MARCOS ALBERTO MOCHINSKI

USO DE *MACHINE LEARNING* PARA POSICIONAMENTO DE ROTEADORES E *GATEWAYS* EM REDES DE COMUNICAÇÃO SEM FIO EM *SMART GRIDS*

Curitiba - PR, Brasil

2024

MARCOS ALBERTO MOCHINSKI

USO DE *MACHINE LEARNING* PARA POSICIONAMENTO DE ROTEADORES E *GATEWAYS* EM REDES DE COMUNICAÇÃO SEM FIO EM *SMART GRIDS*

Tese apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná, como requisito parcial para obtenção do título de Doutor em Informática.

Pontifícia Universidade Católica do Paraná - PUCPR

Programa de Pós-Graduação em Informática - PPGIa

Orientador: Prof. Dr. Fabrício Enembreck

Coorientador: Prof. Dr. Marcelo Eduardo Pellenz

Curitiba - PR, Brasil

2024

Dados da Catalogação na Publicação
Pontifícia Universidade Católica do Paraná
Sistema Integrado de Bibliotecas – SIBI/PUCPR
Biblioteca Central
Edilene de Oliveira dos Santos CRB 9 / 1636

M688u
2024

Mochinski, Marcos Alberto
Uso de machine learning para posicionamento de roteadores e gateways em redes de comunicação sem fio em smart grids / Marcos Alberto Mochinski ; orientador: Fabrício Enembreck ; coorientador: Marcelo Eduardo Pellenz. -- 2024.
201 f. : il. ; 30 cm

Tese (doutorado) – Pontifícia Universidade Católica do Paraná, Curitiba, 2024
Bibliografia: f. 167-179

1. Informática. 2. Aprendizado do computador. 3. Redes elétricas inteligentes. 4. Roteadores (Redes de computadores). 5. Sistemas de comunicação sem fio. I. Enembreck, Fabrício. II. Pellenz, Marcelo Eduardo. III. Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Informática. IV. Título

CDD 20. ed. – 004

Pontifícia Universidade Católica do Paraná
Escola Politécnica
Programa de Pós-Graduação em Informática

Curitiba, 15 de maio de 2024.

37-2024

DECLARAÇÃO

Declaro para os devidos fins, que **Marcos Alberto Mochinski** defendeu a tese de Doutorado intitulada "**USO DE *MACHINE LEARNING* PARA POSICIONAMENTO DE ROTEADORES E *GATEWAYS* EM REDES DE COMUNICAÇÃO SEM FIO EM *SMART GRIDS*"**, na área de concentração Ciência da Computação no dia 04 de abril de 2024, no qual foi aprovado.

Declaro ainda, que foram feitas todas as alterações solicitadas pela Banca Examinadora, cumprindo todas as normas de formatação definidas pelo Programa.

Por ser verdade firmo a presente declaração.

gov.br Documento assinado digitalmente
EMERSON CABRERA PARAISO
Data: 15/05/2024 16:55:07-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Emerson Cabrera Paraiso
Coordenador do Programa de Pós-Graduação em Informática

MARCOS ALBERTO MOCHINSKI

USO DE *MACHINE LEARNING* PARA POSICIONAMENTO DE ROTEADORES E *GATEWAYS* EM REDES DE COMUNICAÇÃO SEM FIO EM *SMART GRIDS*

Tese apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná, como requisito parcial para obtenção do título de Doutor em Informática.

Trabalho _____. Curitiba - PR, Brasil, 04 de abril de 2024:

Prof. Dr. Fabrício Enembreck
(PPGIa – PUCPR) – Orientador

Prof. Dr. Marcelo Eduardo Pellenz
(PPGIa – PUCPR) – Coorientador

Prof. Dr. Voldi Costa Zambenedetti
(CISEI – PUCPR) – Convidado 1

Prof. Dr. Edgard Jamhour
(CISEI – PUCPR) – Convidado 2

Prof. Dr. Alexandre Rasi Aoki
(UFPR) – Convidado 3

Curitiba - PR, Brasil
2024

Este trabalho é dedicado à minha família e a todos que me auxiliaram no seu desenvolvimento.

Agradecimentos

Agradeço a Deus, à minha família, ao meu orientador, ao meu coorientador, aos professores e integrantes da equipe técnico-administrativa do curso de Doutorado em Informática da Pontifícia Universidade Católica (PUCPR), e a todos os demais que me deram o apoio e o incentivo necessários para o desenvolvimento deste trabalho.

Um agradecimento especial aos professores Fabrício Enembreck (orientador) e Marcelo Eduardo Pellenz (coorientador) pelas orientações e contribuições dadas nas dezenas de reuniões semanais que foram realizadas durante todo o desenvolvimento desta pesquisa, bem como por estarem sempre disponíveis para o apoio e esclarecimento das diversas dúvidas que precisaram ser sanadas ao longo do trabalho.

Meus agradecimentos se estendem a todos os integrantes das equipes envolvidas no desenvolvimento do Projeto de Pesquisa ANEEL/COPEL-DIS, PD-02866-0478/2017 (Ferramenta de Planejamento Ótimo de Comunicação e Tecnologias Emergentes para Automação e Monitoramento de Redes de Comunicação), incluindo pesquisadores e profissionais da PUCPR, COPEL e LACTEC, que contribuíram no desenvolvimento desta pesquisa, viabilizando-a em seus aspectos acadêmicos, técnicos, financeiros e administrativos.

Por todo o apoio e orientações, agradeço ao professor Voldi Costa Zambenedetti, gerente/coordenador do projeto de pesquisa na PUCPR, e aos professores Edgar Jamhour e Ivan Jorge Chueiri pelas contribuições técnicas e apoio.

Agradecimentos, também, a Maurício Biczkowski, gerente do projeto de pesquisa por parte da COPEL Distribuição, SSG - Superintendência de Smart Grid e Projetos Especias, por todo o apoio técnico dado para a execução deste trabalho.

Esta pesquisa foi apoiada pelo programa de pesquisa e desenvolvimento tecnológico da 'Companhia Paranaense de Energia - Copel', por intermédio do projeto PD-02866-0478/2017, regulado pela ANEEL.

This work was supported by the 'Companhia Paranaense de Energia - COPEL' research and technological development program, through the PD-02866-0478/2017 project, regulated by ANEEL.

"Data is food for AI." — Andrew Ng

Resumo

O posicionamento de roteadores e *gateways* de uma rede de comunicação sem fio para *Smart Grid* (Rede Elétrica Inteligente) é um problema *NP-Hard*, visto que o número de topologias possíveis cresce exponencialmente de acordo com o número de postes e medidores inteligentes a serem considerados. O perfil do terreno pode levar a perdas de qualidade de sinal entre um medidor e roteadores e *gateways* instalados em postes selecionados, tornando o problema ainda mais difícil. Adicionalmente, a topologia de comunicação deve levar em consideração a posição dos dispositivos de automação de distribuição (DAs) previamente instalados para suportar a operação remota da rede elétrica. Neste estudo, o problema de posicionamento de roteadores e *gateways* é explorado com o uso de duas abordagens. A primeira sugere o uso de um método heurístico analítico, e foi denominada de AIDA (*AI-driven AMI network planning with DA-based information and a link-specific propagation model*), que usa a Potência Recebida Estimada no Enlace (*Link Received Power*, LRP) como métrica para avaliar a possibilidade de conexão entre medidores e posições candidatas. O método também usa os valores de LRP calculados para as arestas de uma *Minimum Spanning Tree* (Árvore Geradora Mínima) para propor uma análise simplificada de conexões com múltiplos saltos. Outra abordagem, denominada de AIDA-ML, avalia o uso de algoritmos de *Machine Learning* (Aprendizagem de Máquina) para implementar uma estratégia de posicionamento mais eficiente que a utilizada pelo método analítico proposto, capaz de aprender a partir dele. Para a implementação do método baseado em *Machine Learning*, um processo de *Feature Engineering* (Engenharia de Características) é utilizado para a criação de *datasets* com características que consigam reproduzir o funcionamento do método analítico e seus resultados. O uso de uma estratégia de *machine learning* tem o propósito principal de alcançar resultados comparáveis aos obtidos com o método analítico, demandando, porém, menor tempo de processamento. Em experimentos realizados com dados de cenários reais, em uma rede aérea de distribuição, incluindo informações sobre 26 municípios do estado do Paraná, Brasil, com coordenadas geográficas de 466.237 medidores inteligentes e 352.867 postes, os resultados obtidos com diferentes algoritmos de *machine learning* sugerem que o método AIDA-ML é capaz de assegurar a cobertura de conexão de medidores inteligentes dentro dos mesmos parâmetros mínimos estabelecidos para o método analítico, com a vantagem de reduzir em 87,60%, em média, o tempo de processamento em comparação ao que seria consumido pela abordagem heurística. O AIDA-ML também é capaz de reduzir em 96,86% o número de cálculos de LRP exigidos para o posicionamento de roteadores/*gateways* em comparação ao efetuado por AIDA.

Palavras-chave: *Machine Learning*, *Smart Grid*, Posicionamento de Roteadores e Gateways.

Abstract

The placement of routers and gateways of a Smart Grid wireless communication network is an NP-Hard problem as the number of possible topologies grows exponentially according to the number of poles and smart meters to consider. The terrain profile can lead to signal quality losses between a meter and routers and gateways installed on selected poles, making the problem even more difficult. Additionally, the communication topology must consider the position of the distribution automation devices (DAs) previously installed to support the remote operation of the electrical network. This study explores the problem of positioning routers and gateways using two approaches. The first one suggests the use of a heuristic, analytical method and it is called AIDA (AI-driven AMI network planning with DA-based information and a link-specific propagation model), which uses the Link Received Power (LRP) as a metric to assess the connectivity between meters and candidate positions. The method also uses the calculated LRP values for the edges of a Minimum Spanning Tree proposing a simplified analysis of multihop connections. Another approach, called AIDA-ML, evaluates the use of Machine Learning algorithms to implement a more efficient positioning strategy than the one used by the proposed analytical method, capable of learning from it. For the machine learning-based method's implementation, a feature engineering process is used to create datasets with characteristics that can reproduce the operation of the analytical method and its results. Using a machine learning strategy aims to achieve results comparable to those obtained with the analytical method but demanding less processing time. In experiments carried out with data from real scenarios, in an overhead electrical power transmission network, including information about 26 cities from the state of Paraná, Brazil, with geographic coordinates of 466,237 smart meters and 352,867 poles, the results obtained using different machine learning algorithms suggest that AIDA-ML can ensure the connection coverage of smart meters within the same minimum parameters established for the analytical method, with the advantage of reducing by 87.60%, on average, the processing time compared to what would be consumed by the heuristic approach. AIDA-ML is also capable of reducing by 96.86% the number of LRP calculations demanded for the positioning of routers/gateways compared to AIDA.

Keywords: Machine Learning, Smart Grid, Routers and Gateways positioning.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

AI	*Artificial Intelligence*
AIDA	*AI-driven AMI network planning with DA-based information and a link-specific propagation model*
AIDA-ML	AIDA *Machine Learning*
AMI	*Advanced Metering Infrastructure*
ANH	*Average Number of Hops*
AP	*Access Point*
AutoML	*Automated Machine Learning*
BU	Abordagem *Bottom-Up* do método AIDA
CART	*Classification and Regression Trees*
CD	*Coordinating Devices*
CL	*Clustering*
COD	Centro de Operação de Distribuição
CP	*Candidate Position*
CPP	*Capacitated Placement Problem*
C-RAN	*Cloud Radio Access Network, Cloud RAN* ou *Centralized-RAN*
CT	*Constraint Programming*
DA	*Distribution Automation device*
DAP	*Data Aggregation Point*
DB	*Distance-based Analysis*
DCU	*Data Concentrator Unit*
DL	*Diffraction Loss*
DNP3	*Distributed Network Protocol 3*
DT	*Decision Tree*

DTN	*Delay Tolerant Network*
EN	*End-node*
FN	*False Negative*
FNR	*False Negative Rate*
FP	*False Positive*
FPR	*False Positive Rate*
GA	*Genetic Algorithm*
GNP	*Gateway Node Placement Problem*
GW	*Gateway*
HAN	*Home Area Network*
HE	*Heuristics*
HPO	*Hyperparameter Optimization*
HTS	*High Throughput Satellite*
IDSS	*Intelligent Decision Support System*
IA	Inteligência Artificial
IA/ML	Inteligência Artificial/*Machine Learning*
ILP	*Integer Linear Programming*
IoT	*Internet of Things*
IP	*Internet Protocol*
ITU	*International Telecommunication Union*
Iter.	Iterações ou Número de Iterações
LS	*Local Search*
LoRa	*Long Range*
LoRaWAN	*Protocolo de comunicação para rede LoRa*
LPL	*Link Power Loss*
LPWAN	*Low Power Wide Area Network*

LR	*Logistic Regression*
LRP	*Link Received Power*
MH	*Metaheuristics*
MI	Medidor Inteligente
MILP	*Mixed Integer Linear Programming*
MINLP	*Mixed-Integer Non-Linear Programming*
MIP	*Mixed Integer Programming*
ML	*Machine Learning*
MST	*Minimum Spanning Tree*
NAN	*Neighborhood Area Network*
NN	*Neural Network*
OP	*Optimization*
PDR	*Packet Delivery Ratio*
PL	*Path Loss*
PLC	*Power-line Communication*
PSA	*Pareto Simulated Annealing*
PSO	*Particle Swarm Optimization*
PR	*Probabilistic Model / Stochastic Model*
Quant.	Quantidade
QoS	*Quality of Service*
RF	*Random Forest*
RG	*Regression Model*
RNN	*Recurrent Neural Network*
RPL	*IPv6 Routing Protocol for Low Power and Lossy Networks*
RSSI	*Received Signal Strength Indication*
RT	Roteador

SA	*Simulated Annealing*
SCADA	*Supervisory Control and Data Acquisition*
SDN	*Software Defined Network*
SG	*Smart Grid*
SINR	*Signal to Interference and Noise Ratio*
SL	*Supervised Learning*
SM	*Smart Meter*
SNR	*Signal to Noise Ratio*
ST	*Statistical Analysis*
TD	Abordagem *Top-Down* do método AIDA
TN	*True Negative*
TNR	*True Negative Rate*
TP	*True Positive*
TPR	*True Positive Rate*
UC	*Utility Center*
VLAN	*Virtual Local Area Network*
WAN	*Wide Area Network*
Wi-SUN	*Wireless Smart Utility Network*
WMN	*Wireless Mesh Network*
WOM	*Whale Optimization Method*
XGB	*XGBoost*

Sumário

1 INTRODUÇÃO

O uso de técnicas de inteligência artificial (IA) e *machine learning* (ML) está cada vez mais evidente no cotidiano das mais diversas áreas da sociedade, como na segurança, na saúde, nos esportes, na detecção de fraudes, nas finanças, na automação industrial, na recomendação de produtos, no processamento de linguagem natural, entre outras (LI; ZHANG, 2017; ISLAM et al., 2020; ZHANG, 2021; ASSUNÇÃO et al., 2022; HUANG et al., 2022). Em relação à sua aplicação em redes de comunicação sem fio, em especial na área de *Smart Grids* (SG), a crescente variedade de tecnologias desenvolvidas aplicáveis à área, como os sistemas celulares 5G e tecnologias de rede como Wi-SUN e LoRa, sugerem oportunidades de pesquisas em diferentes frentes, como no planejamento da rede, no monitoramento de eventos ou no controle da operação do sistema, entre outras (MINHAJ et al., 2023; KUNDACINA et al., 2022; ZHENG et al., 2022; MIRZAEE et al., 2021; SONG et al., 2021; DELIGIANNIS; KOUTROUBINAS; KORONIAS, 2019).

Contexto da Pesquisa

A comunicação entre os nós de uma rede sem fio (rede *wireless*) de múltiplos saltos (*multihop*) depende da existência de elementos como roteadores, que auxiliam no roteamento de pacotes de dados, e *gateways*, que atuam como concentradores e permitem a interconexão com outras redes. Nesse contexto, posicionar tais elementos de comunicação não é uma tarefa trivial.

O posicionamento de roteadores e *gateways* inclui as tarefas de estabelecer a quantidade necessária de equipamentos e a localização em que devem ser instalados de forma a possibilitar a maior quantidade de benefícios em critérios como cobertura, desempenho da rede e custo total da solução. Sob o ponto de vista de planejamento, é importante buscar um *trade-off* entre tais critérios, considerando as características técnicas dos equipamentos a serem instalados e requisitos de qualidade estabelecidos para o projeto.

Questão de Pesquisa

A questão de pesquisa que esta tese busca responder pode ser formulada da seguinte forma: *Dado um conjunto de posições candidatas para o posicionamento de roteadores/gateways, e obtendo características do cenário (medidores, postes, equipamentos de automação) e topografia em seu entorno, é possível utilizar técnicas de machine learning para determinar as posições para instalação de roteadores e gateways de forma a assegurar conectividade e desempenho em redes de comunicação de smart grids?*

1.1 DEFINIÇÃO DO PROBLEMA

O posicionamento de roteadores e *gateways* em redes *wireless* é uma tarefa complexa (*NP-Hard*), (AOUN et al., 2006; AALAMIFAR et al., 2014; LANG et al., 2022; HELLER; SHERWOOD; MCKEOWN, 2012), dada a grande quantidade de posições possíveis para a instalação de tais equipamentos e a multiplicidade de possibilidades de conexões entre medidores e os dispositivos concentradores. Tal posicionamento reflete diretamente nas métricas de desempenho da rede, impactando, especialmente, a cobertura, a latência e *throughput* do sistema como um todo.

A Figura 1 apresenta um cenário geral de aplicação do problema de posicionamento de equipamentos de comunicação em *smart grid*, mais especificamente na área de infraestrutura de medição avançada (AMI, *Advanced Metering Infrastructure*). Nesse cenário, que considera que as residências, instalações comerciais, industriais e rurais fazem uso de medidores inteligentes de energia, podem ser observadas regiões com alta concentração desses equipamentos e regiões mais esparsas. Além disso, a topografia do terreno é variável, com regiões mais planas e áreas mais acidentadas, favorecendo diferentes qualidades de sinal transmitido, visto que a comunicação entre os equipamentos é *wireless* (comunicação sem-fio). Elementos da rede elétrica, em especial os postes de energia, são considerados como posições de interesse para a instalação de equipamentos de comunicação, como roteadores e *gateways*. É por intermédio de tais equipamentos que os dados de leituras dos medidores inteligentes são encaminhados para o Centro de Operação de Distribuição (COD) do sistema.

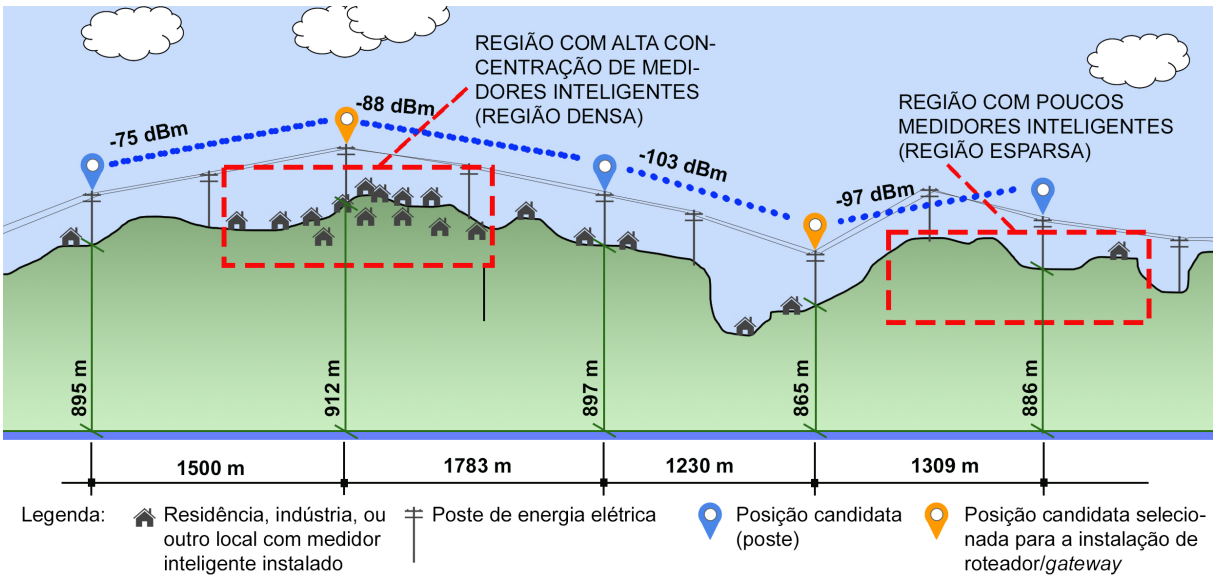


Figura 1 – Cenário de interesse para o posicionamento de roteadores/*gateways* em *smart grid*.

Fonte: Adaptado de (MOCHINSKI et al., 2024).

Estabelecer a quantidade ideal de equipamentos apresenta reflexos no desempenho

e no custo total da solução. Estabelecer um conjunto muito pequeno de tais equipamentos pode ser insuficiente para garantir a cobertura necessária e o desempenho esperado para o sistema. Em contrapartida, definir como solução um conjunto excessivo de equipamentos pode resultar em desperdício de investimento sem, necessariamente, conseguir obter um desempenho ideal para o sistema.

A definição da posição ideal para a instalação de roteadores e *gateways*, deve considerar posições candidatas com capacidade técnica suficiente para viabilizar que tais equipamentos recebam a energia elétrica que necessitam para a sua ativação. Além disso este posicionamento deve ocorrer em locais onde a propagação de sinal seja mais favorável. Em redes elétricas inteligentes, os postes existentes em determinada região podem ser indicados como posições candidatas para a instalação desses dispositivos, desde que possuam os requisitos técnicos suficientes. Esses requisitos incluem a tensão máxima exigida pelo dispositivo e a ausência de incompatibilidade técnica com os equipamentos já instalados nesse mesmo poste.

Deve-se dar destaque, também, à necessidade de minimizar a quantidade de posições candidatas selecionadas para a instalação de roteadores e *gateways*, visto que a quantidade de dispositivos tem impacto direto no custo de infra-estrutura.

1.2 OBJETIVOS DA PESQUISA

Objetivo Geral:

O objetivo geral da presente pesquisa consiste em propor uma estratégia baseada no uso de *machine learning* para posicionamento de *gateways* e roteadores em redes de comunicação sem fio em *smart grids*. A topologia da rede proposta por esse método deve garantir cobertura e desempenho dentro dos valores estabelecidos, a um custo reduzido; ou seja, com quantidade de equipamentos suficiente para assegurar os requisitos de qualidade estabelecidos para o projeto. O uso de uma abordagem baseada em *machine learning* visa mitigar a complexidade do problema e da solução proposta, em especial quando comparada ao uso de uma abordagem analítica.

Objetivos Específicos:

Os objetivos específicos estabelecidos para esta pesquisa incluem:

- Avaliar estratégias de *machine learning* existentes, aplicáveis ao planejamento de redes *wireless*, e propor um método capaz de recomendar posicionamento que assegure desempenho geral da rede dentro de parâmetros pré-estabelecidos, indicados pela indústria e pelo operador da rede.

- Desenvolver um método analítico de posicionamento que permita rotular posições candidatas a fim de viabilizar a construção de uma base de dados de treinamento que possa ser utilizada pelos algoritmos de *machine learning* (aprendizagem de máquina) para geração do modelo final de posicionamento. O desenvolvimento desse método é justificado pela inexistência na literatura de uma base de dados disponível para aprendizagem de posicionamento de dispositivos de comunicação de redes sem fio para *smart grids*.

- Analisar técnicas de extração de *features* a partir de objetos existentes no entorno de posições candidatas e estabelecer conjunto de características suficientes para o uso de algoritmos de *machine learning* para posicionamento de roteadores/*gateways*.

1.3 HIPÓTESES DE PESQUISA

Nesta seção são formuladas as hipóteses a serem investigadas pela pesquisa e responder à questão de pesquisa formulada na Seção 1.

H01 (Hipótese Nula 01): Não é possível estabelecer posições de roteadores/*gateways* em redes *wireless* a partir de características de elementos do cenário existente no entorno de posições candidatas.

HA1 (Hipótese Alternativa 01): É possível estabelecer posições de roteadores/*gateways* em redes *wireless* a partir de características de elementos do cenário existente no entorno de posições candidatas.

H02 (Hipótese Nula 02): Não é possível atingir qualidade das conexões da rede em parâmetros aceitáveis usando técnicas de *machine learning* para posicionamento de roteadores/*gateways*.

HA2 (Hipótese Alternativa 02): É possível atingir qualidade das conexões da rede em parâmetros aceitáveis usando técnicas de *machine learning* para posicionamento de roteadores/*gateways*.

H03 (Hipótese Nula 03): O estabelecimento de posições candidatas com o uso de algoritmos de ML não diminui o espaço de conexões entre medidores e postes a serem avaliadas em comparação ao utilizado por um método analítico.

HA3 (Hipótese Alternativa 03): O estabelecimento de posições candidatas com o uso de algoritmos de ML diminui o espaço de conexões entre medidores inteligentes e postes a serem avaliadas em comparação ao utilizado por um método analítico.

1.4 MÉTODO DE PESQUISA

Esta pesquisa foi desenvolvida com a utilização de um método experimental (experimentação). Inicialmente, foi estabelecido um modelo de trabalho e a definição do método avaliativo a ser utilizado. Depois disso, um processo iterativo de realização de experimentos, observação de resultados e ajustes de modelo foi utilizado.

Com foco especial na análise da aplicabilidade de métodos de *machine learning* para o posicionamento de dispositivos de comunicação, uma fase inicial incluiu o desenvolvimento de um método analítico, inovador em suas características em relação a outros existentes na literatura, para servir como base para a caracterização, implementação e experimentação do método baseado em aprendizagem de máquina.

1.5 DESAFIOS

Vários foram os desafios considerados no desenvolvimento da pesquisa.

Um primeiro desafio foi o de selecionar um conjunto de características capazes de qualificar posições candidatas de forma a auxiliar o processo de classificação por métodos de *machine learning*. Essa dificuldade advém do fato de que muitas das estratégias de posicionamento de dispositivos de comunicação encontradas na literatura são baseadas em técnicas heurísticas e modelos de otimização que não fazem uso de técnicas de *machine learning*.

Outro desafio a ser destacado foi o de selecionar algoritmo de *machine learning* que consiga demonstrar nível de aprendizagem capaz de posicionar roteadores e *gateways* em localizações que assegurem desempenho da rede *wireless* dentro de parâmetros estabelecidos, suficientes para o atendimento de demandas de comunicação.

O problema de posicionamento de dispositivos em *smart grids* é aplicado a cenários que podem ter de milhares a centenas de milhares (ou mesmo milhões) de medidores inteligentes e postes a serem considerados. Por isso, outro grande desafio é o de estruturar a aplicação de forma que o resultado final de posicionamento seja alcançado no menor tempo de processamento possível, dando preferência a execuções que possam ser realizadas em questões de horas e não dias no caso de processamento de dados de uma grande cidade.

Além disso, deve-se considerar que as necessidades de planejamento não se limitam à proposição de topologia para novas regiões em que o *smart grid* será instalado, mas também para a atualização de eventual rede de comunicação pré-existente.

Finalmente, outro desafio enfrentado incluiu vencer as dificuldades impostas pela diversidade de cenários em que a aplicação final será utilizada, visto que diferentes regiões podem apresentar condições geográficas distintas e concentrações variadas de medidores

a serem conectados. Dessa forma, o método desenvolvido deve evitar o viés causado por regiões de alta densidade, comum em regiões urbanas altamente densas de moradias. Com isso, deve-se assegurar que o método consiga apresentar capacidade de classificação tanto em regiões rurais esparsas como em regiões urbanas densas.

1.6 CONTRIBUIÇÕES

As contribuições esperadas como resultado do desenvolvimento desta pesquisa incluem diferentes aspectos, descritos a seguir.

Inicialmente, pode-se citar o desenvolvimento de um método inovador para o posicionamento de roteadores e *gateways* em redes *wireless*, baseado em uma abordagem heurística e que utiliza um modelo de propagação que leva em consideração as perdas decorrentes do perfil topográfico de terreno existente entre medidores e posições de roteadores e *gateways*. Diferenciando-se de outros métodos encontrados na literatura, o método proposto utiliza abordagens que minimizam a quantidade de conexões a avaliar, pelo uso de árvore geradora mínima (*Minimum Spanning Tree, MST*), tornando-o aplicável a cenários de larga escala. Essa aplicabilidade a grandes cenários se dá pela capacidade de diminuir a complexidade de um método de posicionamento ao se reduzir a quantidade de posições candidatas a serem avaliadas. Trata-se de um método com dependência forte de processamento e complexidade computacionais.

Outra contribuição (e principal resultado buscado por esta pesquisa) está no desenvolvimento de um modelo de posicionamento que aprende com os resultados gerados por outro método (no caso, um método analítico). Para isso, a contribuição a ser destacada está no desenvolvimento de um método baseado em técnicas de *machine learning* que seja capaz de diminuir o espaço de posições candidatas a serem avaliadas e alcançar resultados comparáveis aos obtidos com o método analítico proposto, mas capaz de apresentar, também, ganhos significativos de processamento, tornando-o mais aderente à realidade de uma aplicação final.

Finalmente, um diferencial a ser destacado é o da realização de experimentos com dados de cenários reais de aplicação de *smart grids*, com características suficientes para avaliar o desempenho dos métodos, como com a existência de regiões com diferentes características geográficas e grande variação na quantidade e densidade de equipamentos a serem avaliados.

1.7 PUBLICAÇÕES

Durante o desenvolvimento da pesquisa para a elaboração desta tese de doutorado, duas publicações foram realizadas em jornais de relevância técnica e acadêmica. Essas publicações representam uma extensão significativa do trabalho apresentado nesta tese, fornecendo *insights* e contribuições importantes para o campo da pesquisa, em especial nas áreas de posicionamento de dispositivos e de inteligência artificial.

O primeiro artigo, intitulado "*Towards an Efficient Method for Large-Scale Wi-SUN-Enabled AMI Network Planning*" (MOCHINSKI et al., 2022), foi publicado no jornal MDPI Sensors, direcionado a publicações na área de ciência e tecnologia de sensores, e está acessível pelo link <https://doi.org/10.3390/s22239105>. Esse artigo apresenta o método analítico de posicionamento (AIDA) desenvolvido nesta pesquisa e compara suas características com outros métodos encontrados na literatura, destacando seus diferenciais. Experimentos são realizados com dados de cenários reais de larga-escala e com a avaliação do comportamento do método com dois modelos de propagação de sinal.

O segundo artigo, intitulado "*Developing an Intelligent Decision Support System for large-scale smart grid communication network planning*" (MOCHINSKI et al., 2024), foi publicado no jornal *Knowledge-Based Systems*, da Editora Elsevier, direcionado a pesquisas na área de inteligência artificial, e está acessível pelo link <https://doi.org/10.1016/j.knosys.2023.111159>. No artigo, o método baseado em *machine learning* (AIDA-ML) é apresentado como uma abordagem preliminar para o desenvolvimento de um *Intelligent Decision Support System* (IDSS, ou Sistema Inteligente de Suporte à Decisão) e foca na importância do processo de Engenharia de Características do cenário na definição de uma estratégia efetiva para o posicionamento de equipamentos de comunicação em *redes wireless* em *smart grids*.

Essas publicações demonstram o comprometimento em contribuir ativamente para o avanço do conhecimento na área de estudo desta pesquisa, enfatizando a relevância da pesquisa para o contexto acadêmico atual.

Através da integração de trabalhos publicados e com a pesquisa desenvolvida, espera-se enriquecer ainda mais o diálogo acadêmico e promover o avanço contínuo da área de estudo.

1.8 ESTRUTURA DA TESE

A presente tese está organizada em capítulos. Além do capítulo de Introdução, inclui os seguintes capítulos:

- Capítulo 2 – Fundamentação Teórica: visa contextualizar o leitor sobre conceitos

relevantes, apresentando seções sobre a arquitetura de rede de *smart grid* considerada pela pesquisa, o modelo de perda de percurso utilizado pelos métodos propostos e apresentados no trabalho, a descrição de características de métodos de *machine learning* e técnicas de otimização de modelos, entre outros conceitos úteis, para a melhor compreensão do texto.

- Capítulo 3 – Estado-da-Arte: apresenta os resultados de pesquisa bibliográfica efetuada para a identificação de aplicação de técnicas de *machine learning* em problemas de posicionamento de dispositivos em redes de comunicação sem fio.
- Capítulo 4 – Método AIDA: descreve as características de um método heurístico e resultados de experimentos efetuados para o posicionamento de roteadores/*gateways* com o uso dessa abordagem.
- Capítulo 5 – Método AIDA-ML: apresenta um método que utiliza uma abordagem baseada em *machine learning* para o posicionamento de roteadores/*gateways*.
- Capítulo 6 – Experimentos com o Método AIDA-ML e Resultados: descreve os experimentos e apresenta resultados obtidos com o uso de AIDA-ML para o posicionamento dos dispositivos de comunicação. Compara os resultados obtidos com essa abordagem em relação aos obtidos com o método analítico AIDA.
- Capítulo 7 – Conclusão: analisa as contribuições da pesquisa, faz uma revisão dos objetivos e hipóteses estabelecidos e, por fim, apresenta uma lista com oportunidades de trabalhos futuros que podem evoluir para outras contribuições científicas relacionadas ao tema da tese.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, inicialmente é apresentada a arquitetura de uma rede de comunicação de um *smart grid* com as características técnicas e restrições que foram consideradas para o desenvolvimento dos métodos de posicionamento propostos por este trabalho. Esse detalhamento é necessário dada a diversidade de tecnologias disponíveis e as várias implementações de redes elétricas inteligentes que podem ser encontradas no mercado.

Em seguida, o conceito de posições candidatas é explicado por sua importância para o processo de seleção de recursos considerados para a instalação de roteadores e *gateways*. O termo *posições candidatas* será utilizado com frequência ao longo do texto e, por isso, merece uma explicação neste capítulo.

O problema de posicionamento de roteadores e *gateways* que se busca resolver com este trabalho é amplamente discutido na literatura e caracterizado, entre diferentes possibilidades, como uma variação do *problema das p-Medianas*. Esse tema é explorado nesta seção para melhor caracterizar a forma clássica de conceituar esse tipo de problema.

Outro conceito apresentado nesta seção é referente ao modelo de perda de percurso do enlace. Para estabelecer a conexão entre medidores inteligentes e roteadores/*gateways* é necessário que a potência de sinal recebido no enlace seja suficiente para assegurar o processo de comunicação e transferência de pacotes de dados pela rede de comunicação. Por esse motivo, nesta seção são explicados alguns conceitos relevantes, os tipos de perdas considerados e o modelo de cálculo de potência recebida utilizado pelos métodos propostos pelo estudo.

Na sequência, o termo árvore geradora mínima é apresentado devido à sua relevância para o processo de avaliação de conexões entre medidores inteligentes explorado pelos métodos de posicionamento propostos por este estudo. O uso desse tipo de árvore visa minimizar a quantidade de conexões entre medidores a serem consideradas, e o uso dessa abordagem pode refletir em ganhos no processo de análise de cenários de larga escala e, por isso, a importância de apresentá-lo nesta seção.

Além desses conceitos, uma seção apresenta o funcionamento geral dos três algoritmos de *machine learning* utilizados nos experimentos realizados neste trabalho, incluindo os métodos: Regressão Logística, *Random Forest* e XGBoost.

Finalmente, conceitos básicos relacionados à otimização de modelos de *machine learning* são introduzidos para auxiliar no entendimento de técnicas que serão exploradas neste estudo.

2.1 ARQUITETURA DE UMA REDE DE COMUNICAÇÃO DE UM *SMART GRID*

Os autores em (AMIN; WOLLENBERG, 2005) introduzem o conceito de *smart grid* (SG), ou rede elétrica inteligente, apresentando diferentes características inerentes a um sistema dessa natureza. De acordo com os autores, os sistemas modernos de infraestrutura estão cada vez mais interconectados, podendo fazer com que uma falha em determinado ponto da rede venha a comprometer o funcionamento em outros pontos. Por esse motivo Amin e Wollenberg (2005) explicam que a substituição de um modelo de rede convencional por um *smart grid* possibilita que o controlador do sistema tenha ferramentas para monitorar e manter o sistema sob controle. Para os autores, o conceito de SG agrega inteligência a um sistema de transmissão de energia, ao sugerir o uso de um sistema distribuído composto por dispositivos/sensores inteligentes ou processadores independentes em todos os componentes da rede, incluindo subestações e usinas de energia. A capacidade de autorrecuperação também é uma característica destacada por Amin e Wollenberg (2005) para o conceito de *smart grid* que deve ser entendido, então, como um termo abrangente por envolver aspectos relativos a tecnologias de informação, monitoramento e controle, participação de mercado, regulação e planejamento.

O conceito de uma rede elétrica inteligente vai além da digitalização dos processos de controle e distribuição de energia, pois inclui, também, conforme explicado pelos autores em (VLASOV; ADAMOVA; SELIVANOV, 2021), a incorporação de conceitos como fontes de energia renováveis, eficiência energética e novos aspectos relacionados ao armazenamento e consumo de energia. Num *smart grid*, busca-se, também, um processo de distribuição de energia mais confiável, mais seguro e de mais fácil gerenciamento pela distribuidora e pelos próprios consumidores. Diferentemente do que usualmente ocorre numa rede elétrica tradicional, num SG a comunicação é bidirecional, permitindo fluxo de informações do centro de operação de distribuição às extremidades da rede (como instalações residenciais, indústrias e equipamentos de automação) e vice-versa.

Em (CECATI et al., 2010), os autores explicam que, num *smart grid*, os medidores de consumo de energia são denominados de medidores inteligentes (*smart meters*, SMs) e são componentes principais de uma estrutura denominada de *Advanced Metering Infrastructure* (AMI), ou infraestrutura de medição avançada, que inclui funções de monitoramento e controle de dispositivos e aparelhos. Num AMI, as tecnologias de comunicação mais viáveis de acordo com Cecati et al. (2010), incluem a comunicação sem fio (*wireless*) e a *Power Line Communication* (PLC), que utiliza a própria rede de energia elétrica para a transmissão de dados. É importante destacar que, na rede *backhaul* que faz a ligação de *gateways* e dispositivos de automação, além da comunicação sem fio, podem existir infraestruturas de comunicação disponíveis que utilizam cabos de fibra ótica como meio de transmissão.

Para o escopo deste trabalho, os dois principais componentes de uma rede de comunicação de um *smart grid* incluem a rede AMI e a rede de automação. Para ambas as redes, é importante estabelecer comunicação bidirecional com o centro de operação de distribuição para fins de aquisição e gerenciamento de dados. Usando uma arquitetura de comunicação sem fio padronizada, a rede AMI conecta *smart meters* (SMs), roteadores e *gateways*. A rede de automação é de missão crítica porque conecta os dispositivos de automação (DA, *Distribution Automation*) à infraestrutura de comunicação do *smart grid*, tornando-se altamente dependente do posicionamento correto dos equipamentos de comunicação.

2.1.1 Arquitetura Geral da Rede

Este estudo considera o uso do padrão de comunicação sem fio Wi-SUN (*Wireless Smart Utility Network*) (Wi-SUN Alliance® (Wi-SUN Alliance, —-)), que implementa uma arquitetura de rede *mesh* baseada no padrão IEEE 802.15.4g (IEEE SA - Standards Association, 2012), usando RPL (*IPv6 Routing Protocol for Low Power and Lossy Networks,* RFC6550) (ALEXANDER et al., 2012) como o protocolo de roteamento na camada de rede. A rede *mesh* permite a comunicação através de múltiplos saltos (*multihop*) entre medidores, roteadores e *gateways*. Nesse contexto, os medidores são capazes de efetuar o roteamento de pacotes, encaminhando mensagens entre os medidores no seu entorno e as posições de *gateways*.

Nesta tese, ao mencionarmos o termo roteador, nos referimos ao posicionamento de um tipo especial de equipamento denominado de extensor que, geralmente, pode ser compreendido como um equipamento que pode atuar como medidor inteligente, se necessário, e que apresenta maior potência de transmissão que um medidor comum. Os extensores devem ser posicionados de forma estratégica para garantir a comunicação em posições de mais difícil acesso ou com condição de comunicação mais restrita. Assim, de uma forma geral, ao nos referirmos ao posicionamento de roteadores/*gateways*, estamos nos referindo à identificação de posições para a instalação de extensores e *gateways* propriamente ditos.

O planejamento de rede envolve muitos elementos, incluindo medidores inteligentes, *gateways*, roteadores, postes e componentes da rede *backhaul*, e pode ser classificado como um problema NP-Hard. Além disso, um conjunto de restrições está associado ao projeto da rede, para garantir o máximo rendimento com a menor latência possível a um custo reduzido. O desempenho da rede é altamente dependente da instalação, em posições adequadas, de dispositivos para o processo de comunicação entre a rede NAN (*Neighborhood Area Network*, ou rede da área de vizinhança) do *smart grid*, os *gateways* e a rede WAN (*Wide Area Netowrk*, ou rede de longa distância), onde o Centro de Operação de Distribuição (*COD*) está instalado.

A Figura 2.a apresenta os elementos de interesse para o cenário de *smart grid* tratado por este estudo. A comunicação sem fio entre os elementos da região NAN ocorre de acordo com o protocolo de mensagens utilizado pela rede. Neste caso, considera-se a utilização do protocolo RPL, que permite a existência de diferentes rotas visando minimizar os pontos de falha pela possibilidade de utilizar nós pais alternativos (nós de *backup*) para o encaminhamento de mensagens. Analisando do ponto de vista de cada medidor inteligente, diferentes opções de comunicação são possíveis, seja conectando o medidor diretamente a um *gateway*, a um roteador, ou mesmo usando encaminhamento de mensagens com o uso de múltiplos saltos através de outros medidores.

Para o escopo desta tese, denomina-se de *"dispositivo de comunicação"* ou *"equipamento de comunicação"* um elemento da rede de comunicação com capacidade para atuar como concentrador e/ou fazer o encaminhamento de pacotes de dados, o controle de tráfego e/ou a interconexão de redes. O termo pode ser usado no texto como sinônimo para dispositivos dos tipos roteadores e *gateways*, em especial no que se referir ao processo de posicionamento de tais elementos de comunicação.

Os medidores inteligentes, roteadores, *gateways* e dispositivos DA estão na região NAN. Os dispositivos DA incluem reguladores de tensão e religadores automáticos, entre outros equipamentos. Todos esses elementos devem estar conectados para garantir a comunicação com a rede *backhaul*, que conecta os principais elementos de comunicação e estabelece um canal confiável de acesso bidirecional da NAN ao COD da rede inteligente.

Em um cenário típico de rede inteligente, o posicionamento correto dos equipamentos de comunicação (no caso, os *gateways* e roteadores) garante a comunicação entre um grande número de SMs (*smart meters*, medidores inteligentes) e o COD, e entre os dispositivos DA e o COD. Além disso, é fundamental destacar que, geralmente, esses elementos (medidores inteligentes e DAs) estão dispersos por uma grande área geográfica, trazendo complexidade ao planejamento do posicionamento.

Na prática, roteadores e *gateways* são, comumente, instalados em postes. A instalação geralmente é feita em áreas com grande concentração de medidores e equipamentos a serem conectados. Com isso, um conjunto de posições candidatas pode ser estabelecido a partir do conjunto de postes da região. Para o planejamento da rede AMI e posicionamento dos principais dispositivos, os métodos propostos neste estudo sugerem dar especial preferência ao uso de postes que hospedam DAs, pois geralmente esses dispositivos de automação já estão instalados na região, possuem conexão direta com a rede *backhaul* e são elementos importantes na infraestrutura da rede elétrica. Em alguns casos, os DAs já podem estar interligados via cabo de fibra ótica, reforçando o uso preferencial desses recursos. Considerando que, em especial, os *gateways* devem se conectar diretamente a elementos da rede *backhaul* (como os roteadores do *backhaul*), ao selecionar postes que possuam dispositivos já conectados a essa rede, favorece-se a criação de uma topologia que

exija menos elementos para atender a essa necessidade especial dos *gateways*. O número de postes com DAs é, geralmente, bem inferior à demanda de uma rede AMI; portanto, depois de priorizar o uso das posições com DAs, devem ser selecionados outros postes em posições adequadas para o processo de comunicação.

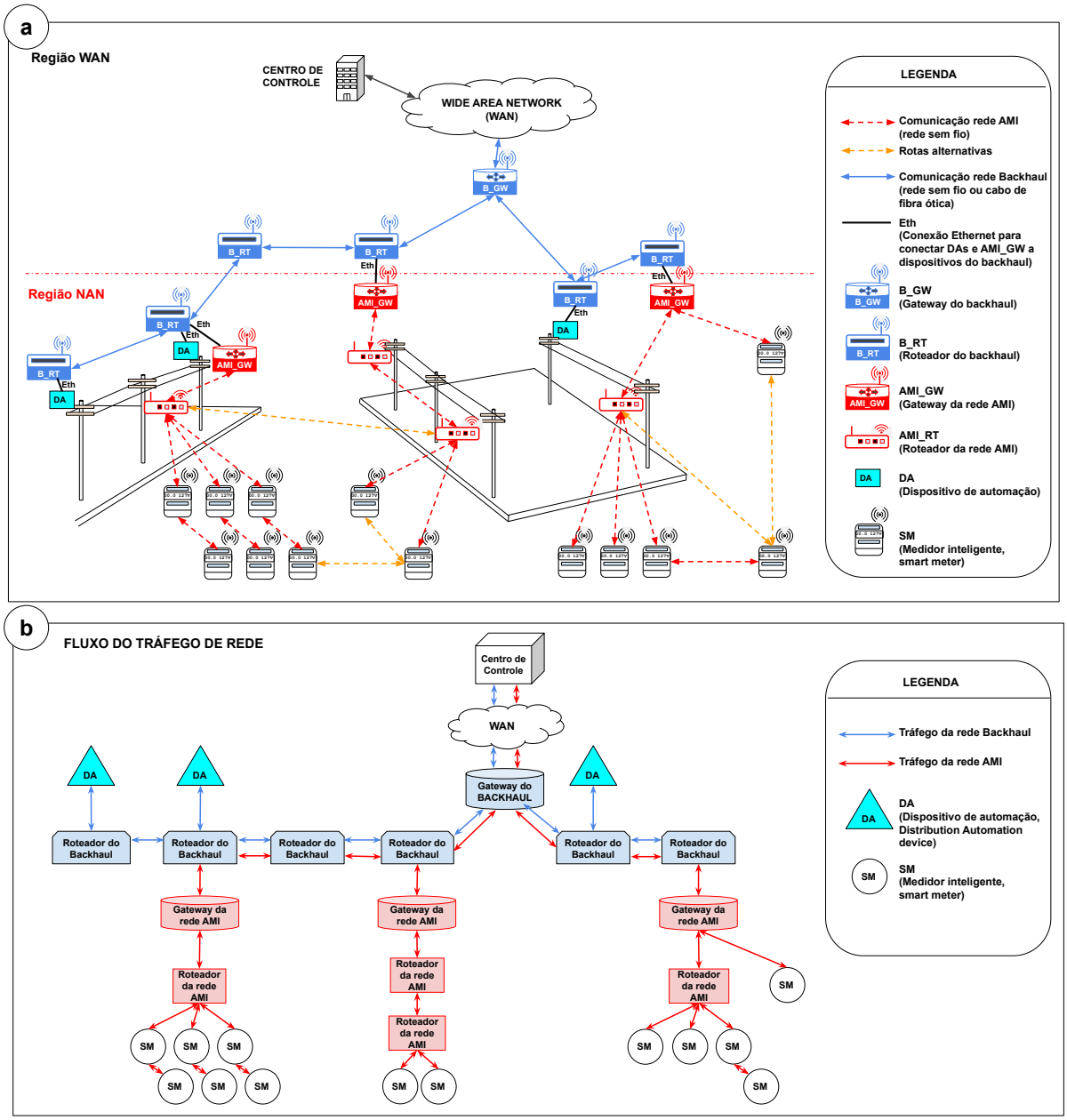


Figura 2 – Arquitetura de uma rede de comunicação de um *smart grid*. (**a**) Visão geral de um cenário de *smart grid*, destacando os principais elementos nas regiões NAN e WAN. (**b**) Diagrama dos fluxos de tráfego das redes *backhaul* e AMI para demonstrar os principais elementos na transferência de informações entre os dispositivos *end-point* e o centro de operação de distribuição.

Fonte: Adaptado de (MOCHINSKI et al., 2022).

2.1.2 Restrições de Planejamento de Rede

As restrições apresentadas nesta seção são baseadas em diretrizes para a implementação e operação do *smart grid* de uma grande empresa de energia elétrica no estado do Paraná, no sul do Brasil. A lista de restrições inclui:

- A estrutura do *smart grid* compreende duas redes sem fio (Figura 2.b): a rede *backhaul* e a rede sem fio AMI baseada na tecnologia Wi-SUN. Além disso, a rede *backhaul* é conectada a uma rede óptica (*backbone* WAN) nas subestações elétricas.

- A rede *wireless backhaul* é segmentada em três redes locais virtuais (VLANs) com diferentes prioridades de tráfego. A primeira VLAN serve para monitoramento de rádio e tem a prioridade mais alta. A segunda VLAN é para automação de equipamentos da rede de distribuição de energia e tem a segunda maior prioridade. Finalmente, a terceira VLAN é para o tráfego de comunicação AMI. Essa VLAN transporta o tráfego de dados de medição inteligente da rede Wi-SUN e tem a prioridade mais baixa. As redes de comunicação AMI e DA são separadas por VLANs em cada ponto de entroncamento com a rede física (subestação, estações VHF ou ramal), pois isso aumenta o nível de segurança da rede de comunicação como um todo.

- Os principais elementos de interesse na topologia de rede AMI para esta pesquisa incluem (i) medidores inteligentes, que medem o consumo de energia; (ii) roteadores AMI, com os quais os medidores se conectam e que são responsáveis pelo encaminhamento de mensagens pela rede; e (iii) *gateways* AMI que aceitam conexões de roteadores, bem como conexões diretas de medidores e, além de retransmitirem mensagens, servem como interface de comunicação entre a rede AMI e a rede *backhaul*.

- Em relação à rede *backhaul*, os elementos de interesse para este estudo são os roteadores do *backhaul*, com os quais se ligam os dispositivos DA e que também permitem a ligação de *gateways* AMI, e o *gateway backhaul*, que faz a interface entre a rede *backhaul* e a rede WAN para encaminhar mensagens de/para o centro de operação de distribuição. Neste estudo, ao nos referirmos a roteadores e *gateways*, estamos nos referindo de forma simplificada a roteadores e *gateways* da rede AMI.

- A infra-estrutura da rede de comunicação de automação e medição é baseada na existência de postes, como ocorre em diversas empresas no mundo. A vantagem da utilização dos postes se baseia no fato de fazerem parte dos ativos da empresa, minimizando a necessidade de contratação de infraestrutura de terceiros. Além disso, os postes oferecem a tensão de alimentação necessária para a configuração e operação dos dispositivos de comunicação e apresentam uma altura favorável para o posicionamento dos roteadores e *gateways*.

- Os DAs são instalados em postes e principalmente na rede elétrica aérea. As redes subterrâneas de energia estão restritas a pequenas áreas no Brasil, em alguns centros urbanos, e são tratadas como exceção (fora do escopo deste estudo). A comunicação com o equipamento DA usa o protocolo de rede distribuída (DNP3) sobre IP usando *pooling*, sem fazer uso de mensagens não solicitadas (mensagens enviadas automaticamente pelos dispositivos ao servidor) devido a uma limitação do sistema de Supervisão e Aquisição de Dados (*Supervisory Control and Data Acquisition*, SCADA).

- O gerenciamento do fluxo de informações dos *end-points* para o centro de operação de distribuição considera que os dados dos elementos da rede AMI (por exemplo, medidores inteligentes) e os DAs compartilharão a infraestrutura física da rede *backhaul*. No entanto, as informações fluem por diferentes VLANs e com diferentes prioridades, explicadas a seguir: (i) As informações de monitoramento de consumo de energia e tensão dos medidores geralmente são obtidas na rede AMI por meio de um mecanismo de *pooling* gerenciado pelo centro de operação de distribuição, que usa um algoritmo para fazer um *pooling* programado para distribuir a leitura ao longo do dia e evitar congestionamentos. Esse algoritmo, em geral, pode controlar a leitura espacialmente (estabelecendo diferentes regiões para a leitura) e temporalmente (para realizar a leitura de diferentes áreas em diferentes períodos). Um exemplo de um algoritmo de leitura de medidor inteligente programado é apresentado pelos autores em (KEMAL; OLSEN; SCHWEFEL, 2018); (ii) Quanto aos DAs (integrantes da rede *backhaul*), eles são considerados dispositivos de alta prioridade; assim, seu status é lido com mais frequência (leitura de alta frequência), pois o centro de operação de distribuição os monitora continuamente e age sobre eles com a rapidez necessária. Apesar dessa leitura de alta frequência, é fundamental destacar que o número de DAs em uma rede inteligente é consideravelmente inferior ao número de medidores inteligentes. Assim, seu tráfego representa uma alta frequência de leituras, mas para uma relativa pequena quantidade de dispositivos.

- Finalmente, a função de rede AMI não se restringe a medição e faturamento. Ela deve oferecer suporte a uma comunicação bidirecional que permita que um controle remoto desligue/religue a energia da casa dos consumidores — além de suportar alarmes de *last-gasp* (último suspiro) informando a falta de energia nas casas dos consumidores, facilitando a identificação de trechos defeituosos e a coordenação de equipes de manutenção com maior assertividade. A rede de comunicação dos DAs (*wireless backhaul network*) é provida por um sistema de *backup* de energia (baterias) para permitir manobras mesmo durante paradas de fornecimento de energia.

2.2 POSIÇÕES CANDIDATAS

Para este estudo, o termo posições candidatas (*candidate positions*, CPs) se refere às posições dos postes da rede de distribuição de energia elétrica, incluindo os postes com equipamentos de automação a eles anexados. As posições são chamadas de candidatas porque sua promoção a uma posição de roteador ou *gateway* dependerá da existência de SMs conectados à sua posição ao final do processo de planejamento.

Os métodos de posicionamento propostos por este estudo, inicialmente priorizam a seleção de postes com dispositivos DA, pois esses equipamentos precisam estar conectados diretamente à rede de comunicação do *backhaul*. Depois disso, postes comuns são considerados para conectar os medidores inteligentes restantes.

A utilização de postes para a instalação de equipamentos se justifica por esses elementos fazerem parte do rol de ativos da distribuidora de energia e podem ser facilmente configurados para atender aos requisitos técnicos de instalação de roteadores e *gateways*.

Considerando a irregularidade do terreno das regiões e o elevado número de postes existentes em cada cidade, os métodos desenvolvidos neste estudo utilizam uma abordagem de *grid* (grade) para fazer uma seleção otimizada de um subconjunto de postes e minimizar o esforço computacional necessário para escolher as coordenadas mais adequadas para o posicionamento dos dispositivos de comunicação. Mais detalhes sobre este processo são apresentados na Seção 4.2.3.

2.3 PROBLEMA DAS p-MEDIANAS

A atividade de posicionamento de *gateways* consiste em estabelecer as melhores posições para a instalação de tais equipamentos num cenário de rede de comunicação. As tecnologias de rede, os equipamentos e as aplicações envolvidas podem ser as mais diversas possíveis. As tecnologias de rede podem incluir redes WMN (*Wireless Mesh Network*), redes 5G, LoRaWAN, redes WSN (*Wireless Sensors Networks*), redes SDN (*Software-Defined Network*), Wi-Fi, Wi-SUN, entre outras. Os equipamentos envolvidos podem assumir diferentes nomenclaturas (particulares de suas funções no cenário em que são aplicados) como roteadores, *gateways*, controladores, *switches*, coordenadores, concentradores, e podem ser utilizados em aplicações como comunicação celular 5G, *smart grid*, redes de sensores IoT, entre outros. De uma forma geral, apesar dessa diversidade, é comum que esse problema seja modelado sob a denominação de um problema das p-Medianas.

O problema das p-Medianas, abordado inicialmente por (HAKIMI, 1964; HAKIMI, 1965) para o posicionamento de *switching centers* em redes de telefonia, é amplamente aplicado a problemas de posicionamento de recursos, instalações (*facilities*) e dispositivos.

Pode ser estendido/adaptado e encontrado sob a denominação de *Facility Location problems*, *Controller Placement problems* (CPP), dentre outros nomes, e tem aplicabilidade em um universo amplo de situações como na identificação da melhor posição para a construção ou posicionamento de instalações numa rede de distribuição, ou estabelecer qual a melhor posição para a instalação de escolas numa cidade, hospitais, etc. Considere o cenário de exemplo em que uma rede de lojas de departamento possui diversas filiais espalhadas pelo país e precisa estabelecer qual a melhor localidade para a instalação de um centro de distribuição. Pode-se dizer que a melhor posição para o centro de distribuição deve ser aquela que assegure que a rede conseguirá atender à demanda dos clientes de forma satisfatória. Do ponto de vista da empresa, espera-se que o posicionamento assegure ganhos para a empresa, tal como mais agilidade no processo de distribuição, a redução de custos de transporte, ou mesmo a possibilidade de expandir o volume de vendas pela abertura de novas frentes de comercialização como, por exemplo, *e-commerce*.

Aplicado ao cenário de redes de comunicação, o problema das p-Medianas pode ser utilizado de forma a identificar qual a melhor posição para a instalação de *gateways* ou concentradores de forma a assegurar maior *throughput*, menor latência, economia de gastos de instalação pela minimização do número de dispositivos a serem instalados, ou para atender a outros requisitos estabelecidos de acordo com as necessidades de cada projeto. A Figura 3 apresenta um cenário simplificado de um *smart grid* em que é necessário posicionar roteadores/*gateways* para a comunicação entre medidores inteligentes e o centro de operação de distribuição da companhia de energia. É possível observar que, apesar de existirem vários postes candidatos para a instalação de roteadores/*gateways*, apenas alguns serão selecionados para a conexão dos medidores inteligentes. A quantidade depende de diferentes fatores, como o orçamento disponível, a potência de transmissão dos medidores, a localização dos postes candidatos, o volume de informações a serem trafegadas, a quantidade de medidores, entre outros.

De acordo com (MARIANOV; SERRA, 2009) o problema é denominado de problema das p-Medianas porque, numa rede ou grafo, o vértice mediano é o vértice para o qual a soma dos comprimentos dos caminhos mais curtos para todos os outros vértices é a menor. Em relação à complexidade, o problema das p-Medianas é considerado um problema NP-Hard mesmo para redes com estrutura simples (exemplo: grafo planar com grau máximo de vértice 3), dado o grande número de posições possíveis disponíveis para a instalação dos equipamentos. Por isso, a busca por diferentes abordagens para a sua resolução é frequente na literatura. Sobre isso, os autores em (LORENA et al., 2001) citam referências que abordam sobre o uso de vários métodos heurísticos e métodos que exploram busca em árvore, bem como o uso combinado de técnicas de relaxação lagrangeana e otimização por subgradientes de um ponto de vista primal-dual para o problema das p-Medianas.

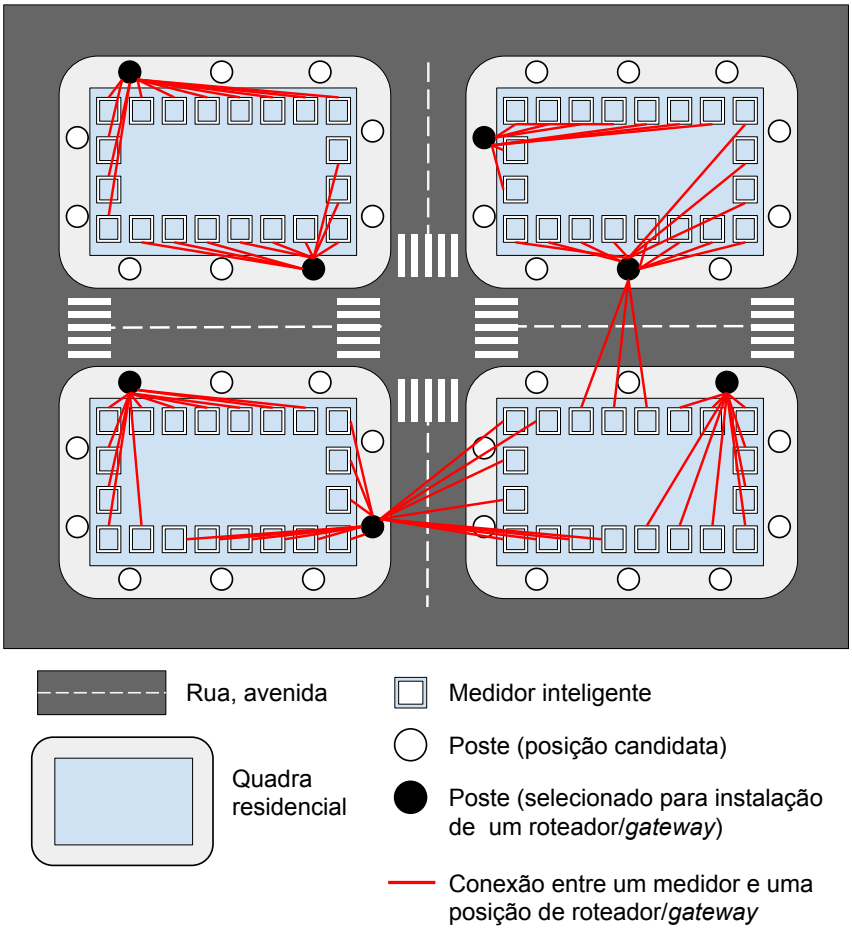


Figura 3 – Cenário-exemplo de aplicação do problema das p-Medianas para o posicionamento de roteadores/*gateways* no contexto de um *smart grid*.

Fonte: Autoria própria.

Sob o ponto de vista formal, o problema das p-Medianas pode ser modelado matematicamente como um problema de programação linear inteira (ILP, *Integer Linear Programming*). Em (BELTRAN; TADONKI; VIAL, 2006) os autores detalham que no problema das p-Medianas o objetivo é selecionar os locais para o posicionamento de p instalações a partir de um conjunto de m posições candidatas para atender a um conjunto de n clientes e associar cada cliente a uma única instalação. De acordo com os autores, o custo dessa associação corresponde à soma das distâncias mais curtas c_{ij} de um cliente a uma instalação. Tal distância pode, às vezes, ser ponderada por um fator apropriado para representar, por exemplo, a demanda em um nó cliente. O objetivo do problema das p-Medianas é minimizar essa soma e, de acordo com Beltran, Tadonki e Vial (2006), pode ser formulado conforme indicado na Equação (2.1):

$$z^* = \min_{x,y} \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{2.1}$$

sujeito a:

$$\sum_{i=1}^{m} x_{ij} = 1, \quad \forall j,$$
$$\sum_{i=1}^{m} y_i = p,$$
$$x_{ij} \leq y_i, \quad \forall i, j,$$
$$x_{ij}, y_i \in \{0, 1\}$$

onde m corresponde ao total de posições candidatas para o posicionamento de instalações, p corresponde ao total de instalações a serem posicionadas, n corresponde ao total de clientes a serem atendidos pelas instalações que serão posicionadas, c_{ij} corresponde à distância entre a posição do cliente j à posição da instalação i, $x_{ij} = 1$ se a instalação i serve o cliente j, senão $x_{ij} = 0$ e $y_i = 1$ se uma instalação for posicionada na posição candidata i, senão $y_i = 0$.

2.4 MODELO DE PERDA DE PERCURSO DO ENLACE

Nesta seção, é descrito o modelo de cálculo de perda de percurso do enlace utilizado para o cálculo de potência recebida estimada no enlace (*link received power*, LRP). Neste trabalho, o valor de LRP é usado como métrica para estabelecer a conexão entre um medidor inteligente e uma posição candidata e sua descrição foi adaptada de (MOCHINSKI et al., 2022).

O LRP é calculado com base na potência de transmissão, ganhos de antena e modelo de perda de percurso (*path loss*, PL) do canal de rádio.

Para estimar a perda de potência do enlace (*link power loss*, LPL), considera-se a perda de percurso do enlace e a perda por difração (*diffraction loss*, DL). O LPL considera o perfil detalhado do terreno, que é construído da seguinte forma: inicialmente, são identificadas as coordenadas e o comprimento do trajeto entre o medidor inteligente e a CP. O caminho é dividido em 100 pontos equidistantes, obtendo-se a elevação do terreno em cada ponto. Com as medidas de elevação e a posição de cada ponto, obtém-se o perfil detalhado do terreno.

A utilização de uma análise detalhada do perfil do terreno evita (ou minimiza) a necessidade de classificação empírica do terreno, pois é muito difícil (ou impreciso) definir se, para uma região específica, o terreno é, por exemplo, totalmente acidentado ou plano, ou apenas 50% montanhoso com densidade de árvores leves ou pesadas, e assim por diante.

A *International Telecommunication Union* (ITU)[1] apresenta em suas recomendações

[1] <https://www.itu.int/> (acessado em 8 de maio de 2022)

os modelos para determinar as perdas por difração de enlaces de rádio. O método *Delta-Bullington* é apresentado na ITU-R P.526-13 (International Telecommunication Union, 2013) e tem como objetivo determinar a difração de um enlace de rádio considerando os múltiplos obstáculos determinados pelo perfil do terreno entre as coordenadas dos dispositivos transmissor e receptor. A perda por difração do método *Bullington* para o caminho é dada pela Equação (2.2):

$$L_b^{\mathrm{dB}} = L_{uc}^{\mathrm{dB}} + (1 - e^{-L_{uc}^{\mathrm{dB}}/6}) \cdot (10 + 0,02 \cdot d) \tag{2.2}$$

onde L_{uc}^{dB} é a *knife-edge loss* (perda de gume de faca) para o ponto *Bullington*, e d é a distância (em km) entre o transmissor e o receptor. O modelo inclui três tipos de perda por difração (DL) (mais detalhes em (International Telecommunication Union, 2013)):

- Bullington DL para o perfil real do percurso (*actual path profile*) (L_{ba}^{dB}): Para o cálculo de L_{ba}^{dB}, aplica-se o método de *Bullington* através da Equação (2.2) considerando o perfil real do terreno com todas as suas elevações. O obstáculo que causa a maior difração é considerado para o cálculo.
- Bullington DL para um perfil suavizado do percurso (*smooth path profile*) (L_{bs}^{dB}): Esta perda de difração considera um terreno sem elevações. Para o cálculo de L_{bs}^{dB}, aplica-se o método de *Bullington* utilizando a Equação (2.2) considerando um obstáculo equivalente com alturas equivalentes das antenas transmissora e receptora.
- Perda por difração da Terra esférica (*spherical-Earth diffraction loss*) (L_{sph}^{dB}): Essa perda por difração leva em consideração a curvatura da Terra e é calculada como a perda por difração interpolada, dada pela Equação (2.3):

  $$L_{sph}^{dB} = [1 - h/h_{req}] \cdot A_h \tag{2.3}$$

  onde h é a menor altura livre entre o caminho curvo da Terra e o raio entre as antenas, h_{req} é o espaço livre necessário para perda por difração ser igual a zero e A_h é a perda por difração para o percurso usando o raio da Terra modificado. Se A_h for negativo, a perda por difração para o caminho é zero e nenhum cálculo adicional é necessário.

A perda por difração do enlace (*link diffraction loss*, LDL) para o percurso é:

$$LDL\ (\mathrm{dB}) = L_{ba}^{\mathrm{dB}} + \max(L_{sph}^{\mathrm{dB}} - L_{bs}^{\mathrm{dB}}, 0). \tag{2.4}$$

A perda de percurso corresponde à redução da densidade de potência de uma onda de rádio à medida que ela se propaga pelo canal (WU et al., 2020). Essa atenuação de

sinal geralmente é o resultado de fenômenos físicos de propagação, como reflexão, refração, difração e espalhamento (POPOOLA et al., 2019). Considerando a frequência de operação dos rádios, o fenômeno de difração é particularmente relevante para estimativas precisas de perda de percurso. Para o método proposto neste estudo, PL é calculado considerando o modelo *log-distance path loss* definido na Equação (2.5), onde λ é o comprimento de onda, e d é a distância entre transmissor e receptor, em metros. O parâmetro β é o expoente de perda do percurso e d_0 é a distância de referência. O modelo *log-distance path loss* estabelece que a perda de potência é diretamente proporcional ao logaritmo da distância entre transmissor e receptor e implica que, quanto maior a distância, maior a perda de potência.

$$PL\ (\text{dB}) = 10 \cdot \log_{10}\left[\left(\frac{4\pi d_0}{\lambda}\right)^2\right] + 10 \cdot \beta \cdot \log_{10}\left(\frac{d}{d_0}\right) \tag{2.5}$$

A perda de potência no enlace é dada pela Equação (2.6).

$$LPL\ (\text{dB}) = PL\ (\text{dB}) + LDL\ (\text{dB}) \tag{2.6}$$

Por fim, a potência recebida estimada no enlace (LRP) entre um medidor inteligente (SM) e um roteador (RT) e/ou *gateway* (GW) é calculada usando (2.7):

$$\begin{aligned} LRP\ (\text{dBm}) = & P_{tx}^{SM}\ (\text{dBm}) + G_{tx}^{SM}\ (\text{dBi}) \\ & + G_{rx}^{RT/GW}\ (\text{dBi}) - LPL\ (\text{dB}) \end{aligned} \tag{2.7}$$

onde P_{tx}^{SM} é a potência de transmissão do medidor SM, G_{tx}^{SM} é o ganho da antena do SM (*smart meter*, medidor inteligente) e $G_{rx}^{RT/GW}$ é o ganho da antena do roteador/*gateway*. A Equação (2.7) também é válida para calcular o LRP para o enlace entre dois medidores inteligentes.

2.5 ÁRVORE GERADORA MÍNIMA

Um grafo $G = (V, E)$ é composto por um conjunto de vértices V e arestas (*edges*) E, conforme ilustrado na Figura 4. A representação matemática, computacional ou gráfica da relação entre vértices e arestas de um grafo permite identificar a interdependência entre os elementos que o compõem. Os grafos possuem aplicações em diferentes áreas como no mapeamento das relações entre os participantes de uma rede social, na representação de redes de transporte, no processamento de linguagem natural, entre outras. Para o interesse deste estudo, o grafo será utilizado para representar a relação entre medidores inteligentes num *smart grid*.

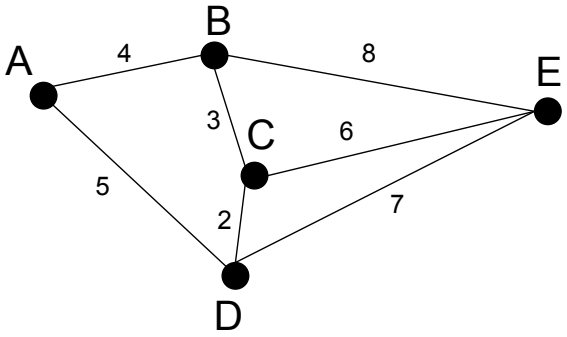


Figura 4 – Exemplo de grafo com 5 vértices e 7 arestas.

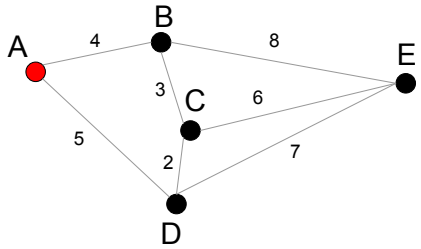
Fonte: Autoria própria.

Uma árvore geradora mínima (MST) corresponde a uma árvore formada pela seleção de arestas de um grafo de forma que o caminho resultante da soma dos pesos (custos ou dimensões) das arestas seja o menor, que conecte todos os vértices e que não gere ciclos. Diferentes algoritmos podem ser utilizados para a geração de uma MST, entre eles: algoritmo de *Prim* (PRIM, 1957), algoritmo de *Kruskal* (KRUSKAL, 1956), o algoritmo de tempo linear randomizado proposto por (KARGER; KLEIN; TARJAN, 1995), o algoritmo *Boruvka* de árvore dupla (MARCH; RAM; GRAY, 2010), entre outros.

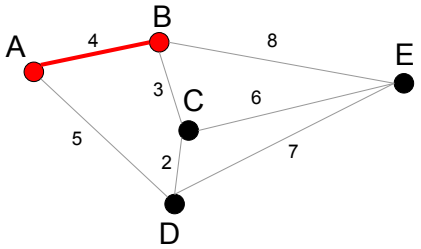
Os algoritmos diferem entre si pela abordagem como efetuam a geração da árvore. Mas, de forma geral, todos buscam formar uma árvore de menor custo que conecte todos vértices sem a formação de ciclos. O algoritmo *Kruskal*, por exemplo, para um determinado grafo G, estrutura a árvore geradora de forma iterativa a partir da escolha e adição à arvore da aresta disponível que apresente o menor peso (menor custo ou menor comprimento) sem formar nenhum ciclo com as arestas já escolhidas. Com isso, diferentes subárvores vão se formando até que, ao final, todas as arestas pertençam à mesma árvore. O algoritmo *Prim*, por sua vez, inicia a construção da árvore pela seleção aleatória de um vértice. A partir desse vértice, avalia os pesos (custo ou comprimento) das arestas conectadas a esse vértice e seleciona como próximo vértice da árvore aquele com o qual a aresta de ligação possui menor custo. Também de forma iterativa, o algoritmo repete esse processo (analisando vértices já pertencentes à árvore e os pesos das arestas a eles conectados) até que todos os vértices pertençam à arvore geradora mínima. A Figura 5 ilustra a construção de uma MST para o grafo da Figura 4 usando o algoritmo *Prim*.

2.6 ALGORITMOS DE *MACHINE LEARNING*

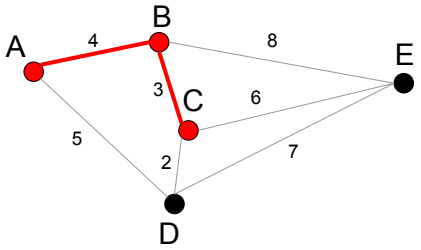
Nesta seção são descritas as características principais dos algoritmos considerados nos experimentos desenvolvidos para este estudo, em especial na Seção 6 que apresentam resultados com uso dos métodos de *machine learning* no método AIDA-ML.



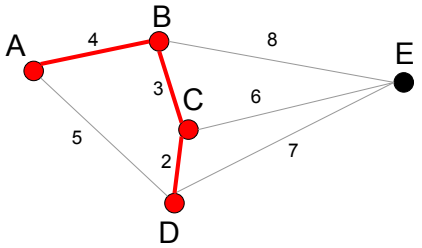
Iteração 1 - Um vértice é selecionado aleatoriamente, no caso o vértice A. Esse vértice passa a pertencer ao conjunto dos vértices que comporão a MST. Os demais ficam reservados para as próximas iterações. São analisadas as arestas de A para identificar qual a que possui menor peso (no caso, a aresta {A,B} com valor 4).



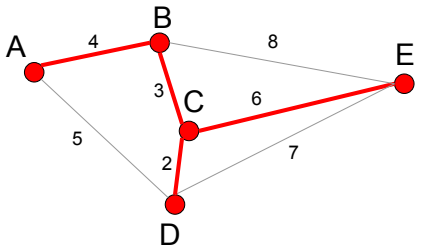
Iteração 2 - O vértice B passa a pertencer ao conjunto dos vértices que comporão a MST juntamente com o vértice A. Os demais ficam reservados para as próximas iterações. São analisadas as arestas disponíveis de A ({A,D}) e B ({B,C} e {B,E}) para identificar qual a que possui menor peso (no caso, a aresta {B,C} com valor 3).



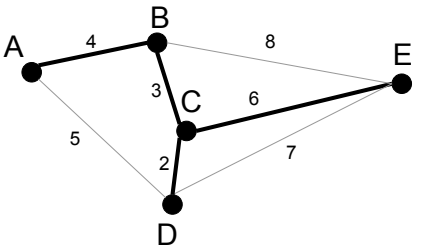
Iteração 3 - O vértice C passa a pertencer ao conjunto dos vértices que comporão a MST juntamente com os vértices A e B. Os demais ficam reservados para as próximos iterações. São analisadas as arestas disponíveis de A, B e C para identificar qual a que possui menor peso e que, se escolhida não forma ciclo (no caso, a aresta {C,D} com valor 2).



Iteração 4 - O vértice D passa a pertencer ao conjunto dos vértices que comporão a MST juntamente com os vértices A, B e C. Os demais ficam reservados para as próximos iterações. São analisadas as arestas disponíveis de A, B, C e D para identificar qual a que possui menor peso e que, se escolhida não forma ciclo (no caso, a aresta {C,E} com valor 6).



Iteração 5 - O vértice E passa a pertencer ao conjunto dos vértices que comporão a MST juntamente com os vértices A, B, C e D. Com isso, todos vértices já compõem a árvore mínima.



Árvore geradora mínima (MST) construída com o algoritmo Prim.

Figura 5 – Exemplo de execução do algoritmo *Prim* para a construção da MST.

Fonte: Autoria própria.

O autor em (MITCHELL, 1997) define *machine learning* como a área de conhecimento cujo objetivo consiste em desenvolver sistemas que aprendem com experiências passadas e melhoram seu desempenho à medida que novas experiências são fornecidas a ele. Sob um ponto de vista formal, Mitchell (1997) define que um programa de compu-

tador aprende com a experiência E em relação a alguma classe de tarefas T e medida de desempenho P, se seu desempenho em tarefas em T, medido por P, melhora com a experiência E.

Para o escopo desta pesquisa, o foco será dado no uso da aprendizagem supervisionada, que pode ser usada para problemas de classificação ou regressão, e que utiliza um conjunto de dados de treinamento rotulado para modelar a relação entre as variáveis de entrada e prever uma variável de saída, denominada de classe ou *target*, e que corresponde ao resultado do processo de classificação.

Os algoritmos descritos nesta seção incluem os métodos de Regressão Logística (*Logistic Regression*, LR), *Random Forest* (RF) e *XGBoost* (XGB), selecionados por utilizarem abordagens diferentes para o processo de classificação e por serem amplamente utilizados na literatura.

2.6.1 Regressão Logística (LR)

A regressão logística é um modelo estatístico originalmente proposto pelo autor em (COX, 1958), aplicável a problemas de classificação binária e para a estimativa de probabilidade de classe. Os autores em (BARTOSIK; WHITTINGHAM, 2021) explicam que a regressão logística utiliza uma combinação linear de recursos e aplica a eles uma função sigmoide não linear para redução dos erros residuais. Além disso, explicam que a regressão logística depende do conceito de chance do evento, que é a probabilidade de um evento ocorrer dividida pela probabilidade de não ocorrer. De maneira similar ao que ocorre com a regressão linear, a regressão logística utiliza pesos (coeficientes) associados às variáveis de entrada. Para a regressão logística, a relação entre os pesos e o resultado do modelo é exponencial, não linear. De acordo com os autores em (MOLNAR, 2022), o modelo de regressão logística usa a função logística para assegurar que a saída de uma equação linear esteja no intervalo entre 0 e 1. A função logística é definida pela Equação (2.8):

$$\text{logistic}(x) = \frac{1}{1 + e^{-x}} \tag{2.8}$$

A Figura 6 ilustra o gráfico da função logística definida pela Equação (2.8), para o intervalo de -6 a 6,

Os autores (MOLNAR, 2022) explicam que num modelo de regressão linear, a relação entre o resultado e os recursos são modelados com uma equação linear, tal como indicado na Equação (2.9),

$$\hat{y}^{(i)} = \beta_0 + \beta_1 {x_1}^{(i)} + \cdots + \beta_p {x_p}^{(i)} \tag{2.9}$$

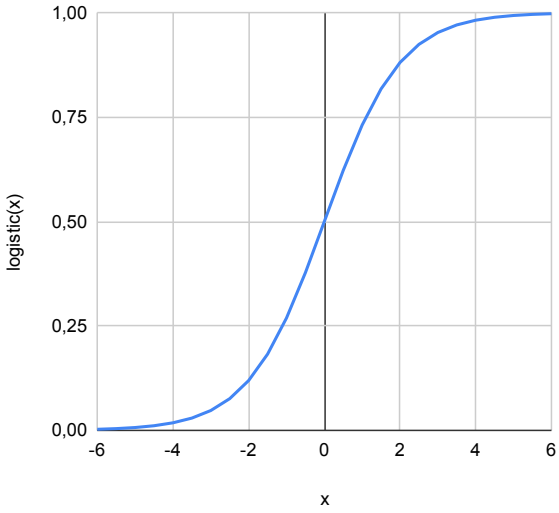


Figura 6 – Gráfico da função logística.

Fonte: Adaptado de (MOLNAR, 2022).

onde $\hat{y}^{(i)}$ corresponde ao valor previsto de uma instância i e equivale a uma soma ponderada de seus p recursos. Os parâmetros β_1 a β_p representam os pesos ou coeficientes das características e o primeiro peso na soma (β_0) é chamado de intercepto. Ele equivale ao ponto de interseção da reta de regressão no eixo vertical.

Para a classificação, os autores (MOLNAR, 2022) explicam que as probabilidades de classe devem variar entre 0 e 1. Para conseguir isso, então, a expressão (2.9) é envolvida na função logística, tal como indicado na Equação (2.10),

$$P(y^{(i)} = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 {x_1}^{(i)} + \cdots + \beta_p {x_p}^{(i)})}} \tag{2.10}$$

onde $P(y^{(i)} = 1)$ corresponde à probabilidade da classe ser igual a 1.

A Figura 7, adaptada de (MOLNAR, 2022) apresenta um exemplo de aplicação da regressão logística. Nesse exemplo, amostras de tumores são classificadas com a função logística quanto à sua malignidade conforme o tamanho que apresentam. Com a regressão logística, a classificação qualifica a observação de forma categórica, indicando se se trata de um tumor maligno ou benigno dentro de certa faixa de probabilidade.

De acordo com (GUDIVADA et al., 2016), os coeficientes do modelo de regressão logística equivalem, aproximadamente, aos pesos das características, de forma a mapear os pesos de cada recurso para um valor entre 0 e 1 por meio da função logística em forma de S. Esse valor pode ser interpretado como a probabilidade de uma instância pertencer a uma determinada classe. Para a classificação, o algoritmo de aprendizagem ajusta os pesos para classificar corretamente os dados de treinamento. O ajuste de pesos deve levar em conta a preocupação para minimizar a possibilidade de *overfitting*. Para isso, métodos

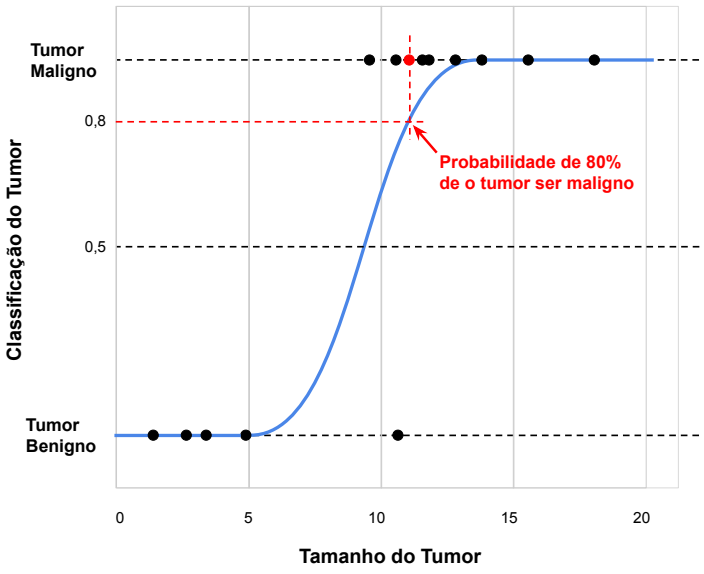


Figura 7 – Exemplo de aplicação de regressão logística.

Fonte: Adaptado de (MOLNAR, 2022).

como o método de descida do gradiente e outras variantes são opções utilizadas. Uma vez escolhidos os pesos, a função logística é aplicada a qualquer exemplo não visto para obter a probabilidade de pertencer a uma classe.

Ainda de acordo com (GUDIVADA et al., 2016), a regressão logística é um método simples e, popularmente, usado como um algoritmo inicial para problemas de classificação. O método é pouco propenso a problemas de *overfitting* e o uso da descida de gradiente para o ajuste de parâmetros torna rápido o processo de treinamento do modelo.

2.6.2 Random Forest (RF)

O algoritmo *Random Forest* (floresta aleatória) é um método de aprendizagem supervisionada que, de acordo com o autor em (BREIMAN, 2001), faz uso de uma combinação de classificadores do tipo árvore de decisão em que cada árvore depende dos valores de um vetor aleatório amostrado independentemente com a mesma distribuição para todas as árvores que compõem a floresta. Os autores em (CUTLER; CUTLER; STEVENS, 2011) explicam que uma floresta aleatória (*random forest*) é uma combinação (um *ensemble*) baseado em árvore em que cada árvore depende de uma coleção de variáveis aleatórias.

O *Random Forest* pode ser utilizado em problemas de regressão e de classificação. Para o treinamento, o algoritmo RF aplica a técnica de *bootstrapping*, criando J modelos, sendo um para cada árvore criada pelo método, a partir de amostras com reposição de dados extraídas aleatoriamente do *dataset* de treinamento. O RF aplica, também, o que é

denominado de *feature bagging*, fazendo a seleção aleatória de atributos ou características consideradas no treinamento. Após a criação de J modelos, o resultado do processo de predição para uma nova instância x (não considerada no treinamento), será dado pela Equação (2.11) (para problemas de regressão) e pela Equação (2.12) (para problemas de classificação) (CUTLER; CUTLER; STEVENS, 2011), que fazem a agregação dos resultados gerados pelos diferentes modelos. Em resumo, o algoritmo RF usa o conceito de *bagging* (*bootstrapping + aggregation*) que corresponde ao uso de vários modelos para o treinamento e a agregação das predições geradas por cada árvore da floresta.

$$\hat{f}(x) = \frac{1}{J}\sum_{j=1}^{J}\hat{h}_j(x) \tag{2.11}$$

$$\hat{f}(x) = arg\,max_y \sum_{j=1}^{J} I(\hat{h}_j(x) = y) \tag{2.12}$$

onde $\hat{h}_j(x)$ é a previsão da variável de resposta da instância x usando a árvore de índice j.

Para problemas de classificação, que são o principal interesse deste estudo, o *Random Forest* usa o método de votação majoritária (*majority voting*) para definir a classe que será atribuída a uma determinada instância submetida ao processo de classificação. Com esse método, o resultado da predição corresponderá ao resultado que ocorrer mais vezes no total de árvores que foram geradas no processo de treinamento do modelo.

A Figura 8 ilustra, de forma simplificada, o funcionamento do método RF, em especial quanto ao processo de definição do valor do atributo *target* ao realizar a classificação.

Para o treinamento do método *Random Forest*, diferentes hiperparâmetros podem ser ajustados, como, por exemplo: o número de árvores da floresta, a função a ser usada para medir a qualidade da divisão de um nó (*split*), a profundidade máxima das árvores, o número mínimo de amostras exigidas para dividir um nó interno, bem como o número mínimo de amostras para um nó folha, entre outros.

2.6.3 XGBoost (XGB)

XGBoost (acrônimo para *eXtreme Gradient Boosting*), (CHEN; GUESTRIN, 2016), é um algoritmo de *machine learning* baseado em árvores de decisão que utiliza um processo de *boosting* para a construção do modelo de aprendizagem. Nesse processo, o modelo é construído de forma sequencial, com a incorporação de modelos simples (no caso, modelos baseados em árvores de decisão) de forma incremental com o objetivo de corrigir erros de predição observados na iteração anterior e alcançar um modelo final mais assertivo. O modelo é ajustado para corrigir erros do modelo anterior usando gradientes de uma função de perda. Por isso a denominação de *gradient boosting*.

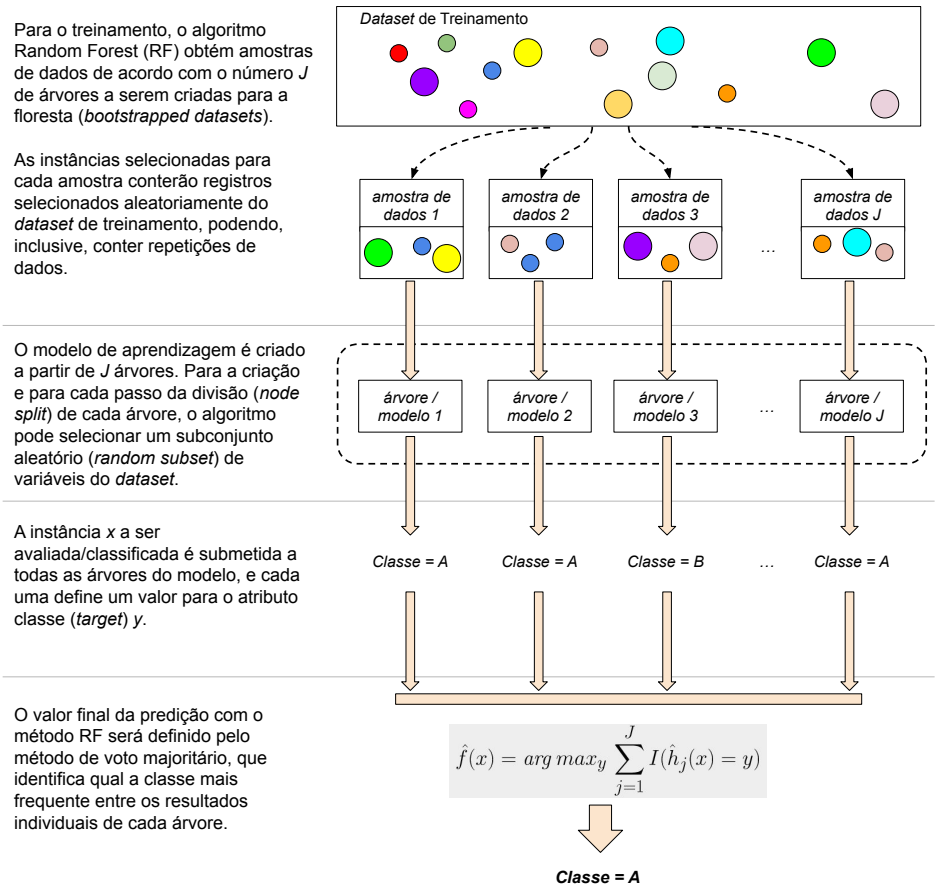


Figura 8 – Imagem ilustrativa e simplificada do funcionamento do método *Random Forest*.

Fonte: Adaptado de (SANDI, 2021; CUTLER; CUTLER; STEVENS, 2011).

O atributo de "*extreme*" certamente lhe é atribuído por suas qualidades, conforme destacadas por (CHEN; GUESTRIN, 2016), que incluem: escalabilidade, capacidade para lidar com dados esparsos, suporte para uso em computação distribuída e paralela, entre outras.

O XGBoost utiliza um processo de regularização (denominado de *regularized learning objective*, ou objetivo de aprendizagem regularizada) para minimizar a ocorrência de *overfitting*. De acordo com (XGBoost developers, 2022), num *ensemble* de árvores, tais como os usados pelo algoritmo *Random Forest*, os *scores* (pontuação) de predição de cada árvore para uma dada instância são somados para definir o score final do modelo, conforme Equação (2.13).

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), f_k \in \mathcal{F} \tag{2.13}$$

onde K é o número de árvores, f_k é uma função no espaço funcional $\mathcal{F}$, e $\mathcal{F}$ é o conjunto de todas as árvores de decisão possíveis. A função objetivo a ser otimizada é dada pela Equação (2.14):

$$obj(\theta) = \sum_{i}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \omega(f_k) \tag{2.14}$$

onde $\omega(f_k)$ corresponde à complexidade da árvore f_k.

No caso do XGBoost, o treinamento do modelo é feito de forma incremental (*additive training*), treinando uma árvore, ajustando o que foi aprendido e adicionando outra árvore em seguida e repetindo o processo para melhorar o modelo. Os autores em (XGBoost developers, 2022) explicam que, num cenário de *boosting* uma predição é dada pela Equação (2.15):

$$\begin{aligned}
\hat{y}_i^{(0)} &= 0 \\
\hat{y}_i^{(1)} &= f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i) \\
\hat{y}_i^{(2)} &= f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i) \\
&\dots \\
\hat{y}_i^{(t)} &= \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i)
\end{aligned} \tag{2.15}$$

onde $\hat{y}_i^{(t)}$ corresponde ao valor da predição da instância i no passo t. O autor em (AGUIAR, 2020) explica que na Equação (2.15), a predição na iteração t è equivalente a predição na iteração $t-1$ somada à predição de um novo modelo, f_t, e o erro no modelo (*loss*) na iteração t é definido pela Equação (2.16):

$$L_t = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{t-1} + f_t(x_i)) \tag{2.16}$$

onde t representa o número da iteração, n é o número total de amostras, l é uma função de erro (por exemplo: MSE, *mean squared error*), y_i é o valor do *target*, $\hat{y}_i^{(t-1)}$ corresponde à predição do (t-ésimo-1) modelo para a amostra x_i. O autor em (AGUIAR, 2020) complementa que, adicionando um termo de regularização ($\Omega(f_t)$) para minimizar a ocorrência de *overfitting*, tem-se a Equação (2.17):

$$L_t = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{t-1} + f_t(x_i)) + \Omega(f_t) \tag{2.17}$$

O autor complementa que, para o XGBoost, a função *loss* deve ser vista como um problema de otimização e, por isso, deve-se buscar um valor de f_t que minimiza L. Após algumas transformações na expressão original, a Equação (2.18) apresenta a função de perda para a iteração t:

$$L_t \approx \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t(x_i)^2 \right] + \Omega(f_t) \tag{2.18}$$

onde g_i corresponde a um gradiente e h_i a uma hessiana. A Figura 9 ilustra, de forma simplificada, o funcionamento do método XGBoost, em especial quanto ao processo *gradient boosting*.

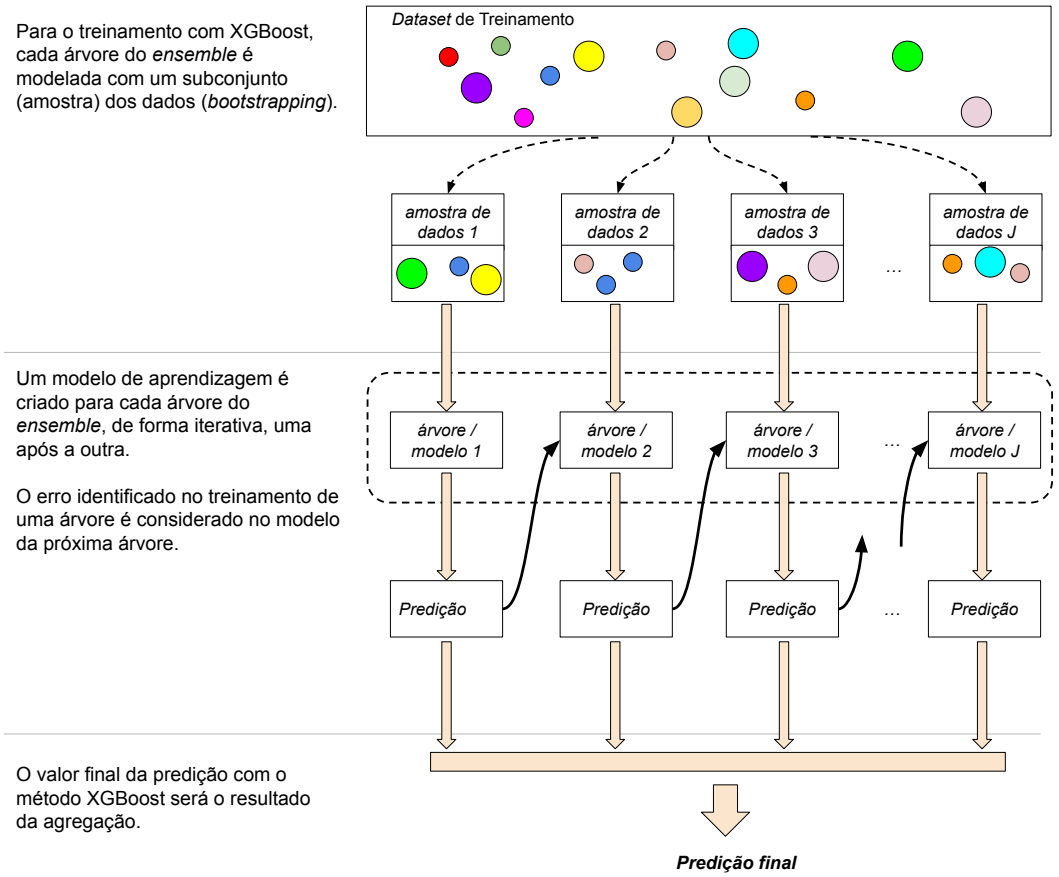


Figura 9 – Exemplo simplificado do funcionamento do processo de *gradient boosting* do método XGboost.

Fonte: Adaptado de (SANDI, 2021).

2.7 OTIMIZAÇÃO DE MODELOS DE *MACHINE LEARNING*

Os algoritmos de *machine learning*, tal como os descritos na Seção 2.6, e outros algoritmos encontrados na literatura podem, de uma forma geral e em diferentes níveis, ter seu desempenho otimizado pelo uso de algumas estratégias como as de seleção de características e de otimização de hiperparâmetros, descritas nesta seção.

2.7.1 Seleção de Características

O processo de seleção de características (*features*) visa identificar o subconjunto de atributos de um *dataset* com o qual se obtém melhores resultados com o modelo de aprendizagem gerado, seja nos resultados obtidos com esse subconjunto (por exemplo: favorecendo o alcance de uma melhor acurácia) ou no desempenho geral em termos de custo computacional (reduzindo o tempo de processamento, por exemplo). Um conjunto reduzido de características também ajuda a gerar modelos mais simples, facilitando a compreensão do modelo de ML gerado. Outros exemplos de benefícios da seleção de características são destacados por Guyon e Elisseeff (2003) e incluem: facilitar a visualização e compreensão dos dados, reduzir os requisitos de medição e armazenamento, diminuir os tempos de treinamento, e desafiar a Maldição da Dimensionalidade (*Curse of Dimensionality*, (BELLMAN; KALABA, 1959; BELLMAN, 1961)) de forma a melhorar o desempenho do processo de predição.

Os autores em (BLUM; LANGLEY, 1997) abordam sobre a questão da relevância de uma *feature*. Abaixo, seguem algumas de suas definições:

- ***Feature relevante ao target****: Uma característica x_i é relevante para uma função-target c (função-alvo c) se existir um par de exemplos A e B no espaço de instâncias, tal que A e B diferem apenas em sua atribuição a x_i e $c(A) \neq c(B)$.* Ou seja, conforme os autores explicam, essa definição estabelece que x_i é relevante se existir algum exemplo no espaço de instâncias para o qual ajustar o valor de x_i, mantendo fixos os valores das demais *features*, afeta a classificação dada pela função-*target c*.

- ***Feature fortemente relevante para a amostra/distribuição****: Uma característica x_i é fortemente relevante para a amostra S se existirem exemplos A e B em S que diferem apenas em sua atribuição a x_i e têm rótulos diferentes (ou têm distribuições diferentes de rótulos se aparecerem em S várias vezes). Da mesma forma, x_i é fortemente relevante para a função-target c e a distribuição D se existirem exemplos A e B com probabilidade não nula sobre D que diferem apenas em sua atribuição a x_i e satisfazem $c(A) \neq c(B)$.* De acordo com os autores, essa definição é similar à definição anterior (*Feature relevante ao target*), com a diferença de que A e B devem estar em S, ou devem ter probabilidade não nula.

- ***Feature fracamente relevante para a amostra/distribuição****: Uma característica x_i é fracamente relevante para a amostra S (ou para o alvo c e a distribuição D) se for possível remover um subconjunto das características de modo que x_i se torne fortemente relevante.*

- ***Feature com utilidade incremental****: Dada uma amostra de dados S, um algoritmo de aprendizagem L e um conjunto de características A, a característica x_i é*

incrementalmente útil para L em relação a A se a acurácia da hipótese que L gera usando o conjunto de características $\{x_i\} \cup A$ for melhor do que a acurácia alcançada usando apenas o conjunto de características A.

Apesar da possibilidade de se avaliar a relevância de uma determinada *feature*, os autores em (KOHAVI; JOHN, 1997) destacam que a relevância de uma característica não implica, necessariamente, que ela esteja no subconjunto ótimo de características. Complementam ainda que o contrário também é verdadeiro, ao indicar que a irrelevância de uma característica não implica que ela não deva estar no subconjunto ótimo de características. Para esse último caso, mencionam um exemplo de um cenário em que um *dataset* possui uma *feature* que apresenta sempre o valor 1. Nesse caso, ela aparenta ser irrelevante, porém pode ser essencial a depender do tipo de algoritmo de classificação a que o *dataset* seja submetido.

Uma visão mais recente apresentada por Li et al. (2017) adiciona a seleção de características sob a perspectiva dos dados. Para isso, avaliam se os dados são estáticos ou dinâmicos (*streaming*), se podem ser considerados dados convencionais ou heterogêneos, se representam fluxos de dados ou fluxos de características, entre outros aspectos, e selecionam o método mais adequado para a escolha das *features* de interesse.

Para este estudo, o foco será dado no uso de uma ou mais abordagens tradicionais, a serem selecionadas entre métodos de Filtro, *Wrapper*, Embutido e Híbrido, discutidos pelos autores em (JOHN; KOHAVI; PFLEGER, 1994; KOHAVI; JOHN, 1997; GUYON; ELISSEEFF, 2003; LI et al., 2017) e descritos a seguir.

Seleção do tipo Filtro:

Na seleção com o uso de filtro, as características são avaliadas independentemente umas das outras, sem considerar como uma combinação delas pode afetar o desempenho do modelo, e a seleção pode ocorrer sem levar em conta o algoritmo de *machine learning* que será utilizado. Métodos estatísticos ou heurísticos são geralmente usados para avaliar a relevância de cada característica.

Para a filtragem, pode-se avaliar, por exemplo: a quantidade de informação que uma *feature* fornece sobre a variável alvo (classe); a relevância da *feature*, ou a força de associação entre uma *feature* e a variável alvo; a autocorrelação entre uma *feature* e as demais características. Entre as técnicas utilizadas para fazer essa seleção, podem ser usadas, entre outras, medidas de correlação e de ganho de informação.

Os autores em (LI et al., 2017), mencionam que os métodos de seleção baseados em estatísticas (*statistical-based methods*) podem ser entendidos como métodos baseados em filtros, e sugerem o uso de algumas abordagens como a análise de variância dos dados (sugerindo, por exemplo, a remoção de atributos nos casos em que apresentem variância

igual a 0); a análise de *t-score* ou a análise de *Chi-Square score*, dando maior importância a características com maiores valores nesses indicadores; a análise de *Gini index*, dando prioridade à seleção de features com menor valor desse indicador; entre outras possibilidades.

Seleção com o uso de *Wrappers*:

A seleção com o uso de *wrappers* envolve o treinamento do modelo usando diferentes subconjuntos de características e a avaliação do desempenho do modelo com base em sua capacidade de prever corretamente. Métodos *wrapper* usam algoritmos de aprendizado para avaliar diferentes combinações de características.

Os autores em (KOHAVI; JOHN, 1997) destacam a importância de usar métodos de avaliação apropriados para estimar o desempenho do modelo com diferentes subconjuntos de características. Esses métodos podem incluir validação cruzada e separação do conjunto de dados em conjunto de treinamento e conjunto de teste. De acordo com os mesmos autores, o uso de técnicas de *wrapper* pode favorecer a obtenção de melhores resultados que os obtidos com técnicas de filtro, especialmente quando o conjunto de dados possuir muitas *features* irrelevantes. No entanto, os *wrappers* tendem a ser mais lentos que os filtros, por necessitarem treinar o algoritmo de aprendizagem múltiplas vezes.

Entre exemplos de técnicas de *wrapper*, pode-se citar:

- *Sequential forward selection* (SFS): É um método de seleção de características que começa com um conjunto vazio de *features* e, a cada iteração, adiciona a *feature* que tiver o maior impacto no desempenho do modelo. O processo é repetido até que o número desejado de características seja alcançado.

- *Recursive feature elimination* (RFE): É um processo de seleção de características que inicia com o conjunto completo de *features* e, a cada iteração, treina o modelo e avalia a importância de cada característica. A *feature* menos importante é removida. O processo é repetido até se atingir a quantidade de *features* determinada pelo usuário. O critério para escolher qual característica que deve ser removida é o impacto que ela tem no desempenho do modelo. Geralmente, isso é medido pela pontuação de importância atribuída a cada característica pelo modelo.

- *Sequential backward selection* (SBS): É uma técnica de seleção de características que começa com um conjunto completo de características e, a cada iteração, remove a característica que causa a menor diminuição no desempenho do modelo. O processo é repetido até que o número desejado de características seja alcançado. A principal diferença entre o SBS e o RFE é que o SBS remove as *features* com base na sua importância para o desempenho do modelo, enquanto o RFE remove as *features* com base na sua importância relativa às outras características. Em outras palavras, o SBS

é mais sensível ao impacto da remoção de uma característica no desempenho geral do modelo, enquanto o RFE é mais sensível à importância relativa das características.

É importante destacar que, como esses processos são iterativos, é possível avaliar após a sua execução qual combinação de *features* apresenta melhor resultado conforme a métrica de qualidade escolhida pelo usuário no momento da execução.

Seleção do tipo Embutida (*Embedded*):

A seleção com o uso de método *embedded* incorpora a seleção de *features* diretamente no processo de treinamento do modelo. Isso significa que a seleção de características ocorre embutida/integrada ao processo de construção do modelo de *machine learning*.

O método de regularização L1 (Lasso), por exemplo, pode ser considerado como um método embutido de seleção de *features*. Isso ocorre porque a penalidade L1 pode forçar alguns dos coeficientes do modelo a serem zero. Em outras palavras, a regularização L1 pode fazer com que o modelo ignore algumas das *features*. O algoritmo *Random Forest* é outro exemplo que, por sua vez, usa um conjunto de árvores de decisão para estimar a importância de cada característica para o modelo de ML.

Seleção Híbrida:

A seleção híbrida usa uma combinação das técnicas descritas anteriormente.

Um método híbrido pode, por exemplo, usar uma estratégia de filtro para reduzir o número de características e, em seguida, usar uma técnica de *wrapper* para selecionar as *features* mais relevantes. Ou então, pode-se utilizar, inicialmente, uma técnica de *wrapper* para selecionar as características mais relevantes e, depois, usar uma técnica *embedded* para refinar a seleção.

Além dos processos de seleção de *features* apresentados, os autores em (GUYON; ELISSEEFF, 2003), por sua vez, exploram também a construção automática de *features* e o uso de técnicas para a redução de dimensionalidade pela transformação dos dados. Os autores destacam que um desempenho melhor pode ser alcançado com o uso de características derivadas da entrada original. Além disso, sugerem que a construção de características pode ser vista como uma oportunidade para incorporar conhecimento específico do domínio nos dados. Entre os métodos de construção de *features*, destacam o uso de técnicas como agrupamento, transformações lineares básicas das variáveis de entrada (PCA–*Principal Component Analysis*, SVD–*Singular Value Decomposition*, LDA–*Linear Discriminant Analysis*), transformações lineares mais sofisticadas (como o uso de transformadas espectrais (por exemplo, Fourier, Hadamard, entre outros)), ou mesmo a

aplicação de funções simples a subconjuntos de variáveis.

As possibilidades de seleção de *features* são diversas. Com isso, a experimentação com diferentes abordagens pode possibilitar a exploração de diferentes cenários e comportamentos dos algoritmos de aprendizagem.

2.7.2 Otimização de Hiperparâmetros

O processo de otimização de hiperparâmetros busca identificar os valores para os hiperparâmetros de um algoritmo de *machine learning* que favorecem a obtenção de melhores resultados na classificação de determinado conjunto de dados em análise. O tema pode ser encontrado na literatura sob diferentes denominações, mas em especial como *tuning* de hiperparâmetros ou *Hyperparameter Optimization* (HPO). Em (ZöLLER; HUBER, 2021), os autores explicam em detalhes o histórico sobre o processo de HPO e comparam diferentes ferramentas disponíveis.

A escolha dos valores dos hiperparâmetros pode ser feita de forma empírica, com a exploração manual (busca manual) de valores conforme a experiência do cientista de dados que está fazendo o uso de determinado algoritmo de *machine learning*, ou por técnicas que facilitam a busca automatizada em um espaço maior de valores.

Entre as possibilidades existentes, a técnica de *grid search* (disponível para o Python, por exemplo, na função *sklearn.model_selection.GridSearchCV* (PEDREGOSA et al., 2011)) é bem difundida e faz uma busca exaustiva em um espaço amplo e pré-definido de valores para os hiperparâmetros do modelo. Com o objetivo de reduzir o espaço de busca, uma estratégia de *random search* (BERGSTRA; BENGIO, 2012), ou busca aleatória, também pode ser utilizada; nesse caso, ao invés de percorrer todos os valores do domínio de busca estabelecido, o processo faz uma busca aleatória de valores no intervalo estabelecido para cada parâmetro, geralmente numa quantidade de iterações determinada pelo usuário.

Outra possibilidade de HPO inclui o uso de ferramentas de AutoML (*Automated Machine Learning*) (FERREIRA et al., 2021), que buscam automatizar os processos de seleção de algoritmos e de *tuning* de hiperparâmetros. O espaço de atuação do AutoML é amplo, podendo incluir diferentes etapas de um processo de *machine learning*. De acordo com Salehin et al. (2023), o escopo do AutoML abrange vários estágios podendo incluir os processos de preparação de dados, *feature engineering*, seleção de modelo de aprendizagem e *tuning* de hiperparâmetros.

Para este estudo, o interesse está no uso de AutoML como uma técnica de HPO, uma vez que pode simplificar esse processo usando abordagens como, por exemplo, pesquisa aleatória, otimização bayesiana e algoritmos genéticos para encontrar os melhores valores de hiperparâmetros de forma automatizada.

Atualmente, várias são as ferramentas de AutoML disponíveis na literatura. Para

este estudo, foram selecionadas três bibliotecas para o Python que implementam funções para o processo de HPO e que estão descritas a seguir.

Auto-sklearn:

A biblioteca Auto-sklearn (FEURER et al., 2015; FEURER et al., 2020) é uma ferramenta de AutoML que emprega um processo de otimização bayesiana para automatizar o processo de seleção de modelos e ajuste de hiperparâmetros.

Em relação aos modelos de aprendizagem, o Auto-sklearn utiliza os algoritmos disponibilizados pela biblioteca *scikit-learn*. Com isso, o Auto-sklearn se beneficia de uma infraestrutura de modelos e métodos já implementados e validados.

A estrutura de funcionamento do Auto-sklearn está representada na Figura 10 e envolve, de forma geral, as seguintes etapas:

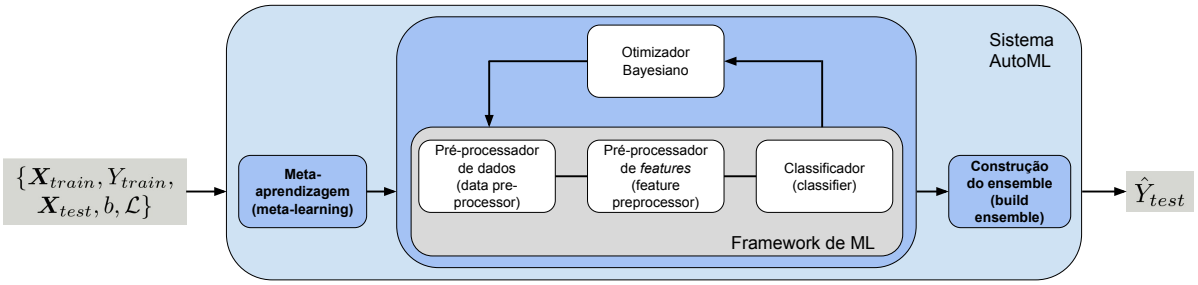


Figura 10 – Estrutura de Auto-sklearn.

Fonte: Adaptado de (FEURER et al., 2015).

1. Inicialmente o usuário fornece um conjunto de dados de treino e de teste e pode especificar diferentes parâmetros adicionais, como métrica de avaliação do modelo, tempo máximo de processamento, número de modelos a serem avaliados, limite de memória a ser considerado, etapas a serem consideradas (por exemplo: se deve fazer ou não a seleção de *features*), entre outros.

2. Em seguida, o Auto-sklearn realiza, automaticamente, algumas etapas de pré-processamento, como por exemplo a eliminação de valores nulos e o tratamento de variáveis categóricas.

3. Segue, então, com a construção de modelos de ML, explorando automaticamente um espaço de busca de modelos e de seus hiperparâmetros. Para essa busca, utiliza um otimizador bayesiano, que procura priorizar para etapas posteriores configurações que apresentaram melhores resultados em iterações anteriores.

4. Em relação à seleção de *features* e de hiperparâmetros, o Auto-*sklearn* pode fazer uso de métodos eficientes de busca como o modelo SMAC (*Sequential Model-based Algo-*

rithm Configuration) (HUTTER; HOOS; LEYTON-BROWN, 2011), para encontrar uma configuração ótima.

5. O Auto-sklearn avalia a execução dos vários modelos produzidos e seleciona o modelo mais apropriado considerando o desempenho alcançado e as configurações de execução estabelecidas pelo usuário.

O processo de otimização bayesiana é utilizado por várias ferramentas de AutoML. Os autores em (FEURER et al., 2015) destacam que as principais características que diferem o Auto-sklearn de outros métodos são, em primeiro lugar, a inclusão de uma etapa de meta-aprendizagem (*meta-learning*), usada para iniciar o procedimento de otimização bayesiana, que auxilia na obtenção de melhoria na eficiência. Em segundo lugar, a inclusão de uma etapa automatizada de construção de *ensemble* de modelos, que permite considerar todos os classificadores avaliados pelo processo de otimização bayesiana. O Auto-sklearn aplica o *meta-learning* para selecionar instâncias do *framework* de ML que podem favorecer melhor desempenho para um novo conjunto de dados. Para criar essa base de meta-aprendizagem, o Auto-sklearn coleta dados de desempenhos obtidos para diferentes *datasets* para um conjunto de *meta-features*, ou características que podem ser extraídas de um conjunto de dados e aplicadas a novos conjuntos de dados para comparação.

TPOT:

A biblioteca TPOT (*Tree-based Pipeline Optimization Tool*) (OLSON et al., 2016) é uma ferramenta de AutoML que utiliza programação genética na otimização de *pipelines* de *machine learning*. O TPOT usa algoritmos de aprendizagem e funções de transformação de dados disponíveis no *scikit-learn*. Além disso, o TPOT integra a otimização de Pareto com o objetivo de auxiliar na produção de *pipelines* compactos sem comprometer a performance da classificação.

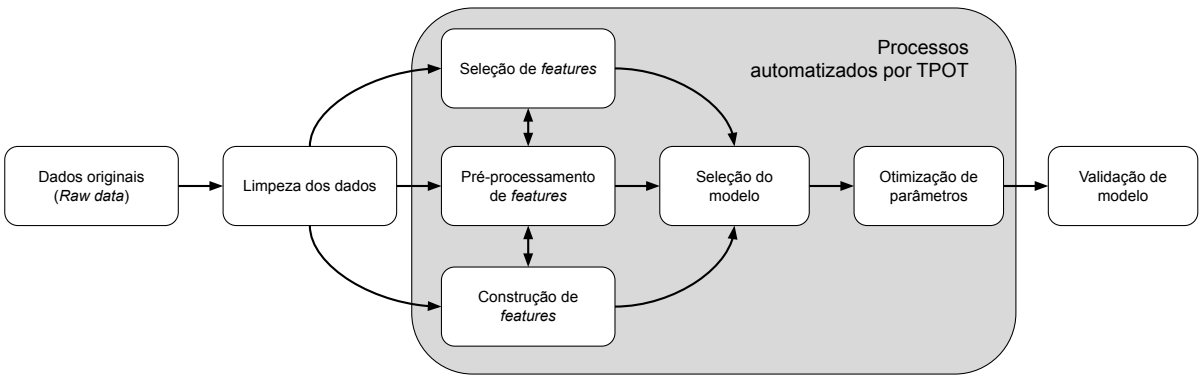


Figura 11 – Exemplo de um *pipeline* de *machine learning*. Os elementos contidos no quadro cinza indicam processos que podem ser automatizados por TPOT.

Fonte: Adaptado de (OLSON et al., 2016).

Na Figura 11 é apresentado um *pipeline* de um processo de *machine learning* e, no quadro em cinza, estão indicadas as etapas que podem ser automatizadas com o TPOT. De acordo com Olson et al. (2016), os principais tipos de operadores de *pipeline* (*pipeline operators*) implementados no TPOT incluem:

- **Pré-processadores:** implementam um operador de escalonamento padrão (*StandardScaler*) que usa a média e a variância da amostra para padronizar as características; além disso, oferecem outro operador, denominado de escalonamento robusto (*RobustScaler*) que usa a mediana e a amplitude interquartil da amostra para dimensionar as características; e também dispõem de um operador denominado de *PolynomialFeatures* que gera *features* por meio de combinações polinomiais de características numéricas.
- **Decomposição:** para a decomposição, implementam uma variante da Análise de Componentes Principais (PCA), denominada de *RandomizedPCA* que usa a Decomposição Singular de Valores Randomizada (SVD).
- **Seleção de características:** para a seleção de características, implementam diferentes abordagens como a eliminação recursiva de características (*Recursive Feature Elimination* (RFE)); uma outra estratégia denominada de *SelectKBest*, que escolhe as k-melhores características; uma seleção com base no percentil superior das características (*SelectPercentil*); e uma estratégia que remove as características que não atendem a um limite mínimo de variância (*Variance Threshold*).
- **Modelos:** para a aprendizagem supervisionada, que é o interesse deste estudo, implementam modelos baseados em árvores (*Decision Tree*, *Random Forest*, *Gradient Boosting*, entre outros), modelos lineares como SVM e Regressão Logística, e *k-nearest neighbors*.

Em comparação com o Auto-sklearn, os autores do TPOT (OLSON et al., 2016) destacam que o Auto-sklearn explora um conjunto fixo de *pipelines* com apenas um pré-processador de dados, um pré-processador de características e um modelo, não conseguindo, dessa forma, gerar *pipelines* grandes/complexos, muitas vezes necessários para problemas de AutoML. Para contornar isso, os autores do TPOT identificaram que o uso de programação genética (GP, ou *genetic programming*) na otimização de *pipelines*, poderia favorecer o projeto de *pipelines* mais adequadas para problemas de classificação supervisionada.

Scikit-Optimize (*skopt*):

A biblioteca *Scikit-Optimize*, disponível em <https://scikit-optimize.github.io/>, implementa métodos de otimização baseados em modelos sequenciais que podem ser usados

na minimização de funções. Para o escopo deste estudo, o interesse nessa biblioteca está em seu uso como uma ferramenta de AutoML para a otimização de hiperparâmetros.

Para isso, são de interesse as seguintes funções de *skopt*:

- *skopt.dummy_minimize* — Busca aleatória por amostragem uniforme dentro de limites estabelecidos. A função gera pontos no espaço de busca de forma aleatória e avalia a função objetivo nesses pontos. Essa função tende a requerer grande quantidade de avaliações da função objetivo para encontrar uma solução.

- *skopt.gbrt_minimize* — Otimização sequencial usando *Gradient-Boosted Regression Trees* (GBRT). A função *skopt.gbrt_minimize* utiliza o modelo GBRT para fazer previsões sobre o valor da função objetivo em locais não explorados. Em seguida, avalia a função objetivo nos pontos onde o modelo GBRT sugere que há uma alta probabilidade de encontrar um mínimo. Esses pontos são escolhidos de maneira inteligente para explorar áreas promissoras do espaço de busca, minimizando o número de avaliações necessárias.

- *skopt.gp_minimize* - Otimização bayesiana usando processos gaussianos para minimizar uma função objetivo. Um processo gaussiano é um tipo de modelo probabilístico em que cada conjunto de variáveis aleatórias, em qualquer intervalo de tempo, segue uma distribuição gaussiana (distribuição normal). O processo gaussiano é usado para modelar a função objetivo, permitindo que o algoritmo de otimização faça previsões sobre o valor da função objetivo em locais não explorados.

De uma forma geral, para todas as funções de *skopt* descritas nesta seção, o usuário deve determinar a quantidade de iterações que devem ser realizadas.

2.8 CONSIDERAÇÕES FINAIS

Neste capítulo, o conceito de arquitetura do *smart grid* foi apresentado para melhor delimitar o cenário de estudo, e o termo *posições candidatas* foi introduzido para explicar, de forma geral, como se dá a seleção de postes no processo de posicionamento.

O *problema das p-Medianas* serve como referência para a modelagem teórica do método analítico proposto por este estudo. O modelo de cálculo de perdas e de potência recebida no enlace apresentado serve, por sua vez, para avaliar a possibilidade de conexão entre medidores e posições candidatas.

O conceito de *árvore geradora mínima* é explorado nesta pesquisa na análise de conexões com múltiplos saltos.

Além disso, uma explicação geral sobre alguns algoritmos de *machine learning* foi apresentada por se referirem aos métodos explorados nos experimentos realizados por esta pesquisa.

Por último, a seção explorou também os conceitos de seleção de características e otimização de hiperparâmetros por serem processos que podem ser usados na busca por melhores resultados no desempenho dos modelos de ML criados nos experimentos.

Os termos apresentados neste capítulo, além de servirem como referência teórica para os temas discutidos nesta tese como um todo, podem ser abordados, também, pela literatura existente sobre o posicionamento de roteadores/*gateways*, bem como as que exploram problemas de comunicação em redes de comunicação sem fio e o uso de abordagens de *machine learning*. No próximo capítulo, referências que abordam sobre o posicionamento de roteadores/*gateways* e o uso de técnicas de inteligências artificial/*machine learning* são apresentadas.

3 ESTADO-DA-ARTE

Nos anos recentes, termos como IoT (*Internet of Things*), *Big Data*, redes *mesh*, redes *5G*, entre outros, têm se incorporado ao vocabulário cotidiano das pessoas em diferentes áreas. Isso pode ser justificado pelo uso cada vez mais intenso de tecnologias sem fio, com a aplicação difundida em diversas áreas e facilitada pelo surgimento de dispositivos de custo mais acessíveis, com o surgimento de dispositivos de transmissão em longo alcance e com as tecnologias que permitem velocidades de transmissão e de processamento cada vez maiores. Conceitos como cidades inteligentes (*smart cities*) e redes elétricas inteligentes (*smart grids*) surgem como áreas de aplicação dessas novas tecnologias de comunicação sem fio. Juntamente com esses conceitos, o uso de tecnologias de inteligência artificial e *machine learning* tem se intensificado com o propósito de viabilizar o gerenciamento e processamento do volume de dados gerados e, também, para possibilitar antever as demandas geradas por esses cenários. (YARALI, 2022a; SHAFIQUE et al., 2020; YARALI, 2022b; PONCHA et al., 2018; AL-SAMAWI; SINGH, 2022).

Apesar do uso crescente de tecnologias de comunicação sem fio na área de *smart grids*, não se identificou na literatura recente um estudo que apresente uma visão geral das aplicações de IA/ML para comunicação sem fio, mais precisamente no que se refere ao uso de técnicas de *machine learning* como ferramenta principal para posicionamento de *gateways*. Este é um problema em que tais tecnologias podem ser usadas na etapa de projeto de topologia da rede de comunicação. É possível encontrar trabalhos como os desenvolvidos pelos autores em (MOCANU, 2017), (DHARMADHIKARI et al., 2021), (MIN et al., 2021), (TESTI et al., 2019), (HORIHATA et al., 2020) e (ZHAO et al., 2022), que abordam diferentes usos de IA/ML em problemas de comunicação sem fio, alguns mais focados em aspectos como alocação de recursos, otimização de consumo de energia, análise de consumo, enquanto outros abordam temas como segurança, topologia de rede e portfólio de investimento. No entanto, nenhum com foco específico na aplicação de ML na área de interesse deste trabalho.

Dessa forma, com o objetivo de preencher essa lacuna, este capítulo apresenta os resultados de uma pesquisa de referências bibliográficas sobre a utilização de técnicas de inteligência artificial e *machine learning* em problemas de posicionamento de *gateways* que avalia um conjunto de referências buscadas tendo como foco principal a área de redes de comunicação, porém com certo viés à procura por referências na área de *smart grids* e tecnologias aplicáveis a essa área. Com isso, diferentes tecnologias de redes foram identificadas e avaliadas quanto ao problema de posicionamento de dispositivos e processos de planejamento de infraestrutura.

Como resultado, é apresentado um panorama das técnicas utilizadas objetivando identificar as encontradas com mais frequência e as tendências observadas. Além disso, o estudo procurou localizar referências com abordagem similar à explorada como tema principal desta pesquisa.

3.1 PRINCIPAIS TÉCNICAS DE POSICIONAMENTO

Nesta seção, são apresentadas informações sobre as referências selecionadas para análise por este estudo, com o objetivo de identificar artigos que abordem o tema de posicionamento de roteadores/*gateways* de forma a evidenciar as principais estratégias utilizadas e a aplicação de técnicas de IA/*machine learning* nesse cenário.

O interesse principal é o de identificar a aplicabilidade na área de *smart grid* (SG). Para isso, foi feita uma exploração geral sobre o tema posicionamento de *gateways* em redes *wireless* de uma forma geral para avaliar as diferentes abordagens utilizadas e que podem contribuir com a área de *smart grid*, visto que o problema geral denominado de *gateway placement problem* é de aplicação em diferentes cenários. Algumas referências podem abordar questões de infraestrutura e foram selecionadas a título de complementação do estudo como exploração de uso das técnicas de IA/ML.

Sobre a terminologia, roteadores e *gateways* são explorados na literatura sob diferentes denominações, tais como: concentradores (*concentrators*), controladores (*controllers*), *data concentrator unit*, *data aggregation points*, *access points*, entre outros. Para este estudo, no entanto, tais termos devem ser entendidos como sinônimos de roteadores/*gateways*, pois o interesse é o de identificar técnicas de posicionamento de dispositivos de redes de comunicação que podem servir como elementos concentradores que atuam como interface de comunicação, recebendo pacotes de dados por sua interface de entrada (oriundos de diferentes dispositivos) e encaminhando tais pacotes de informação em direção a um centro de operação de distribuição que concentra e processa todos os dados. O caminho de comunicação em sentido inverso também é válido nesse cenário.

Para a seleção de referências, diferentes bases foram consultadas, incluindo: IEEE Xplore, ACM Digital Library, SpringerLink, Elsevier (ScienceDirect) e pesquisas abertas no buscador Google. Uma exploração inicial em tais bases buscou identificar a aplicação de técnicas de *machine learning* em redes de comunicação sem fio de uma forma geral. Dessa busca, 112 referências foram selecionadas e classificadas. As pesquisas nas bases foram feitas considerando os seguintes termos: *"smart grid machine learning"*, *"smart grid gateway placement machine learning"*, *"smart grid"+"gateway placement"*, *"smart grid"+"gateway location"*, *"smart grid"+"device placement"*, *"smart grid"+"machine learning"+"positioning*, *"smart grid"+"device location"* e variações dessas combinações de forma a não limitar a busca apenas a aplicações em *smart grids*, permitindo encontrar referências que abordassem

sobre outras tecnologias de rede, como redes *mesh*, redes de sensores, entre outras. A busca inicial procurou identificar referências datadas do ano 2018 em diante, encontrando artigos até o ano de 2022. Para esse intervalo, 84 referências foram encontradas. As demais, datam de anos anteriores, sendo 26 do ano 2010 até o ano 2017 e o restante de 2006 a 2009 (mantidas nos resultados por sua relevância). Das 112 referências encontradas, 88 se referiam a artigos de conferências e monografias, e 24 classificadas como surveys. Em relação ao principal tema explorado por essas referências, tem-se a seguinte distribuição: *controller placement* (43,5%), análise de dados e monitoramento (21,0%), eficiência energética (16,9%), segurança (8,9%) e detecção de anomalias (8,5%).

Após a identificação das 112 referências, uma filtragem foi feita de forma a priorizar as referências que abordavam sobre o problema de posicionamento de dispositivos, que foram classificadas conforme o quadro apresentado na Figura 12, resultando em 38 referências de interesse. A classificação qualifica a referência em relação ao método utilizado (IA/ML ou Outro Método) e quanto à área de aplicação do trabalho (se direcionado principalmente para *smart grid* ou direcionado para outras áreas/outros temas, mas de interesse para o estudo sobre o posicionamento de *gateways*). É importante ressaltar que, um trabalho classificado com a indicação de usuário de métodos de IA/ML, não está limitado ao uso dessas abordagens, uma vez que é comum a modelagem de problemas de posicionamento como problemas de Programação Linear Inteira, pelo conjunto de restrições que possui, mas resolvidos com técnicas de clusterização ou outra técnica de IA/ML como algoritmo principal. Nesses casos, o trabalho foi classificado como usuário de método IA/ML. A classificação de trabalho como uma aplicação de *smart grid* pode ser identificada pelo título ou explicitamente indicada no texto do trabalho (algumas vezes indicados como problemas de *smart grid*, ou problemas de *advanced metering infrastructure*).

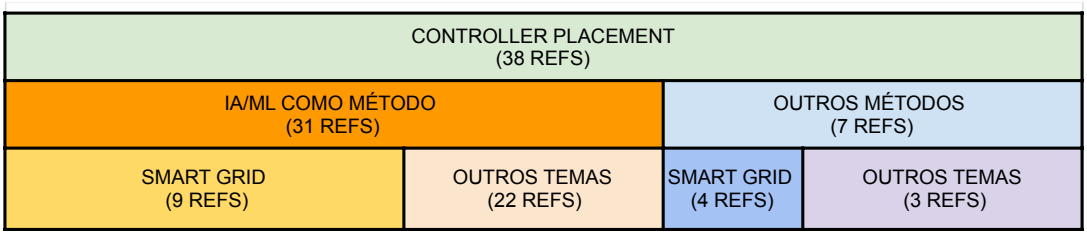


Figura 12 – Classificação das referências selecionadas.

Fonte: Autoria própria.

Do total de referências selecionadas para a análise, 34,2% abordam *smart grid* como tema principal. Essa indicação pode estar no título do trabalho, ou descrita explicitamente no texto. Mas também é importante verificar que uma boa parcela das referências abordam sobre outras áreas de aplicação com o intuito de deixar o trabalho mais amplo. Redes *mesh*, por exemplo, têm grande aplicabilidade no cenário de *smart grids*, e trabalhos que abordam sobre o posicionamento de *gateways* em redes *mesh* podem ser de interesse para

o escopo do estudo. O mesmo é válido para outras tecnologias. Medidores inteligentes podem ser considerados sensores especializados; dessa forma, trabalhos que exploram o posicionamento de dispositivos em redes de sensores também podem ser relevantes para o estudo e foram incluídos na lista de trabalhos selecionados. Da mesma forma, o uso de estratégias para posicionamento de dispositivos em redes LoRaWAN, SDN, 5G, entre outros, foram explorados para avaliar as técnicas de IA/ML ou outra utilizadas em problemas de posicionamento de dispositivos. Um exemplo da importância de se considerar diferentes tecnologias de rede pode ser evidenciado no trabalho de (SCARAMELLA et al., 2022) que explora a comunicação LoRaWAN sobre redes Wi-SUN, que é uma tecnologia amplamente utilizada em redes elétricas inteligentes.

A diversidade de tecnologias de redes existentes no cenário da comunicação *wireless* pode ser visualizada na Figura 13.

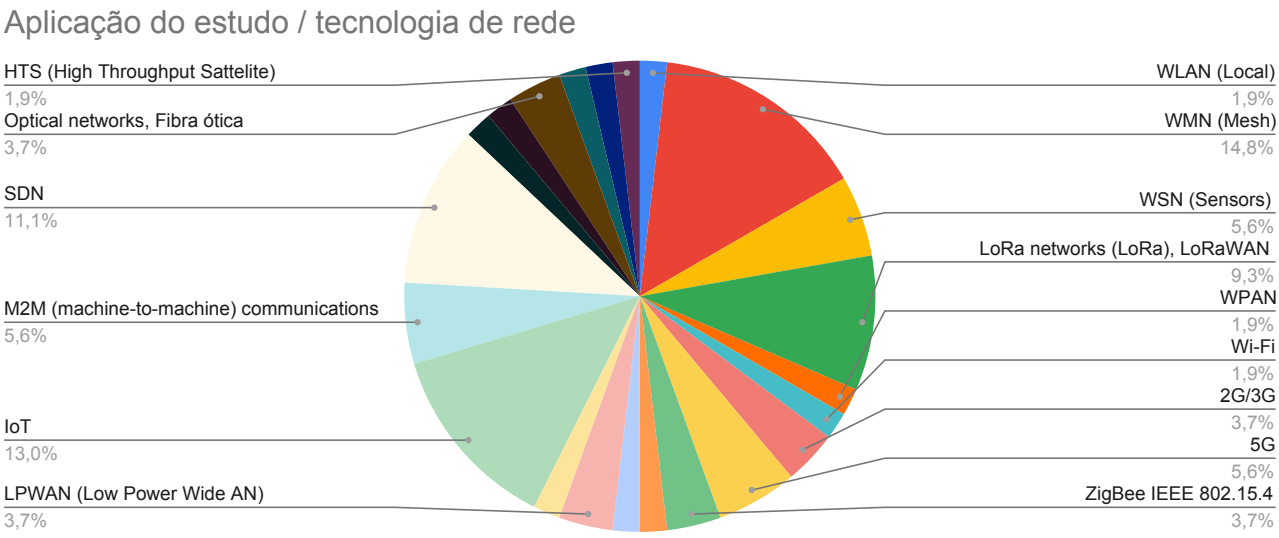


Figura 13 – Classificação das referências por área/tecnologia de aplicação.

Fonte: Autoria própria.

Quanto à técnica utilizada por cada referência, 81,6% utilizam ou citam o uso de métodos baseados em IA ou o uso de técnicas de ML (principalmente como método auxiliar). Do total de referências, 92,7% apresentam foco principal no processo de posicionamento de *gateways* (ou *controller placement*). As demais referências abordam aspectos relativos ao planejamento da rede sob diferentes focos como eficiência energética, segurança, expansão da rede, roteamento, entre outros. Adicionalmente, 88,4% dos trabalhos consultados faz uso de experimentos. Isso é relevante pois possibilita, além de identificar o tipo de método utilizado pelos autores da referência, verificar os resultados obtidos com as diferentes abordagens propostas.

3.1.1 IA/ML para posicionamento de dispositivos de comunicação em *smart grid*

Para o posicionamento de DAPs (*Data Aggregation Points*) em NAN, os autores em (GALLARDO; AHMED; JARA, 2021) propõem o uso de técnicas de agrupamento para minimizar a distância entre DAPs e SMs, dividindo as vizinhanças em sub-redes. Com isso, também procuram reduzir o número de saltos a até 3 saltos. Eles introduzem uma nova métrica denominada de densidade de cobertura (*coverage density*), que define se o planejamento feito para determinada zona assegura a cobertura necessária. Os autores citam que o problema de posicionamento de DAPs é um tópico subexplorado no domínio de *smart grids*. Os métodos utilizados pelos autores incluem distância de *Haversine*, *Floyd-Warshall algorithm* (para encontrar rota de menor caminho entre um determinado nó e outro nó da rede) e o algoritmo de clusterização *k-Medoids*, que levam em consideração a distância entre medidores inteligentes e o alcance de transmissão dos medidores.

A minimização de distância entre DAPs e medidores inteligentes atendidos por eles pode ser um requisito de um problema de posicionamento de dispositivos. Sobre isso, os autores em (WANG et al., 2018) usam uma abordagem de particionamento da rede, com o uso de algoritmo de clusterização *Clustering-based DAP Placement Algorithm* que faz o agrupamento buscando a minimização de distância. Os autores utilizam, também, o *Floyd Warshall algorithm* para a busca de menor caminho, e a distância entre os nós como informações consideradas pelo método.

As análises de topologias da rede de energia e da rede de comunicação podem ser feitas de forma conjunta conforme demonstrado pelos autores em (MAHDY et al., 2017). Esse estudo tem como objetivos principais estabelecer as coordenadas dos DAPs e minimizar o atraso médio total do sistema, considerando, para isso, o volume de tráfego de dados e a minimização de custos. O trabalho objetiva fazer isso sem comprometer QoS (*Quality of Service*). Isso é alcançado focando na ideia de agregar e compactar dados associados ao mesmo alimentador de energia no DAP apropriado antes de serem enviados para o UC (*Utility Center*). Os autores destacam que ter medidores inteligentes alimentados por diferentes alimentadores da rede de energia, conectando-se ao mesmo DAP na rede de comunicação, pode levar a uma ineficiente agregação de dados. Portanto, de acordo com os autores, o problema de posicionamento de DAPs não deve considerar a rede de comunicação isolada da rede elétrica. O problema é formulado pelos autores como um *mixed integer non-linear optimization problem*, e a otimização é feita com algoritmo genético. As características de rede consideradas pelo estudo incluem: *delay*, número de *smart meters* (nós) por *cluster*, *path loss*, tamanho de pacotes, taxa de transmissão, capacidade do enlace entre DAP e UC, SINR (*Signal-to-Interference plus Noise Ratio*) e coordenadas dos *gateways*.

Um estudo que aborda o planejamento de infraestrutura de rede em vários níveis,

com o objetivo de minimizar custos de construção da rede e maximizar o número de nós facilitadores no nível da rede de forma a assegurar a flexibilidade é apresentado pelos autores em (SILVA, 2012). Nesse estudo, os autores sugerem que ele é aplicável para redes *smart grid* e usos similares, pois a mesma arquitetura é válida para rede de telecomunicações para redes elétricas, água, gás. A abordagem utilizada pelos autores inclui o uso de algoritmos evolucionários. Sugerem um método denominado de algoritmo multinível para projetos de rede (*Multilevel Algorithm for Network Design* - MAND). O método proposto usa o algoritmo de *Dijkstra* para encontrar o melhor caminho entre os nós e algoritmos de otimização de objetivo único (como GA – *Genetic Algorithm* –, ou PSO – *Particle Swarm Optimization*) e multiobjetivos (como NSGA-II e MOPSO). Para o processamento, eles levam em consideração a quantidade de nós de demanda, nós facilitadores e custos de trajetos entre nós de demanda, nós facilitadores e nós ascendentes. Os resultados obtidos são comparados com a análise individual de cada nível da rede com a análise multinível proposta pelo método desenvolvido pelo estudo (MAND). Os métodos GA ou PSO, e NSGA-II e MOPSO foram escolhidos, e avaliam redes de até 5 níveis. Os autores também comparam os resultados com as soluções que seriam obtidas com projetistas usando ferramentas básicas.

Os autores em (SOUZA et al., 2013) propõem um algoritmo para determinar a localização ótima de concentradores em *smart grids* para um cenário baseado no protocolo de comunicação *ZigBee Mesh* IEEE 802.15.4 no acesso final e comunicação GPRS (*global packet radio service*) no concentrador da rede *mesh*, conectando diretamente com o *head-end system* da concessionária (*backhaul*). O estudo pressupõe uma arquitetura de *smart grid* com *neighborhood area network*, onde estão localizados os medidores, *wide area network*, onde estão posicionados os coletores (geralmente instalados em postes de iluminação), e um *head-end system*, camada em que é efetuado o gerenciamento da rede. A comunicação entre coletores e interface de gerenciamento (*Backhaul*) é baseada em comunicação por IP (*Internet Protocol*). Os autores indicam que uma das grandes dificuldades no projeto de redes NAN em *smart grids* é o posicionamento de nós coletores. O posicionamento ideal otimiza custos e melhora o desempenho do sistema. Para isso, os autores propõem um método que considera o posicionamento de concentradores em postes. Então, todos os postes disponíveis passam a ser posições possíveis (posições candidatas) para a instalação do concentrador. A meta consiste em encontrar a menor quantidade de saltos entre os medidores e os concentradores. Ou seja, encontrar o menor caminho, que minimize o custo das rotas e assegure desempenho do sistema. Usam algoritmos que procuram estabelecer a melhor rota entre dois pontos. No caso, utilizam os seguintes algoritmos: *Dijkstra*, *Bellman-Ford* e BFS (*Breadth-First Search*). No caso em que definem que mais de um concentrador deve ser posicionado, os autores utilizam *k-Means* para a formação de *clusters*, tendo como centroides as posições candidatas (postes).

O posicionamento de concentradores é geralmente definido de forma a atingir deter-

minados objetivos como, por exemplo, a minimização de saltos ou o balanceamento de carga. No estudo de (FERREIRA et al., 2015), os autores avaliam dois métodos: *Kmeans-Dijkstra* (utilizado pelos autores em (SOUZA et al., 2013)), e o algoritmo Recursivo (proposto pelos autores em (AOUN et al., 2006)). Avaliam a quantidade de nós por concentrador, a carga nos nós dominantes e a quantidade de saltos permitida na comunicação dos medidores com seus respectivos concentradores. Para esse balanceamento, eles consideram como fatores que afetam o desempenho da rede (parâmetros de QoS) os seguintes elementos: vazão, atrasos e perdas. O algoritmo *Kmeans-Dijkstra* procura minimizar a quantidade de saltos na rede. Os concentradores são considerados centros de massa do algoritmo *k-Means*. O *Kmeans-Dijkstra* é capaz de minimizar a quantidade de saltos, mas não leva em conta critérios de posicionamento e rotas de pacotes entre medidores e concentradores. O algoritmo Recursivo, por sua vez, é capaz de levar em conta restrições como: carga, carga nos nós dominantes e quantidade máxima de saltos.

Um processo iterativo de minimização de número de saltos, com consequente maximização de *throughput* e diminuição de tempo de espera (atrasos na rede) é proposto pelos autores em (TANAKORNPINTONG et al., 2017) para redes IEEE 802.15.4. Os autores indicam que, apesar de muitos artigos tratarem do problema de otimização de posicionamento de dispositivos em redes *wireless*, poucos são os que tratam, especificamente, de localização de concentradores em redes AMI (*advanced metering infrastructure*). Além disso, os autores comentam que estabelecer o posicionamento de concentradores (DCU, *data concentrator unit*) é uma tarefa desafiadora. Citam a falta de estudo de posicionamento de DCU em redes AMI que levem em consideração o *throughput* médio, o atraso e a otimização de número de saltos. Os autores não fazem uso de um algoritmo ou técnica específica de otimização, mas utilizam um processo iterativo de minimização de número de saltos, com consequente maximização de *throughput* e diminuição de tempo de espera (atrasos na rede). Inicialmente, é utilizado um algoritmo de agrupamento (*clustering*) como o *k-Means* para definir o posicionamento de concentradores nos centroides estabelecidos por esse algoritmo. Consideram como métrica de desempenho para o estabelecimento dos centroides o tempo de entrega de pacotes entre medidores e concentradores (*end-to-end delay*). Um processo iterativo é realizado até que todos os medidores estejam associados a um concentrador. Para a análise de *throughput*, estimam a probabilidade de perda de pacote, a quantidade de tempo de operação do sistema sem erros no canal *wireless*, e definem a taxa de erro de pacote. Consideram também a potência de transmissão e a distância (saltos) entre os nós e estimam a relação sinal-ruído (SNR). O *throughput* é definido pelo número médio de pacotes recebidos por unidade de tempo no receptor, que, no caso, é o coletor (DCU) em análise. Pode ser entendido, também, como a taxa de serviço de pacotes em um determinado nó (*x-hop node*) que não são bloqueados pelos nós intermediários entre o determinado nó (*x-hop*) e o coletor. A probabilidade de bloqueio de pacotes em cada salto considera um modelo de fila *M/M/1/K* em que o sistema consiste

de 1 servidor e um *buffer* de tamanho K. O *throughput* médio por nó pode ser usado como medida de eficiência média por nó. Para a análise de *delay*, é considerado que o tempo para entrega de um pacote de um nó até o destino (no caso um coletor) corresponde à soma do tempo de transmissão e do tempo de espera (atraso) nas filas existentes por todos os nós intermediários.

Em (XING et al., 2016), os autores abordam sobre o posicionamento de APs (*access points*) em *smart grids*, com rede de comunicação implementada com PLC (*power-line communication*) *network* que, primeiramente, estabelece um modelo de otimização para a localização do AP que minimiza o custo de instalação de APs, enquanto satisfaz as restrições de confiabilidade, atraso de rede e resiliência. Em seguida, um algoritmo genético melhorado é proposto para resolver o problema de otimização. Quanto aos aspectos de restrições de projeto, citam: a) minimização do custo de construção da rede, principalmente dos custos de instalação dos APs; b) maximização do nível médio de confiabilidade em condições normais de operação; c) garantia de que os ENs (*end-nodes*) estejam sempre conectados a pelo menos um AP, a fim de manter a rede funcionando em um nível adequado, mesmo que um enlace fique indisponível (ou seja, garantindo que a confiabilidade da rede permaneça acima de um limite predefinido); e d) redução do atraso de comunicação para atender aos requisitos da aplicação. Nesse trabalho, o algoritmo utilizado para resolver o problema de planejamento é um algoritmo genético melhorado. Para evitar o ótimo local e manter a diversidade populacional, a função densidade foi introduzida com base no GA padrão. Entre os elementos de rede avaliados, destacam-se: *transmission delay*, custo de implantação do AP, número de nós (*end-nodes*), número de posições candidatas para os APs, conjunto de caminhos entre APs e ENs, e confiabilidade do caminho entre APs e ENs.

A minimização de latência da rede é uma necessidade comum a diferentes estudos e abordada por (WANG et al., 2017). Os autores formulam um *DAP problem* e depois usam uma abordagem de *clustering* para o particionamento da rede (*network partitioning*) com o objetivo de minimizar a latência máxima de propagação dos dados entre cada DAP e os medidores associados a ele. Utilizam um Algoritmo de *Dijkstra* para o cálculo de menor caminho entre dois nós e levam em conta a Distância de *Haversine* entre os elementos. Usam uma abordagem *clustering-based DAP placement* (CDP) para o posicionamento. As principais características de rede consideradas pelos autores incluem: quantidade de medidores, posição dos medidores, alcance de transmissão e o número de DAPs.

3.1.2 IA/ML para posicionamento de dispositivos de comunicação em redes *wireless*

A tecnologia 5G é um tema que desperta interesse de pesquisa em diferentes aspectos de forma cada vez mais frequente. Os autores em (RAITHATHA et al., 2021) exploram o problema de posicionamento de *gateways* em redes Ultra-Densas com o objetivo de minimizar o número médio de saltos (ANH, *average number of hops*) e maximizar a capacidade da rede *backhaul* (*BNC, backhaul network capacity*). Eles exploram métodos heurísticos baseados em IA/ML para definir as posições de *gateways* e associar *small cells* da rede a tais *gateways*. O trabalho sugere uma abordagem denominada de K-GA, que faz uso de *k-Means*, algoritmo genético e algoritmo do menor caminho *Dijkstra* (utilizado para associar *small cells* a *gateways*). Os autores escolhem utilizar GA por considerarem que esse é o método mais usado para o problema das *p-Medianas* (comumente utilizado para a modelagem de problemas de posicionamento de controladores). Para a execução do método, avaliam o alcance de sinal de cada micro-célula da rede, além do número de saltos e capacidade da rede.

O posicionamento de *gateways* em LPWAN/redes de sensores considerando cancelamento de interferência (*interference cancellation*) de forma a maximizar a PDR (*packet delivery ratio*, ou taxa de entrega de pacotes) é abordado pelos autores em (TIAN; WEITNAUER; NYENGELE, 2018). Para isso, eles avaliam a minimização do número de contenção média, que é inversamente proporcional ao PDR utilizando algoritmos gulosos: WBG (*weight bipartite graph*); e PGL, abordagem que limita posições de *gateways* a pontos ou *pixels* num *grid* retangular. Os autores sugerem essa abordagem quando o número de *sensor nodes* (SN) passa a ser muito grande. O estudo é direcionado para redes TO LPWAN (*transmit only LPWAN*), em que os sensores possuem apenas a função de *transmitter* ativada (*receiver* desligado), para economia de energia. *Gateways* coordenam a recepção de dados para reduzir perda de pacotes e maximizar a eficiência do canal, fornecendo diversidade contra desvanecimento (*fading*) e IC (*interference cancellation*) para mitigar colisões de pacotes. O método avalia o cancelamento de interferência de sinal. Quando múltiplos sensores enviam sinais para o mesmo *gateway*, pode haver interferência/sobreposição de sinais, colisões. Pacotes com maior nível de sinal tendem a ser interpretados com maior facilidade, sofrendo menos com o efeito de colisões. Os *gateways* são posicionados de forma que, juntos, eles atendam ao número máximo de pares de sensores.

A minimização de latência é abordada também por (AOUN et al., 2006) em estudo sobre o posicionamento de *gateways* em WMN com restrições de QoS. Para minimizar a latência (atraso), os autores citam que é preciso determinar um raio de *cluster* ou uma profundidade máxima de árvore entre origem e *gateway*. Eles propõem um algoritmo recursivo denominado de *Recursive_DS*, caracterizado como um *polynomial time near-optimal algorithm*. O algoritmo divide o conjunto de nós em *clusters* disjuntos. Nesses

clusters, um nó atua como *gateway* tendo os demais nós conectados a ele. Em cada *cluster*, uma *spanning tree* com raiz no *gateway* é usada para agregar e assegurar o tráfego/encaminhamento de mensagens. O algoritmo é baseado no conceito de *dominant set* (DS), fazendo uso de aproximações recursivas do problema de mínimo DS. Além disso, usam um algoritmo guloso para fazer a seleção do nó que será o centroide de um *cluster*. Os autores usam ILP para formular o problema e mostrar que o problema de posicionamento de *gateways* é NP-hard. Para o posicionamento, avaliam o efeito da carga no *relay*, e efeitos do tamanho e do raio do *cluster*.

O uso combinado de algoritmos de *machine learning* e algoritmo heurístico é abordado pelos autores em (HE et al., 2017b) para o posicionamento de controladores e para a otimização de redes de comunicação (*data-driven optimization*). Primeiramente, o modelo proposto usa um algoritmo heurístico que estabelece uma solução inicial e, na sequência, predições com algoritmos de *machine learning* podem resultar em soluções de melhor qualidade e com tempo de execução reduzido. Eles caracterizam o problema como um problema das p-Medianas (*weighted controller placement problem (WCPP)*) e usam um algoritmo guloso para minimizar a função objetivo, implementado como um algoritmo de busca local que iterativamente altera a posição de controladores e avalia a solução. Algoritmos de *machine learning* são explorados para aprender a partir da distribuição de tráfego observada no passado e serem usados em processo de classificação *multilabel*. Os algoritmos experimentados incluem *Decision Tree* (DT), *Classification and Regression Trees* (CART), *Neural Network* (NN) e *Logistic Regression*. O estudo é aplicado a redes SDN/*communication networks* e usam como elementos de avaliação no processo de otimização, principalmente, a intensidade de tráfego em cada nó da rede. Adicionalmente, avaliam a latência entre os nós, o conjunto de chaves (*switches*, ou roteadores) e o conjunto de controladores (*controllers*). Como resultado, apresentam o conjunto de nós indicando a melhor posição para os controladores.

Os autores em (LANGE et al., 2015) propõem um *framework* denominado *Pareto-based Optimal COntroller placement* (POCO) que sugere um posicionamento "*pareto optimal*" respeitando diferentes métricas de desempenho. O estudo é direcionado a redes SDN, cujo princípio chave consiste em separar o plano de dados do plano de controle. As características de rede consideradas pelo processo incluem a latência (entre nós e controladores, e entre controladores), balanceamento de carga e tolerância a falhas (resiliência em relação a interrupções do controlador). Para pequenas redes, o *framework* realiza uma busca exaustiva pelas posições possíveis. Para redes maiores, faz uso de abordagem heurística com acurácia menor, mas que alcança um resultado em curto espaço de tempo computacional. Essa abordagem heurística do domínio da otimização combinatorial multi-objetivo (*multiobjective combinatorial optimization* (MOCO)), é denominada de *Pareto Simulated Annealing* (PSA).

Em (MATNI, 2020) os autores apresentam um método denominado de *Place*, que propõe uma estratégia de posicionamento ótimo de *gateways* em redes LoRaWAN que usa o método estatístico Gap para definir o número de *gateways* (*clusters*) e o algoritmo *Fuzzy C-Means* para o posicionamento. Utilizam o *packet delivery ratio* (PDR) como principal elemento avaliado pelo método. Computam *CAPEX* e *OPEX* para as simulações realizadas e procuram atender a requisitos de *QoS*. Uma variação do método, denominada de *DPLACE* é apresentada pelos autores em (MATNI et al., 2020). Nessa versão, os autores utilizam o *Fuzzy C-Means* para determinar a quantidade de *clusters*, o método estatístico *GAP* para o posicionamento e o *k-Means* para ajustar as posições finais de forma a resultar em menores valores de *CAPEX* e *OPEX*, mantendo bom nível de *PDR* e *Packet Delay*. Os autores ainda avaliam valores de *RSSI* para estimar se o *gateway* teria alcance para se comunicar com os dispositivos no entorno. Consideram *RSSI* e a distância entre os dispositivos para efetuar a clusterização, além de avaliar *PDR* e *Packet Delay*.

O posicionamento de *gateway* levando em conta a eficiência energética é explorado pelos autores em (PATIL; GOKHALE, 2021). Para o posicionamento de *gateways* os autores usam um método de otimização baseado, especialmente, em distância (Euclidiana e de *Manhattan*), levando também em consideração para a definição da posição final dos *gateways*, a distância entre CDs (*coordinating devices*) e *gateways*, o *throughput*, o consumo de energia, o balanceamento de carga e a capacidade do enlace. O método é executado em dois estágios: no 1^o estágio, é feita a seleção de posição candidata usando a distância Euclidiana e, no 2^o estágio, é feita a seleção da localização do *gateway* usando a distância de *Manhattan*.

O posicionamento de *gateways* de forma a aumentar a capacidade da rede *backhaul* pela minimização do número médio de saltos é abordado pelos autores em (CHAUDHRY et al., 2020). O estudo tem aplicação em redes *5G* ultra-densas. Os métodos utilizados pelos autores incluem clusterização (com o uso de algoritmos *k-Means* e *k-Medoids*) e algoritmo *Dijkstra* usado para encontrar o número médio de saltos e para associar *small cells* a *gateways* pela identificação do menor caminho. As características da rede avaliadas pelo método proposto incluem o número de saltos, o *throughput* e o número de transmissões simultâneas.

Os autores em (MAHMOUD; ISMAIL; DARWEESH, 2020) abordam sobre o posicionamento de *gateways* em *C-RAN* (*Cloud Radio Access Network*, *Cloud RAN* ou *Centralized-RAN*) baseado em demanda de tráfego e *throughput*. Utilizam o algoritmo de triangulação *Delaunay* para identificar a topologia da rede. O algoritmo de posicionamento é analisado para rotas estáticas (*static, maximum occupancy*) e dinâmicas (*dynamic, partial occupancy*). Autores também propõem um algoritmo para tratar a recuperação de desastres. Os métodos utilizados pelos autores incluem *GLA* (*gateway locations algorithm*), que discute a resolução do problema de otimização da rede, a avaliação de Distância Euclidiana para

determinar a melhor rota e *GA-MST* (*genetic algorithm minimum spanning tree*) usado nos cenários de recuperação de desastre (*backup scenario*). Entre as principais informações utilizadas no posicionamento, estão a distância entre os nós e o tráfego na rede.

O posicionamento de *gateways* em solo (*ground segment*) em sistemas HTS (*high throughput satellite*) é estudado pelos autores em (CORNEJO et al., 2020) para obter alta disponibilidade e ótimo número de *gateways* redundantes. Utilizam técnica de *deep learning* que permite predizer a atenuação provocada por chuva pela análise de séries temporais de cada área de *gateway* em conjunto com cadeias de *Markov* para estabelecer o número de *gateways* e otimizar o segmento de solo. Essa técnica permite estabelecer um melhor mecanismo de *switch* entre *NGWs* (*nominal gateways*) e *PGWs* (*redundant gateways*). Os métodos utilizados pelos autores incluem *deep learning*, *LSTM* (*long-short term memory*) que é uma *RNN* (*recurrent neural network*), e *Markov chain*, e avaliam as seguintes informações da rede: capacidade do enlace, *rain attenuation* (atenuação da chuva) em *gateways NGW* e *PGW*, *Carrier-to-noise and interference ratio* (*CNIR*) e *uplink* do alimentador (*feeder uplink*).

O conceito de *gateway node placement problem* (*GNP*) é explorado por (WZOREK; BERGER; DOHERTY, 2021) com o objetivo de estabelecer o menor número possível de *gateways* de forma a satisfazer requisitos de QoS em ambientes de busca e resgate em rede *mesh*. Os autores avaliam o posicionamento de gateways em conjunto com o problema de *router node placement problem* (*RNP*), que faz o posicionamento de roteadores. As técnicas consideradas pelos autores incluem: estratégias de *clustering*, decomposição de área e abordagem heurística (*heuristic graph clustering technique*). O método proposto pelos autores é executado em duas etapas: a primeira etapa do algoritmo garante que os posicionamentos calculados dos nós do roteador para uma determinada região de implantação atendam aos objetivos do problema *RNP* e suas restrições. Isso significa que a configuração de rede *backbone WMN* resultante maximiza a cobertura da rede, mantendo sua conectividade e minimizando o número de *router nodes* usados. A segunda etapa (*GNP*) garante que o número de nós de *gateway* atribuídos seja mínimo e a divisão do grafo de topologia da rede em um conjunto de *clusters* disjuntos (sub-redes) satisfaça a três restrições de QoS: *RQoS* (*maximum communication delay*), *LQoS* (*maximum relay load for each RN, router node*) e *SQoS* (*gateway throughput*). Ainda sobre o método, os autores propõem o algoritmo *RRT-WMN* (onde *RRT = rapidly exploring random trees*, ou árvore aleatória de exploração rápida) e fazem uso combinado com uma abordagem heurística de agrupamento de grafos. Os autores aplicam o algoritmo *RRT-WMN* para resolver o posicionamento de roteadores. Em seguida, o grafo de topologia de rede resultante, juntamente com as restrições de *QoS*, são usados como entrada para uma abordagem de agrupamento de grafos (que integra o *weighted recursive dominating set algorithm*). Dentre as restrições consideradas pelos métodos estão: obstáculos, raio de alcance de sinal, atraso (*delay*), carga nos roteadores, *throughput* / capacidade dos *gateways*. Os autores

destacam que uma importante medida de desempenho da rede WMN é a conectividade da rede que quantifica o quão bem interconectados estão os roteadores. Eles indicam que a conectividade, inclusive, é mais importante que a cobertura da rede ou cobertura do cliente, pois garante que os nós dos roteadores estão interligados entre si. O problema de posicionamento de *gateways* é tratado pelos autores de forma combinada com o problema de posicionamento de roteadores. Em suma, o problema *GNP* trata de encontrar um número mínimo de *GNs* e seus posicionamentos para garantir um nível suficiente de *QoS* com base em critérios que influenciem diretamente as medidas de desempenho da rede, como atraso de comunicação, carga do roteador e limites de capacidade dos *gateways*.

A otimização de *throughput* é o objetivo do estudo de (LI et al., 2008) que aborda sobre o posicionamento de *gateways* em *WMN*. Levam em consideração o número de *gateways* a serem posicionados e o modelo de interferência da rede. O método proposto pelos autores pode ser estendido para redes *mesh* multi-canais e multi-rádios. Os autores propõem o posicionamento baseado num *grid*, avaliam diferentes posições para os *gateways* e selecionam a combinação que assegure maior *throughput*. A estratégia proposta é comparada com o posicionamento aleatório e com o posicionamento fixo. De acordo com os autores, o posicionamento baseado em *grid* foi o que apresentou melhor resultado nos experimentos realizados. Em relação ao método apresentado pelos autores, citam o uso de: *mixed integer linear programming* para a otimização/maximização de *throughput* (*routing problem*); *greedy algorithm* para *interference-free link scheduling*; e *grid-based gateway placement scheme* (que usa o método de programação linear usado para o *throughput* como ferramenta de avaliação) para seleção da posição. Para o processamento do posicionamento, as principais informações consideradas pelo método incluem a análise de alcance (para análise de alcance de interferência), *achieved flow* (relação entre o fluxo obtido em relação ao fluxo demandado, tratado como uma *constraint*), e *total schedule traffic*.

Os autores em (HAJDU; DáVID; KRéSZ, 2021) abordam sobre características de redes de sensores e o uso de métodos gulosos para o posicionamento levando em conta objetivos como: *GLL* (*global load objective*), que busca minimizar a soma das cargas na rede; *MIL* (*minimal load*), que busca minimizar a carga dos sensores mais críticos, com maior carga; e *BAL* (*balance load*), cujo melhor cenário ocorre quando todos os sensores possuem a mesma carga. As principais informações consideradas nesse processo, incluem a análise da carga nos nós e número de *gateways*.

O posicionamento de *gateways* em *LoRa Network*, levando em conta de forma conjunta os problemas de posicionamento de *gateway*, atribuição de fator de espalhamento e alocação de energia, é tratado pelos autores em (OUSAT; GHADERI, 2019). Eles estabelecem uma abordagem para definir a atribuição ótima de *spreading factor* para cada célula (conjunto de *end devices*) individual e então aplicam uma abordagem gulosa para instalar os *gateways*. Têm como objetivo maximizar a eficiência energética e *throughput*,

com posicionamento mínimo de *gateways* de forma a ter menos custos de implantação e operação. Formulam o problema como um *mixed-integer non-linear program* (*MINLP*), e usam uma abordagem heurística/gulosa (*greedy LoRaPlanning*) para o posicionamento de *gateways*. Entre os elementos de rede considerados pelo método, incluem: posições candidatas de *gateways*, posições dos nós *end devices*, fator de espalhamento (*spreading factor*) e potência de sinal.

O uso de redes tolerantes a atrasos (*delay tolerant networks, DTNs*) como *backbone* para comunicação em cidades inteligentes é explorado pelos autores em (MADAMORI, 2019). Eles avaliam o desempenho de redes usando algoritmos de problemas de *set-cover* e de maximização de influência. Têm como objetivo minimizar a quantidade de *gateways*, visto que o interesse é fazer o projeto de *low-cost smart cities* (cidades inteligentes de baixo custo). Avaliam o uso de abordagem gulosa (*maximal sensor Coverage, MSC*), que busca maximizar a cobertura dos *gateways* (abordagem *minimal set cover problem*), e abordagens heurísticas e gulosas para análise de latência, com os métodos *CELF-MDD* e *Greedy-MDD*. As principais características de rede avaliadas incluem: latência, rota e posição de sensores.

Abordagens meta-heurísticas são exploradas por (ALI, 2016) para o posicionamento de *gateways* em WMN. Para isso, utilizam GA e SA (*simulated annealing*), considerando o número de *gateways* e o número de saltos que os pacotes precisam trafegar entre a origem e o destino (*router/gateway*). Têm como objetivo minimizar a variação de saltos entre roteadores (MR, *mesh routers*) e *gateways* (*VAR-MR-IG-Hop*) para garantir que os *gateways* (*IG, Internet gateways*) estejam devidamente posicionados. Além do uso de *GA* e *SA*, os autores citam o uso de algoritmo de *Dijkstra* para calcular o menor caminho entre cada roteador e todos os *gateways* da rede. Entre as principais características de rede consideradas no processo de posicionamento, observam-se: o número de saltos e a quantidade de roteadores associados a cada *gateway*, de forma a alcançar melhor balanceamento de carga/tráfego na rede.

Os autores em (TANG; CHEN, 2017) abordam sobre o posicionamento de *gateway* baseado no agrupamento de grafos e no uso de algoritmo genético de reparo (*RGA, repairing genetic algorithm*) para trabalhar com tais grafos, de forma a reparar soluções inviáveis. O *RGA* difere do *GA* por detectar e reparar soluções inviáveis geradas pelas operações de cruzamento e mutação, além de se mostrar computacionalmente eficiente, com tempo reduzido de processamento quando comparado ao *GA*. As principais características da rede consideradas pelo método incluem: máximo número de saltos e capacidade dos *gateways*.

Uma abordagem para o posicionamento de *gateways* em redes *LoRaWAN* usando *PSO* modificado é explorada pelos autores em (NYIRENDA, 2021), introduzindo o uso de medidas de distâncias dos *gateways* na fase inicial e no tempo de voo. O objetivo do processo de otimização é obter partículas que alcancem os maiores valores de *PDR*. De

acordo com os autores, *gateways* apropriadamente distanciados asseguram alta cobertura da rede e aumento na taxa de entrega de pacotes. Denominam o método apresentado de *PSODIST* (*PSO* modificado para considerar distâncias) e as principais características de rede avaliadas incluem a PDR e a distância entre *gateways*.

Os autores em (NANDA; KUMAR, 2016) estudam o posicionamento de *gateways* em rede *mesh* híbrida, com interface *2G/3G* com a Internet em alguns nós para alívio de carga, quando essa está alta em determinado *cluster* da rede. Têm como objetivo principal melhorar *QoS* sem, necessariamente, posicionar *gateways* que seriam subutilizados em parte de seu tempo. O trabalho avalia o conceito de *offloading clusters* (*clusters* de descarregamento) e os valores de *QoS* do *gateway* de acordo com os *clusters* de descarregamento. A abordagem apresentada pelos autores incluem o uso do método estatístico *ARIMA* (*autoregressive integrated moving average*), treinado e usado para predizer a carga no *grid* no próximo intervalo de tempo, denominado de época, e o uso de *GSO* (*glowworm swarm optimization*, otimização por colônia/enxame de vagalume) usado para definir a melhor posição dos *gateways*. Quanto às informações de rede avaliadas, consideram: carga predita para determinado ponto da rede, carga atual em determinado ponto e a posição dos *clusters* de descarregamento (*offload clusters*).

O posicionamento de controladores em *SDN* de redes híbridas *5G-Satellite* é explorado pelos autores em (TORKZABAN; BARAS, 2021). Fazem isso usando *MILP* (*mixed integer linear programming*) com objetivo de minimizar a probabilidade média de falhas em caminhos de controle da *SDN* e garantir que os *switches* da *SDN* vão receber as instruções de forma confiável. O método objetiva, também, implantar os controladores *SDN* próximos aos *gateways* de satélite para garantir que a conexão entre as duas camadas ocorra com a menor latência. Para o cálculo de menor caminho entre cada par de nós, autores usam *Yen's algorithm*. Usam *MILP* para posicionamento de *gateways* e comparam com um algoritmo guloso. A latência e a probabilidade de falhas nos caminhos de controle são as principais características avaliadas.

A necessidade de prever a expansão da rede é considerada pelos autores em (LOH et al., 2021). Para isso, avaliam a capacidade do *gateway* deixando reserva para acomodar novos sensores quando necessário. Os autores usam uma abordagem baseada em *capacitated geometric set cover problem* (*combinatorial optimization problem set cover*) e avaliam o *QoS* baseado na investigação da probabilidade de colisão. O objetivo geral é minimizar os *spreading factors* necessários ao transmitir pacotes *LoRa* para todos os sensores sem sobrecarregar a rede com *gateways*. Além disso, os autores observam que limitar o número de sensores por *gateway* no posicionamento tem pequena influência no resultado. Eles utilizam os métodos *ILP* e *VoronoiLocalSearch* (*GeometricLocalSearch*), e usam como informações de rede a posição dos nós sensores e o alcance de sinal de sensores e *gateways*. A metodologia utilizada consiste de três etapas: na primeira etapa, avaliam as entradas

geradas pelos sensores, as posições possíveis para a instalação de *gateways*, o alcance de transmissão e o número de sensores por *gateway*; na segunda etapa, parâmetros de transmissão *LoRa* são avaliados e os sensores são mapeados para *gateways* que estiverem mais próximos; na etapa três, é avaliada a probabilidade de colisão de pacotes de cada sensor e da rede como um todo.

3.1.3 Posicionamento de dispositivos de comunicação com o uso de outras técnicas

Nesta seção são descritas as referências cuja abordagem não evidencia o uso de uma técnica de IA/ML como ferramenta principal ou complementar para a resolução do problema descrito pelos autores de cada trabalho. De um modo geral, pode-se evidenciar que os métodos alternativos às técnicas de IA/ML incluem, especialmente, os relacionados à área de programação linear, sendo a programação linear inteira ou programação linear inteira mista as encontradas com mais frequência nas referências que abordam o problema de posicionamento de dispositivos.

Os autores em (FATEH; GOVINDARASU; AJJARAPU, 2013) apresentam um estudo sobre o projeto de rede para monitoramento de linhas de transmissão de energia, quanto ao posicionamento de equipamentos com o objetivo de minimizar custos operacionais e custos de instalação, além de satisfazer requisitos de latência e largura de banda. Consideram uma rede hierárquica composta de rede cabeada, rede sem fio e tecnologia celular e formulam um modelo ILP para definir a posição ótima para a instalação de torres de transmissão celular, a serem usadas no monitoramento de linhas de transmissão de energia. Avaliam a variação na largura de banda de fluxo, variação na latência do fluxo, tamanho de rede, variação na cobertura celular, período operacional, custo baseado na utilização do enlace, falta de confiabilidade do enlace e implantação incremental.

Técnicas de programação linear são encontradas, também, no trabalho dos autores (ZHEN et al., 2019) que apresentam um estudo sobre o posicionamento de concentradores em projetos de expansão de *smart grids*, maximizando *QoS* e fazendo análise de *path loss propagation model* para estabelecer raio de comunicação. O estudo objetiva maximizar a capacidade de *buffer* residual da rede sob os impactos de restrições orçamentárias, requisitos de conectividade de rede, limitações de distância devido a *path loss* (perda de percurso) e enlaces de comunicação redundantes para aumentar a robustez de uma rede de comunicação de rede inteligente em expansão. Utilizam métodos baseados em *mixed integer non-linear programming* e *mixed integer linear programming*, denominados de *SGEP-A* (que maximiza a capacidade residual média do *buffer*), *SGEP-MM* (que maximiza a capacidade residual mínima do *buffer*), *SGEP-R* (que visa minimizar a capacidade total de *buffer* residual recíproco). Para a análise de *path loss propagation model*, utilizam o *SUI model* para estabelecer o modelo de perdas associado às características da região em

análise.

Um modelo de otimização pode ser utilizado para o posicionamento de *gateways* de forma a balancear o tempo de vida da rede e o custo de implantação (maximização do tempo de vida e minimização do custo da rede). Isso é demonstrado pelos autores em (LIU et al., 2011) que modelam o problema e utilizam uma nova métrica de desempenho que definem como *RLC* (*ratio of lifetime to cost*), que leva em consideração a energia consumida para a transmissão de pacotes e o tempo de vida da rede (*network lifetime*).

A resiliência da rede é um tema pouco explorado. Sobre isso, os autores em (BOONKAJAY et al., 2021) destacam a existência de referências sobre o posicionamento de *gateways* quanto a *QoS* e balanceamento de carga, mas não identificaram estudos quanto a resiliência da rede. Os autores, então, apresentam um trabalho sobre o posicionamento de *gateways* de emergência e propõem o uso de uma interface NB-IoT (*narrowband Internet-of-Things*) para que alguns medidores inteligentes possam agir como *gateways* de emergência em redes Wi-SUN. O posicionamento é definido com o uso de uma abordagem de *ILP problem*. A previsão de *gateways* de emergência é importante para manter a rede operacional em situações em que a interferência de sinal possa estar comprometendo a operação normal da rede. Entre as características de rede utilizadas pelos autores, destacam-se: a quantidade de medidores, a diversidade de caminhos, o número de saltos entre medidores e *gateways*, a proximidade dos *clusters* (medido pelo *path loss* médio entre nós e os *gateways* de emergência) e o consumo de energia.

Os autores em (HELLER; SHERWOOD; MCKEOWN, 2012) apresentam um estudo teórico sobre o problema de posicionamento de controladores em redes *SDN*. Discutem o tema apresentando e discutindo métricas de avaliação e cenários de utilização do processo de instalação de controladores em diferentes posições da rede. Citam que é importante avaliar, por exemplo, quantos controladores são necessários, qual a melhor posição para esses controladores em determinada topologia, e sugerem verificar se um controlador é eficiente, se o posicionamento afeta a latência e quais vantagens/desvantagens identificadas em cada topologia experimentada. Quanto a métricas de posicionamento, sugerem avaliar a latência média da rede, a latência do pior caso, bem como o posicionamento de controladores de forma a maximizar o número de nós dentro de certa faixa de latência. Utilizam ILP para fazer experimentos que complementam o estudo teórico do artigo.

O problema de posicionamento de controladores em *SDN* é tratado por (YAO et al., 2014) como um *capacitated placement problem* (*CPP*) que avaliam a carga nos controladores para definir sua melhor posição. Consideram que um algoritmo de k-centro (*k-center algorithm*) nem sempre é aplicável. Sugerem, então, considerar um *capacitated K-problem* e mostram que o uso dessa abordagem é capaz de alcançar uma latência menor que implementações baseadas em CPP. Um modelo de programação inteira é utilizado para encontrar o número mínimo de controladores com um raio r específico. Utilizam

uma abordagem baseada em restrições, e consideram a carga dos controladores (devido a limitação da capacidade do servidor, latência do processamento da mensagem, ou falha). Para isso, procuram minimizar a latência média. Em resumo, a avaliação é feita sob o ponto de vista de carga e de raio (máxima distância/latência de cada *switch* ao controlador). De acordo com os autores, a diferença do problema clássico é que a carga de cada vértice não é constante com base na análise da última subseção. Uma variante do problema é minimizar a latência média em vez da latência máxima. Os autores discutem, principalmente, como minimizar a latência máxima de forma a evitar que alguns *switches* fiquem muito longe de seus controladores atribuídos.

O posicionamento de controladores em redes *SDN* considerando fluxos dinâmicos de tráfego é apresentado por (HE et al., 2017a), com o objetivo de minimizar o tempo de *setup* do fluxo. Os autores levam em consideração as latências do processo de *setup* e implementam uma abordagem de *MIP* (*mixed integer programming*). Eles também comparam o modelo proposto *CTR-SW*, que considera a movimentação de controlador e *switches* (interruptores) ao mesmo tempo, com outros dois modelos (*CTR* e *SW*) que consideram ou a movimentação de controlador ou a movimentação de *switch*, separadamente. O modelo *CTR* (*Controller-Topology-Route model)* é usado num cenário em que a atribuição *switch-to-controller* é fixa. Os domínios de controle permanecem inalterados enquanto se avalia o tempo necessário para estabelecer novas conexões de fluxo na rede SDN considerando diferentes posições do controlador. Duas restrições estabelecidas para a solução garantem que haja apenas um controlador para cada domínio de controle e as chaves de domínio sejam atribuídas a esse controlador. No modelo *SW* (*Switch-Based model*) é considerado um cenário em que os controladores permanecem em seus lugares e os interruptores mudam suas atribuições. Uma restrição estabelecida para a solução fixa a localização de cada controlador.

3.2 MÉTODOS IDENTIFICADOS PARA O POSICIONAMENTO E TAXONOMIA

Nesta seção é apresentado um resumo com a lista dos principais métodos e propriedades de rede utilizados pelas referências consultadas para o processo de posicionamento de *gateways* ou problemas de planejamento. A Figura 14 mostra a distribuição de métodos utilizados pelas diferentes referências consultadas.

Dada a diversidade de métodos observados nas referências, a Figura 15 procura concentrar os métodos em categorias com o objetivo de identificar aquelas que mais se destacam. É importante mencionar que uma mesma referência pode fazer uso de diferentes abordagens como, por exemplo usar, um método de clusterização para um agrupamento inicial dos equipamentos e o uso de método heurístico para avaliar se o agrupamento gerado inicialmente pode ser otimizado quanto a alguma métrica.

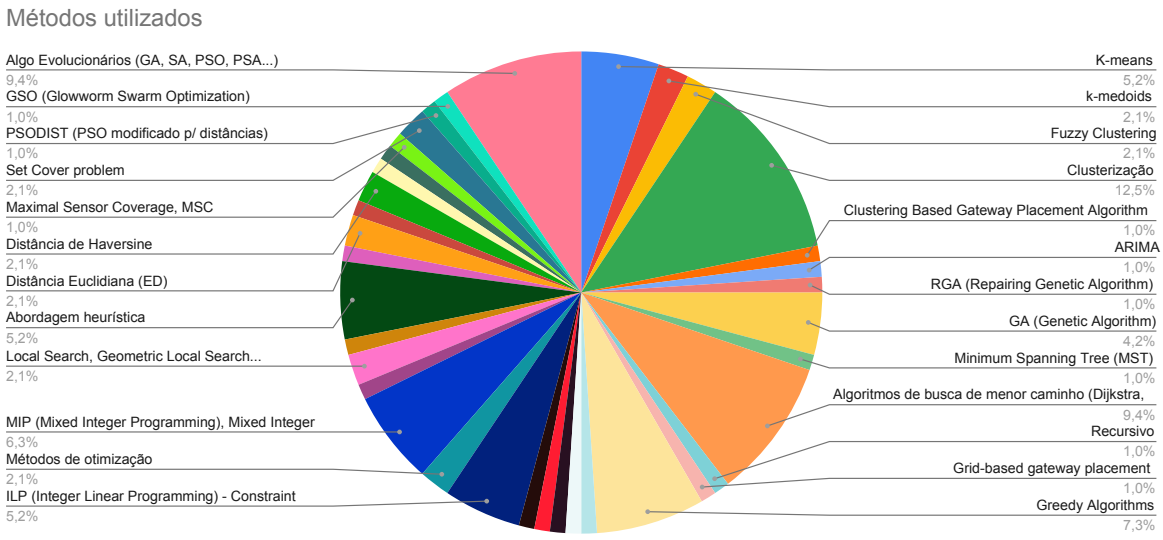


Figura 14 – Classificação das referências por tipo de método utilizado.

Fonte: Autoria própria.

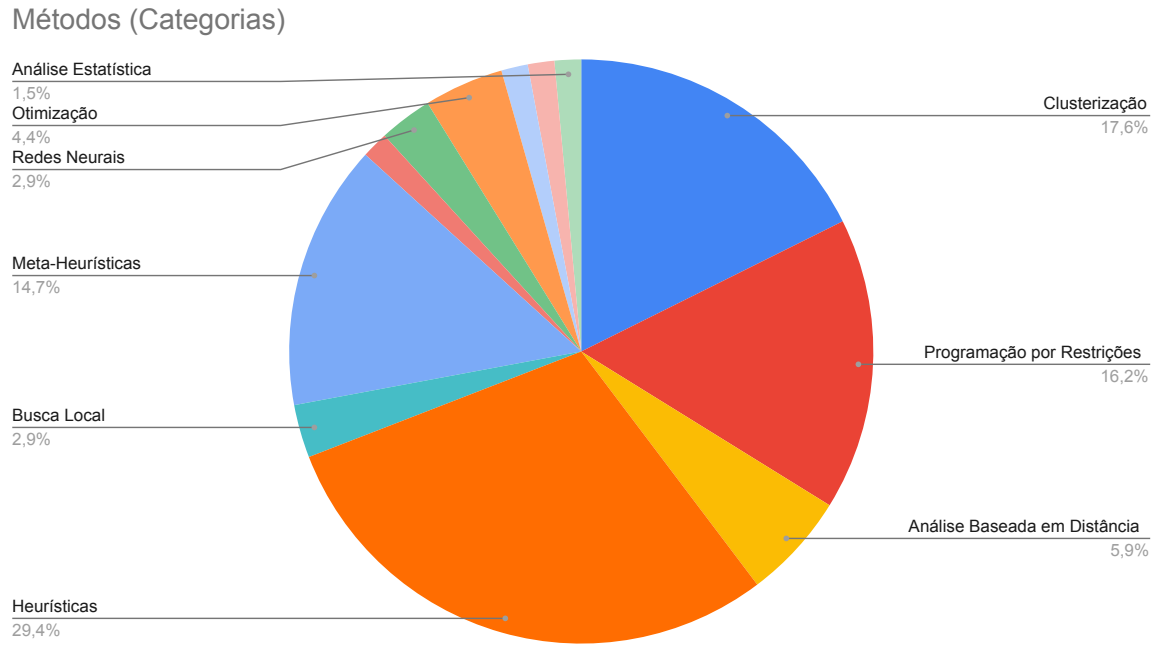


Figura 15 – Classificação das referências por categoria de método utilizado.

Fonte: Autoria própria.

Em relação aos elementos avaliados no processo de posicionamento de dispositivos, várias características ou métricas podem ser levadas em consideração. A Figura 16 dá um panorama geral dos principais elementos utilizados pelas referências.

Uma vez identificados os métodos utilizados com maior frequência na literatura consultada para resolução de problemas de posicionamento de *gateway*, este estudo propõem

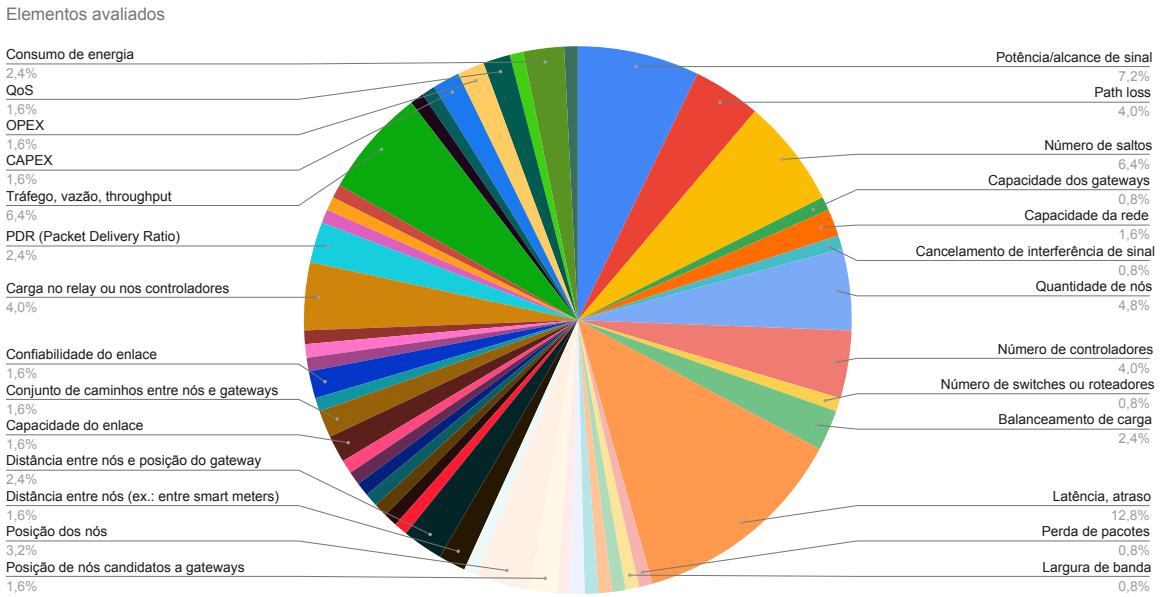


Figura 16 – Classificação das referências por elemento avaliado/utilizado no processo de posicionamento ou planejamento.

Fonte: Autoria própria.

uma taxonomia que inclui a categoria e identificação de métodos utilizados/citados ou propostos pelas referências consultadas.

As Tabelas 1 e 2 apresentam a lista de referências consultadas com as indicações de método e categoria do método citado ou proposto pela referência. Destacam-se os métodos heurísticos, meta-heurísticos, clusterização e programação por restrições (*constraint programming*) como os mais utilizados.

Tabela 1 – Lista de métodos de posicionamento de dispositivos de comunicação por categoria.

Início da Tabela de Lista de Métodos por Categoria		
Categoria	Método	Referências que citam, utilizam ou propõem o método
(HE) Heuristics	Minimum Spanning Tree (MST)	(MAHMOUD; ISMAIL; DARWEESH, 2020)
	Algoritmos de busca de menor caminho (Dijkstra, BFS, Bellman-Ford, Floyd-Warshall, Yen's algorithm.)	(RAITHATHA et al., 2021), (SOUZA et al., 2013), (FERREIRA et al., 2015), (CHAUDHRY et al., 2020), (GALLARDO; AHMED; JARA, 2021), (WANG et al., 2018), (ALI, 2016), (TORKZABAN; BARAS, 2021), (WANG et al., 2017)
	Recursivo	(AOUN et al., 2006), (FERREIRA et al., 2015)
	Grid-based gateway placement	(LI et al., 2008)
	Greedy Algorithms	(TIAN; WEITNAUER; NYENGELE, 2018),(AOUN et al., 2006),(LI et al., 2008),(HAJDU; DáVID; KRéSZ, 2021),(OUSAT; GHADERI, 2019),(MADAMORI, 2019), (TORKZABAN; BARAS, 2021)
	Greedy-MDD (Greedy Minimal Delivery Delay)	(MADAMORI, 2019)

Continuação da Tabela 1 - Lista de Métodos por Categoria		
Categoria	Método	Referências que citam, utilizam ou propõem o método
	CELF-MDD (Cost-Effective Lazy Forward Minimal Delivery Delay)	(MADAMORI, 2019)
	Set Cover problem	(LOH et al., 2021), (MADAMORI, 2019)
	Decomposição de área	(WZOREK; BERGER; DOHERTY, 2021)
	Abordagem heurística	(LANGE et al., 2015), (TANAKORNPINTONG et al., 2017),(WZOREK; BERGER; DOHERTY, 2021),(OUSAT; GHADERI, 2019),(MADAMORI, 2019)
(MH) Metaheuristics	RGA (Repairing Genetic Algorithm)	(TANG; CHEN, 2017)
	GA (Genetic Algorithm)	(RAITHATHA et al., 2021), (MAHMOUD; ISMAIL; DARWEESH, 2020), (XING et al., 2016), (MAHDY et al., 2017)
	Pareto Simulated Annealing (PSA)	(LANGE et al., 2015)
	PSODIST (PSO modificado para considerar distâncias)	(NYIRENDA, 2021)
	GSO (Glowworm Swarm Optimization)	(NANDA; KUMAR, 2016)
	Algoritmos Evolucionários (GA, SA, PSO, PSA...)	(RAITHATHA et al., 2021), (SILVA, 2012), (LANGE et al., 2015), (ALI, 2016), (TANG; CHEN, 2017), (NYIRENDA, 2021), (NANDA; KUMAR, 2016), (XING et al., 2016), (MAHDY et al., 2017)
(CL) Clustering	k-Means	(RAITHATHA et al., 2021), (FERREIRA et al., 2015), (TANAKORNPINTONG et al., 2017), (MATNI et al., 2020), (CHAUDHRY et al., 2020)
	k-Medoids	(CHAUDHRY et al., 2020), (GALLARDO; AHMED; JARA, 2021)
	Fuzzy Clustering	(MATNI, 2020), (MATNI et al., 2020)
	Clusterização	(RAITHATHA et al., 2021), (AOUN et al., 2006), (FERREIRA et al., 2015), (TANAKORNPINTONG et al., 2017), (MATNI, 2020), (MATNI et al., 2020), (CHAUDHRY et al., 2020), (GALLARDO; AHMED; JARA, 2021), (WZOREK; BERGER; DOHERTY, 2021), (WANG et al., 2018), (TANG; CHEN, 2017), (WANG et al., 2017)
	Clustering Based Gateway Placement Algorithm	(WANG et al., 2017)
(CT) Constraint Programming/Linear Programming	ILP (Integer Linear Programming), Constraint Programming	(FATEH; GOVINDARASU; AJJARAPU, 2013), (AOUN et al., 2006), (YAO et al., 2014), (MADAMORI, 2019), (BOONKAJAY et al., 2021)
	MIP (Mixed Integer Programming), Mixed Integer non-Linear programming (MINLP), Mixed Integer Linear Programming (MILP)	(HE et al., 2017a), (ZHEN et al., 2019), (LI et al., 2008), (OUSAT; GHADERI, 2019), (TORKZABAN; BARAS, 2021), (MAHDY et al., 2017)
(SL) Supervised Learning	DT (Decision Tree)	(HE et al., 2017b)
(DB) Distance-Based Analysis	Distância Euclidiana (ED)	(PATIL; GOKHALE, 2021), (MAHMOUD; ISMAIL; DARWEESH, 2020)
	Distância de Manhattan (MD)	(PATIL; GOKHALE, 2021)
	Distância de Haversine	(GALLARDO; AHMED; JARA, 2021),(WANG et al., 2018)
(NN) Neural Networks	LSTM (RNN)	(CORNEJO et al., 2020)
	NN (Neural Networks), ANN (Artificial NN)	(HE et al., 2017b)

Continuação da Tabela 1 - Lista de Métodos por Categoria		
Categoria	Método	Referências que citam, utilizam ou propõem o método
(OP) Optimization method	Métodos de otimização	(HELLER; SHERWOOD; MCKEOWN, 2012), (LIU et al., 2011)
	Maximal Sensor Coverage, MSC	(MADAMORI, 2019)
(LS) Local Search, Geometric Local Search	VoronoiLocalSearch	(LOH et al., 2021)
	Local Search, Geometric Local Search...	(HE et al., 2017b), (LOH et al., 2021)
(RG) Regression model	LR (Logistic Regression)	(HE et al., 2017b)
(ST) Statistical Analysis	ARIMA	(NANDA; KUMAR, 2016)
(PR) Probabilistic model	Cadeias de Markov	(CORNEJO et al., 2020)
Final da Tabela 1 - Lista de Métodos por Categoria		

A Tabela 3, adaptada de (MOCHINSKI et al., 2022), apresenta um comparativo entre o método AIDA, proposto por este estudo, e outros métodos encontrados na literatura. A tabela evidencia as abordagens mais usuais e destaca a experimentação com dados de larga-escala utilizada nos testes com o método AIDA, enquanto o mais comum é o uso de base de dados reduzidas ou a experimentação com dados sintéticos.

3.3 DISCUSSÃO

3.3.1 Características/elementos de rede considerados para o posicionamento de gateways

Os métodos baseados em IA ou métodos de *machine learning* mais utilizados pelas referências são os métodos de clusterização, abordagens heurísticas e meta-heurísticas. Especificamente, sobre métodos de *machine learning*, é possível notar um uso mais expressivo de técnicas de clusterização, ou aprendizagem não-supervisionada. Quanto às técnicas de aprendizagem supervisionada e redes neurais, elas geralmente são utilizadas como técnicas acessórias ou complementares em conjunto com outras estratégias.

Técnicas baseadas em *constraint programming*, como *ILP* e suas extensões aparecem em grande número nas referências, citadas muitas vezes como método principal, ou então como método para estabelecer a modelagem formal do problema. É de conhecimento da literatura que problemas de posicionamento de *gateways* são classificados como problemas NP-Hard devido à grande quantidade de restrições geralmente associadas ao cenário de aplicação. O posicionamento de *gateways*, de um modo geral, procura estabelecer o melhor local para a instalação dos equipamentos de comunicação da rede. Para isso, diferentes elementos ou diferentes características da rede podem ser levados em consideração no momento de avaliar o resultado do método proposto. Na Figura 16 é possível observar que as características de interesse principal (as citadas com mais frequência) incluem: latência ou atraso (12,8%); potência ou alcance de sinal (7,2%); tráfego, vazão ou *throughput* (6,4%); número de saltos (6,4%); quantidade de nós (4,8%); número de controladores ou gateways

Tabela 2 – Referências consultadas e categorias dos métodos citados em cada referência.

Ref	HE Heuristics	MH Metaheuristics	CL Clustering	CT Constraint Programming	SL Supervised Learning	NN Neural Networks	DB Distance-based Analysis	OP Optimization Method	LS Local Search	PR Probabilistic Model	RG Regression Model	ST Statistical Analysis
(RAITHATHA et al., 2021)	✓	✓	✓									
(TIAN; WEITNAUER; NYENGELE, 2018)	✓											
(SILVA, 2012)		✓										
(FATEH; GOVINDARASU; AJJARAPU, 2013)				✓								
(AOUN et al., 2006)	✓		✓	✓								
(HELLER; SHERWOOD; MCKEOWN, 2012)								✓				
(HE et al., 2017b)					✓	✓			✓		✓	
(YAO et al., 2014)				✓								
(HE et al., 2017a)				✓								
(LANGE et al., 2015)	✓	✓										
(SOUZA et al., 2013)	✓											
(FERREIRA et al., 2015)	✓		✓									
(TANAKORNPINTONG et al., 2017)	✓		✓									
(ZHEN et al., 2019)				✓								
(MATNI, 2020)			✓									
(MATNI et al., 2020)			✓									
(PATIL; GOKHALE, 2021)							✓					
(LIU et al., 2011)								✓				
(CHAUDHRY et al., 2020)	✓		✓									
(MAHMOUD; ISMAIL; DARWEESH, 2020)	✓	✓					✓					
(CORNEJO et al., 2020)						✓				✓		
(GALLARDO; AHMED; JARA, 2021)	✓		✓				✓					
(WZOREK; BERGER; DOHERTY, 2021)	✓		✓									
(WANG et al., 2018)	✓		✓				✓					
(LI et al., 2008)	✓			✓								
(HAJDU; DáVID; KRéSZ, 2021)	✓											
(LOH et al., 2021)	✓								✓			
(OUSAT; GHADERI, 2019)	✓			✓								
(MADAMORI, 2019)	✓			✓				✓				
(ALI, 2016)	✓	✓										
(TANG; CHEN, 2017)		✓	✓									
(NYIRENDA, 2021)		✓										
(NANDA; KUMAR, 2016)		✓										✓
(TORKZABAN; BARAS, 2021)	✓			✓								
(XING et al., 2016)		✓										
(MAHDY et al., 2017)		✓		✓								
(BOONKAJAY et al., 2021)				✓								
(WANG et al., 2017)	✓		✓									

a serem posicionados (4,0%); *path loss* (4,0%); carga nos nós controladores (4,0%); posição dos nós (3,2%); consumo de energia (2,4%).

Tabela 3 – Comparativo de características entre o método AIDA e referências sobre posicionamento de dispositivos de comunicação em *smart grids*.

	Referências													
	AIDA	(WANG et al., 2017)	(MAHDY et al., 2017)	(GALLARDO; AHMED; JARA, 2021)	(LIU et al., 2011)	(ROLIM et al., 2018)	(AALAMIFAR et al., 2014)	(SOUZA et al., 2013)	(KONG, 2019)	(LANG et al., 2022)	(INGA et al., 2018)	(PIRAK et al., 2017)	(STIRI et al., 2022)	(ZHEN et al., 2019)
Heurística	✓					✓	✓	✓	✓	✓	✓			
Meta-Heurística			✓										✓	
Particionamento de Rede		✓		✓										
Clusterização	✓	✓	✓	✓			✓	✓	✓	✓		✓	✓	
Modelagem como Problema de Programação Linear/Não-linear			✓			✓	✓			✓				✓
Cobertura de Conjuntos						✓			✓		✓			
Localização de Instalações				✓									✓	✓
Atribuição de Rota														✓
Modelo Analítico					✓									
Modelo de Propagação com Análise do Perfil Topográfico do Terreno	✓													
Modelo de Propagação com Análise Simplificada do Terreno			✓			✓	✓		✓	✓		✓	✓	✓
Uso de Postes como Posições Candidatas	✓					✓	✓	✓		✓	✓	✓		✓
Prioriza o Uso de Postes com Equipamentos de Automação	✓													
Total de medidores (Experimento com Dados Reais)	234797	294		891		29002		67			381	31		
Total de medidores (Experimento com Dados Sintéticos)			348		81	N.D.*	17121		24011	8020			5000	275

* N.D. - Não Disponível. Fonte: Adaptado de (MOCHINSKI et al., 2022)

3.3.2 Tendências

Ao analisar as abordagens utilizadas pelos autores das referências consultadas, fica evidente que o uso de clusterização, heurísticas e meta-heurísticas, bem como técnicas de programação linear, são dominantes na resolução de problemas de posicionamento de *gateways*.

A utilização de técnicas de redes neurais tem sido experimentada em diferentes cenários da comunicação sem fio, resultado do aumento na capacidade de processamento dos computadores modernos. Dispositivos com funções mais especializadas (como os sensores, por exemplo), apesar de apresentarem capacidade computacional maior que as apresentadas por dispositivos similares mais antigos, devem priorizar a eficiência energética como um dos elementos a serem avaliados durante o seu funcionamento, restringindo assim, tarefas mais complexas (mais consumidoras de energia) a centros de controle ou dispositivos especializados. Apesar da existência de literatura sobre o uso de redes neurais e redes neurais profundas em redes de comunicação sem fio (exemplos: (MOCANU, 2017), (CORNEJO et al., 2020), (TESTI et al., 2019), (HE et al., 2017b)), geralmente as aplicações de tais técnicas não têm foco na definição de posições de dispositivos de comunicação, mas sim como instrumentos para o controle, análise de demandas de uso energético, tendências de consumo de energia, entre outros. Com isso, o uso de abordagens de clusterização e modelos baseados em heurísticas ou meta-heurísticas devem se manter como técnicas principais no processo de posicionamento de roteadores/*gateways*.

Com o uso cada vez mais expressivo de sistemas de comunicação sem fio, com geração elevada de dados a serem processados, pode-se dizer que uma tendência a ser explorada é a ampliação da utilização de técnicas de processamento de fluxos de dados, para que o processamento de informações e tomada de ação possam ocorrer em tempo de resposta mais rápidos. Um exemplo de aplicação é apresentado pelos autores em (DHARMADHIKARI et al., 2021), que explora temas como segurança, análise de consumo de energia e eficiência energética.

3.4 CONSIDERAÇÕES FINAIS

A pesquisa desenvolvida neste estudo para identificação da literatura recente existente em relação ao uso de técnicas de IA/*machine learning* para posicionamento de *gateways* mostra que não existe uma técnica dominante a ser adotada. A diversidade de tecnologias de rede e cenários de aplicação certamente justificam essa constatação, uma vez que existem particularidades de cada estudo em relação à área e cenário de aplicação.

A grande variedade de tecnologias se evidencia nos estudos mais recentes para atender a necessidades de diferentes cenários de aplicação, como no uso de sensores ou outros dispositivos de IoT, redes *5G*, dispositivos LoRa, entre outros. A demanda cada

vez maior pela busca de um consumo mais equilibrado de energia também estimula o desenvolvimento e adoção de novas tecnologias.

Fica evidente, pelo resultado da pesquisa efetuada, que não foram identificadas referências que utilizem a análise de características de um cenário de *smart grid* como base para o treinamento e classificação com métodos de *machine learning* para a definição de posições de roteadores e *gateways*. Por esse motivo, a pesquisa efetuada procurou identificar as abordagens mais usuais para o posicionamento de forma a deixar claro que a proposta apresentada neste estudo é inovadora.

Técnicas clássicas como o uso de técnicas de otimização como abordagens de *constraint programming* se fazem presentes na literatura, juntamente com outras abordagens. Nenhuma técnica, no entanto, parece ser adotada como padrão dominante, inclusive abrindo possibilidades para a exploração para o uso combinado de diferentes técnicas. Apesar disso, técnicas de clusterização e abordagens heurísticas/meta-heurísticas apresentam grande representatividade de uso nos cenários avaliados, sugerindo que técnicas baseadas em IA/ML apresentam relevante importância para resolução de problemas de posicionamento de *gateways*.

4 MÉTODO AIDA

Este capítulo apresenta uma proposta de método analítico para o projeto preliminar de redes de comunicação em redes elétricas inteligentes (*smart grids*), na camada que integra medidores inteligentes, roteadores e *gateways*. Mais especificamente, avalia uma determinada região geográfica e propõe as posições de equipamentos de comunicação para o processo de comunicação entre a região NAN de um *smart grid* e *gateways* responsáveis pela integração com a região WAN onde o centro de operação de distribuição está instalado.

O método AIDA está descrito detalhadamente no artigo "*Towards an Efficient Method for Large-Scale Wi-SUN-Enabled AMI Network Planning*" (MOCHINSKI et al., 2022), publicado no jornal MDPI Sensors, acessível pelo link <https://doi.org/10.3390/s22239105>. No artigo, experimentos são realizados com 4 regiões do estado do Paraná. Além disso, o método é avaliado com dois modelos de perda para o cálculo de potência recebida: o *Delta-Bullington* e o *Erceg-SUI propagation model* (SUI *model)*, implementado tal como apresentado pelos autores em (ERCEG et al., 1999) e (ERCEG et al., 2001) (IEEE 802.16.3c-01/29r4, seção *Suburban Path Loss model*). Neste documento, por sua vez, os experimentos são realizados com uma base que contém 26 municípios e a análise de desempenho avalia os resultados das duas diferentes abordagens de clusterização implementadas pelo método.

Em uma rede de comunicação AMI, posicionar os principais dispositivos (roteadores e *gateways*) é uma tarefa complexa. Inicialmente, requer conhecimento sobre a posição de medidores inteligentes, dispositivos de automação de distribuição e postes. Conhecer as informações técnicas sobre a tecnologia de comunicação e a área geográfica também é importante. O planejamento de uma rede AMI envolve preocupações particulares com relação aos custos e desempenho geral da rede de comunicação, pois o posicionamento adequado dos principais dispositivos pode reduzir significativamente o custo total de implantação. Portanto, ao projetar uma rede de comunicação para um *smart grid*, além de priorizar a conectividade dos medidores inteligentes, também é importante levar em consideração as posições dos dispositivos de automação da distribuição. A rede de comunicação para gerenciamento dos equipamentos DA deve ter alto desempenho e confiabilidade, sendo assim essencial ter sempre em mente a importância de sua conexão com a rede *backhaul*.

Além disso, uma avaliação mais detalhada das conexões entre dispositivos na fase de planejamento ajuda a antecipar a ocorrência de *enlaces* potencialmente não confiáveis durante a fase de implementação. Na pesquisa desenvolvida para a proposição do método apresentado nesta seção, o interesse consiste em propor um método para posicionar os dispositivos de comunicação, apresentando um bom *trade-off* entre a cobertura da rede

e o número de dispositivos de comunicação a serem instalados. Conforme apresentado na Seção 3 e destacado por Mochinski et al. (2022), a literatura existente explora o tema apresentando diferentes técnicas para o posicionamento de dispositivos. Ainda assim, raras são as referências que exploram cenários reais de grande escala ou que exploram meticulosamente a viabilidade de comunicação entre medidores e equipamentos de comunicação considerando a topografia detalhada entre os dispositivos. Adicionalmente, diferentemente de outros métodos da literatura, o método de posicionamento proposto nesta seção procura priorizar posições candidatas em que estejam instalados equipamentos DAs.

AIDA (acrônimo para *AI-driven AMI network planning with DA-based information and a link-specific propagation model*)[1] é um método heurístico e iterativo que visa minimizar o número de medidores não conectados a cada iteração do método. Durante a execução, um conjunto de medidores não conectados é submetido ao método para ser processado a cada iteração. Após o término da iteração corrente, a porcentagem de medidores não conectados (P_u) é avaliada. Uma nova iteração é iniciada se essa porcentagem for maior que o critério de parada (P_u^{max}), que estabelece a quantidade máxima de medidores não conectados aceitos pela simulação. Uma lista de medidores não conectados restantes é processada na execução subsequente. O fato de o método admitir um resultado em que nem todos os medidores estão conectados deriva da razão de o mesmo poder ser utilizado para uma análise ou estudo preliminar, em que se pode tolerar um certo nível de não cobertura de sinal.

As principais contribuições obtidas com o desenvolvimento do método AIDA incluem: (i) A proposição de um método eficiente de planejamento da camada de comunicação de uma rede AMI com foco em larga escala. Para avaliar isso, os experimentos realizados com o método utilizam um conjunto com dados reais de 26 municípios do estado do Paraná e um total de 466.237 medidores inteligentes. A exploração de cenários de grande escala não é comum na literatura e permite avaliar o método proposto em condições reais, com características de terreno distintas de forma a avaliar o desempenho do método em regiões urbanas e áreas rurais, com diferentes concentrações de medidores (regiões com alta densidade e regiões esparsas); (ii) Para o cálculo de potência recebida estimada no enlace, o estudo avalia a utilização de um modelo de propagação que não depende da classificação empírica do terreno por fazer uso de um modelo capaz de avaliar, de forma detalhada, o perfil do terreno para o cálculo das perdas por difração; (iii) A aplicação de uma heurística baseada em *grid* para determinar as posições candidatas para a instalação dos equipamentos de comunicação visa minimizar o número de posições de postes a serem avaliadas. Adicionalmente, o método propõe o uso de um mecanismo simplificado para a análise de conectividade com múltiplos saltos (*multihop*) baseado no uso de uma árvore

[1] Em tradução livre, AIDA pode ser lido como “Planejamento de rede AMI orientado por técnicas de IA com informações baseadas em posições de DAs e um modelo de propagação específico”.

geradora mínima para reduzir o número de conexões a serem analisadas. As estratégias selecionadas visam equilibrar complexidade e qualidade da solução final. É importante destacar que o método AIDA faz uso de uma seleção de posições candidatas baseada em *grid* que prioriza o uso de posições de dispositivos de automação, combinada com uma heurística MST para explorar as conexões de múltiplos saltos e minimizar o número de conexões a serem analisadas. O objetivo de priorizar a utilização de postes com dispositivos de automação visa possibilitar, sempre que possível, o posicionamento de roteadores e *gateways* em locais próximos à rede *backhaul*.

O método proposto explora algoritmos de *machine learning* não supervisionados (no caso, algoritmos de agrupamento), algoritmos de processamento de grafos (árvore geradora mínima), algoritmos de busca exaustiva e de busca gulosa. A potência média recebida estimada no enlace é utilizada como base para a análise de conectividade. O LRP é calculado usando um modelo de propagação detalhado, que inclui o cálculo de perda de difração que leva em consideração o perfil topográfico entre as coordenadas geográficas dos pontos na análise do enlace. O método *Delta-Bullington* (International Telecommunication Union, 2013) é usado para calcular a perda de difração usando as características do perfil do terreno entre as coordenadas. O método estima o LRP com base nas características técnicas dos dispositivos, incluindo potência de transmissão, sensibilidade do receptor e ganhos das antenas.

4.1 PROBLEMA DE OTIMIZAÇÃO MULTIOBJETIVO

O método caracteriza-se como um problema de otimização multiobjetivo e busca atingir as funções objetivo descritas nesta subseção. A primeira função objetivo visa maximizar o valor LRP obtido para uma conexão entre um medidor inteligente v_i e uma posição de um equipamento de comunicação (no caso, um roteador ou *gateway*) k_j (4.1):

$$\text{maximizar } LRP_{v_i \to k_j}, \quad \forall \ v_i \in \{\mathcal{V}\}, \ k_j \in \{\mathcal{K}\} \tag{4.1}$$

onde $\mathcal{V} = \{v_1, \ldots, v_u\}$ é o conjunto de medidores inteligentes e $\mathcal{K}$ é o conjunto de equipamentos de comunicação posicionados. Para reduzir os custos de instalação, a segunda função objetivo visa minimizar o número de equipamentos posicionados (4.2).

$$\text{minimizar } |\mathcal{K}| \tag{4.2}$$

onde $\mathcal{K}$ representa um subconjunto do conjunto de posições candidatas $\mathcal{C} = \{c_1, \ldots, c_z\}$. Considerando os equipamentos de comunicação posicionados, a terceira função objetivo visa maximizar o número de equipamentos instalados em postes com equipamentos de automação da distribuição, DA (4.3):

$$\text{maximizar } |\mathcal{K}_{DA}| \tag{4.3}$$

onde $\mathcal{K}_{DA} \subseteq \mathcal{K}$. Posicionar um equipamento de comunicação em postes com DA minimiza o custo de configuração, pois garante que o dispositivo será instalado em um ponto onde a rede de *backhaul* já foi configurada. A quarta função objetivo maximiza o número de medidores inteligentes conectados a um equipamento de comunicação para construir o conjunto de SMs conectados e otimizar o uso de cada equipamento (4.4):

$$\text{maximizar } |\mathcal{V}_{con_{k_j}}| \tag{4.4}$$

onde $\mathcal{V}_{con_{k_j}}$ é igual ao número de medidores inteligentes $v_i \in \{\mathcal{V}\}$ conectados ao dispositivo $k_j \in \{\mathcal{K}\}$. A quinta função objetivo maximiza o valor médio de LRP para a solução (4.5):

$$\text{maximizar } \left(\frac{1}{n}\sum_{i=1}^{n} LRP_{v_i \to k_j}\right), \ \ \forall \ v_i \in \{\mathcal{V}\}, \ k_j \in \{\mathcal{K}\} \tag{4.5}$$

onde n é o número de medidores inteligentes conectados. A última função objetivo minimiza a porcentagem de medidores inteligentes não conectados, P_u, avaliados a cada iteração do método pelo critério de parada (4.6). O percentual máximo deve ser ajustado de acordo com os requisitos do projeto, e é definido como parâmetro para a execução do método:

$$\text{minimizar } P_u, \ \ \forall \ v_i \in \{\mathcal{V}\}, \ k_j \in \{\mathcal{K}\} \tag{4.6}$$

onde $P_u \leq P_u^{max}$, e P_u^{max} representa o critério de parada das iterações do método.

O posicionamento de roteadores e *gateways* é considerado um problema NP-Hard, caracterizado por múltiplas funções objetivo, exigindo resolução complexa. Ele pode ser tratado como um problema das p-Medianas (Seção 2.3) e o uso de uma abordagem heurística orientada por IA visa reduzir sua complexidade e alcançar uma solução para o problema em um tempo de processamento razoável. Portanto, é útil no planejamento de cenários de grande escala.

4.2 ESTRUTURA DO MÉTODO AIDA

AIDA é um método que visa minimizar o número de medidores inteligentes não conectados a cada iteração do método. Um conjunto de SMs não conectados é submetido ao método para ser processado a cada iteração. Após o término da iteração atual, a porcentagem de SMs não conectados (P_u) é avaliada. Uma nova iteração é iniciada se esta porcentagem for maior que o critério de parada (P_u^{max}), que estabelece o máximo de SMs não conectados aceitos pela simulação. Uma lista de SMs não conectados restantes é processada na execução subsequente. O método AIDA inclui as etapas indicadas na Figura 17.

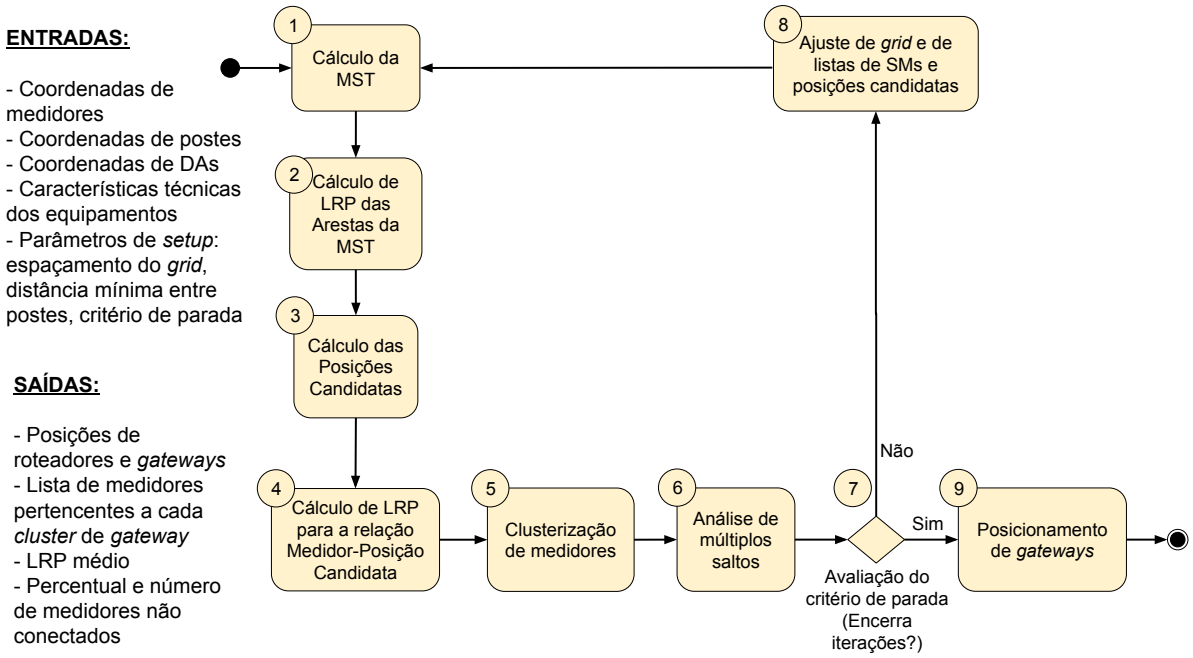


Figura 17 – Etapas do método AIDA.

Fonte: Adaptado de (MOCHINSKI et al., 2022).

4.2.1 Etapa 1 – Cálculo da MST

Nessa etapa, uma única árvore geradora mínima baseada nas coordenadas geográficas dos SMs é computada usando um algoritmo *Boruvka* de árvore dupla (*DualTreeBoruvka algorithm*) proposto pelos autores em (MARCH; RAM; GRAY, 2010). Para o cálculo da MST, todos os medidores inteligentes da iteração (que correspondem aos medidores não conectados na iteração anterior do método) são considerados como vértices de um grafo.

A estratégia de utilizar uma MST visa identificar os medidores inteligentes que sejam vizinhos mais próximos entre si e, com isso, minimizar o número de conexões possíveis a serem avaliadas. Trata-se de uma estratégia com foco na aplicação do método em cenários de larga escala.

4.2.2 Etapa 2 – Cálculo de LRP das arestas da MST

Após a construção da MST interligando os medidores da iteração, o cálculo de LRP é executado para todas as arestas da árvore geradora mínima construída na Etapa 1 para identificar a potência recebida estimada no enlace que conecta cada par de medidores da árvore.

Os valores de LRP computados para a MST são classificados em três categorias: (1) Arestas azuis indicam enlaces de alta qualidade; (2) Arestas em cor laranja indicam enlaces de qualidade média, com possibilidade de conexão incerta; (3) e Arestas em vermelho, indicam enlaces de baixa qualidade, que não apresentam condições de conexão.

As faixas de valores que estabelecem os limites para enlaces de alta, média e baixa qualidade são especificados na Tabela 5. Com as arestas da MST coloridas conforme as faixas estabelecidas, é possível visualizar um mapa de calor que apresenta um cenário geral da qualidade de conexão por toda a região, conforme mostra a Figura 18. A Equação (2.7) é considerada para o cálculo do LRP, conforme apresentado na Seção 2.4.

4.2.3 Etapa 3 – Cálculo das posições candidatas

O conjunto de posições candidatas para instalação de equipamentos de comunicação (roteadores/*gateways*) é estabelecido com base nas posições dos postes com e sem dispositivos DA. O método AIDA usa uma abordagem de *grid* para fazer a seleção otimizada de um subconjunto de postes e minimizar o esforço computacional necessário para escolher as posições mais ideais para a colocação dos roteadores e *gateways*. Além disso, opta pela utilização de um *grid* para estabelecer a cobertura uniforme de toda a região. Normalmente, a área territorial das cidades é irregular, apresentando uma concentração (densidade) variada de postes e medidores, incluindo áreas urbanas muito densas e áreas rurais esparsas. Abordagens baseadas em *grids* são capazes de apresentar bons resultados, como a apresentada pelos autores em (LI et al., 2008).

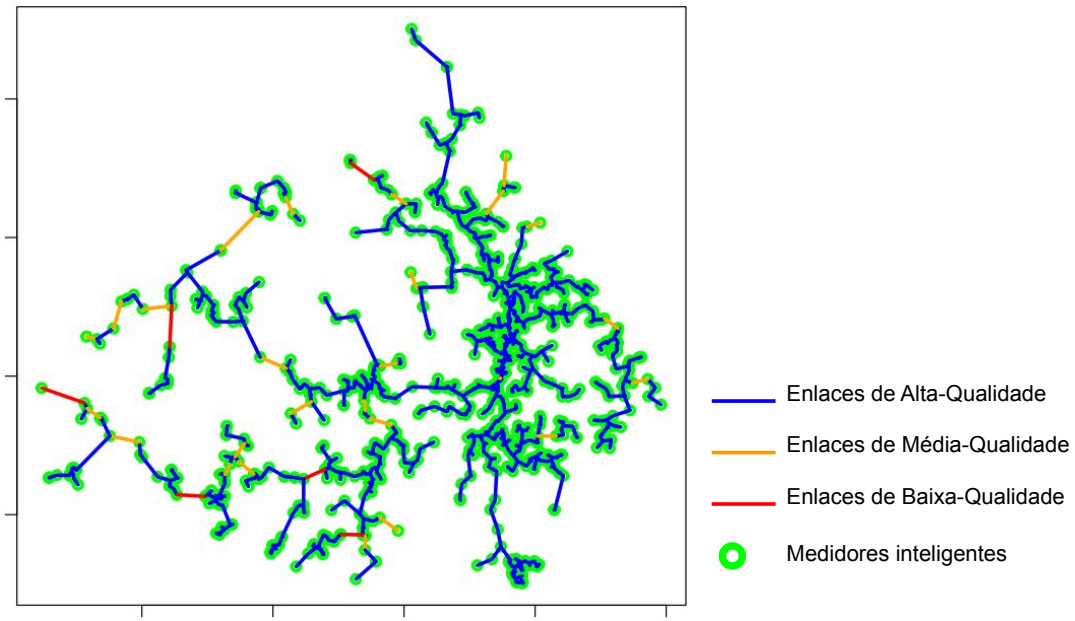


Figura 18 – Exemplo de MST com arestas coloridas de acordo com os valores de LRP.
Fonte: Adaptado de (MOCHINSKI et al., 2022).

Primeiramente, a área da região é dividida em uma grade com o mesmo espaçamento horizontal e vertical, considerando um alcance teórico de transmissão para os dispositivos roteadores/*gateways*. Uma vez estabelecida a grade inicial, os pontos da grade são ajustados para as posições mais próximas dos postes disponíveis na região (Figura 19), priorizando o uso de postes com DA (função objetivo (4.3)).

O método tenta manter os pontos da grade dentro de uma distância mínima de separação para minimizar a alocação de posições candidatas. Pontos da grade que estejam muito longe dos postes são descartados. A resolução da grade é ajustada a cada iteração do AIDA para diminuir a distância entre os pontos. A cada nova iteração do método, os postes selecionados para serem posições de CP (*candidate position*, posição candidata) em iterações anteriores são ignorados.

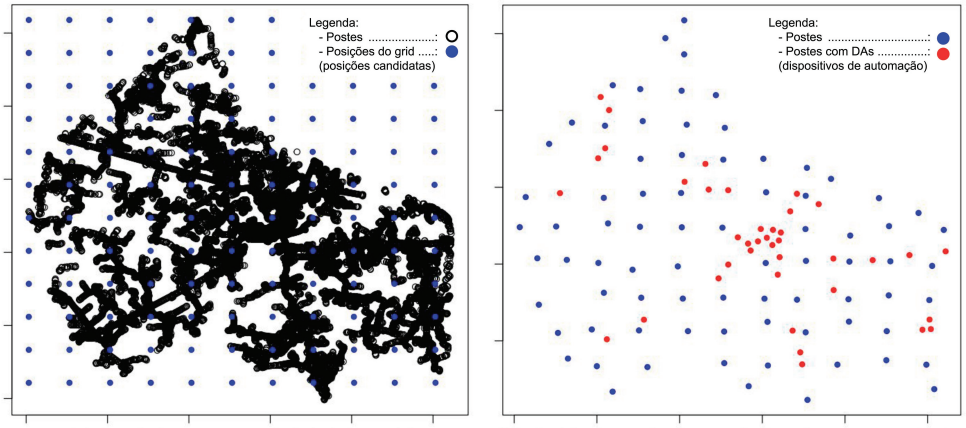


Figura 19 – Grid inicial e posicionamento de posições candidatas.

Fonte: Adaptado de (MOCHINSKI et al., 2022).

4.2.4 Etapa 4 – Cálculo de LRP para a relação medidor–posição candidata

Essa etapa calcula os valores de LRP entre cada medidor inteligente e as posições candidatas mais próximas dentro de um raio específico pré-estabelecido.

Considerando que o número de medidores inteligentes costuma ser alto e sabendo que as posições candidatas estabelecidas para cada iteração estão dispersas por toda a região, o método AIDA busca minimizar a quantidade de cálculos de LRP entre medidores e CPs. Para isso, estabelece um alcance teórico e calcula a potência recebida apenas para as posições dentro desse raio.

Inicialmente, para identificar as CPs dentro da faixa de cada medidor, são calculadas as distâncias entre cada medidor e todas as CPs da iteração. Depois disso, o LRP é calculado e armazenado em uma estrutura auxiliar apenas para as relações em que a distância entre o medidor e a CP forem menores que c_j^r, que corresponde ao raio de comunicação estabelecido para as posições candidatas. Esse raio de comunicação e outros parâmetros de interesse para o método AIDA são relacionados na Tabela 4 e, na Tabela 5, podem ser obtidas as características técnicas de equipamentos e valores considerados para o alcance de comunicação roteador/*gateway*, entre outros parâmetros utilizados na execução de experimentos com AIDA.

Nessa etapa, apenas os valores de LRP são calculados para todas as conexões possíveis. No entanto, nenhuma verificação de capacidade é realizada em relação ao número de conexões com as CPs. A verificação de capacidade é executada pelos processos de *clustering* (agrupamento) definidos na Etapa 5—Clusterização de medidores (Seção 4.2.5). Para o cálculo do LRP, é considerada a Equação (2.7) (ver a Seção 2.4 para mais detalhes).

Tabela 4 – Parâmetros de entrada e saída dos algoritmos.

Parâmetro	**Descrição**
$\mathcal{V} = \{v_1, \ldots, v_u\}$	Conjunto de medidores inteligentes, v_i
$\mathcal{C} = \{c_1, \ldots, c_z\}$	Conjunto de CPs, c_j, onde $c_j =< c_j^n, c_j^r >$
c_j^n	Número de SMs conectados a c_j
c_j^r	Raio de alcance de comunicação de c_j
$\mathcal{L}$	Conjunto de valores de LRP, l_{v_i,c_j}, para os enlaces $v_i \leftrightarrow c_j$
$\mathcal{V}_{con}$	Conjunto de SMs conectados a CPs, $< v_i, c_j, l_{v_i,c_j}^{max} >$
$\mathcal{M}, \mathcal{N}$	Subconjuntos de $\mathcal{L}$
$\mathcal{Q}$	Subconjunto de $\mathcal{C}$
l_{min}	Valor de LRP mínimo para estabelecer uma conexão
n_{max}	Número máximo de SMs por CP
$dist(v_i, c_j)$	Distância entre SM v_i e a CP c_j
c_s	CP selecionada para conectar um SM v_i
n_v	Contador de v_i no alcance de uma CP c_j

Tabela 5 – Características técnicas de equipamentos e parâmetros para o funcionamento do método AIDA.

Parâmetro	**Descrição**
Frequência de operação Wi-SUN	920 MHz
Potência de transmissão do medidor inteligente (P_{tx})	26 dBm
Ganho de antena do medidor inteligente (G_{tx})	2 dBi
Altura de instalação da antena do medidor inteligente	1.5 m
Ganho de antena do roteador/*gateway* (G_{rx})	6.25 dBi
Altura de instalação do roteador/*gateway*	7 m
Alcance de comunicação do roteador/*gateway*	3000 m
Número máximo de conexões aceitas pelo roteador/*gateway* (n_{max})	2000
Separação mínima entre postes e postes com DAs	1000 m
Valor mínimo de LRP para estabelecer uma conexão (l_{min})	-95 dBm
Critério de parada (*Stopping criteria*)	$P_u^{max} = 2\%$
Número máximo de saltos (h_{max})	7
Critério para enlace de Alta-qualidade (*High-quality*, HQ)	LRP ≥ -95 dBm
Critério para enlace de Média-qualidade (*Medium-quality*, MQ)	$-105 \leq$ LRP < -95 dBm
Critério para enlace de Baixa-qualidade (*Low-quality*, LQ)	LRP < -105 dBm

4.2.5 Etapa 5 – Clusterização de medidores

Essa etapa se refere ao agrupamento de SMs a uma determinada posição candidata. Nesse processo, são avaliados os valores de LRP calculados na etapa anterior, assim como as conexões máximas aceitas pelas CPs (quantidade máxima de SMs que podem ser conectados a uma CP). Uma conexão para uma relação $SM \Leftrightarrow CP$ pode ser estabelecida se o LRP mínimo (l_{min}) for alcançado.

Duas abordagens heurísticas são avaliadas para o agrupamento: Bottom-up e Top-down. A cada iteração do AIDA, ambas as abordagens são executadas, e aquela que resultar no menor número de CPs será tomada como base para definir a lista de SMs e postes para a próxima iteração. Ambas as abordagens visam alcançar os objetivos expressos pelas funções objetivo (4.1), (4.2), (4.4) e (4.5).

As abordagens de agrupamento são descritas a seguir (a Tabela 4 contém informações sobre seus parâmetros de entrada e de saída):

- Abordagem Bottom–Up (BU): nessa abordagem (Algoritmo 1, Figura 20), uma estratégia de busca exaustiva é usada para avaliar os valores de LRP calculados para o enlace entre cada medidor inteligente e as CPs ao seu alcance. Uma conexão $SM \Leftrightarrow CP$ é estabelecida com a posição que apresentar o valor de LRP mais alto. Isso visa maximizar o valor de LRP entre o medidor inteligente e o roteador/*gateway* ao qual ele será conectado.

- Abordagem Top–Down (TD): nessa abordagem (Algoritmo 2, Figura 21), uma estratégia de busca gulosa é usada para conectar o número máximo de SMs a cada CP, apresentando $LRP \geq l_{min}$, priorizando a conexão aos SMs com maiores valores de potência recebida. Isso visa maximizar o uso da CP, conectando a ele o máximo de medidores possível, limitado a n_{max} (ver Tabela 4).

4.2.6 Etapa 6 – Análise de múltiplos saltos

As abordagens Bottom-Up e Top-Down estabelecem a conexão viável de um salto entre as posições candidatas e os medidores inteligentes. Os medidores inteligentes não conectados pelas abordagens descritas devem, então, se conectar a um *cluster* de SMs usando conexões de múltiplos saltos.

A MST calculada com as posições dos SMs é considerada para a análise teórica da conexão de múltiplos saltos. Primeiro, são identificados os medidores inteligentes pertencentes a *clusters* de SMs (ou seja, SMs já conectados a posições candidatas) e seus enlaces com arestas da MST. Então, para cada SM que já pertence a um *cluster*, é realizada uma busca por vizinhos adjacentes da MST (busca nos vértices da MST) que ainda não

Algoritmo 1 Abordagem Bottom–Up (BU) – Adaptado de (MOCHINSKI et al., 2022)

Entrada: $\mathcal{V}$, $\mathcal{C}$, $\mathcal{L}$, l_{min}, n_{max}
Saída: $\mathcal{V}_{con}$
$\mathcal{V}_{con} \leftarrow \{\}$
para todo $v_i \in \mathcal{V}$ **faça**
  $\mathcal{K} \leftarrow$ Selecione todo $c_j \in \{\mathcal{C} \mid (c_j^n < n_{max}) \wedge (dist(v_i, c_j) \leq c_j^r)\}$
  $l_{max} \leftarrow l_{min}$
  $c_s \leftarrow \{\}$
  para todo $c_k \in \mathcal{K}$ **faça**
    $l_{con} \leftarrow \{l_{v_i,c_k} | (l_{v_i,c_k} \in \mathcal{L})\}$
    se $l_{con} \geq l_{max}$ **então**
      $l_{max} \leftarrow l_{con}$
      $c_s \leftarrow c_k$
    fim-se
  fim-para
  se $c_s \neq \{\}$ **então**
    $\mathcal{V}_{con} \leftarrow \mathcal{V}_{con} \cup \{< v_i, c_s, l_{max} >\}$
    $c_s^n \leftarrow c_s^n + 1$
  fim-se
fim-para

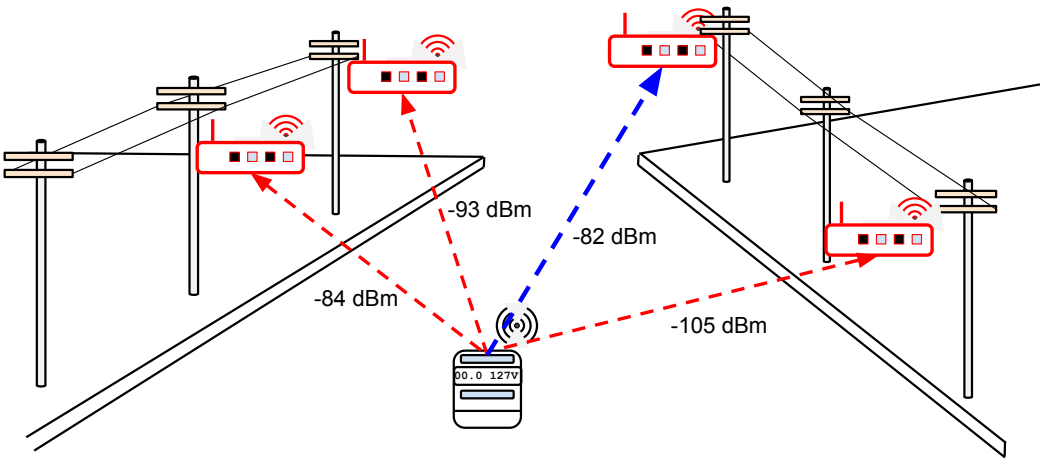


Figura 20 – Estratégia de conexão entre medidor e posição candidata utilizada pela abordagem Bottom-UP (BU).

estão conectados, mas têm valor de potência recebida suficiente para estabelecer a conexão (ou seja, correspondem a enlaces azuis da MST, com $LRP \geq l_{min}$). A busca é feita até o limite máximo de saltos (h_{max}) estabelecido pelo método. Vizinhos com $LRP \geq l_{min}$

Algoritmo 2 Abordagem Top–Down (TD) – Adaptado de (MOCHINSKI et al., 2022)

Entrada: $\mathcal{V}$, $\mathcal{C}$, $\mathcal{L}$, l_{min}, n_{max}
Saída: $\mathcal{V}_{con}$
$\mathcal{M} \leftarrow$ Selecione todo $v_i, c_j \in \{\mathcal{L} \mid l_{v_i,c_j} \geq l_{min}\}$
$\mathcal{Q} \leftarrow$ Selecione os distintos c_j de $\mathcal{M}$
$\mathcal{V}_{con} \leftarrow \{\}$
enquanto Verdadeiro **faça**
    $\mathcal{K} \leftarrow$ Selecione todo $c_j, (count(v_i)$ como $n_v) \in \{\mathcal{M} \mid n_{v_z} > n_{v_{z+1}}\}$
    se $\mathcal{K} \neq \{\}$ **então**
        $s_i \leftarrow \{k_1 \mid k_1$ é o primeiro c_j de $\mathcal{K}\}$
        $\mathcal{N} \leftarrow$ Selecione $(v_i, c_j)_n \in \{\mathcal{M} \mid (c_j = s_i) \wedge (n \in \{1, 2, .., n_{max}\}) \wedge (dist(v_i, c_j) \leq s_i^r) \wedge ((l_{v_i,c_j})_z > (l_{v_i,c_j})_{z+1})\}$
        $\mathcal{M} \leftarrow \mathcal{M} - \mathcal{N}$
        $\mathcal{Q} \leftarrow \mathcal{Q} - \{s_i\}$
        $\mathcal{V}_{con} \leftarrow \mathcal{V}_{con} \cup \{< v_i, s_i, l_{v_i,s_i} > \ \forall\ v_i \in \mathcal{N}\}$
    senão
        exit // sai do laço do enquanto
    fim-se
    se $\mathcal{Q} = \{\}$ **então**
        exit // sai do laço do enquanto
    fim-se
fim-enquanto

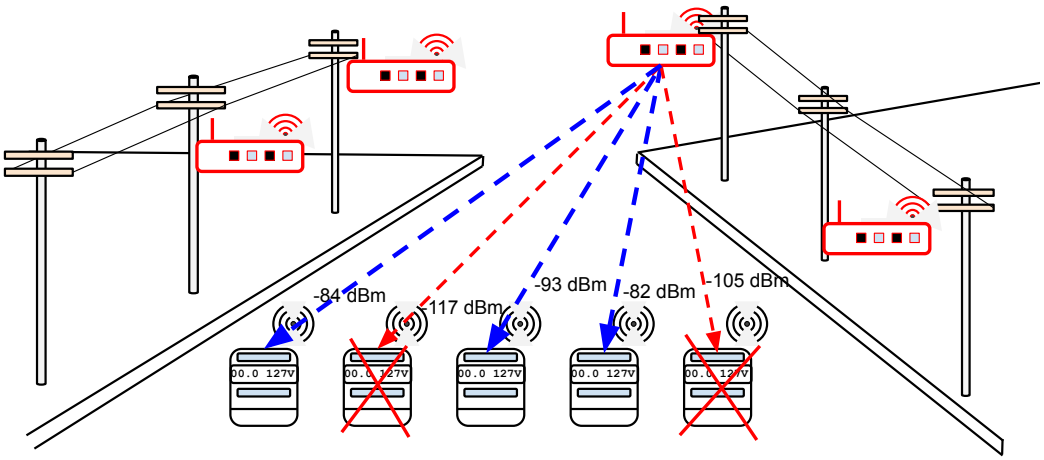


Figura 21 – Estratégia de conexão entre posição candidata e medidores utilizada pela abordagem Top-Down (TD).

são conectados ao *cluster* SM. Ao encontrar um vizinho com valor de potência recebida menor que o mínimo requerido, tal nó é descartado e permanece desconectado por não

apresentar condições técnicas para a conexão.

4.2.7 Etapa 7 – Avaliação do critério de parada

Para o método AIDA, uma nova iteração é contada para cada ciclo de execução do método que inclui os passos 1, 2, 3, 4, 5 e 6 (Figura 17). O número de iterações executadas pelo método depende do número de iterações necessárias para atingir a cobertura esperada (total de medidores inteligentes conectados), e irá variar de acordo com as condições da região, sua forma, número de postes, quantidade e concentração de medidores inteligentes.

Após cada iteração de AIDA, o método verifica se todos os SMs estão conectados ou se o critério de parada (Tabela 5) foi alcançado. Caso o percentual de SMs não conectados (P_u) seja maior que o estabelecido (P_u^{max}), uma nova iteração deve ser realizada para minimizá-lo (função objetivo (4.6)) e a Etapa 8—Ajuste de grid e de listas de SMs e posições candidatas é executado (Seção 4.2.8). Caso contrário, o método pode executar a análise de posicionamento de *gateways* (Etapa 9—Posicionamento de gateways, Seção 4.2.9).

4.2.8 Etapa 8 – Ajuste de grid e de listas de SMs e posições candidatas

O AIDA é um método iterativo e, após a execução de uma iteração, caso ainda não tenha sido atingido o percentual estabelecido de SMs não conectados (P_u^{max}), alguns parâmetros devem ser ajustados antes de iniciar a execução de outro método. A Figura 22 ilustra o processo iterativo apresentando, inicialmente, uma imagem com todos os medidores não conectados e, em seguida, imagens com as posições candidatas consideradas e os medidores conectados em cada iteração do método.

Os parâmetros a serem considerados em uma nova iteração incluem a lista de SMs que permaneceram desconectados na iteração anterior e a lista de CPs utilizadas. Com relação ao cálculo das CPs para a nova iteração, o *grid* utilizado para identificar as posições candidatas deve ser ajustado, geralmente adotando um novo espaçamento horizontal e vertical entre os pontos da grade que represente metade do valor usado na iteração anterior, resultando em um *grid* mais denso (conforme indicado na Tabela 6). A informação sobre as CPs utilizadas nas iterações anteriores é importante porque, para a nova iteração, as posições dos postes anteriormente considerados como CPs são ignoradas nas novas iterações.

Tabela 6 – Parâmetros de iteração para definição de grid e separação de postes.

Parâmetro	**Iteração 1**	**Iteração 2**	**Iteração 3**
Espaçamento do grid (km)	5.0	2.5	1.25
Distância mínima entre postes (km)	3.0	1.5	0.75
Distância mínima entre postes e postes com DA (km)	1.0	1.0	1.0

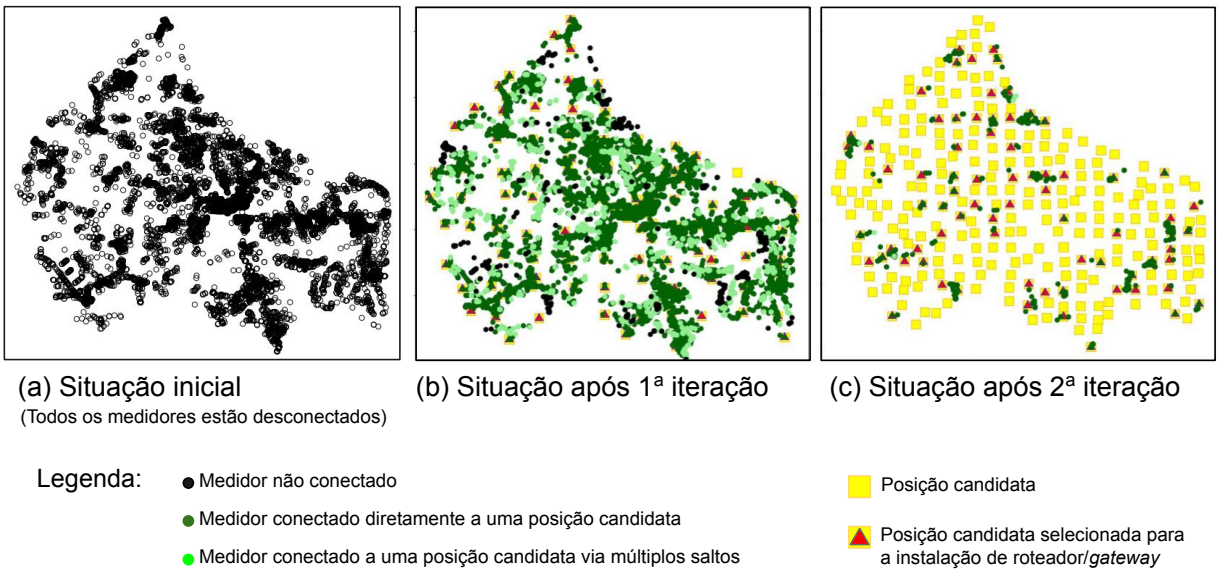


Figura 22 – Ilustração do processo iterativo do método AIDA.

Fonte: Adaptado de (MOCHINSKI et al., 2022).

4.2.9 Etapa 9 – Posicionamento de *gateways*

Com a execução das etapas anteriores, o método AIDA é capaz de identificar as melhores posições de postes para instalação de roteadores e *gateways*. Inicialmente, as posições de postes são selecionadas com a ajuda de uma grade e promovidas a posições candidatas. A partir dessas posições candidatas, algumas posições são selecionadas como as mais adequadas para o posicionamento dos principais dispositivos (ou seja, roteadores e *gateways*). Esta seleção de CPs é feita na Etapa 5—Clusterização de medidores do método AIDA (Seção 4.2.5). As CPs consideradas no posicionamento do *gateway* serão aquelas calculadas pela abordagem de agrupamento SM (*Bottom–Up* ou *Top–Down*) que minimiza efetivamente o número de CPs necessárias.

Na etapa de posicionamento do *gateway*, descrita nesta seção, as posições candidatas selecionadas para uma região (ver exemplo na Figura 23) são avaliadas para determinar se serão consideradas posições para instalação de roteadores ou *gateways*. A princípio, todas as posições candidatas selecionadas são consideradas posições válidas para a instalação de roteadores.

Posteriormente, um processo de agrupamento é usado para determinar o conjunto de roteadores que serão conectados ao mesmo *gateway*. Para isso, o algoritmo *Weighted K-means* (ou *k-Means* ponderado) (INDARJO, 2020) é usado para selecionar a lista de roteadores que devem ser incluídos no mesmo *cluster* e estabelecer a melhor posição para o *gateway* de cada grupo. A Figura 24 ilustra o resultado do processo de agrupamento dessa etapa. Os parâmetros de entrada desse algoritmo de agrupamento incluem a lista de CPs selecionadas (já consideradas como posições de roteadores), o número de medidores

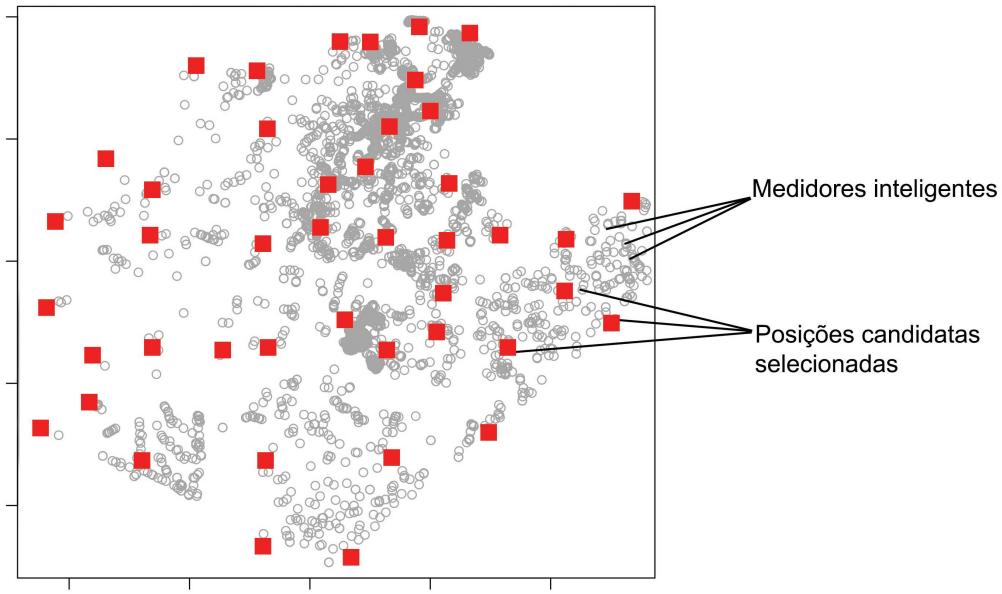


Figura 23 – Exemplo de cenário com a indicação das posições candidatas selecionadas pelo método AIDA.

Fonte: Adaptado de (MOCHINSKI et al., 2022).

conectados a cada posição candidata e o máximo de conexões aceitas por um *gateway*. O número de medidores conectados a cada CP define o peso de cada CP. O processo de agrupamento visa agrupar em um mesmo *cluster* uma lista das CPs mais próximas cuja soma de medidores conectados a cada uma não exceda o limite de conexões estabelecido para o *gateway*. Esse limite é um parâmetro determinado para a execução do método, mas pode variar de acordo com as características técnicas do equipamento.

Em algumas situações, pode ocorrer que um determinado grupo calculado pelo algoritmo *Weighted K-Means* contenha apenas uma posição candidata selecionada. Isso pode acontecer porque o número de medidores conectados à CP é igual ou próximo ao limite de medidores estabelecido para o *cluster*. Neste caso, esta posição será classificada como *gateway*. Em outros casos, o grupo será formado por um conjunto de roteadores e um *gateway* (selecionado dentre as posições das CPs no *cluster*). Após esse processo de agrupamento, as posições dos equipamentos de comunicação são estabelecidas e o método AIDA é concluído.

4.3 EXPERIMENTOS E RESULTADOS PRELIMINARES

Para experimentos com o método AIDA, foram utilizados os dados de 26 municípios do estado do Paraná. A lista de municípios considerados no estudo inclui as regiões (cidades) indicadas na Tabela 7. Desses municípios, são consideradas como dados de entrada para o método AIDA as coordenadas geográficas de medidores inteligentes e de postes. Além disso, são consideradas as posições de postes utilizadas para o posicionamento de dispositivos de

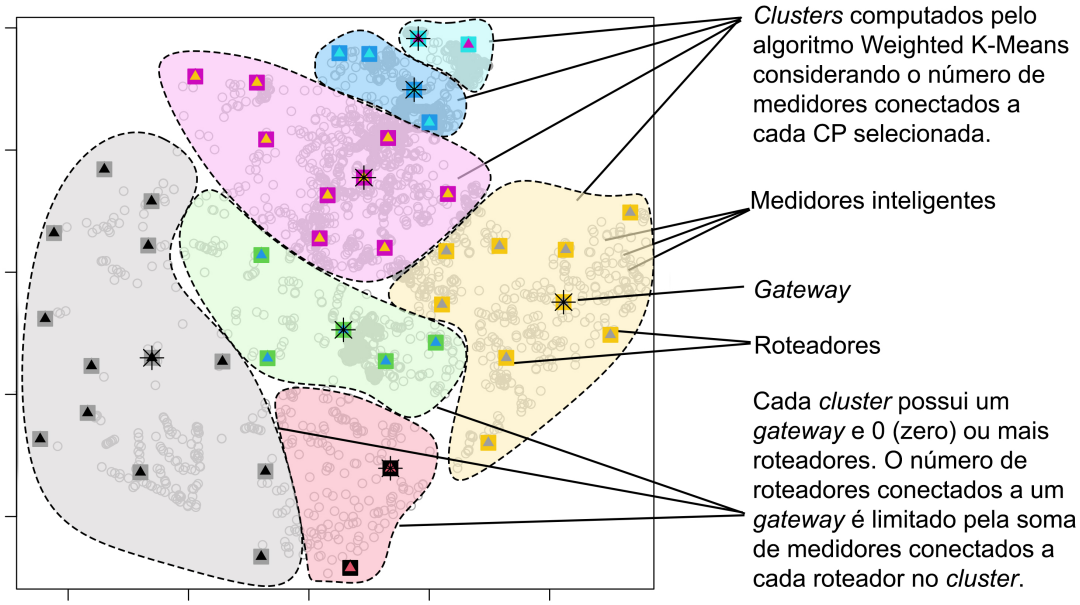


Figura 24 – Posicionamento de *gateways* pelo método AIDA. A figura apresenta os diferentes *clusters* (grupos) calculados pelo algoritmo *Weighted K-Means*. Cada grupo possui um *gateway* posicionado e pode ter zero ou mais roteadores.

Fonte: Adaptado de (MOCHINSKI et al., 2022).

automação da rede. A soma total de medidores inteligentes utilizados nos experimentos é de 466.237 medidores. O uso de posições reais dos dispositivos e de uma grande quantidade de medidores e postes servem como base para a avaliação do desempenho do método em cenários de larga escala.

Tabela 7 – Dados gerais das cidades utilizadas nos experimentos.

Região (Cidade)	Medidores	Postes	BB Area km^2	Medidores/km^2	Postes/km^2
AGUDOS DO SUL	5383	7357	459,0	11,7	16,0
ARAUCARIA	52836	18076	322,0	164,1	56,1
BALSA NOVA	6106	8250	622,6	9,8	13,3
CAMPO DO TENENTE	4224	7146	1057,5	4,0	6,8
CARAMBEI	7512	5310	659,8	11,4	8,0
CONTENDA	10319	13132	674,1	15,3	19,5
FAZENDA RIO GRANDE	56157	15754	177,2	316,9	88,9
GUAMIRANGA	3727	6499	671,8	5,5	9,7
IMBITUVA	10171	8920	939,7	10,8	9,5
INACIO MARTINS	5275	10576	2439,2	2,2	4,3
IRATI	23027	17360	1102,8	20,9	15,7
IVAI	6392	10092	2448,5	2,6	4,1
LAPA	19339	22959	2761,5	7,0	8,3
MANDIRITUBA	11689	13775	547,6	21,3	25,2
PALMEIRA	14887	20684	3907,1	3,8	5,3
PIEN	5303	7017	445,6	11,9	15,7
PONTA GROSSA	150951	62412	4427,2	34,1	14,1
PORTO AMAZONAS	2375	4357	949,8	2,5	4,6
PRUDENTOPOLIS	18982	22997	3594,1	5,3	6,4
QUITANDINHA	6761	9527	491,1	13,8	19,4
REBOUCAS	5224	5707	868,8	6,0	6,6
RIO AZUL	5309	8110	1004,2	5,3	8,1
RIO NEGRO	1610	4127	524,4	3,1	7,9
SAO JOAO DO TRIUNFO	6241	9714	1244,5	5,0	7,8
SAO MATEUS DO SUL	21583	26005	3001,8	7,2	8,7
TEIXEIRA SOARES	4854	7004	1079,7	4,5	6,5
TOTAIS	**466.237**	**352.867**	**36.421,6**	**12,8**	**9,7**

A Tabela 8 apresenta uma comparação entre o desempenho do método AIDA para suas duas abordagens (*Top-Down* e *Bottom-Up*). Ao avaliar a quantidade de CPs selecionadas por cada abordagem, é possível observar que a abordagem TD demanda uma quantidade menor de CPs para a conexão que a abordagem BU. A coluna *Ganho* nessa tabela é computada de acordo com a Equação (4.7) que define o ganho relativo da abordagem *Top–Down* (*TD*) em comparação à abordagem *Bottom–Up (BU)* em relação à redução no número de CPs selecionadas.

$$G_r(\%) = \left(1 - \frac{CP_s^{TD}}{CP_s^{BU}}\right) \cdot 100 \tag{4.7}$$

Tabela 8 – Tabela com resultados do método AIDA analítico.

		Abordagem TD (Top-Down)				Abordagem BU (Bottom-Up)				
Cidade	SMs	CPs Usadas	Quant. SMs não conectados	% SMs não conectados	LRP médio	CPs Usadas	Quant. SMs não conectados	% SMs não conectados	LRP médio	Ganho (G_r(%))
AGUDOS DO SUL	5383	44	77	1,430	-82,271	48	77	1,430	-81,033	8,3%
ARAUCARIA	52836	56	0	0,000	-78,798	69	0	0,000	-72,879	18,8%
BALSA NOVA	6106	44	4	0,066	-75,574	48	4	0,066	-71,936	8,3%
CAMPO DO TENENTE	4224	65	4	0,095	-77,643	74	4	0,095	-75,168	12,2%
CARAMBEI	7512	41	0	0,000	-73,021	45	0	0,000	-69,523	8,9%
CONTENDA	10319	43	178	1,725	-74,678	46	174	1,686	-71,239	6,5%
FAZENDA RIO GRANDE	56157	44	0	0,000	-79,140	58	0	0,000	-73,720	24,1%
GUAMIRANGA	3727	67	4	0,107	-78,006	76	4	0,107	-74,518	11,8%
IMBITUVA	10171	61	9	0,088	-70,961	66	9	0,088	-70,365	7,6%
INACIO MARTINS	5275	142	34	0,645	-79,043	160	29	0,550	-77,732	11,3%
IRATI	23027	105	17	0,074	-74,369	114	17	0,074	-71,959	7,9%
IVAI	6392	117	39	0,610	-79,727	128	39	0,610	-78,349	8,6%
LAPA	19339	171	9	0,047	-76,498	194	9	0,047	-75,075	11,9%
MANDIRITUBA	11689	62	11	0,094	-76,413	68	11	0,094	-72,666	8,8%
PALMEIRA	14887	164	17	0,114	-73,315	180	17	0,114	-70,638	8,9%
PIEN	5303	39	19	0,358	-76,120	40	19	0,358	-73,740	2,5%
PONTA GROSSA	150951	249	3	0,002	-73,437	294	3	0,002	18,930	15,3%
PORTO AMAZONAS	2375	47	4	0,168	-71,215	51	4	0,168	-66,079	7,8%
PRUDENTOPOLIS	18982	196	16	0,084	-76,664	223	16	0,084	-74,468	12,1%
QUITANDINHA	6761	55	3	0,044	-79,595	61	3	0,044	-75,255	9,8%
REBOUCAS	5224	53	3	0,057	-72,859	59	3	0,057	-70,017	10,2%
RIO AZUL	5309	65	0	0,000	-77,419	77	0	0,000	-75,915	15,6%
RIO NEGRO	1610	38	7	0,435	-77,391	43	7	0,435	-73,931	11,6%
SAO JOAO DO TRIUNFO	6241	86	13	0,208	-78,686	93	16	0,256	-74,644	7,5%
SAO MATEUS DO SUL	21583	169	0	0,000	-72,857	190	0	0,000	-67,576	11,1%
TEIXEIRA SOARES	4854	62	17	0,350	-76,229	67	17	0,350	-73,514	7,5%
MÉDIAS:	**17932**	**88**	**19**	**0,262**	**-76,228**	**99**	**19**	**0,258**	**-69,731**	**11,2%**

Ao analisar as Tabelas 8 e 9 é possível extrair alguns resultados que ajudam a compreender melhor as características que diferem as duas abordagens de clusterização do método (TD e BU).

A abordagem *Bottom–Up* se destaca por sua capacidade de assegurar um percentual médio de medidores não conectados (0,258%) ligeiramente menor que a apresentada pela

abordagem TD (que foi de 0,262%). A diferença mais perceptível, no entanto, de BU em comparação a TD, é quanto ao valor médio de LRP apresentado pela solução, em que a abordagem BU apresenta LRP médio de -69,731 dBm e a abordagem TD um LRP médio igual a -76,228 dBm.

As quantidades finais de CPs selecionadas pelo método analítico usando a abordagem Top-Down, que foi a que apresentou maiores ganhos, estão indicadas na Tabela 9.

Tabela 9 – Resultados de AIDA Analítico (quantidade de iterações e números de CPs) obtidos com a abordagem *Top-Down*.

	Resultados AIDA (analítico)						
Região (Cidade)	Medidores	Iterações	CPs Não Selecionadas	CPs Selecionadas	Total CPs	CPs selecionadas/Total CPs	SMs/CPs*
AGUDOS DO SUL	5.383	2	35	44	79	0,557	122,3
ARAUCARIA	52.836	3	182	56	238	0,235	943,5
BALSA NOVA	6.106	2	67	44	111	0,396	138,8
CAMPO DO TENENTE	4.224	2	92	65	157	0,414	65,0
CARAMBEI	7.512	2	68	41	109	0,376	183,2
CONTENDA	10.319	1	3	43	46	0,935	240,0
FAZENDA RIO GRANDE	56.157	3	102	44	146	0,301	1276,3
GUAMIRANGA	3.727	2	65	67	132	0,508	55,6
IMBITUVA	10.171	2	69	61	130	0,469	166,7
INACIO MARTINS	5.275	2	160	142	302	0,470	37,1
IRATI	23.027	2	87	105	192	0,547	219,3
IVAI	6.392	2	118	117	235	0,498	54,6
LAPA	19.339	2	219	171	390	0,438	113,1
MANDIRITUBA	11.689	2	60	62	122	0,508	188,5
PALMEIRA	14.887	2	260	164	424	0,387	90,8
PIEN	5.303	2	43	39	82	0,476	136,0
PONTA GROSSA	150.951	3	996	249	1245	0,200	606,2
PORTO AMAZONAS	2.375	2	77	47	124	0,379	50,5
PRUDENTOPOLIS	18.982	2	199	196	395	0,496	96,8
QUITANDINHA	6.761	2	49	55	104	0,529	122,9
REBOUCAS	5.224	2	58	53	111	0,477	98,6
RIO AZUL	5.309	2	78	65	143	0,455	81,7
RIO NEGRO	1.610	2	48	38	86	0,442	42,4
SAO JOAO DO TRIUNFO	6.241	2	107	86	193	0,446	72,6
SAO MATEUS DO SUL	21.583	2	229	169	398	0,425	127,7
TEIXEIRA SOARES	4.854	2	105	62	167	0,371	78,3
TOTAIS:	**466.237**	−−	**3.576**	**2.285**	**5.861**	**0,390**	**204,0**

* SMs/CPs = Total de Medidores divididos pelo total de CPs Selecionadas

4.4 CONSIDERAÇÕES FINAIS

Neste capítulo, o método AIDA foi apresentado, bem como os resultados com experimentos realizados com dados de larga escala. Pode-se observar que o método é capaz de assegurar conectividade dentro dos parâmetros estabelecidos, atendendo a todas as restrições modeladas para o problema.

No próximo capítulo, um novo método é apresentado para o posicionamento de roteadores/*gateways*. Para esse outro método, uma abordagem inovadora, baseada em técnicas de *machine learning*, é utilizada com o objetivo de apresentar resultados competitivos com os obtidos pelo método analítico, com o uso uma estratégia de execução mais simples e eficiente.

5 MÉTODO AIDA-ML

No capítulo anterior, foi apresentado o método analítico AIDA que faz o posicionamento de roteadores e *gateways* levando em consideração o resultado da análise ponto a ponto da conectividade entre as posições de medidores inteligentes de uma região e as posições de postes selecionados como posições candidatas.

Na abordagem analítica, a análise é detalhada e custosa do ponto de vista computacional, visto que avalia várias possibilidades de combinações, além de fazer a análise detalhada do perfil de terreno para todas essas combinações, até chegar à solução final.

Neste capítulo, é apresentado o método AIDA-ML que, utilizando a experiência adquirida com o método analítico e com o uso de estratégias de *machine learning*, procura simplificar o processo de seleção de posições candidatas para a instalação de roteadores e *gateways*, reduzindo o volume de análises de perfis de terreno e diminuindo expressivamente o número de combinações de conectividade entre medidores e posições candidatas a serem avaliadas.

O método AIDA-ML está descrito no artigo "*Developing an Intelligent Decision Support System for large-scale smart grid communication network planning*" (MOCHINSKI et al., 2024), publicado no jornal *Knowledge-Based Systems*, da Editora Elsevier, acessível pelo link <https://doi.org/10.1016/j.knosys.2023.111159>. No artigo, o método é apresentado como uma abordagem preliminar para o desenvolvimento de um *Intelligent Decision Support System* (IDSS, ou Sistema Inteligente de Suporte à Decisão) e foca na importância do processo de Engenharia de Características do cenário na definição de uma estratégia efetiva para o posicionamento de equipamentos de comunicação em *redes wireless* em *smart grids*.

Conforme descrito anteriormente nesta pesquisa, o posicionamento de roteadores/*gateways* é um problema complexo, dado que os cenários típicos de aplicação envolvem grandes quantidades de medidores inteligentes, postes e várias posições candidatas para selecionar os pontos ideais para a instalação de equipamentos.

Ao analisar a estrutura funcional do método analítico AIDA, observa-se a oportunidade de explorar a criação de uma alternativa técnica que, com o uso de uma estrutura mais simples, consiga alcançar resultados competitivos e, também, ganhos em aspectos como menor tempo de processamento e menor quantidade de cálculos necessários para chegar a uma solução. A ideia de criar um método baseado em aprendizagem de máquina, que aprende com base no modo operacional e nos resultados do método analítico, surge não apenas visando alcançar bons resultados, mas, especialmente, possibilitando avaliar a aplicabilidade de uma estratégia inovadora de posicionamento, pouco explorada pela

literatura.

Com essa ideia em mente, a implementação de um método baseado em *machine learning* parte da análise do modo operacional do método heurístico, passa pela compreensão de sua dinâmica iterativa (que visa estabelecer conexões entre medidores e posições candidatas em cada execução do método) e na observação dos resultados gerados. Como resultado dessa análise, foi concluído que um processo de engenharia de características poderia ser empregado para representar (através da implementação de *datasets* de características) a funcionalidade do método AIDA e servir de base para treinar um modelo de aprendizagem de máquina de classificação binária, caracterizando assim o método AIDA-ML.

Na seção a seguir, o processo de engenharia de características é descrito com o objetivo de explicar como o funcionamento do método AIDA pode ser representado em *features* para uso de um método de *machine learning*. Na sequência, o método AIDA-ML é apresentado em detalhes.

5.1 ENGENHARIA DE CARACTERÍSTICAS

O processo de engenharia de características (*feature engineering*) é parte essencial do método AIDA-ML, uma vez que as características (*features*) serão elementos básicos para os processos de treinamento e classificação.

A proposta do método consiste em simplificar a abordagem utilizada pelo método analítico AIDA, de forma a dispensar a análise detalhada de perfis de terreno para todos os medidores e posições candidatas e tornar o processo de posicionamento mais simples, mais rápido, de forma a alcançar resultados dentro de limites comparáveis entre os métodos. A questão principal em relação a isso, então, consiste em identificar como fazer para que a versão ML do método consiga sugerir posições adequadas para a instalação de roteadores e *gateways* sem necessitar efetuar uma análise detalhada tal como a executada pelo método analítico. Para isso as características utilizadas como base para o processo de classificação devem refletir propriedades e comportamentos da região em análise, porém sem a exigência de uma análise ponto a ponto. A Figura 1, apresentada na Seção 1.1, dá uma ideia geral sobre as características que podem ser evidenciadas num cenário de *smart grid*, com medidores instalados em terrenos com perfis variados e em regiões com alta densidade de medidores e regiões esparsas.

5.1.1 Considerações sobre posições candidatas e sua importância no processo de definição de características

O estudo do conjunto de posições candidatas e da conexão de medidores inteligentes após diferentes iterações do método analítico AIDA é importante para estabelecer as informações a serem extraídas do cenário e que podem compor o conjunto de características de tais posições.

Considerando o cenário da Figura 25, é possível observar que o conjunto de medidores disponíveis para conexão é alterado a cada iteração do método analítico AIDA: os medidores conectados a posições candidatas em cada iteração devem ser desconsiderados em iterações futuras por não estarem mais disponíveis para conexão. O mesmo ocorre em relação às posições de postes usados em iterações anteriores, que devem ser ignoradas em iterações seguintes.

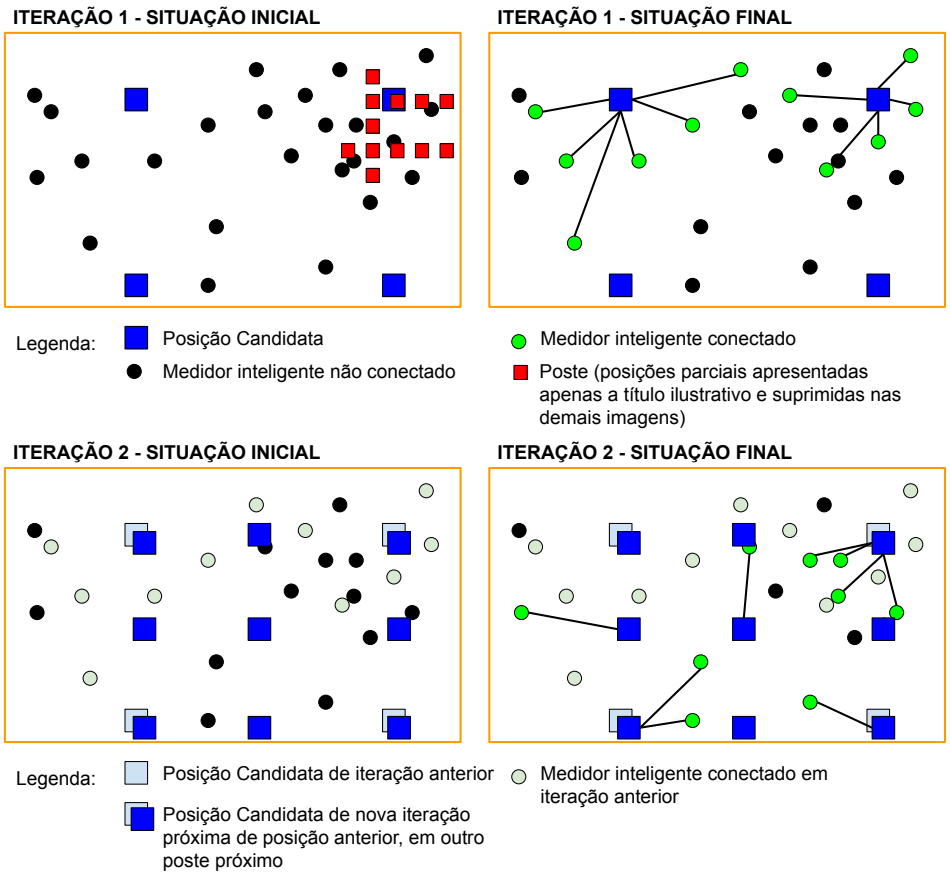


Figura 25 – Cenário de conectividade de medidores a posições candidatas após diferentes iterações do método AIDA.

Fonte: Autoria própria.

Na sequência apresentada na Figura 26, são representados (de forma simplificada) os passos do processo iterativo utilizado pelo método analítico AIDA para a conexão entre

medidores e posições candidatas. Os gráficos levam em consideração os resultados obtidos com a abordagem *Top-Down*.

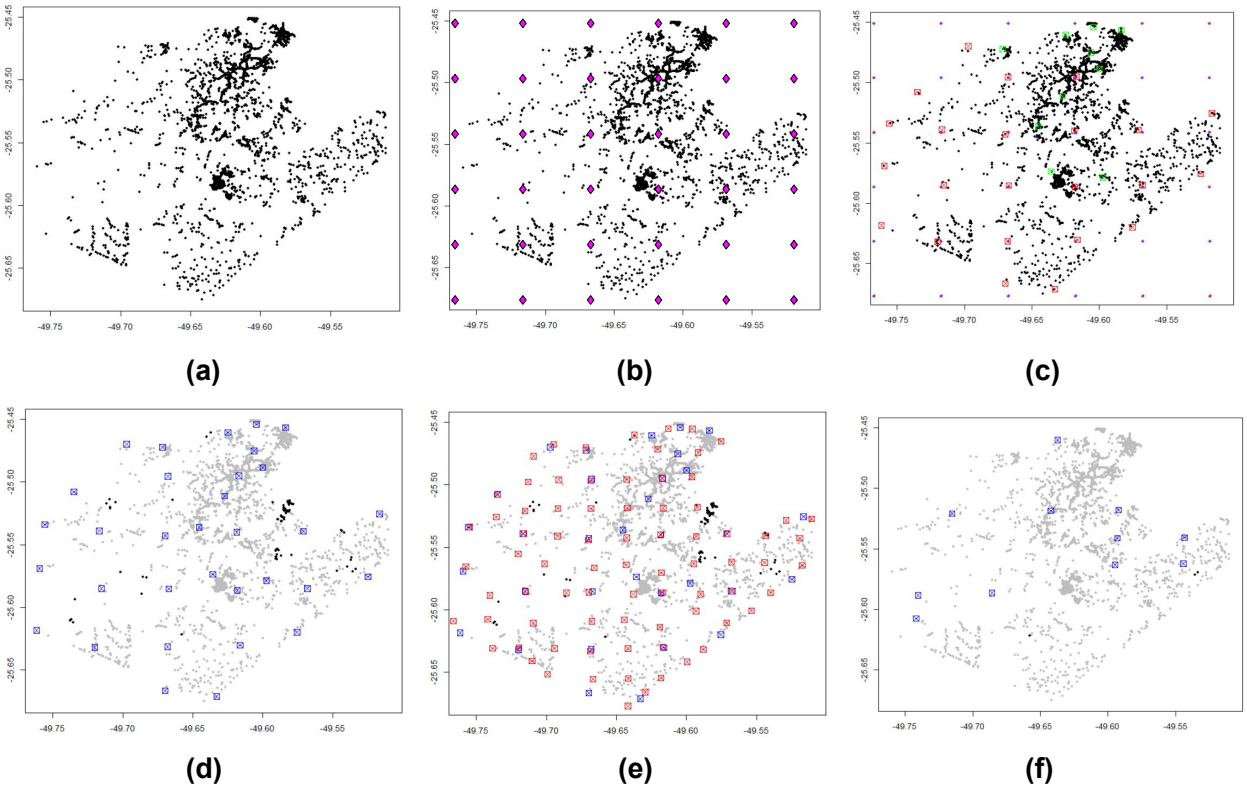


Figura 26 – Exemplo do processo iterativo realizado por AIDA.

Fonte: Autoria própria.

Na Figura 26.a, todos os medidores inteligentes estão desconectados e indicados em preto. Nas Figuras 26.b e 26.c, o *grid* virtual é calculado e plotado (pontos roxos). Na sequência, as posições candidatas são calculadas, ajustadas para posições de postes próximos e selecionadas para evitar posições muito próximas. Em vermelho estão indicadas as posições candidatas referentes a postes e, em verde, as posições das coordenadas de DAs.

A Figura 26.d apresenta as posições candidatas da iteração 1 que foram utilizadas (indicadas em azul e que, no caso, correspondem a todas as posições que estavam disponíveis), os medidores conectados (em cinza) e os medidores remanescentes (a conectar), indicados em preto.

A Figura 26.e apresenta as posições candidatas calculadas e posicionadas para a iteração 2 (indicadas em vermelho). Para comparação, a figura mantém indicadas (em azul) as posições usadas na iteração 1.

A Figura 26.f apresenta o resultado após a iteração 2, indicando as 11 posições candidatas utilizadas pelo método Top-Down nessa iteração (posições em azul) e os 4

medidores que ainda restaram sem conexão (indicados em preto). Em cinza estão indicados os medidores conectados. É possível observar com as etapas indicadas na Figura 26 que, à medida que as iterações vão sendo processadas, o *grid* fica mais *denso* e as posições dos medidores sem conexão mais dispersas.

Para o processo de criação de *features* para uma nova região que venha a ser processada com o método de *machine learning* para a determinação de posição de CPs válidas, existe uma preocupação sobre como apresentar para a classificação pelos métodos ML uma quantidade de posições que evite a geração de muitos falsos positivos que, por suas características iniciais poderiam sugerir que seriam posições candidatas válidas.

No processo iterativo, a quantidade de medidores disponíveis para análise diminui a cada iteração, dado que foram conectados em iterações anteriores do método. Isso afeta a densidade de medidores e posições candidatas disponíveis para análise a cada iteração pelo método analítico AIDA, sendo necessário reproduzir essa característica de alguma forma no processo de criação de *features* a ser implementado.

Os cenários da Figura 25 e da Figura 26 sugerem questões para reflexão e determinação das *features* a serem criadas para representar as características de cada posição candidata:

Q.1) Em relação a uma posição candidata, é sabida a quantidade de medidores e postes no seu entorno, e podem ser extraídas informações relativas ao relevo e qualidade de sinal na região. Pelo processamento das cidades usadas nos experimentos com o método analítico, é possível identificar se determinada posição foi utilizada ou não e a quantidade de medidores que, ao fim do processamento, foram associados a cada posição candidata. A informação de medidores conectados a uma posição candidata é relevante para saber quanto do total de medidores uma posição candidata é capaz de absorver. Porém, essa é uma informação que não pode ser registrada como uma *feature*, pois não é uma informação conhecida antes do processamento. Como fazer, então, para qualificar uma posição candidata de forma a indicar seu potencial de conectividade (potencial de medidores a conectar)? Medidores no entorno estão acima da capacidade de conexão da posição candidata? Acredita-se que *features* e *flags* desse tipo podem ser úteis.

Q.2) O conjunto de medidores disponíveis no entorno de uma posição candidata de uma iteração deve considerar todos os medidores no seu raio de alcance que não estejam conectados, mesmo que esse total ultrapasse o limite de conexões da posição candidata? Em regiões com alta densidade e elevado número de medidores, é muito provável que nem todos os medidores estarão conectados ao final da iteração, demandando nova etapa de processamento. No entanto, computar o número total de medidores no entorno e classificar a posição candidata como posição válida para o posicionamento de roteadores e *gateways*, certamente deverá aumentar a complexidade de calibragem do modelo de aprendizagem, visto que várias posições candidatas podem concorrer pelo mesmo conjunto de medidores

no seu entorno.

Q.3) O conjunto original de postes de uma cidade é muito maior que o número de posições candidatas estabelecidas por um *grid*. A análise de todos os postes como posições além de acarretar elevado tempo de processamento, pode indicar posições candidatas como posições efetivas para a instalação de roteadores em número que acarretará elevado custo de implantação e subutilização de equipamentos. Com isso, a opção pelo uso de uma abordagem baseada em *grid* pode ajudar a diminuir o espaço de busca de soluções e diminuir a possibilidade de indicação excessiva de posições a serem usadas para a instalação de roteadores e *gateways*. Diferentemente do método analítico, cujo número de iterações do método é definido em tempo de execução (que é executado até que se atinja o percentual de medidores conectados especificado), a criação de *datasets* para classificação deve estabelecer uma estratégia artificial de iterações para determinar a lista de posições candidatas a serem avaliadas.

O processo de engenharia de *features* deve privilegiar características que tenham significado para os resultados do modelo. Sabendo disso, é possível inferir que os seguintes conjuntos de características impactam no modelo de ML: i) Características sobre o relevo; ii) Características sobre o processo iterativo do método analítico; iii) Características com informações sobre a densidade de medidores no entorno de posições candidatas. Além disso, considera-se que as *features* a serem usadas por AIDA-ML devem refletir as restrições e heurísticas que o modelo analítico utiliza.

5.1.2 Tipos de características

Para a caracterização de uma região, diferentes *features* são estabelecidas de forma a qualificar as posições candidatas à instalação de roteadores e *gateways*. A Figura 27 serve como referência para facilitar o entendimento das descrições dos tipos de características. Nessa figura, a posição central "R zero" (R0) corresponde à posição candidata de interesse, para a qual se tem interesse de criar características para a sua descrição. No entorno de R0 é possível identificar a existência de medidores inteligentes, postes e regiões de outras posições candidatas (R1 a R8). Essas regiões no entorno podem ser entendidas como as regiões que estão mais próximas da posição candidata (CP) central que está em análise. As características propostas visam representar as propriedades da região possibilitando analisar aspectos relativos a relevo, qualidade de sinal e densidade de equipamentos, sem depender de uma análise minuciosa das possibilidades de interconexão entre todos os medidores próximos e uma posição candidata como a utilizada pelo método analítico AIDA.

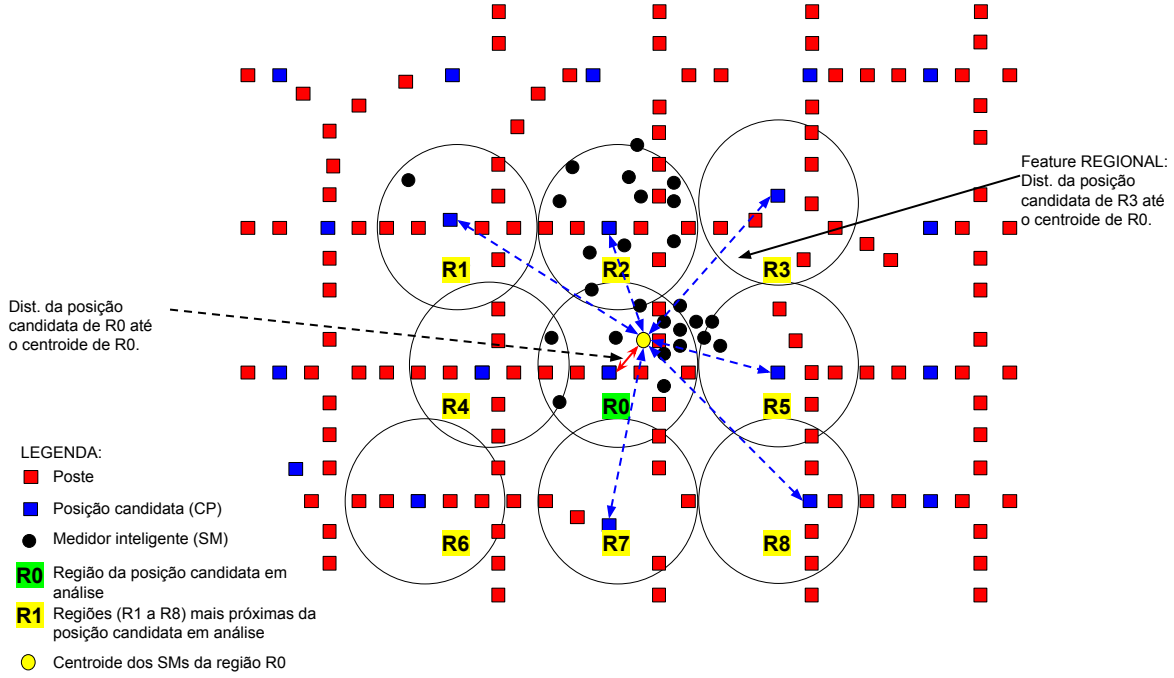


Figura 27 – Regiões (R1 a R8) localizadas no entorno de uma posição candidata central indicada como R0 ("R zero") .

Fonte: Autoria própria.

Features de Identificação:

As *features* de identificação representam informações chave para o cadastro e identificação de posições candidatas. São de caráter único e capazes de distinguir uma posição de outra.

Nos processos de treinamento e classificação, essas *features* são desconsideradas, por servirem apenas como atributos de identificação.

Features Locais:

As *features* locais têm o objetivo de representar informações relacionadas à área no alcance de comunicação de uma posição candidata em análise, conforme mostrado na Figura 28. A região $R0$ ao redor da CP é considerada para calcular as *features locais*. Nessa figura, os ícones azuis indicam postes selecionados como posições candidatas. No centro da região está a CP_{R0}, ou seja, a posição candidata em análise no processo de engenharia de características. Na figura, para os ícones que representam casas, considera-se que todas tenham um medidor inteligente instalado com capacidade de comunicação sem fio.

As *features* locais abrangem várias características, incluindo o número de medidores inteligentes e postes dentro da faixa da posição candidata (CP), informações do terreno ao redor da CP (variações de elevação), qualidade do sinal para diferentes posições ao redor da CP para caracterizar a qualidade esperada da comunicação na região e a distância

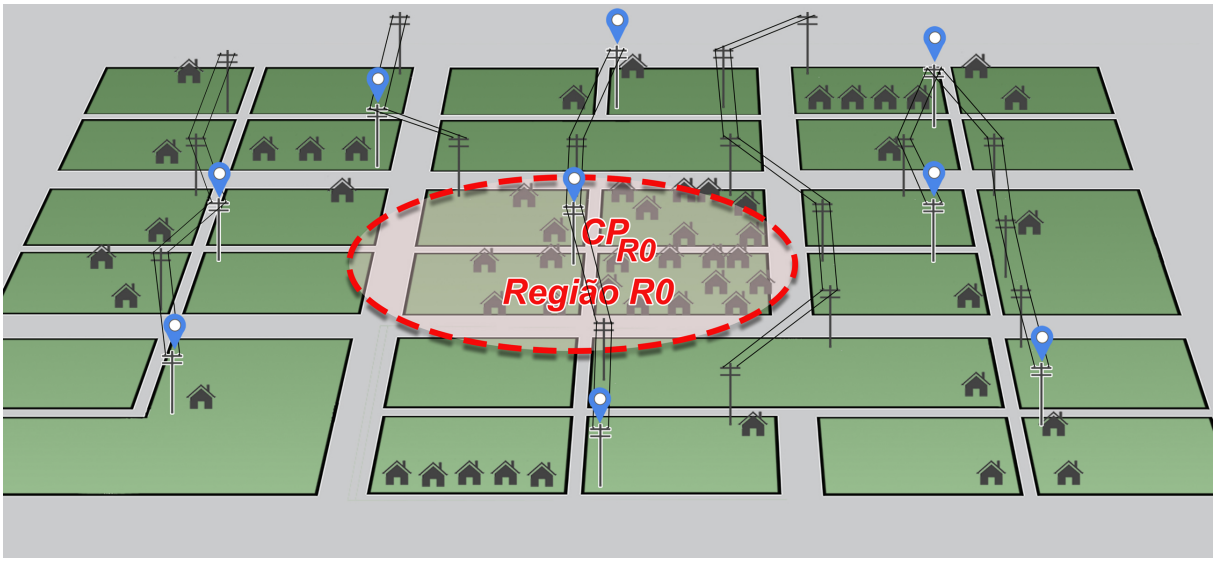


Figura 28 – *Features* Locais – Delimitação de área de abrangência.

Fonte: Adaptado de (MOCHINSKI et al., 2024).

entre a CP e a concentração principal de medidores inteligentes, entre outros.

Ainda em relação a essa categoria de *features*, pode-se dizer que:

- O número de medidores inteligentes ao redor da CP é importante para determinar se a posição candidata sozinha é suficiente para conectar todos os medidores na região ou se são necessárias CPs adicionais.
- Para caracterizar a topografia ao redor da CP, valores de elevação são registrados em diferentes direções (Norte, Nordeste, Leste, Sudeste, Sul, Sudoeste, Oeste e Noroeste) e em várias distâncias (100 m, 200 m, 300 m, 400 m, 500 m, 1000 m, 1500 m, 2000 m, 2500 m, 3000 m).
- A qualidade do sinal, representada pelo valor de LRP entre um medidor inteligente e a posição da CP, é calculada usando o mesmo método que o utilizado por AIDA (analítico). Esse cálculo considera a potência de transmissão dos medidores inteligentes, valores de ganho das antenas e a perda de percurso do canal.
- Informações sobre os postes presentes na região também são capturadas. O número total de postes na região é calculado, e a posição do centroide desses elementos é usada para determinar se a CP em análise está próxima ou distante de uma região com alta concentração de postes.

Features Regionais:

As *features* regionais representam características que procuram capturar propriedades de posições candidatas no entorno da posição em análise e que, por esse motivo,

podem concorrer/competir na conexão dos mesmos medidores que existem na região. As regiões de interesse se referem às 8 posições (R1, R2,... R8) localizadas no entorno de uma região central indicada como R0 (Figura 29). Toda posição candidata terá 8 regiões no seu entorno. Elas correspondem às 8 posições candidatas mais próximas da posição candidata original, visto que a escolha de posições dispostas em posições cardeais reais (N, NE, E, SE, S, SW, W e NW) seria inviável dado que os postes mais próximos nem sempre estão dispostos ortogonalmente. É importante observar que a ortogonalidade não ocorre apenas por questões de irregularidade das quadras, mas também porque, após o estabelecimento do *grid*, cada ponto do *grid* pode ser deslocado para as coordenadas de um poste verdadeiro e mais próximo da região em análise. De uma forma geral, as principais informações relacionadas a essa categoria de *features* incluem:

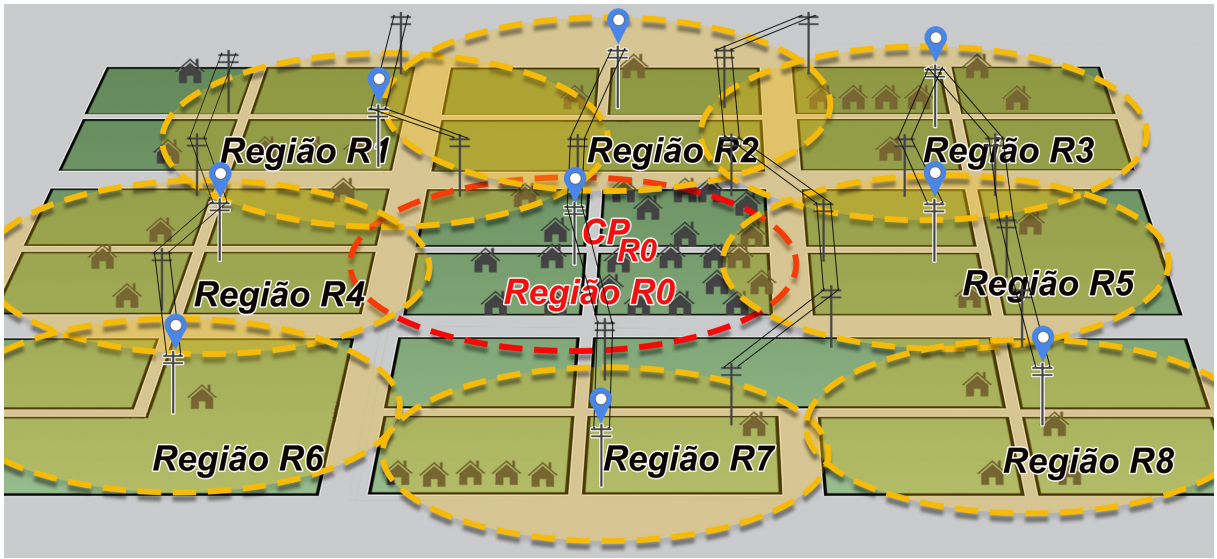


Figura 29 – *Features* Regionais – Exemplo ilustrativo do posicionamento das regiões no entorno da região R0.

Fonte: Adaptado de (MOCHINSKI et al., 2024).

- Características que indicam o número de medidores inteligentes em cada região ao redor da CP em análise.
- Características relacionadas à qualidade do sinal entre a CP em análise e a CP em cada uma das regiões $R1$ a $R8$.
- Características que representam a distância entre a CP em análise e o centroide das regiões no entorno.

Features de Iteração Anterior:

As *features* de iteração anterior representam características que identificam relações entre uma determinada iteração e a iteração que a antecedeu. São necessárias para

incorporar ao processo de treinamento informações que permitam identificar a necessidade de seleção de múltiplas posições candidatas próximas para assegurar a conexão de medidores que existam na região.

Numa região com alta densidade de medidores, por exemplo, em número superior à capacidade de conexões de um roteador, é provável que mais de uma posição candidata seja necessária para estabelecer a cobertura de todos os medidores existentes. Ou seja, posições candidatas selecionadas numa iteração anterior podem ser insuficientes para a conexão dos medidores existentes, demandando a necessidade de seleção de posições candidatas adicionais em novas iterações.

Numa região esparsa, por sua vez, é possível que uma posição candidata selecionada numa iteração anterior tenha conseguido conectar todos os medidores no seu entorno, diminuindo a necessidade de alocar uma nova posição candidata numa iteração posterior na mesma região.

Features Globais:

As *features* globais buscam caracterizar a relevância da posição candidata em análise em relação ao cenário geral. As principais informações relacionadas a esse tipo de *feature* incluem:

- A criação de uma característica que representa a porcentagem de medidores inteligentes dentro da faixa da CP em relação ao total de medidores da cidade em análise, e que tem como objetivo indicar a importância da CP no processo de cobertura geral de comunicação.
- Informações sobre a área total da região, largura e comprimento do terreno, o número total de medidores e postes, o número de medidores por quilômetro quadrado (medidores/km^2), e o número de postes por quilômetro quadrado (postes/km^2).
- Além disso, essa categoria inclui características que representam a posição do centroide dos medidores inteligentes na região (indicando a concentração de medidores inteligentes), informações de elevação do terreno em vários pontos e detalhes sobre a qualidade do sinal (possibilitando a identificação de regiões com diferentes condições de estabelecimento de conexão).

Feature Classe:

Corresponde ao atributo que rotula a posição candidata em análise, classificando-a como uma posição válida (selecionada) ou não para o posicionamento de um roteador ou *gateway*. Corresponde ao atributo classe ou *target* do registro servindo, portanto, para caracterizar o problema como um problema de classificação binária.

Esse atributo é preenchido nos *datasets* de treinamento, para ser avaliado pelo processo de aprendizagem, porém deixado em branco nos *datasets* que passarão por processo de classificação, visto que esse processo é responsável pela determinação de seu valor no método AIDA-ML.

5.1.3 Relação de características

Com base nos tipos de *features* estabelecidas para o processo de *machine learning*, esta seção apresenta o conjunto de *features* a serem criadas na estruturação de *datasets* utilizados pelo método AIDA-ML.

O conjunto de características inclui um total de **333 *features*** descritas no Apêndice A. Para o treinamento, são consideradas 318 *features* mais o atributo *target* (*CLASSE_CP*). As *features* ignoradas são as reservadas para a identificação do registro ou informações complementares. Na estrutura, todos os atributos são numéricos, com exceção de *ID_NomeCidade*.

5.2 ESTRUTURA DO MÉTODO AIDA-ML

O método AIDA-ML implementa uma abordagem de *machine learning* para posicionamento de roteadores e *gateways*, e inclui os módulos de construção do modelo de aprendizagem, de preparação de dados de uma nova região, e de classificação. A Figura 30 apresenta os módulos componentes de AIDA-ML e sua relação com o método AIDA. A descrição detalhada dos módulos de AIDA-ML é apresentada nesta seção.

5.2.1 Construção do Modelo de Aprendizagem

O módulo de Construção do Modelo de Aprendizagem é utilizado para construir um modelo de *machine learning* (Modelo ML) a partir do conhecimento gerado pelo método analítico AIDA, suas características e comportamento. Este módulo inclui duas etapas principais: Engenharia de Características e Treinamento.

a) Engenharia de Características:

Esse módulo gera um conjunto de dados de treinamento com base nos resultados produzidos pelo método analítico AIDA.

O *dataset* de treinamento é composto por uma lista de posições candidatas (CP) utilizadas por AIDA para o planejamento AMI de diversas cidades. Para cada posição candidata, as características são criadas conforme relação apresentada na Seção 5.1.3. O atributo *target* (classe) é definido com base nos valores produzidos pelo método AIDA

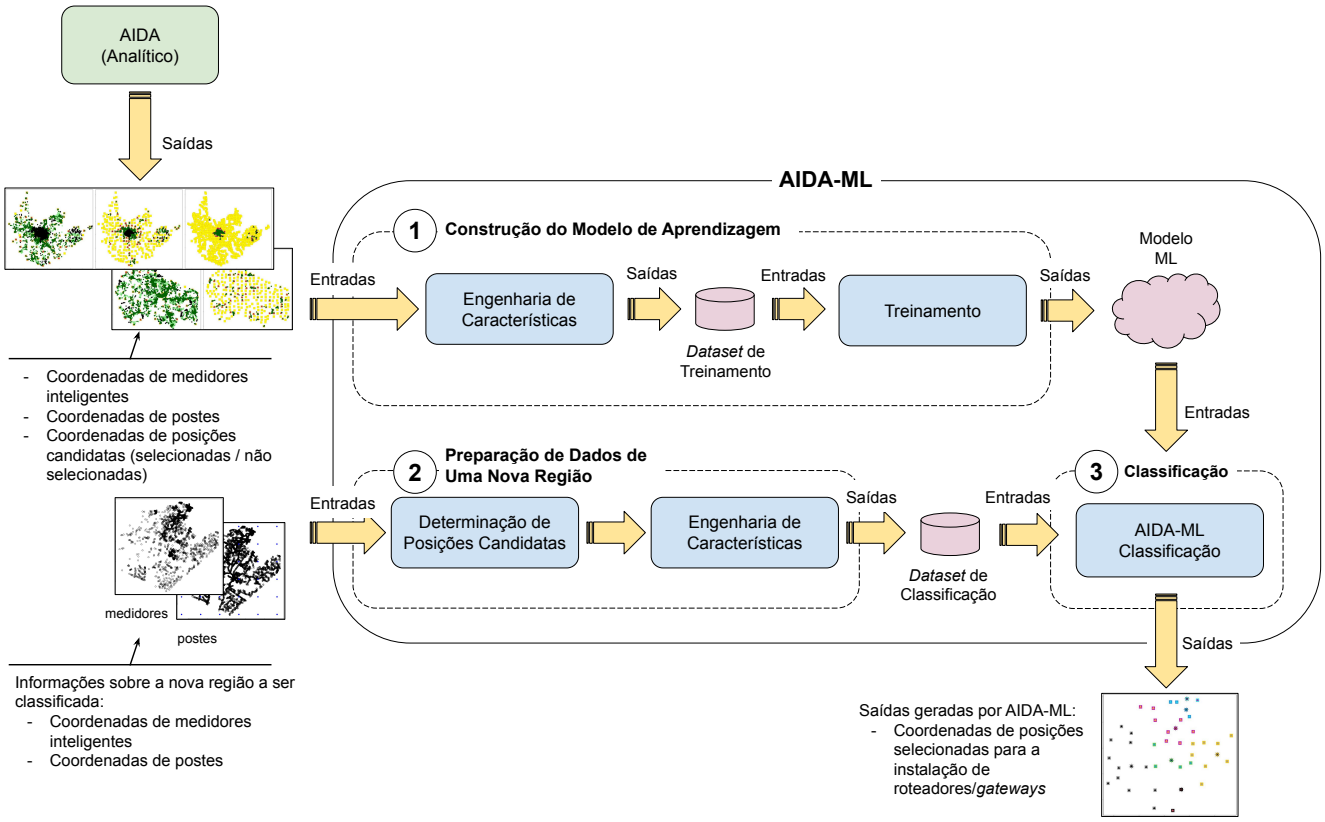


Figura 30 – Estrutura do método AIDA-ML.

Fonte: Adaptado de (MOCHINSKI et al., 2024).

e classifica uma posição candidata como válida (1) ou inválida (0) para instalação de roteadores/*gateways*.

O processo incorpora dados de diversas regiões para criar um conjunto de dados de treinamento abrangente, capaz de capturar diversos cenários e servir como uma base de conhecimento robusta. Essa abordagem garante que os processos de aprendizagem realizados pela etapa de Treinamento possam avaliar com eficácia regiões com características geográficas distintas e concentrações variadas de medidores inteligentes. Ao incluir dados de diferentes regiões, o AIDA-ML pode adquirir uma compreensão mais ampla e melhorar a sua capacidade de classificar e avaliar uma vasta gama de cenários geográficos.

Os parâmetros de entrada (Entradas) da etapa consistem nas coordenadas dos medidores inteligentes e dos postes considerados em cada iteração do método AIDA. Além disso, o módulo leva em consideração as coordenadas das posições candidatas de todas as iterações realizadas por AIDA e a lista de posições selecionadas pelo método analítico para o posicionamento de roteadores e *gateways* (*target*).

O processo de Engenharia de Características gera um conjunto de dados contendo os recursos criados (*Dataset* de Treinamento), que são posteriormente utilizados para o treinamento e criação do modelo de ML.

b) Treinamento:

Essa etapa usa o conjunto de dados de treinamento (*Dataset* de Treinamento) criado pela etapa anterior como entrada para criar um modelo de *machine learning* (Modelo ML). O algoritmo de aprendizagem selecionado (no caso, o XGBoost) analisa os recursos do conjunto de dados e suas relações para determinar os critérios de classificação que criarão um modelo capaz de selecionar ou rejeitar uma posição candidata para o posicionamento de roteador e *gateway*.

5.2.2 Preparação de Dados de Uma Nova Região

O módulo de Preparação de Dados de Uma Nova Região foi projetado para gerar conjuntos de dados contendo as características de posições candidatas para a classificação de uma nova região. O objetivo principal desse módulo é produzir uma lista de posições candidatas (e suas características) que passarão pelo processo de Classificação. Cada elemento da lista é avaliado para determinar sua elegibilidade como posição para a instalação final de roteadores e *gateways* de uma nova região.

As características das posições candidatas geradas por esse módulo seguem a estrutura especificada na Seção 5.1.3. O atributo *target* (*classe*) nesse conjunto de dados é deixado em branco, pois será estabelecido posteriormente pelo processo de classificação.

Os parâmetros de entrada do módulo incluem informações específicas da região, como as coordenadas de medidores inteligentes e as coordenadas de postes da região. Como saída, o módulo gera um conjunto de dados (*Dataset* de Classificação) que incorpora as características criadas a partir dos dados da região a ser classificada.

Este módulo inclui duas etapas principais: Determinação de Posições Candidatas e Engenharia de Características.

a) Determinação de Posições Candidatas:

A determinação dos postes que serão considerados como posições candidatas para a instalação de roteadores/*gateways* na nova região a ser analisada, utiliza o mesmo processo de cálculo de posições candidatas usado pelo método AIDA (descrito na Seção 4.2.3), utilizando um *grid* para a escolha de um subconjunto de postes da região.

Diferentemente do método analítico, que estabelece posições candidatas para cada iteração que venha a realizar, o método AIDA-ML, por sua vez, cria posições candidatas simulando a ocorrência de 3 (três) iterações, visto que deve submeter à classificação os dados de todas as posições candidatas de uma única vez. Essa quantidade de iterações foi estabelecida após experimentos com o método, e visa dar cobertura tanto a cidades menores

(que podem ser resolvidas pelo método analítico com 1 iteração) quanto a cidades com grande número de medidores (normalmente resolvidas em até 3 iterações do método AIDA).

b) Engenharia de Características:

Esse módulo gera um conjunto de dados de classificação para as posições candidatas estabelecidas para a nova região.

A criação de características é feita da mesma forma que a utilizada para o processo de Engenharia de Características descrito no módulo de Construção de Modelo de Aprendizagem (Seção 5.2.1), com a diferença de que, ao invés de considerar posições estabelecidas por AIDA, considera as posições candidatas identificadas pela etapa anterior deste módulo (Identificação de Posições Candidatas). Além disso, ao invés de gerar dados rotulados, deixa o atributo Classe em branco para ser definido pelo processo de classificação.

5.2.3 Classificação

O módulo de *Classificação* usa o modelo de aprendizagem (Modelo ML) criado pelo módulo de *Construção do Modelo de Aprendizagem* para classificar o *Dataset* de Classificação com posições candidatas geradas pelo módulo de *Preparação de Dados de Uma Nova Região* de AIDA-ML.

A classificação das posições candidatas é realizada por meio de uma tarefa de *machine learning* de classificação binária que classifica uma CP como selecionada (1 ou viável para instalação de roteadores/*gateways*) ou não selecionada (0 ou inviável).

Como resultado, o módulo de *Classificação* do AIDA-ML gera um conjunto de dados contendo as coordenadas de posição selecionadas para instalação de roteadores/*gateways* na região analisada.

5.3 CONSIDERAÇÕES FINAIS

Neste capítulo foi apresentado o método AIDA-ML, que implementa uma estratégia de *machine learning* para posicionamento de roteadores e *gateways*.

O método AIDA-ML é composto por módulos com funções que incluem: i) A construção de *dataset* de treinamento e de um modelo de *machine learning* que incorpore características de cenários avaliados originalmente pelo método AIDA (analítico); ii) A preparação de dataset de classificação, com dados de uma região cujos dados não tenham sido avaliados no processo de treinamento; iii) A classificação de dados de posições candidatas de uma nova região, que é responsável pela identificação das melhores posições para a instalação de roteadores/*gateways* para a região em análise.

Para a criação de *datasets* de treinamento e de teste, um processo de engenharia de características é utilizado de forma a gerar atributos capazes de qualificar as posições candidatas de interesse para uma região. Entre as características, o processo de geração das *features* deve ser capaz de computar informações quanto à identificação, aos aspectos locais (topografia, densidade de medidores, qualidade de sinal) e às propriedades de regiões do entorno da posição candidata em análise. Levando-se em consideração que o método AIDA analítico é um processo iterativo, as *features* criadas para AIDA-ML devem, também, ser capazes de caracterizar tal processo, visto que a concorrência por recursos entre uma iteração e outra do método analítico é dinâmica, em razão do cenário de medidores e postes disponíveis para a conexão se alterar completamente a cada iteração.

Com um conjunto de *features* capazes de sintetizar todos os aspectos funcionais e resultados do método analítico, uma base de conhecimento consistente pode ser formada para o processo de treinamento e classificação do método baseado em *machine learning*.

Sob o ponto de vista de implementação, o método AIDA-ML é um método mais simples que o método analítico, tendo sua maior complexidade no processo de engenharia de características.

Em relação aos resultados do processo de classificação, o método AIDA-ML busca ser uma alternativa ao método AIDA analítico, capaz de efetuar o posicionamento de roteadores/*gateways* em localizações que assegurem a conectividade de medidores inteligentes dentro dos parâmetros estabelecidos, com a vantagem de apresentar ganhos no processamento.

No capítulo a seguir, são descritos os experimentos e resultados obtidos com o método AIDA-ML e um comparativo do seu desempenho em relação ao método AIDA analítico.

6 EXPERIMENTOS COM O MÉTODO AIDA-ML E RESULTADOS

No capítulo anterior foi apresentado o método AIDA-ML, que utiliza uma abordagem baseada em *machine learning* para posicionamento de roteadores e *gateways* e que busca ser uma alternativa otimizada ao uso do método analítico. Neste capítulo, os experimentos visam verificar o desempenho do método AIDA-ML e comparar os seus resultados com os valores gerados pela versão analítica do método.

6.1 PROTOCOLO EXPERIMENTAL

O protocolo experimental utilizado para validação do método AIDA-ML engloba as seguintes tarefas:

- Descrição de dados de entrada: identificação de fontes e tipos de dados, análise exploratória e proporção entre as classes.
- Protocolo de avaliação: definição da estratégia de execução dos experimentos (em especial os processos de treinamento, validação e testes) no que se refere à forma como os dados serão organizados/particionados.
- Métricas de avaliação: estabelecimento das métricas que permitem a comparação de desempenho dos algoritmos nos diferentes cenários avaliados.
- Seleção de *features*: utilização de diferentes abordagens para avaliar o desempenho de modelos de aprendizagem com diferentes conjuntos de *features*.
- Otimização de hiperparâmetros: uso de técnicas de otimização de hiperparâmetros para buscar melhor desempenho dos modelos de *machine learning*.

6.1.1 Descrição de dados de entrada

Os dados de entrada utilizados nos processos de geração de *features*, treinamento e testes de classificação incluem informações obtidas de 26 municípios do estado do Paraná indicados na Tabela 7 (Seção 4.3). Os resultados obtidos com o processamento dessas cidades com o método AIDA analítico, em especial em relação à lista de CPs selecionados, estão indicados na Tabela 9 (Seção 4.3).

Tal como para o método AIDA analítico, os dados de interesse desses municípios incluem as coordenadas geográficas de medidores inteligentes e de postes, bem como as posições dos dispositivos de automação da rede (DA *devices*).

Na Tabela 9, ao analisar as colunas referentes ao número de CPs, é possível observar o desbalanceamento na distribuição de valores do atributo *Classe* que indica se determinada posição candidata foi ou não selecionada como coordenada para a instalação de roteadores/*gateways*. Do total de 5.861 posições candidatas criadas pelo método AIDA para as cidades do estudo, 38,99% foram selecionadas e 61,01% não foram. É importante destacar que não foi identificado nenhum comportamento padronizado quanto ao número de CPs selecionadas ser sempre menor ou maior que o número de CPs não selecionadas analisando cada cidade individualmente.

Quanto ao número de iterações realizadas por AIDA, o número médio calculado foi igual a 2,08. Do total de cidades, uma demandou 1 iteração (no caso, Contenda), 3 necessitaram de 3 iterações (Araucária, Fazenda Rio Grande e Ponta Grossa) e o restante (22 cidades) tiveram a solução obtida com 2 iterações cada.

O número médio de medidores por CP selecionada (coluna SMs/CPs da Tabela 9) expressa o volume de medidores conectados (em média) a cada roteador/*gateway* da região. Nessa coluna, os valores variam de 37,1 (para a cidade de Inácio Martins) a 1.276,3 (para a cidade de Fazenda Rio Grande) indicando que a concentração de medidores é variável entre os municípios, dependente de sua área territorial e do tipo da região (urbana, urbana densa, ou rural/esparsa). Em média, considerando todas as cidades utilizadas no experimento, foram obtidos 204 SMs por CP selecionada.

6.1.2 Protocolo de avaliação

Para o treinamento dos algoritmos, foi utilizada uma abordagem denominada de *Leave-One-Out Cross-Validation* (SAMMUT; WEBB, 2010), mais especificamente um protocolo denominado de LOSO-CV *Leave-One-Subject-Out* (FAZLI et al., 2009; GHOLAMIANGONABADI; KISELOV; GROLINGER, 2020; PAULI; POHL; GOLZ, 2021), ou simplesmente LOSO, implementado da seguinte forma: considerando a existência de 26 cidades para os experimentos, o processo de treinamento (*fitting*) utiliza os dados de 25 cidades reservando os dados de uma cidade para efetuar os testes (Figura 31). O processo é repetido para cada uma das 26 cidades, gerando um modelo de aprendizagem para cada processo de treinamento. As métricas de desempenho (por exemplo, acurácia (POWERS, 2008) e AUC-PR (DAVIS; GOADRICH, 2006; KEILWAGEN; GROSSE; GRAU, 2014)) são calculadas para cada cidade individualmente. Adicionalmente, um valor médio do desempenho obtido pela classificação de dados de todas as cidades é calculado com o objetivo de obter um resultado geral. De acordo com o autor em (BROWNLEE, 2020a), a vantagem de utilizar essa abordagem é que, com a criação e avaliação de diferentes

modelos, obtém-se uma estimativa mais robusta do desempenho.

O uso da abordagem *Leave-One-Subject-Out* (LOSO) tal como proposta para este estudo busca simular uma situação real de avaliação de topologia para uma cidade que não tenha participado do processo de treinamento, tornando o processo de testes mais próximo ao de um cenário real de aplicação do método.

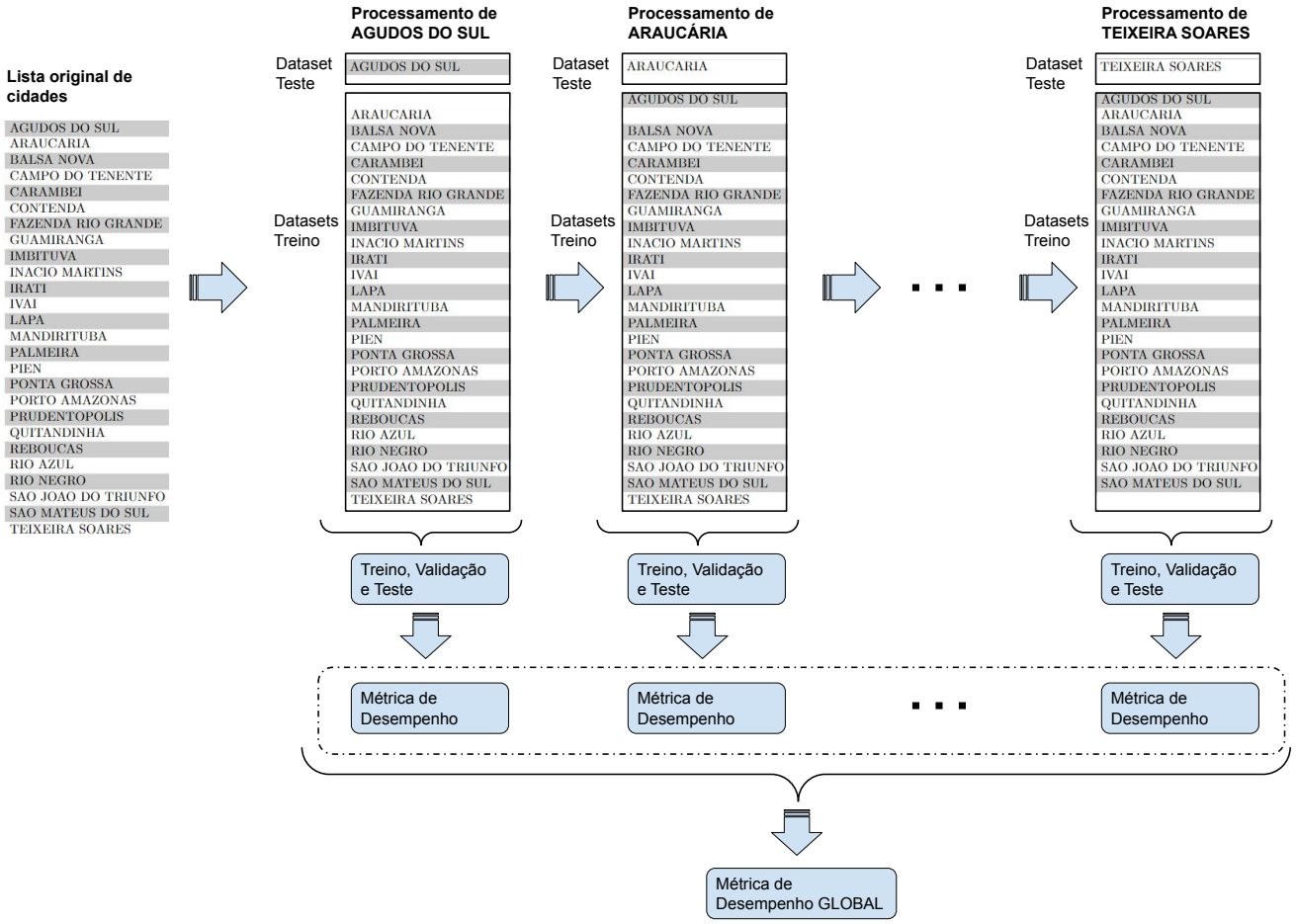


Figura 31 – Protocolo *Leave-One-Subject-Out* (LOSO).

Fonte: Autoria própria.

Ainda em relação ao treinamento, uma vez agrupados os dados de 25 cidades (deixando os dados de uma cidade reservados para teste), 90% dos registros são usados para treinamento e 10% para validação do processo de treino.

O desempenho dos algoritmos selecionados para os experimentos iniciais (Regressão Logística, *Random Forest* e *XGBoost*) é verificado em testes preliminares considerando o uso de sua configuração *default* de hiperparâmetros, conforme estabelecidos pela biblioteca de funções de *machine learning* para o Python, o *scikit-learn* (PEDREGOSA et al., 2011).

6.1.3 Métricas de avaliação

Os autores em (DAVIS; GOADRICH, 2006) destacam que a análise do desempenho de algoritmos de aprendizagem não deve estar limitada à análise de acurácia apenas. A escolha pela melhor métrica deve levar em conta a distribuição da classe, verificando o quanto a base de dados está ou não balanceada (diferença na contagem de exemplos com classe rotulada como positiva ou negativa).

Em Davis e Goadrich (2006), seus autores avaliam e comparam o uso de curvas ROC (*receiver operating characteristic*) e curvas PR (*precision-recall*) de forma a auxiliar na escolha da melhor métrica de análise de desempenho. Para tarefas com grande distorção (bastante desequilíbrio) na distribuição de classes, o uso de curvas *Precision-Recall* se mostra mais adequado. Brownlee (2020b) aborda sobre a aplicação de curvas ROC e curvas PR em modelos de classificação binária, em especial para dados desbalanceados. A curva ROC (DAVIS; GOADRICH, 2006) plota uma relação entre a taxa de falsos positivos (no eixo horizontal) versus a taxa de verdadeiros positivos (no eixo vertical), e a curva PR faz uma relação entre *Precision* (precisão) e *Recall* (revocação ou sensibilidade). O autor complementa que a análise de curvas AUC (*area under curve*), no caso AUC-ROC e AUC-PR, sumarizam as métricas e podem ser usadas para a comparação de classificadores. Para este estudo, as principais métricas selecionadas para a comparação de desempenho dos algoritmos de aprendizagem incluem a análise de acurácia e o uso da curva AUC-PR.

A acurácia pode ser expressa pela Equação (6.1):

$$acurácia = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.1}$$

onde TP corresponde a classificações *True Positive* (Positivo Verdadeiro, registro classificado como verdadeiro sendo realmente verdadeiro), TN significa *True Negative* (Verdadeiro Negativo, ou seja, resultados negativos classificados como negativos), FP corresponde a resultados *False Positive* (Falso Positivo, ou seja, resultados classificados como positivos, mas que na realidade são negativos), e FN equivale a resultados *False Negative* (Falso Negativo, ou seja, resultados classificados como negativos, mas que deveriam ser classificados como positivos).

Para a curva AUC-PR, os autores em (GOUTTE; GAUSSIER, 2005) definem, conforme as equações (6.2) e (6.3), os conceitos de *Precision* e *Recall*. É importante explicar que *Precision* procura identificar do total de registros classificados como positivos, quantos são realmente positivos; e *Recall* avalia de um total de exemplos positivos, quantos são classificados como positivos; ou seja, quantos exemplos positivos foram corretamente classificados pelo modelo (LEKHTMAN, 2019).

$$precision = \frac{TP}{TP + FP} \tag{6.2}$$

$$recall = \frac{TP}{TP + FN} \tag{6.3}$$

Do ponto de vista do cenário de aplicação, em especial para comparar o desempenho dos métodos AIDA e AIDA-ML, outros indicadores são analisados, entre eles: total de CPs selecionadas, quantidade de medidores não conectados, percentual de medidores não conectados e valor médio de LRP.

6.1.4 Seleção de *features*

Para este estudo optou-se pelo uso de duas abordagens distintas de seleção de *features*, a saber:

- Seleção com o uso de filtro: Corresponde ao processo de seleção de característica utilizado nos experimentos principais deste estudo (Seções 6.3, 6.4 e 6.5). Para essa filtragem, optou-se pela análise do atributo *feature_importances_* disponibilizado pelo *sklearn*, que estabelece a importância de cada *feature* para o modelo de treinamento gerado. Ao invés de uma filtragem simples pela importância da característica para o modelo de ML, a estratégia criada para este estudo utiliza uma avaliação de relevância das características para um conjunto mais amplo de modelos (um para cada cidade utilizada no treinamento). Essa estratégia é apresentada com mais detalhes na Seção 6.3.1.

- Seleção automática de *features* com o uso de abordagens *wrapper*. A seleção sequencial de *features*, que é uma técnica de *wrapper*, é utilizada nos experimentos adicionais deste estudo (Seção 6.7), e envolve o uso de um modelo de aprendizado de máquina (no caso, o XGBoost) para avaliar a qualidade das diferentes combinações de *features*. Em outras palavras, ela incorpora o modelo de *machine learning* como uma parte integral do processo de seleção de *features*. Para a seleção sequencial de *features*, duas formas foram selecionadas para uso neste estudo: i) *Sequential Backward Selection* (SBS), que efetua o treinamento inicial do modelo de aprendizagem com todas as *features* do *dataset* e, gradualmente, vai eliminando as *features* (uma de cada vez), avaliando para isso o resultado obtido no processo de classificação; ii) *Sequential Forward Selection* (SFS), que inicia o treinamento do modelo de aprendizagem com uma *feature* de cada vez e, gradualmente, vai acrescentando outras *features* até se chegar à quantidade de *features* de interesse. O objetivo de ambas as abordagens é identificar a quantidade e combinação de *features* com as quais se obtém melhores resultados no processo de classificação. Para os experimentos com essas abordagens,

foram usadas funções da biblioteca *MLxtend* (RASCHKA, 2018), disponível para o Python.

6.1.5 Otimização de hiperparâmetros

Neste estudo, além da seleção manual de hiperparâmetros (utilizada para os experimentos principais, descritos nas Seções 6.3, 6.4 e 6.5), duas abordagens distintas de otimização de hiperparâmetros são utilizadas, incluindo o *grid search*, que faz uma busca sistemática em um espaço pré-definido de valores para os hiperparâmetros do modelo, e técnicas de AutoML (*Automated Machine Learning*). Entre os diferentes métodos, bibliotecas e abordagens de AutoML disponíveis, foram selecionados os seguintes métodos para uso nos experimentos adicionais (Seção 6.7) deste estudo:

- Biblioteca *Auto-sklearn* (FEURER et al., 2015; FEURER et al., 2020).
- Biblioteca *TPOT* (Tree-based Pipeline Optimization Tool) (OLSON et al., 2016).
- Biblioteca *Scikit-Optimize* (skopt), disponível em <https://scikit-optimize.github.io/>, que implementa métodos de otimização baseados em modelos sequenciais:
  - *skopt.dummy_minimize* — Busca aleatória por amostragem uniforme dentro de limites estabelecidos.
  - *skopt.gbrt_minimize* — Otimização sequencial usando *gradient-boosted trees*.
  - *skopt.gp_minimize* - Otimização bayesiana usando processos gaussianos.

6.2 EXPERIMENTOS E RESULTADOS

Esta seção descreve os experimentos realizados com o método AIDA-ML. Inicialmente, apresenta os resultados obtidos a partir de *datasets* de treinamento utilizando a abordagem *Leave-One-Subject-Out*. Em seguida, avalia o desempenho do método com experimentos realizados com *datasets* gerados pela etapa de preparação de dados de uma nova região, realizados para avaliar o número de iterações a serem simuladas de acordo com características da cidade a classificar.

Após a execução de todas as etapas do método AIDA-ML, um processo especial foi criado para validar e comparar os resultados do método de *machine learning* e avaliar a aderência de AIDA-ML em relação ao método analítico. Nesse processo, as posições de CPs selecionadas pelo método de classificação de AIDA-ML são utilizadas como posições candidatas (dados de entrada) no método AIDA analítico que executa uma iteração apenas e obtém a cobertura de conexão dos medidores da região em análise.

Quanto às linguagens de programação utilizadas no desenvolvimento de AIDA-ML, o processo de engenharia de características, que cria os *datasets* de treinamento e de teste, foi desenvolvido na Linguagem R, versão 4.2.1. Os processos de treinamento/criação de modelo ML e de classificação de AIDA-ML, por sua vez, foram implementados na linguagem Python, versão 3.8.10.

6.2.1 Experimentos iniciais com a abordagem *Leave-One-Subject-Out*

Experimentos iniciais foram realizados utilizando a abordagem *Leave-one-subject-out* com os algoritmos Regressão Logística (biblioteca *sklearn*, método *LogisticRegression*), *Random Forest* (biblioteca *sklearn*, método *RandomForestClassifier*) e XGBoost (biblioteca *xgboost*, método *XGBClassifier*). Esses algoritmos foram selecionados por possuírem características distintas para a criação do modelo de aprendizagem. O algoritmo de Regressão Logística é um método de fácil entendimento, e os demais fazem uso de abordagens distintas para a combinação de modelos simples, geralmente, árvores de decisão, facilitando a compreensão do resultado obtido e da relevância dos atributos considerados. Além disso, os algoritmos Random Forest e XGBoost usam estratégias para a redução da dimensionalidade do espaço de características consideradas para a construção de cada árvore.

Os resultados obtidos estão expressos nas Tabelas 10, 11 e 12. Para os experimentos, 10% dos dados foram usados para validação do processo de treinamento e os testes realizados considerando os dados da cidade em análise (desconsiderada no processo de treinamento e validação).

Observação: os experimentos iniciais foram realizados com um conjunto de *features* que ainda não contemplava todas as características apresentadas no Apêndice A, mas tiveram grande importância para a seleção do algoritmo de *machine learning* a ser usado para a construção do modelo de treinamento de AIDA-ML. Esses testes iniciais foram realizados com *datasets* de testes que ainda não utilizavam o processo Engenharia de Características da etapa de "Preparação de Dados de Uma Nova Região" de AIDA-ML, e consideravam para o processo de classificação as posições candidatas estabelecidas pelo método analítico AIDA. A natureza iterativa do método de pesquisa utilizado nos experimentos possibilitou, em fase posterior, identificar a necessidade de criação de grupos de *features* adicionais, que passaram a incluir características sobre a Iteração Anterior e *features* Globais. Tais grupos de *features* se fizeram necessários para possibilitar a validação do número ideal de iterações a serem consideradas pelo processo de Engenharia de Características e a obtenção de percentual de cobertura de comunicação de medidores inteligentes em valores competitivos com os obtidos com o método analítico. Antes da inclusão dessas *features*, era possível perceber uma tendência a obter muitos resultados falsos positivos, classificando como posições válidas para a instalação de roteadores/*gateways*

um número muito superior ao computado pelo método AIDA para as mesmas regiões dos experimentos.

Os resultados obtidos com os experimentos iniciais (classificação de base de testes) com esses três algoritmos foram os seguintes:

- Regressão Logística: Acurácia = 90,054%, AUC-PR = 0,912 (Hiperparâmetros: *penalty='l2', dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None*).
- Random Forest: Acurácia = 90,388%, AUC-PR=0,915 (Hiperparâmetros: *n_estimators=100, criterion='gini', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='sqrt', max_leaf_nodes=None, min_impurity_decrease=0.0, bootstrap=True, oob_score=False, n_jobs=None, random_state=None, verbose=0, warm_start=False, class_weight=None, ccp_alpha=0.0, max_samples=None*).
- XGBoost: Acurácia = 90,695%, AUC-PR=0,918 (Hiperparâmetros: *tree_method='hist', learning_rate=0.3, max_depth=3, reg_lambda = 1, n_estimators=100*).

De uma forma geral, os resultados com os algoritmos foram muito próximos, com valores levemente superiores para o método XGBoost. Por causa disso e pelo desempenho demonstrado em diferentes competições do site Kaggle[1], o algoritmo XGBoost (CHEN; GUESTRIN, 2016) foi selecionado para o processo de classificação de AIDA-ML. Após experimentos com variação manual de hiperparâmetros, os valores selecionados para o processo de classificação com XGBoost foram os seguintes: *tree_method = 'hist', learning_rate = 1, max_depth = 15, reg_lambda = 20, n_estimators= 500.*

6.2.2 Análise de iterações de AIDA-ML

Os resultados apresentados nas Tabelas 10, 11 e 12 consideram como dados de treinamento e testes os dados gerados pelo processamento das saídas computadas pelo método AIDA analítico. Os dados nessas estruturas consideram todas as iterações que o processo analítico necessitou para conectar os medidores disponíveis em cada cidade (respeitando o critério de parada estabelecido pelo método). A quantidade de medidores observados nas cidades disponíveis variou de 1.610 a 150.951 (Tabela 7). O número de iterações exigidos pelo AIDA analítico para alcançar a conectividade dos medidores dentro dos limites estabelecidos variou de 1 (uma) a 3 (três) iterações. Para a cidade com menos medidores (Rio Negro, com 1.610 medidores) foram necessárias 2 iterações do método

[1] https://www.kaggle.com/

Tabela 10 – Métricas de desempenho (LR com hiperparâmetros *default*)

Cidade	Teste total	Teste 0	Teste 1	TP	FP	TN	FN	TPR	FPR	TNR	FNR	LogisticR accu teste	LogisticR aucPR teste
AGUDOS DO SUL	79	35	44	38	6	29	6	86,364	17,143	82,857	13,636	84,81	0,902
ARAUCARIA	238	182	56	50	7	175	6	89,286	3,846	96,154	10,714	94,538	0,898
BALSA NOVA	111	67	44	40	2	65	4	90,909	2,985	97,015	9,091	94,595	0,949
CAMPO DO TENENTE	157	92	65	57	5	87	8	87,692	5,435	94,565	12,308	91,72	0,924
CARAMBEI	109	68	41	35	8	60	6	85,366	11,765	88,235	14,634	87,156	0,861
CONTENDA	46	3	43	43	3	0	0	100	100	0	0	93,478	0,967
FAZENDA RIO GRANDE	146	102	44	36	5	97	8	81,818	4,902	95,098	18,182	91,096	0,876
GUAMIRANGA	132	65	67	55	9	56	12	82,09	13,846	86,154	17,91	84,091	0,886
IMBITUVA	130	69	61	53	6	63	8	86,885	8,696	91,304	13,115	89,231	0,914
INACIO MARTINS	302	160	142	126	16	144	16	88,732	10	90	11,268	89,404	0,914
IRATI	192	87	105	82	8	79	23	78,095	9,195	90,805	21,905	83,854	0,906
IVAI	235	118	117	102	21	97	15	87,179	17,797	82,203	12,821	84,681	0,882
LAPA	390	219	171	150	19	200	21	87,719	8,676	91,324	12,281	89,744	0,909
MANDIRITUBA	122	60	62	58	6	54	4	93,548	10	90	6,452	91,803	0,937
PALMEIRA	424	260	164	150	17	243	14	91,463	6,538	93,462	8,537	92,689	0,923
PIEN	82	43	39	35	4	39	4	89,744	9,302	90,698	10,256	90,244	0,922
PONTA GROSSA	1245	996	249	212	32	964	37	85,141	3,213	96,787	14,859	94,458	0,875
PORTO AMAZONAS	124	77	47	42	4	73	5	89,362	5,195	94,805	10,638	92,742	0,923
PRUDENTOPOLIS	395	199	196	176	20	179	20	89,796	10,05	89,95	10,204	89,873	0,923
QUITANDINHA	104	49	55	50	3	46	5	90,909	6,122	93,878	9,091	92,308	0,95
REBOUCAS	111	58	53	47	9	49	6	88,679	15,517	84,483	11,321	86,486	0,89
RIO AZUL	143	78	65	55	10	68	10	84,615	12,821	87,179	15,385	86,014	0,881
RIO NEGRO	86	48	38	34	4	44	4	89,474	8,333	91,667	10,526	90,698	0,918
SAO JOAO DO TRIUNFO	193	107	86	76	5	102	10	88,372	4,673	95,327	11,628	92,228	0,937
SAO MATEUS DO SUL	398	229	169	149	22	207	20	88,166	9,607	90,393	11,834	89,447	0,902
TEIXEIRA SOARES	167	105	62	57	5	100	5	91,935	4,762	95,238	8,065	94,012	0,934
MÍNIMO								**78,095**	**2,985**	**0**	**0**	**83,854**	**0,861**
MÁXIMO								**100**	**100**	**97,015**	**21,905**	**94,595**	**0,967**
MÉDIA								**88,205**	**12,324**	**87,676**	**11,795**	**90,054**	**0,912**

analítico. A cidade de Contenda, com 10.319 medidores, foi resolvida com 1 iteração e Ponta Grossa, com 150.951 medidores, demandou 3 iterações, assim como outras cidades com grande número de medidores como Fazenda Rio Grande (com 56.157) e Araucária (com 52.836 medidores) que também foram solucionadas com 3 iterações, sugerindo que esse seja um número que atende a cidades com esta característica.

A geração de dados de testes para uma nova região, em especial o processo de engenharia de características, tem como parâmetros de entrada conhecidos as coordenadas de medidores e as coordenadas de postes da cidade a classificar. Para esse caso, o número de iterações que o método analítico precisaria executar é conhecido apenas após a execução do método. Para os experimentos realizados com AIDA-ML, então, ficou estabelecido que o total de iterações a serem utilizadas para a geração de *features* para bases de dados de classificação (*dataset* de classificação) deve ser igual a 3 (três), aplicável a todos os municípios, independentemente da quantidade de medidores que possua, deixando para o processo de classificação, indicar quais devem ser definidas como CPs selecionadas.

Tabela 11 – Métricas de desempenho (RF com hiperparâmetros *default*)

Cidade	Teste total	Teste 0	Teste 1	TP	FP	TN	FN	TPR	FPR	TNR	FNR	RF accu teste	RF aucPR teste
AGUDOS DO SUL	79	35	44	41	4	31	3	93,182	11,429	88,571	6,818	91,139	0,94
ARAUCARIA	238	182	56	49	3	179	7	87,5	1,648	98,352	12,5	95,798	0,923
BALSA NOVA	111	67	44	40	2	65	4	90,909	2,985	97,015	9,091	94,595	0,949
CAMPO DO TENENTE	157	92	65	59	5	87	6	90,769	5,435	94,565	9,231	92,994	0,934
CARAMBEI	109	68	41	33	8	60	8	80,488	11,765	88,235	19,512	85,321	0,842
CONTENDA	46	3	43	43	3	0	0	100	100	0	0	93,478	0,967
FAZENDA RIO GRANDE	146	102	44	39	6	96	5	88,636	5,882	94,118	11,364	92,466	0,894
GUAMIRANGA	132	65	67	60	10	55	7	89,552	15,385	84,615	10,448	87,121	0,903
IMBITUVA	130	69	61	52	5	64	9	85,246	7,246	92,754	14,754	89,231	0,917
INACIO MARTINS	302	160	142	129	18	142	13	90,845	11,25	88,75	9,155	89,735	0,915
IRATI	192	87	105	78	7	80	27	74,286	8,046	91,954	25,714	82,292	0,901
IVAI	235	118	117	100	15	103	17	85,47	12,712	87,288	14,53	86,383	0,898
LAPA	390	219	171	151	15	204	20	88,304	6,849	93,151	11,696	91,026	0,922
MANDIRITUBA	122	60	62	57	5	55	5	91,935	8,333	91,667	8,065	91,803	0,94
PALMEIRA	424	260	164	150	17	243	14	91,463	6,538	93,462	8,537	92,689	0,923
PIEN	82	43	39	36	5	38	3	92,308	11,628	88,372	7,692	90,244	0,919
PONTA GROSSA	1245	996	249	225	40	956	24	90,361	4,016	95,984	9,639	94,859	0,886
PORTO AMAZONAS	124	77	47	43	2	75	4	91,489	2,597	97,403	8,511	95,161	0,951
PRUDENTOPOLIS	395	199	196	169	18	181	27	86,224	9,045	90,955	13,776	88,608	0,917
QUITANDINHA	104	49	55	50	3	46	5	90,909	6,122	93,878	9,091	92,308	0,95
REBOUCAS	111	58	53	46	12	46	7	86,792	20,69	79,31	13,208	82,883	0,862
RIO AZUL	143	78	65	55	9	69	10	84,615	11,538	88,462	15,385	86,713	0,888
RIO NEGRO	86	48	38	33	4	44	5	86,842	8,333	91,667	13,158	89,535	0,909
SAO JOAO DO TRIUNFO	193	107	86	76	7	100	10	88,372	6,542	93,458	11,628	91,192	0,926
SAO MATEUS DO SUL	398	229	169	149	21	208	20	88,166	9,17	90,83	11,834	89,698	0,904
TEIXEIRA SOARES	167	105	62	55	5	100	7	88,71	4,762	95,238	11,29	92,814	0,923
MÍNIMO								**74,286**	**1,648**	**0**	**0**	**82,292**	**0,842**
MÁXIMO								**100**	**100**	**98,352**	**25,714**	**95,798**	**0,967**
MÉDIA								**88,591**	**11,921**	**88,079**	**11,409**	**90,388**	**0,915**

6.3 SELEÇÃO DE CARACTERÍSTICAS

O processo de geração de *features* utilizado pelos módulos de AIDA-ML faz, originalmente, a geração de 333 características. Desse total, 318 características podem ser utilizadas para os processos de treinamento e classificação. Apesar dessas características expressarem informações particulares acerca de cada posição candidata, pode ocorrer o fato de nem todas terem caráter relevante para os algoritmos de *machine learning*, em especial por não serem suficientemente discriminantes para a definição do atributo classe.

Por essa questão, nesta seção são apresentados os resultados de experimentos que analisaram e identificaram as *features* mais relevantes para AIDA-ML, de forma a estabelecer uma estratégia para a redução de dimensionalidade das bases de dados utilizadas no experimento.

Tabela 12 – Métricas de desempenho (XGB com hiperparâmetros *default*)

Cidade	Teste total	Teste 0	Teste 1	TP	FP	TN	FN	TPR	FPR	TNR	FNR	XGBoost accu teste	XGBoost aucPR teste
AGUDOS DO SUL	79	35	44	41	5	30	3	93,182	14,286	85,714	6,818	89,873	0,931
ARAUCARIA	238	182	56	51	9	173	5	91,071	4,945	95,055	8,929	94,118	0,891
BALSA NOVA	111	67	44	42	1	66	2	95,455	1,493	98,507	4,545	97,297	0,975
CAMPO DO TENENTE	157	92	65	57	6	86	8	87,692	6,522	93,478	12,308	91,083	0,916
CARAMBEI	109	68	41	35	7	61	6	85,366	10,294	89,706	14,634	88,073	0,871
CONTENDA	46	3	43	43	3	0	0	100	100	0	0	93,478	0,967
FAZENDA RIO GRANDE	146	102	44	38	3	99	6	86,364	2,941	97,059	13,636	93,836	0,916
GUAMIRANGA	132	65	67	60	11	54	7	89,552	16,923	83,077	10,448	86,364	0,897
IMBITUVA	130	69	61	54	5	64	7	88,525	7,246	92,754	11,475	90,769	0,927
INACIO MARTINS	302	160	142	129	18	142	13	90,845	11,25	88,75	9,155	89,735	0,915
IRATI	192	87	105	80	8	79	25	76,19	9,195	90,805	23,81	82,813	0,901
IVAI	235	118	117	100	13	105	17	85,47	11,017	88,983	14,53	87,234	0,906
LAPA	390	219	171	154	22	197	17	90,058	10,046	89,954	9,942	90	0,91
MANDIRITUBA	122	60	62	56	7	53	6	90,323	11,667	88,333	9,677	89,344	0,921
PALMEIRA	424	260	164	154	16	244	10	93,902	6,154	93,846	6,098	93,868	0,934
PIEN	82	43	39	35	5	38	4	89,744	11,628	88,372	10,256	89,024	0,911
PONTA GROSSA	1245	996	249	219	33	963	30	87,952	3,313	96,687	12,048	94,94	0,886
PORTO AMAZONAS	124	77	47	43	3	74	4	91,489	3,896	96,104	8,511	94,355	0,941
PRUDENTOPOLIS	395	199	196	173	19	180	23	88,265	9,548	90,452	11,735	89,367	0,921
QUITANDINHA	104	49	55	51	2	47	4	92,727	4,082	95,918	7,273	94,231	0,964
REBOUCAS	111	58	53	46	7	51	7	86,792	12,069	87,931	13,208	87,387	0,899
RIO AZUL	143	78	65	57	10	68	8	87,692	12,821	87,179	12,308	87,413	0,892
RIO NEGRO	86	48	38	33	4	44	5	86,842	8,333	91,667	13,158	89,535	0,909
SAO JOAO DO TRIUNFO	193	107	86	73	8	99	13	84,884	7,477	92,523	15,116	89,119	0,909
SAO MATEUS DO SUL	398	229	169	154	24	205	15	91,124	10,48	89,52	8,876	90,201	0,907
TEIXEIRA SOARES	167	105	62	58	5	100	4	93,548	4,762	95,238	6,452	94,611	0,94
MÍNIMO								**76,19**	**1,493**	**0**	**0**	**82,813**	**0,871**
MÁXIMO								**100**	**100**	**98,507**	**23,810**	**97,297**	**0,975**
MÉDIA								**89,425**	**12,015**	**87,985**	**10,575**	**90,695**	**0,918**

6.3.1 Estratégia implementada

A estratégia utilizada para a seleção de *features* considera uma abordagem de filtro que avalia o atributo *feature_importances_* disponibilizado pelo *sklearn* para um modelo treinado com um classificador (no caso, o XGBoost).

Etapas da estratégia:

1) Para cada cidade do experimento, foi montada uma base de dados de treinamento considerando todas as cidades da lista, exceto a cidade em análise. A base de dados gerada foi treinada com o classificador e os valores de *feature importance* foram armazenados em um arquivo externo em ordem decrescente de importância.

2) Depois de se obter os dados de importância para cada cidade, todos os *datasets* foram agrupados e os valores individuais de cada atributo somados para formar um *dataset* com os campos *Feature* e *SomaImportanciaFeature*.

3) O *dataset* com a soma de importâncias das *features* foi ordenado por *SomaImportanciaFeature* em ordem decrescente. Uma vez ordenado, foram obtidas as listas com as *n-features* mais relevantes, selecionando as 20, 40, 80 e 120 características mais importantes.

Depois disso, foram preparados *datasets* de treino e de testes com os atributos selecionados e realizados os processos de classificação de AIDA-ML e a validação com AIDA analítico das posições selecionadas.

6.3.2 Características Selecionadas

Nesta seção, estão indicadas as características identificadas como as mais relevantes para o processo de treinamento. Para a análise e seleção das *features*, foram considerados *datasets* originais com 318 características. Os conjuntos com as 20, 40, 80 e 120 *features* mais importantes são indicados a seguir e as descrições de cada *feature* estão disponíveis no Apêndice A.

Conjunto das 20 *features* mais importantes = {I_Dist_CP_CP7, I_Dist_CP_CP8, I_Dist_CP_CP5, ID_Iteracao, I_LRP_CP_CP1, I_Dist_CP_CP6, I_Dist_CP_CP1, I_L_SMs_CP5, I_L_DA_CP1, I_LRP_CP_Centroide_CP7, G_perc_SMs, I_LRP_CP_CP2, I_Dist_CP_CP2, G_BboxArea_SMs, L_DistMin_SMs, R_SMs_R2, L_SMs_CP, R_LRP_R1, R_LRP_R2, L_Postes_CP}

Conjunto das 40 *features* mais importantes = {I_Dist_CP_CP7, I_Dist_CP_CP8, I_Dist_CP_CP5, ID_Iteracao, I_LRP_CP_CP1, I_Dist_CP_CP6, I_Dist_CP_CP1, I_L_SMs_CP5, I_L_DA_CP1, I_LRP_CP_Centroide_CP7, G_perc_SMs, I_LRP_CP_CP2, I_Dist_CP_CP2, G_BboxArea_SMs, L_DistMin_SMs, R_SMs_R2, L_SMs_CP, R_LRP_R1, R_LRP_R2, L_Postes_CP, L_DiffElev_SE_500m, I_Dist_CP_CP4, I_L_SMs_CP1, L_LRP_W_100m, R_SMs_R7, L_DiffElev_SW_200m, L_DistDevPad_SMs, I_Dist_CP_CP3, L_DiffElev_W_200m, L_CentroidePostes_LON, R_SMs_R4, G_LRP_Centroide_NW, L_DiffElev_NW_1000m, L_Centroide_LON, R_SMs_R1, L_Elev_CP, L_LRP_NE_100m, L_DiffElev_E_200m, I_L_SMs_CP4, L_LRP_NW_100m}

Conjunto das 80 *features* mais importantes = {I_Dist_CP_CP7, I_Dist_CP_CP8, I_Dist_CP_CP5, ID_Iteracao, I_LRP_CP_CP1, I_Dist_CP_CP6, I_Dist_CP_CP1, I_L_SMs_CP5, I_L_DA_CP1, I_LRP_CP_Centroide_CP7, G_perc_SMs, I_LRP_CP_CP2, I_Dist_CP_CP2, G_BboxArea_SMs, L_DistMin_SMs, R_SMs_R2, L_SMs_CP, R_LRP_R1, R_LRP_R2, L_Postes_CP, L_DiffElev_SE_500m, I_Dist_CP_CP4, I_L_SMs_CP1, L_LRP_W_100m, R_SMs_R7, L_DiffElev_SW_200m, L_DistDevPad_SMs, I_Dist_CP_CP3, L_DiffElev_W_200m, L_CentroidePostes_LON, R_SMs_R4, G_LRP_Centroide_NW, L_DiffElev_NW_1000m, L_Centroide_LON, R_SMs_R1, L_Elev_CP, L_LRP_NE_100m, L_DiffElev_E_200m, I_L_SMs_CP4, L_LRP_NW_100m, L_DiffElev_SE_300m, R_Dist_R1, I_L_SMs_CP3, R_Dist_R2,

L_DiffElev_NW_200m, I_LRP_CP_CP4, L_DiffElev_N_3000m, L_DiffElev_NW_500m, L_DistPoste_S, L_DiffElev_E_300m, L_DiffElev_N_1000m, L_DiffElev_W_100m, G_DiffElev_E, L_LRP_NE_2000m, L_DiffElev_SW_100m, I_L_DA_CP5, I_L_DA_CP3, L_DiffElev_N_300m, I_LRP_CP_CP5, G_LRP_Centroide_S, R_LRP_R4, L_DiffElev_NW_2500m, L_DiffElev_NW_2000m, G_LRP_Centroide_NE, L_DiffElev_W_1500m, L_DiffElev_S_1000m, R_LRP_R8, I_LRP_CP_CP3, L_LRP_N_100m, L_DiffElev_SW_300m, L_Dist_CP_CentroideG, R_SMs_R5, L_DiffElev_W_500m, L_DiffElev_W_300m, L_LRP_SW_100m, L_DiffElev_CP_CentroideG, L_DiffElev_SE_200m, L_DiffElev_N_500m, L_LRP_Centroide, L_Dist_E}

Conjunto das 120 *features* mais importantes = {I_Dist_CP_CP7, I_Dist_CP_CP8, I_Dist_CP_CP5, ID_Iteracao, I_LRP_CP_CP1, I_Dist_CP_CP6, I_Dist_CP_CP1, I_L_SMs_CP5, I_L_DA_CP1, I_LRP_CP_Centroide_CP7, G_perc_SMs, I_LRP_CP_CP2, I_Dist_CP_CP2, G_BboxArea_SMs, L_DistMin_SMs, R_SMs_R2, L_SMs_CP, R_LRP_R1, R_LRP_R2, L_Postes_CP, L_DiffElev_SE_500m, I_Dist_CP_CP4, I_L_SMs_CP1, L_LRP_W_100m, R_SMs_R7, L_DiffElev_SW_200m, L_DistDevPad_SMs, I_Dist_CP_CP3, L_DiffElev_W_200m, L_CentroidePostes_LON, R_SMs_R4, G_LRP_Centroide_NW, L_DiffElev_NW_1000m, L_Centroide_LON, R_SMs_R1, L_Elev_CP, L_LRP_NE_100m, L_DiffElev_E_200m, I_L_SMs_CP4, L_LRP_NW_100m, L_DiffElev_SE_300m, R_Dist_R1, I_L_SMs_CP3, R_Dist_R2, L_DiffElev_NW_200m, I_LRP_CP_CP4, L_DiffElev_N_3000m, L_DiffElev_NW_500m, L_DistPoste_S, L_DiffElev_E_300m, L_DiffElev_N_1000m, L_DiffElev_W_100m, G_DiffElev_E, L_LRP_NE_2000m, L_DiffElev_SW_100m, I_L_DA_CP5, I_L_DA_CP3, L_DiffElev_N_300m, I_LRP_CP_CP5, G_LRP_Centroide_S, R_LRP_R4, L_DiffElev_NW_2500m, L_DiffElev_NW_2000m, G_LRP_Centroide_NE, L_DiffElev_W_1500m, L_DiffElev_S_1000m, R_LRP_R8, I_LRP_CP_CP3, L_LRP_N_100m, L_DiffElev_SW_300m, L_Dist_CP_CentroideG, R_SMs_R5, L_DiffElev_W_500m, L_DiffElev_W_300m, L_LRP_SW_100m, L_DiffElev_CP_CentroideG, L_DiffElev_SE_200m, L_DiffElev_N_500m, L_LRP_Centroide, L_Dist_E, L_DistMax_SMs, L_DistMed_SMs, L_DiffElev_W_1000m, L_DiffElev_S_3000m, L_Dist_S, L_Centroide_LAT, L_DiffElev_SW_500m, G_Total_Postes, L_LRP_W_200m, L_DistPoste_NW, I_Dist_CP_Centroide_CP6, I_Dist_CP_Centroide_CP1, R_Dist_R8, G_DiffElev_S, L_DiffElev_E_3000m, L_LRP_N_2000m, L_LRP_W_1000m, L_DiffElev_N_2000m, I_L_SMs_CP8, G_Postes_km2, L_Dist_SE, L_DiffElev_SE_1000m, I_L_DA_CP8, I_Dist_CP_Centroide_CP3, L_DistPoste_SE, L_DiffElev_NE_1500m, G_DiffElev_W, R_LRP_R5, L_DiffElev_S_500m, L_LRP_NW_3000m, L_DiffElev_NE_2000m,

G_Dist_Centroide_W, L_DiffElev_SE_3000m, L_DiffElev_SE_400m, I_LRP_CP_Centroide_CP2, L_LRP_S_400m, R_Dist_R4, I_Dist_CP_Centroide_CP2, L_DiffElev_SW_400m, L_DiffElev_NW_3000m}

6.4 ANÁLISE DE DESEMPENHO COM DIFERENTES CONJUNTOS DE *FEATURES*

Nesta seção são apresentados os resultados de experimentos com diferentes conjuntos de *features* (20, 40, 80, 120 e todas as 318 *features*) com o objetivo de selecionar a combinação que apresentar melhor desempenho. Para essa análise, *datasets* com diferentes quantidades de *features* mais relevantes foram utilizados pelo processo de classificação de AIDA-ML.

A análise de desempenho do processo de classificação de AIDA-ML considerou: (i) O número de CPs selecionadas pelo processo de classificação em relação ao total de CPs selecionadas pelo processo AIDA analítico; (ii) A cobertura de conexão (% de medidores conectados) obtida com as CPs selecionadas pela Classificação de AIDA-ML. Observação: essa análise de cobertura (e de qualidade média de sinal) foi feita submetendo as CPs selecionadas ao método AIDA analítico, de forma a verificar (em uma única iteração) se as posições propostas por AIDA-ML garantem boa cobertura. Esse processo foi denominado de Validação-ML (Figura 32) e busca avaliar a aderência dos resultados de AIDA-ML em relação ao método analítico.

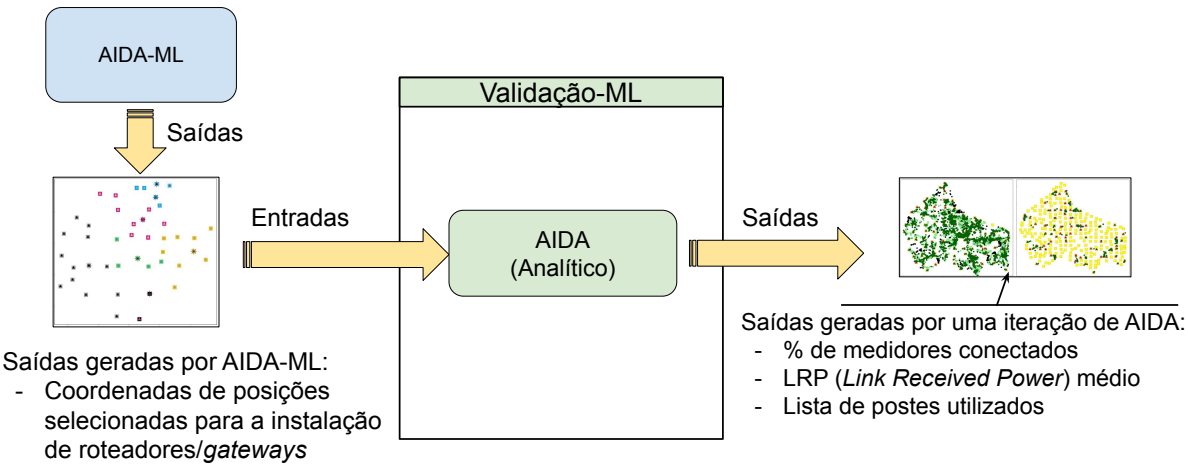


Figura 32 – Estrutura do processo Validação-ML.

Fonte: Adaptado de (MOCHINSKI et al., 2024).

A Tabela 13 apresenta um comparativo entre as quantidades de CPs selecionadas pelo processo de classificação para os diferentes conjuntos de *features*.

Tabela 13 – Tabela comparativa de quantidades de CPs selecionadas por AIDA-ML para experimentos efetuados com diferentes quantidades de *features*. Experimentos com XGBoost e n_estimators=500.

		AIDA Analítico		AIDA-ML		Total de CPs selecionadas pelo processo de classificação				
Cidade	SMs	Iter.	CPs AIDA analítico	Iter.	CPs computadas por AIDA-ML	318 *features*	20 *features*	40 *features*	80 *features*	120 *features*
AGUDOS DO SUL	5383	2	44	3	230	34	36	34	35	38
ARAUCARIA	52836	3	56	3	237	48	55	50	49	51
BALSA NOVA	6106	2	44	3	346	41	44	45	43	38
CAMPO DO TENENTE	4224	2	65	3	449	62	61	61	64	58
CARAMBEI	7512	2	41	3	286	43	38	40	39	37
CONTENDA	10319	1	43	3	436	56	60	56	65	61
FAZENDA RIO GRANDE	56157	3	44	3	146	36	41	41	38	36
GUAMIRANGA	3727	2	67	3	408	60	58	62	62	58
IMBITUVA	10171	2	61	3	380	51	52	53	52	51
INACIO MARTINS	5275	2	142	3	869	122	133	121	122	125
IRATI	23027	2	105	3	575	78	83	86	83	80
IVAI	6392	2	117	3	749	107	102	109	108	115
LAPA	19339	2	171	3	1208	152	165	151	159	154
MANDIRITUBA	11689	2	62	3	384	52	55	58	57	51
PALMEIRA	14887	2	164	3	1260	163	169	168	156	158
PIEN	5303	2	39	3	237	33	39	38	35	37
PONTA GROSSA	150951	3	249	3	1227	200	192	209	209	198
PORTO AMAZONAS	2375	2	47	3	367	45	42	40	45	45
PRUDENTOPOLIS	18982	2	196	3	1257	175	187	194	190	193
QUITANDINHA	6761	2	55	3	337	51	56	52	53	51
REBOUCAS	5224	2	53	3	327	46	52	51	53	46
RIO AZUL	5309	2	65	3	447	64	57	66	67	60
RIO NEGRO	1610	2	38	3	254	33	35	34	33	37
SAO JOAO DO TRIUNFO	6241	2	86	3	587	79	92	78	82	78
SAO MATEUS DO SUL	21583	2	169	3	1226	146	162	150	167	159
TEIXEIRA SOARES	4854	2	62	3	491	61	58	57	59	57

Experimentos de validação com AIDA Analítico adaptado para uma única iteração, ou seja, com Validação-ML, foram efetuados considerando as diferentes quantidades de CPs selecionadas pelo processo de Classificação para *datasets* com diferentes conjuntos de *features* (Tabela 13) e estão disponíveis no Apêndice B.

Na Tabela 14 são apresentados os resultados médios obtidos pelo método de Validação-ML com as diferentes configurações e serve para selecionar a abordagem com melhores resultados.

Tabela 14 – Resultados médios obtidos com Validação-ML (validação com AIDA analítico) para quantidades de CPs definidas em experimentos com AIDA-ML considerando diferentes quantidades de *features*.

		Valores Médios					
	Item	AIDA	TopN20	TopN40	TopN80	TopN120	318 *features*
	CPs selecionadas (para AIDA, considera valor de TD; para os demais (AIDA-ML), considera o total médio de CPs classificadas como selecionadas)	88	82	81	82	80	79
	Quant. de cidades cujo número de CPs selecionadas por AIDA-ML é superior ao de AIDA analítico	N/A	4	4	2	1	2
Abordagem TD	(TD) CPs usadas (média)	88	78	77	77	76	75
	(TD) % médio SMs não conectados	0,262	3,485	3,313	3,231	3,588	4,308
	(TD) LRP Médio	-76,228	-74,342	-74,740	-74,654	-74,750	-75,151
	(TD) Quant. de cidades com 2% ou menos de SMs não conectados	26	13	14	12	14	13
Abordagem BU	(BU) CPs usadas (média)	99	81	80	81	79	78
	(BU) % médio SMs não conectados	0,258	2,935	2,659	2,843	3,236	3,383
	(BU) LRP Médio	-69,731	-71,297	-71,327	-71,453	-71,757	-72,131
	(BU) Quant. de cidades com 2% ou menos SMs não conectados	26	14	16	14	15	15

A estrutura de *dataset* que apresentou melhores resultados no processo de Validação-ML foi a TopN40 (que utiliza *dataset* com 40 *features*), pois apresentou melhores resultados que os demais nos seguintes itens: quantidade de cidades com 2% ou menos de medidores não conectados e percentual médio de medidores não conectados, ambos para as abordagens TD e BU.

Analisando as *features* que compõem esse conjunto de 40 características mais relevantes, é importante destacar:

- 15 *features* (exemplo: I_Dist_CP_CP1, I_Dist_CP_CP2, entre outras) se referem a características de CPs da iteração anterior, sugerindo que é importante observar, na classificação de uma CP, as posições de iterações anteriores que podem concorrer pelos mesmos recursos, no caso os mesmos medidores a serem conectados;
- 5 *features* contém informações sobre variações na elevação do terreno (como L_DiffElev_E_200m e L_DiffElev_NW_1000m, entre outras), indicando que a topografia é um fator importante na definição da classe;
- 6 características se referem a informações sobre regiões no entorno (como R_LRP_R2 e R_SMs_R1, por exemplo).
- Outras características trazem informações sobre a área da região (G_BboxArea_SMs), o total de medidores da região da CP (L_SMs_CP), o percentual de medidores na área da CP em relação ao total de medidores a serem conectados (G_perc_SMs) e informações sobre postes da região (L_Postes_CP, L_CentroidePostes_LON).
- No geral, as *features* procuram sintetizar informações sobre topografia, qualidade de sinal e quantidade de dispositivos no entorno da região em análise.

6.5 ANÁLISE DE AIDA-ML UTILIZANDO *DATASETS* COM 40 *FEATURES*

Nesta seção são apresentados os resultados de experimentos efetuados com o *dataset* TopN40, com 40 *features*, com o objetivo de avaliar o comportamento desse cenário ao efetuar variações na quantidade de CPs selecionadas como uma fase de pós-processamento.

Duas estratégias foram avaliadas: i) Aumentar a quantidade de CPs selecionadas além da quantidade observada quando se analisa a classificação original obtida com XGBoost; ii) Avaliar diferentes pontos de corte de probabilidade (ajuste de *threshold*) para determinação da classe adotada por XGBoost. Nesse último caso, inicialmente é importante destacar que, de uma forma padrão, quando o XGBoost faz a definição do valor do atributo *target* (classe) definindo-o como 0 ou 1, faz isso a partir da determinação de uma probabilidade para a classe. Instâncias cuja probabilidade de $classe = 1$ (PROBA_1) é maior ou igual a 50% (0,50) são diretamente definidos como 1 (positivas). No entanto, considerando que o volume de dados disponíveis para treinamento é muito limitado, ou então quando há um desbalanceamento de valores do atributo *target* (quantidade de posições selecionadas << quantidade de posições candidatas), pode-se explorar o comportamento para diferentes pontos de corte. Nos experimentos adicionais efetuados com o *dataset* de 40 *features*, foram utilizados PROBA_1 maior ou igual a 0,20 e maior ou igual a 0,30.

Experimentos de validação com AIDA analítico foram efetuados considerando as diferentes abordagens de definição de número de CPs selecionadas após processo de classificação para *datasets* com 40 *features* e estão disponíveis no Apêndice C.

Na Figura 33 é possível visualizar uma comparação entre as posições candidatas selecionadas pelos métodos AIDA analítico e AIDA-ML para uma mesma região. Pode-se notar que, para a maioria dos casos, as posições selecionadas pelo método de *machine learning* coincidem com as do método analítico. Quando não coincidem, o método de *machine learning* é capaz de selecionar posições próximas capazes de assegurar a cobertura. No caso da figura, o método analítico posicionou 105 dispositivos deixando 0,074% de medidores não conectados com a abordagem TD, totalizando 17 medidores. O método AIDA-ML selecionou 99 posições, e avaliações com o método Validação-ML e abordagem TD demonstraram que deixou de conectar 0,573% dos medidores (total 132 medidores), dentro do limite estabelecido parametrizado como 2%. O valor médio de LRP obtido para essa região foi de -66,807 dBm, portanto, dentro da faixa que estabelece enlaces com capacidade de comunicação.

Na Tabela 15 são apresentados os resultados médios obtidos com as diferentes configurações e serve para identificar a abordagem com melhores resultados.

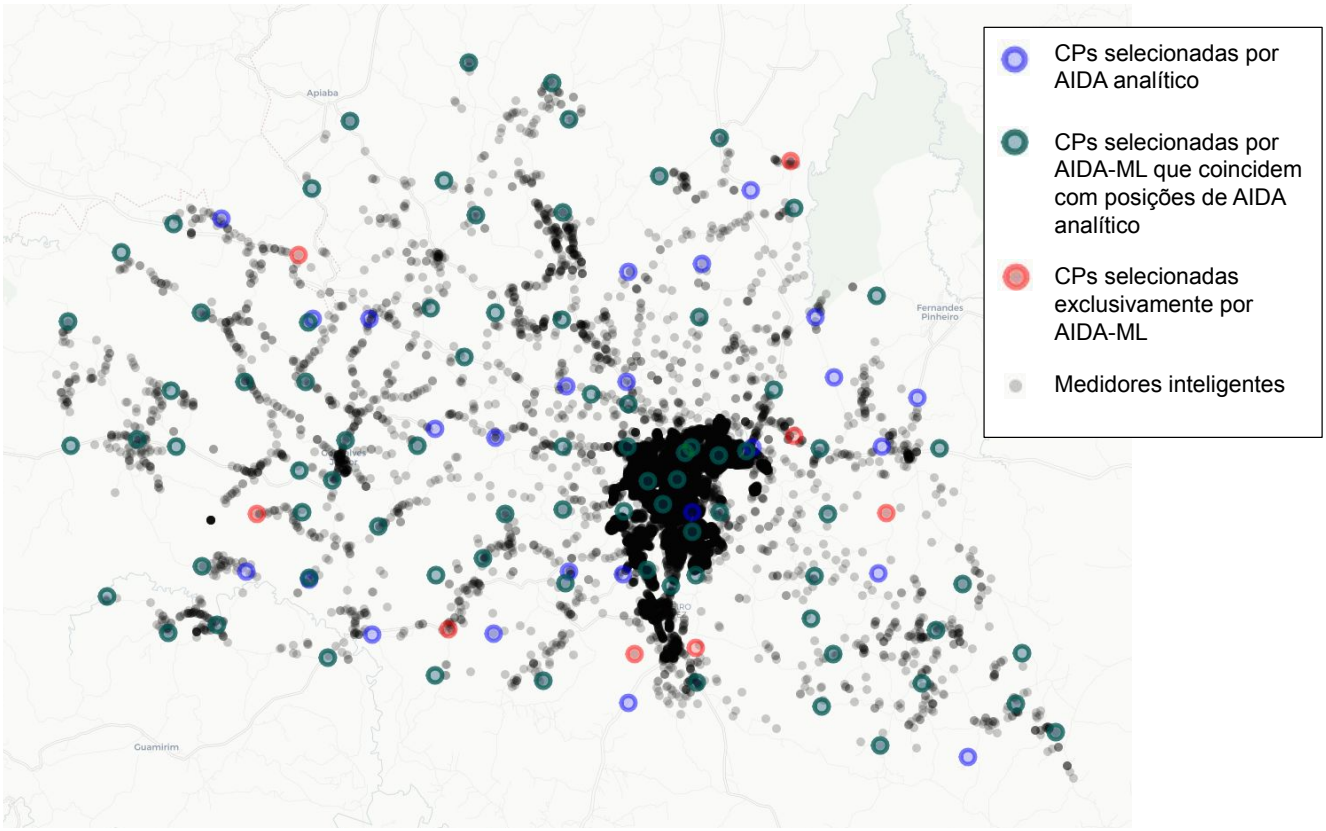


Figura 33 – Comparativo de seleção de CPs entre método AIDA analítico e AIDA-ML.

Fonte: Autoria própria.

Tabela 15 – Resultados médios obtidos com Validação-ML (validação com AIDA analítico) para quantidades de CPs definidas em experimentos com AIDA-ML considerando 40 *features*.

			Valores Médios (AIDA-ML com 40 features)				
	Item	AIDA	TopN40	+10% CPs	+15% CPs	PROBA_1 $\geq$ 0.20	PROBA_1 $\geq$ 0.30
	CPs selecionadas (para AIDA, considera valor de TD; para os demais (AIDA-ML), considera o total MÉDIO de CPs classificadas como selecionadas)	88	81	89	94	94	88
	Quant. de cidades cujo número de CPs selecionadas por AIDA-ML é superior ao de AIDA analítico	N/A	4	15	20	18	10
Abordagem TD	(TD) CPs usadas (média)	88	77	83	86	86	82
	(TD) % médio SMs não conectados	0,262	3,313	1,770	1,455	1,616	1,969
	(TD) LRP Médio	-76,228	-74,740	-73,790	-73,350	-73,665	-73,917
	(TD) Quant. de cidades com 2% ou menos de SMs não conectados	26	14	19	19	19	18
Abordagem BU	(BU) CPs usadas (média)	99	80	88	92	92	87
	(BU) % médio SMs não conectados	0,258	2,659	1,387	1,041	1,233	1,558
	(BU) LRP Médio	-69,731	-71,327	-69,818	-69,099	-69,454	-70,026
	(BU) Quant. de cidades com 2% ou menos SMs não conectados	26	16	20	23	21	19

Um dos requisitos principais do processo de posicionamento de roteadores/*gateways* considerados nesta tese visa obter um percentual máximo de medidores não conectados igual ou inferior a 2% (que corresponde ao limite estabelecido para o método AIDA Analítico). Ao observar os valores obtidos com AIDA-ML (Tabela 15), e ao avaliar os

resultados da coluna ”+15% CPs” (que corresponde a seleção de mais 15% de CPs que o estabelecido por PROBA_1 >= 0.5), nota-se que os percentuais de medidores não conectados resultaram em 1,455% para a abordagem TD e 1,041% na abordagem BU nas validações das CPs estabelecidas pelo processo de classificação de AIDA-ML.

Do ponto de vista de potência recebida (LRP), é possível observar na mesma Tabela 15, na coluna ”+15% CPs”, que o valor de LRP médio obtido pela abordagem TD é de -73,350 dBm. Ao comparar esse valor com o obtido por AIDA analítico (-76,228 dBm) pode-se dizer que a potência média recebida obtida com AIDA-ML é cerca de 3,8% maior. Evidentemente, nesse caso, é necessário considerar aspectos relativos a custo de instalação versus qualidade percebida. Isso porque a quantidade de roteadores demandada por AIDA-ML é maior e a cobertura (apesar de estar acima dos 98% exigidos) ainda é inferior à obtida com AIDA analítico.

6.6 ANÁLISE DE TEMPO DE EXECUÇÃO E QUANTIDADE DE CÁLCULOS DE LRP

Nesta seção é apresentada uma análise comparativa do tempo de execução e da quantidade de cálculos de LRP dos métodos AIDA analítico e AIDA-ML.

A Tabela 16 apresenta os tempos de processamento e a quantidade de cálculos de LRP demandados pelos métodos para o posicionamento de roteadores/*gateways* para as 26 cidades do experimento. Para os experimentos com AIDA-ML, foram considerados *datasets* com 40 *features*. Os experimentos foram executados em um servidor com sistema operacional Linux, *kernel version* #44 20.04.2-Ubuntu SMP, *release* 5.11.0-40-generic, x86_64, Intel(R) Xeon(R) Gold 5220R CPU @2.20GHz, 96 CPU(s), 396 GB RAM.

Em relação ao tempo de processamento, destaca-se que o método AIDA-ML é capaz de completar a seleção de posições candidatas em um tempo médio que corresponde a apenas 12,40% do tempo requerido pelo método analítico; ou seja, um ganho de 87,60% no tempo de processamento em relação ao que seria exigido pelo método analítico AIDA.

Os resultados da Tabela 16 também destacam que, com o uso do método de *machine learning*, foi possível reduzir significativamente o tamanho do espaço de busca de conexões entre medidores e postes em comparação com o método analítico. Essa redução no espaço de busca contribui para a notável diminuição no tempo de processamento, pois elimina a necessidade de calcular a potência recebida estimada no enlace (LRP) para todas as conexões possíveis entre medidores e posições de postes selecionados para cada região analisada. Ao invés disso, o AIDA-ML se concentra na geração de características para um conjunto mínimo de *features* (no caso, 40 *features*) para cada posição candidata.

Considerando as 26 regiões dos experimentos, o método heurístico efetuou 4.215.064

cálculos de LRP. No entanto, ao considerar os experimentos com os conjuntos de dados de 40 *features*, o AIDA-ML exigiu o cálculo de apenas 132.480 valores de LRP, resultando em uma redução de 96,86% na quantidade total de cálculos. Ao posicionar roteadores/*gateways* para a região com maior número de medidores no experimento (Ponta Grossa, com 150.951 medidores), o método analítico AIDA calcula 1.999.370 valores de LRP para diversas conexões entre medidores, bem como entre medidores e posições candidatas (CPs), usando três iterações. O método AIDA-ML, por sua vez, necessita calcular características para apenas 1.227 posições candidatas para três iterações simuladas. Sabendo-se que a configuração com 40 características é a que produz melhores resultados, o número total de *features* dependentes de cálculos de LRP a serem computados foi equivalente a 1.227×9 (pois apenas 9 características se referem a valores de LRP), resultando em 11.043 cálculos de LRP. Da mesma forma, para outra região com menos medidores, como a cidade de

Tabela 16 – Ganho de AIDA-ML vs AIDA em relação ao tempo de processamento e ao tamanho do espaço de busca. As regiões são classificadas em ordem decrescente de ganho no tempo de processamento. O número de cálculos de LRP para AIDA-ML se referem a *datasets* com 40 *features*. Tempos de processamento em (hh:mm:ss).

			Tempo de processamento			**#Cálculos de LRP**		
Região	**Cidade**	**#SMs**	**AIDA**	**AIDA-ML**	**Ganho de AIDA-ML**	**AIDA**	**AIDA-ML**	**Ganho de AIDA-ML**
7	FAZENDA RIO GRANDE	56157	05:23:00	0:12:30	96,13%	675.148	1.314	99,81%
2	ARAUCARIA	52836	04:28:38	0:10:36	96,05%	490.386	2.133	99,57%
17	PONTA GROSSA	150951	20:06:56	1:42:40	91,49%	1.999.370	11.043	99,45%
14	MANDIRITUBA	11689	00:39:12	0:05:09	86,86%	52.324	3.456	93,40%
11	IRATI	23027	01:29:44	0:12:30	86,07%	175.392	5.175	97,05%
9	IMBITUVA	10171	00:37:10	0:05:13	85,96%	48.642	3.420	92,97%
16	PIEN	5303	00:17:59	0:02:38	85,36%	24.428	2.133	91,27%
5	CARAMBEI	7512	00:26:17	0:03:57	84,97%	40.774	2.574	93,69%
1	AGUDOS DO SUL	5383	00:17:15	0:02:45	84,06%	21.327	2.070	90,29%
6	CONTENDA	10319	00:34:05	0:05:48	82,98%	54.296	3.924	92,77%
20	QUITANDINHA	6761	00:22:54	0:03:58	82,68%	30.743	3.033	90,13%
3	BALSA NOVA	6106	00:20:34	0:04:18	79,09%	23.051	3.114	86,49%
21	REBOUCAS	5224	00:17:59	0:04:30	74,98%	20.287	2.943	85,49%
13	LAPA	19339	01:16:06	0:20:50	72,62%	106.329	10.872	89,78%
25	SAO MATEUS DO SUL	21583	01:20:23	0:22:36	71,88%	130.876	11.034	91,57%
22	RIO AZUL	5309	00:18:42	0:05:16	71,84%	20.307	4.023	80,19%
24	SAO JOAO DO TRIUNFO	6241	00:21:53	0:06:48	68,93%	26.374	5.283	79,97%
19	PRUDENTOPOLIS	18982	01:07:37	0:21:29	68,23%	99.522	11.313	88,63%
26	TEIXEIRA SOARES	4854	00:16:40	0:05:19	68,10%	18.324	4.419	75,88%
8	GUAMIRANGA	3727	00:12:23	0:04:18	65,28%	12.937	3.672	71,62%
15	PALMEIRA	14887	00:53:35	0:19:15	64,07%	76.817	11.340	85,24%
4	CAMPO DO TENENTE	4224	00:14:13	0:05:40	60,14%	13.772	4.041	70,66%
12	IVAI	6392	00:21:52	0:08:58	58,99%	21.731	6.741	68,98%
23	RIO NEGRO	1610	00:06:01	0:02:40	55,68%	6.097	2.286	62,51%
18	PORTO AMAZONAS	2375	00:08:04	0:04:17	46,90%	5.417	3.303	39,03%
10	INACIO MARTINS	5275	00:17:39	0:10:27	40,79%	20.393	7.821	61,65%
	Total	**466237**	**42:16:51**	**05:14:30**	**87,60%**	**4.215.064**	**132.480**	**96,86%**

Pien com 5.303 medidores, o método analítico AIDA realiza cálculos de LRP para 24.428 conexões, enquanto o método AIDA-ML requer apenas cálculos para 2.133 *features* de LRP no conjunto de dados de classificação. A disparidade no número de conexões avaliadas por AIDA e por AIDA-ML tende a ser significativa na maioria dos casos.

Considerando os valores totais da Tabela 16, estima-se uma média de 1,626 h (1h37min34s) para o processamento de cada cidade com o método analítico. Considerando que o estado do Paraná possui 399 municípios, pode-se estimar, então, que o tempo total de processamento com AIDA seria de, aproximadamente, 648 horas. O método AIDA-ML, por sua vez, efetuaria esse cálculo em cerca de 80,5 horas.

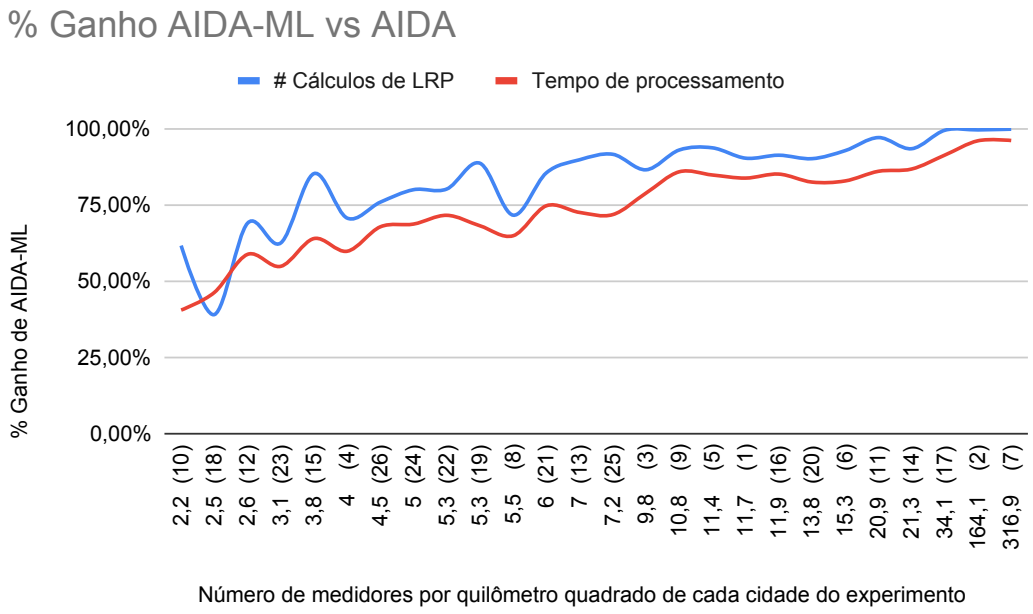


Figura 34 – Análise da variação do ganho percentual de AIDA-ML (comparado a AIDA) em relação ao tempo de processamento e ao número de cálculos de LRP. Os valores entre parênteses identificam as cidades às quais os valores pertencem.

Fonte: Adaptado de (MOCHINSKI et al., 2024).

Pode-se dizer, também, que AIDA-ML tem um desempenho melhor que AIDA no que diz respeito ao tempo de processamento e ao número de cálculos de LRP à medida que o número de medidores inteligentes por km^2 aumenta (Figura 34). Observando essa figura, em relação ao tempo de processamento, o menor ganho de AIDA-ML foi observado para Inácio Martins (10), com ganho de 40,79%. E, quanto ao número de cálculos de LRP, o menor ganho de AIDA-ML (39,03%) foi computado para Porto Amazonas (18).

6.7 EXPERIMENTOS ADICIONAIS COM TÉCNICAS DE AUTOML, OTIMIZAÇÃO DE HIPERPARÂMETROS E SELEÇÃO DE *FEATURES*

Nas seções anteriores, foi evidenciado que o uso de um *dataset* com 40 *features* selecionadas a partir da análise da importância das *features* para o modelo de treinamento (em especial na combinação "TopN40 + 15% CPs") é capaz de fornecer resultados melhores em termos de cobertura de conexão que os obtidos com o uso de um *dataset* original com todas as características do cenário. Experimentos avaliando o tempo de processamento e o espaço de busca também demonstram grande ganho em relação ao método heurístico AIDA.

Nesta seção, são descritos os experimentos adicionais realizados com o objetivo de verificar se uma combinação diferente de características, o uso de hiperparâmetros diferentes para XGBoost ou mesmo o uso de outro algoritmo de aprendizagem é capaz de alcançar melhoria nos resultados, em especial quanto à acurácia e cobertura de conexão dos medidores inteligentes.

Em relação ao processo de otimização de hiperparâmetros (HPO) e análise de diferentes algoritmos, os experimentos adicionais incluíram:

- O uso de técnica de *Grid Search* para a otimização de hiperparâmetros.
- O uso da biblioteca *Scikit-Optimize* (*skopt*) considerando as funções *skopt.dummy_minimize* (busca aleatória por amostragem uniforme dentro de limites estabelecidos), *skopt.gbrt_minimize* (otimização sequencial usando *gradient-boosted trees*) e *skopt.gp_minimize* (otimização bayesiana usando processos gaussianos) para a otimização de hiperparâmetros.
- O uso de técnicas de AutoML, incluindo o uso de funções das bibliotecas *Auto-sklearn* e *TPOT*. O objetivo de uso de AutoML visa avaliar o processo de treinamento de forma a verificar se outros algoritmos de ML, além do XGBoost, podem apresentar desempenho melhor que os obtidos anteriormente.

Em relação ao processo de seleção de *features*, os experimentos adicionais incluíram:

- Experimentos com *wrappers* com o objetivo de buscar uma combinação de *features* capaz de obter melhores valores de acurácia com o protocolo LOSO que os obtidos anteriormente.

Os diferentes experimentos realizados nesta etapa visam verificar a possibilidade de obtenção de resultados melhores em relação aos valores obtidos com o *dataset* "TopN40

+ 15% CPs", com o qual foi possível obter, nos experimentos principais deste estudo, e com a abordagem BU, um percentual médio de medidores não conectados igual a 1,041% com uma média de 92 CPs usadas por cidade.

Nesta etapa, essa combinação "TopN40 + 15% CPs" será designada de *baseline*, servindo, portanto, de parâmetro base de comparação em relação aos demais experimentos realizados. Para essa *baseline*, a acurácia obtida com o protocolo LOSO nos experimentos principais foi de 0,8589 com os hiperparâmetros de XGBoost iguais a: *tree_method='hist', learning_rate=1, max_depth=15, reg_lambda=20, n_estimators=500*.

As tabelas com os valores de acurácia obtidos com as diferentes estratégias utilizadas nos experimentos adicionais estão disponíveis no Apêndice D, com resultados para os experimentos realizados com 40 e 318 *features*, e também para 36 e 79 *features* (selecionadas pelo processo de seleção sequencial de *features*). Ao todo foram realizados 37 experimentos para avaliação de acurácia, sendo: 18 experimentos com *datasets* de 40 *features*, 13 experimentos com *datasets* de 318 *features*, 3 experimentos com *datasets* de 36 *features* e 3 experimentos com *datasets* de 79 *features*.

Em experimentos com seleção sequencial de *features* (*sequential feature selection*) na modalidade de busca *Backward* (SBS), cujo processo inicia com a análise de todas as características e, gradualmente extrai uma *feature* e analisa os valores de acurácia, o melhor valor foi encontrado para um total de 36 features (Figura 35), com uma acurácia de 0,8729 em experimentos com validação cruzada com 5 *folds*. Com o protocolo LOSO, experimentos realizados em *datasets* com as 36 *features* selecionadas pelo método SBS alcançaram uma acurácia de 0,8634 (Tabela 32 do Apêndice D) após processo de HPO com *skopt.dummy_minimize*. Para a estratégia de busca *Forward* (SFS), que inicia com nenhuma *feature* e, gradualmente vai incrementando o número de características, o melhor resultado foi alcançado com 79 *features*, com uma acurácia de 0,8770 (Figura 36) em experimentos com validação cruzada com 5 *folds*. Com o protocolo LOSO, a melhor acurácia obtida em experimentos realizados com *dataset* contendo as 79 *features* selecionadas pelo método SFS foi de 0,8678 (Tabela 33 do Apêndice D) após processo de HPO com *skopt.dummy_minimize*.

Em relação aos experimentos com 318 *features*, é possível observar na Tabela 31 do Apêndice D que a melhor acurácia obtida foi de 0,8646 em experimentos com busca aleatória de hiperparâmetros (*skopt.dummy_minimize*).

Analisando as tabelas do Apêndice D, observa-se que os maiores valores de acurácia foram obtidos com *datasets* de 40 *features*. Por isso, para a comparação com a *baseline*, este estudo focará na análise dos valores da Tabela 30 do apêndice.

Na Tabela 17 (que sumariza os melhores resultados encontrados na Tabela 30 do Apêndice D), observa-se que os três maiores valores de acurácia obtidos com os

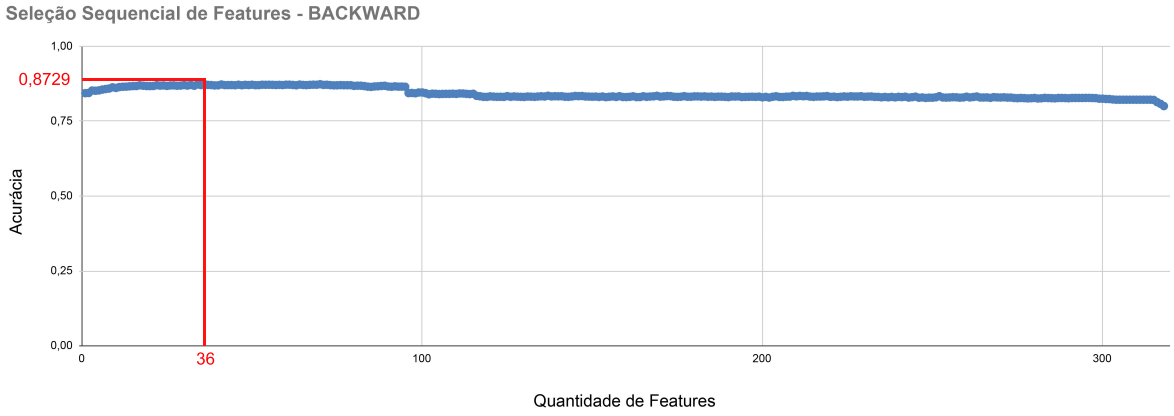


Figura 35 – Resultado de processo de seleção sequencial de *features* - *Backward*. Melhor valor de acurácia obtido para 36 *features* com validação cruzada com 5 *folds*.

Fonte: Autoria própria.

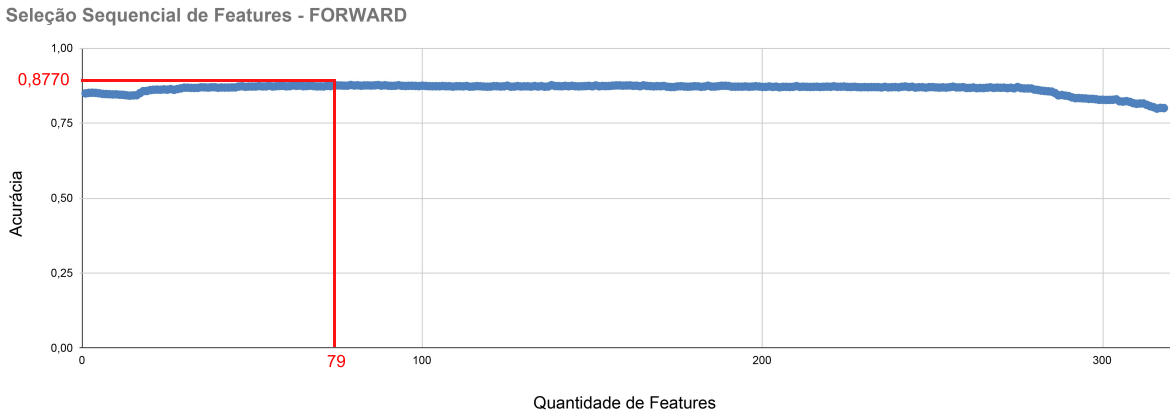


Figura 36 – Resultado de processo de seleção sequencial de *features* - *Forward*. Melhor valor de acurácia obtido para 79 *features* com validação cruzada com 5 *folds*.

Fonte: Autoria própria.

experimentos adicionais para *datasets* com 40 *features* foram: 0,8734 (obtido com XGBoost, estratégia de busca aleatória de hiperparâmetros e 500 iterações); 0,8728 (obtido com XGBoost, estratégia de busca aleatória de hiperparâmetros e 100 iterações); e 0,8725 (obtido com LightGBM, otimização bayesiana usando processos gaussianos para a busca de hiperparâmetros, e 500 iterações).

Em comparação com os experimentos realizados com 36, 79 e 318 *features*, nota-se que acurácias obtidas nos experimentos adicionais com 40 features se destacam em relação às demais.

As três configurações que produziram maior acurácia para 40 *features* serão, então, usadas como referência para comparação com os valores obtidos pela *baseline*, e passarão a ser referidas a partir deste ponto do texto como:

- REF1 - estratégia com a melhor acurácia.
- REF2 - estratégia com a 2^a melhor acurácia.
- REF3 - estratégia com a 3^a melhor acurácia.

Tabela 17 – Comparativo entre os 3 maiores valores de acurácia obtidos com os experimentos adicionais de AIDA-ML e a *baseline*.

Features	**Técnica**	**Acurácia (LOSO)**	**Classificador**	**Hiperparâmetros**
40	*BASELINE*	0,8589	XGBoost (xgboost.XGBClassifier)	tree_method='hist', learning_rate=1, max_depth=15, reg_lambda=20, n_estimators=500
40	(REF1) skopt.dummy_minimize (Busca aleatória, 500 calls)	0,8734	XGBoost (xgboost.XGBClassifier)	learning_rate = 0,02504, n_estimators = 2238, max_depth = 12, min_child_weight = 11, subsample = 0,70774, colsample_bynode = 0,30457, num_parallel_tree = 2, gamma = 1,92242, colsample_bytree = 0,61318
40	(REF2) skopt.dummy_minimize (Busca aleatória, 100 calls)	0,8728	XGBoost (xgboost.XGBClassifier)	learning_rate = 0,01661, n_estimators = 1354, max_depth = 11, min_child_weight = 10, subsample = 0,74514, colsample_bynode = 0,33373, num_parallel_tree = 6, gamma = 1,51908, colsample_bytree = 0,868934
40	(REF3) skopt.gp_minimize (otimização bayesiana usando processos gaussianos, 500 calls)	0,8725	LightGBM (lightgbm.LGBMClassifier)	max_bin = 1227, learning_rate = 0,027542196740280862, n_estimators = 2500, num_leaves = 65

Para comparações com a *baseline*, as posições candidatas selecionadas pelo processo de classificação de AIDA-ML realizado para as três estratégias, foram submetidas ao processo de Validação-ML com o objetivo de avaliar o percentual de cobertura que cada estratégia seria capaz de alcançar. Adicionalmente, além das posições candidatas originais de REF1, REF2 e REF3, experimentos complementares foram realizados com a adição de mais "15% de CPs", gerando as configurações "REF1 + 15% CPs", "REF2 + 15% CPs" e "REF2 + 15% CPs", submetendo-as também ao processo de Validação-ML.

A Tabela 18 mostra um comparativo dos resultados consolidados de Validação-ML obtidos pelos experimentos principais realizados com 40 *features* (TopN40 e "TopN40 + 15% CPs") e os valores obtidos com validações realizadas com REF1, REF2 e REF3 e suas variações.

Comparando o valor de acurácia obtida por REF1 (0,8734) com a acurácia da *baseline* (0,8589), observa-se um ganho de 1,688%. No entanto, apesar de as acurácias calculadas com o protocolo LOSO terem sido maiores para REF1, REF2 e REF3 quando comparadas ao valor obtido pela *baseline*, o que se observa em relação aos percentuais de cobertura calculados com Validação-ML é que a configuração utilizada pela *baseline* é capaz de alcançar com a abordagem BU (*Bottom-Up*) um percentual médio de medidores não conectados igual a 1,041%, enquanto esse percentual é de 1,235%, 1,459% e 1,436% para REF1, REF2 e REF3, respectivamente.

Tabela 18 – Comparativo de desempenho de resultados com 40 *features* versus resultados após HPO (REF1, REF2 e REF3). Os valores foram computados com o uso do método de Validação-ML.

			Valores Médios (AIDA-ML com 40 features vs. resultados após HPO)							
	Item	AIDA	TopN40	TopN40 +15% CPs	REF1	REF1 +15% CPs	REF2	REF2 +15% CPs	REF3	REF3 +15% CPs
	CPs selecionadas (para AIDA, considera valor de TD; para os demais (AIDA-ML), considera o total médio de CPs classificadas como selecionadas)	88	81	94	79	92	80	92	79	92
	Quant. de cidades cujo número de CPs selecionadas por AIDA-ML é superior ao de AIDA analítico	N/A	4	20	1	19	1	17	1	17
Abordagem TD	(TD) CPs usadas (média)	88	77	86	76	85	76	86	76	85
	(TD) % médio SMs não conectados	0,262	3,313	1,455	3,787	1,744	3,620	2,052	2,993	1,972
	(TD) LRP Médio	-76,228	-74,740	-73,350	-74,997	-73,941	-74,876	-73,973	-74,675	-74,107
	(TD) Quant. de cidades com 2% ou menos de SMs não conectados	26	14	19	13	19	14	18	13	18
Abordagem BU	(BU) CPs usadas (média)	99	80	92	78	91	79	91	78	90
	(BU) % médio SMs não conectados	0,258	2,659	**1,041**	3,271	**1,235**	3,085	**1,459**	2,261	**1,436**
	(BU) LRP Médio	-69,731	-71,327	-69,099	-72,097	-69,898	-71,909	-69,913	-71,495	-70,097
	(BU) Quant. de cidades com 2% ou menos SMs não conectados	26	16	23	14	22	15	20	15	21

O mapa de calor apresentado na Figura 37 mostra os percentuais de medidores não conectados por cidade, calculados para a *baseline* e para REF1, REF2 e REF3. A configuração *baseline* consegue melhores resultados de cobertura com *datasets* acrescidos de 15% de CPs em relação a REF1, REF2 e REF3 em 14 das 26 cidades avaliadas. Isso significa que a diferença de acurácia nos testes com protocolo LOSO ocorre em diferentes conjuntos de cidade. Avaliando as cidades com maiores números de medidores (Ponta Grossa, com 150.951 medidores, Fazenda Rio Grande, com 56.157 medidores, e Araucária, com 52.836 medidores), observa-se que a *baseline* (TopN40 + 15% de CPs) obtém melhores resultados nessas cidades quando comparada aos demais experimentos, com exceção apenas nos testes com REF1 + 15% CPs para a cidade de Araucária.

Merece destaque especial nesse mesmo mapa de calor (Figura 37) os resultados obtidos na faixa central delimitados com uma moldura na cor verde. Pode-se observar que cidades com número de medidores entre 6.106 (Balsa Nova) e 23.027 (Irati) apresentam maiores quantidades de valores grifados em verde, sugerindo melhores resultados de AIDA-ML para cidades nesse intervalo. Vale ressaltar, no entanto, que os valores máximos de não cobertura buscados devem ser menores que 2%; portanto, outros valores grifados em amarelo também são bons resultados. Os valores tendendo à cor vermelha denotam configurações em que a cobertura mínima esperada não é alcançada com o método.

		Percentuais de medidores não conectados							
Cidade	Núm. Medidores	TopN40	TopN40 +15% CPs	REF1	REF1 + 15% CPs	REF2	REF2 + 15% CPs	REF3	REF3 + 15% CPs
PONTA GROSSA	150951	7,612	1,063	10,503	5,128	8,089	5,127	3,638	1,666
FAZENDA RIO GRANDE	56157	5,996	1,813	11,546	3,139	11,557	4,854	4,851	2,929
ARAUCARIA	52836	3,199	1,660	4,086	1,395	4,086	2,487	3,182	2,667
IRATI	23027	0,725	0,599	0,552	0,282	0,617	0,308	0,782	0,512
SAO MATEUS DO SUL	21583	0,978	0,537	0,857	0,338	0,737	0,315	0,917	0,533
LAPA	19339	6,764	1,432	6,572	1,406	1,515	1,401	6,784	6,195
PRUDENTOPOLIS	18982	0,479	0,195	0,390	0,284	0,358	0,227	0,774	0,248
PALMEIRA	14887	1,437	0,343	1,021	0,584	0,967	0,497	1,532	0,544
MANDIRITUBA	11689	0,539	0,248	0,479	0,291	0,479	0,291	0,582	0,359
CONTENDA	10319	0,378	0,116	0,630	0,262	0,378	0,262	0,475	0,262
IMBITUVA	10171	1,376	0,236	8,514	0,403	8,514	0,226	0,462	0,246
CARAMBEI	7512	0,612	0,506	0,612	0,399	0,612	0,399	0,612	0,572
QUITANDINHA	6761	0,562	0,118	0,399	0,104	0,385	0,104	1,139	0,104
IVAI	6392	1,737	1,267	1,392	1,126	1,471	1,126	2,206	1,502
SAO JOAO DO TRIUNFO	6241	1,330	0,336	1,106	0,689	1,058	0,545	1,041	0,609
BALSA NOVA	6106	0,311	0,295	0,328	0,311	0,328	0,311	1,752	0,311
AGUDOS DO SUL	5383	9,530	3,771	7,449	3,418	7,449	3,028	5,759	3,938
RIO AZUL	5309	0,603	0,490	3,899	0,170	3,974	3,748	4,125	1,582
PIEN	5303	0,924	0,754	1,207	0,849	1,207	0,849	1,207	0,849
INACIO MARTINS	5275	7,052	5,460	7,393	5,592	7,147	5,763	6,521	5,156
REBOUCAS	5224	0,708	0,632	0,727	0,689	0,727	0,670	0,708	0,651
TEIXEIRA SOARES	4854	3,028	2,658	2,905	0,762	2,905	0,742	0,845	0,680
CAMPO DO TENENTE	4224	2,060	0,568	2,131	1,870	2,249	1,894	2,131	1,563
GUAMIRANGA	3727	2,844	0,563	2,415	1,261	5,983	1,395	3,086	1,422
PORTO AMAZONAS	2375	1,768	0,589	1,347	0,421	0,842	0,421	1,389	1,305
RIO NEGRO	1610	6,584	0,807	6,584	0,932	6,584	0,932	2,298	0,932
MÉDIA:	17932	2,659	1,041	3,271	1,235	3,085	1,459	2,261	1,436

Figura 37 – Mapa de calor comparando percentuais de medidores não conectados obtidos pela *baseline* e experimentos REF1, REF2 e REF3. Os dados estão classificados por ordem decrescente de número de medidores, e as cidades com maiores quantidades de medidores estão em destaque. O quadro central destaca a faixa de cidades em que os resultados obtidos com AIDA-ML são melhores.

Fonte: Autoria própria.

Sob o ponto de vista de totais de medidores conectados (Tabela 19), pode-se observar que a *baseline* com (TopN40 + 15% CPs) totaliza 461.096 medidores conectados e o segundo melhor total é obtido por REF3 + 15% CPs, que totalizou 457.967 medidores conectados do total geral de 466.237 medidores, conseguindo melhores resultados que a *baseline* em 5 cidades.

Em relação ao total de CPs selecionadas, a Tabela 18 mostra que a *baseline* seleciona uma média de 92 CPs por cidade (em média) para a abordagem BU, enquanto as demais abordagens (REF1 + 15% CPs, REF2 + 15% CPs e REF3 + 15% CPs) selecionam 91, 91 e 90 CPs, respectivamente. Esse posicionamento de menor quantidade de CPs pode justificar o motivo pelo qual a cobertura com esses métodos é ligeiramente menor que a

Tabela 19 – Total de medidores conectados obtidos pela *baseline* e experimentos REF1, REF2 e REF3.

Cidade	**Núm. Medidores**	**TopN40**	**TopN40 + 15%CPs**	**REF1**	**REF1 + 15% CPs**	**REF2**	**REF2 + 15% CPs**	**REF3**	**REF3 + 15% CPs**
AGUDOS DO SUL	5383	4870	5180	4982	5199	4982	5220	5073	5171
ARAUCARIA	52836	51146	51959	50677	52099	50677	51522	51155	51427
BALSA NOVA	6106	6087	6088	6086	6087	6086	6087	5999	6087
CAMPO DO TENENTE	4224	4137	4200	4134	4145	4129	4144	4134	4158
CARAMBEI	7512	7466	7474	7466	7482	7466	7482	7466	7469
CONTENDA	10319	10280	10307	10254	10292	10280	10292	10270	10292
FAZENDA RIO GRANDE	56157	52790	55139	49673	54394	49667	53431	53433	54512
GUAMIRANGA	3727	3621	3706	3637	3680	3504	3675	3612	3674
IMBITUVA	10171	10031	10147	9305	10130	9305	10148	10124	10146
INACIO MARTINS	5275	4903	4987	4885	4980	4898	4971	4931	5003
IRATI	23027	22860	22889	22900	22962	22885	22956	22847	22909
IVAI	6392	6281	6311	6303	6320	6298	6320	6251	6296
LAPA	19339	18031	19062	18068	19067	19046	19068	18027	18141
MANDIRITUBA	11689	11626	11660	11633	11655	11633	11655	11621	11647
PALMEIRA	14887	14673	14836	14735	14800	14743	14813	14659	14806
PIEN	5303	5254	5263	5239	5258	5239	5258	5239	5258
PONTA GROSSA	150951	139461	149346	135097	143210	138740	143211	145459	148436
PORTO AMAZONAS	2375	2333	2361	2343	2365	2355	2365	2342	2344
PRUDENTOPOLIS	18982	18891	18945	18908	18928	18914	18939	18835	18935
QUITANDINHA	6761	6723	6753	6734	6754	6735	6754	6684	6754
REBOUCAS	5224	5187	5191	5186	5188	5186	5189	5187	5190
RIO AZUL	5309	5277	5283	5102	5300	5098	5110	5090	5225
RIO NEGRO	1610	1504	1597	1504	1595	1504	1595	1573	1595
SAO JOAO DO TRIUNFO	6241	6158	6220	6172	6198	6175	6207	6176	6203
SAO MATEUS DO SUL	21583	21372	21467	21398	21510	21424	21515	21385	21468
TEIXEIRA SOARES	4854	4707	4725	4713	4817	4713	4818	4813	4821
TOTAL:	**466237**	**445668**	**461096**	**437134**	**454415**	**441682**	**452745**	**452385**	**457967**

obtida pela *baseline*.

Com o objetivo de comparar os valores de cobertura (percentual de medidores conectados, que corresponde ao valor complementar dos dados da Figura 37 necessários para totalizar 100%) obtidos com os experimentos realizados com 40 *features* (TopN40, REF1, REF2, REF3 e suas variações com acréscimo de 15% de CPs), foi utilizado o Teste de Friedman (teste estatístico não-paramétrico de Friedman, (FRIEDMAN, 1937)).

Entre os resultados do Teste de Friedman, é de interesse para este estudo a análise do valor obtido para *p-value* (valor-p) que é uma medida estatística que ajuda a avaliar a evidência contra uma hipótese nula. Assumindo que tal hipótese nula corresponde à não evidenciação de diferença estatisticamente significativa entre os resultados avaliados, pode-se estabelecer que uma hipótese alternativa indica que existe, sim, uma diferença entre os valores. Para este estudo, o nível de significância considerado para a análise de *p-value* foi estabelecido em 0,05. Portanto, valores inferiores a esse limite, indicam a ocorrência de diferença estatisticamente significativa.

O valor de *p-value* obtido com o Teste de Friedman ($p\text{-value} < 2,2 \cdot 10^{-16}$) sugere que existe uma diferença estatística entre os resultados dos experimentos. No entanto, ao comparar com um Diagrama de Diferença Crítica (*Critical Difference Diagram*, (DEMŠAR, 2006)) os valores de cobertura obtidos por cidade (Figura 38), observa-se que não há uma diferença estatisticamente significativa entre os resultados obtidos quando analisados os experimentos com conjuntos originais de CPs entre si e os experimentos com *datasets* acrescidos de 15% no número de posições candidatas entre si. Nesse tipo de diagrama, os resultados que não apresentam diferenças estatisticamente significativas estão conectados por uma linha horizontal e, quando não há ligação entre os experimentos, isso indica que os resultados dos experimentos são estatisticamente diferentes.

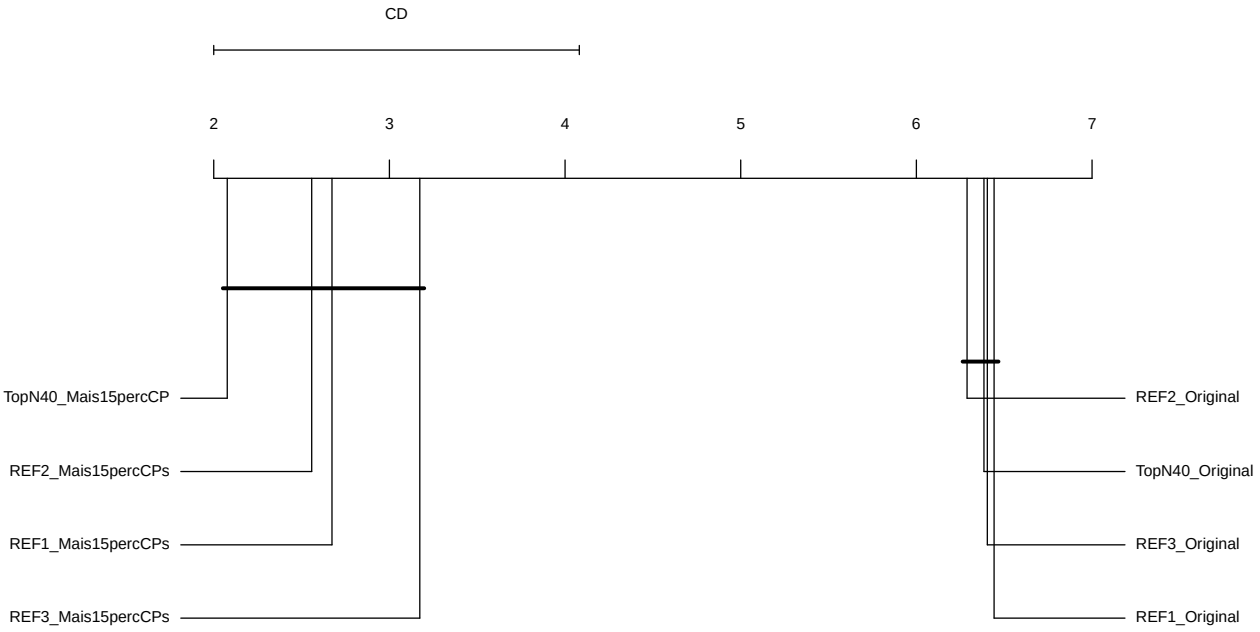


Figura 38 – Diagrama de Diferença Crítica avaliando a cobertura obtida com TopN40 e experimentos realizados com REF1, REF2 e REF3. Nível de significância (α) igual a 0,05.

Fonte: Autoria própria.

Avaliando exclusivamente os resultados de experimentos com *datasets* acrescidos de 15% de CPs, foi obtido como resultado do Teste de Friedman um valor de *p-value* igual a 0,04042, que possibilita dizer que os resultados obtidos pelos experimentos são estatisticamente diferentes. Uma análise mais detalhada, realizada com um teste *post-hoc*, no caso o Nemenyi Test (NEMENYI, 1962), que realiza uma comparação pareada entre os diferentes experimentos, revelou os resultados destacados na Figura 39, que mostram que a diferença mais significativa ocorre na comparação entre os resultados dos experimentos com o *dataset* TopN40 + 15% de CPs e os experimentos com REF3 + 15% de CPs.

Ao analisar o Diagrama de Diferença Crítica para os experimentos com os *datasets*

TopN40, REF1, REF2 e REF3 acrescidos de 15% de CPs (Figura 40), pode-se observar que a configuração TopN40 + 15% de CPs (ou seja, a *baseline*) se destaca das demais, apresentando uma diferença estatisticamente significativa em relação aos resultados obtidos com REF3 + 15% de CPs (visto que não há nenhuma linha conectando os resultados desses dois experimentos). Em relação aos demais experimentos, no entanto, a diferença não pode ser dita como estatisticamente significativa, porém a posição de TopN40 + 15% de CPs no gráfico (primeira posição mais à esquerda do gráfico) qualifica esse experimento como o que apresenta melhores resultados.

Com base nos resultados obtidos com os experimentos adicionais realizados nesta seção, pode-se considerar que a escolha de algoritmo de aprendizagem (no caso, o XGBoost), bem como a configuração de hiperparâmetros utilizada nos experimentos principais deste estudo (em especial para o *dataset* "TopN40 + 15% CPs"), estão devidamente estabelecidas, em especial por viabilizarem o alcance de um maior valor de cobertura

```
        Pairwise comparisons using Nemenyi-Wilcoxon-Wilcox all-pairs test for a two-way
 balanced complete block design

data: y, groups and blocks

                    REF1_Mais15percCPs REF2_Mais15percCPs REF3_Mais15percCPs
REF2_Mais15percCPs  0.950              -                  -
REF3_Mais15percCPs  0.768              0.435              -
TopN40_Mais15percCP 0.314              0.639              0.036
```

Figura 39 – Resultados do teste *post-hoc* Nemenyi Test, que faz uma análise pareada dos resultados obtidos pelos experimentos realizados com os *datasets* TopN40, REF1, REF2 e REF3 acrescidos de 15% de CPs.

Fonte: Autoria própria.

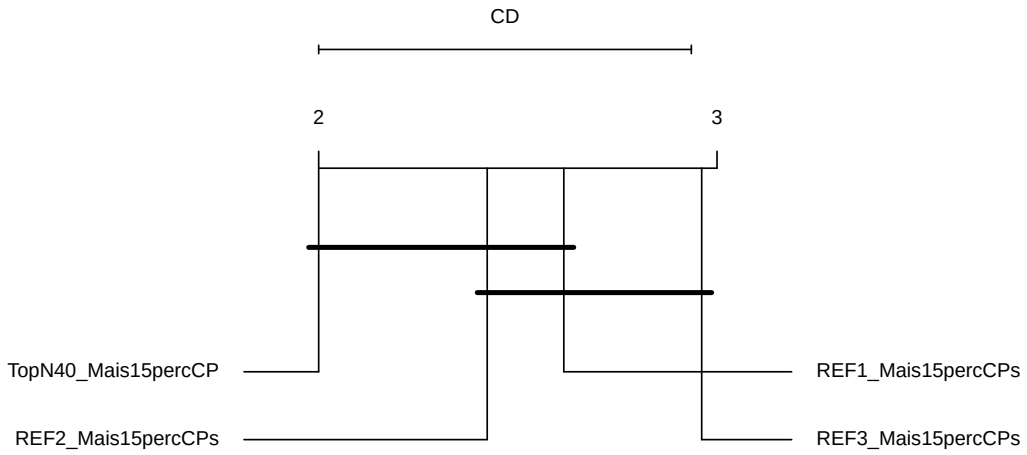


Figura 40 – Diagrama de Diferença Crítica avaliando a cobertura obtida com os *datasets* TopN40, REF1, REF2 e REF3 acrescidos de 15% de CPs. Nível de significância (α) igual a 0,05.

Fonte: Autoria própria.

médio em comparação com os resultados obtidos após HPO. Com essa configuração, é possível observar, de uma forma geral, resultados satisfatórios (com cobertura média dentro dos parâmetros estabelecidos) para o conjunto de cidades avaliadas, tanto para aquelas com maiores quantidades de medidores, como para as demais.

6.8 CONSIDERAÇÕES FINAIS

Os resultados obtidos com os experimentos realizados com AIDA-ML em diferentes conjuntos de *features* demonstram que, com o uso de uma abordagem baseada em *machine learning*, é possível alcançar desempenho compatível com AIDA analítico no que se refere, em especial, ao percentual de medidores conectados.

Isso é demonstrado com os testes efetuados com *datasets* de 40 *features* (em suas variações) com os quais foi possível obter percentuais médios de medidores não conectados inferiores a 2%, que corresponde ao valor de referência considerado pelo método analítico como *stopping criteria*. Entretanto, é importante destacar que os percentuais de medidores não conectados obtidos com o método analítico foram sempre inferiores aos de AIDA-ML.

Em relação ao tempo de processamento, por sua vez, o método AIDA-ML consegue se destacar, fazendo a seleção de posições candidatas em tempo que corresponde, em média, a 12,40% do tempo utilizado pelo método analítico, se considerados os valores obtidos no processamento das 26 cidades do experimento; ou seja, uma redução de 87,60%, em média, no tempo de processamento em comparação ao que seria demandado por AIDA analítico. O ganho com o uso de AIDA-ML também fica evidente em relação ao número de cálculos de LRP necessários, visto que com o método de ML tem-se um ganho médio de 96,86% em relação a esse aspecto.

7 CONCLUSÃO

Nesta seção são analisadas as contribuições da pesquisa e as confrontamos com os requisitos iniciais pré-estabelecidos. De forma geral, observa-se que as contribuições atenderam às expectativas científicas e técnicas. Por tratar-se de uma problemática complexa, a presente tese certamente não aborda todas as nuances e condições relacionadas ao posicionamento de dispositivos de redes de comunicação sem fio projetadas como suporte de automação para *smart grids*; portanto, concluímos a seção com oportunidades de pesquisas (trabalhos futuros) que poderiam evoluir para outras contribuições científicas relacionadas ao tema desta tese.

O problema de posicionamento de roteadores e *gateways* é tratado na literatura, na maior parte dos casos, com o uso de abordagens heurísticas ou como um problema de otimização numérica.

Neste estudo, dois métodos inovadores foram apresentados: (i) AIDA (analítico), que corresponde a uma abordagem heurística aplicada para problemas de larga-escala, que faz uso de uma análise detalhada de perfil de terreno existente para o cálculo de perdas no canal de comunicação, que minimiza a necessidade de análises a serem efetuadas entre medidores e posições candidatas pelo uso de uma MST, e que explora duas abordagens de clusterização para a conexão entre medidores, roteadores e *gateways*; e (ii) AIDA-ML, que corresponde a um método baseado em técnicas de *machine learning* para efetuar o posicionamento de roteadores/*gateways*, com o propósito de ser uma abordagem tão eficiente quanto à abordagem analítica, e que é capaz de apresentar ganhos nos resultados finais, em especial em relação ao tempo de processamento e ao volume de cálculos efetuados.

Para viabilizar a abordagem de *machine learning* proposta pelo estudo, os resultados obtidos com o método analítico serviram como base para treinamento de dados. Isso transfere ao método ML proposto uma característica inovadora, para a qual não foi identificada estratégia similar na literatura recente consultada.

Os experimentos demonstram que o método analítico, com seu modelo de cálculo de perdas, que faz uma análise detalhada de perfil, apresenta uma abordagem pouco explorada na literatura e é capaz de obter resultados melhores que métodos com uso de um modelo geral de cálculo de potência recebida.

Em relação ao método AIDA-ML, os experimentos demonstraram que ele é capaz de obter resultados comparáveis (dentro dos limites estabelecidos) aos do método analítico, com ganho importante, em especial, em relação ao tempo de processamento. A capacidade do método AIDA-ML de conseguir selecionar posições candidatas num tempo que equivale, em média, a 12,40% do tempo exigido pelo método analítico (ou seja, uma redução de

87,60%), sugere grande capacidade de ganho computacional ao que seria necessário para efetuar o processamento de centenas de cidades, com diferentes características geográficas e diferentes quantidades de elementos (postes e medidores inteligentes) envolvidos. O ganho também poderia ser percebido de forma expressiva em relação ao total de cálculos de LRP efetuados, uma vez que AIDA-ML foi capaz de realizar uma quantidade de cálculos 96,86% inferior ao demandado por AIDA.

Em contraste com o apresentado pelos autores em (HE et al., 2017a), que propõem o uso combinado de método heurístico e *machine learning* para o posicionamento de controladores em redes SDN baseado na intensidade de tráfego em cada nó da rede e usam técnicas de *machine learning* como método de otimização, na presente tese a técnica de *machine learning* é implementada em AIDA-ML para o planejamento de AMI, ou seja para todo o processo de definição da topologia da rede, aprendendo com os resultados de outro sistema (no caso, o método analítico AIDA). Além disso, confrontando os resultados obtidos com esta pesquisa em relação aos trabalhos da literatura consultada e apresentada na Seção 3, pode-se dizer que o presente trabalho apresenta uma abordagem inovadora para o posicionamento de roteadores/gateways pelo uso de estratégias de *machine learning* e de extração de *features* em cenário de *smart grid* (incluindo, como resultado, a criação de uma base de dados de treinamento obtida em cenários reais), enquanto as abordagens usuais se baseiam em clusterização e métodos essencialmente heurísticos ou de programação por restrições.

Por fim, a questão de pesquisa estabelecida para este estudo procurava avaliar se "*Dado um conjunto de posições candidatas para o posicionamento de gateways, e obtendo características do cenário (medidores, postes, equipamentos de automação) e topografia em seu entorno, é possível utilizar técnicas de machine learning para determinar as posições para instalação de roteadores e gateways de forma a assegurar conectividade e desempenho em redes de comunicação de smart grids?*". Com base na análise dos resultados, hipóteses e objetivos, tem-se a percepção que sim, é possível utilizar técnicas de *machine learning* como uma alternativa às técnicas tradicionais de posicionamento.

7.1 ANÁLISE DE OBJETIVOS E HIPÓTESES

Em relação aos objetivos de pesquisa apresentados na Seção 1.2 pode-se dizer que o **objetivo** de "*Avaliar estratégias de machine learning existentes, aplicáveis ao planejamento de redes wireless, e propor um método capaz de recomendar posicionamento que assegure desempenho geral da rede dentro de parâmetros pré-estabelecidos, indicados pela indústria e pelo operador da rede*" foi **atingido** com a proposição do método AIDA-ML descrito neste documento, capaz de obter resultados comparáveis aos obtidos com o método analítico, com a vantagem de consumir um tempo de processamento expressivamente menor, demandando

apenas 12,40%, em média, do tempo exigido pelo método AIDA analítico.

Em relação ao **objetivo** de "*Desenvolver um método analítico de posicionamento que permita rotular posições candidatas a fim de viabilizar a construção de uma base de dados de treinamento que possa ser utilizada pelos algoritmos de machine learning para geração do modelo final de posicionamento*", os resultados obtidos com a abordagem de *machine learning* demonstram que o uso de bases de dados de treinamento geradas a partir de resultados do método AIDA analítico é capaz de produzir modelos de aprendizagem com características suficientes para a discriminação das posições candidatas submetidas para o processo de classificação, qualificando-as dentro dos parâmetros estabelecidos e fazendo a seleção de posições candidatas em quantidades e posições que asseguram conectividade e qualidade de enlace capazes de obter o nível de cobertura esperado. Com isso, considera-se que esse objetivo também foi **atingido**.

Quanto ao **objetivo** de "*Analisar técnicas de extração de features a partir de objetos existentes no entorno de posições candidatas e estabelecer conjunto de características suficientes para o uso de algoritmos de machine learning para posicionamento de roteadores/gateways*" pode-se considerar que ele foi **atingido**, uma vez que identificou-se um conjunto de características (40 *features*) suficientemente simples e robusto que permitiu aos métodos de aprendizagem o alcance de resultados do posicionamento de roteadores/*gateways* dentro dos limites de referência estabelecidos para os experimentos.

Em relação às hipóteses apresentadas na Seção 1.3 observa-se que a **hipótese nula H01**, que estabelece que "*Não é possível estabelecer posições de roteadores/gateways em redes wireless a partir de características de elementos do cenário existente no entorno de posições candidatas*", foi **refutada** com os estudos e experimentos realizados, visto que o uso de um método de ML para o posicionamento baseado em características de elementos do cenário e suas relações é capaz de efetuar o posicionamento de roteadores/*gateways* dentro dos parâmetros estabelecidos. Os parâmetros definidos para o método estabelecem a observância da quantidade máxima de medidores não conectados $P_u^{max} = 2\%$ e os resultados com AIDA-ML mostram percentuais abaixo desse limite uma vez que, em experimentos com a configuração "$TopN40 + 15\%CPs$", o percentual médio de medidores não conectados totalizou 1,041%. Além disso, para a mesma configuração, o valor médio de LRP calculado foi de -69,099 dBm, que é bem superior à exigência de buscar uma solução que garanta $LRP \geq -95$ dBm nas conexões entre os dispositivos de comunicação. Com isso, é possível afirmar que a **hipótese alternativa, HA1, é válida**.

Quanto à **hipótese nula H02**, que estabelece que "*Não é possível atingir qualidade das conexões da rede em parâmetros aceitáveis usando técnicas de machine learning para posicionamento de roteadores/gateways*", destaca-se que ela foi refutada pelos estudos e experimentos realizados, visto que o uso do método AIDA-ML foi capaz de posicionar os dispositivos de comunicação de forma a assegurar conectividade entre medidores inteligentes

e posições de roteadores/*gateways* com garantia de potência média recebida estimada no enlace (LRP) superior ao mínimo estabelecido ($LRP \geq -95$ dBm), com percentual de cobertura dentro dos limites também considerados pelo método analítico e respeitando o número máximo de saltos definido como limite ($h_{max} = 7$). Dessa forma, estabelece-se que a **hipótese alternativa, HA2, é válida**.

Finalmente, em relação à **hipótese nula H03**, que determina que "*O estabelecimento de posições candidatas com o uso de algoritmos de ML não diminui o espaço de conexões entre medidores e postes a serem avaliadas em comparação ao utilizado por um método analítico*", é possível refutá-la com os estudos e experimentos realizados, visto que o uso do método AIDA-ML possibilita tempo de processamento expressivamente menor que o demandado pelo método analítico (87,60% menor que o exigido por AIDA) por reduzir o cômputo de LRP entre medidores e posições de postes selecionados da região em análise, dependendo apenas da geração de características para um conjunto mínimo de *features* (da ordem de 40 *features*) para cada posição candidata analisada. Ao efetuar o posicionamento de roteadores/*gateways* para a cidade com maior número de medidores do experimento (Ponta Grossa, com 150.951 medidores), por exemplo, o método AIDA analítico faz o cálculo de 1.999.370 valores de LRP para as diferentes conexões possíveis entre medidores e outros medidores, e entre medidores e posições candidatas (CPs), considerando a execução de 3 iterações. O método AIDA-ML, por sua vez, precisa efetuar o cálculo de *features* para 1.227 posições candidatas (total para as 3 iterações simuladas). Considerando que a configuração com 40 *features* foi a que apresentou melhores resultados, a quantidade de valores de LRP a serem computados totaliza 1.227 x 9 (visto que das 40 features, apenas 9 demandam cálculo de LRP), que é igual a 11.043 cálculos. Para outra cidade com menos medidores (Pien, com 5.303 medidores), o método AIDA analítico executa a análise de LRP para 24.428 conexões, enquanto o método AIDA-ML efetua o cálculo para 2.133 atributos de LRP para o *dataset* de classificação com 40 *features*. A diferença na quantidade de conexões avaliadas por AIDA e por AIDA-ML tende a ser expressiva na maioria dos casos; exceções, no entanto, podem ocorrer em regiões com baixa quantidade de medidores e que demandem grande quantidade de CPs a serem avaliadas devido à extensão geográfica e baixa densidade de medidores por km^2. Por exemplo, na cidade de Porto Amazonas, com 2.375 medidores e 2,5 medidores por km^2, são efetuados 5.417 cálculos de LRP por AIDA contra 3.303 por AIDA-ML. Considerando apenas os cálculos de LRP demandados por AIDA-ML em comparação aos efetuados por AIDA, observa-se (considerando as 26 cidades dos experimentos) um ganho médio de 96,86% para AIDA-ML. Levando em consideração que a redução de conexões avaliadas é expressiva para a maior parte dos casos avaliados, estabelece-se que a **hipótese alternativa, HA3, é válida**.

7.2 TRABALHOS FUTUROS

Neste estudo foram apresentados dois métodos para o posicionamento de roteadores/*gateways* em redes de comunicação de *smart grids*, sendo um denominado de AIDA, que usa uma abordagem analítica para o posicionamento, e outro que usa uma abordagem baseada em *machine learning* e é denominado de AIDA-ML. Ambos os métodos fazem a seleção de posições candidatas (postes) para a instalação de equipamentos de comunicação levando em consideração as posições de postes e medidores inteligentes existentes na região em análise.

Em relação ao método AIDA (analítico), em especial no processo de seleção de posições candidatas com o uso de *grid*, sugere-se avaliar a priorização de postes instalados em posições mais altas, que tendem a favorecer melhor qualidade na transmissão/recepção de sinal. Além disso, em relação à otimização do número de posições candidatas selecionadas (que reflete no total de equipamentos a serem instalados computado pela solução), pode-se avaliar ajustes nas estratégias de *clustering* (abordagens BU e TD) de forma a priorizar as análises de comunicação via múltiplo salto antes de dar prioridade ao uso das posições candidatas em suas capacidades máximas.

Deve-se avaliar, também, para o método analítico, a viabilidade de sua adaptação para suportar a seleção de diferentes modelos de propagação de sinal, utilizados como base para o cálculo dos valores de potência recebidos nos enlaces entre medidores e dispositivos de comunicação.

É importante destacar que, ao adaptar o método analítico para o uso de diferentes modelos de propagação, é essencial o retreinamento de modelo do método AIDA-ML, visto que ele é altamente dependente das configurações utilizadas para o processamento do método analítico.

Outras possibilidades de estudo para trabalho futuro incluem: i) O uso dos métodos AIDA e AIDA-ML em projetos de expansão da rede. Para isso, um estudo de otimização multiestágio (PRÉKOPA, 1995) pode ser uma base para a proposição de estratégia para a modelagem do problema; ii) A generalização dos métodos para os cenários de *smart cities* ou de *multiutilities* visto que uma infraestrutura de comunicação como a adotada para o contexto de *smart grids* pode servir, também, como referência para o estudo de sistemas de fornecimento de gás encanado e água, por exemplo. iii) Avaliar a viabilidade de se criar um método híbrido que possa tomar a decisão entre usar o método AIDA (analítico) ou o método AIDA-ML de acordo com as características da cidade; essa avaliação é motivada pela percepção obtida com a análise do mapa de calor da Figura 37, que apresenta melhores resultados de AIDA-ML para cidades com número de medidores no intervalo central (cidades com número de medidores entre 6.106 e 23.027). Além disso, outras características das cidades devem ser avaliadas, para verificar de que forma a

concentração de medidores e as dimensões das cidades, entre outras informações, podem impactar na seleção do melhor método.

Referências

AALAMIFAR, Fariba; SHIRAZI, Ghasem Naddafzadeh; NOORI, Moslem; LAMPE, Lutz. Cost-efficient data aggregation point placement for advanced metering infrastructure. In: *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*. [S.l.: s.n.], 2014. p. 344–349. Citado 2 vezes nas páginas 32 e 94.

AGUIAR, Alan de. *XGBoost - A Matemática passo a passo*. 2020. <https://medium.com//@aln.deaguiar/xgboost-a-matem%C3%A1tica-passo-a-passo-29d34fa561dc>. "On-line; acessado em 08/03/2023". Citado na página 59.

AL-SAMAWI, Mazen Abdualmajed Ali; SINGH, Manwinder. Effect of 5G on IoT and daily life application. In: *2022 3rd International Conference for Emerging Technology (INCET)*. [S.l.: s.n.], 2022. p. 1–5. Citado na página 71.

ALEXANDER, Roger; BRANDT, Anders; VASSEUR, JP; HUI, Jonathan; PISTER, Kris; THUBERT, Pascal; LEVIS, P; STRUIK, Rene; KELSEY, Richard; WINTER, Tim. *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*. RFC Editor, 2012. RFC 6550. (Request for Comments, 6550). Disponível em: <https://www.rfc-editor.org/info/rfc6550>. Citado na página 41.

ALI, AWADALLAH MOHAMMED AHMED. *Optimizing Gateway Placement in Wireless Mesh Network using Genetic Algorithm and Simulated Annealing*. Tese (Doutorado) — College of Computer Science and Information Technology. Sudan University of Science & Technology, January 2016. Citado 4 vezes nas páginas 84, 90, 91 e 93.

AMIN, S. Massoud; WOLLENBERG, B.F. Toward a smart grid: power delivery for the 21st century. *IEEE Power and Energy Magazine*, v. 3, n. 5, p. 34–41, 2005. Citado na página 40.

AOUN, B.; BOUTABA, R.; IRAQI, Y.; KENWARD, G. Gateway placement optimization in wireless mesh networks with QoS constraints. *IEEE Journal on Selected Areas in Communications*, v. 24, n. 11, p. 2127–2136, 2006. Citado 6 vezes nas páginas 32, 77, 79, 90, 91 e 93.

ASSUNÇÃO, Gustavo; PATRãO, Bruno; CASTELO-BRANCO, Miguel; MENEZES, Paulo. An overview of emotion in artificial intelligence. *IEEE Transactions on Artificial Intelligence*, v. 3, n. 6, p. 867–886, 2022. Citado na página 31.

BARTOSIK, Aleksandra; WHITTINGHAM, Hannes. Chapter 7 - evaluating safety and toxicity. In: ASHENDEN, Stephanie Kay (Ed.). *The Era of Artificial Intelligence, Machine Learning, and Data Science in the Pharmaceutical Industry*. Academic Press, 2021. p. 119–137. ISBN 978-0-12-820045-2. Disponível em: <https://www.sciencedirect.com/science/article/pii/B9780128200452000088>. Citado na página 54.

BELLMAN, R. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961. (Princeton Legacy Library). ISBN 9780691079011. Disponível em: <https://books.google.com.br/books?id=POAmAAAAMAAJ>. Citado na página 61.

BELLMAN, Richard; KALABA, Robert. On adaptive control processes. *IRE Transactions on Automatic Control*, IEEE, v. 4, n. 2, p. 1–9, 1959. Citado na página 61.

BELTRAN, C.; TADONKI, Claude; VIAL, Jean-Philippe. Solving the p-median problem with a semi-lagrangian relaxation. *Computational Optimization and Applications*, v. 35, p. 239–260, 01 2006. Citado na página 48.

BERGSTRA, James; BENGIO, Yoshua. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, JMLR.org, v. 13, n. null, p. 281–305, feb 2012. ISSN 1532-4435. Citado na página 65.

BLUM, Avrim L.; LANGLEY, Pat. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, v. 97, n. 1, p. 245–271, 1997. ISSN 0004-3702. Relevance. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0004370297000635>. Citado na página 61.

BOONKAJAY, Amnart; TAN, Peng Hui; GOH, Lee Kee; AHMED, Syed Naveen Altaf; SUN, Sumei. Enhancing Wi-SUN AMI network resilience by using emergency gateway with optimal placement. In: *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*. [S.l.: s.n.], 2021. p. 1–5. Citado 3 vezes nas páginas 87, 91 e 93.

BREIMAN, Leo. Random Forests. *Mach. Learn.*, Kluwer Academic Publishers, USA, v. 45, n. 1, p. 5–32, oct 2001. ISSN 0885-6125. Disponível em: <https://doi.org/10.1023/A:1010933404324>. Citado na página 56.

BROWNLEE, Jason. *LOOCV for Evaluating Machine Learning Algorithms*. 2020. Https://machinelearningmastery.com/loocv-for-evaluating-machine-learning-algorithms. Accessed on Aug 9, 2022. Disponível em: <https://machinelearningmastery.com/loocv-for-evaluating-machine-learning-algorithms>. Citado na página 132.

BROWNLEE, Jason. *ROC Curves and Precision-Recall Curves for Imbalanced Classification*. 2020. Https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-imbalanced-classification/. Acessado em 09/08/2022. Disponível em: <https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-imbalanced-classification/>. Citado na página 134.

CECATI, Carlo; MOKRYANI, Geev; PICCOLO, Antonio; SIANO, Pierluigi. An overview on the smart grid concept. In: *IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society*. [S.l.: s.n.], 2010. p. 3322–3327. Citado na página 40.

CHAUDHRY, Aizaz U.; RAITHATHA, Mital; HAFEZ, Roshdy H.M.; CHINNECK, John W. Using machine learning to locate gateways in the wireless backhaul of 5G ultra-dense networks. In: *2020 International Symposium on Networks, Computers and Communications (ISNCC)*. [S.l.: s.n.], 2020. p. 1–5. Citado 4 vezes nas páginas 81, 90, 91 e 93.

CHEN, Tianqi; GUESTRIN, Carlos. XGBoost: A scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2016. (KDD '16), p. 785–794. ISBN 9781450342322. Disponível em: <https://doi.org/10.1145/2939672.2939785>. Citado 3 vezes nas páginas 57, 58 e 138.

CORNEJO, Andres; LANDEROS-AYALA, Salvador; MATIAS, Jose M.; MARTINEZ, Ramon. Applying learning methods to optimize the ground segment for HTS systems. In: *2020 IEEE 11th Latin American Symposium on Circuits Systems (LASCAS)*. [S.l.: s.n.], 2020. p. 1–4. Citado 5 vezes nas páginas 82, 91, 92, 93 e 95.

COX, David R. The regression analysis of binary sequences. *Journal of the Royal Statistical Society*, XX, n. 2, p. 215–242, 1958. Citado na página 54.

CUTLER, Adele; CUTLER, David; STEVENS, John. Random Forests. In: _____. [S.l.: s.n.], 2011. v. 45, p. 157–176. ISBN 978-1-4419-9325-0. Citado 3 vezes nas páginas 56, 57 e 58.

DAVIS, Jesse; GOADRICH, Mark. The relationship between Precision-Recall and ROC curves. In: *Proceedings of the 23rd International Conference on Machine Learning*. New York, NY, USA: Association for Computing Machinery, 2006. (ICML '06), p. 233–240. ISBN 1595933832. Disponível em: <https://doi.org/10.1145/1143844.1143874>. Citado 2 vezes nas páginas 132 e 134.

DELIGIANNIS, Paraskevas; KOUTROUBINAS, Stelios; KORONIAS, George. Predicting energy consumption through machine learning using a smart-metering architecture. *IEEE Potentials*, v. 38, n. 2, p. 29–34, 2019. Citado na página 31.

DEMŠAR, Janez. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine learning research*, v. 7, n. 1, p. 1–30, 2006. Citado na página 159.

DHARMADHIKARI, S.C.; GAMPALA, Veerraju; RAO, Ch. Mallikarjuna; KHASIM, Syed; JAIN, Shafali; BHASKARAN, R. A smart grid incorporated with ML and IoT for a secure management system. *Microprocessors and Microsystems*, v. 83, p. 103954, 2021. ISSN 0141-9331. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0141933121001332>. Citado 2 vezes nas páginas 71 e 95.

ERCEG, V.; GREENSTEIN, L.J.; TJANDRA, S.Y.; PARKOFF, S.R.; GUPTA, A.; KULIC, B.; JULIUS, A.A.; BIANCHI, R. An empirically based path loss model for wireless channels in suburban environments. *IEEE Journal on Selected Areas in Communications*, v. 17, n. 7, p. 1205–1211, 1999. Citado na página 97.

ERCEG, V.; HARI, K.V.S.; SMITH, M.; BAUM, D.s; SHEIKH, Kalsoom; TAPPENDEN, C.; COSTA, J.; BUSHUE, C.; SARAJEDINI, A.; SCHWARTZ, R.; BRANLUND, D. Channel models for fixed wireless application. *IEEE 802.16 Broadband Wireless Access Working Group, Tech Rep*, 01 2001. Citado na página 97.

FATEH, Benazir; GOVINDARASU, Manimaran; AJJARAPU, Venkataramana. Wireless network design for transmission line monitoring in smart grid. *IEEE Transactions on Smart Grid*, v. 4, n. 2, p. 1076–1086, 2013. Citado 3 vezes nas páginas 86, 91 e 93.

FAZLI, Siamac; GROZEA, Cristian; DANóCZY, Márton; BLANKERTZ, Benjamin; POPESCU, Florin; MüLLER, Klaus-Robert. Subject independent EEG-based BCI decoding. In: . [S.l.: s.n.], 2009. v. 22, p. 513–521. Citado na página 132.

FERREIRA, Luís; PILASTRI, André; MARTINS, Carlos Manuel; PIRES, Pedro Miguel; CORTEZ, Paulo. A comparison of AutoML tools for machine learning, deep learning and

XGBoost. In: *2021 International Joint Conference on Neural Networks (IJCNN)*. [S.l.: s.n.], 2021. p. 1–8. Citado na página 65.

FERREIRA, Marcus; SOUZA, Gustavo; CASTRO, Marcelo; ARAùJO, Sérgio; VIEIRA, Flávio Henrique; BORGES, Vinicius; CARDOSO, Alisson. Posicionamento de concentradores para uma infraestrutura avançada de medição inteligente em redes máquina a máquina. In: *XXXIII Simpósio Brasileiro de Telecomunicações (SBrT2015)*. [S.l.: s.n.], 2015. p. 1–5. Citado 4 vezes nas páginas 77, 90, 91 e 93.

FEURER, Matthias; EGGENSPERGER, Katharina; FALKNER, Stefan; LINDAUER, Marius; HUTTER, Frank. Auto-sklearn 2.0: Hands-free AutoML via meta-learning. *arXiv:2007.04074 [cs.LG]*, 2020. Citado 2 vezes nas páginas 66 e 136.

FEURER, Matthias; KLEIN, Aaron; EGGENSPERGER, Katharina; SPRINGENBERG, Jost; BLUM, Manuel; HUTTER, Frank. Efficient and robust automated machine learning. In: *Advances in Neural Information Processing Systems 28 (2015)*. [S.l.: s.n.], 2015. p. 2962–2970. Citado 3 vezes nas páginas 66, 67 e 136.

FRIEDMAN, Milton. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, Taylor & Francis, v. 32, n. 200, p. 675–701, 1937. Citado na página 158.

GALLARDO, José Luis; AHMED, Mohamed A.; JARA, Nicolás. Clustering algorithm-based network planning for advanced metering infrastructure in smart grid. *IEEE Access*, v. 9, p. 48992–49006, 2021. Citado 5 vezes nas páginas 75, 90, 91, 93 e 94.

GHOLAMIANGONABADI, Davoud; KISELOV, Nikita; GROLINGER, Katarina. Deep neural networks for human activity recognition with wearable sensors: Leave-one-subject-out cross-validation for model selection. *IEEE Access*, v. 8, p. 133982–133994, 2020. Citado na página 132.

GOUTTE, Cyril; GAUSSIER, Eric. A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In: . [S.l.: s.n.], 2005. v. 3408, p. 345–359. ISBN 978-3-540-25295-5. Citado na página 134.

GUDIVADA, V.N.; IRFAN, M.T.; FATHI, E.; RAO, D.L. Chapter 5 - cognitive analytics: Going beyond big data analytics and machine learning. In: GUDIVADA, Venkat N.; RAGHAVAN, Vijay V.; GOVINDARAJU, Venu; RAO, C.R. (Ed.). *Cognitive Computing: Theory and Applications*. Elsevier, 2016, (Handbook of Statistics, v. 35). p. 169–205. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0169716116300517>. Citado 2 vezes nas páginas 55 e 56.

GUYON, Isabelle; ELISSEEFF, André. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, JMLR.org, v. 3, n. null, p. 1157–1182, mar 2003. ISSN 1532-4435. Citado 3 vezes nas páginas 61, 62 e 64.

HAJDU, László; DáVID, Balázs; KRéSZ, Miklós. Gateway placement and traffic load simulation in sensor networks. *Pollack Periodica*, Akadémiai Kiadó, Budapest, Hungary, v. 16, n. 1, p. 102 – 108, 2021. Disponível em: <https://akjournals.com/view/journals/606/16/1/article-p102.xml>. Citado 3 vezes nas páginas 83, 90 e 93.

HAKIMI, S.L. Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations Research*, v. 12 (3), p. 450–459, 06 1964. Citado na página 46.

HAKIMI, S.L. Optimum distribution of switching centers in a communication network and some related graph theoretic problems. *Operations Research*, v. 13(3), p. 462–475, 06 1965. Citado na página 46.

HE, Mu; BASTA, Arsany; BLENK, Andreas; KELLERER, Wolfgang. Modeling flow setup time for controller placement in SDN: Evaluation for dynamic flows. In: *2017 IEEE International Conference on Communications (ICC)*. [S.l.: s.n.], 2017. p. 1–7. Citado 4 vezes nas páginas 88, 91, 93 e 164.

HE, Mu; KALMBACH, Patrick; BLENK, Andreas; KELLERER, Wolfgang; SCHMID, Stefan. Algorithm-data driven optimization of adaptive communication networks. In: *2017 IEEE 25th International Conference on Network Protocols (ICNP)*. [S.l.: s.n.], 2017. p. 1–6. Citado 5 vezes nas páginas 80, 91, 92, 93 e 95.

HELLER, Brandon; SHERWOOD, Rob; MCKEOWN, Nick. The controller placement problem. In: *Proceedings of the First Workshop on Hot Topics in Software Defined Networks*. New York, NY, USA: Association for Computing Machinery, 2012. (HotSDN '12), p. 7–12. ISBN 9781450314770. Disponível em: <https://doi.org/10.1145/2342441.2342444>. Citado 4 vezes nas páginas 32, 87, 92 e 93.

HORIHATA, Kenshi; KANAI, Kazuki; HASEGAWA, Rei; KOYANAGI, Yoshio; ICHIKAWA, Yasufumi. A study of machine learning using wireless and physical environment data at a factory. In: *2020 IEEE International Symposium on Antennas and Propagation and North American Radio Science Meeting*. [S.l.: s.n.], 2020. p. 1099–1100. Citado na página 71.

HUANG, Changwu; ZHANG, Zeqi; MAO, Bifei; YAO, Xin. An overview of artificial intelligence ethics. *IEEE Transactions on Artificial Intelligence*, p. 1–21, 2022. Citado na página 31.

HUTTER, Frank; HOOS, Holger H.; LEYTON-BROWN, Kevin. Sequential model-based optimization for general algorithm configuration. In: COELLO, Carlos A. Coello (Ed.). *Learning and Intelligent Optimization*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. p. 507–523. ISBN 978-3-642-25566-3. Citado na página 67.

IEEE SA - Standards Association. *IEEE 802.15.4g-2012. IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 3: Physical Layer (PHY) Specifications for Low-Data-Rate, Wireless, Smart Metering Utility Networks*. 2012. Https://standards.ieee.org/ieee/802.15.4g/5053/. Accessed on May 8, 2022. Disponível em: <https://standards.ieee.org/ieee/802.15.4g/5053/>. Citado na página 41.

INDARJO, Pararawendy. *Using Weighted K-Means Clustering to Determine Distribution Centres Locations. Another use case of a modified version of K-Means algorithm you might not know*. 2020. Https://towardsdatascience.com/using-weighted-k-means-clustering-to-determine-distribution-centres-locations-2567646fc31d. Accessed: 2021-12-13. Citado na página 109.

INGA, Esteban; CAMPAñA, Miguel; HINCAPIé, Roberto; CéSPEDES, Sandra. Optimal placement of data aggregation points for smart metering using wireless heterogeneous networks. In: *2018 IEEE Colombian Conference on Communications and Computing (COLCOM)*. [S.l.: s.n.], 2018. p. 1–6. Citado na página 94.

International Telecommunication Union. Propagation by diffraction. In: *Recommendation ITU-R P.526-13, International Telecomunication Union, ITU-R*. Electronic Publication, Geneva, 2013. P Series, Radiowave propagation, p. 1–41. Disponível em: <https://www.itu.int/dms_pubrec/itu-r/rec/p/R-REC-P.526-13-201311-S!!PDF-E.pdf>. Citado 2 vezes nas páginas 50 e 99.

ISLAM, Muhammad Nazrul; INAN, Toki Tahmid; RAFI, Suzzana; AKTER, Syeda Sabrina; SARKER, Iqbal H.; ISLAM, A. K. M. Najmul. A systematic review on the use of AI and ML for fighting the COVID-19 pandemic. *IEEE Transactions on Artificial Intelligence*, v. 1, n. 3, p. 258–270, 2020. Citado na página 31.

JOHN, George H.; KOHAVI, Ron; PFLEGER, Karl. Irrelevant features and the subset selection problem. In: COHEN, William W.; HIRSH, Haym (Ed.). *Machine Learning Proceedings 1994*. San Francisco (CA): Morgan Kaufmann, 1994. p. 121–129. ISBN 978-1-55860-335-6. Disponível em: <https://www.sciencedirect.com/science/article/pii/B9781558603356500234>. Citado na página 62.

KARGER, David R.; KLEIN, Philip N.; TARJAN, Robert E. A randomized linear-time algorithm to find minimum spanning trees. *J. ACM*, Association for Computing Machinery, New York, NY, USA, v. 42, n. 2, p. 321–328, mar 1995. ISSN 0004-5411. Disponível em: <https://doi.org/10.1145/201019.201022>. Citado na página 52.

KEILWAGEN, Jens; GROSSE, Ivo; GRAU, Jan. Area under precision-recall curves for weighted and unweighted data. *PloS one*, v. 9, p. e92209, 03 2014. Citado na página 132.

KEMAL, Mohammed S.; OLSEN, Rasmus L.; SCHWEFEL, Hans-Peter. Optimized scheduling of smart meter data access for real-time voltage quality monitoring. In: *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*. [S.l.: s.n.], 2018. p. 1–6. Citado na página 45.

KOHAVI, Ron; JOHN, George H. Wrappers for feature subset selection. *Artificial Intelligence*, v. 97, n. 1, p. 273–324, 1997. ISSN 0004-3702. Relevance. Disponível em: <https://www.sciencedirect.com/science/article/pii/S000437029700043X>. Citado 2 vezes nas páginas 62 e 63.

KONG, Peng-Yong. Cost efficient data aggregation point placement with interdependent communication and power networks in smart grid. *IEEE Transactions on Smart Grid*, v. 10, n. 1, p. 74–83, 2019. Citado na página 94.

KRUSKAL, Joseph B. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, American Mathematical Society, v. 7, n. 1, p. 48–50, 1956. ISSN 00029939, 10886826. Disponível em: <https://doi.org/10.2307/2033241>. Citado na página 52.

KUNDACINA, Ognjen; FORCAN, Miodrag; COSOVIC, Mirsad; RACA, Darijo; DZAFE-RAGIC, Merim; MISKOVIC, Dragisa; MAKSIMOVIC, Mirjana; VUKOBRATOVIC, Dejan. Near real-time distributed state estimation via AI/ML-empowered 5G networks.

In: *2022 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. [S.l.: s.n.], 2022. p. 284–289. Citado na página 31.

LANG, Adrian; WANG, Yi; FENG, Cheng; STAI, Eleni; HUG, Gabriela. Data aggregation point placement for smart meters in the smart grid. *IEEE Transactions on Smart Grid*, v. 13, n. 1, p. 541–554, 2022. Citado 2 vezes nas páginas 32 e 94.

LANGE, Stanislav; GEBERT, Steffen; ZINNER, Thomas; TRAN-GIA, Phuoc; HOCK, David; JARSCHEL, Michael; HOFFMANN, Marco. Heuristic approaches to the controller placement problem in large scale SDN networks. *IEEE Transactions on Network and Service Management*, v. 12, n. 1, p. 4–17, 2015. Citado 3 vezes nas páginas 80, 91 e 93.

LEKHTMAN, Alon. *Data Science in Medicine — Precision & Recall or Specificity & Sensitivity?* 2019. Https://towardsdatascience.com/should-i-look-at-precision-recall-or-specificity-sensitivity-3946158aace1. Acessado em 09/08/2022. Disponível em: <https://towardsdatascience.com/should-i-look-at-precision-recall-or-specificity-sensitivity-3946158aace1>. Citado na página 134.

LI, Fan; WANG, Yu; LI, Xiang-Yang; NUSAIRAT, Ashraf; WU, Yanwei. Gateway placement for throughput optimization in wireless mesh networks. *Mobile Networks and Applications*, v. 13, n. 1, p. 198–211, Apr 2008. ISSN 1572-8153. Disponível em: <https://doi.org/10.1007/s11036-008-0034-8>. Citado 5 vezes nas páginas 83, 90, 91, 93 e 102.

LI, Jundong; CHENG, Kewei; WANG, Suhang; MORSTATTER, Fred; TREVINO, Robert P.; TANG, Jiliang; LIU, Huan. Feature selection: A data perspective. *ACM Comput. Surv.*, Association for Computing Machinery, New York, NY, USA, v. 50, n. 6, dec 2017. ISSN 0360-0300. Disponível em: <https://doi.org/10.1145/3136625>. Citado na página 62.

LI, Xiuquan; ZHANG, Tao. An exploration on artificial intelligence application: From security, privacy and ethic perspective. In: *2017 IEEE 2nd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*. [S.l.: s.n.], 2017. p. 416–420. Citado na página 31.

LIU, Qiang; LENG, Supeng; MAO, Yuming; ZHANG, Yan. Optimal gateway placement in the smart grid machine-to-machine networks. In: *2011 IEEE GLOBECOM Workshops (GC Wkshps)*. [S.l.: s.n.], 2011. p. 1173–1177. Citado 4 vezes nas páginas 87, 92, 93 e 94.

LOH, Frank; BAU, Dominique; ZINK, Johannes; WOLFF, Alexander; HOßFELD, Tobias. Robust Gateway Placement for Scalable LoRaWAN. In: *2021 13th IFIP Wireless and Mobile Networking Conference (WMNC)*. [S.l.: s.n.], 2021. p. 71–78. Citado 4 vezes nas páginas 85, 91, 92 e 93.

LORENA, Luiz Antonio Nogueira; SENNE, Edson Luiz França; PAIVA, João Argemiro de Carvalho; PEREIRA, Marcos Antonio. Integração de modelos de localização a sistemas de informações geográficas. *Gestão & Produção*, Universidade Federal de São Carlos, v. 8, n. Gest. Prod., 2001 8(2), p. 180–195, Aug 2001. ISSN 0104-530X. Disponível em: <https://doi.org/10.1590/S0104-530X2001000200006>. Citado na página 47.

MADAMORI, Oluwashina. *Optimal Gateway Placement in Low-cost Smart Cities*. Dissertação (Master's Thesis. First Advisor: Dr. Corey E. Baker) — University of Kentucky, Theses and Dissertations–Computer Science. 92, https://uknowledge.uky.edu/cs_etds/92, 2019. Citado 5 vezes nas páginas 84, 90, 91, 92 e 93.

MAHDY, Amany; KONG, Peng-Yong; ZAHAWI, Bashar; KARAGIANNIDIS, George K. Data aggregate point placement for smart grid with joint consideration of communication and power networks. In: *2017 7th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO)*. [S.l.: s.n.], 2017. p. 1–5. Citado 4 vezes nas páginas 75, 91, 93 e 94.

MAHMOUD, Haitham H. M.; ISMAIL, Tawfik; DARWEESH, M. Saeed. Dynamic traffic model with optimal gateways placement in IP cloud heterogeneous CRAN. *IEEE Access*, v. 8, p. 119062–119070, 2020. Citado 4 vezes nas páginas 81, 90, 91 e 93.

MARCH, William B.; RAM, Parikshit; GRAY, Alexander G. Fast euclidean minimum spanning tree: Algorithm, analysis, and applications. In: *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2010. (KDD '10), p. 603–612. ISBN 9781450300551. Disponível em: <https://doi.org/10.1145/1835804.1835882>. Citado 2 vezes nas páginas 52 e 101.

MARIANOV, Vladimir; SERRA, Daniel. Median problems in networks. *SSRN Electronic Journal*, v. 155, 03 2009. Citado na página 47.

MATNI, Nagib. *Fuzzy C-Means Based Gateway Placement Algorithm for LoRaWAN*. Dissertação (Master Thesis. Advisor: Denis Lima do Rosário, Co-Advisors: Eduardo Coelho Cerqueira) — Programa de Pós-Graduação em Engenharia Elétrica, Instituto de Tecnologia, Universidade Federal do Pará, Belém, Brazil, 2020. Citado 3 vezes nas páginas 81, 91 e 93.

MATNI, Nagib; MORAES, Jean; OLIVEIRA, Helder; ROSáRIO, Denis; CERQUEIRA, Eduardo. LoRaWAN gateway placement model for dynamic internet of things scenarios. *Sensors*, v. 20, n. 15, 2020. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/20/15/4336>. Citado 3 vezes nas páginas 81, 91 e 93.

MIN, Li; ALNOWIBET, Khalid Abdulaziz; ALRASHEEDI, Adel Fahad; MOAZZEN, Farid; AWWAD, Emad Mahrous; MOHAMED, Mohamed A. A stochastic machine learning based approach for observability enhancement of automated smart grids. *Sustainable Cities and Society*, v. 72, p. 103071, 2021. ISSN 2210-6707. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2210670721003553>. Citado na página 71.

MINHAJ, Syed Usama; MAHMOOD, Aamir; ABEDIN, Sarder Fakhrul; HASSAN, Syed Ali; BHATTI, Muhammad Talha; ALI, Syed Haider; GIDLUND, Mikael. Intelligent resource allocation in LoRaWAN using machine learning techniques. *IEEE Access*, v. 11, p. 10092–10106, 2023. Citado na página 31.

MIRZAEE, Parya Haji; SHOJAFAR, Mohammad; POORANIAN, Zahra; ASEFY, Pedram; CRUICKSHANK, Haitham; TAFAZOLLI, Rahim. Fids: A federated intrusion detection system for 5G smart metering network. In: *2021 17th International Conference on Mobility, Sensing and Networking (MSN)*. [S.l.: s.n.], 2021. p. 215–222. Citado na página 31.

MITCHELL, Tom M. *Machine Learning*. [S.l.]: McGraw-Hill Education, 1997. ISBN: 978-0070428072. Citado na página 53.

MOCANU, Elena. *Machine learning applied to smart grids*. Tese (Doutorado) — Eindhoven University of Technology, out. 2017. Citado 2 vezes nas páginas 71 e 95.

MOCHINSKI, Marcos Alberto; BICZKOWSKI, Mauricio; CHUEIRI, Ivan Jorge; JAMHOUR, Edgard; ZAMBENEDETTI, Voldi Costa; PELLENZ, Marcelo Eduardo; ENEMBRECK, Fabrício. Developing an intelligent decision support system for large-scale smart grid communication network planning. *Knowledge-Based Systems*, v. 283, p. 111159, 2024. ISSN 0950-7051. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0950705123009097>. Citado 8 vezes nas páginas 32, 37, 115, 122, 123, 126, 144 e 151.

MOCHINSKI, Marcos Alberto; VIEIRA, Marina Luísa de Souza Carrasco; BICZKOWSKI, Mauricio; CHUEIRI, Ivan Jorge; JAMHOUR, Edgar; ZAMBENEDETTI, Voldi Costa; PELLENZ, Marcelo Eduardo; ENEMBRECK, Fabrício. Towards an efficient method for large-scale Wi-SUN-Enabled AMI network planning. *Sensors*, v. 22, n. 23, 2022. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/22/23/9105>. Citado 15 vezes nas páginas 37, 43, 49, 92, 94, 97, 98, 101, 102, 103, 106, 107, 109, 110 e 111.

MOLNAR, Christoph. *Interpretable Machine Learning: A Guide For Making Black Box Models*. 2022. <https://christophm.github.io/interpretable-ml-book/>. "On-line; acessado em 22/02/2023". Citado 3 vezes nas páginas 54, 55 e 56.

NANDA, P.; KUMAR, Josephine Prem. Gateway placement in mesh network using traffic offloading through 2G/3G networks. In: *Global Journal of Computer Science and Technology*. [S.l.: s.n.], 2016. v. 16, n. 6. Citado 4 vezes nas páginas 85, 91, 92 e 93.

NEMENYI, Peter. Distribution-free multiple comparisons. In: INTERNATIONAL BIOMETRIC SOC 1441 I ST, NW, SUITE 700, WASHINGTON, DC 20005-2210. *Biometrics*. [S.l.], 1962. v. 18, n. 2, p. 263. Citado na página 159.

NYIRENDA, Clement N. On the efficacy of particle swarm optimization for gateway placement in LoRaWAN networks. In: VAKHANIA, Prof. Nodari (Ed.). *Optimization Algorithms*. Rijeka: IntechOpen, 2021. cap. 3. Disponível em: <https://doi.org/10.5772/intechopen.98649>. Citado 3 vezes nas páginas 84, 91 e 93.

OLSON, Randal S.; BARTLEY, Nathan; URBANOWICZ, Ryan J.; MOORE, Jason H. Evaluation of a tree-based pipeline optimization tool for automating data science. In: *Proceedings of the Genetic and Evolutionary Computation Conference 2016*. New York, NY, USA: Association for Computing Machinery, 2016. (GECCO '16), p. 485–492. ISBN 9781450342063. Disponível em: <https://doi.org/10.1145/2908812.2908918>. Citado 3 vezes nas páginas 67, 68 e 136.

OUSAT, Behnam; GHADERI, Majid. LoRa network planning: Gateway placement and device configuration. In: *2019 IEEE International Congress on Internet of Things (ICIOT)*. [S.l.: s.n.], 2019. p. 25–32. Citado 4 vezes nas páginas 83, 90, 91 e 93.

PATIL, Suvarna; GOKHALE, Prasad. Distance aware gateway placement optimization for machine-to-machine (M2M) communication in IoT network. In: *Turkish Journal of Computer and Mathematics Education*. [S.l.: s.n.], 2021. v. 12, n. 2 (2021), p. 1995–2005. Citado 3 vezes nas páginas 81, 91 e 93.

PAULI, Martin; POHL, Constantin; GOLZ, Martin. Balanced leave-one-subject-out cross-validation for microsleep classification. *Current Directions in Biomedical Engineering*, v. 7, p. 147–150, 10 2021. Citado na página 132.

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. Citado 2 vezes nas páginas 65 e 133.

PIRAK, Chaiyod; SANGSUWAN, Tanayoot; TANAKORNPINTONG, Songserm; MATHAR, Rudolf. Channel-aware optimal placement algorithm for data concentrator unit in smart grid systems. In: *2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*. [S.l.: s.n.], 2017. p. 447–450. Citado na página 94.

PONCHA, Lemayian Joel; ABDELHAMID, Sherin; ALTURJMAN, Sinem; EVER, Enver; AL-TURJMAN, Fadi. 5G in a convergent internet of things era: An overview. In: *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*. [S.l.: s.n.], 2018. p. 1–6. Citado na página 71.

POPOOLA, Segun I.; JEFIA, Abigail; ATAYERO, Aderemi A.; KINGSLEY, Ogbeide; FARUK, Nasir; OSENI, Olasunkanmi F.; ABOLADE, Robert O. Determination of neural network parameters for path loss prediction in very high frequency wireless channel. *IEEE Access*, v. 7, p. 150462–150483, 2019. Citado na página 51.

POWERS, David. Evaluation: From precision, recall and f-factor to ROC, informedness, markedness & correlation. *Mach. Learn. Technol.*, v. 2, 01 2008. Citado na página 132.

PRÉKOPA, András. Multi-stage stochastic programming problems. In: _____. *Stochastic Programming*. Dordrecht: Springer Netherlands, 1995. p. 425–446. ISBN 978-94-017-3087-7. Disponível em: <https://doi.org/10.1007/978-94-017-3087-7_13>. Citado na página 167.

PRIM, R. C. Shortest connection networks and some generalizations. *The Bell System Technical Journal*, v. 36, n. 6, p. 1389–1401, 1957. Citado na página 52.

RAITHATHA, Mital; CHAUDHRY, Aizaz U.; HAFEZ, Roshdy H. M.; CHINNECK, John W. A fast heuristic for gateway location in wireless backhaul of 5G ultra-dense networks. *IEEE Access*, v. 9, p. 43653–43674, 2021. Citado 4 vezes nas páginas 79, 90, 91 e 93.

RASCHKA, Sebastian. Mlxtend: Providing machine learning and data science utilities and extensions to Python's scientific computing stack. *The Journal of Open Source Software*, The Open Journal, v. 3, n. 24, apr 2018. Disponível em: <https://joss.theoj.org/papers/10.21105/joss.00638>. Citado na página 136.

ROLIM, Guilherme; PASSOS, Diego; ALBUQUERQUE, Célio; MORAES, Igor. MOSKOU: A heuristic for data aggregator positioning in smart grids. *IEEE Transactions on Smart Grid*, v. 9, n. 6, p. 6206–6213, 2018. Citado na página 94.

SALEHIN, Imrus; ISLAM, Md.Shamiul; SAHA, Pritom; NOMAN, S.M.; TUNI, Azra; HASAN, Md.Mehedi; BATEN, Md.Abu. Automl: A systematic review on automated machine learning with neural architecture search. *Journal of*

Information and Intelligence, 2023. ISSN 2949-7159. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2949715923000604>. Citado na página 65.

Leave-one-out cross-validation. In: SAMMUT, Claude; WEBB, Geoffrey I. (Ed.). *Encyclopedia of Machine Learning*. Boston, MA: Springer US, 2010. p. 600–601. ISBN 978-0-387-30164-8. Disponível em: <https://doi.org/10.1007/978-0-387-30164-8_469>. Citado na página 132.

SANDI, Younes. *Bootstrapping, Bagging & Boosting*. 2021. <https://www.linkedin.com/pulse/bootstrapping-bagging-boosting-younes-sandi/>. "On-line; acessado em 22/02/2023". Citado 2 vezes nas páginas 58 e 60.

SCARAMELLA, Geovana; HECK, Giancarlo Covolo; JUNIOR, Lourival Lippmann; HEXSEL, Roberto A.; SANTANA, Tiago; GOMES, Victor B. Enabling LoRaWAN communication over Wi-SUN smart grid networks. In: *ICC 2022 - IEEE International Conference on Communications*. [S.l.: s.n.], 2022. p. 4842–4847. Citado na página 74.

SHAFIQUE, Kinza; KHAWAJA, Bilal A.; SABIR, Farah; QAZI, Sameer; MUSTAQIM, Muhammad. Internet of things (IoT) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5G-IoT scenarios. *IEEE Access*, v. 8, p. 23022–23040, 2020. Citado na página 71.

SILVA, Hideson Alves da. *Algoritmo de otimização multinível aplicado a problemas de planejamento de redes*. Dissertação (Tese (doutorado). Orientador: Alceu Soares Britto Jr, Co-orientador: Luiz Eduardo Soares) — Programa de Pós-Graduação em Informática (PPGIa), Pontifícia Universidade Católica do Paraná, PUCPR, Brasil, 2012. Citado 3 vezes nas páginas 76, 91 e 93.

SONG, Xi; LI, Wenhui; LIU, Chao; WANG, KeMin; HOU, Yuting; BAO, Zhengrui; YUAN, Yaling. AI-Enabled quality prediction of 5G wireless network in smart grid. In: *2021 13th International Conference on Wireless Communications and Signal Processing (WCSP)*. [S.l.: s.n.], 2021. p. 1–6. Citado na página 31.

SOUZA, Gustavo Batista de Castro; VIEIRA, Flávio Henrique Teles; LIMA, Cláudio Ribeiro; JUNIOR, Getulio Antero de Deus; CASTRO, Marcelo Stehling de; ARAúJO, Sérgio Granato de. Optimal positioning of GPRS concentrators for minimizing node hops in smart grids considering routing in mesh networks. In: *2013 IEEE PES Conference on Innovative Smart Grid Technologies (ISGT Latin America)*. [S.l.: s.n.], 2013. p. 1–7. Citado 5 vezes nas páginas 76, 77, 90, 93 e 94.

STIRI, Souhaima; CHAOUB, Abdelaali; GRILO, António; BENNANI, Rachid; LAKSSIR, Brahim; TAMTAOUI, Ahmed. Hybrid PLC and LoRaWAN smart metering networks: Modeling and optimization. *IEEE Transactions on Industrial Informatics*, v. 18, n. 3, p. 1572–1582, 2022. Citado na página 94.

TANAKORNPINTONG, Songserm; TANGSUNANTHAM, Natthanan; SANGSUWAN, Tanayoot; PIRAK, Chaiyod. Location optimization for data concentrator unit in IEEE 802.15.4 smart grid networks. In: *2017 17th International Symposium on Communications and Information Technologies (ISCIT)*. [S.l.: s.n.], 2017. p. 1–6. Citado 3 vezes nas páginas 77, 91 e 93.

TANG, Maolin; CHEN, Chien-An. Wireless network gateway placement by evolutionary graph clustering. In: LIU, Derong; XIE, Shengli; LI, Yuanqing; ZHAO, Dongbin; EL-ALFY, El-Sayed M. (Ed.). *Neural Information Processing*. Cham: Springer International Publishing, 2017. p. 894–902. ISBN 978-3-319-70090-8. Citado 3 vezes nas páginas 84, 91 e 93.

TESTI, Enrico; FAVARELLI, Elia; PUCCI, Lorenzo; GIORGETTI, Andrea. Machine learning for wireless network topology inference. In: *2019 13th International Conference on Signal Processing and Communication Systems (ICSPCS)*. [S.l.: s.n.], 2019. p. 1–7. Citado 2 vezes nas páginas 71 e 95.

TIAN, Hongxian; WEITNAUER, Mary Ann; NYENGELE, Gedeon. Optimized gateway placement for interference cancellation in transmit-only LPWA networks. *Sensors*, v. 18, n. 11, 2018. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/18/11/3884>. Citado 3 vezes nas páginas 79, 90 e 93.

TORKZABAN, Nariman; BARAS, John S. *Controller Placement in SDN-enabled 5G Satellite-Terrestrial Networks*. arXiv, 2021. Disponível em: <https://arxiv.org/abs/2108.09176>. Citado 4 vezes nas páginas 85, 90, 91 e 93.

VLASOV, Andrey; ADAMOVA, Arina; SELIVANOV, Kirill. Development of smart grid technologies: organizational and communication aspects. *E3S Web of Conferences*, v. 250, p. 08001, 01 2021. Citado na página 40.

WANG, Guodong; ZHAO, Yanxiao; HUANG, Jun; WINTER, Robb M. On the data aggregation point placement in smart meter networks. In: *2017 26th International Conference on Computer Communication and Networks (ICCCN)*. [S.l.: s.n.], 2017. p. 1–6. Citado 5 vezes nas páginas 78, 90, 91, 93 e 94.

WANG, Guodong; ZHAO, Yanxiao; YING, Yulong; HUANG, Jun; WINTER, Robb M. Data aggregation point placement problem in neighborhood area networks of smart grid. *Mobile Networks and Applications*, v. 23, n. 4, p. 696–708, Aug 2018. ISSN 1572-8153. Disponível em: <https://doi.org/10.1007/s11036-018-1002-6>. Citado 4 vezes nas páginas 75, 90, 91 e 93.

Wi-SUN Alliance. *What We Do*. —-. Https://wi-sun.org/about/. Accessed on May 8, 2022. Disponível em: <https://wi-sun.org/about/>. Citado na página 41.

WU, Lina; HE, Danping; AI, Bo; WANG, Jian; QI, Hang; GUAN, Ke; ZHONG, Zhangdui. Artificial neural network based path loss prediction for wireless communication network. *IEEE Access*, v. 8, p. 199523–199538, 2020. Citado na página 50.

WZOREK, Mariusz; BERGER, Cyrille; DOHERTY, Patrick. Router and gateway node placement in wireless mesh networks for emergency rescue scenarios. *Autonomous Intelligent Systems*, v. 1, n. 1, p. 14, Dec 2021. ISSN 2730-616X. Disponível em: <https://doi.org/10.1007/s43684-021-00012-0>. Citado 3 vezes nas páginas 82, 91 e 93.

XGBoost developers. *dmlc XGBoost - XGBoost Documentation*. 2022. <https://xgboost.readthedocs.io/en/stable/index.html>. "On-line; acessado em 08/03/2023". Citado 2 vezes nas páginas 58 e 59.

XING, Ningzhe; ZHANG, Sidong; SHI, Yue; GUO, Shaoyong. PLC-oriented access point location planning algorithm in smart-grid communication networks. *China Communications*, v. 13, n. 9, p. 91–102, 2016. Citado 3 vezes nas páginas 78, 91 e 93.

YAO, Guang; BI, Jun; LI, Yuliang; GUO, Luyi. On the capacitated controller placement problem in software defined networks. *IEEE Communications Letters*, v. 18, n. 8, p. 1339–1342, 2014. Citado 3 vezes nas páginas 87, 91 e 93.

YARALI, Abdulrahman. Artificial intelligence, 5g, and iot. In: _____. *Intelligent Connectivity: AI, IoT, and 5G*. [S.l.: s.n.], 2022. p. 251–268. Citado na página 71.

YARALI, Abdulrahman. Big data and artificial intelligence. In: _____. *Intelligent Connectivity: AI, IoT, and 5G*. [S.l.: s.n.], 2022. p. 299–326. Citado na página 71.

ZHANG, Jianhua. Clustering and evolution of artificial intelligence technology in international sports. In: *2021 World Automation Congress (WAC)*. [S.l.: s.n.], 2021. p. 77–79. Citado na página 31.

ZHAO, Wenxin; LIU, Xubin; WU, Yujie; ZHANG, Tao; ZHANG, Luao. A learning-to-rank-based investment portfolio optimization framework for smart grid planning. *Frontiers in Energy Research*, v. 10, 2022. ISSN 2296-598X. Disponível em: <https://www.frontiersin.org/article/10.3389/fenrg.2022.852520>. Citado na página 71.

ZHEN, Todd; ELGINDY, Tarek; ALAM, S.M. Shafiul; HODGE, Bri-Mathias; LAIRD, Carl D. Optimal placement of data concentrators for expansion of the smart grid communications network. *IET Smart Grid*, v. 2, n. 4, p. 537–548, 2019. Disponível em: <https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-stg.2019.0006>. Citado 4 vezes nas páginas 86, 91, 93 e 94.

ZHENG, Weijun; CHEN, Ding; FANG, Jinghui; TANG, Jinjiang; WU, Guoqing; YAO, Jiming. Machine learning-based quality evaluation of 5G wireless network in smart grid. In: *2022 IEEE 6th Information Technology and Mechatronics Engineering Conference (ITOEC)*. [S.l.: s.n.], 2022. v. 6, p. 2006–2010. Citado na página 31.

ZöLLER, Marc-André; HUBER, Marco F. Benchmark and survey of automated machine learning frameworks. *Journal of Artificial Intelligence Research, JAIR, AI Access Foundation*, v. 70, p. 409–472, 2021. ISSN 1076 - 9757. Disponível em: <https://www.jair.org/index.php/jair/article/view/11854>. Citado na página 65.

Apêndices

APÊNDICE A – Lista de Características de Posições Candidatas

Tabela 20 – Lista de características de posições candidatas.

Início da Tabela de Características de Posições Candidatas		
Tipo	Característica	Descrição
Identificação	ID_CP	Identificador da posição candidata (CP)
Identificação	ID_Latitude	Latitude da coordenada da CP
Identificação	ID_Longitude	Longitude da coordenada da CP
Identificação	ID_NomeCidade	Nome da cidade ou região da CP
Identificação	ID_Iteracao	Número da iteração de AIDA a qual os resultados se referem
Local	L_SMs_CP	Total de medidores no entorno da posição, indicada como posição candidata CP. Considera o total de medidores no raio de alcance da CP
Local	L_SMs_SupACapacidade	*Flag* que indica que o total de medidores existentes na região é superior à capacidade da posição
Local	L_DA	*Flag* que indica se a CP possui (1) ou não possui (0) equipamento de automação (DA) instalado
Local (relevo)	L_Elev_CP	Elevação (em m) da coordenada geográfica da CP
Local (relevo)	L_DiffElev_N_100m	Diferença de elevação (em m) em relação a L_Elev_CP da coordenada a 100 m ao N da CP
Local (relevo)	L_DiffElev_N_200m	Diferença de elevação (em m) em relação a L_Elev_CP da coordenada a 200 m ao N da CP
Local (relevo)	L_DiffElev_N_300m	Diferença de elevação (em m) em relação a L_Elev_CP da coordenada a 300 m ao N da CP
Local (relevo)	L_DiffElev_N_400m	Diferença de elevação (em m) em relação a L_Elev_CP da coordenada a 400 m ao N da CP
Local (relevo)	L_DiffElev_N_500m	Diferença de elevação (em m) em relação a L_Elev_CP da coordenada a 500 m ao N da CP
Local (relevo)	L_DiffElev_N_1000m	Diferença de elevação (em m) em relação a L_Elev_CP da coordenada a 1000 m ao N da CP
Local (relevo)	L_DiffElev_N_1500m	Diferença de elevação (em m) em relação a L_Elev_CP da coordenada a 1500 m ao N da CP
Local (relevo)	L_DiffElev_N_2000m	Diferença de elevação (em m) em relação a L_Elev_CP da coordenada a 2000 m ao N da CP
Local (relevo)	L_DiffElev_N_2500m	Diferença de elevação (em m) em relação a L_Elev_CP da coordenada a 2500 m ao N da CP
Local (relevo)	L_DiffElev_N_3000m	Diferença de elevação (em m) em relação a L_Elev_CP da coordenada a 3000 m ao N da CP
...	...	...
Local (relevo)	L_DiffElev_X_NNNm	Diferença de elevação (em m) em relação a L_Elev_CP da coordenada a diferentes distâncias de determinada posição cardeal da CP, onde X={NE, E, SE, S, SW, W, NW} e NNN={100, 200, 300, 400, 500, 1000, 1500, 2000, 2500, 3000}
...	...	...
Local (relevo, qualidade de sinal)	L_LRP_N_100m	LRP entre a CP e a coordenada a 100 m ao N da CP.
Local (relevo, qualidade de sinal)	L_LRP_N_200m	LRP entre a CP e a coordenada a 200 m ao N da CP.

Continuação da Tabela 20 - Características de Posições Candidatas		
Tipo	Característica	Descrição
Local (relevo, qualidade de sinal)	L_LRP_N_300m	LRP entre a CP e a coordenada a 300 m ao N da CP.
Local (relevo, qualidade de sinal)	L_LRP_N_400m	LRP entre a CP e a coordenada a 400 m ao N da CP.
Local (relevo, qualidade de sinal)	L_LRP_N_500m	LRP entre a CP e a coordenada a 500 m ao N da CP.
Local (relevo, qualidade de sinal)	L_LRP_N_1000m	LRP entre a CP e a coordenada a 1000 m ao N da CP.
Local (relevo, qualidade de sinal)	L_LRP_N_1500m	LRP entre a CP e a coordenada a 1500 m ao N da CP.
Local (relevo, qualidade de sinal)	L_LRP_N_2000m	LRP entre a CP e a coordenada a 2000 m ao N da CP.
Local (relevo, qualidade de sinal)	L_LRP_N_2500m	LRP entre a CP e a coordenada a 2500 m ao N da CP.
Local (relevo, qualidade de sinal)	L_LRP_N_3000m	LRP entre a CP e a coordenada a 3000 m ao N da CP.
...	...	...
Local (relevo)	L_LRP_X_NNNm	LRP entre a CP e a coordenada a diferentes distâncias de determinada posição cardeal da CP, onde X={NE, E, SE, S, SW, W, NW} e NNN={100, 200, 300, 400, 500, 1000, 1500, 2000, 2500, 3000}
...	...	...
Local (relevo, qualidade de sinal)	L_LRP_N	Valor de LRP entre a CP de R0 e extremo N do alcance.
Local (relevo, qualidade de sinal)	L_LRP_NE	Valor de LRP entre a CP de R0 e extremo NE do alcance.
Local (relevo, qualidade de sinal)	L_LRP_E	Valor de LRP entre a CP de R0 e extremo E do alcance.
Local (relevo, qualidade de sinal)	L_LRP_SE	Valor de LRP entre a CP de R0 e extremo SE do alcance.
Local (relevo, qualidade de sinal)	L_LRP_S	Valor de LRP entre a CP de R0 e extremo S do alcance.
Local (relevo, qualidade de sinal)	L_LRP_SW	Valor de LRP entre a CP de R0 e extremo SW do alcance.
Local (relevo, qualidade de sinal)	L_LRP_W	Valor de LRP entre a CP de R0 e extremo W do alcance.
Local (relevo, qualidade de sinal)	L_LRP_NW	Valor de LRP entre a CP de R0 e extremo NW do alcance.
Local (densidade medidores)	L_Centroide_LAT	Latitude do centroide calculado com base nas posições dos medidores no alcance da CP de R0.
Local (densidade medidores)	L_Centroide_LON	Longitude do centroide calculado com base nas posições dos medidores no alcance da CP de R0.
Local (densidade medidores)	L_Dist_N	Distância do centroide de medidores de R0 até posição N no limite do alcance da posição.
Local (densidade medidores)	L_Dist_NE	Distância do centroide de medidores de R0 até posição NE no limite do alcance da posição.
Local (densidade medidores)	L_Dist_E	Distância do centroide de medidores de R0 até posição E no limite do alcance da posição.
Local (densidade medidores)	L_Dist_SE	Distância do centroide de medidores de R0 até posição SE no limite do alcance da posição.
Local (densidade medidores)	L_Dist_S	Distância do centroide de medidores de R0 até posição S no limite do alcance da posição.
Local (densidade medidores)	L_Dist_SW	Distância do centroide de medidores de R0 até posição SW no limite do alcance da posição.
Local (densidade medidores)	L_Dist_W	Distância do centroide de medidores de R0 até posição W no limite do alcance da posição.
Local (densidade medidores)	L_Dist_NW	Distância do centroide de medidores de R0 até posição NW no limite do alcance da posição.

Continuação da Tabela 20 - Características de Posições Candidatas		
Tipo	Característica	Descrição
Local (relevo, qualidade de sinal)	L_LRP_Centroide	Valor de LRP entre R0 e o centroide da região.
Local (densidade medidores)	L_DistMin_SMs	Distância entre a CP e o medidor mais próximo e dentro do raio de alcance.
Local (densidade medidores)	L_DistMax_SMs	Distância entre a CP e o medidor mais distante, mas dentro do raio de alcance.
Local (densidade medidores)	L_DistMed_SMs	Distância média entre a CP e os medidores dentro do raio de alcance.
Local (densidade medidores)	L_DistDevPad_SMs	Desvio padrão das distâncias entre a CP e os medidores dentro do raio de alcance.
Local (densidade de postes)	L_Postes_CP	Total de Postes no entorno da posição indicada como CP. Considera total de Postes no raio de alcance da CP.
Local (densidade de postes)	L_CentroidePostes_LAT	Latitude do centroide calculado com base nas posições dos Postes no alcance da CP de R0.
Local (densidade de postes)	L_CentroidePostes_LON	Longitude do centroide calculado com base nas posições dos Postes no alcance da CP de R0.
Local (densidade de postes)	L_DistPoste_N	Distância do centroide de Postes de R0 até posição N no limite do alcance da posição.
Local (densidade de postes)	L_DistPoste_NE	Distância do centroide de Postes de R0 até posição NE no limite do alcance da posição.
Local (densidade de postes)	L_DistPoste_E	Distância do centroide de Postes de R0 até posição E no limite do alcance da posição.
Local (densidade de postes)	L_DistPoste_SE	Distância do centroide de Postes de R0 até posição SE no limite do alcance da posição.
Local (densidade de postes)	L_DistPoste_S	Distância do centroide de Postes de R0 até posição S no limite do alcance da posição.
Local (densidade de postes)	L_DistPoste_SW	Distância do centroide de Postes de R0 até posição SW no limite do alcance da posição.
Local (densidade de postes)	L_DistPoste_W	Distância do centroide de Postes de R0 até posição W no limite do alcance da posição.
Local (densidade de postes)	L_DistPoste_NW	Distância do centroide de Postes de R0 até posição NW no limite do alcance da posição.
Local (relevo, qualidade de sinal)	L_Dist_CP_CentroideG	Distância (em m) entre a CP em análise e o centroide de SMs da região
Local (relevo, qualidade de sinal)	L_DiffElev_CP_CentroideG	Diferença de elevação (em m) entre a elevação da CP em análise e o centroide de SMs da região
Local (relevo, qualidade de sinal)	L_LRP_CP_CentroideG	Valor de LRP entre a CP em análise e o centroide de SMs da região
Regional (densidade de medidores)	R_SMs_R1	Total de medidores na região R1
Regional (densidade de medidores)	R_SMs_R2	Total de medidores na região R2
Regional (densidade de medidores)	R_SMs_R3	Total de medidores na região R3
Regional (densidade de medidores)	R_SMs_R4	Total de medidores na região R4
Regional (densidade de medidores)	R_SMs_R5	Total de medidores na região R5
Regional (densidade de medidores)	R_SMs_R6	Total de medidores na região R6
Regional (densidade de medidores)	R_SMs_R7	Total de medidores na região R7
Regional (densidade de medidores)	R_SMs_R8	Total de medidores na região R8
Regional (densidade de medidores)	R_perc_SMs	Percentual de medidores da CP em relação ao Total de medidores das regiões em análise (R1 a R8)

Continuação da Tabela 20 - Características de Posições Candidatas		
Tipo	Característica	Descrição
Regional (densidade de medidores)	R_Dist_R1	Distância do centroide de R0 à CP da região R1
Regional (densidade de medidores)	R_Dist_R2	Distância do centroide de R0 à CP da região R2
Regional (densidade de medidores)	R_Dist_R3	Distância do centroide de R0 à CP da região R3
Regional (densidade de medidores)	R_Dist_R4	Distância do centroide de R0 à CP da região R4
Regional (densidade de medidores)	R_Dist_R5	Distância do centroide de R0 à CP da região R5
Regional (densidade de medidores)	R_Dist_R6	Distância do centroide de R0 à CP da região R6
Regional (densidade de medidores)	R_Dist_R7	Distância do centroide de R0 à CP da região R7
Regional (densidade de medidores)	R_Dist_R8	Distância do centroide de R0 à CP da região R8
Regional (relevo, qualidade de sinal)	R_LRP_R1	LRP entre o centroide de R0 e a CP da região R1. Observação: Considera o centroide de R0 (ao invés das coordenadas da CP) como base para poder verificar qual a capacidade de conexão entre as demais regiões e o centro de concentração dos medidores de R0.
Regional (relevo, qualidade de sinal)	R_LRP_R2	LRP entre o centroide de R0 e a CP da região R2
Regional (relevo, qualidade de sinal)	R_LRP_R3	LRP entre o centroide de R0 e a CP da região R3
Regional (relevo, qualidade de sinal)	R_LRP_R4	LRP entre o centroide de R0 e a CP da região R4
Regional (relevo, qualidade de sinal)	R_LRP_R5	LRP entre o centroide de R0 e a CP da região R5
Regional (relevo, qualidade de sinal)	R_LRP_R6	LRP entre o centroide de R0 e a CP da região R6
Regional (relevo, qualidade de sinal)	R_LRP_R7	LRP entre o centroide de R0 e a CP da região R7
Regional (relevo, qualidade de sinal)	R_LRP_R8	LRP entre o centroide de R0 e a CP da região R8
Iteração (I) anterior	I_ID_CP1	ID_CP da CP da iteração anterior que é a 1^a mais próxima (CP1) da CP em análise
Iteração (I) anterior	I_L_SMs_CP1	Total de SMs no entorno da posição CP1 (mesmo valor já calculado para a CP na iteração anterior)
Iteração (I) anterior	I_L_SMs_SupACapacidade_CP1	*Flag* que indica que o total de SMs existentes na região de CP1 é superior à capacidade da posição (mesmo valor já calculado para a CP na iteração anterior)
Iteração (I) anterior	I_L_DA_CP1	*Flag* que indica se a posição CP1 possui (1) ou não possui (0) equipamento de automação (DA) instalado (mesmo valor já calculado para a CP na iteração anterior)
Iteração (I) anterior	I_Dist_CP_CP1	Distância entre a CP em análise e CP1. Distância da CP da iteração anterior que está mais próxima (CP1) da CP em análise
Iteração (I) anterior	I_Dist_CP_Centroide_CP1	Distância entre a CP em análise e o centroide dos SMs de CP1
Iteração (I) anterior	I_LRP_CP_CP1	LRP entre a posição de CP e a posição de CP1
Iteração (I) anterior	I_LRP_CP_Centroide_CP1	LRP entre a posição de CP e o centroide de SMs de CP1
...	...	...
Iteração (I) anterior	I_ID_CP8 (ignorar no treinamento)	ID_CP da CP da iteração anterior que é a 8^a mais próxima (CP8) da CP em análise

Continuação da Tabela 20 - Características de Posições Candidatas		
Tipo	Característica	Descrição
Iteração (I) anterior	I_L_SMs_CP8	Total de SMs no entorno da posição CP8. (mesmo valor já calculado para a CP na iteração anterior)
Iteração (I) anterior	I_L_SMs_SupACapacidade_CP8	Flag que indica que o total de SMs existentes na região de CP8 é superior à capacidade da posição (mesmo valor já calculado para a CP na iteração anterior)
Iteração (I) anterior	I_L_DA_CP8	Flag que indica se a posição CP8 possui (1) ou não possui (0) equipamento de automação (DA) instalado (mesmo valor já calculado para a CP na iteração anterior)
Iteração (I) anterior	I_Dist_CP_CP8	Distância entre a CP em análise e CP8
Iteração (I) anterior	I_Dist_CP_Centroide_CP8	Distância entre a CP em análise e o centroide dos SMs de CP8
Iteração (I) anterior	I_LRP_CP_CP8	LRP entre a posição de CP e a posição de CP8
Iteração (I) anterior	I_LRP_CP_Centroide_CP8	LRP entre a posição de CP e o centroide de SMs de CP8
Global (densidade de medidores)	G_perc_SMs	Percentual de medidores da CP em relação ao total de medidores da região em análise
Global (dimensões da região)	G_BboxArea_SMs	BoundingBox Area (em km^2) da região delimitada pelos medidores posicionados nos extremos da região
Global (dimensões da região)	G_Compriimento_W_E	Dimensão horizontal (de W para E) da região (em m). Comprimento medido horizontalmente entre os medidores dos pontos extremos em Leste e Oeste
Global (dimensões da região)	G_Comprimento_N_S	Dimensão vertical (de N para S) da região (em m). Comprimento medido verticalmente entre os medidores dos pontos extremos em Norte e Sul
Global (dimensões da região)	G_ComprimentoD_SW_NE	Dimensão diagonal (de SW para NE) da região (em m). Comprimento medido diagonalmente pontos extremos (inferior esquerdo e superior direito)
Global (qualidade de sinal)	G_LRP_Diagonal_SW_NE	Valor de LRP entre o extremos da diagonal de SW para NE
Global (qualidade de sinal)	G_LRP_Diagonal_NW_SE	Valor de LRP entre o extremos da diagonal de NW para SE
Global (densidade de medidores)	G_Total_SMs	Total de medidores da região em análise
Global (densidade de postes)	G_Total_Postes	Total de postes da região em análise
Global (densidade de medidores)	G_SMs_km2	Quantidade de medidores por km^2
Global (densidade de postes)	G_Postes_km2	Quantidade de postes por km^2
Global (posição)	G_Centroide_LAT (ignorar no treinamento)	Latitude do Centroide dos SMs da região
Global (posição)	G_Centroide_LON (ignorar no treinamento)	Longitude do Centroide dos SMs da região
Global (relevo, qualidade de sinal)	G_Dist_Centroide_N	Distância entre o centroide de SMs e o ponto mais a N
Global (relevo, qualidade de sinal)	G_Dist_Centroide_NE	Distância entre o centroide de SMs e o ponto mais a NE
Global (relevo, qualidade de sinal)	G_Dist_Centroide_E	Distância entre o centroide de SMs e o ponto mais a E
Global (relevo, qualidade de sinal)	G_Dist_Centroide_SE	Distância entre o centroide de SMs e o ponto mais a SE
Global (relevo, qualidade de sinal)	G_Dist_Centroide_S	Distância entre o centroide de SMs e o ponto mais a S
Global (relevo, qualidade de sinal)	G_Dist_Centroide_SW	Distância entre o centroide de SMs e o ponto mais a SW
Global (relevo, qualidade de sinal)	G_Dist_Centroide_W	Distância entre o centroide de SMs e o ponto mais a W

Continuação da Tabela 20 - Características de Posições Candidatas		
Tipo	Característica	Descrição
Global (relevo, qualidade de sinal)	G_Dist_Centroide_NW	Distância entre o centroide de SMs e o ponto mais a NW
Global (relevo, qualidade de sinal)	G_DiffElev_N	Diferença de elevação (em m) entre o centroide de SMs e a posição mais a N
Global (relevo, qualidade de sinal)	G_DiffElev_NE	Diferença de elevação (em m) entre o centroide de SMs e a posição mais a NE
Global (relevo, qualidade de sinal)	G_DiffElev_E	Diferença de elevação (em m) entre o centroide de SMs e a posição mais a E
Global (relevo, qualidade de sinal)	G_DiffElev_SE	Diferença de elevação (em m) entre o centroide de SMs e a posição mais a SE
Global (relevo, qualidade de sinal)	G_DiffElev_S	Diferença de elevação (em m) entre o centroide de SMs e a posição mais a S
Global (relevo, qualidade de sinal)	G_DiffElev_SW	Diferença de elevação (em m) entre o centroide de SMs e a posição mais a SW
Global (relevo, qualidade de sinal)	G_DiffElev_W	Diferença de elevação (em m) entre o centroide de SMs e a posição mais a W
Global (relevo, qualidade de sinal)	G_DiffElev_NW	Diferença de elevação (em m) entre o centroide de SMs e a posição mais a NW
Global (relevo, qualidade de sinal)	G_LRP_Centroide_N	Valor de LRP entre o centroide de SMs e o ponto mais a N
Global (relevo, qualidade de sinal)	G_LRP_Centroide_NE	Valor de LRP entre o centroide de SMs e o ponto mais a NE
Global (relevo, qualidade de sinal)	G_LRP_Centroide_E	Valor de LRP entre o centroide de SMs e o ponto mais a E
Global (relevo, qualidade de sinal)	G_LRP_Centroide_SE	Valor de LRP entre o centroide de SMs e o ponto mais a SE
Global (relevo, qualidade de sinal)	G_LRP_Centroide_S	Valor de LRP entre o centroide de SMs e o ponto mais a S
Global (relevo, qualidade de sinal)	G_LRP_Centroide_SW	Valor de LRP entre o centroide de SMs e o ponto mais a SW
Global (relevo, qualidade de sinal)	G_LRP_Centroide_W	Valor de LRP entre o centroide de SMs e o ponto mais a W
Global (relevo, qualidade de sinal)	G_LRP_Centroide_NW	Valor de LRP entre o centroide de SMs e o ponto mais a NW
Classe (target, rótulo)	CLASSE_CP	Classifica a posição candidata (CP) entre “1” (escolhida para a instalação de roteador) e “0” (descartada)
Final da Tabela 20 - Características de Posições Candidatas		

APÊNDICE B – Tabelas com Resultados de Experimentos Principais com AIDA-ML Utilizando Diferentes Quantidades de *Features*

Tabela 21 – Tabela com resultados do processo de Validação-ML do processamento de CPs definidas por AIDA-ML para *datasets* com *top-n* 20 *features*

			Abordagem TD (Top-Down)				Abordagem BU (Bottom-Up)			
Cidade	CPs AIDA Analitico	CPs AIDA-ML	CPs Usadas	Quant. SMs não conectados	% SMs não conectados	LRP médio	CPs Usadas	Quant. SMs não conectados	% SMs não conectados	LRP médio
AGUDOS DO SUL	44	36	34	359	6,67	-82,130	36	373	6,929	-79,076
ARAUCARIA	56	55	52	2862	5,417	-68,798	55	1612	3,051	-67,511
BALSA NOVA	44	44	44	17	0,278	-77,235	44	18	0,295	-70,976
CAMPO DO TENENTE	65	61	60	32	0,758	-75,454	61	29	0,687	-73,006
CARAMBEI	41	38	33	315	4,193	-75,349	37	33	0,439	-71,188
CONTENDA	43	60	55	63	0,611	-76,516	60	65	0,63	-69,117
FAZENDA RIO GRANDE	44	41	38	4807	8,56	-69,021	41	3972	7,073	-69,648
GUAMIRANGA	67	58	54	256	6,869	-80,109	58	252	6,761	-77,852
IMBITUVA	61	52	50	510	5,014	-70,796	52	404	3,972	-71,057
INACIO MARTINS	142	133	125	308	5,839	-79,793	130	304	5,763	-78,218
IRATI	105	83	81	186	0,808	-67,965	83	143	0,621	-63,594
IVAI	117	102	101	381	5,961	-77,155	101	373	5,835	-76,385
LAPA	171	165	157	155	0,801	-69,934	165	70	0,362	-67,282
MANDIRITUBA	62	55	53	74	0,633	-76,417	55	71	0,607	-71,452
PALMEIRA	164	169	155	110	0,739	-70,963	165	135	0,907	-67,302
PIEN	39	39	39	116	2,187	-78,120	39	121	2,282	-74,567
PONTA GROSSA	249	192	182	19432	12,873	-67,985	190	17626	11,677	-68,863
PORTO AMAZONAS	47	42	39	28	1,179	-68,387	42	28	1,179	-67,528
PRUDENTOPOLIS	196	187	180	154	0,811	-71,670	187	157	0,827	-69,247
QUITANDINHA	55	56	54	32	0,473	-78,296	56	30	0,444	-71,657
REBOUCAS	53	52	49	43	0,823	-72,028	51	43	0,823	-70,925
RIO AZUL	65	57	54	463	8,721	-75,576	56	245	4,615	-75,225
RIO NEGRO	38	35	31	97	6,025	-77,468	34	99	6,149	-72,582
SAO JOAO DO TRIUNFO	86	92	90	31	0,497	-76,714	92	35	0,561	-69,970
SAO MATEUS DO SUL	169	162	155	165	0,764	-73,353	160	160	0,741	-65,958
TEIXEIRA SOARES	62	58	56	151	3,111	-75,657	58	150	3,09	-73,545
Média:	88	82	78	1198	3,485	-74,342	81	1021	2,935	-71,297

Tabela 22 – Tabela com resultados do processo de Validação-ML do processamento de CPs definidas por AIDA-ML para *datasets* com *top-n* 40 *features*

			Abordagem TD (Top-Down)				Abordagem BU (Bottom-Up)			
Cidade	CPs AIDA Analitico	CPs AIDA-ML	CPs Usadas	Quant. SMs não conectados	% SMs não conectados	LRP médio	CPs Usadas	Quant. SMs não conectados	% SMs não conectados	LRP médio
AGUDOS DO SUL	44	34	32	508	9,437	-86,399	33	513	9,53	-84,482
ARAUCARIA	56	50	48	2885	5,46	-68,810	50	1690	3,199	-68,381
BALSA NOVA	44	45	45	18	0,295	-77,039	45	19	0,311	-70,614
CAMPO DO TENENTE	65	61	58	99	2,344	-84,648	61	87	2,06	-81,375
CARAMBEI	41	40	34	315	4,193	-75,328	38	46	0,612	-71,684
CONTENDA	43	56	52	41	0,397	-76,606	56	39	0,378	-69,540
FAZENDA RIO GRANDE	44	41	38	4668	8,312	-68,969	41	3367	5,996	-69,331
GUAMIRANGA	67	62	57	110	2,951	-78,706	62	106	2,844	-75,773
IMBITUVA	61	53	51	817	8,033	-71,454	53	140	1,376	-68,935
INACIO MARTINS	142	121	113	375	7,109	-80,022	116	372	7,052	-78,376
IRATI	105	86	77	166	0,721	-67,376	86	167	0,725	-61,078
IVAI	117	109	105	121	1,893	-76,983	108	111	1,737	-73,072
LAPA	171	151	147	1230	6,36	-73,502	151	1308	6,764	-74,254
MANDIRITUBA	62	58	55	64	0,548	-76,239	58	63	0,539	-71,399
PALMEIRA	164	168	154	214	1,437	-69,390	164	214	1,437	-65,284
PIEN	39	38	38	44	0,83	-77,104	38	49	0,924	-73,070
PONTA GROSSA	249	209	195	14442	9,567	-66,843	205	11490	7,612	-66,359
PORTO AMAZONAS	47	40	38	42	1,768	-69,212	40	42	1,768	-68,199
PRUDENTOPOLIS	196	194	187	91	0,479	-71,431	194	91	0,479	-68,801
QUITANDINHA	55	52	51	34	0,503	-78,380	52	38	0,562	-71,818
REBOUCAS	53	51	48	37	0,708	-71,875	50	37	0,708	-70,756
RIO AZUL	65	66	60	38	0,716	-72,864	65	32	0,603	-70,593
RIO NEGRO	38	34	31	104	6,46	-77,219	33	106	6,584	-73,897
SAO JOAO DO TRIUNFO	86	78	78	101	1,618	-77,752	78	83	1,33	-71,590
SAO MATEUS DO SUL	169	150	146	220	1,019	-73,512	149	211	0,978	-66,270
TEIXEIRA SOARES	62	57	55	144	2,967	-75,570	57	147	3,028	-69,559
Média:	88	81	77	1036	3,313	-74,740	80	791	2,659	-71,327

Tabela 23 – Tabela com resultados do processo de Validação-ML do processamento de CPs definidas por AIDA-ML para *datasets* com *top-n* 80 *features*

			Abordagem TD (Top-Down)				Abordagem BU (Bottom-Up)			
Cidade	CPs AIDA Analitico	CPs AIDA-ML	CPs Usadas	Quant. SMs não conectados	% SMs não conectados	LRP médio	CPs Usadas	Quant. SMs não conectados	% SMs não conectados	LRP médio
AGUDOS DO SUL	44	35	33	398	7,394	-85,798	35	401	7,449	-83,787
ARAUCARIA	56	49	46	3109	5,884	-69,633	49	2215	4,192	-69,549
BALSA NOVA	44	43	43	27	0,442	-76,795	43	28	0,459	-69,973
CAMPO DO TENENTE	65	64	58	92	2,178	-84,368	64	80	1,894	-80,785
CARAMBEI	41	39	33	288	3,834	-75,185	38	19	0,253	-71,534
CONTENDA	43	65	56	51	0,494	-76,654	65	49	0,475	-68,549
FAZENDA RIO GRANDE	44	38	36	4112	7,322	-68,804	38	3164	5,634	-69,768
GUAMIRANGA	67	62	57	71	1,905	-78,227	62	69	1,851	-75,608
IMBITUVA	61	52	50	1009	9,920	-71,804	52	866	8,514	-72,955
INACIO MARTINS	142	122	116	301	5,706	-79,535	119	299	5,668	-78,081
IRATI	105	83	78	176	0,764	-67,531	83	176	0,764	-61,508
IVAI	117	108	105	351	5,491	-76,918	107	319	4,991	-75,788
LAPA	171	159	151	1237	6,396	-73,120	159	1247	6,448	-73,655
MANDIRITUBA	62	57	55	67	0,573	-76,202	57	66	0,565	-71,051
PALMEIRA	164	156	144	209	1,404	-69,373	152	207	1,390	-65,506
PIEN	39	35	35	128	2,414	-78,093	35	144	2,715	-74,650
PONTA GROSSA	249	209	192	14562	9,647	-66,424	203	13026	8,629	-66,434
PORTO AMAZONAS	47	45	43	23	0,968	-68,560	45	23	0,968	-66,328
PRUDENTOPOLIS	196	190	186	138	0,727	-71,412	190	143	0,753	-68,729
QUITANDINHA	55	53	52	26	0,385	-78,534	53	30	0,444	-71,899
REBOUCAS	53	53	49	35	0,670	-71,867	52	35	0,670	-70,698
RIO AZUL	65	67	61	30	0,565	-73,006	67	24	0,452	-70,672
RIO NEGRO	38	33	30	33	2,050	-76,572	32	34	2,112	-73,264
SAO JOAO DO TRIUNFO	86	82	80	185	2,964	-78,246	82	167	2,676	-71,960
SAO MATEUS DO SUL	169	167	157	224	1,038	-73,287	165	217	1,005	-65,983
TEIXEIRA SOARES	62	59	57	139	2,864	-75,059	59	143	2,946	-69,059
Média:	88	82	77	1039	3,231	-74,654	81	892	2,843	-71,453

Tabela 24 – Tabela com resultados do processo de Validação-ML do processamento de CPs definidas por AIDA-ML para *datasets* com *top-n* 120 *features*

			Abordagem TD (Top-Down)				Abordagem BU (Bottom-Up)			
Cidade	CPs AIDA Analitico	CPs AIDA-ML	CPs Usadas	Quant. SMs não conectados	% SMs não conectados	LRP médio	CPs Usadas	Quant. SMs não conectados	% SMs não conectados	LRP médio
AGUDOS DO SUL	44	38	36	243	4,514	-81,366	37	254	4,719	-77,523
ARAUCARIA	56	51	48	3107	5,88	-69,593	51	2163	4,094	-69,208
BALSA NOVA	44	38	38	28	0,459	-77,286	38	33	0,54	-71,116
CAMPO DO TENENTE	65	58	53	130	3,078	-85,309	58	113	2,675	-81,999
CARAMBEI	41	37	34	314	4,18	-75,296	37	45	0,599	-71,698
CONTENDA	43	61	56	51	0,494	-76,704	61	49	0,475	-69,173
FAZENDA RIO GRANDE	44	36	35	5630	10,025	-68,966	36	4425	7,88	-69,993
GUAMIRANGA	67	58	54	41	1,1	-77,423	58	39	1,046	-74,601
IMBITUVA	61	51	49	1020	10,029	-71,821	51	877	8,623	-73,087
INACIO MARTINS	142	125	119	347	6,578	-80,191	121	346	6,559	-78,502
IRATI	105	80	76	276	1,199	-68,165	80	228	0,99	-62,470
IVAI	117	115	105	351	5,491	-77,375	111	346	5,413	-76,875
LAPA	171	154	148	2897	14,98	-75,407	153	3115	16,107	-76,983
MANDIRITUBA	62	51	49	196	1,677	-76,878	51	196	1,677	-72,217
PALMEIRA	164	158	146	233	1,565	-71,301	154	232	1,558	-67,590
PIEN	39	37	37	58	1,094	-77,329	37	74	1,395	-73,551
PONTA GROSSA	249	198	185	14850	9,838	-66,914	193	13070	8,658	-66,619
PORTO AMAZONAS	47	45	44	31	1,305	-68,857	45	31	1,305	-66,258
PRUDENTOPOLIS	196	193	189	110	0,579	-71,150	192	103	0,543	-68,482
QUITANDINHA	55	51	51	40	0,592	-78,675	51	47	0,695	-72,085
REBOUCAS	53	46	43	43	0,823	-72,148	45	43	0,823	-71,145
RIO AZUL	65	60	56	36	0,678	-73,270	59	30	0,565	-71,252
RIO NEGRO	38	37	33	33	2,05	-76,085	36	34	2,112	-72,406
SAO JOAO DO TRIUNFO	86	78	78	75	1,202	-77,444	78	80	1,282	-71,747
SAO MATEUS DO SUL	169	159	151	216	1,001	-73,391	158	183	0,848	-66,047
TEIXEIRA SOARES	62	57	54	139	2,864	-75,155	57	143	2,946	-73,048
Média:	88	80	76	1173	3,588	-74,750	79	1012	3,236	-71,757

Tabela 25 – Tabela com resultados do processo de Validação-ML do processamento de CPs definidas por AIDA-ML para *datasets* com 318 *features*

			Abordagem TD (Top-Down)				Abordagem BU (Bottom-Up)			
Cidade	CPs AIDA Analitico	CPs AIDA-ML	CPs Usadas	Quant. SMs não conectados	% SMs não conectados	LRP médio	CPs Usadas	Quant. SMs não conectados	% SMs não conectados	LRP médio
AGUDOS DO SUL	44	34	32	269	4,997	-81,954	33	279	5,183	-80,559
ARAUCARIA	56	48	47	5208	9,857	-70,874	48	4026	7,62	-70,541
BALSA NOVA	44	41	41	27	0,442	-77,273	41	27	0,442	-71,311
CAMPO DO TENENTE	65	62	59	82	1,941	-84,723	61	66	1,563	-81,308
CARAMBEI	41	43	36	103	1,371	-74,806	43	33	0,439	-68,175
CONTENDA	43	56	52	50	0,485	-76,810	56	51	0,494	-69,787
FAZENDA RIO GRANDE	44	36	34	7075	12,599	-70,678	36	6208	11,055	-73,111
GUAMIRANGA	67	60	56	74	1,986	-78,138	60	70	1,878	-75,360
IMBITUVA	61	51	51	795	7,816	-71,292	51	119	1,17	-68,799
INACIO MARTINS	142	122	112	342	6,483	-80,040	115	342	6,483	-78,681
IRATI	105	78	76	379	1,646	-68,548	78	300	1,303	-64,110
IVAI	117	107	103	370	5,788	-77,648	106	360	5,632	-75,857
LAPA	171	152	147	1534	7,932	-74,594	152	1359	7,027	-74,381
MANDIRITUBA	62	57	54	81	0,693	-76,414	57	81	0,693	-71,644
PALMEIRA	164	163	151	123	0,826	-69,344	162	121	0,813	-65,375
PIEN	39	33	33	139	2,621	-78,531	33	152	2,866	-75,187
PONTA GROSSA	249	200	182	21854	14,478	-68,915	194	20831	13,8	-68,955
PORTO AMAZONAS	47	45	44	16	0,674	-68,255	45	16	0,674	-66,356
PRUDENTOPOLIS	196	177	174	1269	6,685	-75,025	177	719	3,788	-72,102
QUITANDINHA	55	51	51	36	0,532	-78,616	51	41	0,606	-72,035
REBOUCAS	53	46	43	70	1,34	-72,294	45	72	1,378	-71,556
RIO AZUL	65	64	60	441	8,307	-74,912	64	223	4,2	-74,141
RIO NEGRO	38	33	31	98	6,087	-77,966	32	100	6,211	-74,895
SAO JOAO DO TRIUNFO	86	82	79	65	1,041	-77,205	82	45	0,721	-71,292
SAO MATEUS DO SUL	169	146	140	1032	4,782	-75,115	145	266	1,232	-68,567
TEIXEIRA SOARES	62	61	58	29	0,597	-73,968	61	33	0,68	-71,311
Média:	88	79	75	1599	4,308	-75,151	78	1382	3,383	-72,131

APÊNDICE C – Tabelas com Resultados de Experimentos Principais com AIDA-ML Utilizando *Datasets* com 40 *Features*

Tabela 26 – Tabela com resultados do processo de Validação-ML do processamento de CPs definidas por AIDA-ML para *datasets* com *top-n* 40 *features* com acréscimo de 10% de CPs

			Abordagem TD (Top-Down)				Abordagem BU (Bottom-Up)			
Cidade	CPs AIDA Analitico	CPs AIDA-ML	CPs Usadas	Quant. SMs não conectados	% SMs não conectados	LRP médio	CPs Usadas	Quant. SMs não conectados	% SMs não conectados	LRP médio
AGUDOS DO SUL	44	38	35	246	4,57	-84,431	37	266	4,941	-82,336
ARAUCARIA	56	55	53	2884	5,458	-69,206	55	1687	3,193	-68,299
BALSA NOVA	44	50	48	18	0,295	-76,776	50	19	0,311	-69,492
CAMPO DO TENENTE	65	68	63	73	1,728	-84,088	68	60	1,42	-80,11
CARAMBEI	41	44	35	36	0,479	-66,799	42	38	0,506	-64,88
CONTENDA	43	62	55	29	0,281	-76,447	62	27	0,262	-68,519
FAZENDA RIO GRANDE	44	46	42	2358	4,199	-67,711	46	1435	2,555	-66,908
GUAMIRANGA	67	69	63	46	1,234	-77,33	68	42	1,127	-73,332
IMBITUVA	61	59	56	577	5,673	-70,354	59	24	0,236	-65,302
INACIO MARTINS	142	134	124	305	5,782	-79,288	129	303	5,744	-77,3
IRATI	105	95	85	141	0,612	-66,796	95	147	0,638	-60,813
IVAI	117	120	115	99	1,549	-76,802	119	89	1,392	-72,588
LAPA	171	167	159	162	0,838	-71,267	166	284	1,469	-70,096
MANDIRITUBA	62	64	60	47	0,402	-75,869	64	46	0,394	-70,099
PALMEIRA	164	185	162	86	0,578	-68,97	179	86	0,578	-64,702
PIEN	39	42	40	35	0,66	-77,362	42	40	0,754	-72,079
PONTA GROSSA	249	230	210	5338	3,536	-65,245	226	3458	2,291	-62,843
PORTO AMAZONAS	47	44	42	29	1,221	-68,908	44	29	1,221	-67,71
PRUDENTOPOLIS	196	214	201	50	0,263	-71,288	213	50	0,263	-68,423
QUITANDINHA	55	58	56	9	0,133	-78,019	58	9	0,133	-71,322
REBOUCAS	53	57	53	33	0,632	-71,649	56	33	0,632	-70,39
RIO AZUL	65	73	64	32	0,603	-72,734	72	26	0,49	-69,861
RIO NEGRO	38	38	34	16	0,994	-75,993	37	18	1,118	-72,442
SAO JOAO DO TRIUNFO	86	86	85	51	0,817	-77,038	86	55	0,881	-70,723
SAO MATEUS DO SUL	169	165	157	165	0,764	-73,293	164	156	0,723	-65,87
TEIXEIRA SOARES	62	63	60	132	2,719	-74,884	62	136	2,802	-68,821
Média:	88	89	83	500	1,770	-73,790	88	329	1,387	-69,818

Tabela 27 – Tabela com resultados do processo de Validação-ML do processamento de CPs definidas por AIDA-ML para *datasets* com *top-n* 40 *features* com acréscimo de 15% de CPs

			Abordagem TD (Top-Down)				Abordagem BU (Bottom-Up)			
Cidade	CPs AIDA Analitico	CPs AIDA-ML	CPs Usadas	Quant. SMs não conectados	% SMs não conectados	LRP médio	CPs Usadas	Quant. SMs não conectados	% SMs não conectados	LRP médio
AGUDOS DO SUL	44	40	37	193	3,585	-83,073	39	203	3,771	-79,744
ARAUCARIA	56	58	56	2496	4,724	-68,845	58	876	1,66	-67,077
BALSA NOVA	44	52	49	17	0,278	-76,791	52	18	0,295	-68,802
CAMPO DO TENENTE	65	71	63	24	0,568	-80,087	71	24	0,568	-76,623
CARAMBEI	41	46	36	36	0,479	-66,8	44	38	0,506	-64,871
CONTENDA	43	65	58	12	0,116	-76,092	65	12	0,116	-67,671
FAZENDA RIO GRANDE	44	48	44	2092	3,725	-66,471	48	1018	1,813	-65,413
GUAMIRANGA	67	72	65	23	0,617	-76,751	71	21	0,563	-72,544
IMBITUVA	61	61	57	577	5,673	-70,33	61	24	0,236	-65,274
INACIO MARTINS	142	140	129	290	5,498	-79,144	135	288	5,46	-76,953
IRATI	105	99	89	132	0,573	-66,807	99	138	0,599	-59,47
IVAI	117	126	121	91	1,424	-76,599	125	81	1,267	-72,302
LAPA	171	174	166	154	0,796	-71,225	173	277	1,432	-69,935
MANDIRITUBA	62	67	63	29	0,248	-75,795	67	29	0,248	-69,611
PALMEIRA	164	194	168	52	0,349	-68,91	188	51	0,343	-64,502
PIEN	39	44	42	35	0,66	-77,327	44	40	0,754	-72,038
PONTA GROSSA	249	241	217	3275	2,17	-64,744	236	1605	1,063	-61,643
PORTO AMAZONAS	47	46	44	14	0,589	-67,873	46	14	0,589	-66,634
PRUDENTOPOLIS	196	224	208	41	0,216	-71,118	223	37	0,195	-68,1
QUITANDINHA	55	60	58	9	0,133	-77,876	60	8	0,118	-71,114
REBOUCAS	53	59	53	33	0,632	-71,649	58	33	0,632	-70,37
RIO AZUL	65	76	65	32	0,603	-72,579	75	26	0,49	-69,585
RIO NEGRO	38	40	36	12	0,745	-75,682	39	13	0,807	-72,088
SAO JOAO DO TRIUNFO	86	90	89	17	0,272	-76,776	90	21	0,336	-70,115
SAO MATEUS DO SUL	169	173	163	124	0,575	-73,038	171	116	0,537	-65,544
TEIXEIRA SOARES	62	66	61	125	2,575	-74,713	65	129	2,658	-68,55
Média:	88	94	86	382	1,455	-73,350	92	198	1,041	-69,099

Tabela 28 – Tabela com resultados do processo de Validação-ML do processamento de CPs definidas por AIDA-ML para *datasets* com *top-n* 40 *features* com PROBA_1 ≥ 0.20

			Abordagem TD (Top-Down)				Abordagem BU (Bottom-Up)			
Cidade	CPs AIDA Analitico	CPs AIDA-ML	CPs Usadas	Quant. SMs não conectados	% SMs não conectados	LRP médio	CPs Usadas	Quant. SMs não conectados	% SMs não conectados	LRP médio
AGUDOS DO SUL	44	41	37	193	3,585	-82,99	40	203	3,771	-79,648
ARAUCARIA	56	53	51	2885	5,46	-69,204	53	1690	3,199	-68,353
BALSA NOVA	44	48	47	18	0,295	-76,881	48	19	0,311	-69,643
CAMPO DO TENENTE	65	64	61	94	2,225	-84,393	64	82	1,941	-80,751
CARAMBEI	41	45	36	36	0,479	-66,799	43	38	0,506	-64,879
CONTENDA	43	66	59	12	0,116	-76,091	66	12	0,116	-67,63
FAZENDA RIO GRANDE	44	45	42	2663	4,742	-68,709	45	1878	3,344	-68,044
GUAMIRANGA	67	75	67	21	0,563	-76,62	74	19	0,51	-72,014
IMBITUVA	61	62	58	577	5,673	-70,329	62	24	0,236	-65,268
INACIO MARTINS	142	143	131	288	5,46	-79,095	138	286	5,422	-76,78
IRATI	105	92	83	151	0,656	-67,294	92	157	0,682	-60,925
IVAI	117	143	130	65	1,017	-77,135	141	52	0,814	-71,241
LAPA	171	180	168	153	0,791	-71,136	179	276	1,427	-69,804
MANDIRITUBA	62	65	61	47	0,402	-75,844	65	46	0,394	-69,903
PALMEIRA	164	186	163	84	0,564	-68,967	180	84	0,564	-64,692
PIEN	39	43	41	35	0,66	-77,352	43	40	0,754	-72,063
PONTA GROSSA	249	245	220	2678	1,774	-64,434	239	491	0,325	-61,017
PORTO AMAZONAS	47	49	46	9	0,379	-67,768	49	9	0,379	-66,43
PRUDENTOPOLIS	196	245	223	35	0,184	-70,909	244	35	0,184	-67,574
QUITANDINHA	55	57	55	22	0,325	-78,189	57	26	0,385	-71,552
REBOUCAS	53	55	52	37	0,708	-71,718	54	37	0,708	-70,463
RIO AZUL	65	76	65	32	0,603	-72,579	75	26	0,49	-69,585
RIO NEGRO	38	36	32	22	1,366	-75,986	35	24	1,491	-72,62
SAO JOAO DO TRIUNFO	86	86	85	51	0,817	-77,038	86	55	0,881	-70,723
SAO MATEUS DO SUL	169	169	160	131	0,607	-73,081	168	122	0,565	-65,604
TEIXEIRA SOARES	62	65	61	125	2,575	-74,738	64	129	2,658	-68,593
Média:	88	94	86	402	1,616	-73,665	92	225	1,233	-69,454

Tabela 29 – Tabela com resultados do processo de Validação-ML do processamento de CPs definidas por AIDA-ML para *datasets* com *top-n* 40 *features* com PROBA_1 ≥ 0.30

			Abordagem TD (Top-Down)				Abordagem BU (Bottom-Up)			
Cidade	CPs AIDA Analitico	CPs AIDA-ML	CPs Usadas	Quant. SMs não conectados	% SMs não conectados	LRP médio	CPs Usadas	Quant. SMs não conectados	% SMs não conectados	LRP médio
AGUDOS DO SUL	44	38	35	246	4,57	-84,431	37	266	4,941	-82,336
ARAUCARIA	56	51	49	2885	5,46	-69,21	51	1690	3,199	-68,365
BALSA NOVA	44	47	46	18	0,295	-76,933	47	19	0,311	-69,789
CAMPO DO TENENTE	65	61	58	99	2,344	-84,648	61	87	2,06	-81,375
CARAMBEI	41	43	35	36	0,479	-66,799	41	38	0,506	-64,882
CONTENDA	43	61	55	29	0,281	-76,449	61	27	0,262	-68,571
FAZENDA RIO GRANDE	44	44	41	2909	5,18	-68,789	44	2007	3,574	-68,41
GUAMIRANGA	67	69	63	46	1,234	-77,33	68	42	1,127	-73,332
IMBITUVA	61	59	56	577	5,673	-70,354	59	24	0,236	-65,302
INACIO MARTINS	142	135	124	305	5,782	-79,305	130	303	5,744	-77,238
IRATI	105	89	80	164	0,712	-67,347	89	165	0,717	-60,98
IVAI	117	124	119	92	1,439	-76,675	123	82	1,283	-72,429
LAPA	171	166	156	162	0,838	-71,274	165	284	1,469	-70,111
MANDIRITUBA	62	61	58	51	0,436	-76,016	61	50	0,428	-70,451
PALMEIRA	164	179	161	95	0,638	-69,124	173	95	0,638	-64,866
PIEN	39	40	38	44	0,83	-77,491	40	49	0,924	-72,234
PONTA GROSSA	249	224	206	9228	6,113	-65,922	220	6617	4,384	-64,1
PORTO AMAZONAS	47	45	43	14	0,589	-68,136	45	14	0,589	-66,936
PRUDENTOPOLIS	196	222	208	41	0,216	-71,118	221	37	0,195	-68,112
QUITANDINHA	55	55	54	22	0,325	-78,231	55	26	0,385	-71,611
REBOUCAS	53	53	50	37	0,708	-71,829	52	37	0,708	-70,688
RIO AZUL	65	73	64	32	0,603	-72,734	72	26	0,49	-69,861
RIO NEGRO	38	36	32	22	1,366	-75,986	35	24	1,491	-72,62
SAO JOAO DO TRIUNFO	86	80	79	89	1,426	-77,452	80	71	1,138	-71,259
SAO MATEUS DO SUL	169	161	154	205	0,95	-73,363	160	196	0,908	-65,988
TEIXEIRA SOARES	62	63	60	132	2,719	-74,884	62	136	2,802	-68,821
Média:	88	88	82	676	1,969	-73,917	87	477	1,558	-70,026

APÊNDICE D – Tabelas com Resultados de Experimentos Adicionais com Técnicas de AutoML, Otimização de Hiperparâmetros e Seleção de *Features*

Tabela 30 – Resultados de experimentos adicionais com *datasets* com 40 *features* após HPO.

Início da Tabela de Resultados com 40 *features* após HPO.				
Features	**Técnica**	**Acurácia (LOSO)**	**Classificador**	**Hiperparâmetros**
40	skopt.dummy_minimize (Busca aleatória, 500 calls)	**0,8734**	XGBoost (xgboost.XGBClassifier)	learning_rate = 0,02504, n_estimators = 2238, max_depth = 12, min_child_weight = 11, subsample = 0,70774, colsample_bynode = 0,30457, num_parallel_tree = 2, gamma = 1,92242, colsample_bytree = 0,61318
40	skopt.dummy_minimize (Busca aleatória, 100 calls)	**0,8728**	XGBoost (xgboost.XGBClassifier)	learning_rate = 0,01661, n_estimators = 1354, max_depth = 11, min_child_weight = 10, subsample = 0,74514, colsample_bynode = 0,33373, num_parallel_tree = 6, gamma = 1,51908, colsample_bytree = 0,868934
40	skopt.gp_minimize (otimização bayesiana usando processos gaussianos, 500 calls)	**0,8725**	LightGBM (lightgbm.LGBMClassifier)	max_bin = 1227, learning_rate = 0,027542196740280862, n_estimators = 2500, num_leaves = 65
40	skopt.gp_minimize (otimização bayesiana usando processos gaussianos, 100 calls)	0,872	XGBoost (xgboost.XGBClassifier)	learning_rate = 0,00603, n_estimators = 2334, max_depth = 20, min_child_weight = 6, subsample = 1,0, colsample_bynode = 0,52707, num_parallel_tree = 7, gamma = 1,20866, colsample_bytree = 0,820437
40	skopt.gbrt_minimize (otimização sequencial usando gradient boosted trees)	0,872	XGBoost (xgboost.XGBClassifier)	learning_rate = 0,003625, n_estimators = 3743, max_depth = 6, min_child_weight = 8, subsample = 0,66296, colsample_bynode = 0,11123, num_parallel_tree = 8, gamma = 1,03936, colsample_bytree = 0,927287
40	skopt.dummy_minimize (Busca aleatória, 30 calls)	0,871	XGBoost (xgboost.XGBClassifier)	learning_rate = 0,00727, n_estimators = 1210, max_depth = 16, min_child_weight = 11, subsample = 0,96985, colsample_bynode = 0,38208, num_parallel_tree = 7, gamma = 2,83169, colsample_bytree = 0,94600
40	skopt.gp_minimize (otimização bayesiana usando processos gaussianos, 100 calls)	0,871	LightGBM (lightgbm.LGBMClassifier)	max_bin = 1486, learning_rate = 0.10038589718153176, n_estimators = 385, num_leaves = 96
40	skopt.dummy_minimize (Busca aleatória, 30 calls)	0,870	XGBoost (xgboost.XGBClassifier)	learning_rate = 0,02126, n_estimators = 249, max_depth = 14, min_child_weight = 9, subsample = 0,5637, colsample_bynode = 0,50318, num_parallel_tree = 5

Continuação da Tabela 30 - Resultados com 40 *features* após HPO.

Features	**Técnica**	**Acurácia (LOSO)**	**Classificador**	**Hiperparâmetros**
40	skopt.gp_minimize (otimização bayesiana usando processos gaussianos, 100 calls)	0,870	Histogram-based Gradient Boosting(sklearn. ensemble. HistGradientBoostingClassifier)	max_iter = 1882, learning_rate = 0,15903, max_depth = 71, l2_regularization = 0,18288
40	TPOT (verbosity=1, population_size = 50, config_dict="TPOT cuML", n_jobs = -1, max_time_mins=5, use_dask=True, random_state=2023)	0,868	XGBoost (xgboost.XGBClassifier)	alpha=1, learning_rate=0.1, max_depth=9, min_child_weight=9, n_estimators=100, n_jobs=1, subsample=0.7000000000000001, tree_method="gpu_hist", verbosity=0
40	Auto-sklearn	0,866	Histogram-based Gradient Boosting(sklearn. ensemble. HistGradientBoostingClassifier)	early_stopping=True, l2_regularization= 0.00487178522148225, learning_rate=0.024432206340259912, max_iter=512, max_leaf_nodes=9, min_samples_leaf=189, n_iter_no_change=1, random_state=2003, validation_fraction = None, warm_start=True
40	skopt.gp_minimize (otimização bayesiana usando processos gaussianos, 300 calls)	0,866	Random Forest (cuml.ensemble. RandomForestClassifier)	n_estimators = 2149, max_features = 'sqrt', max_depth = 27, min_samples_split = 10, min_samples_leaf = 1, bootstrap = False
40	skopt.gp_minimize (otimização bayesiana usando processos gaussianos, 200 calls)	0,865	Random Forest (cuml.ensemble. RandomForestClassifier)	n_estimators = 2500, max_features = 'auto', max_depth = 16, min_samples_split = 5, min_samples_leaf = 1, bootstrap = False
40	RandomizedSearchCV (dask_ml.model_selection. RandomizedSearcCV, 100 iterações)	0,864	XGBoost (xgboost.XGBClassifier)	alpha=0.09999999999999999, learning_rate=0.1, max_depth=9, min_child_weight=2, n_estimators=200
40	LOSO, LGBM, Default params	0,864	LightGBM (lightgbm. LGBMClassifier)	Default, random_state=2023
40	TPOT (verbosity=3, population_size = 500, n_jobs = 60, max_time_mins=900, random_state=2023)	0,864	Gradient Boosting (sklearn.ensemble. GradientBoostingClassifier)	learning_rate=0.1, max_depth=10, max_features=0.55, min_samples_leaf=17, min_samples_split=4, n_estimators=100, subsample=0.8500000000000001
40	LOSO, XGBClassifier, default params, seed = 2022	0,8614	XGBoost (xgboost.XGBClassifier)	Default, seed=2022
40	BASELINE	0,8589	XGBoost (xgboost.XGBClassifier)	tree_method='hist', learning_rate=1, max_depth=15, reg_lambda=20, n_estimators=500
40	skopt.gp_minimize (otimização bayesiana usando processos gaussianos, 100 calls)	0,549	Logistic Regression (cuml. linear_model. logisticregression)	penalty = 'none', C = 93,25641

Final da Tabela 30 - Resultados com 40 *features* após HPO.

Tabela 31 – Resultados de experimentos adicionais com *datasets* com 318 *features* após HPO.

Features	**Técnica**	**Acurácia (LOSO)**	**Classificador**	**Hiperparâmetros**
318	skopt.dummy_minimize (Busca aleatória, 30 calls)	**0,8646**	XGBoost (xgboost.XGBClassifier)	learning_rate = 0,02126, n_estimators = 249, max_depth = 14, min_child_weight = 9, subsample = 0,56375, colsample_bynode = 0,50318, num_parallel_tree = 5
318	skopt.dummy_minimize (Busca aleatória, 30 calls)	0,864	XGBoost (xgboost.dask. DaskXGBClassifier)	learning_rate = 0,02496, n_estimators = 1337, max_depth = 21, min_child_weight = 7, subsample = 0,60669, colsample_bynode = 0,870104, num_parallel_tree = 3
318	TPOT (verbosity=3, population_size = 500, n_jobs = 60, max_time_mins=900, random_state=2023)	0,861	Gradient Boosting (sklearn.ensemble. GradientBoostingClassifier)	learning_rate=0.1, max_depth=10, max_features=0.55, min_samples_leaf=17, min_samples_split=4, n_estimators=100, subsample=0.8500000000000001
318	RandomizedSearchCV (dask_ml.model_selection. RandomizedSearcCV, 100 iterações)	0,861	XGBoost (xgboost.XGBClassifier)	alpha=0.09999999999999999, learning_rate=0.1, max_depth=9, min_child_weight=2, n_estimators=200
318	TPOT (verbosity=1, population_size = 50, config_dict="TPOT cuML", n_jobs = -1, max_time_mins=5, use_dask=True, random_state=2023)	0,860	XGBoost (xgboost.XGBClassifier)	alpha=1, learning_rate=0.1, max_depth=9, min_child_weight=9, n_estimators=100, n_jobs=1, subsample=0.7000000000000001, tree_method="gpu_hist", verbosity=0
318	XGBClassifier, default params, seed = 2022	0,860	XGBoost (xgboost.XGBClassifier)	Default, seed=2022
318	TPOT (verbosity=2, population_size = 500, config_dict="TPOT cuML", n_jobs = 100, max_time_mins=1200, use_dask=True, random_state=2023)	0,859	XGBoost (xgboost.XGBClassifier)	alpha=10, learning_rate=0.1, max_depth=5, min_child_weight=5, n_estimators=100, n_jobs=1, subsample=0.8, tree_method="gpu_hist", verbosity=0
318	TPOT (verbosity=3, population_size = 500, n_jobs = 60, max_time_mins=900, random_state=2023)	0,859	Gradient Boosting (sklearn.ensemble. GradientBoostingClassifier)	learning_rate=0.1, max_depth=10, max_features=0.55, min_samples_leaf=17, min_samples_split=4, n_estimators=100, subsample=0.8500000000000001
318	TPOT (verbosity=2, generations=GENERATIONS, population_size = POP_SIZE, cv=CV, n_jobs = 1, config_dict="TPOT cuML", use_dask=True, random_state=2023)	0,859	XGBoost (xgboost.XGBClassifier)	alpha=10, learning_rate=0.1, max_depth=9, min_child_weight=19, n_estimators=100, n_jobs=1, subsample=0.7500000000000001, tree_method="gpu_hist", verbosity=0
318	Auto-sklearn	0,857	Histogram-based Gradient Boosting(sklearn. ensemble. HistGradientBoostingClassifier)	early_stopping=True, l2_regularization= 0.00487178522148225, learning_rate=0.024432206340259912, max_iter=512, max_leaf_nodes=9, min_samples_leaf=189, n_iter_no_change=1, random_state=2003,validation_fraction = None, warm_start=True
318	BASELINE	0,856	XGBoost (xgboost.XGBClassifier)	tree_method = 'hist', learning_rate=1, max_depth=15, reg_lambda = 20, n_estimators = 500, seed = 2023
318	Teste manual com LGBM, parâmetros default	0,856	LightGBM (lightgbm. LGBMClassifier)	Default, random_state=2023
318	Teste manual com GradientBoostingClassifier, parâmetros default	0,854	Gradient Boosting (sklearn.ensemble. GradientBoostingClassifier)	Default
318	skopt.dummy_minimize (Busca aleatória, 100 calls)	0,673	PCA(40 componentes) + XGBoost (xgboost.XGBClassifier)	learning_rate = 0,00231, n_estimators = 268, max_depth = 3, min_child_weight = 9, colsample_bynode = 0,719572, num_parallel_tree = 1

Tabela 32 – Resultados de experimentos adicionais com *datasets* com 36 *features* após HPO.

Features	**Técnica**	**Acurácia (LOSO)**	**Classificador**	**Hiperparâmetros**
36	skopt.dummy_minimize (Busca aleatória, 500 calls)	**0,8634**	XGBoost (xgboost.XGBClassifier)	learning_rate = 0,0059441870224408765, n_estimators = 2045, max_depth = 6, min_child_weight = 12, subsample = 0,5054952088175283, colsample_bynode = 0,12397857582004676, num_parallel_tree = 6, gamma = 0,3466421353313136, colsample_bytree = 0,8469003238194956, tree_method = 'gpu_hist', random_state = 2023
36	XGBClassifier (sem dask)	0,862	XGBoost (xgboost.XGBClassifier)	learning_rate = 0,02504, n_estimators = 2238, max_depth = 12, min_child_weight = 11, subsample = 0,70774, colsample_bynode = 0,30457, num_parallel_tree = 2, gamma = 1,92242, colsample_bytree = 0,61318
36	XGBClassifier (sem dask)	0,859	XGBoost (xgboost.XGBClassifier)	tree_method = 'gpu_hist', seed = 2022

Tabela 33 – Resultados de experimentos adicionais com *datasets* com 79 *features* após HPO.

Features	**Técnica**	**Acurácia (LOSO)**	**Classificador**	**Hiperparâmetros**
79	skopt.dummy_minimize (Busca aleatória, 500 calls)	**0,8678**	XGBoost (xgboost.XGBClassifier)	learning_rate = 0,023948526542341687, n_estimators = 994, max_depth = 10, min_child_weight = 7, subsample = 0,5648725086828695, colsample_bynode = 0,7304797653575524, num_parallel_tree = 2, gamma = 3,5682439120622123, colsample_bytree = 0,9228165887330854, tree_method = 'gpu_hist', random_state = 2023
79	XGBClassifier (sem dask)	0,860	XGBoost (xgboost.XGBClassifier)	learning_rate = 0,02504, n_estimators = 2238, max_depth = 12, min_child_weight = 11, subsample = 0,70774, colsample_bynode = 0,30457, num_parallel_tree = 2, gamma = 1,92242, colsample_bytree = 0,61318
79	XGBClassifier (sem dask)	0,854	XGBoost (xgboost.XGBClassifier)	tree_method = 'gpu_hist', seed = 2022