

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**ROGÉRIO VICTORIO FERNANDES
BALLESTRIN**

**AVALIAÇÃO DE MODELOS DE
PROPAGAÇÃO INTELIGENTES EM UM
SIMULADOR DE REDE LORA**

Porto Alegre
2024

**ROGÉRIO VICTORIO FERNANDES
BALLESTRIN**

**AVALIAÇÃO DE MODELOS DE
PROPAGAÇÃO INTELIGENTES EM UM
SIMULADOR DE REDE LORA**

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Rio Grande do Sul como parte dos requisitos para a obtenção do título de Mestre em Engenharia Elétrica.
Área de concentração: Controle e Automação

ORIENTADOR: Prof. Dr. Ivan Müller

Porto Alegre
2024

**ROGÉRIO VICTORIO FERNANDES
BALLESTRIN**

**AVALIAÇÃO DE MODELOS DE
PROPAGAÇÃO INTELIGENTES EM UM
SIMULADOR DE REDE LORA**

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Engenharia Elétrica e aprovada em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: _____

Prof. Dr. Ivan Müller, UFRGS

Doutor pela Universidade Federal do Rio Grande do Sul –
Porto Alegre, Brasil

Banca Examinadora:

Prof. Dr. André Lima Ferrer de Almeida, UFC

Doutor pela Universidade Federal do Ceará – Fortaleza, Brasil

Prof. Dr. Wilson Gavião Neto, ISI em Sistemas de Sensoriamento

Doutor pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil

Prof. Dr. Tiago Oliveira Weber, UFRGS

Doutor pela Universidade de São Paulo – São Paulo, Brasil

Coordenador do PPGEE: _____

Prof. Dr. Jeferson Vieira Flores

Porto Alegre, setembro de 2024.

AGRADECIMENTOS

A conclusão deste trabalho não seria possível sem o apoio e a colaboração de muitas pessoas, a quem sou profundamente grato. Primeiramente, agradeço à minha mãe, Cleia, pelo amor incondicional e pelo incentivo constante ao longo de toda a minha jornada acadêmica. À minha companheira de vida, Tatiane, pelo carinho, paciência e compreensão durante os momentos de ausência, e pelas palavras de encorajamento quando mais precisei.

Agradeço ao meu orientador, Prof. Ivan Müller, pelo suporte técnico, orientação e valiosas contribuições durante todo o desenvolvimento desta dissertação. Ao Instituto SENAI de Inovação em Sistemas de Sensoriamento, onde trabalho, pela oportunidade de crescimento profissional e pelo apoio durante minha jornada acadêmica. Em especial, agradeço ao Vitor Nardelli, por sua compreensão e flexibilidade, e ao André Thomazoni, pela colaboração, companheirismo e valiosas discussões.

Aos colegas do LACOM, em especial Max Feldman e Jonas Feijó, pela troca de conhecimentos, ideias e pelo apoio ao longo da jornada. À UFRGS e ao PPGE, pela excelente formação e pelo suporte acadêmico oferecido.

RESUMO

A evolução das redes de comunicação sem fio tem sido um tópico relevante nas últimas décadas, impulsionada pela crescente demanda por aplicações de Internet das Coisas (IoT) em ambientes urbanos e rurais. Essas aplicações abrangem desde o monitoramento remoto até a automação industrial, agricultura de precisão e cidades inteligentes. As redes de sensores sem fio, especialmente aquelas baseadas na tecnologia LoRa, destacam-se por sua capacidade de fornecer comunicações de longo alcance com baixo consumo de energia. No entanto, um dos principais desafios enfrentados no uso dessas redes é a perda de percurso (*path loss*), que afeta diretamente a capacidade de comunicação e o consumo de energia dos dispositivos. Em ambientes urbanos densamente povoados, obstáculos como edifícios altos e estruturas metálicas podem causar atenuação significativa do sinal. Já em áreas rurais, a propagação pode ser afetada por terrenos acidentados ou vegetação densa. Embora existam vários modelos de propagação para prever a perda de percurso, muitos não conseguem lidar com a complexidade dos ambientes reais. Nesta pesquisa, foram implementados e testados modelos clássicos e de Inteligência Artificial usando dois conjuntos de dados: MCLAB, contendo dados de nós móveis, e Medellín, contendo dados de nós fixos. Os melhores modelos identificados foram aqueles derivados da árvore de decisão, como a Floresta Aleatória e o XGBOOST. No conjunto MCLAB, a Floresta Aleatória obteve erro médio quadrático de 6,82 dB, comparado a 9,78 dB do modelo *log-distance*. Já no conjunto Medellín, a Floresta Aleatória apresentou erro médio quadrático de 1,89 dB, comparado a 2,00 dB do modelo *log-distance*. Apesar do significativo esforço computacional na busca pela melhor topologia, nenhuma rede neural superou os modelos baseados em árvore. Todos os modelos aplicados e desenvolvidos foram implementados no simulador LoRaEnergySim, que foi utilizado para avaliar o impacto dos diferentes modelos de propagação no dimensionamento de redes LoRa, confirmando que os modelos clássicos tendem a superestimar as perdas de propagação, o que pode resultar em um dimensionamento incorreto das redes.

Palavras-chave: Perda por propagação, LoRa, Simulador de Rede, Inteligência Artificial.

ABSTRACT

The evolution of wireless communication networks has been a significant topic in recent decades, driven by the growing demand for Internet of Things (IoT) applications in urban and rural environments. These applications range from remote monitoring to industrial automation, precision agriculture, and smart cities. Wireless sensor networks, especially those based on LoRa technology, stand out for their ability to provide long-range communications with low energy consumption. However, one of the main challenges faced by these networks is path loss, which directly affects the communication capacity and energy consumption of devices. In densely populated urban environments, obstacles such as tall buildings and metallic structures can cause significant signal attenuation. In rural areas, signal propagation can be affected by rugged terrain or dense vegetation. Although there are several propagation models to predict path loss, many fail to handle the complexity of real-world environments. In this research, classical models and artificial intelligence models were implemented and tested using two datasets: MCLAB, containing data from mobile nodes, and Medellín, containing data from fixed nodes. The best models identified were those derived from decision trees, such as Random Forest and XGBOOST. In the MCLAB dataset, the Random Forest model achieved an RMSE of 6.82 dB, compared to 9.78 dB for the log-distance model. In the Medellín dataset, the Random Forest model achieved an RMSE of 1.89 dB, compared to 2.00 dB for the log-distance model. Despite the significant computational effort in searching for the best topology, no neural network outperformed the tree-based models. All applied and developed models were implemented in the LoRaEnergySim simulator, which was used to evaluate the impact of different propagation models on the sizing of LoRa networks, confirming that classical models tend to overestimate path losses, which can lead to incorrect network sizing.

Keywords: Path loss, LoRa, Network Simulator, Artificial Intelligence.

LISTA DE ILUSTRAÇÕES

Figura 1 –	Exemplo de 4 símbolos de 2 bits transmitidos com CSS.	17
Figura 2 –	Exemplo de <i>up-chirp</i> com largura de banda de 125 kHz e diferentes fatores de espalhamento.	18
Figura 3 –	Exemplo de pacote de uplink LoRa.	19
Figura 4 –	Palavras-chave mais encontradas na pesquisa.	32
Figura 5 –	Evolução das publicações por país por ano.	32
Figura 6 –	Histograma das variáveis de entrada do <i>dataset</i> Mclab e da perda por propagação.	45
Figura 7 –	Arquitetura da rede desenvolvida por GONZÁLEZ-PALACIO <i>et al.</i> (2022).	46
Figura 8 –	Histograma das variáveis de entrada do <i>dataset</i> Medellín e da perda por propagação.	47
Figura 9 –	Histograma das demais variáveis presentes no dataset Medellín . . .	48
Figura 10 –	Fluxo de dados do simulador LoRaEnergySim.	49
Figura 11 –	Modelo de propagação de caminho livre aplicado ao <i>dataset</i> MCLAB.	54
Figura 12 –	Modelo de propagação de Egli aplicado ao <i>dataset</i> MCLAB.	55
Figura 13 –	Modelo de propagação de Okumura-Hata aplicado ao <i>dataset</i> MCLAB.	55
Figura 14 –	Modelo de propagação COST231-HATA aplicado ao <i>dataset</i> MCLAB.	56
Figura 15 –	Modelo de propagação log-distance aplicado ao <i>dataset</i> MCLAB.	57
Figura 16 –	Avaliação do hiperparâmetro profundidade para árvore de decisão.	58
Figura 17 –	Predição de perda do caminho utilizando árvore de decisão para regressão no <i>dataset</i> MCLAB.	58
Figura 18 –	Avaliação dos hiperparâmetros profundidade e estimadores para a floresta aleatória.	59
Figura 19 –	Predição de perda do caminho utilizando floresta aleatória para regressão no <i>dataset</i> MCLAB.	60
Figura 20 –	Predição de perda do caminho utilizando SVR no <i>dataset</i> MCLAB.	61
Figura 21 –	Avaliação do hiperparâmetro alpha para a Regressão Lasso.	61
Figura 22 –	Predição de perda do caminho utilizando regressão Lasso no <i>dataset</i> MCLAB.	62
Figura 23 –	Avaliação dos hiperparâmetros profundidade e estimadores para o XGBoost.	63
Figura 24 –	Predição de perda do caminho utilizando XGBoost no <i>dataset</i> MCLAB.	63
Figura 25 –	Topologia com camada com nós variáveis explorada como hiperparâmetro.	64
Figura 26 –	Predição de perda do caminho utilizando Rede Neural no <i>dataset</i> MCLAB.	64

Figura 27 –	Análise de componentes principais para o <i>dataset</i> Medellín.	65
Figura 28 –	Sensibilidade da perda de propagação em função das variáveis relacionadas a condição ambiental para o <i>dataset</i> Medellín.	66
Figura 29 –	Sensibilidade da perda de propagação em função das variáveis relacionadas a posição para o <i>dataset</i> Medellín.	66
Figura 30 –	Sensibilidade da perda de propagação em função das variáveis relacionadas a transmissão para o <i>dataset</i> Medellín.	67
Figura 31 –	Modelo de propagação de caminho livre aplicado ao <i>dataset</i> Medellín.	67
Figura 32 –	Modelo de propagação de Egli aplicado ao <i>dataset</i> Medellín.	68
Figura 33 –	Avaliação dos hiperparâmetros profundidade e estimadores para a floresta aleatória com todos dados e com os fatores significativos.	70
Figura 34 –	Predição de perda do caminho utilizando floresta aleatória para regressão no <i>dataset</i> Medellín.	71
Figura 35 –	Predição de perda do caminho utilizando SVR no <i>dataset</i> Medellín.	72
Figura 36 –	Avaliação do hiperparâmetro alpha para a Regressão Lasso.	72
Figura 37 –	Predição de perda do caminho utilizando regressão Lasso no <i>dataset</i> Medellín.	73
Figura 38 –	Avaliação dos hiperparâmetros profundidade e estimadores para o XGBoost.	74
Figura 39 –	Predição de perda do caminho utilizando XGBoost no <i>dataset</i> Medellín.	74
Figura 40 –	Topologia com camada com nós variáveis com todos dados de entrada explorada como hiperparâmetro no <i>dataset</i> Medellín.	75
Figura 41 –	Rede de 10 nós gerada para teste do simulador.	79
Figura 42 –	Rede de 100 nós gerada para teste do simulador.	79
Figura 43 –	Rede de 200 nós gerada para teste do simulador.	80
Figura 44 –	Comparação do tempo de processamento com 1, 10 e 100 nós em uma rede.	80
Figura 45 –	Comparação da métrica de desempenho criada para todos os modelos.	81
Figura 46 –	Comparação da perda para diferentes modelos de propagação até 10 km.	82
Figura 47 –	Comparação da quantidade de pacotes recebidos para diferentes modelos de propagação em diferentes fatores de espalhamento.	85
Figura 48 –	Comparação da quantidade de pacotes recebidos para diferentes modelos de propagação em diferentes tamanhos de célula.	87
Figura 49 –	Comparação da energia consumida para diferentes modelos de propagação em na célula 10.000 x 10.000 m.	88
Figura 50 –	Comparação da quantidade de pacotes recebidos para diferentes quantidade de nós em células do mesmo tamanho.	89

LISTA DE TABELAS

Tabela 1 –	Tempo de Transmissão de um símbolo para largura de banda de 125 kHz	18
Tabela 2 –	Condições dos Ensaios dos Trabalhos Publicados.	41
Tabela 3 –	Contribuições dos Trabalhos Publicados.	42
Tabela 4 –	Hiperparâmetros explorados para definição da rede neural.	51
Tabela 5 –	RMSE para diferentes combinações de variáveis de entrada do <i>dataset</i> Medellín para árvore de decisão.	69
Tabela 6 –	Comparação entre Algoritmos com os <i>datasets</i> MCLAB e Medellín. .	78

LISTA DE ABREVIATURAS

ADR	<i>Adaptive Data Rate</i>
AG2UG	<i>Aboveground-to-Underground</i>
ASTER	<i>Advanced Spaceborne Thermal Emission and Reflection Radiometer</i>
BKG	<i>Bundesamt für Kartographie und Geodäsie</i>
BW	<i>Bandwidth</i>
CR	<i>Code Rate</i>
CSS	<i>Chirp Spread Spectrum</i>
CRC	<i>Cyclic Redundancy Check</i>
EN	<i>End Node</i>
EU-DEM	<i>European Digital Elevation Model</i>
EVRS2000	<i>European Vertical Reference System 2000</i>
GPS	<i>Global Positioning System</i>
GW	<i>Gateway</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IoT	<i>Internet of Things</i>
JSON	<i>JavaScript Object Notation</i>
LoRa	<i>Long Range</i>
LoRaWan	<i>Long Range Wide Area Network</i>
LOS	<i>Line-of-Sight</i>
MAC	<i>Medium Access Control</i>
MAE	<i>Mean Absolute Error</i>
NLOS	<i>Non-line-of-Sight</i>
PCA	<i>Principal Component Analysis</i>
PHY	<i>Physical Layer</i>
PPGEE	<i>Programa de Pós-Graduação em Engenharia Elétrica</i>
RBF	<i>Radial Basis Function</i>

ReLU	<i>Rectified Linear Unit</i>
RMSE	<i>Root Mean Square Error</i>
RSSI	<i>Received Signal Strength Indication</i>
SF	<i>Spreading Factor</i>
SRTM	<i>Shuttle Radar Topography Mission</i>
SVM	<i>Support Vector Machine</i>
SVR	<i>Support Vector Regression</i>
ToA	<i>Time on Air</i>
UG2AG	<i>Underground-to-Aboveground</i>

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Motivação	14
1.2	Objetivos	14
1.2.1	Publicação	15
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	Redes LoRa	16
2.1.1	Pacote LoRa	19
2.2	Perdas por propagação em redes sem fio	19
2.3	Modelos de propagação	20
2.3.1	Perda por caminho em espaço livre	20
2.3.2	Medições de Young e o modelo de Egli	21
2.3.3	<i>Log-Distance</i>	22
2.3.4	Modelo de Okumura-Hata	23
2.3.5	Modelo COST 231-Hata	25
2.4	Inteligência Artificial - Abordagens para Regressão	25
2.4.1	Modelos baseados em árvore de decisão	26
2.4.2	<i>Support Vector Regression</i>	27
2.4.3	Regressão Lasso	28
2.4.4	Rede neural	29
3	REVISÃO DA LITERATURA	31
3.1	Propagação de redes LoRa e LoRaWAN	31
3.1.1	Análise bibliométrica	31
3.1.2	Estado da arte	33
3.2	Uso de inteligência artificial para modelos de propagação de redes sem fio	37
3.3	Simuladores de redes LoRa	38
4	MÉTODOS E MATERIAIS	43
4.1	<i>Datasets</i>	43
4.1.1	<i>Dataset 1: MCLab - Nós móveis</i>	43
4.1.2	<i>Dataset 2: Medellín - Nós fixos</i>	46
4.2	Simulador LoRaEnergySim	47
4.3	Método para treinamento de modelos de predição que usam Inteligência Artificial	50
4.4	Ensaio com simulador	52

5	IMPLEMENTAÇÃO	53
5.1	Modelos clássicos e com IA para <i>dataset</i> MCLAB	53
5.1.1	Espaço Livre	53
5.1.2	Modelo de Egli	53
5.1.3	Okumura-Hata	54
5.1.4	COST231-Hata	55
5.1.5	<i>Log-Distance</i>	56
5.1.6	Árvore de Decisão para regressão	57
5.1.7	Floresta Aleatória para Regressão	59
5.1.8	<i>Support Vector Regression</i>	59
5.1.9	Regressão Lasso	60
5.1.10	XGBoost	62
5.1.11	Rede neural	62
5.2	Modelos clássicos e com IA para <i>dataset</i> Medellín	64
5.2.1	Fatores significativos para predição de perda por propagação	64
5.2.2	Modelos clássicos	66
5.2.3	Árvore de Decisão para regressão	68
5.2.4	Floresta Aleatória para regressão	70
5.2.5	<i>Support Vector Regression</i>	70
5.2.6	Regressão Lasso	71
5.2.7	XGBoost	73
5.2.8	Rede neural	73
5.3	Implementação dos algoritmos no simulador LoraEnergySim	76
6	ANÁLISE DE RESULTADOS	77
6.1	Comparação de resultados dos algoritmos implementados	77
6.2	Redes geradas para testes do simulador	78
6.3	Avaliação do tempo de processamento	79
6.4	Comparação de resultados simulados	82
6.4.1	Perda de propagação com diferentes modelos	82
6.4.2	Avaliação do impacto do modelo de propagação em função do fator de espalhamento	83
6.4.3	Avaliação do impacto do modelo de propagação em função do tamanho da célula	85
6.4.4	Avaliação do impacto do modelo de propagação em função da quantidade de nós	87
7	CONCLUSÃO	90
	REFERÊNCIAS	92

1 INTRODUÇÃO

A evolução das redes de comunicação sem fio tem sido um tópico de pesquisa significativo nas últimas décadas. A demanda por aplicações de IoT em ambientes urbanos e rurais está crescendo exponencialmente (ALLIOUI; MOURDI, 2023), impulsionada pela necessidade de monitoramento remoto, automação industrial, agricultura de precisão e cidades inteligentes. No entanto, para viabilizar essas aplicações em larga escala, é essencial contar com redes de sensores sem fio eficientes e confiáveis, capazes de lidar com os desafios únicos apresentados por diferentes contextos de implantação. As redes de sensores sem fio (*Wireless Sensor Networks*, WSN), como as baseadas na tecnologia LoRa (*Long Range*), têm ganhado destaque devido à sua capacidade de proporcionar comunicações de longo alcance e baixo consumo de energia.

Um dos principais desafios enfrentados pelas redes de sensores sem fio, como aquelas baseadas na tecnologia LoRa, é a perda de percurso (*path loss*), que impacta diretamente a capacidade de comunicação e o consumo de energia dos dispositivos. Em ambientes urbanos densamente povoados, obstáculos como edifícios altos e estruturas metálicas podem causar atenuação significativa do sinal, reduzindo o alcance dessas redes. Da mesma forma, em áreas rurais amplas, a propagação do sinal pode ser afetada por terrenos acidentados ou vegetação densa.

O correto dimensionamento da infraestrutura das redes sem fio depende significativamente da compreensão da propagação do sinal de rádio em diferentes ambientes. Embora existam vários modelos de propagação para prever a perda de percurso, muitos deles não conseguem lidar com a complexidade e a variabilidade dos ambientes de propagação na vida real. Compreender e mitigar esses efeitos é essencial para garantir o desempenho consistente dessas redes.

O estudo da perda por propagação tem papel importante não apenas para a modelagem de redes urbanas em larga escala, mas também desempenha um papel significativo na estimativa da localização de um dispositivo. Conforme FARAHSAARI *et al.* (2022), a abordagem que se baseia na perda de sinal por propagação é a mais direta e amplamente aplicada em cenários de localização de dispositivos sem fio. Esta metodologia depende da estimativa da distância entre os nós de referência, que podem ser gateways de uma WSN,

e os nós móveis para determinar a localização. Comumente, essa distância é modelada pela equação *log-distance*, que é uma forma generalizada da equação de Friis e serve de entrada para algoritmos como os de localização por trilateração.

Melhorar a compreensão da propagação do sinal não apenas aprimora a eficiência energética e a confiabilidade das comunicações, mas também abre caminho para avanços significativos na aplicação prática da IoT. Nesse contexto, os modelos de Inteligência Artificial (IA) surgem como uma solução promissora, capazes de aprender a partir de dados empíricos e adaptar-se a diferentes cenários de propagação. A motivação central desta pesquisa é explorar e aprimorar os modelos de propagação utilizados em redes LoRa, a fim de aprimorar o projeto e o desempenho dessas redes.

1.1 Motivação

A crescente implementação de dispositivos conectados em diferentes aplicações está moldando novas expectativas para a conectividade sem fio. Contudo, desafios como a propagação irregular do sinal em WSN podem dificultar a implementação dessas tecnologias em larga escala. Esta pesquisa se propõe a atuar em uma parte deste desafio, aprimorando os modelos de propagação usados em redes LoRa. Almeja-se otimizar o projeto, e, por consequência, o desempenho de redes IoT concebidas para operar em ambientes urbanos, integrando modelos de IA. Desta forma, objetiva-se não somente melhorar o desempenho da predição de perdas, mas também permitir que novas tecnologias e implementações possam ser testadas em simuladores de rede LoRa mais fidedignos.

1.2 Objetivos

O objetivo principal desta dissertação de mestrado é investigar e identificar o melhor modelo de propagação para redes LoRa urbanas, visando aprimorar os simuladores de redes sem fio utilizados para dimensionamento da infraestrutura necessária. Para tanto, os seguintes objetivos específicos são detalhados:

- **Avaliação de Modelos de Propagação:** Utilizar dados de perda de propagação de redes LoRa para aplicar diferentes modelos de propagação amplamente reconhecidos, como o modelo *log-distance*, o modelo COST 231 Hata, o modelo de Okumura-Hata, entre outros. Comparam-se os resultados dos modelos aplicados com os dados reais para avaliar seu desempenho em prever a perda de percurso em ambientes urbanos densos.
- **Desenvolver Modelos de Propagação baseados em inteligência artificial:** Treinar modelos de perda de propagação de redes LoRa utilizando algoritmos de inteligência artificial, explorando os hiperparâmetros para otimização dos modelos.

- Aprimoramento dos Simuladores: Integrar os modelos de propagação selecionados no simulador de redes sem fio já existente. Verificar e validar os resultados do simulador após a implementação dos novos modelos, realizando comparações entre os modelos de propagação.

A estrutura deste trabalho foi organizada com intuito de proporcionar a compreensão progressiva dos principais elementos abordados. Iniciando pelo Capítulo 2, são introduzidos os conceitos fundamentais que sustentam a dissertação. No Capítulo 3, a revisão bibliográfica é apresentada, resumindo os trabalhos prévios considerados mais relevantes. No Capítulo 4, são detalhados os métodos empregados, bem como os materiais utilizados durante o desenvolvimento deste estudo. O Capítulo 5 é dedicado à exposição da implementação do sistema proposto. Os ensaios conduzidos, acompanhados dos resultados obtidos são apresentados e analisados no Capítulo 6. Finalmente, no Capítulo 7, são apresentadas as conclusões derivadas da pesquisa, além de sugerir direções para futuros desenvolvimentos nesta área de estudo.

1.2.1 Publicação

Os resultados do primeiro objetivo específico deste trabalho foram publicados no artigo intitulado *Exploring Machine Learning Techniques for Path Loss Prediction in LoRa Networks* (BALLESTRIN *et al.*, 2024), apresentado no *19th International Symposium on Wireless Communication Systems (ISWCS)*. Neste artigo, foi explorada a aplicação de técnicas de aprendizado de máquina para a predição de perda de caminho em redes LoRa. A publicação está disponível online e pode ser acessada pelo IEEE Xplore.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção, são apresentados os principais conceitos e modelos teóricos aplicados no desenvolvimento deste trabalho. Inicialmente, a tecnologia LoRa é introduzida, tendo um papel central na pesquisa, incluindo uma análise do pacote LoRa e suas características. Em seguida, são discutidos os principais aspectos relacionados à propagação de sinais em redes sem fio, com ênfase nas perdas por propagação, que influenciam diretamente o desempenho dessas redes. Para uma melhor compreensão dessas perdas, são apresentados modelos de propagação amplamente utilizados na literatura. Por fim, são discutidos alguns modelos de inteligência artificial que podem ser utilizados para a predição das perdas por propagação.

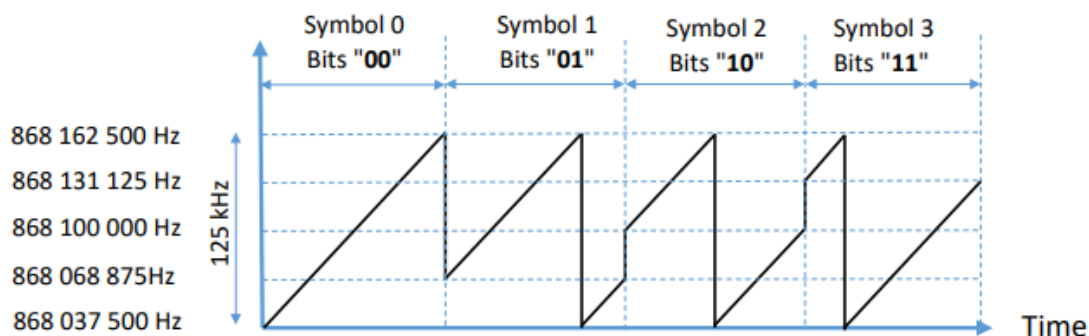
2.1 Redes LoRa

LoRa é uma tecnologia de comunicação sem fio desenvolvido pela empresa *Semtech*, amplamente utilizada para conectar dispositivos na Internet das Coisas (IoT). Sua capacidade de oferecer comunicação de longo alcance com baixo consumo de energia a torna uma escolha atraente para aplicações IoT em áreas como agricultura digital, cidades inteligentes e redes de sensores sem fio em geral.

A tecnologia LoRa utiliza a técnica de modulação de sinais *Chirp Spread Spectrum* (CSS). O CSS utiliza a variação da frequência de transmissão de forma linear ao longo do tempo, criando um "chirp" (em inglês, *chirp*) que se estende por uma ampla faixa de frequência. Um pulso *chirp* é um tipo de sinal cuja frequência muda de forma contínua e gradual ao longo do tempo. Quando essa mudança de frequência é ascendente, é denominada *up-chirp*, e quando é descendente, é chamada de *down-chirp*. Além disso, os *chirps* têm a característica de ciclicamente retornar à frequência inicial após atingir a frequência mais alta. A informação que um sinal *chirp* carrega está relacionada aos padrões dessas mudanças de frequência, ou seja, aos momentos em que ocorrem essas transições entre frequências. Um exemplo de uma transmissão de um símbolo que carrega 2 bits pode ser visto na Figura 1.

Uma característica importante do LoRa é a sua flexibilidade para ajustar a largura de

Figura 1 – Exemplo de 4 símbolos de 2 bits transmitidos com CSS.



Fonte: MONTAGNY (2022)

banda do sinal, permitindo uma adaptação da taxa de transferência de dados e do alcance da comunicação. A largura de banda (BW) define o intervalo de frequências que os transceptores utilizarão para transmitir mensagens, influenciando diretamente a capacidade de transmissão de dados por segundo. O módulo SX1272 da *Semtech*, amplamente empregado em redes LoRa, disponibiliza três opções de largura de banda configuráveis: 125 kHz, 250 kHz e 500 kHz, atendendo tanto às necessidades de transmissão (uplink) quanto às de recepção (downlink) (SEMTECH CORPORATION, 2019).

Outro parâmetro que pode ser escolhido para a transmissão de dados é o fator de espalhamento (*spreading factor*, SF). O SF é a taxa em que a frequência do sinal está variando, ou seja, é um espalhamento no tempo e não na frequência, mas também pode ser visto como a quantidade de bits que um *chirp* possui. Por exemplo, ao transmitir em SF10, está sendo transmitido um símbolo de 10 bits, que pode assumir 2^{10} valores binários diferentes. Mantendo a largura de banda constante, à medida que o valor de SF aumenta, o tempo de transmissão também aumenta, como pode ser visto na Tabela 1. Ao mesmo tempo, a probabilidade de que os sinais LoRa possam ser detectados e recebidos aumenta, e a distância de transmissão também aumenta (GUO; YANG; WEI, 2021). A Figura 2 mostra um *up-chirp* LoRa modulado em uma banda de 125 kHz, em todos os fatores de espalhamento entre 7 e 12 (KIM; LEE; KIM, 2019).

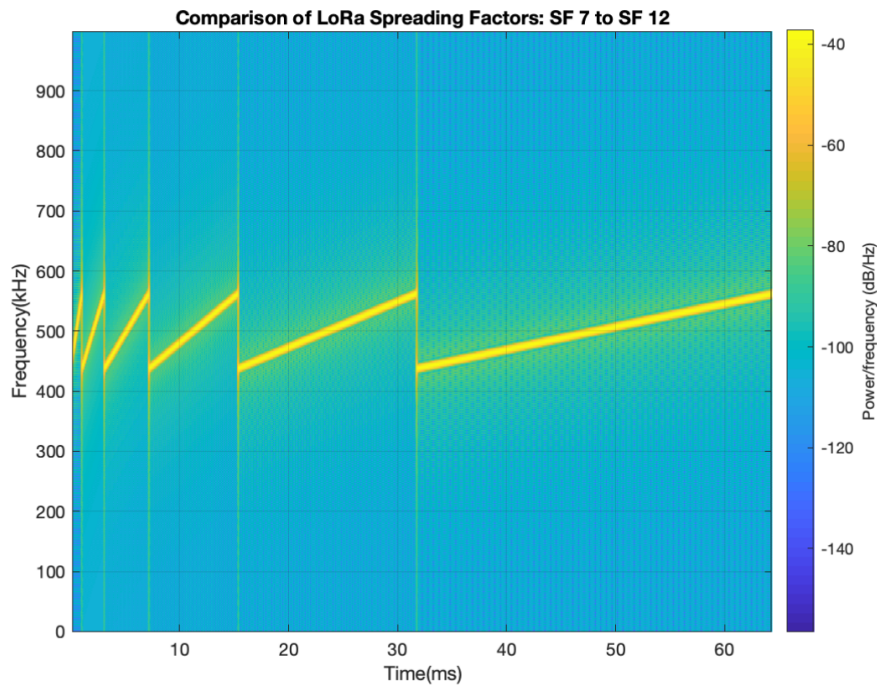
Por fim, o último parâmetro que pode ser trabalhado é a taxa de codificação (*Coding Rate*, CR). O CR se refere à relação entre a quantidade de dados úteis (*payload*) e a quantidade total de dados transmitidos, incluindo redundância e correção de erros, e é expressa como uma fração, onde o numerador representa o *payload* e o denominador representa o total de dados transmitidos (*payload* mais redundância). Uma vez que o CR apenas adiciona redundância, não há impacto na modulação, mas sim na taxa de bits efetiva sendo transmitida. Considerando todos os parâmetros configuráveis, a taxa de transmissão de dados efetiva em bits por segundo (bps) é dada por:

Tabela 1 – Tempo de Transmissão de um símbolo para largura de banda de 125 kHz

Fator de Espalhamento	Tempo de transmissão de Símbolo
SF7	1.024 ms
SF8	2.048 ms
SF9	4.096 ms
SF10	8.192 ms
SF11	16.384 ms
SF12	32.768 ms

Fonte: MONTAGNY (2022).

Figura 2 – Exemplo de *up-chirp* com largura de banda de 125 kHz e diferentes fatores de espalhamento.



Fonte: KIM; LEE; KIM (2019)

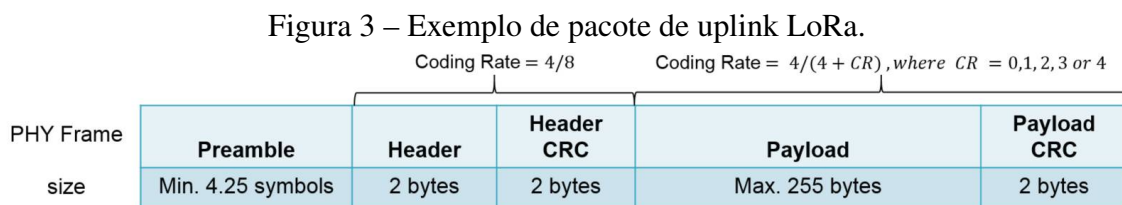
$$BitRate = SF \frac{BW}{2^{SF}} CR \quad (1)$$

A técnica de modulação CSS aliada à esses diferentes parâmetros ajustáveis permite que o sinal LoRa seja altamente resistente a interferências e ruídos, tornando-o adequado para ambientes urbanos, pois é resistente a fenômenos de multicaminho, desvanecimento, Doppler e *jamming*. Por outro lado, acaba utilizando uma largura de banda muito maior do que a necessária para transmitir a informação (RABAÇA, 2018).

2.1.1 Pacote LoRa

Um pacote LoRa é composto por vários campos, incluindo preâmbulo, cabeçalho (*header*), *payload* e um código de correção de erros. Um pacote LoRa começa com um preâmbulo que além de funcionar como função de sincronização, define também o esquema de modulação do pacote, sendo modulado com o mesmo fator de espalhamento que o restante do pacote. O preâmbulo é usado para sincronizar o receptor com o transmissor e pode ser configurado para valores entre 6 e 65535 símbolos, sendo o padrão 8 símbolos (PHAM; EHSAN, 2021). Além do parametrizado, o transmissor de rádio adicionará mais 4,25 símbolos.

O preâmbulo é seguido por um cabeçalho PHY. Este cabeçalho pode ser configurado para o modo implícito ou explícito. Nos casos onde o comprimento do payload, a taxa de codificação de erros e o CRC são fixos ou conhecidos, pode-se reduzir o tempo de transmissão chamando o modo de cabeçalho implícito. Já no modo explícito, o cabeçalho contém informações de comprimento do payload, *code rate* da mensagem e um CRC de cabeçalho que, ao total, possuem 20 bits de comprimento e são codificados com o *code rate* mais robusto (4/8), enquanto o restante do quadro é codificado com a taxa de código especificada no cabeçalho (LORA ALLIANCE, 2019). Um *uplink* lora com CRC de *payload* é exemplificado na Figura 3.



Fonte: PHAM; EHSAN (2021)

2.2 Perdas por propagação em redes sem fio

As ondas eletromagnéticas apresentam características específicas que influenciam a forma como se propagam no espaço. A frequência representa o número de ciclos da onda por segundo, enquanto o comprimento de onda está relacionado à frequência e representa a distância entre dois pontos correspondentes em uma onda. A propagação de ondas de rádio pode ocorrer em duas condições: linha de visada (*Line-of-Sight*, LOS) e não linha de visada (*non-Line-of-Sight*, NLOS). No LOS, há uma linha direta e desobstruída entre o transmissor e o receptor, sem obstruções significativas entre eles, cenário que geralmente resulta em uma propagação com menor atenuação do sinal. Já no NLOS, obstáculos físicos como edifícios, vegetação densa ou terreno acidentado bloqueiam ou refletem o sinal, causando maior atenuação, dispersão e propagação múltipla de sinais (múltiplos caminhos).

A atenuação é a redução da intensidade do sinal à medida que ele se propaga no ambiente. Vários fatores contribuem para a atenuação do sinal de rádio, incluindo a atenuação de percurso (*path loss*), que é a dispersão do sinal em todas as direções conforme se propaga, resultando em diminuição da potência recebida conforme a distância do transmissor. Além dos obstáculos entre o transmissor e o receptor, condições atmosféricas e outros elementos contribuem na atenuação do sinal.

Além da atenuação, outros fenômenos influenciam a propagação de ondas de rádio. A reflexão ocorre quando o sinal reflete em superfícies sólidas, como paredes e pisos, criando múltiplos caminhos de propagação e interferências, que podem ser construtivas ou destrutivas. A difração é outro efeito importante, onde as ondas de rádio podem alterar sua trajetória ao contornar obstáculos, permitindo ocasionalmente que o sinal alcance áreas sombreadas. Por fim, o sombreamento é causado por objetos que criam zonas onde o sinal não chega, resultando em áreas com cobertura deficiente.

2.3 Modelos de propagação

2.3.1 Perda por caminho em espaço livre

As perdas de propagação em espaço livre referem-se à diminuição da intensidade do sinal de rádio à medida que ele se afasta da fonte transmissora e se propaga pelo espaço sem obstruções. Essas perdas ocorrem devido à expansão em forma esférica do sinal, que resulta em uma distribuição de energia em uma área cada vez maior à medida que a distância aumenta. O modelo matemático básico para calcular as perdas de propagação em espaço livre é conhecido como a Lei do Inverso do Quadrado da Distância. A equação geral considerando o ganho das antenas é chamada de Equação de Friis e é dada por (RAPPAPORT, 2001):

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L} \quad (2)$$

Onde:

- P_r é a potência recebida do sinal em watts (W).
- P_t é a potência transmitida do sinal em watts (W).
- G_t é o ganho da antena transmissora.
- G_r é o ganho da antena receptora.
- λ é o comprimento de onda do sinal em metros (m).
- d é a distância entre a antena transmissora e a antena receptora em metros (m).

- L é um fator para as perdas não relacionadas a transmissão, como filtros, cabos e conexões.

Essa equação descreve como a potência do sinal diminui à medida que a distância d aumenta devido à expansão esférica da onda eletromagnética. Quanto maior a distância, menor será a potência recebida. A equação assume condições ideais, ou seja, o vácuo do espaço livre de obstáculos, não considerando efeitos como reflexões, difração ou obstruções no caminho do sinal. Para aplicações práticas, é mais comum encontrar a equação de Friis para calcular a perda por propagação em decibéis (dB), além de considerar o ganho das antenas como unitário. Para este caso simplificado, a equação fica:

$$P_{r,db}(d) = P_{t,db} + 20 \log \frac{\lambda}{4\pi d} \quad (3)$$

2.3.2 Medições de Young e o modelo de Egli

Young realizou medições na cidade de Nova Iorque nas frequências de 150, 450, 950 e 3700 MHz utilizando um veículo em movimento transmitindo dados. Apesar de reportar apenas resultados e traçar algumas hipóteses, os dados de Young reforçam a existência da relação de quarta potência com a distância entre transmissor e receptor do modelo de Terra Plana, adicionando um fator de correção devido às características do meio de propagação utilizado (PARSONS, 2000).

Os dados de Young indicam que é necessário corrigir os modelos em função da topologia da cidade. No experimento realizado, as perdas para distâncias superiores a 10 milhas foram menores do que a tendência encontrada em distâncias menores. A hipótese de Young foi que essas distâncias maiores são medições realizadas com o receptor em áreas suburbanas da cidade de Nova Iorque, que possuíam quantidades menores de edifícios e de alturas menores do que a região mais próxima da antena transmissora (YOUNG, 1952).

Após Young, Egli analisou diferentes medições de ondas propagando na faixa de 90 a 1000 MHz e também encontrou uma tendência semelhante o modelo de Propagação em Terra Plana, onde a potência do sinal decai com a quarta potência da distância entre emissor e transmissor. Um dos pontos importantes do trabalho é demonstrar empiricamente que o modelo de Terra Plana é uma boa aproximação para a mediana dos dados de propagação, mas que há uma distribuição probabilística que depende das irregularidades do terreno. Outra contribuição significativa do modelo de Egli é mostrar que o modelo de Terra Plana, apesar de possuir relações válidas, é uma simplificação válida para frequências abaixo de 40 MHz, sendo necessário corrigir o valor da mediana para frequências superiores. Desta forma, Egli introduz um fator β para corrigir o modelo de terra plana terrenos irregulares (EGLI, 1957), sendo o modelo de perda por propagação definido pela equação:

$$L_{50} = G_t G_r \left(\frac{h_t h_r}{d^2} \right)^2 \beta \quad (4)$$

Com β dado por:

$$\beta = \left(\frac{40}{f} \right)^2 \quad (5)$$

Onde:

- G_t é o ganho da antena transmissora.
- G_r é o ganho da antena receptora.
- h_t é a altura da antena transmissora em metros.
- h_r é a altura da antena receptora em metros.
- f é a frequência do sinal em MHz.
- d é a distância entre a antena transmissora e a antena receptora em metros.

Além de corrigir a mediana, β possui uma distribuição associada. Egli associou o desvio padrão de β às irregularidades do terreno, sendo a equação 5 apenas o valor médio de uma distribuição lognormal apresentada apenas graficamente. As descobertas de Young e Egli serviram de base para modelos que utilizam fatores de correção relacionados às regiões das cidades, à frequência de propagação e também para modelos que consideram distribuições lognormais para as aleatoriedade existente devido à parâmetros complexos de serem modelados, como as irregularidades do terreno.

2.3.3 Log-Distance

O modelo *log-distance* é um dos modelos de propagação mais simples e amplamente utilizados em redes sem fio. Esse modelo é uma abordagem empírica que descreve a perda de potência do sinal em função da distância entre o transmissor e o receptor. O princípio fundamental desse modelo é que a potência do sinal se reduz logaritmicamente à medida que o sinal percorre uma distância maior (RAPPAPORT, 2001).

A equação geral do modelo *log-distance* é dada por:

$$PL(d) = PL(d_0) + 10n \log\left(\frac{d}{d_0}\right) \quad (6)$$

Onde:

- $PL(d)$ é a perda de percurso (Path Loss) na distância em dB;
- $PL(d_0)$ é a perda de percurso em uma distância de referência;
- n é o expoente de perda de percurso, que é uma constante empírica dependente do ambiente e das características do cenário.

O valor de n pode variar de acordo com o ambiente de propagação. Em ambiente aberto, por exemplo, o valor de n é igual a 2, enquanto que em ambientes urbanos densos, por exemplo, com obstáculos e reflexões, o valor de n tende a ser maior (entre 3 a 5). Em ambientes internos e com visada, este valor pode ser menor que 2, devido a reflexões construtivas no sinal. Vale observar que o modelo *log-distance* é uma simplificação e pode não levar em conta alguns efeitos complexos presentes em ambientes reais. Por isso, é essencial calibrar esse modelo com base em medições reais de RSSI (*Received Signal Strength Indicator*) em diferentes locais da rede para encontrar o valor mais adequado de n para o cenário específico de implantação.

2.3.4 Modelo de Okumura-Hata

O modelo de Okumura-Hata é um dos modelos de propagação mais populares utilizados em telecomunicações (SHAKIR *et al.*, 2022). Ele foi desenvolvido por Hata com base em medições realizadas por Okumura em ambiente aberto, urbano e suburbano no Japão, e sua aplicabilidade foi posteriormente estendida para outros tipos de ambientes. O modelo de Okumura, que serviu de base para Hata, leva em consideração a frequência de operação, a distância entre o transmissor e o receptor, a altura das antenas do transmissor e receptor, além de fatores de correção específicos para cada ambiente.

O conceito de modelo de área aberta abrange locais com amplos espaços, onde não há presença de árvores ou prédios altos obstruindo o caminho, e onde a visada se estende limpa por 300 a 400 metros adiante, como em regiões agrícolas. O modelo de área suburbana retrata uma vila ou uma estrada ladeada por árvores e casas, onde alguns obstáculos podem estar próximos ao dispositivo móvel, mas sem causar um congestionamento excessivo. Já o modelo de área urbana é utilizado para uma metrópole densamente construída ou uma cidade grande, onde prédios e casas de dois ou mais andares dominam a paisagem. Pode também representar vilarejos maiores, caracterizados por casas próximas umas das outras e árvores que cresceram até atingirem alturas consideráveis (SEYBOLD, 2005).

Este modelo parte da área aberta como base e, então, aplica fatores de correção para se adaptar a outros meios. Cada um dos fatores de correção é encontrado em um gráfico e utilizado na equação para definir a perda por propagação:

$$PL = L_{fsl} + A_{mu} + H_{tu} + H_{ru} \quad (7)$$

Onde:

- PL é a perda de percurso
- L_{fsl} é a perda de espaço livre para a distância e frequência fornecidas
- A_{mu} é uma função tanto da frequência quanto da distância. É a atenuação mediana em relação à perda de área aberta em uma área urbana, com terreno quase plano, altura efetiva da estação base $h_{(t)e} = 200$ m e altura da antena móvel $h_{(r)e} = 3$ m.

- H_{ru} é o fator de ganho de altura da estação base.
- H_{tu} é o fator de ganho de altura da antena móvel.

Hata equacionou as informações que eram obtidas através dos gráficos de Okumura, criando o modelo de Okumura-Hata. Para cada uma das áreas propostas por Okumura, há uma equação que modela as perdas por propagação. A base para o modelo é a propagação em áreas urbanas e os demais ambientes são corrigidos a partir deste valor. A equação empírica para áreas urbanas é dada por (HATA, 1980):

$$PL_{urbano}(d) = 69.55 + 26.16\log(f_c) - 13.82\log h_t - a(h_r) + [44.9 - 6.55\log(h_t)]\log(d) \quad (8)$$

Onde:

- $PL_{urbano}(d)$ é a perda de percurso na distância em dB;
- f_c é a frequência de operação e deve estar entre 150 e 1500 MHz;
- h_t é a altura da antena do transmissor, em metros, com valores entre 30 e 200 m;
- d é a distância entre transmissor e receptor, em km, com valores entre 1 e 20 km;
- $a(h_r)$ é um fator de correção para a antena do receptor e possui valores diferentes para cidades pequenas a médias e para cidades grandes.

Para cidades pequenas e médias, o fator $a(h_r)$ para h_r entre 1 e 10 m é dado por:

$$a(h_r) = (1.1\log(f_c) - 0.7)h_r - (1.56\log(f_c) - 0.8) \quad (9)$$

No caso de cidades maiores, $a(h_r)$ também é função da frequência, uma vez que o comportamento é diferente para frequências mais baixas:

$$a(h_r) = \begin{cases} 8.29(\log(1.54h_r))^2 - 1.1 & \text{para } f_c \leq 200\text{MHz} \\ 3.2(\log(11.75h_r))^2 - 4.97 & \text{para } f_c \geq 400\text{MHz} \end{cases} \quad (10)$$

Por fim, as equações para áreas suburbanas e áreas abertas é dada por:

$$PL_{suburbano}(d) = PL_{urbano}(d) - 4.78(\log(f_c))^2 + 18.33\log(f_c) - 40.94 \quad (11)$$

$$PL_{aberta}(d) = PL_{urbano}(d) - 2(\log(\frac{f_c}{28}))^2 - 5.4 \quad (12)$$

2.3.5 Modelo COST 231-Hata

Muitos sistemas de comunicação sem fio operam na faixa de frequência de 1500-2000 MHz. Como esperado, foi demonstrado que a perda de sinal pode ser maior nessas frequências do que na faixa de 900 MHz. Alguns estudos sugeriram que a perda de sinal experimentada em 1845 MHz é aproximadamente 10 dB maior do que aquela experimentada em 955 MHz, quando mantidos todos os outros parâmetros constantes (SINGH, 2012). O modelo COST231-Hata estende as bases do modelo original de Hata, com a finalidade de se aplicar também no domínio de frequências de 1500-2000 MHz, onde ocorre uma subestimação da deterioração do sinal pelo modelo. O modelo é expresso pelas equações abaixo:

$$PL(d) = A + B \log(d) + C \quad (13)$$

Onde:

- $PL(d)$ é a perda de percurso na distância em dB;
- $A = 46.3 + 33.9 \log(f_c) - 13.28 \log(h_t) - a(h_r)$;
- $B = 44.9 - 6.55 \log(h_r)$;
- C é um fator de correção de 3 dB para regiões urbanas e 0 para regiões suburbanas e cidades menores.

2.4 Inteligência Artificial - Abordagens para Regressão

Apesar de ser um assunto amplo e com diversas abordagens possíveis, neste trabalho foram exploradas apenas as abordagens de regressão utilizando algoritmos de aprendizado supervisionado, onde o modelo aprende a partir de dados rotulados para realizar previsões em novos conjuntos de dados. O foco está em técnicas voltadas especificamente para regressão, que têm como objetivo prever um valor contínuo, como a perda de sinal em redes LoRa. O uso de modelos supervisionados permite que o sistema desenvolva um entendimento das relações entre as variáveis de entrada e a variável de saída, com potencial para resolver problemas em que o comportamento dos dados pode ser modelado por funções matemáticas.

Um dos maiores desafios no treinamento de modelos de aprendizado supervisionado é o *overfitting*, que ocorre quando o modelo se ajusta demasiadamente aos dados de treinamento, capturando também os ruídos e variações não representativas dos padrões gerais. Isso resulta em um desempenho excelente no conjunto de treino, mas uma capacidade limitada de generalização para novos dados. Para evitar *overfitting*, técnicas como a divisão do conjunto de dados para validação do modelo com dados não utilizados no treinamento,

a regularização e a escolha e ajuste de hiperparâmetros são empregadas, garantindo que o modelo tenha um desempenho consistente em diferentes cenários.

2.4.1 Modelos baseados em árvore de decisão

Uma árvore de decisão é um modelo de aprendizado de máquina que busca representar decisões e suas possíveis consequências de forma hierárquica. O processo de construção da árvore envolve a divisão recursiva do conjunto de dados com base em atributos, de modo a maximizar a pureza das classes em cada ramificação. Cada nó interno representa uma decisão, enquanto as folhas correspondem às conclusões ou classes preditas. O conhecimento extraído não é apenas uma função “caixa preta”, mas pode ser compreendido e interpretado por humanos na forma de uma árvore de decisão legível (ERTEL, 2018).

Há duas abordagens possíveis. A primeira é a busca pelo menor erro, independente da complexidade da árvore final. A segunda forma é realizar a busca pela menor árvore de decisão possível para uma taxa de erro predefinida. Isso significa que, entre todas as árvores com a mesma taxa de erro, a escolha deve recair na menor em termos de número de decisões. Esse princípio visa evitar *overfitting*. Uma árvore menor tende a ser mais transferível para novos conjuntos de dados, facilitando sua compreensão e manutenção.

Florestas Aleatórias, por sua vez, são uma extensão das árvores de decisão. Elas operam através da construção de múltiplas árvores de decisão durante o treinamento e, em seguida, combinam suas previsões para obter uma decisão mais robusta e menos propensa a *overfitting*. Cada árvore é construída com uma amostra aleatória dos dados de treinamento e com um subconjunto aleatório dos atributos em cada divisão, introduzindo aleatoriedade no processo.

Essa abordagem mitigadora do *overfitting*, combinada com a capacidade de lidar com dados complexos e multidimensionais, torna as florestas aleatórias uma escolha popular em diversos problemas de aprendizado de máquina. Elas proporcionam interpretabilidade, escalabilidade e eficiência na predição.

Há diversos hiperparâmetros que devem ser determinados para a floresta aleatória. Neste trabalho, dois serão explorados, que são a “profundidade das árvores” e a “quantidade de estimadores” (ou árvores) na floresta. A profundidade refere-se ao número máximo de níveis que cada árvore da floresta pode ter. Uma maior profundidade permite que as árvores capturem relações mais complexas nos dados de treinamento, mas também pode aumentar o risco de *overfitting*, especialmente se não for devidamente controlada. Já a quantidade, refere-se ao número total de árvores que compõem a floresta. Aumentar a quantidade de estimadores geralmente leva a um modelo mais robusto e estável, melhorando o desempenho das previsões, tornando menos suscetíveis a flutuações nos dados de treinamento. No entanto, existe um ponto de saturação, onde o benefício de adicionar mais árvores diminui, e o custo computacional pode se tornar proibitivo. Quando o tempo de treinamento permite, a melhor abordagem é testar todas as combinações possíveis hiper-

parâmetros até encontrar o ponto de saturação onde o incremento de complexidade não incrementa a redução do erro (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

Outro modelo baseado em árvore é o XGBoost, que é a abreviação de “*Extreme Gradient Boosting*”, é uma técnica de aprendizado de máquina que ganhou destaque por sua eficácia e escalabilidade em uma ampla gama de problemas. É frequentemente utilizado em competições de ciência de dados, demonstrando sua capacidade de lidar com conjuntos de dados complexos e grandes volumes de dados.

Uma das principais características do XGBoost é o seu algoritmo de aumento de árvores. A principal diferença em relação à floresta aleatória é que esta usa uma técnica chamada *bagging* para construir árvores de decisão completas em paralelo, a partir de amostras de inicialização aleatórias do conjunto de dados. Por outro lado, o XGBoost trabalha com *boosting*, que é a melhoria de um único modelo fraco combinando-o com vários outros modelos fracos para gerar um modelo coletivamente forte. O *gradient boosting* é uma extensão do *boosting*, onde o processo de geração aditiva de modelos fracos é formalizado como um algoritmo de descida de gradiente sobre uma função objetivo. O *gradient boosting* define resultados alvo para o próximo modelo na tentativa de minimizar os erros (ZHAO; RAMOS; LI, 2023).

O XGBoost tem a vantagem de ser capaz de lidar com dados esparsos. Ele possui um algoritmo ciente da esparsidade que otimiza o processo de aprendizado para esse tipo de dados. Este algoritmo também utiliza uma técnica chamada “esboço quantil ponderado” para melhorar a eficiência computacional durante o aprendizado da árvore de decisão. Uma árvore de decisão é construída selecionando pontos de corte nos dados que dividem os dados de maneira a maximizar a pureza dos nós resultantes. O esboço quantil ponderado permite que o XGBoost aproxime esses pontos de corte, o que reduz a necessidade de calcular todos os pontos de corte possíveis, melhorando a eficiência computacional.

O XGBoost também se destaca por sua capacidade de fornecer insights sobre padrões de acesso ao cache, compressão de dados e estratégias de fragmentação, permitindo construir sistemas escaláveis que podem lidar com grandes conjuntos de dados em ambientes distribuídos. Entre os recursos avançados do XGBoost estão a penalização inteligente de árvores, uma redução proporcional de nós folha e reforço de Newton, que permitem o ajuste do modelo reduzindo o *overfitting* (CHEN; GUESTRIN, 2016).

2.4.2 Support Vector Regression

A Regressão por Vetores de Suporte (*Support Vector Regression*, SVR) é uma extensão da Máquina de Vetores de Suporte (SVM), projetada para resolver problemas de regressão. Enquanto a SVM é comumente associada à classificação, a SVR é empregada para lidar com tarefas de previsão contínua. A essência da SVM reside no conceito de hiperplano, uma superfície de decisão que otimiza a separação entre classes. No caso da SVR, o foco é encontrar um hiperplano que melhor se ajusta aos dados de treinamento,

minimizando a magnitude dos erros de previsão.

Embora a SVR seja capaz de realizar tarefas de regressão, sua origem na SVM mantém a ênfase na busca por um hiperplano ótimo para separar classes ou, no caso da SVR, para ajustar uma função que descreve a relação entre as variáveis de entrada e saída. O desempenho da SVM em espaços de alta dimensão é uma de suas principais vantagens, tornando-a especialmente útil quando lidamos com conjuntos de dados complexos e multidimensionais (TRIPATHI; Mahmud, 2023).

A escolha do kernel influencia diretamente a capacidade da SVR em modelar relações complexas nos dados de entrada, pois ele permite a transformação de espaços de baixa dimensão em espaços de alta dimensão. O kernel linear é o mais simples, aplicando uma transformação linear direta aos dados de entrada. Ele é eficaz quando os dados são linearmente separáveis. No entanto, para problemas mais complexos, são necessários kernels mais sofisticados. O kernel polinomial introduz não linearidades por meio de uma transformação polinomial nos dados, permitindo a modelagem de relações mais complexas. A escolha do grau do polinômio no kernel polinomial influencia a flexibilidade do modelo, sendo este um hiperparâmetro.

O kernel Radial Basis Function (RBF) ou gaussiano é utilizado devido à sua capacidade de capturar padrões não lineares complexos. Ele mapeia os dados para um espaço de dimensão infinita, sendo especialmente eficaz quando não se conhece a estrutura subjacente dos dados. O parâmetro de largura de banda (γ) no kernel RBF controla a influência de cada exemplo de treinamento na construção do hiperplano de decisão. Por fim, o kernel sigmoide aplica uma função tangente hiperbólica à combinação linear dos dados de entrada.

2.4.3 Regressão Lasso

Na análise de modelos de predição, é necessário entender a relação entre os erros de *bias* e variância. Um modelo com baixo erro nos dados de treino, mas alto erro nos dados de teste, sugere alta variância, indicando que o modelo se ajustou demais aos dados de treino e não generaliza bem para novos dados. Por outro lado, se o erro é alto em ambos, treino e teste, é sinal de alto *bias*, sugerindo que o modelo é muito simplificado para capturar a complexidade dos dados. A regressão Lasso é uma ferramenta para lidar com esses desafios ao introduzir regularização, ou seja, a inserção controlada de *bias* no modelo.

Dentro do contexto da regressão, métodos de regularização como Lasso atuam por meio da aplicação de penalidades aos coeficientes do modelo. O Lasso utiliza a penalização L1, que pode levar alguns coeficientes a zero. Isso é útil em situações onde há múltiplas *features* altamente correlacionadas, pois o Lasso automaticamente seleciona *features*, descartando algumas e simplificando o modelo. Além de mitigar a variância, a regularização Lasso oferece uma solução para o problema de seleção de *features*, contri-

buindo para modelos menos propensos a *overfitting*, especialmente em casos de conjuntos de dados com uma quantidade limitada de amostras.

Durante o treinamento de uma regressão Lasso, há um hiperparâmetro que deve ser ajustado, que é o alfa (α), também chamado de parâmetro de regularização. O α controla a força da penalidade aplicada aos coeficientes do modelo durante o processo de treinamento. Um valor maior de α resulta em uma penalização mais forte, levando a uma simplificação maior do modelo e possivelmente à eliminação de alguns coeficientes. Por outro lado, um α menor permite que o modelo seja mais flexível, ajustando-se mais aos dados de treino (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

2.4.4 Rede neural

As redes neurais são modelos computacionais inspirados na estrutura e funcionamento do cérebro humano. Elas são compostas por unidades interconectadas, chamadas neurônios, que trabalham em conjunto para aprender e representar relações complexas entre entradas e saídas desejadas. Cada neurônio em uma rede neural recebe entradas ponderadas, aplica uma função de ativação para determinar sua saída e, em seguida, transmite essa saída para outros neurônios. Este processo iterativo de ajuste dos pesos das conexões permite que as redes neurais aprendam de maneira eficaz (HAYKIN, 2008).

As funções de ativação introduzem não-linearidades nas redes neurais, permitindo que elas aprendam a representar e generalizar dados complexos. Algumas funções de ativação comuns incluem a função sigmoide, tangente hiperbólica e a função ReLU (*Rectified Linear Unit*). A escolha da função de ativação é geralmente determinada pelo problema em questão e pelas características dos dados, mas também pode ser tratada como um hiperparâmetro e ser definida por testes na base de dados.

Durante o treinamento de uma rede neural, um algoritmo chamado otimizador é usado para ajustar os pesos das conexões entre os neurônios. O objetivo é minimizar uma função de custo que quantifica o erro entre as saídas previstas pela rede e as saídas reais. Alguns otimizadores populares incluem o Gradiente Descendente Estocástico (SGD), o Adam e o RMSprop. Cada otimizador tem suas próprias vantagens e desvantagens em termos de convergência, velocidade de treinamento e robustez, sendo este, outro hiperparâmetro (GOODFELLOW; BENGIO; COURVILLE, 2016).

As camadas em uma rede neural são os blocos fundamentais que compõem sua estrutura. Cada camada é composta por um conjunto de neurônios e pode desempenhar diferentes funções. A camada de entrada recebe os dados de entrada, as camadas ocultas processam e transformam esses dados em representações úteis, e a camada de saída gera as previsões finais do modelo. As redes neurais podem ser organizadas em várias topologias, que se referem à estrutura e organização dos neurônios em camadas. Com base na configuração das camadas, as redes neurais podem ser classificadas em várias categorias, como redes *feedforward*, redes recorrentes (RNN) e redes convolucionais (CNN).

As *feedforward* são as redes neurais mais simples, onde os dados se movem em uma única direção, da camada de entrada para a camada de saída, passando pelas camadas ocultas, caso hajam. Não há ciclos ou loops, o que facilita a análise e o treinamento. Diferentemente das redes *feedforward*, as RNNs possuem conexões cíclicas, permitindo que informações de etapas anteriores influenciem as etapas seguintes. Isso as torna especialmente úteis para tarefas que envolvem dados sequenciais, como processamento de séries temporais. As CNNs são projetadas para processar dados com uma estrutura de grade, como imagens. Elas utilizam camadas convolucionais que aplicam filtros para extrair características locais dos dados, seguidas por camadas de *pooling* que reduzem a dimensionalidade (HAYKIN, 2008).

3 REVISÃO DA LITERATURA

Este capítulo apresenta uma revisão da literatura relevante para a pesquisa, abordando os principais estudos e desenvolvimentos relacionados à propagação de redes LoRa, o uso de inteligência artificial em modelos de propagação e o uso de simuladores para redes LoRa. Primeiramente, é realizada uma análise bibliométrica, onde são definidos os critérios de busca e apresentados os critérios para seleção dos artigos. Em seguida, é detalhado o estado da arte sobre propagação de sinal LoRa, destacando os trabalhos que trouxeram contribuições relevantes ao campo. Na sequência, é explorado o uso da inteligência artificial em modelos de propagação de redes sem fio, focando nos estudos que não envolvem redes LoRa. Por fim, são discutidos os principais simuladores de redes LoRa disponíveis na literatura.

3.1 Propagação de redes LoRa e LoRaWAN

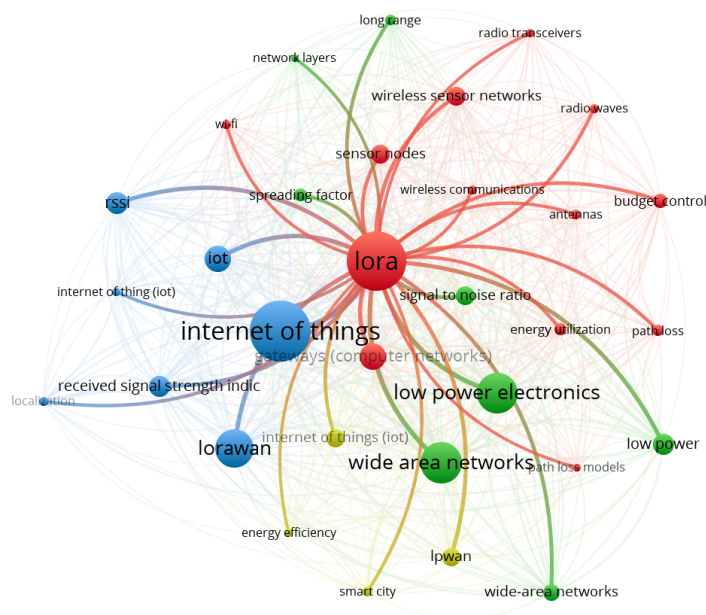
3.1.1 Análise bibliométrica

Na fase inicial da seleção de artigos para esta dissertação, foram realizadas buscas em diversas bases de dados, como IEEE Xplorer, SCOPUS e ELSEVIER, utilizando termos-chave relacionados ao tema de pesquisa, tais como "*LoRa*", "*path loss*", "*propagation*", "*RSSI*", "*channel model*" e "*link budget*".

A Figura 4 ilustra uma nuvem de palavras-chave gerada pelo software VosViver, representando graficamente os termos mais frequentes e relevantes nos artigos encontrados na busca inicial. O tamanho de cada nó na figura reflete a frequência com que as palavras-chave aparecem nos estudos, ou seja, nós maiores indicam palavras mais recorrentes e centrais na área de pesquisa. As conexões entre os nós representam a coocorrência das palavras-chave, sendo que as conexões mais fortes indicam uma relação mais frequente e significativa entre os termos. Com esta análise, além das palavras-chave já utilizadas, um novo termo foi incorporado à busca, o "*budget control*".

A distribuição das publicações por ano e país também foi analisada, principalmente para compreender a evolução cronológica das contribuições acadêmicas em relação à temática, além de delimitar os filtros na etapa seguinte de avaliação. O gráfico da Figura

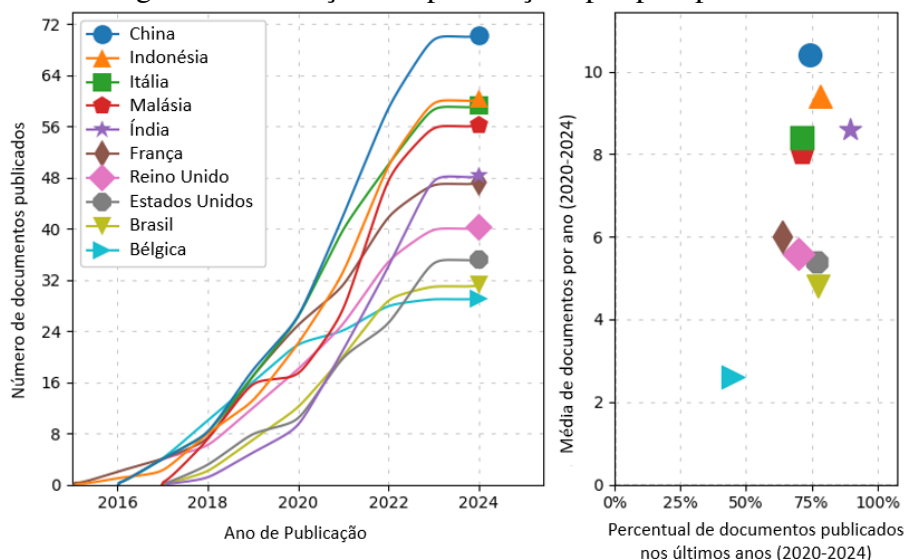
Figura 4 – Palavras-chave mais encontradas na pesquisa.



Fonte: do autor.

5 identifica a distribuição geográfica das publicações, permitindo observar a produção científica ao longo do tempo por país, ajudando a definir os trabalhos mais relevantes e as tendências de produção científica em diferentes regiões.

Figura 5 – Evolução das publicações por país por ano.



Fonte: do autor.

Essa abordagem resultou em uma lista de cerca de 800 artigos. Para refinar a seleção, foi aplicado um filtro inicial que considerou aspectos como relevância, atualidade e enfoque. Os trabalhos que não utilizavam a tecnologia LoRa nos estudos de propagação foram descartados, assim como aqueles trabalhos publicados antes de 2015. Também fo-

ram descartados os artigos publicados em idiomas diferentes do inglês. A filtragem inicial resultou em 120 artigos para análise aprofundada. Na sequência, os resumos, introduções e conclusões desses artigos foram minuciosamente avaliados. Como resultado desse processo de seleção, foram identificados 47 trabalhos que se mostraram particularmente pertinentes para o escopo da dissertação, e todos eles foram submetidos a uma análise integral. Dessas 47 fontes selecionadas, 11 foram considerados os mais relevantes, por apresentarem avanços significativos no estado da arte do objeto de estudo em questão, e suas principais descobertas e contribuições são detalhadamente abordadas nesta revisão. Ao final, são apresentadas tabelas, evidenciando as condições dos ensaios realizados (Tabela 2), e comparações (Tabela 3).

3.1.2 Estado da arte

Há diversos trabalhos na literatura que buscam modelar, de forma teórica ou experimental, a propagação de sinais LoRa. Uma das primeiras pesquisas que buscou modelar o canal foi a de PETAJARVI *et al.* (2015), que coletou dados de nós móveis na frequência de 868 MHz, instalados em um carro e em um barco, com um único gateway em uma cidade da Finlândia. O trabalho resultou em 10 mil medições, com as quais modelou-se o canal de rádio com *Log-distance*, apresentando também o parâmetro *shadow fading*, que engloba toda a variabilidade do meio que não está no modelo, uma vez que somente a distância entre transmissor e receptor é considerada. Este trabalho serviu de base para diversos simuladores LoRa e é frequentemente encontrado na literatura como modelo de Oulu (*Oulu Model*).

Considerando todos trabalhos encontrados nesta avaliação, a pesquisa de RADEMACHER *et al.* (2021) se destaca como a de maior coleta de dados já realizada no contexto de redes LoRa em um ambiente urbano. Essa pesquisa é de grande relevância no estado da arte, pois proporcionou uma visão abrangente e detalhada da propagação de sinais LoRa em uma área urbana de 200 km² ao longo de 230 dias. Cerca de 72% de todos os pacotes foram recebidos por pelo menos dois gateways, indicando uma boa cobertura na cidade e redundância na infraestrutura. Um aspecto único desta pesquisa é a estimativa do tamanho da amostra necessário para realizar um ajuste confiável no modelo de perda de caminho *Log-distance*. Os resultados indicam que são necessárias entre 20 a 30 mil amostras para obter uma precisão satisfatória nesse ajuste. Em termos de desempenho dos modelos de propagação, o estudo revela que o menor RMSE foi alcançado ao calcular os parâmetros para o modelo de *Log-distance* usando os dados coletados, com um valor de 9,9 dBm. Entre os modelos testados sem ajuste de parâmetros, o Okumura-Hata foi o melhor, apresentando um RMSE de 10,32 dB.

A maioria dos trabalhos encontrados na pesquisa utiliza um concentrador (gateway) com antenas mais elevadas e os nós sensores espalhados em uma topologia estrela. Com uma abordagem diferente, CALLEBAUT; PERRE (2020) conduziu uma avaliação da co-

bertura e perda por propagação em links ponto a ponto (P2P) LoRa, com base em campanhas de medição em ambientes urbanos, florestais e costeiros. O estudo revelou que os links P2P apresentam perda maior do que as topologias de rede em estrela, devido a um maior número de obstruções e à menor altura dos transmissores e receptores, resultando em maior perda de pacotes abaixo da sensibilidade do receptor. Para obter parâmetros de PL mais realistas, foi proposta uma abordagem de modelagem de máxima verossimilhança (ML), levando em conta dados que não foram recebidos devido a RSSI baixos. Essa abordagem forneceu parâmetros de PL com erros menores, com RMSE e MAE (*Mean Absolute Error*) de 11,01 e 8,97 dB, respectivamente.

Há um grande interesse na modelagem de nós sensores em ambientes urbanos e suburbanos, uma vez que a tecnologia LoRa pode ser habilitadora para diversas aplicações de cidades inteligentes, porém há aplicações em diferentes meios. Alguns trabalhos, como o de FERREIRA *et al.* (2020), investigaram a propagação em ambientes florestais e compararam com ambientes urbanos e suburbanos, destacando variações significativas no alcance do LoRa, com alcances de até 250 m em ambientes florestais, 1 km em áreas suburbanas e 200 m em áreas urbanas. Ambientes florestais apresentaram conectividade intermitente devido a efeitos multipercursos e ressonância causada pela vegetação. A mobilidade dos nós também impactou a qualidade do sinal e o alcance. O trabalho de ANZUM *et al.* (2022) também utilizou o meio florestal para coleta de dados, realizando a análise da propagação LoRa em canais de frequência de 433 MHz em uma plantação de óleo de palma na Malásia. Os autores caracterizaram a influência da folhagem, propondo um modelo de previsão de perda de caminho baseado em várias camadas, que demonstrou um desempenho superior aos modelos empíricos existentes, com um RMSE médio de 2,74 dB, enfatizando a importância de considerar a influência da folhagem em ambientes específicos como no caso, plantações de óleo de palma.

A propagação de sinais LoRa em ambientes montanhosos e com neve é um caso a parte nas modelagens de propagação, mas é uma rota tecnológica viável para sistemas de busca e resgate em áreas remotas. Os estudos de OLASUPO (2019) e BIANCO *et al.* (2021) abordam essa questão com foco na aplicação de dispositivos IoT em ambientes rochosos e montanhosos para operações de resgate e monitoramento de atividades recreativas em trilhas. Olasupo observou que modelos teóricos existentes apresentam desvios significativos em relação às medições práticas, com perda de sinal média cerca de 8 dB maior em áreas montanhosas do que em modelos destinados a ambientes com elevações menores. Foram desenvolvidos modelos de perda de percurso com base em medições práticas, destacando a importância de dispositivos de frequência mais baixa e maior potência de transmissão para melhor desempenho em ambientes montanhosos. Por sua vez, Bianco *et al.* explorou o uso da tecnologia LoRa para operações de busca e resgate em montanhas, apresentando um sistema baseado em LoRa e um algoritmo de localização com base em medições de potência do sinal. A incerteza do sistema pode atingir a ordem

de metros em torno da posição real.

Os dispositivos LoRa também são utilizados em redes de sensores enterrados, coletando dados úteis para a agricultura de precisão. Com esta aplicação, o trabalho de RENZONE *et al.* (2021) investigou o uso de rádios LoRa para transmitir dados do subsolo para a superfície (*Underground to Above Ground, UG2AG*) em diferentes tipos de solo, como cascalho, areia e argila. Utilizando nós sensores enterrados a profundidades de até 50 cm, foram avaliados parâmetros como RSSI, SNR e perda de pacotes. Os resultados indicaram transmissões bem-sucedidas em todos os tipos de solo até 50 cm de profundidade, com uma perda de pacotes inferior a 2%. No entanto, o estudo não considerou na modelagem teórica variáveis como a umidade do solo e a composição química, que podem influenciar o desempenho. Além disso, o modelo teórico de perda de caminho revelou diferenças de até 25 dBm em relação às medições reais de RSSI. O estudo de LIN; HAO (2021) amplia o trabalho anterior ao trabalhar com variações dos parâmetros físicos dos sensores subterrâneos. Uma contribuição significativa está na validação experimental de que as características de propagação são significativamente diferentes entre UG2AG e superfície para subsolo (*Above Ground to Underground, AG2UG*) em redes LoRa. Com isso, foi desenvolvido um modelo de canal mais preciso que considera todas as perdas, e avalia o impacto dos parâmetros físicos do LoRa, como fator de espalhamento, largura de banda e taxa de código, levando a um RMSE menor que 8 dB.

O trabalho conduzido por CHALL; LAHOUD; HELOU (2019) envolveu a coleta de dados tanto em ambientes internos quanto externos, em cenários urbanos e rurais. Foram aplicados diversos modelos de propagação, como Okumura-Hata, COST231-Hata, 3GPP RMA e UMA para ambientes externos, e ITU-R, Cost-231 MWF e 3GPP cIOT para ambientes internos. Além disso, propôs um modelo baseado em *log-distance*, mas que se diferencia pela introdução de fatores de correção específicos para a presença de paredes e andares em ambientes internos, bem como um fator de correção para a altura da antena em ambientes externos. Esses ajustes foram baseados em dados experimentais coletados em diferentes alturas do transmissor. Por não utilizar diferentes *datasets* para regressão dos parâmetros do modelo e para avaliação do desenvolvimento, o erro médio apresentado é zero para todos os modelos propostos, enquanto os modelos existentes apresentaram erros médios variando de 0,48 a 10,2 dB para ambientes internos e de -1 a 3,9 dB para ambientes externos.

O trabalho de XU *et al.* (2020) concentrou-se exclusivamente em ambientes internos, com testes realizados em quatro tipos diferentes de edifícios, incluindo escritórios, residências, depósitos e estacionamentos, variando de um a seis andares. Foram exploradas uma ampla gama de configurações, incluindo diferentes larguras de banda (125 kHz, 250 kHz e 500 kHz) e faixas de frequência (de 915 MHz a 928 MHz). A principal contribuição reside na proposta de um modelo de propagação baseado no conceito do *log-distance*, adaptando-o com fatores de correção determinados experimentalmente para diferentes ti-

pos de paredes, como paredes de concreto e paredes divisórias. A abordagem resultou em erro médio de 3,5 dB para dispositivos LoRa e gateways localizados no mesmo andar, e 5,4 dB quando em andares diferentes.

Além dos estudos mencionados, uma ampla variedade de pesquisas adicionais ofereceu contribuições específicas no tema. Por exemplo, TORRE *et al.* (2018) conduziu medições sem obstruções na Antártida usando rádios de 433 e 868 MHz, enquanto LIN *et al.* (2019) realizou medições com rádios submersos transmitindo e recebendo sinais para outros rádios submersos ou fora da água. VILLARIM *et al.* (2019) coletou dados e aplicou o modelo *log-distance* para a propagação em florestas densas, enquanto ALMARHABI; ALJOHANI; MOINUDDIN (2021) avaliaram a modelagem da propagação em sistemas em altitudes elevadas em relação ao solo, em equipamentos como satélites e drones. Além disso, BREITEGGER *et al.* (2021) conduziram extensa campanha de medições com rádios em ambientes industriais, especificamente em uma planta química.

Em aplicações específicas, merece destaque a pesquisa de CATHERWOOD *et al.* (2017), que modelou as perdas de propagação para dispositivos vestíveis. Para aplicações em ambientes rurais, MENDES; PASSOS; CORREIA (2022) coletou dados de propagação em uma plantação de laranjas, onde as linhas de plantio eram igualmente espaçadas, enquanto PHAIBOON; PHOKHARATKUL (2023) realizaram medições em uma plantação de mangas na Tailândia.

Especificamente para redes LoRa, foi encontrado apenas um trabalho utilizando inteligência artificial para a predição de perdas por propagação. GONZÁLEZ-PALACIO *et al.* (2023) discutem o uso de técnicas de aprendizado de máquina para modelar a perda de sinal e o desvanecimento em redes LoRaWAN, visando aprimorar a eficiência energética. Os autores coletaram dados de temperatura, umidade, pressão e material particulado com sensores fixos e propuseram modelos de predição de perdas baseados em múltiplas técnicas de aprendizado de máquina, sendo que o melhor resultado alcançou um RMSE de 1,57 dB. Apesar de explorar o mesmo tópico, o trabalho publicado não tem como foco explorar a perda por propagação e sim o impacto das condições ambientais na propagação.

A contribuição principal deste trabalho reside na implementação e avaliação de modelos de predição de perda por propagação em redes LoRa, utilizando tanto abordagens clássicas quanto técnicas de Inteligência Artificial. Diferente do encontrado nesta revisão, além de comparar o desempenho dos modelos existentes com dados reais de nós móveis e fixos, esta pesquisa vai além ao integrar esses modelos em um simulador, permitindo avaliar o impacto das diferentes abordagens no dimensionamento de redes LoRa.

3.2 Uso de inteligência artificial para modelos de propagação de redes sem fio

O estudo da perda de propagação em comunicações sem fio tem recebido atenção considerável devido à sua importância no planejamento de redes de larga escala. Há três abordagens possíveis em relação à modelagem, que são (i) o uso de modelos empíricos, que utilizam, por exemplo, frequência operacional, distância entre transceptores e altura da antena, (ii) modelos estatísticos, que necessitam de quantidade elevada de dados coletados para ajuste de parâmetros, ou o uso de (iii) modelos geométricos baseados em mapas, que requerem mais informações como faixa de elevação entre transceptores, construções no caminho e características de dispersão estatística do meio (HUANG *et al.*, 2022).

O uso de inteligência artificial contribui para a redução de erros dos modelos de propagação das três abordagens. Utilizando uma abordagem de modelos geométricos, mas auxiliado por uma rede neural convolucional profunda, HAYASHI; ICHIGE (2023) propõe um método destinado a aprimorar a precisão da previsão da perda por propagação, combinando informações espaciais em torno de pontos de transmissão e recepção com parâmetros relacionados ao ambiente. O método proposto padroniza o ambiente no caminho de propagação entre os pontos com um tamanho constante, independentemente da distância, capturando características como os caminhos de propagação dominantes que contribuem para as características da perda por propagação. A estimativa realizada foi melhor em 800 MHz do que em 2 GHz, possivelmente devido à zona de Fresnel menor em frequências mais altas. Abordagens parecidas também foram realizadas por INOUE *et al.* (2022) e ZHANG *et al.* (2022) com resultados semelhantes.

O trabalho de LOH *et al.* (2023) propõe um modelo de aprendizado de máquina para prever a localização das estações base em ambientes urbanos. Utilizando dados reais das cidades de Munique e Singapura, a abordagem do aprendizado de máquina permitiu uma previsão melhor da perda por propagação em relação aos modelos empíricos, com um MAE de 3,27 dB em comparação com os 4,75 dB do Okumura-Hata. Posteriormente, o modelo foi utilizado para determinar a área de cobertura e a localização ótima das estações base visando a maximização da cobertura.

Levando em consideração tanto as abordagens dos modelos geométricos quanto dos modelos empíricos, MOROCHO-CAYAMCELA; MAIER; LIM (2020) propõe o uso de aprendizado profundo para segmentar imagens aéreas em ambientes urbanos, suburbanos e rurais, visando estimar com maior precisão a perda de propagação de sinais de RF. A rede atinge precisões de segmentação de 89,41%, 86,47% e 87,37% para classes urbanas, suburbanas e rurais, respectivamente. Na sequência, o modelo de Okumura-Hata é aplicado para cada segmento por onde há propagação do sinal, considerando o tipo de ambiente identificado pelo modelo de inteligência artificial. A abordagem proposta é aplicada em dados reais de duas fontes e reduz o desvio quadrático médio em 3,79 dB e 4,09

dB em comparação com o modelo Okumura-Hata de ambiente único.

Algoritmos de aprendizagem de máquina também podem ser utilizados de forma semelhante aos modelos estatísticos, com grandes *datasets* para prever o comportamento da propagação de sinal em um ambiente. Com dados normalmente utilizados tanto por modelos estatísticos quanto empíricos, como elevação, distância entre transmissor e receptor e distância entre prédios, ISABONA *et al.* (2022) desenvolveu modelos SVR com kernel de função de base radial (RBF) para predição da perda por propagação em uma rede sem fio. O desempenho do modelo de aprendizado de máquina foi comparado com cinco modelos empíricos existentes usando métricas como erro quadrático médio e R-quadrado, e concluiu que o modelo que usa IA apresenta estimativas de perda de caminho melhor do que os modelos empíricos em áreas rurais, suburbanas e urbanas.

3.3 Simuladores de redes LoRa

Devido à necessidade de avaliar o desempenho de redes com alta densidade de nós, que podem ser desafiadoras de serem testadas em ambientes reais, utiliza-se simuladores de redes. Embora a quantidade de pacotes transmitidos por dia e o tempo no ar de cada pacote sejam limitados em redes LoRa, o que dificulta a coleta de uma quantidade grande de dados experimentais, a complexidade surge quando estuda-se redes com muitos nós e diferentes configurações de parâmetros. Para enfrentar esses desafios, a literatura frequentemente utiliza a implementação de redes menores em experimentos controlados, como uma forma de validar parâmetros que seriam aplicáveis a redes maiores. Há diversos simuladores implementados na bibliografia para avaliação de parâmetros controláveis em redes LoRa de larga escala, sendo os principais implementados para o Network Simulator (ns-3) ou em Python.

Um dos primeiros trabalhos publicados sobre simulador de rede LoRa em Python foi o LoRaSim, um simulador de eventos discretos. O simulador utiliza Simpy¹ com foco nos conflitos de pacotes em uma rede LoRa, com o objetivo de avaliar a escalabilidade da rede (BOR *et al.*, 2016). Neste trabalho, foi desenvolvida uma plataforma experimental para coleta de dados para criação do simulador, usando dados reais para definir a sensibilidade máxima no receptor para cada banda e fator de espalhamento. As perdas pelo meio foram modeladas através do modelo *log-distance*, utilizando os parâmetros encontrados por PETAJARVI *et al.* (2015). No simulador, a variância da perda pelo trajeto não é levada em consideração, sendo a intensidade do sinal dependente somente da distância. Uma das contribuições significativas deste trabalho foi o tratamento de conflitos

¹SimPy é uma biblioteca de simulação de processos discretos em Python. Permite a modelagem e simulação de processos que evoluem ao longo do tempo, como filas, eventos ou operações em sistemas. É particularmente útil para simular sistemas complexos e estocásticos, como redes de comunicação ou qualquer sistema onde a ordem e o tempo de eventos sejam importantes (SCHERFKE, STEFAN, 2014).

de pacotes considerando a simultaneidade no envio, a frequência da portadora, o fator de espalhamento e a potência dos sinais recebidos. Por outro lado, não há tratamento em relação a não-ortogonalidade em diferentes fatores de espalhamento.

Mais completo que o LoraSim, CALLEBAUT; OTTOY; VAN DER PERRE (2019) propõem um framework de simulação de diversas camadas LoRaWan chamado LoraEnergySim, visando analisar e otimizar o consumo de energia dos dispositivos IoT. Expandindo trabalhos anteriores, o framework integra a taxa adaptativa de dados e mensagens de downlink, permitindo o monitoramento específico das aplicações ao modificar parâmetros como o tamanho de *payload* e a taxa de transmissão. Utilizando medições de um nó LoRa otimizado para consumo de energia, os valores de consumo de energia são baseados em dados reais, sendo o código-fonte do simulador disponibilizado publicamente. A análise se estende às camadas físicas e de controle de acesso ao meio (MAC), explorando o impacto desses parâmetros na autonomia dos dispositivos IoT.

O LoraEnergySim fornece a possibilidade de simular o impacto da expansão para um grande número de nós, permitindo identificar oportunidades para melhorias nas otimizações das camadas, com foco principalmente em aumentar a eficiência energética da rede como um todo. Por exemplo, os resultados do autor mostram que a ativação da ADR pode levar a uma redução substancial no consumo de energia, até mesmo em uma ordem de magnitude, embora em canais com alta dinâmica, essa taxa pode impactar negativamente a taxa de extração de dados. A estratégia de aumento do tamanho do pacote pode reduzir a energia consumida por byte de *payload*, mas em contrapartida resulta em uma adaptação mais lenta às mudanças nas condições do canal, especialmente quando se trata de *payloads* maiores. Em relação aos modelos de propagação, foram implementados log-distance com os parâmetros do modelo de Oulu e também COST-231.

Apesar de mais completo que outros encontrados na literatura, o modelo desenvolvido originalmente para o LoraEnergySim ainda possuía algumas limitações. O trabalho de STUMPO *et al.* (2022) focou na melhoria do simulador para simular redes LoRa em cenários mais realistas. A versão original do simulador foi expandida para possibilitar simulações de redes com múltiplos gateways, permitindo cobrir áreas maiores e analisar a interferência entre esses gateways. A atualização também considerou a ortogonalidade imperfeita entre os fatores de espalhamento, proporcionando uma modelagem da interferência entre pacotes utilizando diferentes fatores de espalhamento. Estatísticas adicionais, como latência de pacotes, consumo de energia e perda de pacotes, foram coletadas ao término das simulações para melhor avaliação do desempenho da rede. A implementação de redes heterogêneas, variando a densidade de nós e cargas de tráfego, permitiu a simulação de cenários mais diversos. Os autores observaram que a inclusão de suporte para múltiplos gateways acarreta um aumento considerável no custo computacional devido à maior quantidade de informações a serem processadas, enquanto a adição de verificações para a ortogonalidade imperfeita dos fatores de espalhamento não gerou um aumento sig-

nificativo no custo computacional.

Há diversas implementações de rede LoRa para o software Network Simulator, e 4 destes módulos foram avaliados por SILVA *et al.* (2021). O primeiro módulo, desenvolvido em 2016, destaca-se por sua detalhada modelagem da camada física e implementação simples do servidor de rede, embora falte a modelagem de interferência com outros protocolos (MAGRIN; CENTENARO; VANGELISTA, 2017). O segundo módulo (REYNOLDERS; WANG; POLLIN, 2018) oferece uma implementação completa do servidor de rede com uma rede IP entre o gateway e o servidor, apesar de faltar alguns comandos MAC e documentação. Já o módulo 3 (DEN ABEELE *et al.*, 2017) conta com um modelo de erro baseado em simulações MATLAB, mas não inclui um modelo de consumo de energia nem uma implementação completa do servidor de rede. Por fim, o mais recente dos módulos (TO; DUDA, 2018), foi validado por meio de medições, porém carece de documentação e código de exemplo. Não foram encontradas evidências de que algum dos simuladores implementados para o Network Simulator apresente vantagens considerando a modelagem ou custo computacional em relação aos modelos implementados em python.

Tabela 2 – Condições dos Ensaios dos Trabalhos Publicados.

Trabalho	Características do Ensaio	Modelos de Propagação
(CHALL; LAHOUD; HELOU, 2019)	Indoor, Urbano e Rural, SF 12, 868 MHz	Okumura-Hata, COST231-Hata, 3GPP RMA e UMA, ITU-R, Cost-231 MWF e 3GPP cIOT
(OLASUPO, 2019)	Montanha, 928 MHz	Free-space, Log-distance, Two-Ray
(XU <i>et al.</i> , 2020)	Indoor, SF 7 a 10, 915 e 928 MHz	Log-distance
(CALLEBAUT; PERRE, 2020)	Urbano, Florestal e Costeiro, SF 7, 9 e 12, 868 Mhz	Log-distance e Okumura-Hata
(FERREIRA <i>et al.</i> , 2020)	Urbano, Suburbano e Florestal, SF 7 a 12, 868 MHz	N.A.
(RADEMACHER <i>et al.</i> , 2021)	Urbano, SF 12, 868 MHz	Free Space, Log-distance, Winner+, Okumura-Hata, COST-231, Egli, EC33, ITM e ITWOM
(RENZONE <i>et al.</i> , 2021)	Enterrado, SF 12, 868 MHz	UG2AB
(LIN; HAO, 2021)	Enterrado, SF 6, 9 e 12, 433 MHz	UG2AB e AG2UG
(BIANCO <i>et al.</i> , 2021)	Montanha, Neve e Enterrado na Neve, SF 7, 868 MHz	Log-distance
(ANZUM <i>et al.</i> , 2022)	Plantação de Palma (Floresta), SF 7 a 12, 433 MHz	COST-231, ITU, Log-distance, Weissberger
(GONZÁLEZ-PALACIO <i>et al.</i> , 2023)	Urbano, SF 7 a 12, 928 MHz	Rede Neural, Friis, Two-ray e Okumura-Hata
Este trabalho	Urbano, SF 7 a 12, 868 e 928 MHz	Free Space, Egli, Okumura-Hata, COST231-Hata, Log-distance, árvore de decisão, floresta aleatória, SVR, regressão Lasso, XGBoost e rede neural

Tabela 3 – Contribuições dos Trabalhos Publicados.

Trabalho	Contribuições
(CHALL; LAHOUD; HELOU, 2019)	Propõe um modelo baseado no Log-Distance com fatores de correção para altura da antena (outdoor) e paredes e andares (indoor). Utiliza modelos de propagação como Okumura-Hata, COST231-Hata, 3GPP RMA e UMA, ITU-R, Cost-231 MWF e 3GPP cIOT.
(OLASUPO, 2019)	Realiza a primeira medição em ambiente de montanha, identificando perdas médias 8 dBm maiores. Utiliza modelos de propagação como Free-space, Log-distance e Two-Ray.
(XU <i>et al.</i> , 2020)	Desenvolve um modelo baseado em log-distance com fator de correção experimental para diferentes tipos de paredes em ambientes indoor. Utiliza o modelo de propagação Log-distance.
(CALLEBAUT; PERRE, 2020)	Avalia nós P2P com comunicação bidirecional em ambientes urbano, florestal e costeiro, utilizando modelagem de máxima verossimilhança considerando pacotes não entregues. Utiliza modelos Log-distance e Okumura-Hata.
(FERREIRA <i>et al.</i> , 2020)	Compara propagação em ambientes florestais, urbanos e suburbanos, identificando conectividade intermitente devido a efeitos multipercursos e ressonância causada pela vegetação.
(RADEMACHER <i>et al.</i> , 2021)	A maior pesquisa experimental já realizada até a data da publicação, com 112.404 amostras, determinando o tamanho amostral necessário para o modelo log-distance. Utiliza diversos modelos de propagação como Free Space, Log-distance, Winner+, Okumura-Hata, COST-231, Egli, EC33, ITM e ITWOM.
(RENZONE <i>et al.</i> , 2021)	Modela e valida experimentalmente nós enterrados com 3 diferentes tipos de solo, utilizando o modelo UG2AB.
(LIN; HAO, 2021)	Varia diferentes parâmetros da camada física para validação experimental de nós enterrados, utilizando os modelos UG2AB e AG2UG.
(BIANCO <i>et al.</i> , 2021)	Desenvolve modelagem de propagação e algoritmo de trilateração em ambientes de montanha e neve, levando em consideração a propagação através do corpo humano no modelo log-distance.
(ANZUM <i>et al.</i> , 2022)	Estuda a dependência do SF para a perda do caminho em plantações de palma (floresta), propondo um modelo log-distance corrigido pela quantidade de topo de árvore e troncos. Utiliza modelos como COST-231, ITU, Log-distance e Weissberger.
(GONZÁLEZ-PALACIO <i>et al.</i> , 2023)	Considera os efeitos de variáveis ambientais como temperatura, umidade, pressão e material particulado em ambientes urbanos. Utiliza modelos de propagação como Rede Neural, Friis, Two-ray e Okumura-Hata.
Este trabalho	Treinamento de modelos inteligentes. Aprimoramento de um simulador de rede com modelos de propagação com erros menores.

4 MÉTODOS E MATERIAIS

Este capítulo descreve os procedimentos, ferramentas e dados utilizados para atingir os objetivos propostos. Primeiramente, são apresentados os conjuntos de dados utilizados para a avaliação dos modelos existentes e o treinamento dos modelos propostos, incluindo um dataset com nós móveis (MCLab) e outro com nós fixos (Medellín). Em seguida, detalha-se a arquitetura do simulador LoRaEnergySim, empregado para avaliar o impacto dos diferentes modelos de propagação na simulação de redes LoRa. Posteriormente, é descrito o método adotado para o treinamento dos modelos de predição que utilizam técnicas de Inteligência Artificial. Por último, são apresentados os cenários testados com o simulador.

4.1 Datasets

Dois conjuntos de dados provenientes de trabalhos previamente publicados foram utilizados nesta pesquisa. A seguir, são apresentados detalhes sobre as características e métodos de coleta de cada conjunto de dados.

4.1.1 Dataset 1: MCLab - Nós móveis

O conjunto de dados MCLAB, utilizado em um estudo recente apresentado na revisão bibliográfica ((RADEMACHER *et al.*, 2021), foi coletado em Bonn, Alemanha, um ambiente urbano típico com edifícios de diferentes alturas e 330.000 habitantes. A campanha de medição incluiu nove gateways (GW) estacionários e quatro *end nodes* (ENs) com GPS móveis. Os GWs foram estrategicamente posicionados na cidade, levando-se em consideração a disponibilidade de localização (postes e telhados). Diversos modelos de GWs da empresa Kerlink foram empregados, cada um equipado com uma antena omnidirecional de 3 dBi. A medição do RSSI foi realizada no *uplink*, dos ENs para os GWs, e esse método foi escolhido para contornar limitações de ciclo de trabalho nos GWs e simplificar os sensores, evitando o armazenamento de dados. Para obter uma quantidade significativa de amostras de medição de toda a cidade, quatro caminhões de coleta de lixo foram utilizados como veículos móveis. Os sensores foram montados nos telhados desses

caminhões, integrando-se às rotas diárias. Devido às condições adversas nos caminhões, como vibrações e exposição a água, foi desenvolvido um sensor próprio, composto por uma unidade de processamento, um modem LoRa, um acelerômetro, um módulo GPS e uma bateria.

Durante a campanha, cada sensor em um caminhão de lixo transmitiu pacotes LoRa com coordenadas GPS determinadas sempre que o caminhão estava em movimento, sendo recebidos por vários GWs. A campanha abrangeu mais de 200 km² de ambiente urbano, resultando em 175.492 amostras individuais de medição ao longo de 230 dias. Na etapa de pós-processamento, observou-se que a precisão do GPS precisava ser abordada antes da avaliação dos diferentes modelos de perda de percurso (PL). Assim, foram removidas medições que não tinham pelo menos 5 satélites com visada, totalizando cerca de 8% dos dados. Posteriormente, a imprecisão remanescente das medições GPS foi compensada pelo mapeamento das localizações para as ruas mais próximas usando o serviço "Open Source Routing Machine". Medições com um desvio superior a 20 metros entre a localização medida e a rua esperada foram filtradas, representando aproximadamente 22% dos dados. Outras exclusões envolveram medições com altitude superior ao ponto mais alto da cidade (cerca de 1%). Após o pós-processamento, restaram 112.372 amostras.

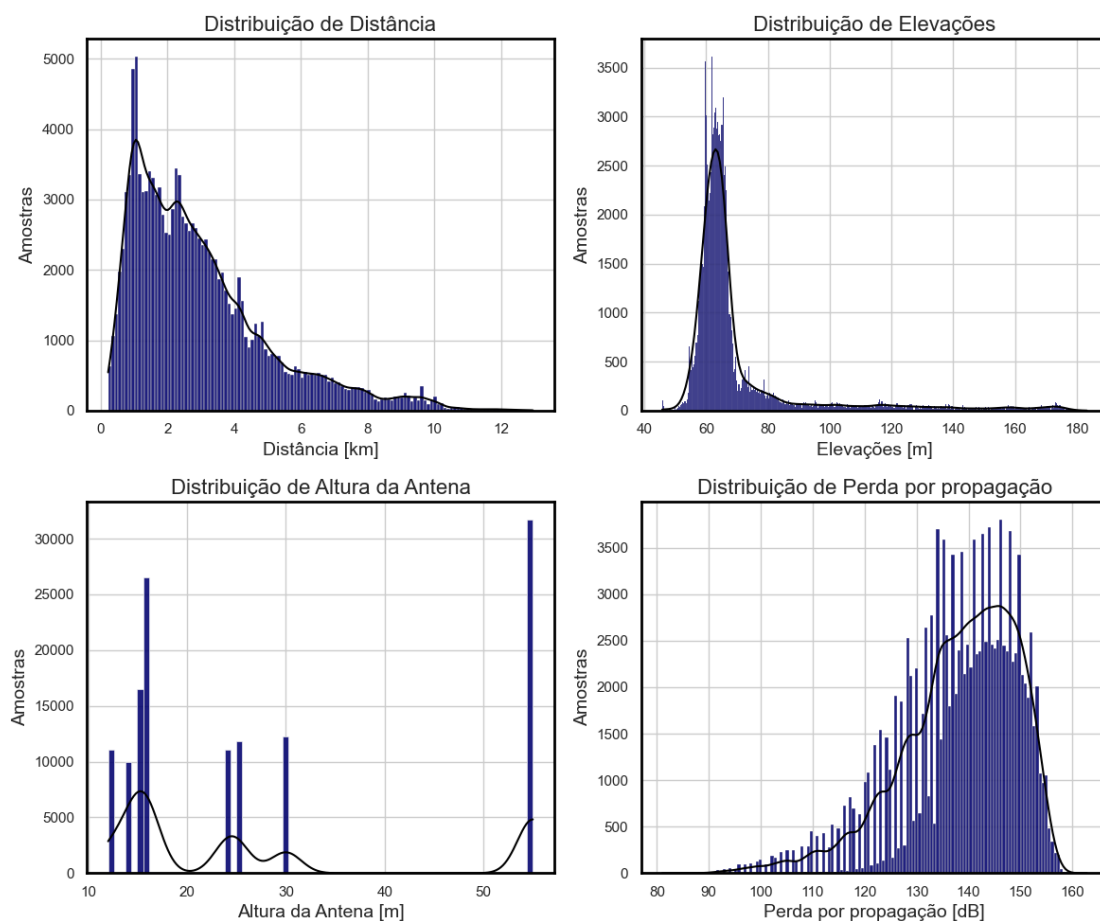
O dataset não possui nenhuma informação sobre elevação do terreno, porém estes dados podem ser obtidos através da coordenada GPS. Assim, para enriquecer o *dataset*, foi realizada uma fusão com o EU-DEM, um conjunto de dados de elevação abrangendo a Europa com resolução de 25 metros. Este *dataset* de elevação é uma implementação do programa Copernicus, gerenciado pela Comissão Europeia. O EU-DEM foi construído ao mesclar dados de elevação dos conjuntos globais SRTM (*Shuttle Radar Topography Mission*) e ASTER (*Advanced Spaceborne Thermal Emission and Reflection Radiometer*), além de mapas topográficos soviéticos em latitudes elevadas. O datum utilizado é o EVRS2000 (*European Vertical Reference System*). Em termos de cobertura geográfica, o conjunto de dados abrange os estados membros da Agência Europeia do Ambiente, além de alguns países a leste. A cobertura se estende a pequenas partes do norte da África, incluindo regiões escandinavas ao norte de 60°, o que não está presente no SRTM (OPEN TOPO DATA, 2023).

Além da fusão com o EU-DEM, também foi considerada a possibilidade de utilizar o *dataset* fornecido pelo BKG (*Bundesamt für Kartographie und Geodäsie*), que disponibiliza diversos conjuntos de dados de elevação para a Alemanha. As resoluções de 200m e 1000m estão disponíveis gratuitamente, enquanto resoluções mais detalhadas, como as de até 5m, exigem pagamento. Devido à melhor resolução, optou-se pelo EU-DEM, que também cobre a área de interesse.

Os gráficos da Figura 6 apresentam as distribuições das variáveis do *dataset*. Observando a distribuição de distâncias, percebe-se uma concentração maior de amostras em distâncias mais curtas. Isso pode ser atribuído a dois possíveis motivos: primeiro, a

presença de uma boa cobertura de gateways na cidade, garantindo que sempre haja um gateway próximo para receber os dados; segundo, em distâncias maiores, há uma maior perda de pacotes, resultando em menos amostras nessas distâncias.

Figura 6 – Histograma das variáveis de entrada do *dataset* Mclab e da perda por propagação.



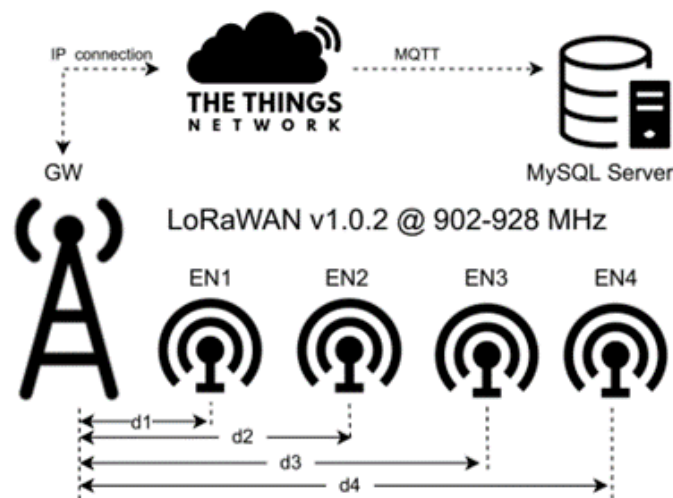
Fonte: do autor.

Ao analisar a distribuição das elevações, observa-se que a maioria das amostras está concentrada em torno de 60 metros, apesar da variação das elevações entre 45 e 180 metros. Poucos pacotes foram recebidos em altitudes acima de 90 metros. A altura da antena dos gateways não fornece informações úteis, pois cada gateway está localizado em alturas variadas, principalmente entre 10 e 30 metros, com apenas um gateway operando acima de 50 metros. Por fim, a distribuição das perdas por propagação indica que a maioria dos pacotes apresenta perdas acima de 120 dB. Nota-se uma queda abrupta próximo de 160 dB, que representa um ponto crítico onde os gateways não possuem mais sensibilidade para interpretar o sinal transmitido.

4.1.2 Dataset 2: Medellín - Nós fixos

Um dos trabalhos vistos na revisão bibliográfica (GONZÁLEZ-PALACIO *et al.*, 2023) utilizou um *dataset* coletado para avaliação do impacto das condições ambientais na propagação de sinais LoRa que havia sido publicado pelo autor anteriormente (GONZÁLEZ-PALACIO *et al.*, 2022). Uma das etapas deste trabalho foi a construção de um banco de dados que abrange o período de outubro de 2021 a março de 2022, totalizando 990.750 observações. A frequência de amostragem foi estabelecida em aproximadamente 15 segundos e utiliza quatro ENs dispostos em diferentes distâncias, variando de 2 a 8 km, comunicando com um GW LoRaWAN. Esses ENs transmitem dados para o GW, que, por sua vez, encaminha essas informações para um *network server*, neste caso, o The Things Network. A arquitetura pode ser vista na Figura 7.

Figura 7 – Arquitetura da rede desenvolvida por GONZÁLEZ-PALACIO *et al.* (2022).



Fonte: GONZÁLEZ-PALACIO *et al.* (2022)

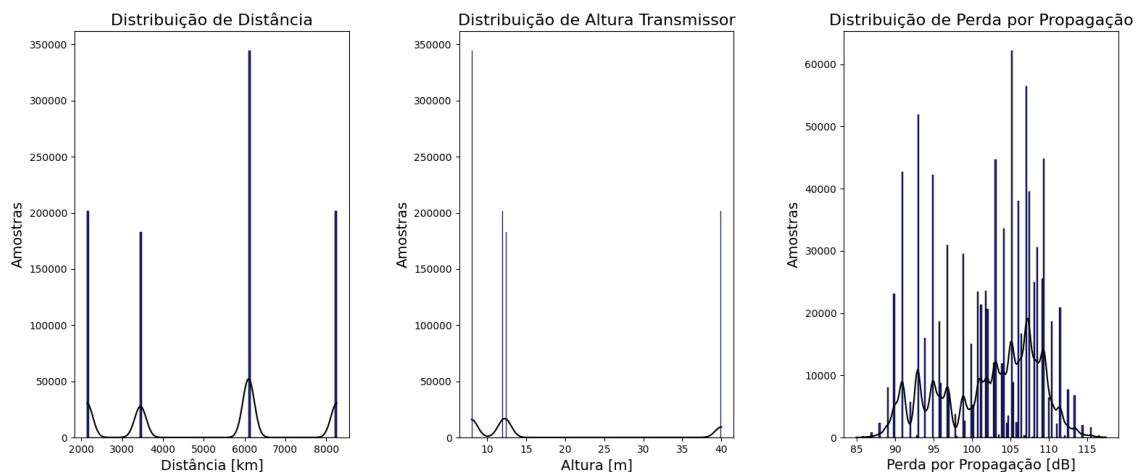
As categorias de dados englobam a identificação das amostras coletadas, as condições físicas e geométricas da instalação (como distância e altura da antena), e as características do aparato experimental, permitindo o cálculo das perdas de propagação, incluindo a potência do transmissor e o ganho de antena. Algumas variáveis ambientais também são registradas, como temperatura, umidade, pressão barométrica e concentração de material particulado no ar.

As métricas relacionadas à propagação de ondas de rádio, tais como RSSI, SNR e o tempo de transmissão (*Time on Air*, TOA), que representa o tempo que o EN permanece enviando dados para o GW, também são disponibilizadas. Além disso, dados previamente calculados são fornecidos para todas as amostras, incluindo a perda por propagação, calculada com base nas características do aparato experimental, e a energia consumida durante a transmissão.

O histograma das variáveis coletadas que aparecem em vários dos modelos para estimar as perdas de propagação, como a distância e a altura do transmissor, assim como

da perda de propagação do sinal medido, podem ser vistos na Figura 8. Por se tratar de uma coleta realizada apenas com nós fixos, aquelas características relacionadas ao EN possuem poucos dados diferentes, como a altura do transmissor e distância.

Figura 8 – Histograma das variáveis de entrada do *dataset* Medellín e da perda por propagação.



Fonte: do autor.

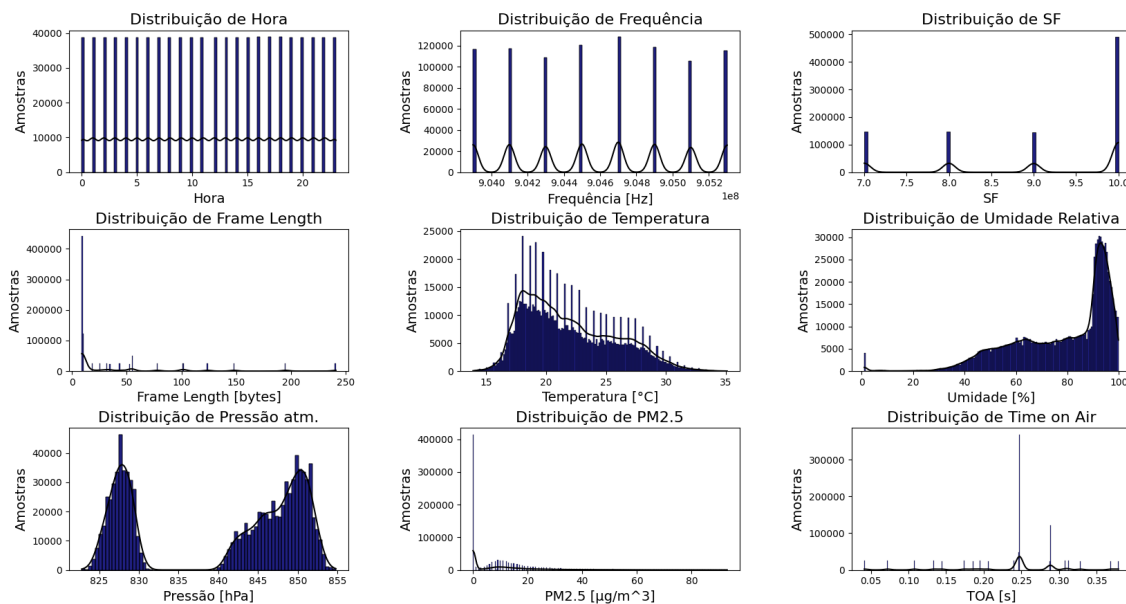
Na Figura 9 estão apresentadas as distribuições das demais variáveis presentes no dataset. Por possuir uma frequência de envio alta, a variável hora está bem distribuída, com quantidade de pacotes semelhantes para cada hora do dia. O histograma de frequência também segue uma distribuição uniforme, o que é esperado ao utilizar LoRaWAN. O SF10 predomina na base de dados, mas há também uma quantidade relevante de informações oriundas de transmissões em SF 7 a 9. O predomínio de dados de TOA mais elevados também é reflexo do SF mais concentrado nos valores mais altos, uma vez que maiores SFs levam a mais tempo para transmissão.

A distribuição de temperatura é mais contínua, com uma maior concentração de amostras entre 15°C a 35°C, conforme esperado pela coleta abrangendo mais de um período do ano e continuamente, levando em consideração variações ao longo do dia e sendo predominantemente um clima moderado. Já a umidade relativa mostra uma distribuição bimodal com uma distribuição mais uniforme entre 35% e 80%, e um pico de umidade próximo de 100%, provavelmente de dias com precipitação. Há pouca variação na pressão atmosférica, com picos bem definidos, evidenciando que os ENs estavam em diferentes altitudes.

4.2 Simulador LoRaEnergySim

O LoRaEnergySim é uma ferramenta de simulação voltada para avaliar o consumo de energia de redes LoRaWAN. O simulador foi desenvolvido com o objetivo principal de avaliar como os parâmetros de comunicação podem afetar o uso de energia, permitindo

Figura 9 – Histograma das demais variáveis presentes no dataset Medellín



Fonte: do autor.

que os usuários otimizem as configurações de rede para maximizar a eficiência energética. O framework do simulador utiliza Python 3 e bibliotecas específicas como SimPy, tornando a ferramenta compatível com diversos hardwares e sistemas operacionais.

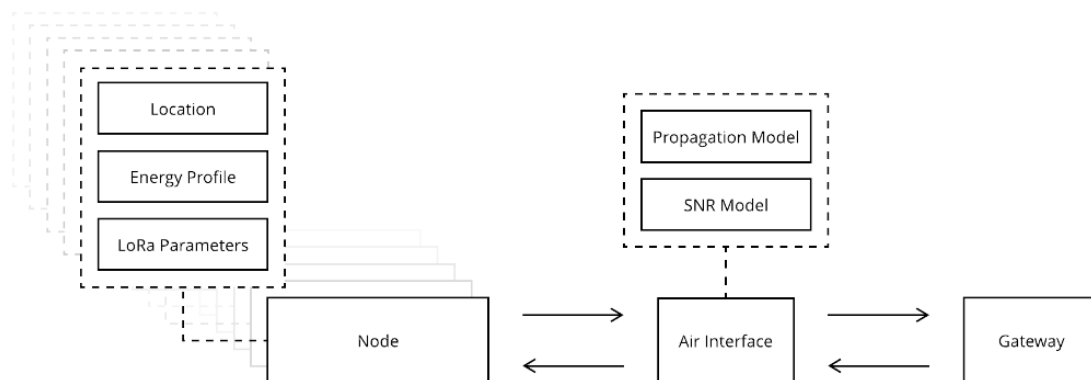
A ferramenta permite a configuração de vários parâmetros de rede para simular diferentes condições em uma rede LoRaWAN. Essas configurações podem incluir ajustes nos intervalos de transmissão, tamanhos de *payload* e modelos de consumo de energia para dispositivos. Os usuários podem experimentar diferentes configurações para observar como cada parâmetro influencia a eficiência energética e o desempenho da rede.

O Framework possui 4 componentes principais, que são o nó, o gateway, o pacote e a interface (canal RF). Os nós são os dispositivos finais simulados que transmitem dados. Eles são configurados para emular vários comportamentos de consumo de energia com base em sua atividade, como envio e recebimento de pacotes de dados. Também é neste módulo que as características de transmissão, como o fator de espalhamento e a largura de banda dos uplinks são definidas. O pacote é o módulo que simula aquilo que será transmitido na rede LoRa. Além dos dados de informação, também são representados os bytes de pré-âmbulo e *header* da rede. A verificação de CRC não foi implementada, uma vez que não há nenhuma distorção do sinal físico implementada na interface, apenas a perda de intensidade do sinal é considerada.

A interface no LoRaEnergySim é modelada para refletir a comunicação LoRa do mundo real. Ela incorpora a simulação de propagação de sinal, podendo levar em consideração fatores como frequência, fator de espalhamento e potência de transmissão. É neste módulo que os modelos de propagação podem ser alterados para que seja possível representar realisticamente o comportamento de perda por propagação de um sinal LoRa.

O fluxo de informações pode ser visto na Figura 10.

Figura 10 – Fluxo de dados do simulador LoRaEnergySim.



Fonte: BOR *et al.* (2016)

Por fim, o gateway atua como ponte entre os nós e o servidor de rede no LoRaWAN. Ele recebe transmissões de vários nós, as agrega e envia os dados para o servidor. Na simulação, ajuda a analisar como os comportamentos do gateway afetam o desempenho da rede e o consumo de energia. Também é neste módulo que são descartadas as mensagens que não chegam com potência suficiente para recepção, as mensagens com conflito de potência, frequência ou fator de espalhamento.

Após executar a simulação, é possível extrair informações detalhadas de cada nó por meio de um dicionário. Esse dicionário, chamado “series”, fornece dados do comportamento e do desempenho de cada nó na rede. As informações incluídas são as seguintes:

- *WaitTimeDC*: O tempo total de espera em modo *sleep* para respeitar o *duty cycle* da rede.
- *NoDLReceived*: O número de vezes em que o nó não recebeu mensagens de down-link.
- *UniquePackets*: O número de pacotes únicos enviados pelo nó.
- *TotalPackets*: O número total de pacotes enviados pelo nó, incluindo retransmissões, se houver.
- *CollidedPackets*: O número de pacotes que colidiram durante a transmissão.
- *RetransmittedPackets*: O número de pacotes que foram retransmitidos devido a colisões ou perda de pacotes.
- *TotalBytes*: O número total de bytes enviados pelo nó.
- *TotalEnergy*: A energia total consumida pelo nó durante a simulação.

- *TxRxEnergy*: A energia consumida pelo nó relacionada à transmissão e recepção de mensagens.

Além das informações disponíveis para cada nó, também é possível extrair dados relacionados ao gateway e à interface após a conclusão da simulação. Este conjunto de dados inclui:

- *BytesReceived*: A quantidade total de bytes recebidos pelo gateway durante a simulação.
- *DLPacketsLost*: O número de pacotes downlink perdidos devido a ocupação da rede.
- *PacketsReceived*: O número total de pacotes recebidos pelo gateway.
- *UniquePacketsReceived*: O número de pacotes únicos recebidos pelo gateway.
- *NumberOfPacketsCollided*: O número de pacotes que colidiram na interface aérea durante a simulação.
- *NumberOfPacketsOnAir*: O número total de pacotes que passaram pela interface.

4.3 Método para treinamento de modelos de predição que usam Inteligência Artificial

A implementação de todos modelos de propagação foi realizada em Python, visando avaliar e comparar o desempenho de diferentes abordagens. A escolha da linguagem Python proporciona facilidade na experimentação com distintas técnicas de modelagem, tanto clássicas quanto utilizando inteligência artificial. Também é a linguagem adotada pelo simulador de rede escolhido, permitindo a fácil migração dos modelos desenvolvidos.

O método desenvolvido para a predição de perdas em redes sem fio utilizando inteligência artificial é composto por várias etapas. Após o tratamento dos conjuntos de dados, foram implementados e avaliados os erros em modelos clássicos de propagação de perdas de sinal sem fio. Estes modelos, sendo o principal o *log-distance*, serviram de base para avaliação dos modelos que usam IA. A seguir, foram desenvolvidos modelos de predição de perdas utilizando algoritmos de regressão que empregam IA. Para os modelos onde isso era necessário, foi realizada uma avaliação sistemática dos hiperparâmetros, testando todas as combinações de valores dentro de um intervalo predefinido. Por causa disso, a aplicação dos modelos de IA adotou uma abordagem de treinamento, validação e teste, dividindo o conjunto de dados em 70% para o treinamento do modelo e 20% para a validação onde os melhores hiperparâmetros são escolhidos, e os últimos 10% foram

utilizados para avaliação do desempenho. Um exemplo disso é a árvore de decisão, onde o hiperparâmetro de profundidade da árvore foi testado no intervalo de 1 a 50. O valor que proporcionou o menor RMSE foi adotado para o modelo final. Para a floresta aleatória e para o XGBOOST, além da profundidade, também foi avaliada a quantidade de estimadores.

Para a criação da rede neural, foi utilizada a interface de aprendizado de máquina scikit-learn para treinamento, com o objetivo de prever a perda por propagação utilizando as mesmas variáveis de entrada dos demais modelos. A rede neural é construída com a ajuda da classe KerasRegressor, que permite integrar modelos Keras (TensorFlow) no fluxo de trabalho do scikit-learn.

Para automatizar a busca dos melhores hiperparâmetros, foi utilizado o GridSearchCV do scikit-learn. Foi definido um conjunto de parâmetros a serem explorados, incluindo o número de neurônios, otimizador e a função de ativação, vistos na Tabela 4. Durante a busca em grade, o GridSearchCV ajusta diversos modelos com as diferentes combinações de hiperparâmetros possíveis. Com os parâmetros escolhidos, 672 redes foram treinadas na busca pelo modelo com o melhor desempenho, avaliado utilizando RMSE como métrica.

Tabela 4 – Hiperparâmetros explorados para definição da rede neural.

Número de neurônios	Função de ativação	Otimizador
10	Softmax	SGD
20	Softplus	RMSprop
30	Softsign	Adagrad
40	ReLU	Adadelata
50	Tanh	Adam
100	Sigmoide	Adamax
200	Hard Sigmoide	Nadam
500	Linear	
1000		
2000		
5000		
10000		

Alguns itens da arquitetura permaneceram constantes, como a dimensão de entrada, a quantidade de neurônios da primeira camada oculta (a mesma da dimensão de entrada) e a quantidade de camadas ocultas, limitada em duas. O número de camadas ocultas não foi explorado pois, segundo o Teorema da aproximação universal, qualquer função contínua pode ser explorada com apenas uma camada oculta, desde que esta seja grande o suficiente. Além disso, a camada de saída também é fixa, com ativação linear e apenas

um neurônio.

Os parâmetros, assim como os valores de MAE e RMSE, foram registrados em um arquivo JSON, permitindo uma análise detalhada das melhores combinações. Uma quantidade menor de neurônios é desejável, pois tanto o tempo de treinamento quanto o de predição é menor, então mais combinações foram avaliadas além do melhor modelo encontrado. A ordenação dos resultados de acordo com o menor RMSE foi adotada, e os 20 melhores modelos foram isolados para uma análise mais aprofundada.

4.4 Ensaios com simulador

Os ensaios foram conduzidos em três etapas principais: geração de redes, avaliação de algoritmos e execução de ensaios com os melhores algoritmos. Foram geradas redes com 10, 100 e 1000 nós, com tamanhos de célula de 100, 1000 e 10000 metros de lado, com o objetivo de avaliar em quais cenários há diferenças significativas entre os modelos já implementados e os novos modelos utilizando IA.

Todos os algoritmos foram avaliados com base no tempo de processamento. Foi criado um indicador de eficiência definido como $\text{Eficiência} = \text{RMSE} \times \text{Tempo de Processamento}$. O melhor algoritmo é aquele com o menor valor deste indicador. Algoritmos que não apresentaram uma melhora significativa na predição em relação ao tempo de processamento foram eliminados.

Os melhores algoritmos foram selecionados para ensaios adicionais. Esses ensaios foram projetados para avaliar as diferenças de predição utilizando algoritmos com erros menores comparados com os algoritmos clássicos normalmente utilizados. Os ensaios incluíram a variação do fator de espalhamento, as dimensões da célula e a quantidade de nós em uma célula de dimensão fixa (10.000m x 10.000m).

5 IMPLEMENTAÇÃO

Neste capítulo, são detalhados os resultados obtidos com a implementação dos modelos utilizando os conjuntos de dados MCLAB e Medellín. Posteriormente, é demonstrada a implementação de alguns desses modelos no simulador LoRaEnergySIM.

5.1 Modelos clássicos e com IA para *dataset* MCLAB

5.1.1 Espaço Livre

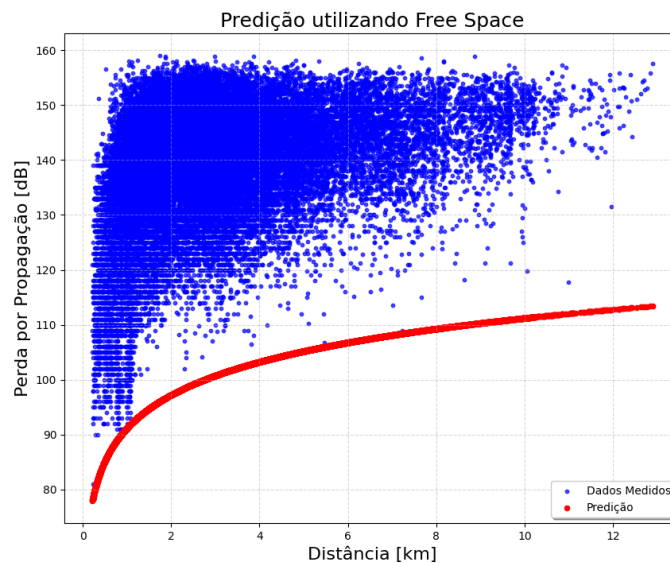
A aplicação do modelo de propagação em espaço livre, visto na Equação 3, ao *dataset* de nós móveis coletados em ambiente urbano, revelou erros significativos, observados na comparação entre as medições reais e as estimativas do modelo. O MAE de 39,72 dB e o RMSE de 40,93 dB indicam a dificuldade do modelo em representar a situação das medições. Estes erros são altos porque o modelo de propagação em ambiente livre não leva em consideração, além de maiores perdas por interação com a matéria, os efeitos da interferência de obstáculos, como edifícios e árvores. Em uma cidade, esses obstáculos são abundantes e podem causar grandes variações na propagação do sinal LoRa.

Como o modelo não considera nenhuma outra entrada além da distância, amostras com condições muito diferentes de altitude, por exemplo, também não são diferenciadas. Os dados reais de perda por propagação e a predição do modelo podem ser vistos na Figura 11.

5.1.2 Modelo de Egli

O modelo de Egli é derivado de diversas medições experimentais realizadas em múltiplas frequências, atingindo 1000 MHz, portanto, na abrangência das frequências LoRa de ambos *datasets* utilizados. Com um MAE de 12,96 dB e um RMSE de 15,35 dB, o modelo apresenta um avanço considerável em relação ao modelo de propagação em espaço livre, ainda que necessite de fatores de correção relacionados a características do rádio e do meio de propagação. Um destes parâmetros, encontrado na Equação 5 é a frequência, corrigindo a atenuação em função da interação com o meio, embora tenha sido considerada apenas a frequência nominal do LoRa, dada a pequena diferença entre os canais, o

Figura 11 – Modelo de propagação de caminho livre aplicado ao *dataset* MCLAB.



Fonte: do autor.

que não impacta significativamente nas métricas de desempenho.

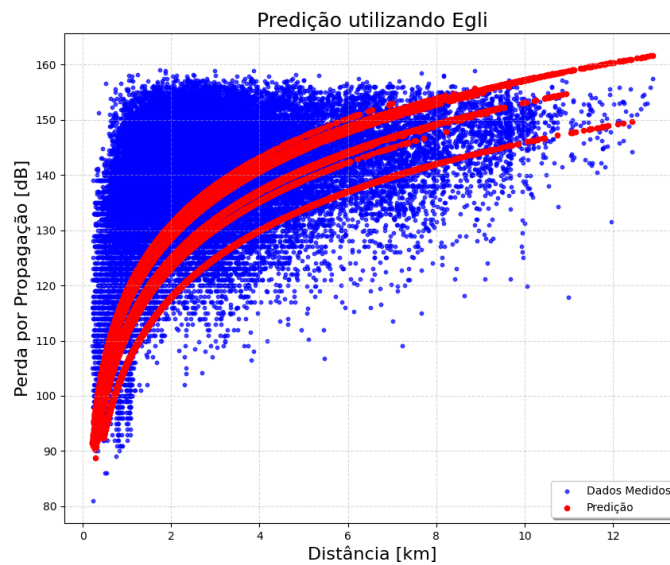
O modelo também leva em consideração a altura da antena do transmissor e do receptor. No caso do receptor, essa altura é considerada constante, pois o modelo não prevê variação de altitude. No entanto, a presença de diferentes alturas de antena para cada gateway introduz variações nas curvas de decaimento, que já são observadas no modelo de espaço livre. Esses fatores de correção alteram o formato e valor médio da curva de atenuação vistos na Figura 12, que segue com um formato relacionado à quarta potência da distância, contribuindo para a redução do erro do modelo.

5.1.3 Okumura-Hata

A aplicação do Modelo de Okumura-Hata ao conjuntos de dados MCLAB proporcionou um MAE de 8,63 dB e um RMSE de 10,77 dB. Nessa implementação, optou-se pela utilização da equação destinada a áreas urbanas (Equação 8), com o fator $a(h_r)$ adequado para cidades pequenas e médias (Equação 9), visto ser o que melhor representa o ambiente de estudo.

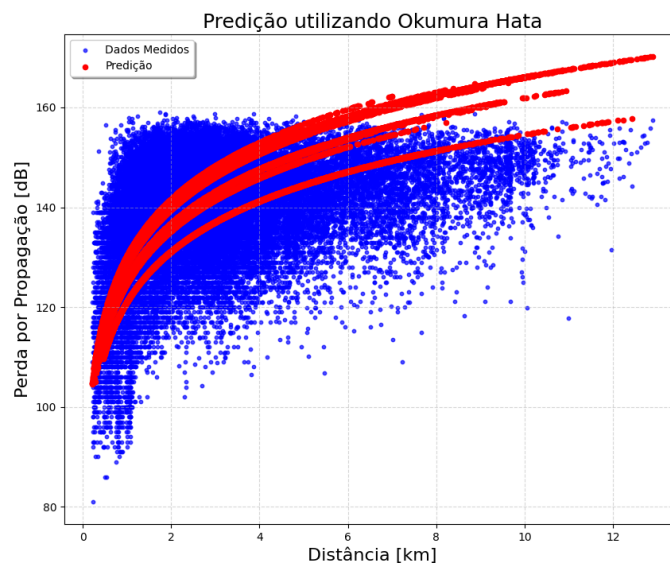
Dada a pequena variação entre os canais de transmissão LoRaWAN utilizados nos conjuntos de dados, novamente apenas a frequência nominal foi considerada (868MHz), pois não há impacto significativo nas métricas de desempenho do modelo. Observou-se que os erros foram mais pronunciados para distâncias superiores, conforme visto na Figura 13, sugerindo que o limite de sensibilidade do receptor pode ter influenciado significativamente. Isto resulta na falha de entrega de diversos pacotes ao gateway, o que pode impactar diretamente na média dos erros.

Figura 12 – Modelo de propagação de Egli aplicado ao *dataset* MCLAB.



Fonte: do autor.

Figura 13 – Modelo de propagação de Okumura-Hata aplicado ao *dataset* MCLAB.



Fonte: do autor.

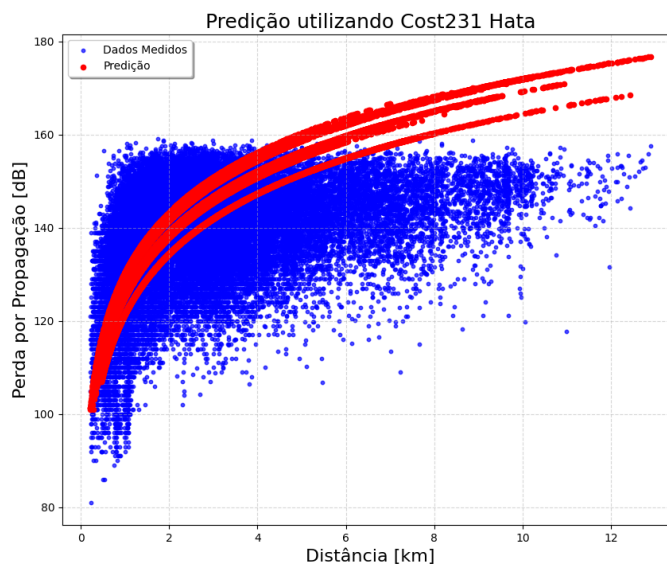
5.1.4 COST231-Hata

A avaliação do desempenho do Modelo de COST231-HATA revelou resultados com um MAE de 10,13 dB e uma RMSE de 12,67 dB. Este modelo, uma expansão do Okumura-Hata para frequências superiores a 1500 MHz, incorpora um fator de correção de 3 dB em vez de uma equação específica para áreas urbanas. No entanto, o fator $a(h_r)$ considerado foi o apropriado para cidades pequenas e médias, abordagem semelhante ao modelo de

Okumura-Hata.

Assim como no modelo de Okumura-Hata, a variação de frequência entre os canais foi considerada desprezível, optando-se pela utilização da frequência nominal dos dispositivos (868 MHz). No entanto, assim como observado no modelo anterior, os erros foram mais expressivos para distâncias superiores, conforme visto na Figura 14 sugerindo que o limite de sensibilidade do receptor pode ter influenciado os resultados. Este comportamento é importante para a interpretação dos resultados, uma vez que a não entrega de pacotes ao gateway pode ter impactado a média dos erros. O erro maior que o encontrado para o modelo de Okumura-Hata já era esperado, uma vez que o COST231-Hata é projetado para frequências superiores àsquelas empregadas nos conjuntos de dados, que operam em faixas sub-GHz.

Figura 14 – Modelo de propagação COST231-HATA aplicado ao *dataset* MCLAB.



Fonte: do autor.

5.1.5 Log-Distance

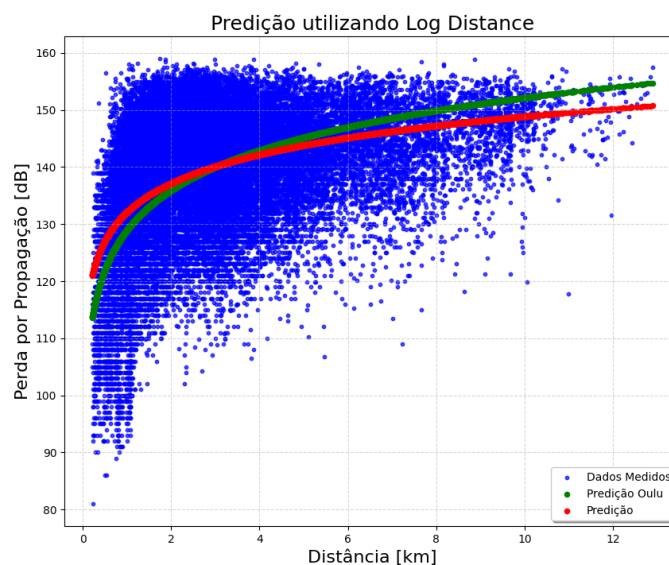
Duas abordagens foram exploradas com o *dataset* MCLAB utilizando o modelo *log-distance*. Na primeira abordagem, os parâmetros da equação foram diretamente aplicados conforme as medições realizadas em Oulu (PETAJARVI *et al.* (2015)), resultando em um MAE de 8,13 dB e uma RMSE de 10,04 dB.

Na segunda variação, reconhecendo a capacidade do modelo *log-distance* de ser adaptado por meio de medições em campo, o conjunto de dados foi dividido em 70% para ajuste dos parâmetros da curva e 10% para teste da equação encontrada, que são os conjuntos de treinamento e teste dos modelos inteligentes, respectivamente. O conjunto de validação, que correspondente a 20% dos dados, é utilizado apenas para ajuste de hi-

perparâmetros nos modelos inteligentes e não foi utilizado aqui, para que fosse possível comparar os modelos clássicos com os modelos inteligentes na sequência.

Conforme esperado, para esta segunda abordagem os erros foram menores, apresentando um MAE de 7,83 dB e um RMSE de 9,82 dB. Esta abordagem, embora mais complexa devido à necessidade de medições em campo, é mais eficaz na personalização do modelo conforme as características específicas dos conjuntos de dados analisados. Ambas curvas encontradas podem ser vistas na Figura 15.

Figura 15 – Modelo de propagação log-distance aplicado ao *dataset* MCLAB.



Fonte: do autor.

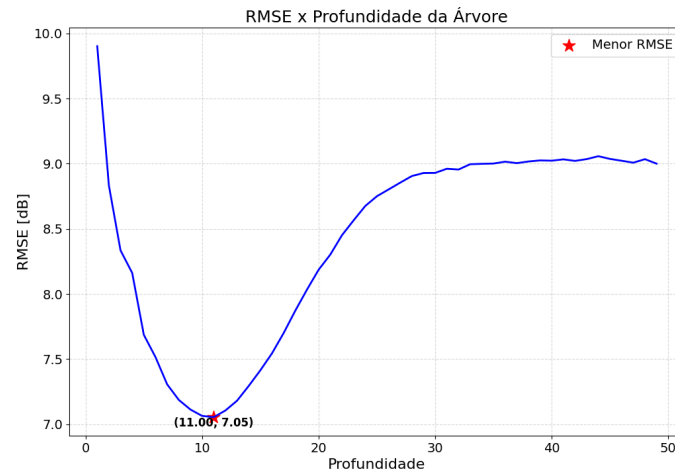
O modelo *log-distance*, que se destaca por sua baixa exigência computacional e dependência apenas da distância entre transmissor e receptor, é uma alternativa viável quando a coleta de dados em campo é possível. Por outro lado, a ausência de outros parâmetros de entrada não permite outros ajustes no modelo, tais como a variação de altura do transmissor e receptor para uma mesma distância entre nós.

5.1.6 Árvore de Decisão para regressão

Uma avaliação sistemática do hiperparâmetro profundidade da árvore foi conduzida no intervalo de 1 a 50. O valor que proporcionou o menor RMSE, identificado como profundidade 11, foi adotado para o modelo final. A avaliação deste hiperparâmetro pode ser vista na Figura 16.

O desempenho resultante foi superior a todos modelos clássicos, atingindo um MAE de 5,46 dB e um RMSE de 7,05 dB durante treinamento e 5,57 dB e 7,14 dB, respectivamente, com dataset de teste. Ao contrário dos modelos clássicos, a regressão por árvore de decisão demonstrou capacidade de aprender nuances geográficas específicas. Isto é

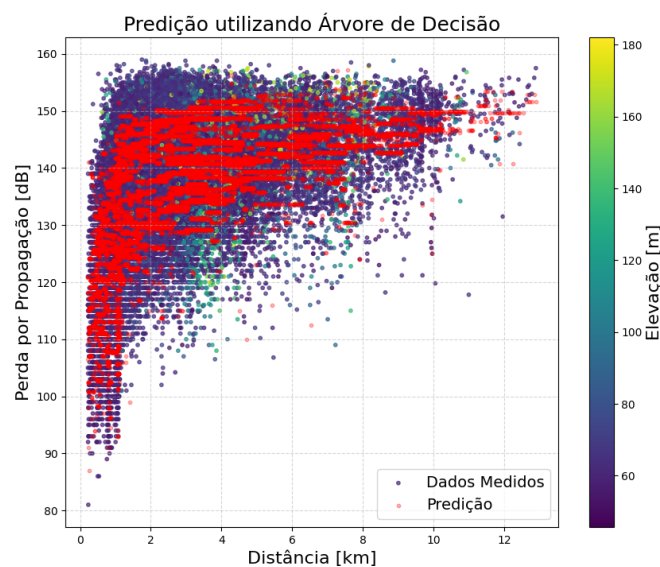
Figura 16 – Avaliação do hiperparâmetro profundidade para árvore de decisão.



Fonte: do autor.

evidenciado pelo comportamento diferenciado em pontos distintos da cidade que possuem a mesma distância entre gateway e dispositivo, revelando a habilidade do modelo em capturar variações relacionadas à altitude e relevo. O gráfico da predição sobreposto às medições reais pode ser visto na Figura 17.

Figura 17 – Predição de perda do caminho utilizando árvore de decisão para regressão no *dataset* MCLAB.

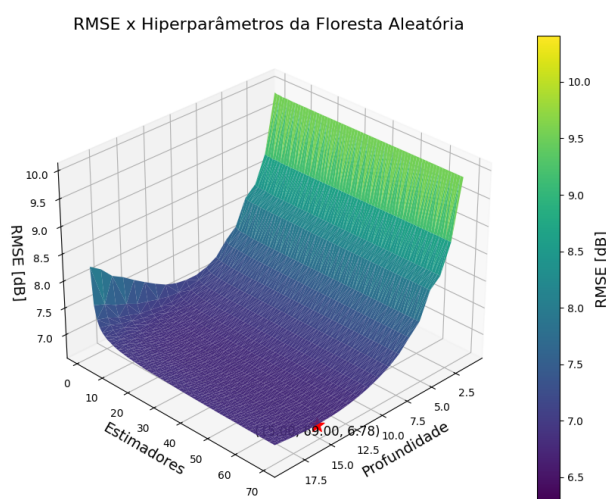


Fonte: do autor.

5.1.7 Floresta Aleatória para Regressão

Utilizando a mesma abordagem da árvore de decisão, o *dataset* também foi dividido na mesma proporção para treinamento, validação e teste do algoritmo de floresta aleatória. Dois hiperparâmetros foram explorados, assim como sua relação, que são a profundidade da árvore e a quantidade de estimadores utilizados. Uma avaliação para todas as combinações de estimadores entre 1 e 70 e profundidade entre 1 e 20 foram explorados e podem ser vistos na Figura 18.

Figura 18 – Avaliação dos hiperparâmetros profundidade e estimadores para a floresta aleatória.



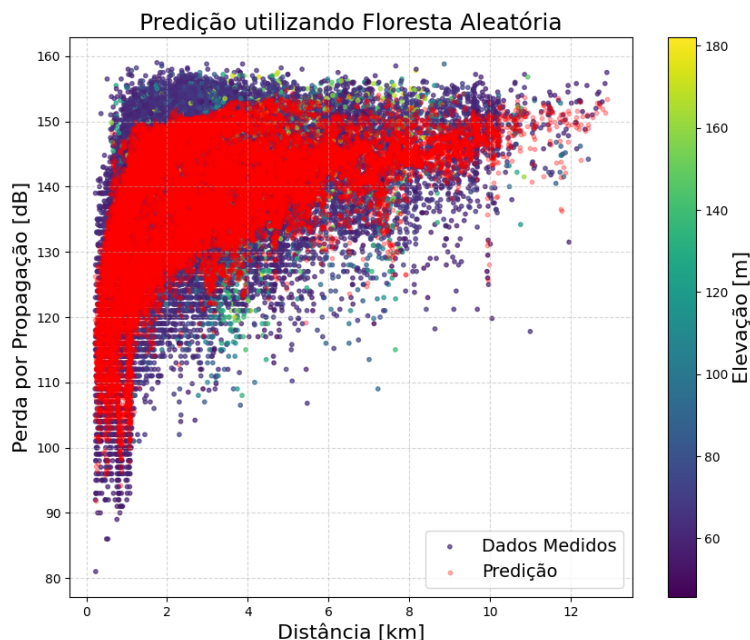
Fonte: do autor.

Desconsiderando o custo computacional e escolhendo os hiperparâmetros apenas pelo menor RMSE, o menor erro foi encontrado para 70 estimadores e 18 de profundidade máxima da raiz até o nó mais extremo. Para esta configuração de hiperparâmetros, os resultados finais exibiram um MAE de 5,23 dB e um RMSE de 6,78 dB durante o treinamento e resultaram em um desempenho de 5,28 e 6,82 dB no teste. Assim como observado na árvore aleatória, a Figura 19 mostra que foi possível capturar um comportamento das perdas de propagação que vai além da distância entre transmissor e receptor.

5.1.8 Support Vector Regression

A implementação do SVR foi feita considerando diferentes kernels para avaliação do desempenho do modelo. O SVR, embora tenha apresentado o segundo maior tempo de treinamento, ficando atrás apenas da rede neural, não obteve o melhor resultado em termos de métricas de erro e por isso as possibilidades de otimização de tempo de processamento não foram exploradas. Dentre os quatro kernels testados, o RBF se destacou como o melhor desempenho, com um MAE de 6,39 dB e um RMSE de 8,21 dB.

Figura 19 – Predição de perda do caminho utilizando floresta aleatória para regressão no *dataset* MCLAB.



Fonte: do autor.

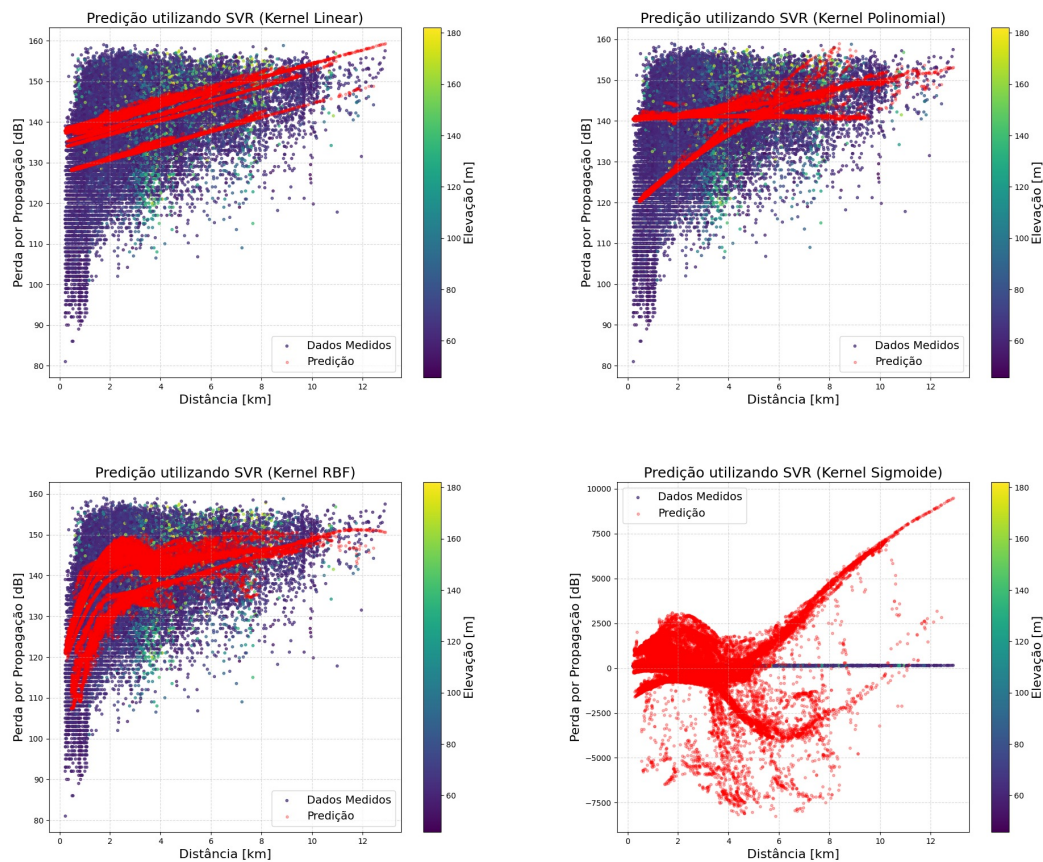
A análise dos outros kernels revelou algumas peculiaridades. O kernel Linear obteve um MAE de 7,45 dB e um RMSE de 9,56 dB, indicando um desempenho razoável, com erros abaixo dos modelos clássicos, mas não tão robusto quanto o RBF. O kernel polinomial apresentou resultados semelhantes ao Linear, com um MAE de 7,36 dB e um RMSE de 9,51 dB. Por fim, o kernel Sigmoide demonstrou resultados pouco plausíveis, com um MAE de 1.177 dB e um RMSE de 1.835 dB, sugerindo uma incompatibilidade significativa com as características do conjunto de dados analisado. O resultado com todos os kernels pode ser visto na Figura 20.

5.1.9 Regressão Lasso

A implementação da Regressão Lasso partiu de uma varredura do hiperparâmetro alfa para determinar o impacto no desempenho do modelo. O alfa foi avaliado em função do RMSE, e foi observado que os menores erros foram encontrados quando o valor estava próximo a zero, conforme visto na Figura 21. Os resultados finais apresentaram um MAE de 7,52 dB e um RMSE de 9,47 dB.

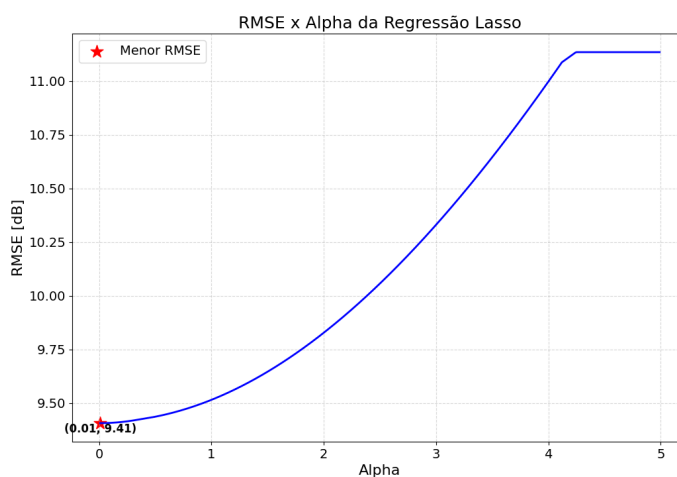
Comparando com o SVR Linear, os valores de erro obtidos pela Regressão Lasso foram bastante semelhantes. Essa similaridade é esperada, pois ambos os modelos compartilham uma resposta linear. Embora os resultados sejam comparáveis, deve-se considerar que o comportamento da Regressão Lasso poderia se destacar de maneira mais significativa se houvesse mais atributos de entrada disponíveis para o modelo pois esta regressão

Figura 20 – Predição de perda do caminho utilizando SVR no *dataset* MCLAB.



Fonte: do autor.

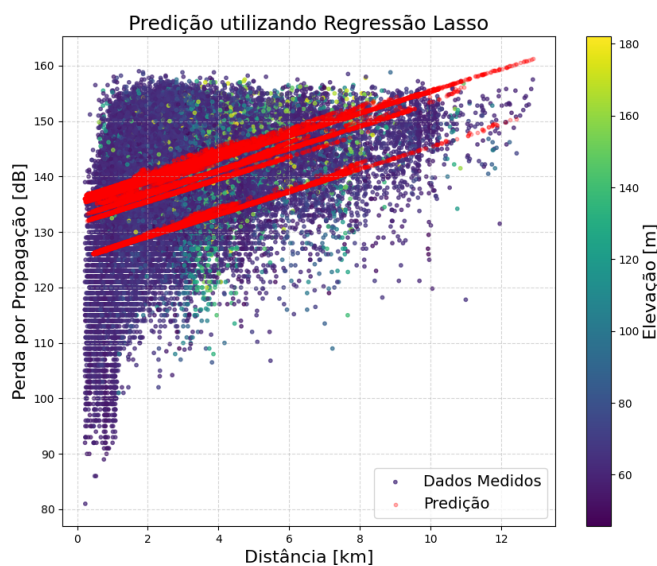
Figura 21 – Avaliação do hiperparâmetro alpha para a Regressão Lasso.



Fonte: do autor.

é conhecida por oferecer desempenho aprimorado quando há um grande número de atributos. O resultado da predição pode ser observado na Figura 22.

Figura 22 – Predição de perda do caminho utilizando regressão Lasso no *dataset* MCLAB.



Fonte: do autor.

5.1.10 XGBoost

O desenvolvimento do modelo de predição utilizando o algoritmo XGBoost iniciou com uma avaliação dos hiperparâmetros, particularmente a profundidade da árvore e o número de estimadores, visto na Figura 23. Ao explorar diferentes configurações, o melhor desempenho durante o teste de hiperparâmetros foi alcançado com uma profundidade de 11 e 39 estimadores, com um RMSE de 6,79 dB. Os resultados finais, com dataset de teste, exibiram um MAE de 5,32 dB e um RMSE de 6,85 dB.

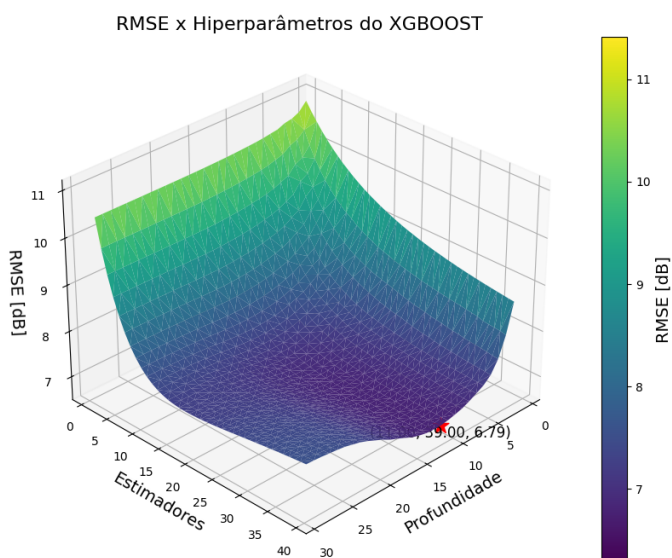
O XGBoost, sendo um modelo baseado em árvores de decisão mas otimizado para evitar *overfitting* e melhor desempenho computacional utilizando de paralelismo para processamento, alcançou valores de erro comparáveis aos obtidos pela Floresta Aleatória, outro modelo baseado em árvores de decisão. O resultado da predição pode ser observado na Figura 24.

5.1.11 Rede neural

Para o caso específico deste *dataset*, a implementação da rede neural utilizou 3 neurônios de entrada, assim como 3 neurônios na primeira camada oculta. A quantidade de neurônios da segunda camada oculta foi testada como um hiperparâmetro, conforme visto na Tabela 4. Um esboço genérico da rede avaliada é visto na Figura 25.

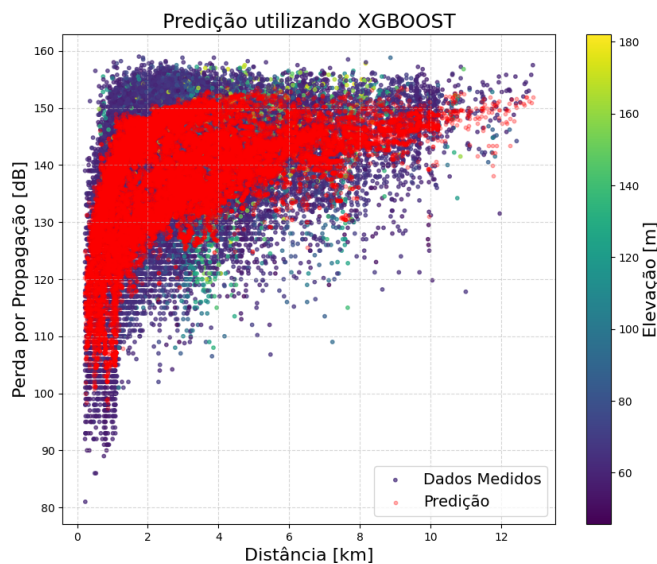
A configuração ótima de hiperparâmetros utilizou uma camada de 10.000 neurônios, aplicando a função de ativação ReLU e o otimizador Adamax. Essa combinação resultou em um MAE de 6,25 dB e um RMSE de 7,83 dB durante o teste e 6,63 e 8,25 dB, respectivamente, aplicando os dados de teste. Dentre as 20 melhores configurações, a

Figura 23 – Avaliação dos hiperparâmetros profundidade e estimadores para o XGBoost.



Fonte: do autor.

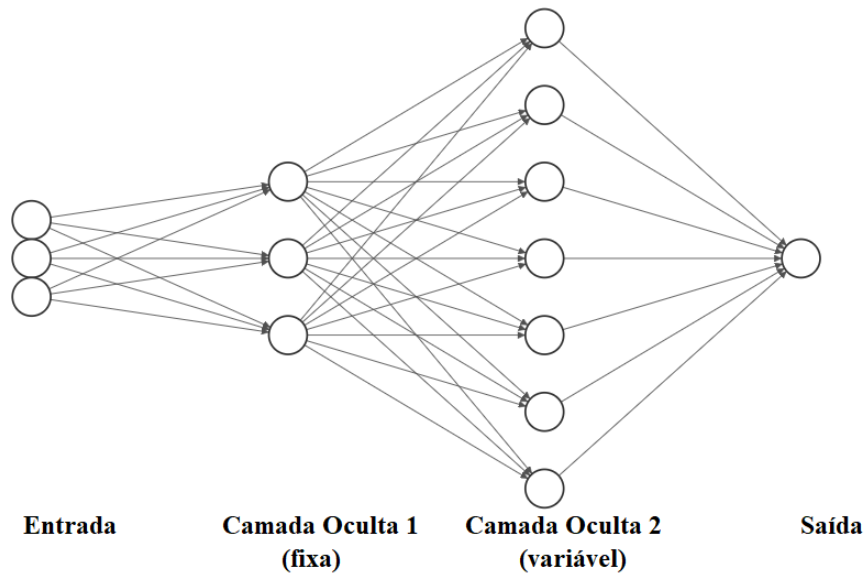
Figura 24 – Predição de perda do caminho utilizando XGBoost no *dataset* MCLAB.



Fonte: do autor.

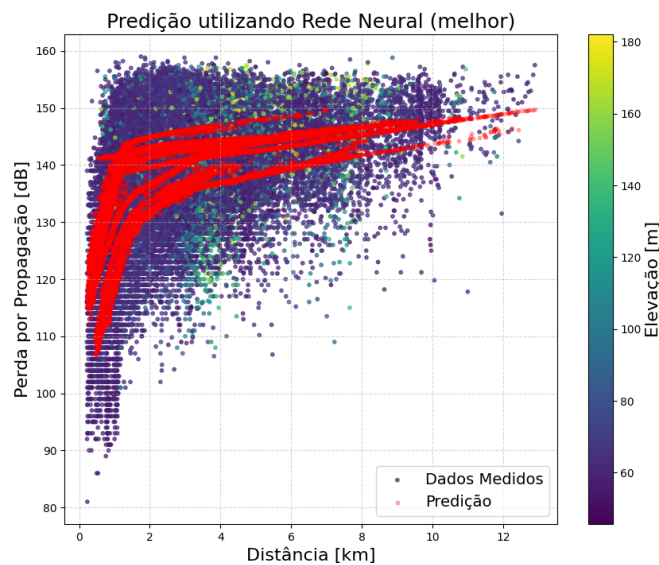
camada com a menor quantidade de neurônios consistiu em 100 unidades, adotando a função de ativação tangente hiperbólica e o otimizador RMSprop. Essa combinação atingiu um MAE de 6,55 dB e um RMSE de 8,20 dB com dataset de teste. O treinamento foi realizado com 20 épocas, com um tamanho de lote (*batch size*) de 100. O resultado final da predição é visto na Figura 26.

Figura 25 – Topologia com camada com nós variáveis explorada como hiperparamêtro.



Fonte: do autor.

Figura 26 – Predição de perda do caminho utilizando Rede Neural no *dataset* MCLAB.



Fonte: do autor.

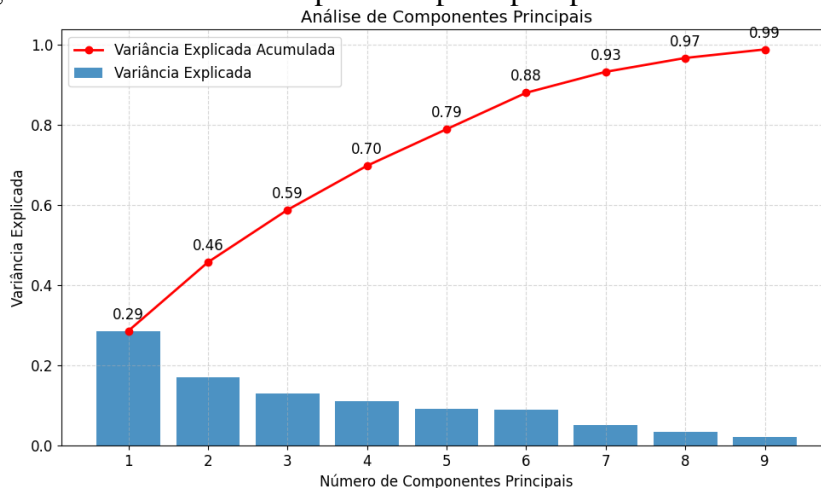
5.2 Modelos clássicos e com IA para *dataset* Medellín

5.2.1 Fatores significativos para predição de perda por propagação

Duas abordagens foram adotadas para explorar os fatores significativos dos dados. Na primeira, a aplicação da Análise de Componentes Principais (PCA) foi realizada na tentativa de identificar possíveis padrões subjacentes e reduzir a dimensionalidade dos atributos. No entanto, os resultados foram inconclusivos, indicando que a variância não

foi adequadamente explicada por um número reduzido de componentes, conforme visto na Figura 27. Além disso, observando os novos componentes criados pelo PCA, nenhuma entrada emergiu como predominantemente influente, sugerindo uma interconexão complexa entre as variáveis e sem dependência de uma única entrada.

Figura 27 – Análise de componentes principais para o *dataset* Medellín.



Fonte: do autor.

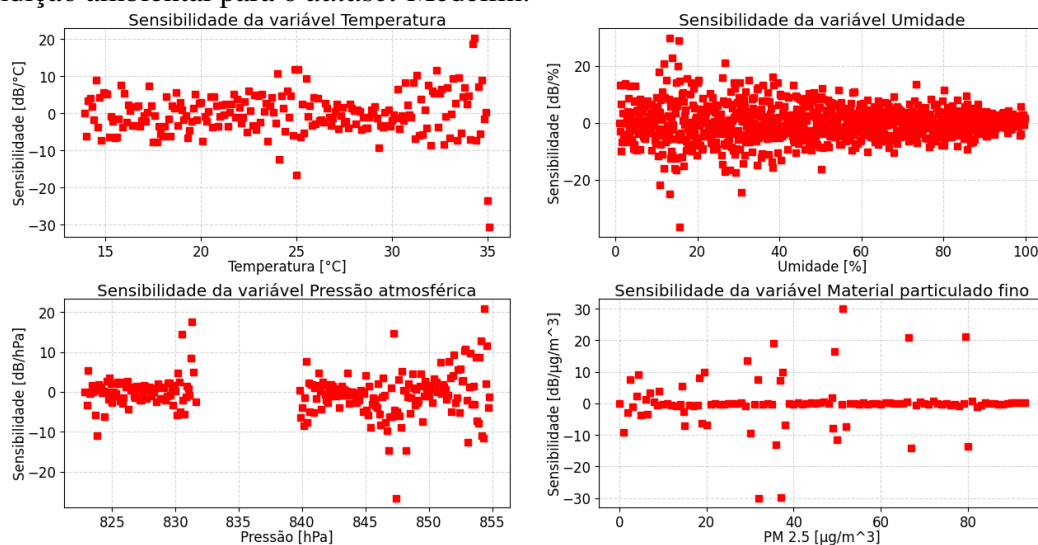
A segunda abordagem envolveu o cálculo da sensibilidade de cada variável por meio da derivada numérica considerando as demais variáveis como constantes, com os resultados visualizados graficamente. Esta análise permite identificar padrões de comportamento e destacar possíveis relações entre variáveis.

No contexto das condições ambientais, incluindo temperatura, umidade, pressão atmosférica e a concentração de material particulado fino ($PM_{2.5}$), a análise de sensibilidade revelou uma dispersão ao redor do eixo X nos gráficos, indicando que essas variáveis não exercem um impacto significativo na sensibilidade. Tal resultado era esperado, uma vez que a frequência da rede LoRa é relativamente baixa no contexto de interação com a matéria, minimizando a interação dessas variáveis ambientais com o sinal transmitido. A Figura 28 apresenta os gráficos correspondentes a cada uma dessas grandezas, fornecendo uma visualização da falta de relação com a perda por propagação.

Quanto às variáveis relacionadas à posição no espaço e no tempo do transmissor, como a hora de transmissão, a altura do transmissor e a distância, observa-se que a distância tem uma influência significativa na perda de propagação, como corroborado por diversos modelos. A altura da antena, no entanto, apresenta resultados inconclusivos, possivelmente devido à visada entre o transmissor e o receptor, o que reduz o impacto dessa variável. A variável "hora" exibe dispersão ao redor do eixo X, indicando uma relação menos definida com a perda por propagação (ver Figura 29).

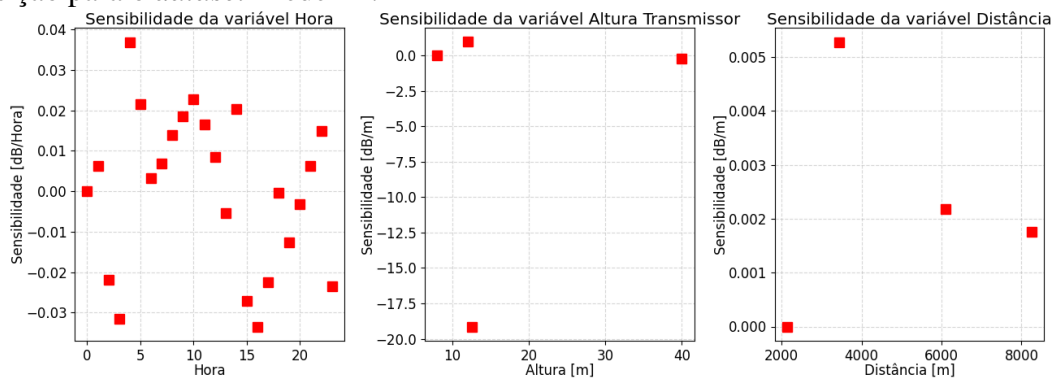
Considerando as variáveis relacionadas à transmissão, como frequência do canal, tempo no ar, tamanho do frame em bytes e fator de espalhamento, apenas o fator de espalhamento exerce influência significativa, mostrando que quanto maior o fator, maior a perda

Figura 28 – Sensibilidade da perda de propagação em função das variáveis relacionadas a condição ambiental para o *dataset* Medellín.



Fonte: do autor.

Figura 29 – Sensibilidade da perda de propagação em função das variáveis relacionadas a posição para o *dataset* Medellín.



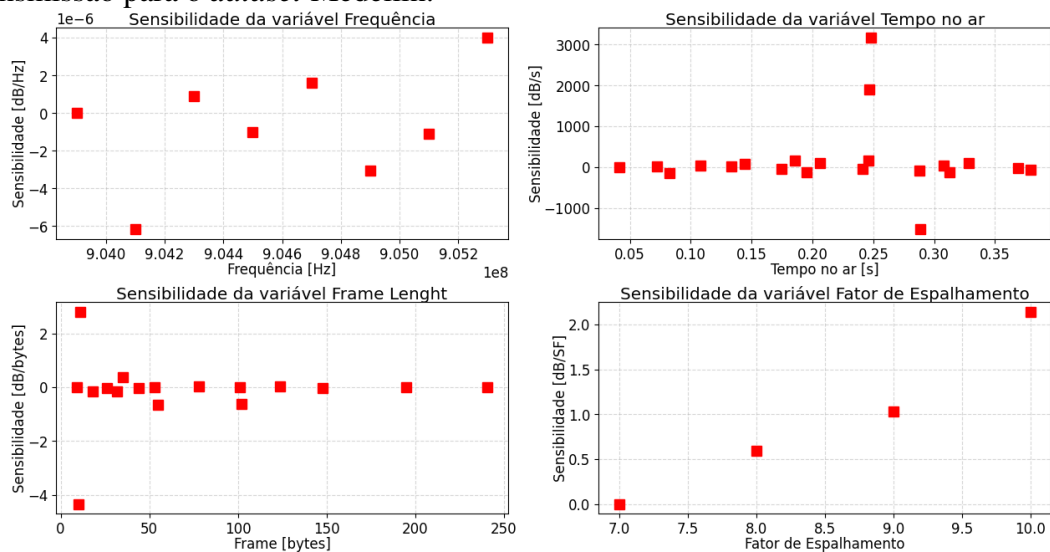
Fonte: do autor.

por propagação. Como o fator de espalhamento influencia diretamente no tempo no ar, esta variável foi incluída na análise, porém esta não parece estar diretamente relacionada com perdas por propagação, conforme visto na Figura 30. O tamanho do quadro, que também tem influência no tempo no ar, não parece ser uma variável significativa na perda por propagação. O impacto das variáveis na predição foi explorado utilizando o modelo de árvores aleatórias e pode ser visto nas próximas seções.

5.2.2 Modelos clássicos

O modelo de espaço livre, que assume condições ideais sem obstruções, exibiu um desempenho bem diferente em relação ao *dataset* MCLAB. Com um MAE de 3,23 dB e um RMSE de 3,99 dB, fica claro que a linha de visada direta entre o transmissor e o receptor é predominante neste ambiente urbanos, contribuindo para uma propagação mais

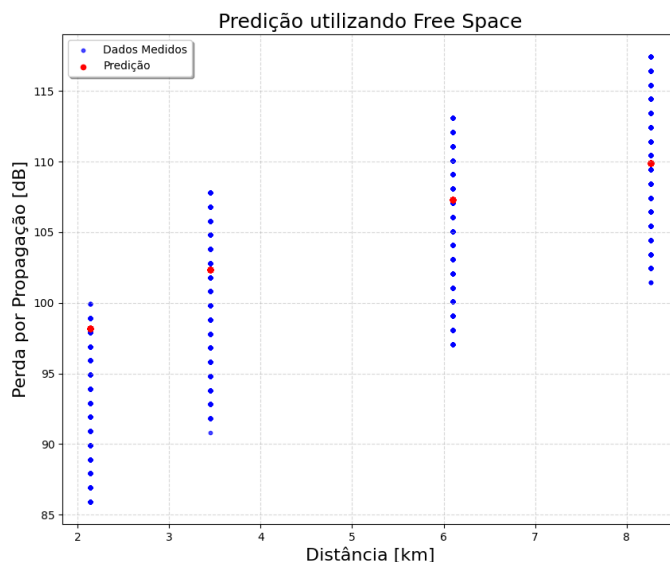
Figura 30 – Sensibilidade da perda de propagação em função das variáveis relacionadas a transmissão para o *dataset* Medellín.



Fonte: do autor.

próxima do cenário ideal. O resultado medido e previsto pode ser observado na Figura 31.

Figura 31 – Modelo de propagação de caminho livre aplicado ao *dataset* Medellín.



Fonte: do autor.

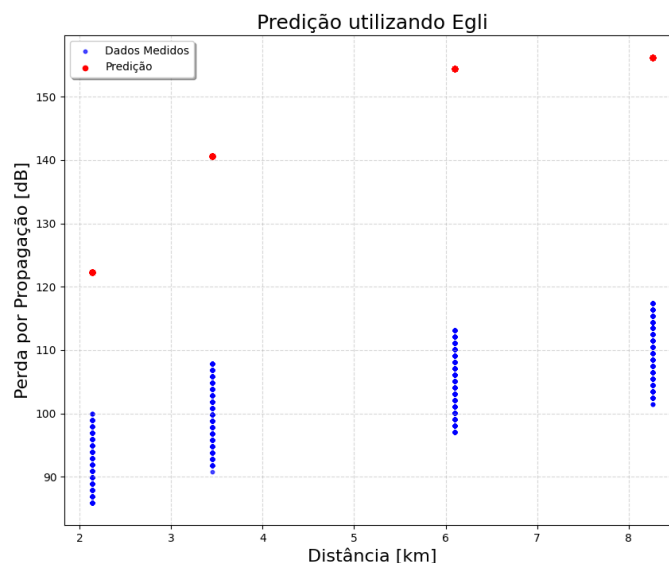
Ao aplicar o modelo de *log-distance* com a curva ajustada pelos próprios dados coletados, os resultados indicaram uma boa adaptação com MAE de 1,99 dB e RMSE de 2,46 dB. Como essa abordagem considera o decaimento logarítmico do sinal com a distância, um formato de curva semelhante ao da propagação livre, era esperado um bom resultado com os parâmetros ajustados. No entanto, ao utilizar os parâmetros do modelo de Oulu,

encontrou-se um MAE de 42,22 dB com RMSE de 42,30 dB.

O modelo de Okumura-Hata, também desenvolvido para ambientes urbanos, apresentou resultados insatisfatórios, com MAE e RMSE superiores a 50 dB. Essa disparidade destaca que as condições de propagação observadas divergem significativamente das premissas urbanas deste modelo. De maneira semelhante ao modelo de Okumura-Hata, o modelo COST231-Hata, uma extensão para frequências abaixo de 1500 MHz, mostrou-se inadequado para os dados coletados, com MAE próximo a 57 dB e RMSE superior a 57 dB. Esses resultados sugerem uma limitação desses modelos em cenários de propagação livre, como os encontrados no conjunto de dados.

O modelo de Egli, que leva em consideração a frequência, altura das antenas e outras características, também apresentou desafios na adaptação aos dados, com MAE de 42,99 dB e RMSE de 43,69 dB. Esses resultados indicam que a complexidade do modelo pode não ser apropriada para os cenários de coleta de dados analisados. Todos os modelos com erros altos apresentaram formatos de curva semelhantes quando plotados em gráfico, como pode ser visto na Figura 32 com a predição usando o modelo de Egli.

Figura 32 – Modelo de propagação de Egli aplicado ao *dataset* Medellín.



Fonte: do autor.

5.2.3 Árvore de Decisão para regressão

A árvore de decisão foi escolhida como algoritmo para testar diferentes combinações de variáveis de entrada, uma vez que apresentou bons resultados na predição de nós móveis e possui um tempo de treinamento rápido, permitindo a avaliação de várias combinações. A variável "distância" foi mantida em todas as configurações, pois é um fator significativo conhecido e utilizado em todos os modelos. Quando utilizada isoladamente,

atinge-se um RMSE de 2,431 dB. As demais variáveis foram adicionadas individualmente, com resultados sendo observados na Tabela 5.

Tabela 5 – RMSE para diferentes combinações de variáveis de entrada do *dataset* Medellín para árvore de decisão.

Variáveis	RMSE [dB]
Distância + Material Particulado Fino	2,279
Distância + Tempo no Ar	2,353
Distância + SF	2,354
Distância + Umidade Relativa	2,358
Distância + Tamanho do Frame	2,368
Distância + Temperatura	2,372
Distância + Frequência	2,376
Distância + Pressão Atmosférica	2,402
Distância + Hora	2,430
Distância + Altura da Antena	2,431

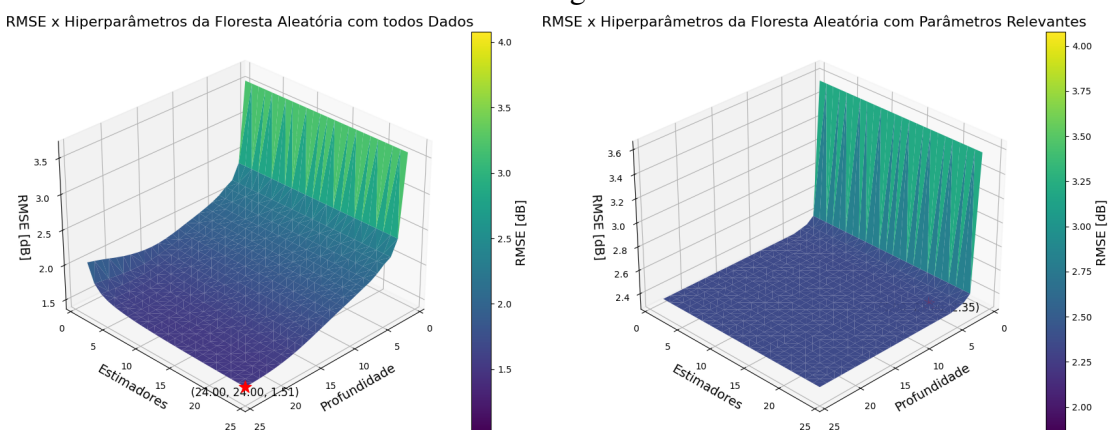
Alguns fatores como hora, altura da antena e pressão atmosférica parecem não trazer novas informações para o modelo, não variando o RMSE. Por outro lado, o material particulado fino, o tempo no ar e o fator de espalhamento foram os fatores mais influentes no desempenho preditivo. Há estudos experimentais que mostram que o material particulado no ar não possui influência na propagação de ondas eletromagnéticas, pois o comprimento de onda é suficientemente grande para que a interação com a matéria seja importante (LAU; HE (2017) e LO SCIUTO (2019)).

O tempo de transmissão é influenciado por duas variáveis: o fator de espalhamento e o tamanho do quadro a ser transmitido. Ao observar que a previsão usando apenas o fator de espalhamento e o tempo de transmissão apresentou resultados bastante semelhantes, foi feita a avaliação da combinação dessas duas variáveis como entradas do modelo combinadas com o tamanho do quadro. O RMSE encontrado foi de 2,353 dB, indicando que essa combinação não adicionou informações relevantes ao modelo. De forma geral, as diferenças observadas na tabela são pequenas e podem não ser relevantes para o desempenho geral do modelo. Portanto, por ser um dado prontamente disponível no momento da transmissão, optou-se por incorporar apenas o fator de espalhamento como variável em todos os modelos. A combinação dos dados de entrada *distância* e *fator de espalhamento* será denominada *variáveis significativas* para os demais modelos treinados. Para validar os resultados, o modelo treinado com as variáveis escolhidas foi então utilizado para realizar a predição do dataset de teste, obtendo um MAE de 1,89 e um RMSE de 2,34 dB.

5.2.4 Floresta Aleatória para regressão

Duas situações foram testadas utilizando a floresta aleatória no *dataset* Medellín. No primeiro teste, todas as variáveis disponíveis foram consideradas, enquanto no segundo teste foram utilizadas apenas as variáveis identificadas como significativas, que foram a distância e o fator de espalhamento. Nos dois casos, foram testadas as combinações possíveis dos hiperparâmetros profundidade e quantidade de estimadores. O gráfico tanto para os fatores significativos quanto para todas as variáveis disponíveis pode ser visto na Figura 33.

Figura 33 – Avaliação dos hiperparâmetros profundidade e estimadores para a floresta aleatória com todos dados e com os fatores significativos.



Fonte: do autor.

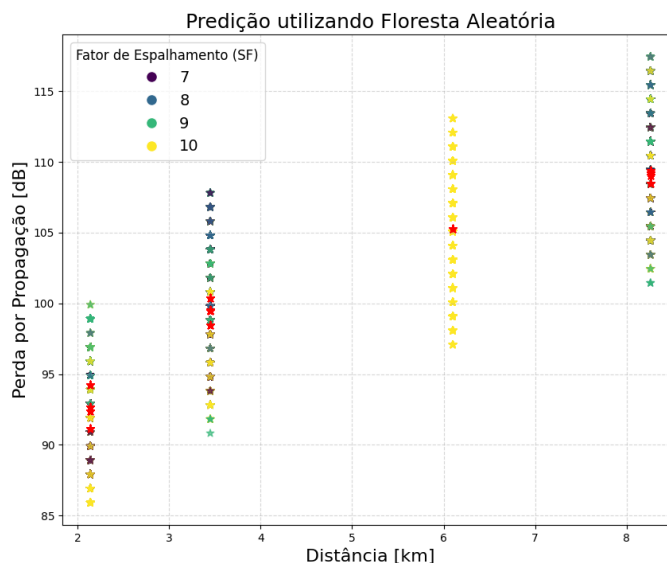
No primeiro teste, que incluiu todas as variáveis, a Floresta Aleatória obteve um MAE de 1,18 dB e um RMSE de 1,51 dB. Esses valores indicam uma capacidade do modelo semelhante ao das árvores aleatórias. No segundo teste, ao considerar apenas as variáveis significativas, o desempenho do modelo foi ligeiramente inferior, conforme esperado. O MAE se manteve semelhante (1,90 dB) e o RMSE subiu para 2,35 dB. Embora os valores de erro tenham aumentado em comparação com o teste que incluiu todas as variáveis, ainda assim os resultados permaneceram bastante satisfatórios para predição das perdas de propagação.

O modelo treinado foi então aplicado ao dataset de teste, obtendo desempenho semelhante, com MAE de 1,89 dB e um RMSE igual a 2,34 dB. A comparação da medição com a predição pode ser vistos na Figura 34.

5.2.5 Support Vector Regression

Ao implementar o SVR no conjunto de dados de Medellín, foram novamente testados quatro kernels diferentes: Linear, Sigmoidal, RBF e Polinomial. Os testes foram realizados considerando todas as variáveis disponíveis e apenas as variáveis significativas identificadas anteriormente. Com o kernel polinomial, tanto para o conjunto completo de

Figura 34 – Predição de perda do caminho utilizando floresta aleatória para regressão no *dataset* Medellín.



Fonte: do autor.

variáveis quanto para as variáveis significativas, os resultados foram razoáveis mas inferiores aos demais modelos. Para o *dataset* completo de variáveis, o MAE foi de 1,52 dB e o RMSE foi de 1,94 dB, enquanto para as variáveis significativas, o MAE foi 4,35 dB, e o RMSE aumentou para 5,99 dBm.

Já para o kernel Sigmoide, os resultados foram insatisfatórios, indicando problemas significativos na predição. Tanto para o conjunto completo de variáveis quanto para as variáveis significativas, o modelo apresentou MAE e RMSE extremamente altos, sugerindo uma falta de adequação do kernel Sigmoide aos dados, assim como no *dataset* de nós móveis.

No caso dos kernels RBF e linear, os resultados foram mais consistentes. Para o kernel RBF, tanto para o conjunto completo de variáveis quanto para as variáveis significativas, os valores de RMSE foram 1,87 e 2,36 dB, respectivamente. Resultado semelhante foi encontrado para o kernel linear, embora com um leve aumento no RMSE para as variáveis significativas (2,92 dB). O resultado da predição, onde pode ser visto o resultado incoerente encontrado para o kernel Sigmoide, pode ser visto na Figura 35.

5.2.6 Regressão Lasso

Na implementação da regressão Lasso foi realizada uma varredura do hiperparâmetro α , que é responsável por controlar a regularização do modelo. Identificou-se que valores menores de α resultaram em um RMSE menor, sugerindo uma melhor adaptação do modelo aos dados, conforme visto na Figura 36.

Figura 35 – Predição de perda do caminho utilizando SVR no *dataset* Medellín.
 Predição utilizando SVR (Kernel Linear)

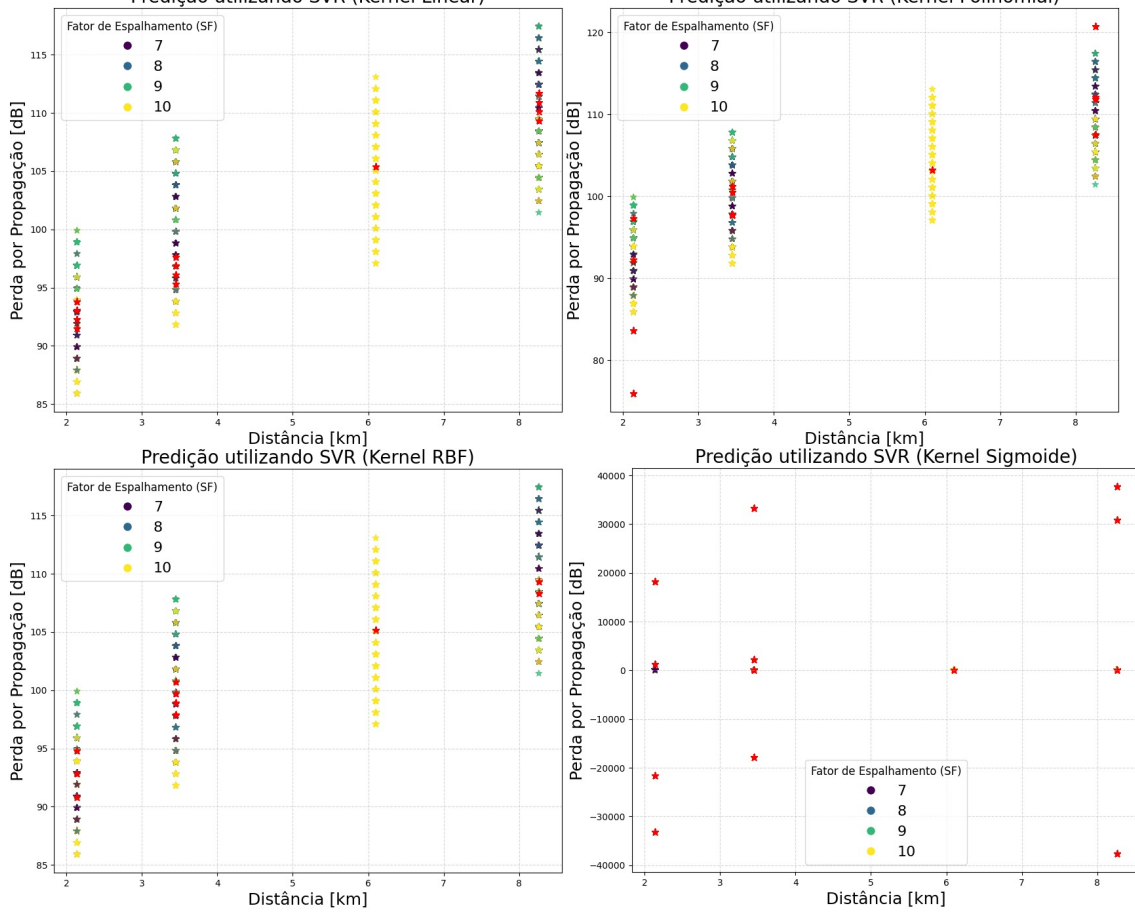
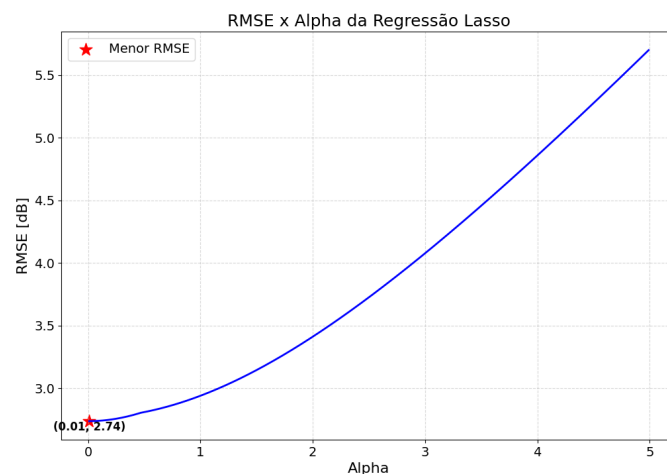


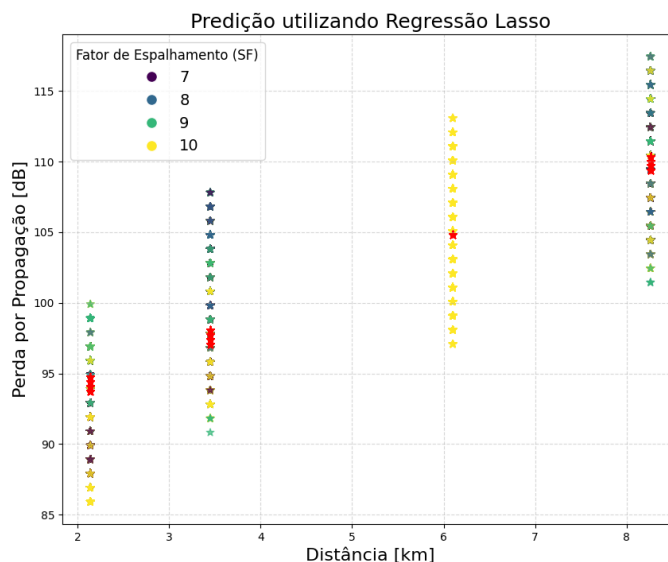
Figura 36 – Avaliação do hiperparâmetro alpha para a Regressão Lasso.



Apesar de também apresentar uma resposta linear, a regressão Lasso obteve resultados semelhantes ao encontrados no SVR com kernel linear. Com todas as variáveis disponí-

veis, a regressão Lasso apresentou um MAE de 1,95 dB e um RMSE de 2,39 dB. Quando consideradas apenas as variáveis significativas, os resultados foram ligeiramente superiores, próximos dos melhores modelos encontrados, com MAE de 2,20 dB e RMSE de 2,72 dB. O gráfico da predição das perdas pode ser visto na Figura 37.

Figura 37 – Predição de perda do caminho utilizando regressão Lasso no *dataset* Medellín.



Fonte: do autor.

5.2.7 XGBoost

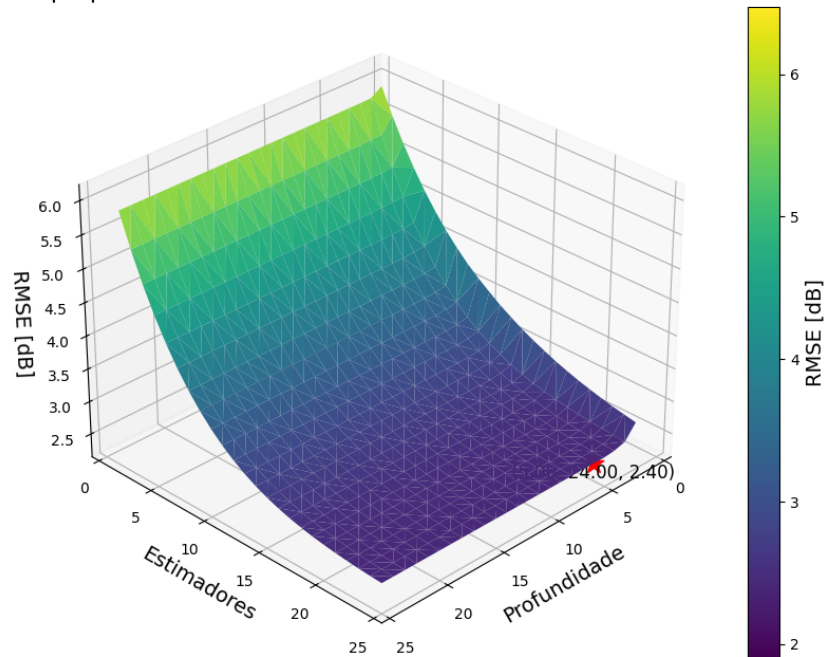
Os resultados obtidos com o algoritmo XGBoost mostram um desempenho semelhante ao da floresta aleatória. Assim como na floresta aleatória, foi realizada uma varredura dos parâmetros de profundidade e estimadores para otimização do modelo, visto na Figura 38. Com todas as variáveis disponíveis, o XGBoost apresentou um MAE de 1,28 dBm e um RMSE de 1,64 dBm. Quando consideradas apenas as variáveis significativas, os erros foram ligeiramente superiores, com um MAE de 1,93 dB e um RMSE de 2,40 dB.

O modelo treinado foi então aplicado ao dataset de validação, obtendo desempenho semelhante, com um MAE de 1,92 db e um RMSE de 2,39 db. A predição pode ser vista na Figura 39.

5.2.8 Rede neural

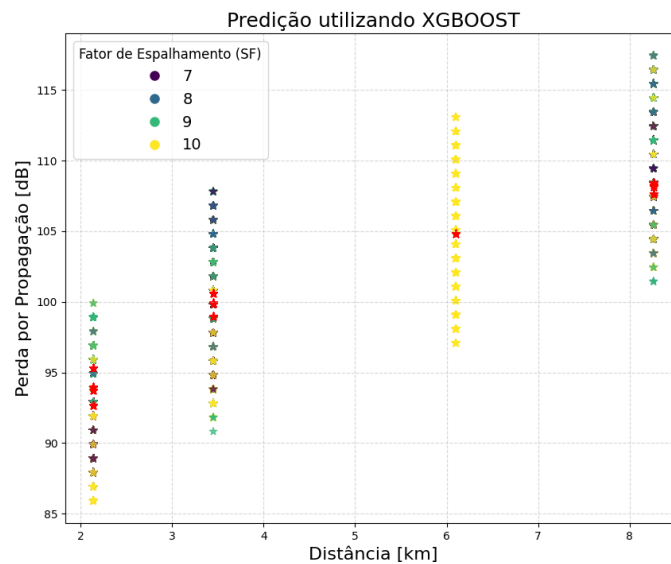
Para a análise do *dataset* Medellín utilizando redes neurais, foi empregado o mesmo framework do *dataset* MCLAB, utilizando o scikit-learn. Uma busca sistemática foi realizada com uma lista de hiperparâmetros pré-definidos, conforme mostrado na Tabela 4. A arquitetura da rede neural permaneceu constante em alguns aspectos: a dimensão de

Figura 38 – Avaliação dos hiperparâmetros profundidade e estimadores para o XGBoost.
RMSE x Hiperparâmetros da Floresta Aleatória com Parâmetros Relevantes



Fonte: do autor.

Figura 39 – Predição de perda do caminho utilizando XGBoost no *dataset* Medellín.

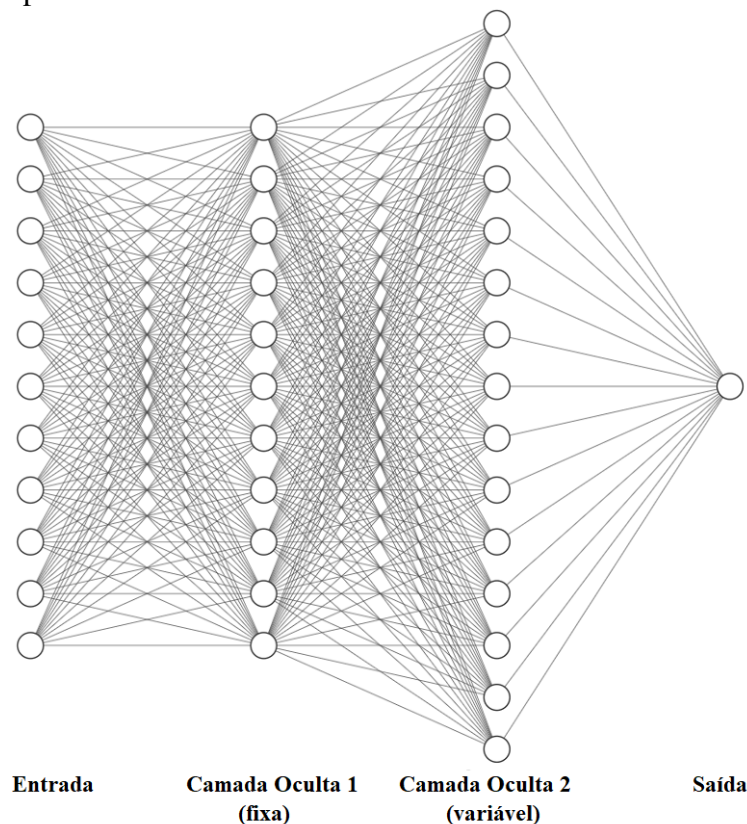


Fonte: do autor.

entrada, a primeira camada oculta com o mesmo número de neurônios que a entrada (2 neurônios para o modelo com as variáveis significativas e 11 neurônios para o modelo com todos os parâmetros), e o limite de duas camadas ocultas. A topologia da rede neural com todos dados de entrada pode ser vista na Figura 40.

Dos modelos gerados, os 20 melhores foram selecionados para uma análise mais de-

Figura 40 – Topologia com camada com nós variáveis com todos dados de entrada explorada como hiperparâmetro no *dataset* Medellín.



Fonte: do autor.

talhada. Na configuração ótima de hiperparâmetros considerando todos os parâmetros de entrada, o melhor modelo utilizou uma camada de 5.000 neurônios, aplicando a função de ativação ReLu e o otimizador Nadam. Este modelo obteve um MAE de 1,56 dB e um RMSE de 1.98 dB. Também foi feita a busca pelo modelo com a menor quantidade de neurônios entre os 20 melhores, uma vez que este pode ter um desempenho computacional melhor. O resultado foi um MAE de 1,66 dB e um RMSE de 1,95 dB, sendo que o modelo utilizou uma camada de 40 neurônios, aplicando a função de ativação *tanh* e o otimizador SGD.

Utilizando apenas os parâmetros de entrada significativos, a condição ótima foi encontrada em um modelo que apresentou uma camada de 10 neurônios, aplicando a função de ativação Sigmoid e o otimizador Nadam, com um MAE de 2,10 dB e um RMSE de 2,66 dB na predição do dataset de teste. O modelo com a menor quantidade de neurônios entre os 20 melhores também foi o melhor modelo, pois os mais complexos levaram rapidamente ao *overfitting*, resultando em desempenho inferior.

5.3 Implementação dos algoritmos no simulador LoraEnergySim

Para permitir a simulação com novos modelos de propagação no framework, algumas melhorias foram implementadas. Primeiramente, foi necessário identificar quais modelos de propagação seriam mais adequados para representar as condições da rede LoRa. Pela simplicidade, todos os modelos clássicos foram implementados, assim como os modelos de inteligência artificial, com a restrição que foram utilizados apenas os modelos treinados com dados do conjunto de dados MCLAB, que oferecem uma representação mais precisa das condições de propagação de sinal em ambientes urbanos, uma vez que os dados coletados no *dataset* Medellín se aproximam de um modelo de propagação livre, com visada entre transmissor e receptor.

No arquivo `PropagationModel`, foram criadas novas classes para cada modelo de propagação, como `FreeSpace`, `Egli`, `OkumuraHata`, `COST231Hata`, `DecisionTree`, `RandomForest`, `SVR`, `Lasso`, `XGBOOST` e `NeuralNetwork`. Cada uma dessas classes possui um método chamado `tp_to_rss`, que recebe os parâmetros de distância e elevação do nó e calcula a intensidade do sinal recebido.

Os modelos de aprendizagem de máquina necessários para esses cálculos foram armazenados em arquivos compactados (.zip) em uma pasta específica. Quando uma instância da classe de propagação é criada, o arquivo correspondente é descompactado e o modelo é carregado utilizando a biblioteca `joblib`, facilitando sua utilização durante a simulação.

Também foi realizada uma implementação para considerar a elevação do nó transmissor, o que não estava previamente contemplado no simulador. Foram feitas alterações na ferramenta de geração de nós da rede a ser simulada, bem como nos módulos onde essa informação é utilizada.

Durante a execução de uma simulação, o método `tp_to_rss` é chamado pelo método `packet_in_air` do módulo de interface e é utilizado para calcular a intensidade do sinal de cada transmissão. A chamada do método também foi alterada para incluir a elevação do nó, que foi implementada como parte desse processo de aprimoramento do simulador.

6 ANÁLISE DE RESULTADOS

6.1 Comparação de resultados dos algoritmos implementados

A Tabela 6 exibe uma comparação entre os diversos algoritmos empregados na modelagem da propagação de sinais LoRa nos conjuntos de dados MCLAB e Medellín. Para o conjunto de dados de Medellín, a comparação é limitada aos algoritmos que levam em conta apenas as variáveis significativas. Esta restrição decorre da dificuldade prática de incorporar parâmetros ambientais ao modelo, pois isso acarretaria em uma quantidade muito grande de dados a serem simulados, dificultando a generalização em um cenário de rede LoRa urbana. Além disso, esses dados nem sempre estão presentes em dispositivo LoRa.

As métricas de MAE e RMSE foram utilizadas para avaliar o desempenho de cada algoritmo. Especificamente o RMSE foi utilizado como métrica principal, uma vez que o MAE pode ser baixo mesmo para modelos que não consigam prever adequadamente a perda por propagação. Estes resultados guiaram a seleção dos modelos que foram implementados no simulador de rede LoRa.

O modelo Free Space, que considera apenas a distância entre transmissor e receptor, demonstra variações significativas nos resultados. No conjunto de dados Medellín, o RMSE é baixo, indicando que a situação de coleta dos dados se aproxima de um campo aberto. No entanto, para o conjunto MCLAB, o RMSE é mais alto (40,90 dB), sugerindo que este modelo não se adapta bem a esses dados.

Os modelos de Egli, Okumura-Hata e COST231-Hata são mais complexos pois já levam em conta a altura das antenas e a topografia, mas também apresentam desafios de adaptação aos diferentes conjuntos de dados. No Medellín, nenhum deles se adaptou bem, enquanto que no MCLAB o melhor modelo foi o Okumura-Hata, ainda que com erros superiores até mesmo ao Log-distance com parâmetros de Oulu.

Entre os modelos que utilizam IA, em ambos os casos os melhores resultados foram encontrados naqueles modelos baseados em árvore, como a própria árvore de decisão, a floresta aleatória e o XGBOOST. Apesar dos esforços de treinamento de redes neurais com diferentes combinações de hiperparâmetros, em nenhum dos dois conjuntos de dados

Tabela 6 – Comparação entre Algoritmos com os *datasets* MCLAB e Medellín.

Algoritmo	MCLAB		Medellín	
	MAE [dB]	RMSE [dB]	MAE [dB]	RMSE [dB]
Free Space	39,72	40,93	3,23	3,99
Egli	12,96	15,37	42,97	43,69
Okumura-Hata	8,63	10,77	53,11	53,53
COST231-HATA	10,13	12,67	56,68	57,10
Log-Distance (Oulu)	8,13	10,04	42,22	42,30
Log-Distance (fit)	7,83	9,82	1,99	2,46
Árvore de Decisão	5,57	7,14	1,89	2,34
Floresta Aleatória	5,28	6,82	1,89	2,34
SVR (RBF)	6,39	8,21	1,87	2,36
SVR (Linear)	7,45	9,56	2,30	2,92
SVR (Polinomial)	7,36	9,51	4,35	5,99
SVR (Sigmoides)	1.177	1.832	12.413	19.475
Regressão Lasso	7,52	9,47	2,20	2,72
XGBoost	5,32	6,85	1,92	2,39
Rede Neural (melhor)	6,63	8,25	2,10	2,66
Rede Neural (menor)	6,55	8,20	2,10	2,66

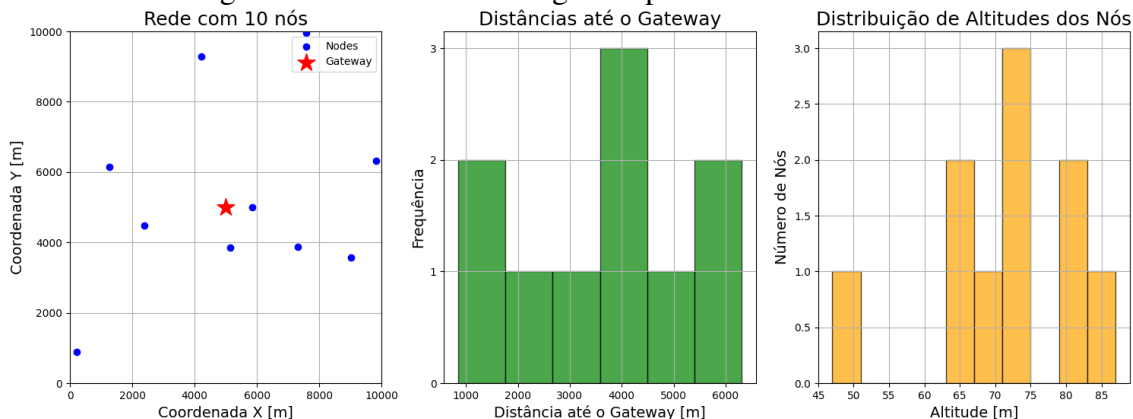
esta foi a melhor escolha para predição de perdas por propagação.

6.2 Redes geradas para testes do simulador

Para garantir a reprodutibilidade dos experimentos, foi necessário gerar uma lista de nós de forma aleatória dentro do espaço de 10.000 por 10.000 metros. O próprio framework utilizado disponibiliza um script para essa geração, o qual foi adaptado para incluir dados de altitude. Todas as redes geradas foram padronizadas com essa dimensão para facilitar a comparação entre os experimentos.

Os limites de altitude foram estabelecidos entre 45 e 90 metros, valores semelhantes às altitudes da cidade onde foram coletados os dados do *dataset* utilizado para treinar os modelos de inteligência artificial. As redes foram geradas com 10, 100 e 200 nós distribuídos nesse espaço, com o gateway sempre posicionado no centro da célula, nas coordenadas 5000, 5000 metros. Para a rede menor, com apenas 10 nós (Figura 41, as distribuições de altitude e distância, que são aleatórias, acabam por não apresentar nenhuma tendência. Por outro lado, o histograma das distâncias até o gateway para redes com 100 e 200 nós (figuras 42 e 43, respectivamente) apresenta uma distribuição que se aproxima de uma normal, conforme esperado para redes com densidades crescentes. Já o

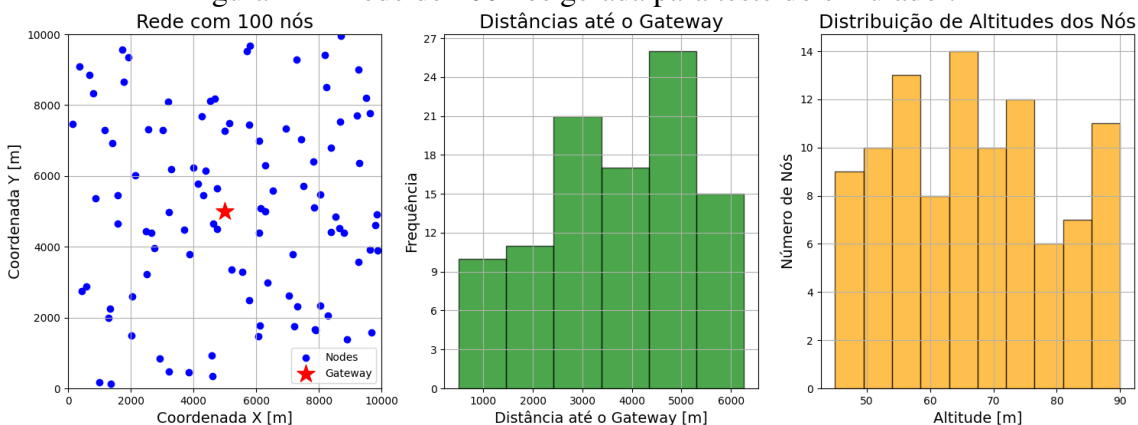
Figura 41 – Rede de 10 nós gerada para teste do simulador.



Fonte: do autor.

histograma de altitudes exibe uma distribuição uniforme entre os limites estabelecidos.

Figura 42 – Rede de 100 nós gerada para teste do simulador.

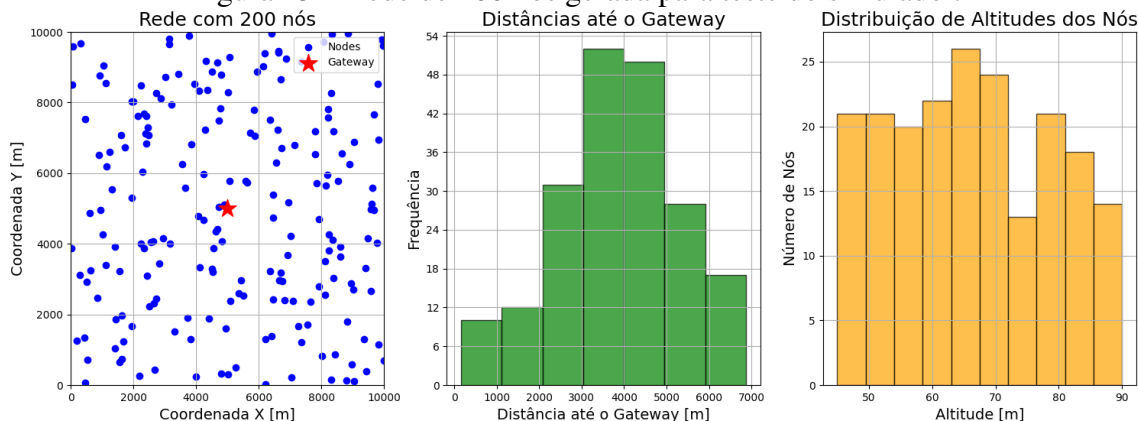


Fonte: do autor.

6.3 Avaliação do tempo de processamento

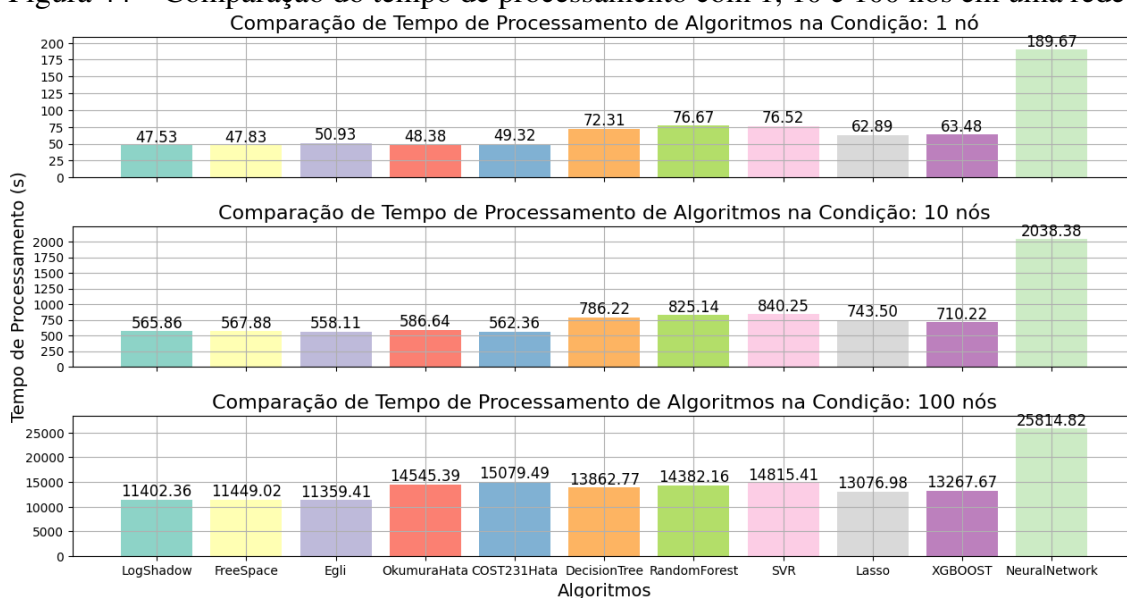
Para avaliar o tempo de processamento utilizando diferentes algoritmos, a simulação foi realizada em condições específicas e constantes. O fator de espalhamento foi mantido fixo em 12, com um pacote transmitido por hora para cada nó na rede, contendo 12 bytes de dados. A simulação abrangeu um período de um mês de execução da rede, considerando uma célula com dimensões de 10.000 por 10.000 metros. Durante a simulação, o algoritmo de ajuste automático de fator de espalhamento (*Adaptive Data Rate, ADR*) foi desativado, e a confirmação de recebimento de pacotes pelo gateway também foi desativada. A Figura 44 ilustra a comparação dos tempos de processamento dos algoritmos de previsão de perdas de propagação implementados, com a variação do número de nós na rede. Estes algoritmos foram testados em três cenários distintos: uma rede com 1 nó, outra com 10 nós e uma terceira com 100 nós.

Figura 43 – Rede de 200 nós gerada para teste do simulador.



Fonte: do autor.

Figura 44 – Comparação do tempo de processamento com 1, 10 e 100 nós em uma rede.



Fonte: do autor.

Foi observado que os algoritmos tradicionais, que incluem o *log-distance* (denominado *LogShadow* pelo simulador, *Free Space*, Egli, OkumuraHata e COST231Hata, exibem tempos de processamento inferiores quando comparados aos algoritmos que empregam inteligência artificial. Apesar de serem amplamente utilizados em propagação de sinal, estes algoritmos clássicos tendem a apresentar erros mais significativos, conforme discutido anteriormente.

Em contrapartida, os algoritmos que fazem uso de inteligência artificial, como a Árvore de Decisão, Floresta Aleatória, XGBOOST, SVR, Lasso e Rede Neural, demonstram tempos de processamento superiores. Entre estes, a Árvore de Decisão e os algoritmos derivados dela, como a Floresta Aleatória e o XGBOOST, se destacam por tempos de simulação menores em comparação aos algoritmos tradicionais.

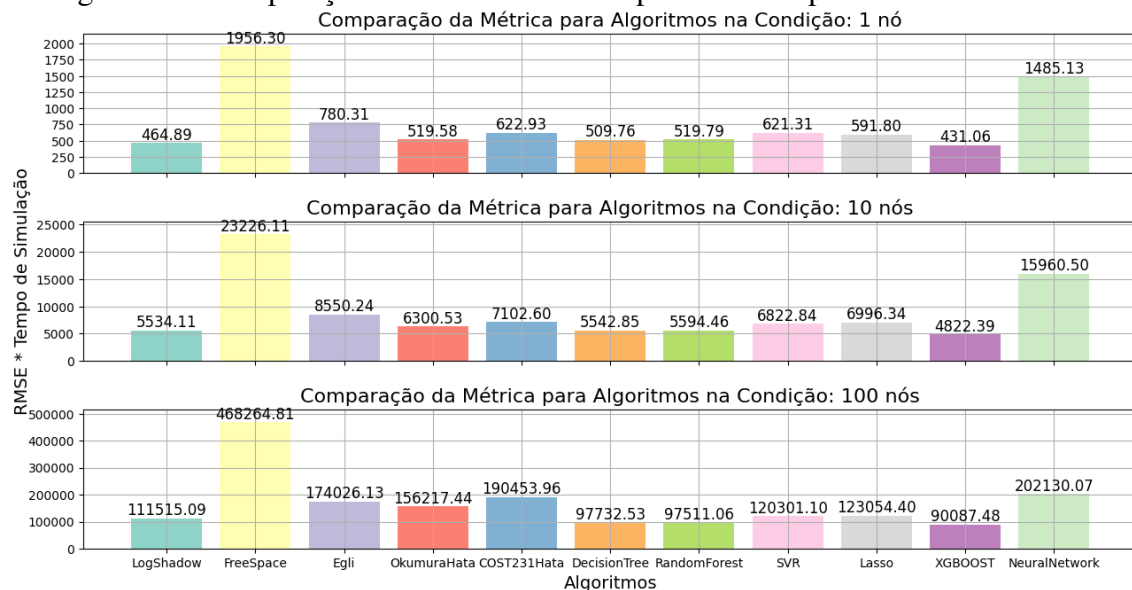
A regressão Lasso, apesar de apresentar um tempo de processamento relativamente

menor entre os algoritmos que utilizam inteligência artificial, não oferece grandes benefícios em termos de erro do modelo. Isso sugere que a simplicidade do Lasso pode não ser adequada para capturar a complexidade dos dados de propagação de sinal. No caso da Rede Neural, além de apresentar um tempo de processamento consideravelmente maior em comparação com os algoritmos derivados da árvore, seu erro também é maior, o que não justifica sua utilização na simulação.

Diferente do esperado, com 100 nós na rede, os tempos de processamento do OkumuraHata e do COST232Hata foram superiores a alguns algoritmos que utilizam inteligência artificial, como a Árvore de Decisão. Isso indica que existe uma variação inesperada no desempenho desses algoritmos em diferentes configurações de rede, e mais investigações são necessárias para entender completamente esse fenômeno.

Para avaliar os modelos que faziam sentido serem explorados nos ensaios subsequentes, foi criada uma métrica que leva em consideração o erro do modelo e o tempo de processamento, conforme explicado na Seção 4.4. O resultado para cada rede e cada algoritmo pode ser visto na Figura 45. Com base nestes resultados, os testes subsequentes com o simulador levarão em consideração apenas os algoritmos Árvore de Decisão, Floresta Aleatória e XGBOOST, pois são os que apresentam o melhor equilíbrio entre erro e tempo de simulação.

Figura 45 – Comparação da métrica de desempenho criada para todos os modelos.



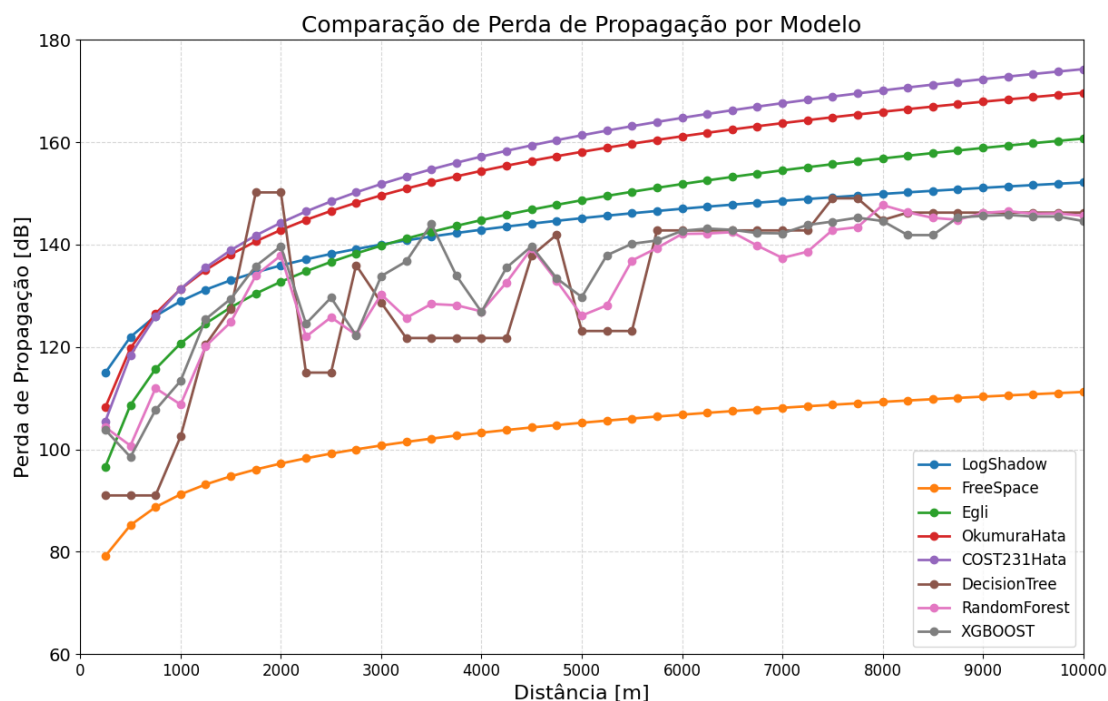
Fonte: do autor.

6.4 Comparação de resultados simulados

6.4.1 Perda de propagação com diferentes modelos

A comparação dos modelos de propagação foi realizada gerando uma série de nós igualmente espaçados entre 250 e 10.000 metros do gateway. A distância máxima foi definida levando em consideração as distâncias máximas encontradas no *dataset* que foi utilizado para treinamento dos modelos com IA. Em seguida, foi realizado o cálculo da perda por propagação em todos os modelos implementados, considerando a potência de transmissão de 0 dBm e a elevação do nó fixa em 60 metros. O resultado de todos os modelos pode ser visto na Figura 46.

Figura 46 – Comparação da perda para diferentes modelos de propagação até 10 km.



Fonte: do autor.

Os resultados obtidos para os modelos clássicos são idênticos aos encontrados no Capítulo 5. Todos apresentam um padrão de curva similar, embora existam variações de até 30 dB entre os modelos, especialmente para distâncias mais extensas. Quanto aos modelos baseados em IA, observam-se variações nos valores, mas o comportamento geral é semelhante entre eles.

Existe uma diferença grande em relação aos modelos clássicos para distâncias mais curtas. No intervalo de 250 a 1000 metros, todos os modelos baseados em IA indicam uma perda de propagação inferior à dos modelos clássicos. Esta situação se inverte entre 1500 e 2000 metros, onde as perdas indicadas pelos modelos treinados aumentam rapidamente, com o modelo de árvore de decisão apresentando a maior perda entre todos os modelos para este intervalo. Posteriormente, as perdas diminuem drasticamente e vol-

tam a seguir um padrão mais próximo do esperado, com perdas sempre inferiores às dos demais modelos.

Assim como nas medições de Young, as perdas de propagação para distâncias maiores foram menores do que as estimadas pelos modelos clássicos. Não é possível afirmar com certeza quais fatores levaram a menores perdas a partir de 6.000 metros, mas a hipótese de Young para sua descoberta foi que distâncias maiores levam o dispositivo para áreas mais suburbanas da cidade, onde a altura dos prédios é menor e a perda começa a ter um impacto reduzido. Uma situação similar pode ter ocorrido com o conjunto de dados utilizado para o treinamento.

6.4.2 Avaliação do impacto do modelo de propagação em função do fator de espalhamento

Foi executada uma série de 48 ensaios utilizando o simulador explorando diferentes cenários. Estes cenários contemplaram seis fatores de espalhamento e oito modelos de propagação, incluindo tanto os modelos clássicos quanto os três modelos baseados em inteligência artificial escolhidos. Cada ensaio foi configurado para simular um mês de operação de uma rede com 100 nós, enviando um pacote por hora, com transmissões de 12 bytes e características específicas, como a ausência de ADR, retransmissão e confirmação de recebimento.

A avaliação desses ensaios focou na quantidade de pacotes recebidos pelo gateway, pacotes que colidiram durante a transmissão e pacotes que não foram recebidos devido à potência de sinal insuficiente. Para determinar a condição de não recebimento, o simulador estabelece limites de potência mínima para cada fator de espalhamento, variando de -121 dBm para SF6 até -139.5 dBm para SF12.

A base para análise foi o modelo *log-distance* com parâmetros do modelo de Oulu, que serviu como referência para comparação com os outros modelos testados. O primeiro modelo avaliado foi o Free Space. Este modelo, que considera uma propagação ideal sem perdas, revelou que, nas distâncias simuladas, nenhum pacote seria perdido devido à atenuação do sinal, independentemente do fator de espalhamento. Essa observação é válida para redes sem obstruções significativas entre os nós e o gateway, refletindo uma condição idealizada de comunicação, diferente da condição encontrada tanto nos experimentos de Oulu, quanto no *dataset* MCLAB utilizado para treinamento dos modelos de IA.

O modelo de Egli revelou resultados semelhantes aos do *log-distance* para o fator de espalhamento 7. No entanto, houve divergência significativa para os fatores de espalhamento 9 e 10. Para a frequência de 868 MHz considerada na simulação, o modelo de Egli é mais sensível à distância do que o *log-distance*, resultando em uma atenuação menor em distâncias menores (abaixo de 1 km) e uma atenuação maior em distâncias maiores. Dado que a maioria dos nós se encontra em distâncias entre 3 e 5 km, é esperado que os resultados deste modelo sejam mais conservadores do que os do modelo *log-distance*.

Embora os resultados sejam mais realistas do que os do modelo de propagação livre, ainda há divergências a serem consideradas. Também foi observada uma redução na incidência de colisões de pacotes em comparação com outros modelos. Um dos tipos de colisão detectados é a colisão por potência, que depende fortemente do modelo de propagação. Essa colisão ocorre quando dois pacotes na mesma frequência e com sobreposição no tempo têm uma diferença de potência menor que 6 dBm, mesmo que estejam em diferentes fatores de espalhamento. Sendo um modelo mais sensível a distância, mesmo em caso de conflito no tempo os pacotes podem chegar com diferenças maiores em potência, sendo recebidos corretamente pelo gateway.

Os modelos Okumura-Hata e COST231-Hata apresentaram perdas muito superiores em relação a todos os demais modelos e parecem não refletir corretamente a realidade. Embora as perdas de pacotes diminuam conforme aumenta o fator de espalhamento, o que é coerente com a distância, os resultados ainda se mostraram preocupantes. Mesmo para fatores de espalhamento 11 e 12, as perdas de pacotes encontradas foram próximas a 70% e 60%, respectivamente. Esses valores estão consideravelmente altos em relação ao que é observado na literatura, especialmente considerando que, para uma célula de 10.000 x 10.000 metros, onde a distância máxima fica próxima de 7 km, espera-se uma taxa de perda muito inferior.

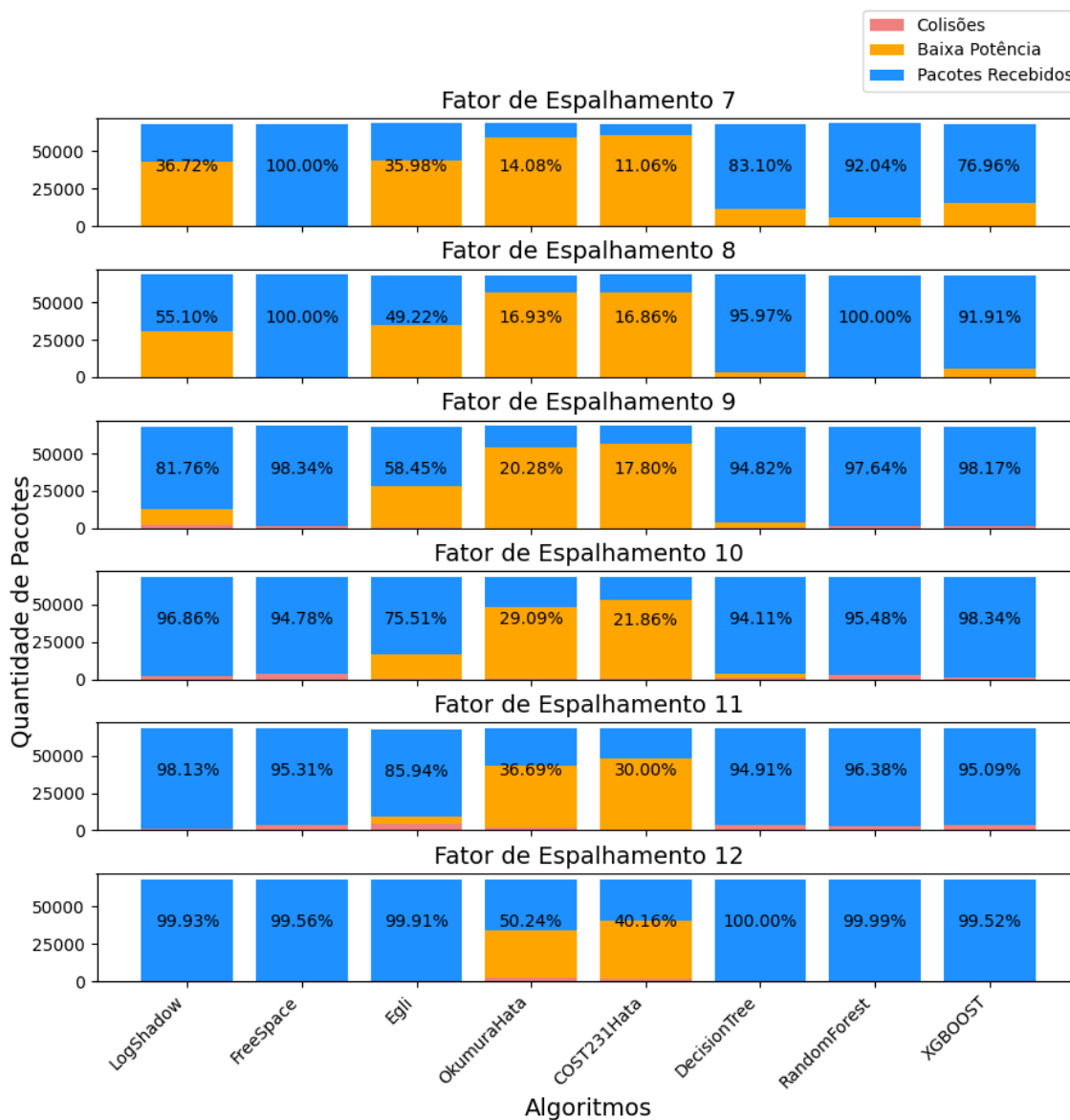
Os três modelos de IA avaliados - árvore de decisão, floresta aleatória e XGBoost - indicaram perdas significativamente menores ao modelo *log-distance* para os fatores de espalhamento 7, 8 e 9. Esses modelos, que possuem erro menor conforme observado na implementação e treinamento, mostram uma convergência em seus resultados, sugerindo que o modelo *log-distance* pode estar subestimando as perdas por propagação em relação ao que é encontrado experimentalmente. Entretanto, foi observado um resultado peculiar do modelo XGBoost, que indicou uma perda de 100% dos pacotes no SF mais baixo. Isso parece pouco provável, considerando que há pelo menos seis nós com distâncias inferiores a 2 km, sendo o mais próximo do gateway a apenas 719 metros.

Outra observação importante é a divergência significativa observada na árvore de decisão em relação aos demais modelos para o SF 7. Enquanto era esperada uma variação maior da taxa de pacotes entregues conforme aumenta-se o fator de espalhamento, houve uma variação pequena, de pouco mais de 8% dos pacotes recebidos com SF8. A floresta aleatória, que obteve os menores erros nos treinamentos, apresentou a resposta mais coerente, tendo uma taxa de pacotes não entregues maior para o SF 7. Por outro lado, a árvore de decisão, que tem o terceiro menor erro, parece subdimensionar as perdas para SF 7 e 8. Exceto pelo SF7, onde as perdas são consideravelmente maiores, o modelo XGBoost parece estar alinhado com os demais modelos em termos de resultados.

Em todos os modelos, as colisões com SF12 foram menores do que as encontradas com SF11 e 10. Valores maiores de SF levam a um aumento do tempo no ar, uma vez que o tamanho do *payload* é constante, o que aumenta a probabilidade de colisão no tempo.

É possível que essa discrepância seja uma limitação na implementação do simulador, que até o momento não foi identificada por outros pesquisadores que utilizaram ele. A Figura 47 mostra o resultado de pacotes recebidos, colididos e não recebidos por potência baixa ao chegar no gateway. O valor em % no gráfico é a quantidade de pacotes recebidos corretamente em relação a quantidade total de pacotes enviados.

Figura 47 – Comparação da quantidade de pacotes recebidos para diferentes modelos de propagação em diferentes fatores de espalhamento.



Fonte: do autor.

6.4.3 Avaliação do impacto do modelo de propagação em função do tamanho da célula

Neste ensaio, foi explorado o desempenho dos oito algoritmos de predição de perdas considerando diferentes cenários de tamanho de células. Os experimentos foram conduzi-

dos em células com dimensões de 100x100 metros, 1.000x1.000 metros e 10.000x10.000 metros, utilizando 100 dispositivos transmitindo um pacote de 12 bytes por hora. Foram criadas novas redes para estes ensaios, exceto para o maior, de 10.000 x 10.000, onde foi utilizada a rede da Figura 42.

O ADR foi ativado para ajustar dinamicamente os fatores de espalhamento de acordo com a perda por propagação entre dispositivos e o gateway. Em um cenário real, o ADR é utilizado com o objetivo de reduzir o consumo de energia e minimizar colisões de pacotes durante a transmissão, então, ambas informações foram coletadas na simulação. Durante os experimentos, foi configurada a retransmissão de pacotes que não receberam confirmação de recebimento do gateway, seja por colisão ou por potência muito baixa. Foram permitidas até 8 tentativas de retransmissão antes que os dispositivos entrassem em modo de baixo consumo de energia novamente.

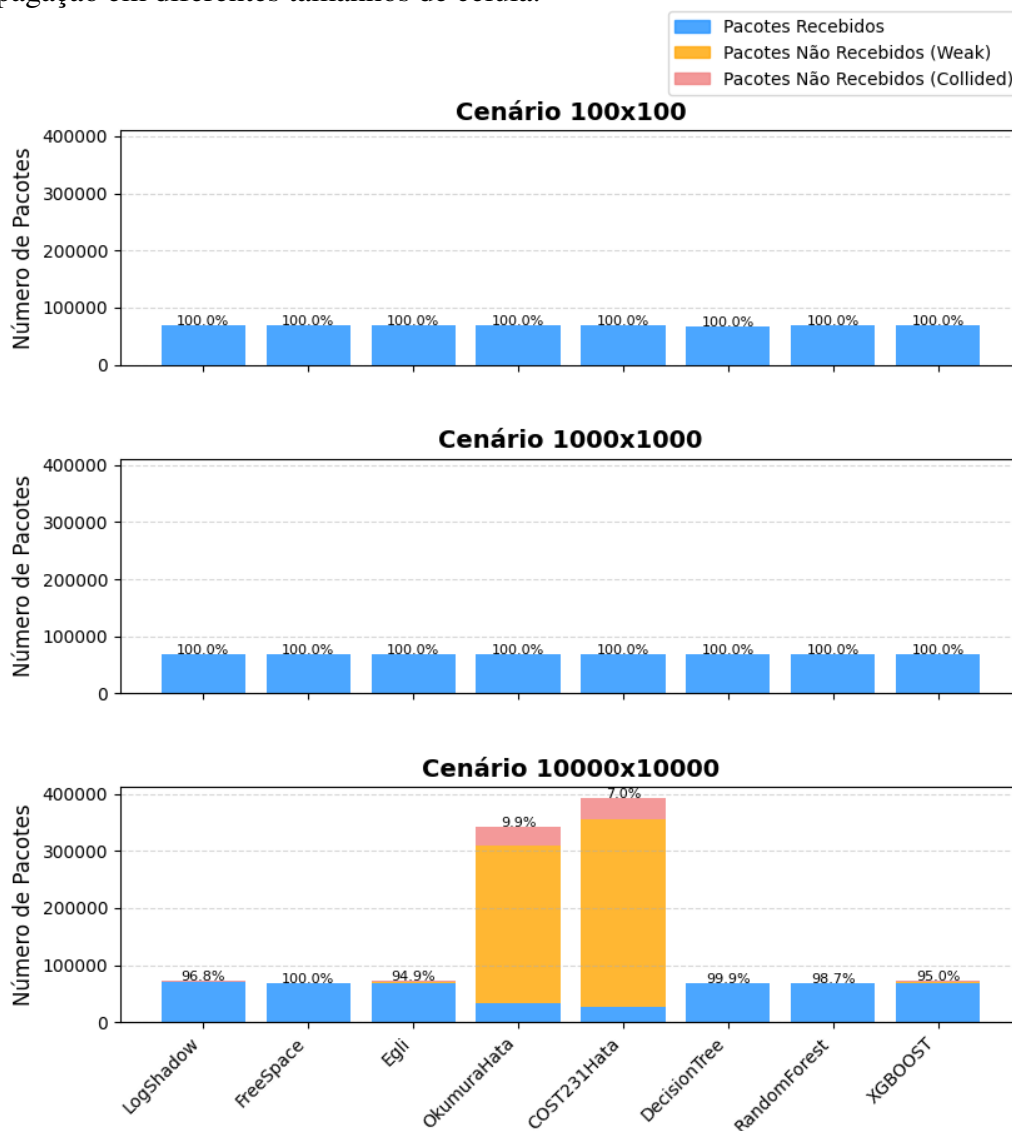
Nos cenários menores de 100x100 e 1.000x1.000 metros, todos os modelos utilizaram os menores fatores de espalhamento, uma vez que não houve perda de pacotes, conforme visto na Figura 48. O consumo de energia entre os modelos foi semelhante nessas condições. Em células menores, não há vantagem em utilizar os modelos de propagação com IA, uma vez que o tempo de processamento é maior, sem que haja diferença nos resultados. No entanto, no cenário maior de 10.000x10.000 metros, foram observadas perdas de pacotes em todos os modelos, com exceção do modelo de espaço livre, que tem as menores atenuações. Os modelos Okumura-Hata e COST231-Hata indicaram perdas por propagação mais elevadas do que os demais, o que levou a também apresentarem uma quantidade significativa de retransmissões de pacotes, resultando em um aumento no consumo de energia e na quantidade de colisões.

O ADR teve um papel importante para a avaliação da correta quantidade de pacotes recebidos, porém mascara as diferenças entre os modelos, ajustando dinamicamente o fator de espalhamento. As diferenças reais entre os modelos puderam ser melhor avaliadas ao observar o consumo total de energia.

Na avaliação do consumo de energia, o modelo de espaço livre serve como referência, consumindo 1.976 J, sempre com transmissões utilizando SF 7, visto na Figura 49. Os modelos com IA apresentaram consumos superiores, porém ainda inferiores aos modelos clássicos, indicando que os modelos tradicionais sobredimensionam as perdas, levando os dispositivos a utilizarem fatores de espalhamento maiores, aumentando o consumo.

Desprezando os modelos com quantidade elevada de retransmissão, o modelo de Egli registrou o maior consumo de energia, totalizando 6.519 J, seguido pelo *log-distance* com 3.947 J. Em contraste, os modelos de Árvore de Decisão e Floresta Aleatória apresentaram consumos mais baixos, com 2.464 J e 2.302 J, respectivamente. O modelo XGBOOST, conforme esperado pelas maiores perdas vistas com SF7, ficou intermediário em termos de consumo, totalizando 3.153 J.

Figura 48 – Comparação da quantidade de pacotes recebidos para diferentes modelos de propagação em diferentes tamanhos de célula.



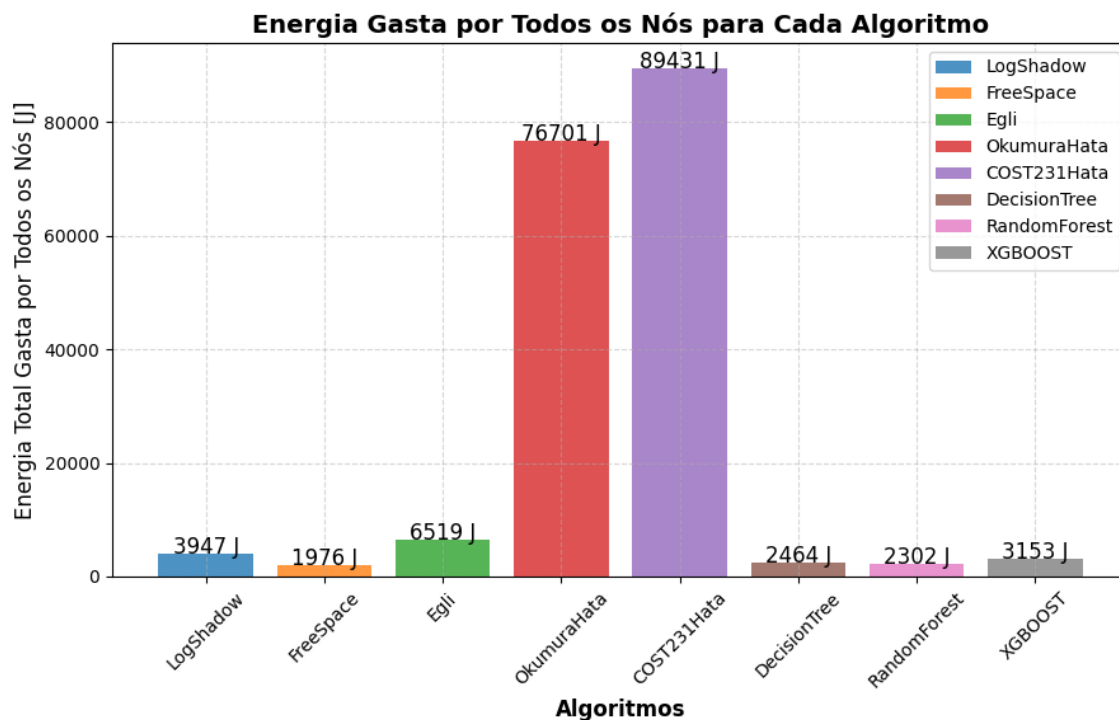
Fonte: do autor.

6.4.4 Avaliação do impacto do modelo de propagação em função da quantidade de nós

No último ensaio destinado a avaliar o impacto dos modelos de propagação no dimensionamento de redes LoRa, foram testados três cenários em uma célula de 10.000 por 10.000 metros, com 10, 100 e 200 nós em cada um. A quantidade de dados transmitida por nó foi mantida constante, sendo um pacote de 12 bytes por hora, resultando em diferentes quantidades de transmissões para cada ensaio. O ADR foi mantido habilitado, assim como as retransmissões de pacotes não recebidos pelo gateway. Os resultados da simulação podem ser vistos na Figura 50.

Os resultados obtidos indicaram um comportamento semelhante ao observado no ensaio de variação do tamanho da célula, com uma perda significativa de pacotes nos mode-

Figura 49 – Comparação da energia consumida para diferentes modelos de propagação em na célula 10.000 x 10.000 m.

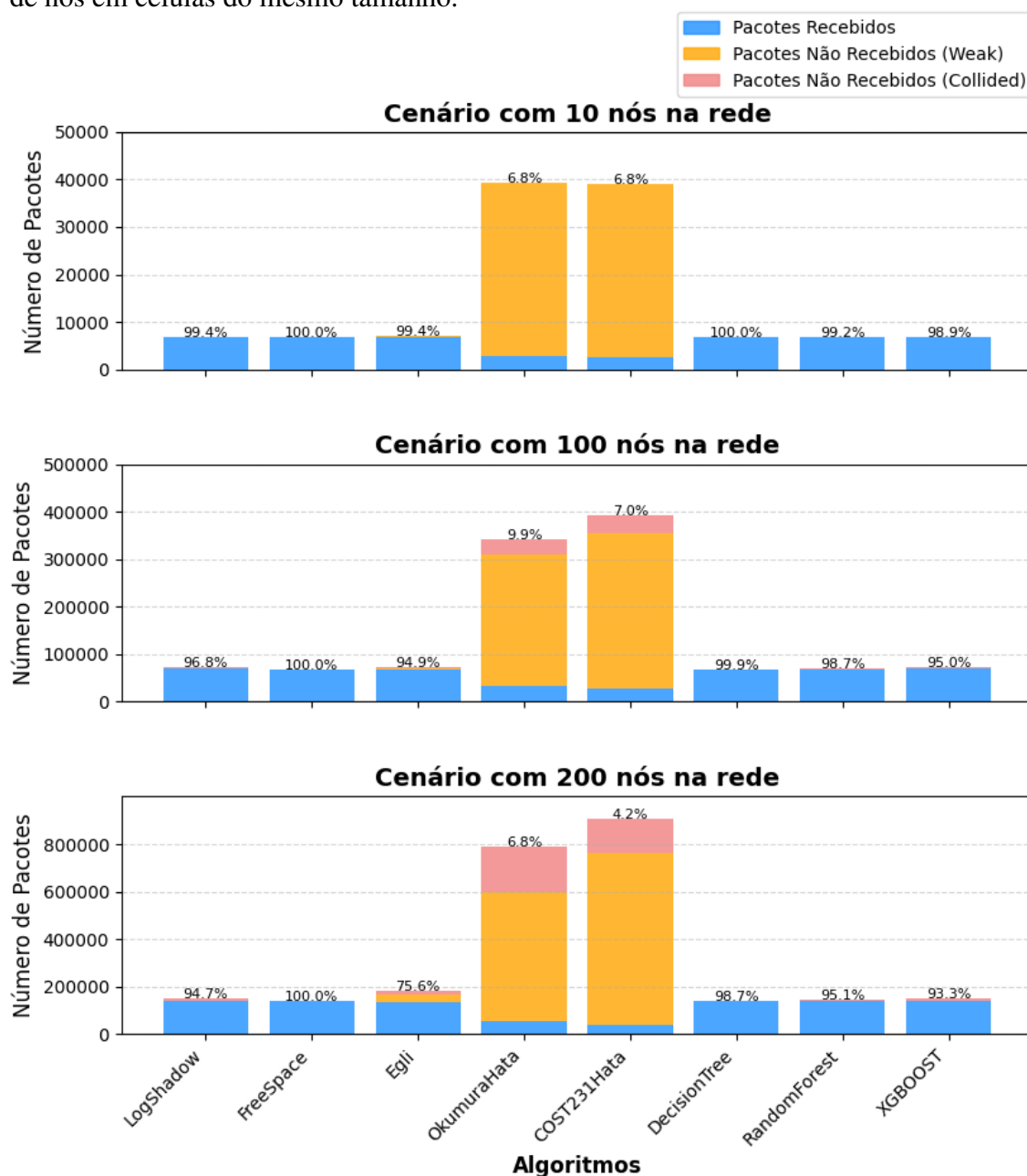


Fonte: do autor.

los de propagação Okumura-Hata e COST231-Hata, particularmente em distâncias maiores. Esses modelos tendem a superestimar as perdas de sinal à medida que a distância aumenta, levando a níveis de sinal que excedem a sensibilidade máxima de recepção do gateway, mesmo quando o fator de espalhamento é aumentado. O modelo de Egli também apresentou perdas superiores em comparação com outros modelos, o que resultou em um número maior de retransmissões. Este problema foi agravado à medida que a densidade da rede aumentou, especialmente no cenário com 200 nós, levando a mais conflitos de pacotes e, conseqüentemente, maiores perdas.

Em relação aos modelos baseados em inteligência artificial, os resultados foram consistentes com as expectativas, sem variações significativas em termos de desempenho. A quantidade de pacotes perdidos manteve-se dentro do esperado, com as perdas por colisão de pacotes aumentando conforme o número de nós na rede crescia.

Figura 50 – Comparação da quantidade de pacotes recebidos para diferentes quantidade de nós em células do mesmo tamanho.



Fonte: do autor.

7 CONCLUSÃO

Este trabalho teve como objetivo investigar e avaliar modelos de predição da perda de percurso em redes LoRa, utilizando tanto abordagens clássicas quanto modelos de IA, aplicadas a dois conjuntos de dados distintos: MCLAB, contendo dados de nós móveis, e Medellín, com dados de nós fixos. O foco foi testar a eficácia desses modelos para prever a propagação do sinal em diferentes cenários e, ao mesmo tempo, avaliar o impacto dessa predição no dimensionamento de redes LoRa. O uso de um simulador, o LoRaEnergy-Sim, possibilitou simular redes reais e observar como diferentes modelos influenciam o desempenho da rede em termos de perda de pacotes e consumo de energia.

Os resultados dos modelos treinados com dados reais demonstraram que os modelos baseados em árvore de decisão, como a Floresta Aleatória e o XGBOOST, tiveram o melhor desempenho em ambos os conjuntos de dados, superando tanto os modelos clássicos quanto as redes neurais. Especificamente no conjunto de dados MCLAB, a Floresta Aleatória obteve um RMSE de 6,78 dB, significativamente melhor que o modelo clássico *log-distance*, que apresentou 9,78 dB. No conjunto Medellín, a Floresta Aleatória também se destacou, com um RMSE de 1,90 dB em comparação a 2,00 dB do *log-distance*. Esses resultados indicam que, em cenários reais, os modelos de IA podem oferecer uma predição melhor da perda de percurso, contribuindo para um dimensionamento mais preciso das redes LoRa. Além disso, observou-se que os modelos clássicos tendem a superestimar as perdas, o que pode resultar em redes superdimensionadas, especialmente em redes com muitos nós ou células urbanas maiores.

Todos os modelos foram implementados no simulador, mas os modelos de inteligência artificial apresentaram tempos de processamento superiores. Para lidar com isso, uma métrica relacionando RMSE e tempo de processamento foi criada para avaliar quais modelos eram viáveis para uso. Além dos modelos clássicos, apenas os modelos baseados em árvore foram considerados viáveis, devido ao seu menor erro e tempo de processamento. Observou-se que os modelos clássicos tendem a superestimar as perdas de propagação, o que indica maior perda de pacotes em fatores de espalhamento mais baixos e em redes maiores. Essa superestimação pode levar a um dimensionamento incorreto das redes, especialmente em casos onde a retransmissão de pacotes é necessária.

No que se refere às limitações de pesquisa, alguns aspectos devem ser considerados. Afim de manter a compatibilidade multiplataforma, estratégias como o uso de GPU para processamento não foram implementadas, o que poderia alterar os resultados de tempo de processamento dos modelos de IA. Além disso, novos dados não foram coletados, o que impede afirmações definitivas sobre a capacidade de generalização dos modelos para diferentes tipos de cidades. Outra limitação é relativa ao simulador utilizado, que posiciona o gateway sempre no centro da célula, impossibilitando a avaliação de estratégias de posicionamento alternativas.

As implicações práticas e teóricas foram avaliadas através da criação de cenários simulados e comparados entre si, variando apenas uma característica de cada vez. Ao simular redes LoRa com distâncias maiores ou com menor distância e menor fator de espalhamento, o uso de modelos de propagação com menor erro pode levar a resultados diferentes. Se o tempo de simulação não for um problema, qualquer um dos três modelos baseados em árvore terá desempenho superior aos modelos clássicos em qualquer situação.

Para pesquisas futuras, sugere-se a coleta de novos dados em diferentes cidades, para comparar os resultados e treinar novamente os modelos de IA, avaliando e melhorando a capacidade de generalização. Além disso, a implementação de algoritmos de posicionamento de gateways e sua validação utilizando o simulador com modelos de erro menor, levaria a uma melhoria significativa. Além disso, sugere-se comparar os modelos com estratégias de estimação de perdas por propagação baseadas no relevo do terreno, como o uso de imagens de satélite, conforme revisado na bibliografia.

REFERÊNCIAS

ALLIOUI, H.; MOURDI, Y. Exploring the Full Potentials of IoT for Better Financial Growth and Stability: a comprehensive survey. **Sensors**, [S.l.], v. 23, n. 19, p. 8015, Sept. 2023.

ALMARHABI, A.; ALJOHANI, A. J.; MOINUDDIN, M. LoRa and High-Altitude Platforms: path loss, link budget and optimum altitude. *In*: INTERNATIONAL CONFERENCE ON INTELLIGENT AND ADVANCED SYSTEMS (ICIAS), 2020., 2021. **Proceedings [...]** IEEE, 2021.

ANZUM, R. *et al.* A Multiwall Path-Loss Prediction Model Using 433 MHz LoRa-WAN Frequency to Characterize Foliage's Influence in a Malaysian Palm Oil Plantation Environment. **Sensors**, [S.l.], v. 22, n. 14, p. 5397, July 2022.

BALLESTRIN, R. *et al.* Exploring Machine Learning Techniques for Path Loss Prediction in LoRa Networks. *In*: INTERNATIONAL SYMPOSIUM ON WIRELESS COMMUNICATION SYSTEMS (ISWCS), 2024., 2024. **Proceedings [...]** IEEE, 2024. p. 1–6.

BIANCO, G. M. *et al.* LoRa System for Search and Rescue: path-loss models and procedures in mountain scenarios. **IEEE Internet of Things Journal**, [S.l.], v. 8, 2021.

BOR, M. *et al.* Do LoRa low-power wide-area networks scale? *In*: MSWIM '16 PROCEEDINGS OF THE 19TH ACM INTERNATIONAL CONFERENCE ON MODELING, ANALYSIS AND SIMULATION OF WIRELESS AND MOBILE SYSTEMS, 2016. **Proceedings [...]** ACM Press, 2016. p. 59–67.

BREITEGGER, J. *et al.* Long-Term LoRa Experiments in a Chemical Plant. *In*: IEEE INTERNATIONAL CONFERENCE ON INDUSTRIAL TECHNOLOGY (ICIT), 2021., 2021. **Proceedings [...]** IEEE, 2021.

CALLEBAUT, G.; OTTOY, G.; VAN DER PERRE, L. Cross-Layer Framework and Optimization for Efficient Use of the Energy Budget of IoT Nodes. *In*: IEEE

WIRELESS COMMUNICATIONS AND NETWORKING CONFERENCE (WCNC), 2019., 2019. **Proceedings [...]** IEEE, 2019.

CALLEBAUT, G.; PERRE, L. V. D. Characterization of LoRa Point-to-Point Path Loss: measurement campaigns and modeling considering censored data. **IEEE Internet of Things Journal**, [S.l.], v. 7, 2020.

CATHERWOOD, P. *et al.* Channel characterisation for wearable LoRaWAN monitors. *In: LOUGHBOROUGH ANTENNAS & AMP PROPAGATION CONFERENCE (LAPC 2017)*, 2017. **Proceedings [...]** Institution of Engineering and Technology, 2017.

CHALL, R. E.; LAHOUD, S.; HELOU, M. E. LoRaWAN network: radio propagation models and performance evaluation in various environments in lebanon. **IEEE Internet of Things Journal**, [S.l.], v. 6, 2019.

CHEN, T.; GUESTRIN, C. XGBoost: a scalable tree boosting system. *In: ACM SIGKDD INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING*, 22., 2016. **Proceedings [...]** ACM, 2016. (KDD '16).

DEN ABEELE, F. V. *et al.* Scalability Analysis of Large-Scale LoRaWAN Networks in ns-3. **IEEE Internet of Things Journal**, [S.l.], v. 4, n. 6, p. 2186–2198, Dec. 2017.

EGLI, J. Radio Propagation above 40 MC over Irregular Terrain. **Proceedings of the IRE**, [S.l.], v. 45, n. 10, p. 1383–1391, 1957.

ERTEL, W. **Introduction to artificial intelligence**. 2. ed. Basel, Switzerland: Springer International Publishing, 2018. (Undergraduate Topics in Computer Science).

FARAHSAARI, P. S. *et al.* A Survey on Indoor Positioning Systems for IoT-Based Applications. **IEEE Internet of Things Journal**, [S.l.], v. 9, n. 10, p. 7680–7699, May 2022.

FERREIRA, A. E. *et al.* A study of the LoRa signal propagation in forest, urban, and suburban environments. **Annals of Telecommunications**, [S.l.], v. 75, p. 333–351, 8 2020.

GONZÁLEZ-PALACIO, M. *et al.* LoRaWAN Path Loss Measurements in an Urban Scenario including Environmental Effects. **Data**, [S.l.], v. 8, n. 1, p. 4, Dec. 2022.

GONZÁLEZ-PALACIO, M. *et al.* Machine-Learning-Based Combined Path Loss and Shadowing Model in LoRaWAN for Energy Efficiency Enhancement. **IEEE Internet of Things Journal**, [S.l.], v. 10, n. 12, p. 10725–10739, June 2023.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>.

- GUO, Q.; YANG, F.; WEI, J. Experimental Evaluation of the Packet Reception Performance of LoRa. **Sensors**, [S.l.], v. 21, n. 4, p. 1071, Feb. 2021.
- HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. **The Elements of Statistical Learning**. [S.l.]: Springer New York, 2009.
- HATA, M. Empirical formula for propagation loss in land mobile radio services. **IEEE Transactions on Vehicular Technology**, [S.l.], v. 29, n. 3, p. 317–325, aug 1980.
- HAYASHI, T.; ICHIGE, K. A Deep-Learning Method for Path Loss Prediction Using Geospatial Information and Path Profiles. **IEEE Transactions on Antennas and Propagation**, [S.l.], v. 71, n. 9, p. 7523–7537, Sept. 2023.
- HAYKIN, S. O. **Neural Networks and Learning Machines**. 3. ed. Upper Saddle River, NJ: Pearson, 2008.
- HUANG, C. *et al.* Artificial Intelligence Enabled Radio Propagation for Communications—Part II: scenario identification and channel modeling. **IEEE Transactions on Antennas and Propagation**, [S.l.], v. 70, n. 6, p. 3955–3969, June 2022.
- INOUE, K. *et al.* Learning-Based Prediction Method for Radio Wave Propagation Using Images of Building Maps. **IEEE Antennas and Wireless Propagation Letters**, [S.l.], v. 21, n. 1, p. 124–128, Jan. 2022.
- ISABONA, J. *et al.* Development of a Multilayer Perceptron Neural Network for Optimal Predictive Modeling in Urban Microcellular Radio Environments. **Applied Sciences**, [S.l.], v. 12, n. 11, p. 5713, June 2022.
- KIM, D.-H.; LEE, E.-K.; KIM, J. Experiencing LoRa Network Establishment on a Smart Energy Campus Testbed. **Sustainability**, [S.l.], v. 11, n. 7, p. 1917, Mar. 2019.
- LAU, L.; HE, J. Investigation into the Effect of Atmospheric Particulate Matter (PM_{2.5} and PM₁₀) Concentrations on GPS Signals. **Sensors**, [S.l.], v. 17, n. 3, p. 508, Mar. 2017.
- LIN, K.; HAO, T. Experimental Link Quality Analysis for LoRa-Based Wireless Underground Sensor Networks. **IEEE Internet of Things Journal**, [S.l.], v. 8, 2021.
- LIN, K. *et al.* Analysis of LoRa Link Quality for Underwater Wireless Sensor Networks: a semi-empirical study. *In: IEEE ASIA-PACIFIC MICROWAVE CONFERENCE (APMC)*, 2019., 2019. **Proceedings [...]** IEEE, 2019.

LO SCIUTO, G. Air pollution effects on the intensity of received signal in 3G/4G mobile terminal. **International Journal of Energy and Environmental Engineering**, [S.l.], v. 10, n. 2, p. 221–229, Feb. 2019.

LOH, W. R. *et al.* Intelligent Base Station Placement in Urban Areas With Machine Learning. **IEEE Antennas and Wireless Propagation Letters**, [S.l.], v. 22, n. 9, p. 2220–2224, Sept. 2023.

LORA ALLIANCE. **RP002-1.0.0 LoRaWAN® Regional Parameters**. Disponível em: <https://resources.lora-alliance.org/document/rp002-1-0-0-lorawan-regional-parameters>. Acesso em: 08 ago. 2024.

MAGRIN, D.; CENTENARO, M.; VANGELISTA, L. Performance evaluation of LoRa networks in a smart city scenario. *In*: IEEE INTERNATIONAL CONFERENCE ON COMMUNICATIONS (ICC), 2017., 2017. **Proceedings [...]** IEEE, 2017.

MENDES, B.; PASSOS, D.; CORREIA, N. Coverage Characterization of LoRaWAN Sensor Networks for Citrus Orchard Monitoring. *In*: INTERNATIONAL YOUNG ENGINEERS FORUM (YEF-ECE), 2022., 2022. **Proceedings [...]** IEEE, 2022.

MONTAGNY, S. **LoRa - LoRaWAN and Internet of Things for beginners**. 2. ed. [S.l.]: Savoie Mont Blanc University, 2022.

MOROCHO-CAYAMCELA, M. E.; MAIER, M.; LIM, W. Breaking Wireless Propagation Environmental Uncertainty With Deep Learning. **IEEE Transactions on Wireless Communications**, [S.l.], v. 19, n. 8, p. 5075–5087, Aug. 2020.

OLASUPO, T. O. Wireless communication modeling for the deployment of tiny iot devices in rocky and mountainous environments. **IEEE Sensors Letters**, [S.l.], v. 3, 2019.

OPEN TOPO DATA. **EU-DEM**. Disponível em: <https://www.opentopodata.org/datasets/eudem/>. Acesso em: 08 ago. 2024.

PARSONS, J. D. **The Mobile Radio Propagation Channel**. 2. ed. Chichester, England: John Wiley & Sons, 2000.

PETAJAJARVI, J. *et al.* On the coverage of LPWANs: range evaluation and channel attenuation model for lora technology. *In*: INTERNATIONAL CONFERENCE ON ITS TELECOMMUNICATIONS (ITST), 2015., 2015. **Proceedings [...]** [S.l.: s.n.], 2015. p. 55–59.

- PHAIBOON, S.; PHOKHARATKUL, P. An Empirical Model For 433 MHz LoRa-WAN in Ruby Mango Plantation. *In: INTERNATIONAL CONFERENCE ON ENGINEERING, APPLIED SCIENCES, AND TECHNOLOGY (ICEAST), 2023., 2023. Proceedings [...]* IEEE, 2023.
- PHAM, C.; EHSAN, M. Dense Deployment of LoRa Networks: expectations and limits of channel activity detection and capture effect for radio channel access. *Sensors*, [S.l.], v. 21, n. 3, p. 825, Jan. 2021.
- RABAÇA, A. F. B. **Aplicação de Tecnologia LoRaWAN à Monitorização de Redes de Distribuição de Energia**. 2018. 109 f. Dissertação (Mestrado) - Universidade de Lisboa. Instituto Superior Técnico de Lisboa, Lisboa, 2018.
- RADEMACHER, M. *et al.* Path Loss in Urban LoRa Networks: a large-scale measurement study. *In: IEEE 94TH VEHICULAR TECHNOLOGY CONFERENCE (VTC2021-FALL), 2021., 2021. Proceedings [...]* IEEE, 2021.
- RAPPAPORT, T. **Wireless Communications: principles and practice**. 2nd. ed. USA: Prentice Hall PTR, 2001.
- RENZONE, G. D. *et al.* LoRaWAN Underground to Aboveground Data Transmission Performances for Different Soil Compositions. **IEEE Transactions on Instrumentation and Measurement**, [S.l.], v. 70, 2021.
- REYNDERS, B.; WANG, Q.; POLLIN, S. A LoRaWAN module for ns-3. *In: WORKSHOP ON NS-3 - WNS3 '18, 10., 2018. Proceedings [...]* ACM Press, 2018.
- SCHERFKE, STEFAN. **Overview - SimPy 4.1.1 documentation**. Disponível em: <https://simpy.readthedocs.io/en/latest/>. Acesso em: 08 ago. 2024.
- SEMTECH CORPORATION. **SX1272/73 - 860 MHz to 1020 MHz Low Power Long Range Transceiver**. Disponível em: <https://www.semtech.com/products/wireless-rf/lora-connect/sx1272>. Acesso em: 08 ago. 2024.
- SEYBOLD, J. S. **Introduction to RF propagation**. [S.l.]: Wiley-Interscience, 2005.
- SHAKIR, Z. *et al.* Performance evaluation for RF propagation models based on data measurement for LTE networks. **International Journal of Information Technology**, [S.l.], v. 14, n. 5, p. 2423–2428, June 2022.
- SILVA, J. *et al.* A Survey of LoRaWAN Simulation Tools in ns-3. **Journal of Communication and Information Systems**, [S.l.], v. 36, n. 1, p. 17–30, 2021.

SINGH, Y. Comparison of Okumura, Hata and COST-231 Models on the Basis of Path Loss and Signal Strength. **International Journal of Computer Applications**, [S.l.], v. 59, n. 11, p. 37–41, Dec. 2012.

STUMPO, D. *et al.* Performance of Extended LoRaEnergySim Simulator in supporting Multi-Gateway scenarios and Interference Management. *In: IEEE/ACM 26TH INTERNATIONAL SYMPOSIUM ON DISTRIBUTED SIMULATION AND REAL TIME APPLICATIONS (DS-RT), 2022., 2022. Proceedings [...]* IEEE, 2022.

TO, T.-H.; DUDA, A. Simulation of LoRa in NS-3: improving LoRa performance with CSMA. *In: IEEE INTERNATIONAL CONFERENCE ON COMMUNICATIONS (ICC), 2018., 2018. Proceedings [...]* IEEE, 2018.

TORRE, P. V. *et al.* Measurement and characterization of dual-band LoRa communication in the Antarctic. *In: EUROPEAN CONFERENCE ON ANTENNAS AND PROPAGATION (EuCAP 2018), 12., 2018. Proceedings [...]* Institution of Engineering and Technology, 2018.

TRIPATHI, S. L.; Mahmud (Ed.). **Explainable machine learning models and architectures**. Indianapolis, IN: Sybex, 2023.

VILLARIM, M. R. *et al.* LoRa Performance Assessment in Dense Urban and Forest Areas for Environmental Monitoring. *In: INTERNATIONAL SYMPOSIUM ON INSTRUMENTATION SYSTEMS, CIRCUITS AND TRANSDUCERS (INSCIT), 2019., 2019. Proceedings [...]* IEEE, 2019.

XU, W. *et al.* Measurement, Characterization, and Modeling of LoRa Technology in Multifloor Buildings. **IEEE Internet of Things Journal**, [S.l.], v. 7, 2020.

YOUNG, W. R. Comparison of Mobile Radio Transmission at 150, 450, 900, and 3700 Mc. **Bell System Technical Journal**, [S.l.], v. 31, n. 6, p. 1068–1085, Nov. 1952.

ZHANG, H. *et al.* An Artificial Intelligence Radio Propagation Model Based on Geographical Information. **IEEE Transactions on Antennas and Propagation**, [S.l.], v. 70, n. 12, p. 12049–12060, Dec. 2022.

ZHAO, Y.; RAMOS, M. F.; LI, B. Integrated framework to integrate Spark-based big data analytics and for health monitoring and recommendation in sports using XGBoost algorithm. **Soft Computing**, [S.l.], v. 28, n. 2, p. 1585–1608, Dec. 2023.