 29, n. 2, p. 238, 2019.

OLIVEIRA, L.; OLIVEIRA, L.; SILVA, G.; TIMOTEO, R.; CUNHA, D. An rss-based regression model for user equipment location in cellular networks using machine learning. *Wireless Networks*, Springer, v. 25, n. 8, p. 4839–4848, 2019.

OLIVEIRA, S.; STEWART, D. E. *Writing scientific software: A guide to good style*. [S.l.]: Cambridge University Press, 2006.

OLSON, R. S.; MOORE, J. H. Tpot: A tree-based pipeline optimization tool for automating machine learning. In: *Automated Machine Learning*. [S.l.]: Springer, 2019. p. 151–160.

PANOV, P.; DŽEROSKI, S. Combining bagging and random subspaces to create better ensembles. In: *Proc. of the International Symposium on Intelligent Data Analysis*. [S.l.: s.n.], 2007. p. 118–129.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011.

PERAL-ROSADO, J. A. del; RAULEFS, R.; LóPEZ-SALCEDO, J. A.; SECO-GRANADOS, G. Survey of cellular mobile radio localization methods: from 1G to 5G. *IEEE Communications Surveys & Tutorials*, IEEE, v. 20, n. 2, Second Quarter, p. 1124 – 1148, 2018.

QUANMING, Y.; MENGSHUO, W.; HUGO, J. E.; ISABELLE, G.; YI-QI, H.; YU-FENG, L.; WEI-WEI, T.; QIANG, Y.; YANG, Y. Taking human out of learning applications: A survey on automated machine learning. 2018. Disponível *on-line* em <https://arxiv.org/abs/1810.13306>. Acesso em Out/2019.

REN, Y.; ZHANG, L.; SUGANTHAN, P. N. Ensemble classification and regression: Recent developments, applications and future directions. *IEEE Computational Intelligence Magazine*, v. 11, n. 1, p. 41–53, 2016. ISSN 1556-603X. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7379058>.

RIZK, H.; SHOKRY, A.; YOUSSEF, M. Effectiveness of data augmentation in cellular-based localization using deep learning. 2018. Disponível *on-line* em <https://arxiv.org/abs/1906.08171>. Acesso em Out/2019.

ROLI, F.; GIACINTO, G.; GIANNI, V. Methods for designing multiple classifier systems. In: *Proc. of the International Workshop on Multiple Classifier Systems*. [S.l.: s.n.], 2001. p. 78–87.

ROONEY, N.; PATTERSON, D.; ANAND, S.; TSYMBAL, A. Dynamic integration of regression models. In: *Roli F., Kittler J., Windeatt T. (eds) Multiple Classifier Systems. MCS 2004. Lecture Notes in Computer Science*. [S.l.: s.n.], 2004. v. 3077, p. 164–173.

SALOMON, A.; MAHAFFEY, K. P. *Mobile communications device payment method utilizing location information*. [S.l.]: Google Patents, 2019. US Patent 2014/0304.165 A1.

SEIDEL, S. Y.; RAPPAPORT, T. S. 914 mhz path loss prediction models for indoor wireless communications in multifloored buildings. *IEEE Trans. on Antennas and Propagation*, v. 40, n. 2, p. 207–217, 1992.

SHRESTHA, D. L.; SOLOMATINE, D. P. Experiments with AdaBoost.RT, an improved boosting scheme for regression. *Neural Computation*, v. 18, n. 7, p. 1678–1710, 2006. ISSN 0899-7667.

SKALAK, D. B. et al. The sources of increased accuracy for two proposed boosting algorithms. In: *Proc. American Association for Artificial Intelligence, AAAI-96, Integrating Multiple Learned Models Workshop*. [S.l.: s.n.], 1996. v. 1129, p. 1133–1273.

SMOLA, A.; SCHÖLKOPF, B. A tutorial on support vector regression. *Statistics and Computing*, v. 14, n. 3, p. 199–222, 2004.

SULTAN, K.; ALI, H.; ZHANG, Z. Big data perspective and challenges in next generation networks. *Future Internet*, Multidisciplinary Digital Publishing Institute, v. 10, n. 7, p. 56, 2018.

SUN, Y.; CHEN, M.; HU, L.; QIAN, Y.; HASSAN, M. M. Asa: Against statistical attacks for privacy-aware users in location based service. *Future Generation Computer Systems*, Elsevier, v. 70, p. 48–58, 2017.

SUZUKI, H. A statistical model for urban radio propogation. *IEEE Trans. Communications*, IEEE, v. 25, n. 7, p. 673–680, 1977.

THOMAS, J.; JOSEPH, A.; JOHNSON, I.; THOMAS, J. Machine learning approach for diabetes prediction. *International Journal of Information*, v. 8, n. 2, 2019.

TIMOTEO, R.; CUNHA, D. C.; SILVA, L. N.; CAVALCANTI, G. A hybrid machine learning approach for mobile user positioning in cellular networks. In: *Proc. of the XXXV SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES E PROCESSAMENTO DE SINAIS (SBrT 2017)*. [S.l.: s.n.], 2017. p. 1–5.

TIMOTEO, R.; SILVA, L.; CUNHA, D.; CAVALCANTI, G. An approach using support vector regression for mobile location in cellular networks. *Computer Networks*, v. 95, n. 2016, p. 51–61, 2016. ISSN 13891286. Disponível em: <http://dx.doi.org/10.1016/j.comnet.2015.12.003>.

TIMOTEO, R. D.; CUNHA, D. C.; CAVALCANTI, G. D. A proposal for path loss prediction in urban environments using support vector regression. In: *Proc. Advanced Int. Conf. Telecommun.* [S.l.: s.n.], 2014. p. 1–5.

TROGH, J.; PLETS, D.; SUREWAARD, E.; SPIESSENS, M.; VERSICHELE, M.; MARTENS, L.; JOSEPH, W. Outdoor location tracking of mobile devices in cellular networks. *EURASIP Journal on Wireless Communications and Networking*, SpringerOpen, v. 2019, n. 1, p. 115, 2019.

UEDA, N.; NAKANO, R. Generalization error of ensemble estimators. In: *Proc. of the IEEE International Conference on Neural Networks*. [S.l.: s.n.], 1996. p. 90–95.

VAPNIK., V. N. *The nature of statistical learning theory*. New York: Springer-Verlag, 1995.

VERZANI, J. *Using R for introductory statistics*. [S.l.]: Chapman and Hall/CRC, 2018.

VIRDIS, A. *Recent advances in network simulation: The OMNeT++ environment and its ecosystem*. [S.l.]: Springer, 2019.

VRIESMANN, L. M.; DE, A.; BRITTO, S.; Soares De Oliveira, L. E.; SABOURIN, R.; HOUNG, A.; KO, R. Improving a dynamic ensemble selection method based on oracle information. *Int. J. Innovative Computing and Applications*, v. 444, n. 3, p. 184–200, 2012. ISSN 1751648X.

WANG, G.; MA, J. A hybrid ensemble approach for enterprise credit risk assessment based on Support Vector Machine. *Expert Systems with Applications*, v. 39, n. 5, p. 5325–5331, 2012. ISSN 09574174.

WANG, Q.; FENG, Y.; ZHANG, X.; SUN, Y.; LU, X. IWKNN: an effective bluetooth positioning method based on isomap and WKNN. *Mobile Information Systems*, Hindawi, v. 2016, 2016.

WANG, X.; GAO, L.; MAO, S. Csi phase fingerprinting for indoor localization with a deep learning approach. *IEEE Internet of Things Journal*, IEEE, v. 3, n. 6, p. 1113–1123, 2016.

WANG, Y.-H.; LI, H.; LUO, X.-L.; SUN, Q.-M.; LIU, J.-N. A 3D fingerprinting positioning method based on cellular networks. *Int. Journal of Distributed Sensor Networks*, Hindawi, v. 10, n. 7, p. 1–8, 2014.

WATT, J.; BORHANI, R.; KATSAGGELOS, A. *Machine learning refined: foundations, algorithms, and applications*. [S.l.]: Cambridge University Press, 2020.

YANG, X.-S. Firefly algorithms for multimodal optimization. In: SPRINGER. *Proc. of the International Symposium on Stochastic Algorithms*. [S.l.], 2009. p. 169–178.

YARKONY, N.; BLAUNSTEIN, N. Prediction of propagation characteristics in indoor radio communication environments. *Progress In Electromagnetics Research*, IET, v. 59, p. 151–174, 2006.

YIU, S.; YANG, K. Gaussian process assisted fingerprinting localization. *IEEE Internet of Things Journal*, IEEE, v. 3, n. 5, p. 683–690, 2016.

ZHANG, C.; MA, Y. *Ensemble machine learning:* ***Methods and applications***. [S.l.]: Springer, 2012.

ZHANG, H.; ZHANG, Z.; ZHANG, S.; XU, S.; CAO, S. Fingerprint-based localization using commercial LTE signals: a field-trial. In: *Proceedings of the 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*. [S.l.: s.n.], 2019. p. 1–5.

ZHOU, Z.-H.; WU, J.; TANG, W. Corrigendum to "Ensembling neural networks: Many could be better than all" [Artificial Intelligence 137 (1–2) (2002) 239–263]. *Artificial Intelligence*, Elsevier B.V., v. 174, n. 18, p. 1570, 2010. ISSN 00043702. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S000437021000158X>.

ZOU, D.; MENG, W.; CHEN, S.; AN, D. A high robustness positioning algorithm for fingerprint localization system. In: IEEE. *Proc. of the Int. Wireless Communications and Mobile Computing Conference (IWCMC 2016)*. [S.l.], 2016(a). p. 730–734.

ZOU, H.; HUANG, B.; LU, X.; JIANG, H.; XIE, L. Standardizing location fingerprints across heterogeneous mobile devices for indoor localization. In: IEEE. *Proc. of the IEEE Wireless Communications and Networking Conference (WCNC 2016)*. [S.l.], 2016(b). p. 1–6.

APÊNDICE A - DATASETS

ABALONE

O *dataset* tem 4.177 instâncias, o objetivo é prever a idade de uma abalone (molusco do mar) baseado em suas medidas físicas. O *dataset* pode ser encontrado na página do Universidade da California Irvine (UCI, *University of California, Irvine*) em *link Abalone*. Na Tabela 15 temos a descrição do *dataset*:

Tabela 15 – Descrição das *features* e *target* do *dataset abalone*

Feature	**Tipo**	**Descrição**
sex	Nominal	M ou F
Length	continuo	cumprimento concha mais longa (mm)
Diameter	continuo	medida perpendicular a Lenght (mm)
Height	continuo	altura considerando a carne (mm)
Whole weight	continuo	peso total (grama)
Shucked weight	continuo	peso carne (grama)
Viscera weight	continuo	peso intestinal após sangramento (grama)
Shell weight	continuo	peso depois da secagem
Rings (alvo)	inteiro	número de anéis

AIRQUALITY

O *dataset* tem 9.358 instâncias, o objetivo é prever a umidade relativa do ar baseado em medições realizadas por sensores. O dataset pode ser encontrado na página do UCI em *link AirQuality*. Na Tabela 16 temos a descrição do *dataset*:

Tabela 16 – Descrição das *features* e *target* do *dataset AirQuality*

Feature	**Tipo**	**Descrição**
Date	Data	data no formato DD/MM/YYYY
Time	Hora	hora no formato HH.MM.SS
CO(GT)	continuo	concentração média horária verdadeira de CO em mg/m^3
PT08.S1(CO)	inteiro	(óxido de estanho) resposta horária média do sensor
NMHC(GT)	inteiro	Altura considerando a carne (mm)
C6H6(GT)	continuo	concentração horária média real de benzeno em microg/m^3
PT08.S2(NMHC)	inteiro	(titânia) resposta horária média do sensor
NOx(GT)t	inteiro	concentração horária média real de NOx em ppb
PT08.S3(NOx)	inteiro	(óxido de tungstênio) resposta horária média do sensor
NO2(GT)	inteiro	concentração horária média real de NO2 em microg/m^3
PT08.S4(NO2)	inteiro	(óxido de tungstênio) resposta horária média do sensor
PT08.S5(O3)	inteiro	(óxido de índio) resposta horária média do sensor
T	continuo	Temperatuar em graus Celsus
RH (alvo)	continuo	umidade Relativa
AH	continuo	umidade Absoluta

AUTO93

O *dataset* tem 93 instâncias, o objetivo é prever o preço de um carro novo baseados em dados do modelo. O *dataset* pode ser encontrado em *link Auto93*. Na Tabela 17 temos a descrição do *dataset*:

Tabela 17 – Descrição das *features* e *target* do *dataset Auto93*

Feature	**Tipo**	**Descrição**
Fabricante	nominal	fabricante do carro
Modelo	nominal	modelo do carro
Tipo	nominal	Tipo do carro
Min.Price	continuo	preço mínimo (em$1000 dolores)
Max.Price	continuo	preço máximo (em $1000 dolores)
MPG.city	continuo	consumo na cidade (milhas/galão)
MPG.highway	continuo	consumo na estrada (milhas/galão)
Cylinders	inteiro	preço máximo (em $1000 dolores)
EngineSize	continuo	Tamanho do motor (litros)
Horsepower	inteiro	potência motor
RPM	inteiro	rotação por minuto (motor)
Rev.per.mile	continuo	preço máximo (em $1000 dolores)
Fuel.tank.capacity	continuo	capacidade do tanque (galão)
Passengers	inteiro	quantidade passageiros
Length	inteiro	cumprimento (inches)
Wheelbase	inteiro	cumprimento eixo (inches)
Width	inteiro	largura (inches)
Origin	nominal	fabricado ou não nos EUA
Turn.circle	continuo	espaço para curva
Rear.seat.room	continuo	espaço bancos traseiros(inches)
Luggage.room	inteiro	capacidade mala (cu. ft.)
Weight	inteiro	peso (pounds)
Price (alvo)	continuo	preço máximo (em $1000 dolores)

AUTOMPG

O *dataset* tem 398 instâncias, o objetivo é prever o consumo do automável tendo como base dados do modelo. O *dataset* pode ser encontrado na página do UCI em *link Autompg*. Na Tabela 18 temos a descrição do *dataset*:

Tabela 18 – Descrição das *features* e *target* do *dataset Autompg*

Feature	**Tipo**	**Descrição**
cylinders	inteiro	quantidade de cilindros no carro
displacement	continuo	espaço interno
horsepower	continuo	potência do motor
weight	continuo	largura
acceleration	continuo	aceleração
modelYear	inteiro	ano do modelo
origin	nominal	origem do modelo
carName	nominal	nome do modelo

BODYFAT

Os dados contêm estimativas do percentual de gordura corporal determinada pela pesagem subaquática e várias medidas de circunferência corporal para 252 homens. Assim, temos 252 instâncias. O *dataset* pode ser encontrado na página *link BodyFat*. Na Tabela 19 temos a descrição do *dataset*:

Tabela 19 – Descrição das *features* e *target* do *dataset Bodyfat*

Feature	**Tipo**	**Descrição**
density	continuo	densidade (subaquática)
percentBFat	continuo	percentual de gordura
age	inteiro	idade
Weight	continuo	peso
Height	continuo	Altura
NeckCirc	continuo	circunferência do pescoço (cm)
ChestCirc	continuo	circunferência peito (cm)
Abdomen2Circ	continuo	circunferência do abdômen 2 (cm)
HipCirc	continuo	circunferência do quadril (cm)
ThighCirc	continuo	circunferência da coxa (cm)
KneeCirc	continuo	circunferência do joelho (cm)
AnkleCirc	continuo	circunferência do tornozelo (cm)
BicepsCirc	continuo	circunferência do bíceps (estendida) (cm)
ForearmCirc	continuo	circunferência do antebraço (cm)
WristCirc	continuo	circunferência do pulso (cm)

BOSTON HOUSING

Os dados contém os valores médios das casas na cidade de Boston nos anos 70. Ao todo, são 506 instâncias com 13 *features* relacionadas ao local do imóvel. O *dataset* pode ser encontrado na página do UCI em *link Housing*. Na Tabela 20 temos a descrição do *dataset*:

Tabela 20 – Descrição das *features* e *target* do *dataset Boston Housing*

Feature	**Tipo**	**Descrição**
CRIM	continuo	taxa de criminalidade per-capita
ZN	continuo	proporção de terrenos residenciais maior que 25.000 ft^2
INDUS	continuo	proporção terrenos comerciais e não comerciais
CHAS	inteiro	1 se o trecho limita o rio; 0 caso contrário
NOX	continuo	concentração de óxidos nítricos (partes por 10 milhões)
RM	continuo	número médio de quartos por habitação
AGE	continuo	proporção de propriedades construídas antes de 1940
DIS	continuo	distância ponderada aos cinco centros de Boston
RAD	inteiro	índice de acessibilidade às rodovias
TAX	inteiro	taxa de imposto sobre a propriedade
PTRATIO	continuo	proporção professor por aluno
B	continuo	$1000(Bk - 0.63)^2$ (bk= prop. terrenos vazios)
LSTAT	continuo	menor status da população
MEDV (alvo)	continuo	valor médio casas ocupadas (em $1000)

CARSEATS

O *dataset* contém 400 instâncias e 10 *features*. O objetivo é prever a quantidade de unidades vendidas de cadeiras de criança utilizadas em carro. O *dataset* pode ser encontrado na pacote R Um introdução ao aprendizagem de máquina (ISL, *An Introduction to Statistical Learning*) em *link Carseats*. Na Tabela 21 temos a descrição do *dataset*:

Tabela 21 – Descrição das *features* e *target* do *dataset Carseats*

Feature	**Tipo**	**Descrição**
Sales (alvo)	continuo	unidades vendidas(x1000) por local.
CompPrice	inteiro	preço competidor no local.
Income	inteiro	nível de renda comunidade.
Advertising	inteiro	orçamento local para propaganda. (x100)
Population	inteiro	população da região (x1000).
Price	inteiro	preço cobrado pela cadeira na região.
ShelveLoc	nominal	nível da localização produto.
Age	continuo	idade média da população.
Education	nominal	nível de eduação região.
Urban	nominal	indica se o local da loja é urbano.
US	nominal	indica se a loja é nos EUA.

CONCRETE

Há 1.030 instâncias no dataset, e o objetivo é prever a resistência à compressão do concreto em função da idade da mistura e do peso por m^3 de 7(sete) componentes, assim há oito *features* no dataset. O dataset pode ser encontrado na página do UCI em *link Concrete*. Na Tabela 22 temos a descrição do *dataset*:

Tabela 22 – Descrição das *features* e *target* do *dataset Concrete*

Feature	**Tipo**	**Descrição**
Cement	continuo	peso componente 1 por m^3
BlastFurnaceSlag	continuo	peso componente 2 por m^3
FlyAsh	continuo	peso componente 3 por m^3
Water	continuo	peso componente 4 por m^3
Superplasticizer	continuo	peso componente 5 por m^3
CoarseAggregate	continuo	peso componente 6 por m^3
FineAggregate	continuo	peso componente 7 por m^3
Ageday	inteiro	idade da mistura em dias
ConCompStrength (alvo)	continuo	resistência do concreto à compressão

CPUS

Dados relativos de desempenho da CPU, os quais são descritos em termos de tempo de ciclo, tamanho da memória, etc. O dataset tem 209 instâncias e 8 *features* e pode ser encontrado na página do UCI em *link Cpus*. Na Tabela 23 temos a descrição do *dataset*:

Tabela 23 – Descrição das *features* e *target* do *dataset Cpus*

Feature	**Tipo**	**Descrição**
vendor	nominal	vendedor
modelName	nominal	nome do modelo
MYCT	inteiro	clico da máquina (nano segundos)
MMIN	inteiro	memória principal mínima (kbytes)
MMIN	inteiro	memória principal máxima (kbytes)
CACH	inteiro	memória cache (kbytes)
CHMIN	inteiro	unidade mínima de canais
CHMAX	inteiro	unidade máxima de canais
PRP	inteiro	performance propaganda
ERP (alvo)	inteiro	performance estimada

FORESTFIRES

O *dataset* tem 513 instâncias e 12 *featuares*, o objetivo é fazer a previsão da área queimada de um parque florestal utilizado as *features* especificadas na Tabela24. O ForestFire *dataset* pode ser encontrado na página do UCI *link ForestFire*. Na Tabela 24 temos a descrição do *dataset*:

Tabela 24 – Descrição das *features* e *target* do *dataset ForestFire*

Feature	**Tipo**	**Descrição**
X	inteiro	coordenada relativa (1 a 9)
y	inteiro	coordenada relativa (1 a 9)
month	nominal	mês do incêndio (jan a dez)
day	nominal	dia do incêndio (seg a dom)
FFMC	continuo	Índice FFMC do sistema FWI: 18,7 a 96,20
DMC	continuo	índice DMC do sistema FWI: 1.1 a 291.3
DC	continuo	Índice DC do sistema FWI: 7,9 a 860,6
ISI	continuo	índice ISI do sistema FWI: 0,0 a 56,10
temp	continuo	temperatura em graus Celsius: 2,2 a 33,30
RH	continuo	umidade relativa do ar em%: 15.0 a 100
wind	continuo	velocidade do vento em km / h: 0,40 a 9,40
rain	continuo	chuva externa em mm / m2: 0.0 a 6.4
area(alvo)	continuo	área queimada da floresta (em ha): 0,00 a 1090,84

HITTERS

O Dataset com 322 instâncias e 19 *features*, a variável alvo é o número de *hits* que jogador irá acertar na temporada. O dataset pode ser encontrado no pacote R ISL em *link Hitters*. Na Tabela 21 temos o detalhamento do *dataset*:

Tabela 25 – Descrição das *features* e *target* do *dataset Hitters*

Feature	**Tipo**	**Descrição**
tBat	inteiro	número de batidas(at bat) em 1986
Hits	inteiro	número de *hits* em 1986
HmRun	inteiro	número de *home runs* em 1986
Runs	inteiro	número de *runs* em 1986
RBI	inteiro	número de *runs batted* em 1986
Walks	inteiro	número de *walks* em 1986
Years	inteiro	número de anos na liga principal
CAtBat	inteiro	número de time durante sua carreira (*at bat*)
CHits	inteiro	número de hits em sua carreira
CHmRun	inteiro	número de *home runs* na carreira
CRuns	inteiro	número de *runs* durante a carreira
CRBI	inteiro	número de *runs batted* na carreira
CWalks	inteiro	número de *walks* na carreira
League	nominal	nível do jogador na liga 1986 (A e N)
Division	nominal	Divisão do Jogador(E e W)
PutOuts	inteiro	número de *put outs* em 1986
Assists	inteiro	número de assitência em 1986
Errors	inteiro	número de erros em 1986
Salary	inteiro	salário anual em 1987 (x1000 dolares)
NewLeague	nominal	Liga do jogado no inicio de 1987 (A e N)

SERVO

O dataset tem 167 instâncias e 5 *features*, a variável alvo é o tempo de resposta de um servomecanismo e as *features* estão relacionadas aos dispositivos usados pelo servomecanismo. O dataset pode ser encontrado na página do UCI *link Servo Dataset*. Na Tabela 26 temos sua descrição:

Tabela 26 – Descrição das *features* e *target* do *dataset Servo*

Feature	**Tipo**	**Descrição**
motor	nominal	A,B,C,D,E
screw	nominal	A,B,C,D,E
pgain	inteiro	3,4,5,6
vgain	inteiro	1,2,3,4,5
class (alvo)	continuo	tempo resposta(0.13 a 7.10)

SLUMP

O *Dataset* tem 103 instâncias, sete *features* e três variáveis pode ser utilizadas como alvo. As variáveis alvos são a queda, a vazão e resistência à compressão do concreto. E as *features* são sete ingredientes utilizados na fabricação do concreto. Mais detalhes sobre o dataset e o arquivo podem ser encontrados na página do UCI em *link Servo Dataset*. Na Tabela 27 temos a descrição do *dataset*:

Tabela 27 – Descrição das *features* e *target* do *dataset Slump*

Feature	**Tipo**	**Descrição**
Cement	inteiro	cimento
Slag	inteiro	bv
FlyAsh	inteiro	cinzas (no ar)
Water	inteiro	água
SP	inteiro	super plastificante
CoarseAggr	inteiro	agregado grosso
FineAggr	inteiro	agregado fino
SLUMP (alvo)	continuo	queda do concreto
FLOW (alvo)	continuo	vazão do concreto
ComprStren28 (alvo)	continuo	resistência à compressão(dia 28)

WAGE

Dataset com salário de 3000 empregados (todos masculinos) e 10 *features*. dataset pode ser encontrado no pacote R ISL em *link Hitters*. Na Tabela 28a descrição do *dataset*:

Tabela 28 – Descrição das *features* e *target* do *dataset Wage*

Feature	**Tipo**	**Descrição**
year	inteiro	Ano que salário foi informado
age	inteiro	idade do trabalhador
maritl	nominal	Estado civil do trabalhador
race	nominal	origem do trabalhador
education	nominal	nível de educação do trabalhador
region	nominal	região do pais
jobclass	nominal	tipo do trabalho
health	nominal	estado de saúde do trabalhador
healthIns	nominal	se trabalhador tem plano de saúde
logwage	continuo	registro de salário
wage	continuo	Salário do trabalhador