

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

EVANDRO MONTEIRO JORGE JÚNIOR

**UTILIZAÇÃO DE INTELIGÊNCIA COMPUTACIONAL NO APRIMORAMENTO
DOS MODELOS DE PROPAGAÇÃO PERDA NO ESPAÇO LIVRE,
OKUMURA-HATA, COST 231, ECC-33, EGLI E A RECOMENDAÇÃO ITU-P.1546**

Uberlândia
2024

EVANDRO MONTEIRO JORGE JÚNIOR

**UTILIZAÇÃO DE INTELIGÊNCIA COMPUTACIONAL NO APRIMORAMENTO
DOS MODELOS DE PROPAGAÇÃO PERDA NO ESPAÇO LIVRE,
OKUMURA-HATA, COST 231, ECC-33, EGLI E A RECOMENDAÇÃO ITU-P.1546**

Exame de Tese apresentado à Universidade
Federal de Uberlândia, como parte dos
requisitos para a obtenção do título de
Doutorado em Engenharia Elétrica.

Prof. Orientador: Antônio Cláudio Paschoarelli Veiga

Membros da Banca:

Dr. Antonio C. P. Veiga (Presidente)/UFU

Dr. Alexandre Coutinho Mateus/UFU

Dr. Gilberto Arantes Carrijo/UFU

Dr. Daniel Moraes Santos/UFVJM

Dr. Tiago Nunes Santos/Algar Telecom

Uberlândia

2024

Ficha Catalográfica Online do Sistema de Bibliotecas da UFU
com dados informados pelo(a) próprio(a) autor(a).

J82 2024	<p>Jorge Júnior, Evandro Monteiro, 1989- Utilização de Inteligência Computacional no Aprimoramento dos Modelos de Propagação Perda no Espaço Livres, Okumura-Hata, Cost 231, Ecc-33, Egli e a Recomendação Itu-P.1546 [recurso eletrônico] / Evandro Monteiro Jorge Júnior. - 2024.</p> <p>Orientador: Antônio Cláudio Paschoarelli Veiga. Tese (Doutorado) - Universidade Federal de Uberlândia, Pós-graduação em Engenharia Elétrica. Modo de acesso: Internet. Disponível em: http://doi.org/10.14393/ufu.te.2024.791 Inclui bibliografia. Inclui ilustrações.</p> <p>1. Engenharia elétrica. I. Veiga, Antônio Cláudio Paschoarelli, 1963-, (Orient.). II. Universidade Federal de Uberlândia. Pós-graduação em Engenharia Elétrica. III. Título.</p> <p>CDU: 621.3</p>
-------------	---

Bibliotecários responsáveis pela estrutura de acordo com o AACR2:

Gizele Cristine Nunes do Couto - CRB6/2091
Nelson Marcos Ferreira - CRB6/3074



ATA DE DEFESA - PÓS-GRADUAÇÃO

Programa de Pós-Graduação em:	Engenharia Elétrica				
Defesa de:	Tese de Doutorado, 343 , PPGEELT				
Data:	Dezenove de dezembro de dois mil e vinte e quatro	Hora de início:	09:30	Hora de encerramento:	12:30
Matrícula do Discente:	11913EEL003				
Nome do Discente:	Evandro Monteiro Jorge Júnior				
Título do Trabalho:	1. UTILIZAÇÃO DE INTELIGÊNCIA COMPUTACIONAL NO APRIMORAMENTO DOS MODELOS DE PROPAGAÇÃO: PERDA NO ESPAÇO LIVRE, OKUMURA-HATA, COST 231, ECC-33, EGLI E A RECOMENDAÇÃO ITU-P.1546				
Área de concentração:	Processamento da Informação				
Linha de pesquisa:	Processamento Digital de Sinais e Redes de Comunicação				
Projeto de Pesquisa de vinculação:	Coordenador do projeto: Antonio Cláudio Paschoarelli Veiga. Título do projeto: Aplicações de machine learning para predição de sinais, estimação de parâmetros e reconhecimento de padrões. Agência financiadora: Não se aplica. Número do processo na agência financiadora: Não se aplica. Vigência do projeto: atual.				

Reuniu-se por meio de videoconferência, a Banca Examinadora, designada pelo Colegiado do Programa de Pós-graduação em Engenharia Elétrica, assim composta:

Doutores: Gilberto Arantes Carrijo (UFU), Alexandre Coutinho Mateus (UFU), Daniel Moraes Santos (UFVJM), Tiago Nunes Santos (Algar Telecom) e Antonio Cláudio Paschoarelli Veiga, orientador do discente.

Iniciando os trabalhos o presidente da mesa, Dr. Antonio Cláudio Paschoarelli Veiga, apresentou a Comissão Examinadora e o candidato, agradeceu a presença do público, e concedeu ao discente a palavra para a exposição do seu trabalho. A duração da apresentação do discente e o tempo de arguição e resposta foram conforme as normas do Programa.

A seguir, o senhor presidente concedeu a palavra, pela ordem sucessivamente, aos examinadores, que passaram a arguir o candidato. Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, atribuiu o resultado final, considerando o candidato:

APROVADO.

Esta defesa faz parte dos requisitos necessários à obtenção do título de Doutor. O competente diploma será expedido após cumprimento dos demais requisitos, conforme as normas do Programa, a legislação pertinente e a regulamentação interna da UFU.

Nada mais havendo a tratar foram encerrados os trabalhos. Foi lavrada a presente ata que após lida e achada conforme, foi assinada pela Banca Examinadora.



Documento assinado eletronicamente por **Antonio Claudio Paschoarelli Veiga, Professor(a) do Magistério Superior**, em 19/12/2024, às 12:35, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Daniel Moraes Santos, Usuário Externo**, em 19/12/2024, às 15:30, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Alexandre Coutinho Mateus, Professor(a) do Magistério Superior**, em 19/12/2024, às 17:59, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Tiago Nunes Santos, Usuário Externo**, em 20/12/2024, às 15:40, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Gilberto Arantes Carrijo, Usuário Externo**, em 20/12/2024, às 17:40, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_confirma&id_orgao_acesso_externo=0, informando o código verificador **5952179** e o código CRC **4C93922A**.

AGRADECIMENTOS

Ao concluir esta etapa tão significativa, expresso minha gratidão a todos que contribuíram para essa conquista. Primeiramente, à minha esposa, Sarah Londe Pessoa, pelo apoio constante e pela presença em cada momento dessa jornada.

Ao meu orientador, Antônio Cláudio Paschoarelli Veiga, agradeço pela orientação e confiança, essenciais para o desenvolvimento deste trabalho.

Aos meus pais, Evandro e Marta, minha gratidão por todo o suporte e pela base que me proporcionaram ao longo da vida.

Também agradeço à minha irmã Verônica e ao meu cunhado Márcio pelo apoio familiar, ao meu primo Rafael pela amizade, e aos amigos Frederico e Tiago pela parceria ao longo desse período.

Reconheço e agradeço o trabalho dos servidores da Faculdade de Engenharia Elétrica e da Biblioteca Eseba, que foram fundamentais para a realização deste projeto.

Por fim, agradeço à Maira, diretora do Sistema de Bibliotecas, e ao colega Josué pelo suporte no ambiente de trabalho.

Deixo aqui também meu sincero agradecimento a todos que, de alguma forma, contribuíram para essa trajetória, seja com palavras de incentivo, gestos de ajuda ou suporte nos momentos necessários.

A todos, meu muito obrigado por fazerem parte dessa conquista.

O objetivo da Ciência é tornar o belo e complexo, compreensível e simples – mas não menos maravilhoso. Hebert A. Simon

RESUMO

Neste trabalho de pesquisa, propõe-se o aprimoramento dos modelos de predição de perda de propagação de ondas eletromagnéticas por meio da integração de redes neurais artificiais, algoritmos genéticos e evolução diferencial. Os modelos tradicionais, como Perda no Espaço Livre, Okumura-Hata, Cost 231, ECC-33, Egli e a Recomendação ITU-P.1546, são utilizados em conjunto com esses métodos para a predição de perdas, possibilitando a geração de expressões analíticas para o cálculo das mesmas. Dessa forma, os novos algoritmos propostos são classificados como modelos híbridos, destacando-se por contraporem as abordagens tradicionais. Uma contribuição relevante deste estudo é a inclusão da altitude como um parâmetro adicional nas expressões analíticas, melhorando a precisão no cálculo da perda de percurso por meio dos modelos de predição desenvolvidos.

Palavras-chave: Algoritmos Genéticos, Inteligência Computacional, Evolução Diferencial, Propagação de Ondas, Redes Neurais Artificiais, Perda de Percurso

ABSTRACT

This research proposes the enhancement of electromagnetic wave propagation loss prediction models through the integration of artificial neural networks, genetic algorithms, and differential evolution. Traditional models, such as Free Space Loss, Okumura-Hata, Cost 231, ECC-33, Egli, and the ITU-P.1546 Recommendation, are combined with these methods to predict losses, enabling the generation of analytical expressions for their calculation. In this context, the proposed algorithms are classified as hybrid models, standing out by contrasting traditional approaches. A significant contribution of this study is the inclusion of altitude as an additional parameter in the analytical expressions, improving accuracy in path loss calculation through the developed prediction models.

Keywords: Genetic Algorithms, Computational Intelligence, Differential Evolution, Wave Propagation, Artificial Neural Networks, Path Loss.

SUMÁRIO

	Páginas
Lista de Figuras	12
Lista de Equações	14
Lista de Abreviaturas	17
Lista de Tabelas	18
Lista de Símbolos	20
1 Introdução	21
A Objetivos	25
A.1 Objetivo Geral	25
A.2 Objetivos Específicos	26
B Estrutura do trabalho	26
2 Revisão de Literatura	28
2.1 Considerações Finais	33
3 Modelos de propagação	35
3.1 Mecanismos de Perda no Espaço Livre	35
3.2 Perda no Espaço Livre	36
3.3 Modelo Okumura-Hata	37
3.3.1 Modelo Okumura	37
3.3.2 Modelo Hata	39
3.4 Cost 231 Hata Model	41
3.5 Modelo Egli	41
3.6 ECC-33	42
3.7 Recomendação ITU P-1546	43
3.8 Considerações finais	45
4 Redes Neurais Artificiais	46
4.1 Modelos de Neurônios	46
4.2 Tipos de função de ativação	48
4.2.1 Função Limiar	49
4.2.2 Função Sigmoide	50
4.2.3 Função Tangente Hiperbólica	51
4.3 Arquitetura da rede	52

4.3.1	Redes <i>Feedforward</i> de camada única	52
4.3.2	Redes <i>Feedforward</i> de múltiplas camadas	53
4.4	Estágios do Treinamento	55
4.4.1	Estágio de Propagação	55
4.4.2	Estágio de Treinamento	56
4.4.3	Estágio de ajuste dos pesos	56
4.5	Algoritmo de treinamento	56
4.5.1	Modificação Levenberg-Marquardt	59
4.6	Considerações finais	61
5	Algoritmos Genéticos e Evolução Diferencial	62
5.1	Algoritmos Genéticos	62
5.1.1	Descrição do Método	63
5.1.2	População inicial	64
5.1.3	Seleção	64
5.1.4	Cruzamento	64
5.1.5	Mutação	65
5.1.6	Critério de parada	65
5.1.7	Função Aptidão	66
5.2	Evolução Diferencial	66
5.2.1	População inicial	67
5.2.2	Mutação	67
5.2.3	Cruzamento	67
5.2.4	Seleção	68
5.3	Considerações finais	68
6	Metodologia	69
6.1	Sistema de Transmissão	69
6.2	Sistema de Recepção	71
6.3	Processamento dos dados	73
6.4	Desenvolvimento das RNAs	76
6.5	Processo de Otimização por Algoritmo Genético e Evolução Diferencial	79
6.5.1	Configuração dos parâmetros do Algoritmo Genético	81
6.5.2	Configuração dos parâmetros da Evolução Diferencial	82
6.6	Considerações Finais	82
7	Resultados da predição obtida pela RNA	83
7.1	Modelo Híbrido Perda no Espaço Livre e RNA	83
7.2	Modelo Híbrido Cost-231 e RNA	85
7.3	Modelo Híbrido Egli e RNA	87

7.4	Modelo Híbrido ECC-33 e RNA	88
7.5	Modelo Híbrido Okumura-Hata e RNA	90
7.6	Modelo Híbrido ITU-P.1546-6 e RNA	93
7.7	Comparação com outros estudos	94
7.8	Considerações Finais	96
8	Resultados da otimização por Algoritmos Genéticos e Evolução Diferencial	97
8.1	Otimização do Modelo Okumura-Hata	97
8.2	Otimização do Modelo ECC-33	102
8.3	Otimização do Modelo Egli	107
8.4	Otimização do Modelo Cost 231	113
8.5	Ranqueamento entre os modelos e métodos	116
8.6	Considerações Finais	121
9	Conclusões	122
9.1	Conclusões acerca do aprimoramento dos modelos de propagação por Redes Neurais Artificiais	123
9.2	Conclusões acerca da otimização dos modelos de propagação por Algoritmos Genéticos e Evolução Diferencial	124
9.3	Comparação entre os resultado obtidos pela RNA e os algoritmos evolutivos . .	125
9.4	Principais Contribuições	126
9.5	Sugestões para Trabalhos Futuros	126
	Referências	128
	Apêndice A	133
	Apêndice B	134
	Apêndice C	137
	Apêndice D	140
	Anexo A	143

LISTA DE FIGURAS

Figura 1.1 - Variação de sinal em pequena escala e grande escala em função da distância	22
Figura 1.2 - Exemplo de causas de perda de percurso	23
Figura 3.1 - Mecanismos de Propagação e Perdas	36
Figura 3.2 - Atenuação média em relação ao espaço livre sobre um terreno quase-liso . .	38
Figura 3.3 - Fator de Correção G_{AREA} para diferentes tipos de terrenos	39
Figura 3.4 - Curvas de intensidade de campo para frequência de 100 MHz e propagação em terra 50% do tempo	44
Figura 4.1 - Modelo de Neurônio não linear	47
Figura 4.2 - Transformação afim causado pelo <i>bias</i>	48
Figura 4.3 - Representação gráfica da função limiar	49
Figura 4.4 - Representação gráfica da função sigmoide	50
Figura 4.5 - Representação gráfica da função sigmoide bipolar	51
Figura 4.6 - Representação gráfica da função tangente hiperbólica	52
Figura 4.7 - RNA de camada única	53
Figura 4.8 - RNA multicamadas	54
Figura 4.9 - RNA Multicamadas com denominações	55
Figura 5.1 - Fluxograma do Algoritmo Genético	63
Figura 5.2 - Técnicas de Cruzamento	65
Figura 5.3 - Operador Mutação	65
Figura 6.1 - Visão dianteira e traseira do transmissor DVI3400	70
Figura 6.2 - Cabo utilizado na linha de transmissão	70
Figura 6.3 - Sistema de Recepção	72
Figura 6.4 - Trajeto onde foram realizadas as medições	73
Figura 6.5 - Mapa com a perda de percurso medida	75
Figura 6.6 - Perda de percurso <i>versus</i> a distância	76
Figura 6.7 - Representação das entradas e saída da RNA	77
Figura 6.8 - Fluxograma com os passos realizados	78
Figura 7.1 - Comparação entre perda de percurso, Perda no Espaço Livre e Perda Espaço Livre-RNA	84
Figura 7.2 - Comparação entre perda de percurso, Cost 231 e Cost231-RNA	86
Figura 7.3 - Comparação entre perda de percurso, Egli e Egli-RNA	88
Figura 7.4 - Comparação entre perda de percurso, ECC-33 e ECC33-RNA	90
Figura 7.5 - Comparação entre perda de percurso, Okumura Hata e Okumura Hata-RNA	92
Figura 7.6 - Comparação entre perda de percurso, ITU-P.1546-6 e ITU-P.1546-6-RNA .	94

Figura 8.1 - PL com otimização por AG (A), DE (B) e pelo modelo Okumura-Hata . . .	100
Figura 8.2 - Ampliação realizada na Figura 8.1 para melhorar a compreensão do efeito da adição do parâmetro altitude	101
Figura 8.3 - Predição de perda de percurso pelo AG, ED e modelo ECC-33	106
Figura 8.4 - Predição de perda de percurso pelo modelo Egli e suas otimizações por AG, ED e AG, ED e altitude	110
Figura 8.5 - Predição de perda de percurso pelo modelo Cost 231 e suas otimizações por AG, ED e AG, ED e altitude	114
Figura 8.6 - Predição de perda de percurso pelo modelo Cost 231 e suas otimizações por AG, ED e AG, ED e altitude	116

LISTA DE EQUAÇÕES

Equação 3.1	36
Equação 3.2	37
Equação 3.3	37
Equação 3.4	37
Equação 3.5	38
Equação 3.6	40
Equação 3.7	40
Equação 3.8	40
Equação 3.9	40
Equação 3.10	40
Equação 3.11	41
Equação 3.12	41
Equação 3.13	41
Equação 3.14	42
Equação 3.15	42
Equação 3.16	43
Equação 3.17	43
Equação 3.18	43
Equação 3.19	43
Equação 3.20	43
Equação 3.21	43
Equação 4.1	47
Equação 4.2	47
Equação 4.3	48
Equação 4.4	49
Equação 4.5	49
Equação 4.6	49
Equação 4.7	50
Equação 4.8	51
Equação 4.9	51
Equação 4.10	52
Equação 4.11	54
Equação 4.12	56
Equação 4.13	57
Equação 4.14	57
Equação 4.15	57
Equação 4.16	57

Equação 4.17	57
Equação 4.18	57
Equação 4.19	57
Equação 4.20	58
Equação 4.21	58
Equação 4.22	58
Equação 4.24	59
Equação 4.25	59
Equação 4.26	59
Equação 4.27	60
Equação 4.28	60
Equação 4.29	60
Equação 4.30	60
Equação 6.2	74
Equação 6.3	78
Equação 6.4	78
Equação 6.5	78
Equação 6.6	78
Equação 6.7	79
Equação 6.8	79

LISTA DE ABREVIATURAS

AG	Algoritmo Genético
Anatel	Agência Nacional de Telecomunicações Anatel
BP	<i>Backpropagation</i>
CDMA	Acesso Múltiplo por Divisão de Código
CoDE	Evolução Diferencial Composta
Cost	Cooperação Européia em ciência e tecnologia
dB	Decibel
ED	Evolução Diferencial
DTDNN	<i>Distributed Time Delay Neural Network</i>
ELM	<i>Extreme Learning Machine</i>
EM	Erro Médio
EMA	Erro Médio Absoluto
EQM	Erro Quadrático Médio
E _{RMS}	Erro RMS
FTDNN	<i>Focused Time Delay Neural Network</i>
GHz	Gigahertz
GPS	Sistema de Posicionamento Global
GSM	Sistema Global para dispositivos móveis
ITU	União Internacional das Telecomunicações
km	Quilômetros
LIBSVM	Biblioteca para Máquina de Vetores de Suporte
LM	Levenber-Marquardt
LOS	Linha de Visada
LRNN	<i>Layer Recurrent Neural Network</i>
m	metros
MHz	Megahertz
MLP	Perceptron Multicamadas
MWMF	Multifrequência <i>multiwall</i>
NLOS	Sem Linha de Visada
PL	Perda de Percurso
R ²	Coeficiente de Determinação
RBF	Função de Base Radial
RFS	<i>Radio Frequency Systems</i>
R _x	Antena Receptora
SCG	<i>Scale Conjugate Gradient</i>
SG3	Grupo de Estudo em Radiocomunicação 3
SHLFNN	Rede Neural Avançada de Camada Oculta Única
SPM	Modelo de Propagação Padrão
SVM	Máquina de Vetores de Suporte
T _x	Antena Transmissora

LISTA DE TABELAS

Tabela 6.1	Características do Sistema Irradiante	71
Tabela 6.2	Descrição estatística do conjunto de dados medidos	74
Tabela 7.1	EQM para o treinamento variando neurônios na camada escondida para modelo híbrido Perda no Espaço Livre-RNA	83
Tabela 7.2	Comparação entre os erros obtidos pelos modelos Perda no Espaço Livre e o híbrido Perda no Espaço Livre-RNA	85
Tabela 7.3	EQM para o treinamento variando neurônios na camada escondida para modelo híbrido Cost 231-RNA	85
Tabela 7.4	Comparação entre os erros obtidos pelos modelos Cost-231 e o híbrido Cost 231-RNA	86
Tabela 7.5	EQM para o treinamento variando neurônios na camada escondida para modelo híbrido Egli-RNA	87
Tabela 7.6	Comparação entre os erros obtidos pelos modelos Egli e o híbrido Egli-RNA	88
Tabela 7.7	EQM para o treinamento variando neurônios na camada escondida para modelo híbrido ECC-33-RNA	89
Tabela 7.8	Comparação entre o ECC-33 e o híbrido ECC-33-RNA	90
Tabela 7.9	EQM para o treinamento variando neurônios na camada escondida para modelo híbrido Okumura-Hata-RNA	91
Tabela 7.10	Comparação entre os erros obtidos pelos modelos Okumura-Hata e o híbrido Okumura-Hata-RNA	92
Tabela 7.11	EQM para o treinamento variando neurônios na camada escondida para modelo híbrido ITU-P.1546-6-RNA	93
Tabela 7.12	Comparação entre os erros obtidos pelos modelos ITU-P.1546-6 e o híbrido ITU-P.1546-6-RNA	94
Tabela 7.13	Resultados obtidos em outros estudos	95
Tabela 7.14	Resultados obtidos neste trabalho	95
Tabela 8.1	Valores das variáveis otimizadas pelo AG e Erro Quadrático Médio para o modelo Okumura-Hata	97
Tabela 8.2	Valores das variáveis otimizadas pelo ED e Erro Quadrático Médio para o modelo Okumura-Hata com adição da altitude	98
Tabela 8.3	Comparação entre o Modelo Okumura-Hata e suas otimizações	102
Tabela 8.4	Valores das variáveis otimizadas pelo ED e Erro Quadrático Médio para o modelo ECC-33	102
Tabela 8.5	Valores das variáveis otimizadas pelo ED e Erro Quadrático Médio para o modelo ECC-33 com adição da altitude	104

Tabela 8.6	Comparação entre o ECC-33 e suas otimizações	107
Tabela 8.7	Valores das variáveis otimizadas pelo ED e Erro Quadrático Médio para o modelo Egli	107
Tabela 8.8	Valores das variáveis otimizadas pelo ED e Erro Quadrático Médio para o modelo Egli com adição da altitude	108
Tabela 8.9	Comparação entre o Egli e suas otimizações	111
Tabela 8.10	Valores das variáveis otimizadas pelo AG e Erro Quadrático Médio para o modelo Cost 231	111
Tabela 8.11	Valores das variáveis otimizadas pelo ED e Erro Quadrático Médio para o modelo Cost-231 com adição da altitude	112
Tabela 8.12	Comparação entre o Modelo Cost 231 e suas otimizações	115
Tabela 8.13	Comparação entre os modelos, e otimização por AG, ED e otimizações com adição do parâmetro altitude	115
Tabela 8.14	Ranqueamento do modelo Okumura-Hata com relação aos Algoritmos de Otimização	117
Tabela 8.15	Ranqueamento do modelo ECC-33 com relação aos Algoritmos de Otimização	117
Tabela 8.16	Ranqueamento do modelo Egli com relação aos Algoritmos de Otimização .	117
Tabela 8.17	Ranqueamento do modelo Cost 231 com relação aos Algoritmos de Otimização	118
Tabela 8.18	Ranqueamento dos modelos com relação aos algoritmos sem otimização . .	118
Tabela 8.19	Ranqueamento dos modelos com relação ao AG	118
Tabela 8.20	Ranqueamento dos modelos com relação ao AG com a adição do Parâmetro Altitude	119
Tabela 8.21	Ranqueamento dos modelos com relação a ED	119
Tabela 8.22	Ranqueamento dos modelos com relação a ED com adição do parâmetro altitude	119
Tabela 8.23	Ranqueamento Geral entre os modelos de predição e Algoritmos de otimização	120
Tabela 9.1	Comparação entre o Modelo Cost 231 e suas otimizações	125

LISTA DE SÍMBOLOS

σ - Desvio Padrão

δ_j - Fator utilizado na correção dos pesos no algoritmo *backpropagation*

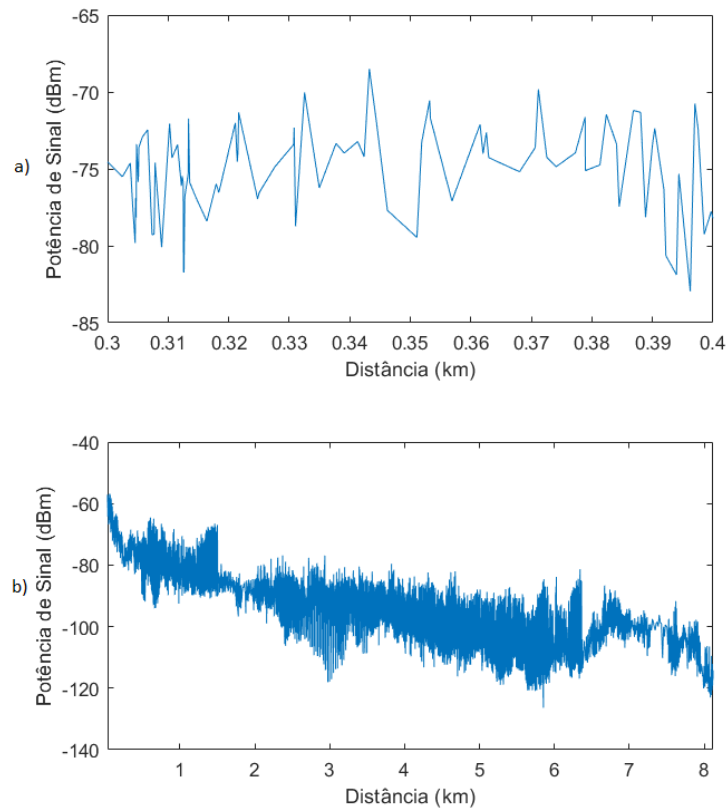
$\varphi(\cdot)$ - Função de Ativação

∇ - Operador Nabla

O estudo da propagação de ondas de rádio é fundamental para o projeto e a otimização de sistemas de radiodifusão. A maneira como as ondas eletromagnéticas se propagam pelo ar e interagem com o ambiente é complexa e influenciada por diversos fenômenos físicos, que afetam significativamente o alcance, a qualidade e a confiabilidade dos sinais transmitidos. Dentre os principais fenômenos, destacam-se:

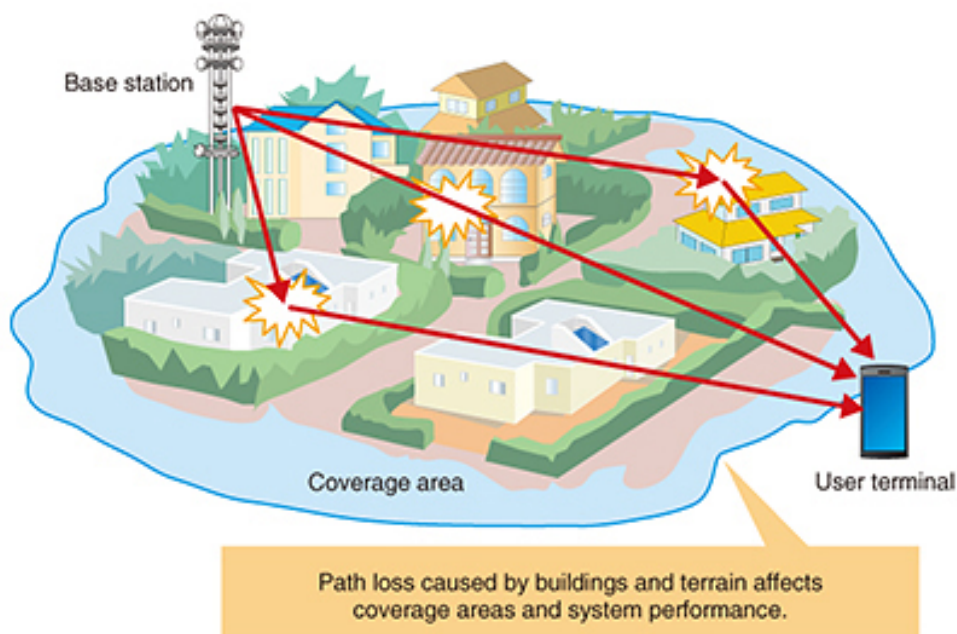
- Reflexão: ocorre quando as ondas de rádio incidem sobre superfícies com dimensões muito maiores que o seu comprimento de onda, como o solo, lagos, rios, edifícios e grandes formações rochosas. A intensidade da onda refletida depende das propriedades elétricas do material refletor e do ângulo de incidência.
- Difração: Manifesta-se quando as ondas encontram obstáculos com dimensões comparáveis ao seu comprimento de onda ou com bordas irregulares, como montanhas, prédios e árvores. A difração permite que as ondas contornem esses obstáculos, propagando-se em regiões de "sombra geométrica" onde não haveria recepção direta.
- Espalhamento: ocorre em meios com uma alta densidade de pequenos objetos, cujas dimensões são da ordem do comprimento de onda ou menores, como folhagens densas, chuva, neblina e partículas em suspensão na atmosfera. O espalhamento causa a dispersão da energia da onda em múltiplas direções, reduzindo a intensidade do sinal na direção original de propagação.

Outros fenômenos relevantes incluem absorção, interferência e refração. Em curtas distâncias, a combinação desses efeitos resulta em rápidas flutuações na intensidade do sinal recebido (Figura 1.1a). Em longas distâncias (Figura 1.1b), a atenuação cumulativa devido à absorção e espalhamento, combinada com a expansão da frente de onda, leva à chamada perda de percurso (PL), que representa a redução da potência do sinal com a distância (FARUK et al., 2021).

Figura 1.1 – Variação de sinal em pequena escala e grande escala em função da distância

Fonte: Autor

A PL representa a diferença entre a potência do sinal recebido e a potência do sinal transmitido em um determinado ambiente de propagação, conforme ilustrado na Figura 1.2 (FENG et al., 2015). Em razão disso, a PL desempenha um papel crucial no planejamento de sistemas de transmissão, permitindo a previsão da potência recebida na região a ser atendida e auxiliando nos estudos de viabilidade, além de facilitar o gerenciamento durante a implantação do sistema.

Figura 1.2 – Exemplo de causas de perda de percurso

Fonte: Sasaki et al. (2016)

Um sistema de telecomunicações pode ser dividido em três partes principais: transmissão, canal e recepção. A modelagem da PL é essencial nesses projetos, pois envolve custos significativos na instalação, manutenção e operação dessas três partes:

- Transmissão;
- Canal;
- Recepção.

Essa modelagem permite atender a toda a área desejada e alcançar o público-alvo. Por essa razão, diversos modelos de previsão de PL como por exemplo os modelos: Okumura-Hata, Cost 231, ECC-33, Egli e a Recomendação ITU-P.1546 foram desenvolvidos e continuamente aprimorados como citado adiante no Capítulo 2, tornando-se mais simples e precisos. Esses modelos ajudam a prever a área de cobertura de um sistema de transmissão e otimizar o desempenho dos enlaces (AYADI; ZINEB; TABBANE, 2017).

Diversos estudos foram realizados com o objetivo de prever a PL, como mencionado no parágrafo anterior, além dos modelos SUI e Ericsson. Com base nesses estudos, foram desenvolvidos modelos de predição da PL, os quais podem ser classificados em analíticos e empíricos. Os modelos analíticos são fundamentados em princípios físicos e exigem um conhecimento aprofundado da propagação das ondas de rádio, bem como a capacidade de realizar cálculos matemáticos complexos e utilizar recursos computacionais avançados. Por outro lado, os modelos empíricos baseiam-se em uma abordagem estatística que estabelece a relação entre as variáveis dependentes da PL (frequência e distância) e uma série de

parâmetros mensurados, como altitude, altura do transmissor, altura do receptor e distância entre os dois. Esses modelos podem ser desenvolvidos de maneira mais simples, com base na medição de dados de atenuação. A principal vantagem dos modelos empíricos reside em sua simplicidade e facilidade de aplicação em diferentes áreas experimentais, quando comparados aos modelos analíticos.

A PL pode ser modelada matematicamente utilizando técnicas de regressão, uma abordagem estatística que busca estabelecer uma relação entre variáveis dependentes e independentes. Em um modelo de regressão, é necessário definir claramente os parâmetros de entrada e saída. Os parâmetros de entrada incluem características como as informações do transmissor (potência transmitida, localização), do receptor (altura, localização), a frequência do sinal, entre outros fatores que influenciam a propagação da onda eletromagnética. A variável de saída é a PL, que representa a diminuição da potência do sinal conforme ele se desloca do transmissor para o receptor (AYADI; ZINEB; TABBANE, 2017).

O objetivo da regressão é identificar uma função matemática que relacione os parâmetros de entrada à variável de saída, de modo a permitir a previsão da PL em diferentes cenários. Esse processo envolve a minimização do erro entre os valores estimados pela função e os valores observados nos dados reais, utilizando métodos como a regressão linear, não linear ou polinomial, dependendo da complexidade do problema. A regressão é utilizada em modelagem de propagação, pois permite gerar modelos preditivos eficientes, adaptando-se às variáveis específicas do ambiente e fornecendo uma ferramenta útil para o planejamento e otimização de sistemas de telecomunicações (AYADI; ZINEB; TABBANE, 2017).

Na previsão de PL, a Inteligência Computacional se destaca ao complementar as técnicas de regressão com algoritmos inteligentes e autoaprendizes, inspirados em processos naturais. Essa abordagem resulta em modelos preditivos mais precisos e adaptáveis, cruciais para lidar com a complexidade e a dinâmica inerentes a esse tipo de previsão (IEEE, 2024).

Esse campo é tradicionalmente baseado em três principais ramos:

- Redes Neural Artificial (RNA),
- Computação Evolutiva
- Sistemas Fuzzy.

RNAs são sistemas de processamento de informações cuja estrutura e funcionalidade simulam os sistemas nervosos e, particularmente, o cérebro de animais e seres humanos. Eles consistem em um grande número de unidades de processamento bastante simples, os neurônios, que trabalham em paralelo. Esses neurônios enviam informações na forma de potenciais de ação através de enlaces direcionados para outros neurônios. Com base no conhecimento sobre a funcionalidade das redes neurais biológicas, tenta-se modelar e imitá-las, especialmente para alcançar a capacidade de aprendizado (IEEE, 2024). No Capítulo 4 será explicado mais detalhadamente sobre esta técnica.

Algoritmos evolutivos são métodos utilizados em otimização e busca de resultados que se inspiram nos princípios da evolução biológica. Eles operam com uma população de soluções potenciais, aplicando operadores inspirados pela natureza, como seleção, mutação e cruzamento, para evoluir as soluções ao longo do tempo (IEEE, 2024). Neste trabalho utilizou-se dois desses métodos: Algoritmo Genético (AG) e Evolução Diferencial (ED). Os AGs são uma classe específica de algoritmos evolutivos que utilizam conceitos de hereditariedade, mutação, seleção natural e recombinação (ou *crossing over*). Eles são implementados como uma simulação computacional onde uma população de soluções é selecionada em busca de soluções melhores. A evolução começa com um conjunto de soluções geradas aleatoriamente e é realizada através de gerações. Em cada geração, a adaptação de cada solução é avaliada, alguns indivíduos são selecionados para a próxima geração, e são recombinados ou mutados para formar uma nova população. Já a ED é outro tipo de algoritmo evolutivo que é particularmente eficaz na otimização de problemas contínuos. Ele trabalha com uma população de soluções candidatas que são submetidas a operações de mutação, cruzamento e seleção. A principal característica da ED é a utilização de diferenças ponderadas entre soluções existentes para perturbar as soluções candidatas atuais, o que ajuda a explorar o espaço de busca de maneira eficiente. Esses dois algoritmos serão explicados detalhadamente no Capítulo 5.

Também fazem parte do ramo da Inteligência Computacional os sistemas fuzzy porém estes não serão abordados uma vez que não fizeram parte deste trabalho.

Desde dezembro de 2018, grandes centros urbanos no Brasil realizaram a migração da TV analógica para a digital (??). Nesse contexto, a Agência Nacional de Telecomunicações do Brasil (Anatel) adotou o modelo de propagação ITU-R P.1546 como referência para o cálculo da PL no Sistema Brasileiro de Televisão Digital (SBTVD). No entanto, a Anatel permite que o projetista escolha o modelo de propagação que melhor se adapte às características específicas da área de estudo (BOTELHO, 2019). Essa flexibilidade na escolha do modelo de propagação é uma das principais justificativas para o desenvolvimento deste estudo, que visa explorar alternativas de modelagem para otimizar a previsão da PL em diferentes cenários.

A OBJETIVOS

A.1 OBJETIVO GERAL

O objetivo geral deste trabalho é combinar técnicas avançadas de Inteligência Computacional com métodos tradicionais reconhecidos na literatura para aprimorar a precisão das previsões de PL. Utilizando dados de potência coletados em diversos pontos da cidade de Uberlândia, o estudo propõe analisar as informações do sistema irradiante da TV Integração (569,142857 MHz), conforme detalhado no Anexo A, com o intuito de desenvolver um modelo preditivo mais robusto e preciso para a PL.

A.2 OBJETIVOS ESPECÍFICOS

- Desenvolver modelos híbridos para a predição da PL, a partir dos dados de potência coletados, combinando os modelos tradicionais Okumura-Hata, Cost 231, Egli, ECC-33 e a Recomendação ITU P.1546-6 com RNAs.
- Analisar qual configuração (quantidade de neurônios na camada escondida) da RNA apresenta melhor desempenho para cada tipo de modelo híbrido, ou seja, modelos tradicionais combinados com RNA, e identificar qual modelo desenvolvido apresenta o melhor desempenho.
- Otimizar os modelos de predição citados no primeiro item (exceto a Recomendação ITU P.1546-6, que não possui equação analítica), utilizando dois tipos de algoritmos evolucionais: AG e ED e classificá-los a partir de um ranqueamento.
- Apresentar como principal contribuição a adição da altitude como parâmetro para o cálculo da PL, através de um modelo de predição que gere equações analíticas mais precisas do que os resultados obtidos pelos modelos de propagação tradicionais.
- Investigar a influência da altitude no cálculo da PL e comparar os resultados obtidos com os resultados práticos. Esta análise se justifica, uma vez que a maioria dos modelos tradicionais de perda de propagação não considera esse parâmetro, que atualmente pode ser facilmente obtido por meio de equipamentos de Sistema de Posicionamento Global (GPS) ou através de *Application Programming Interface* (API) de sites de georreferenciamento.

B ESTRUTURA DO TRABALHO

Neste Capítulo, Introdução, o trabalho apresenta o contexto geral da pesquisa, destacando a importância da melhoria dos modelos de propagação de sinais em redes de comunicação. São delineados os objetivos gerais e específicos da pesquisa, assim como a estrutura do trabalho.

No Capítulo 2, é realizada uma revisão bibliográfica. Um levantamento de trabalhos similares é apresentado neste capítulo, com uma descrição de cada estudo.

O Capítulo 3, Modelos de propagação, detalha os modelos teóricos de propagação de sinais, desde os mecanismos de perda no espaço livre até modelos específicos como Okumura-Hata, Cost 231 Hata, Egli, ECC-33 e a recomendação ITU P-1546. Além disso, são discutidos os sistemas de coleta de dados, abrangendo tanto o sistema de transmissão quanto o de recepção, e o processo de processamento desses dados.

O Capítulo 4 apresenta a RNA, ele explora os fundamentos dessa técnica, começando pelos modelos de neurônios e diferentes tipos de funções de ativação. São apresentadas também as arquiteturas de rede, como as redes *feedforward* de camada única e múltiplas camadas,

e os estágios do treinamento, incluindo o algoritmo de treinamento Modificação Levenberg-Marquardt (LM).

No Capítulo 5 são abordados os AGs e a ED, descrevendo os métodos, a estrutura básica, os operadores genéticos (seleção, cruzamento e mutação), critérios de parada, função aptidão, além do processo de otimização utilizando esses algoritmos.

O Capítulo 6 apresenta a metodologia adotada neste trabalho, iniciando pela descrição detalhada do sistema de transmissão e recepção, essenciais para a coleta de dados. Em seguida, são abordados os processos de tratamento e processamento dos dados, fundamentais para garantir a qualidade das informações utilizadas. O capítulo também explora o desenvolvimento das RNAs e sua integração ao sistema, além de descrever o processo de otimização realizado por meio de AG e ED, com destaque para a configuração de seus parâmetros. Por fim, são apresentadas as considerações finais relacionadas às etapas metodológicas.

O Capítulo 7, intitulado Resultados da predição obtida pela RNA, apresenta os resultados da aplicação das RNAs nos modelos de propagação, detalhando os diferentes modelos híbridos criados com as RNAs e comparando-os com estudos anteriores.

Semelhante ao capítulo anterior, o Capítulo 8, exibe os resultados da otimização dos modelos de propagação utilizando AGs e ED, destacando a eficácia desses métodos na melhoria dos modelos.

Finalmente, o Capítulo 9, Conclusões, sintetiza os principais metas do trabalho, discutindo os resultados alcançados e fornecendo *insights* sobre o aprimoramento dos modelos de propagação por meio de RNAs e algoritmos evolutivos. Também são feitas comparações entre os resultados obtidos pelas RNAs e pelos algoritmos evolutivos, destacando as contribuições de cada abordagem.

REVISÃO DE LITERATURA

Resumo do capítulo

Este Capítulo apresenta a revisão de literatura sobre redes neurais aplicadas à previsão de PL. Trabalhos relevantes, como Popescu et al. (2001) e Cerri et al. (2004), exploraram modelos para ambientes urbanos, enquanto outros estudos, como Kalakh, Kandil e Hakem (2012), abordaram cenários específicos, como mineração. Resultados consistentes demonstram que a RNA supera modelos empíricos tradicionais, sendo eficaz para diferentes ambientes e configurações.

Muitos pesquisadores desenvolveram modelos de RNAs para previsões de PL, neste capítulo é realizado um resumo de quais trabalhos foram consultados no decorrer do desenvolvimento dessa pesquisa. Popescu et al. (2001) desenvolveram redes neurais para a previsão de PL em ambientes urbanos, dois modelos de redes neurais foram considerados: para casos de Linha de Visada (LOS) e Sem Linha de Visada (NLOS). No caso de LOS, dados relevantes (incluindo distância do transmissor e do receptor, altura dos edifícios, largura das ruas, separação do edifício e posição da antena transmissora em relação ao telhado) foram usadas para treinar a RNA. Para o caso NLOS, dois modelos de RNA foram construídos. O primeiro modelo foi treinado com os mesmos parâmetros do caso de LOS, enquanto o segundo modelo foi construído com os mesmos parâmetros, incluindo também as perdas por difração, calculadas pelo algoritmo COST231-Walfisch-Ikegami. A rede híbrida, que combinou algoritmos de PL, dados de rede e a abordagem de rede neural, produziu melhores resultados.

Cerri et al. (2004) sugeriram o uso de uma RNA perceptron multicamadas (MLP) para previsão de PL em áreas urbanas. A RNA foi treinada usando o algoritmo *backpropagation* (BP). A comparação dos resultados com modelos analíticos mostrou que o método proposto é eficiente para planejamento e otimização em enlaces de rádio. Sotiroudis e Siakavara (2015) também propuseram o uso de modelos de RNA para predição de PL em áreas urbanas. Em um trabalho semelhante às outras pesquisas, eles propuseram descobrir quais informações deveriam ser usadas como entrada da rede. A informação de entrada da rede utilizada foi o perfil do terreno do ambiente de propagação e os dados fornecidos foram de pouca quantidade. Os resultados obtidos com a RNA proposta provaram que o modelo foi capaz de prever de forma eficiente a PL em ambiente urbano com erro inferior aos pré-existentes (construído aleatoriamente). Um fato importante que merece ser enfatizado foi a utilização de um pequeno número de entradas com especificações detalhadas da propagação, que rendeu um melhor desempenho com relação às metodologias tradicionais.

Uma análise de modelos empíricos e RNA para previsão de PL foi investigada por Popescu, Nafornta e Constantinou (2005) e Eichie et al. (2017b). Os parâmetros de entradas da RNA foram: distância entre o transmissor e receptor, altitude e potência transmitida. Os parâmetros de saída da RNA foram o PL. Esses parâmetros foram coletados em vias

suburbanas e rurais, percorridas com um veículo. Os resultados de previsão do modelo de RNA foram comparados com os obtidos através dos modelos básicos de PL: Egli, Cost 231, Hata e Ericsson. O modelo de PL baseado em RNA apresentou melhores resultados quando comparado aos modelos básicos. Os autores concluíram que o modelo RNA pode realizar uma previsão precisa de PL em ambientes de propagação rurais e suburbanos.

Ostlin, Zepernick e Suzuki (2010) sugeriram o uso de RNA para previsão de PL de macrocélula. Para treinar e testar o modelo, foram utilizados dados obtidos de uma rede de Acesso Múltiplo por Divisão de Código (CDMA) comercial. A arquitetura de rede neural *feedforward* foi usada juntamente com dois algoritmos de *Backpropagation* (BP): LM e *Scale Conjugate Gradient* (SCG). A distância em relação à antena transmissora, o tipo de vegetação, solo, os parâmetros do perfil do percurso do terreno e a densidade de construções perto da antena receptora foram usados como entradas. Os resultados da previsão de PL do modelo RNA foram comparados com dois outros modelos de propagação (Recomendação ITU-R P.1546 e o modelo Okumura-Hata) e mostrou-se que o modelo RNA treinado produziu resultados mais precisos que os outros dois modelos. Embora a pesquisa tenha dado bons resultados, grandes mudanças de PL não puderam ser previstas, pois havia ausência de dados topográficos.

Piacentini e Rinaldi (2011) propuseram o uso de técnicas de aprendizado de máquina e redução dimensional para previsão de PL. Esses autores avaliaram os efeitos da redução dimensional na precisão da previsão de PL. Além disso, foram examinadas as habilidades das RNAs e Máquina de Vetores de Suporte (SVM) para resolver com eficiência o problema de regressão. Um perceptron de duas camadas foi escolhido como configuração da RNA e sua capacidade de generalização foi avaliada usando a técnica de validação cruzada. As SVMs foram treinadas usando a Biblioteca para Máquina de Vetores de Suporte (LIBSVM) e o Fortran 90 foi usado para implementar os algoritmos para técnicas de redução de dimensionalidade. Notou-se que a redução dimensional melhorou a precisão da previsão dos modelos de aprendizado da máquina, mas a RNA produziu melhores resultados do que o SVM.

Kalakh, Kandil e Hakem (2012) desenvolveram um modelo de rede neural para PL de canal de Banda Ultra Larga (UWB) em um ambiente de mineração. Os autores tiveram como objetivo para este modelo o foco nas alterações na atenuação da PL em relação ao tempo e à distância. O modelo de RNA foi construído usando a arquitetura MLP enquanto empregava o algoritmo de BP para o treinamento. Os resultados obtidos no treinamento e teste da rede neural foram comparados com valores medidos experimentais. O modelo de rede neural previu corretamente as mudanças na atenuação da PL, ao mesmo tempo em que forneceu precisão suficiente com as medidas experimentais. O trabalho revelou a importância e a eficiência das redes neurais para previsão de PL em um ambiente hostil (neste caso, em minerações).

Zaarour et al. (2012) apresentaram um estudo experimental para modelagem de perda de trajetória em canal UWB, em ambiente de mineração, através da implementação da Função de Base Radial (RBF) e MLP, com foco nas variações na atenuação da PL em relação à distância

e frequência. Essa foi considerada uma abordagem diferente na modelagem de PL em um ambiente de mineração. Os resultados obtidos mostraram a eficiência do RBF na modelagem de PL, pois proporcionou menor taxa de erro e maior precisão na previsão de PL. Os autores concluíram que o trabalho de pesquisa auxiliaria na determinação do melhor tipo de RNA para modelagem do canal em um ambiente de mina.

Sotiroudis et al. (2013) forneceram uma abordagem alternativa para a previsão de PL em áreas urbanas aplicando um algoritmo de ED chamado de Evolução Diferencial Composta (CoDE) no desenvolvimento de uma RNA ótima. Os autores propuseram uma metodologia para alcançar a RNA ótima. Os resultados de CoDE foram comparados com os obtidos de várias estratégias de DE. Concluiu-se que o CoDE era altamente eficaz com as vantagens de ser fácil de implementar e ser modificado. Além disso, duas arquiteturas de RNA: duas e três camadas ocultas foram avaliadas usando CoDE, e o desempenho comprovou sua eficácia e precisão em produzir os resultados esperados. Os resultados obtidos foram comparados com os resultados de outros modelos de traçado de raios, e o método proposto apresentou melhor precisão. Outros trabalhos foram recomendados para melhorar as habilidades de generalização e aproximação da RNA utilizando conjuntos de dados para ambientes construídos não uniformes.

Zineb e Ayadi (2015) desenvolveram um novo modelo de propagação usando RNA. Redes neurais e técnicas de mineração de dados foram combinadas para desenvolver um novo modelo. O modelo proposto, derivado do modelo *multiwall*, destinava-se às bandas de frequência mais populares, incluindo UMTS, GSM e Wi-Fi. A rede neural foi modelada usando a estrutura MLP e treinada com dados medidos (incluindo frequência, atenuação do solo, distância e frequência transmissor/receptor) utilizando o algoritmo de aprendizado BP. Os resultados derivados do modelo multifrequência *multiwall* baseado em RNA mostraram melhor desempenho e maior precisão em comparação com um modelo *multiwall* calibrado. Os autores sugeriram que os resultados obtidos poderiam ser melhorados com a inclusão de outros parâmetros, não contidos no modelo, por exemplo: perda de difração, sombra do corpo e assim por diante.

Benmus, Abboud e Shatter (2015) propuseram a previsão de PL nas bandas de 900, 1800 e 2100 MHz por meio da abordagem de RNA. Eles coletaram medições de cinco áreas diferentes: urbanas, urbanas densas, suburbanas, suburbanas densas e rurais, nas faixas de frequência especificadas, que foram utilizadas para o treinamento e avaliação do modelo. Os resultados derivados do modelo foram comparados com o modelo Hata, avaliando o desempenho com base no Erro Quadrático Médio (EQM). O EQM para o modelo proposto foi inferior ao do modelo Hata, mantendo um alto nível de precisão.

Angeles e Dadios (2015) propuseram um modelo alternativo de RNA para a previsão de PL em macrocélulas de TV digital. O modelo foi projetado usando a simulação Longley-Rice, com o auxílio de uma arquitetura de rede neural *feedforward* e treinado usando um algoritmo BP. Os resultados derivados do modelo de predição foram de melhor desempenho em comparação com outros modelos de propagação, como o modelo de propagação no espaço livre e o modelo de Egli. O modelo desenvolvido foi capaz de se adaptar a ambientes arbitrários. Os efeitos

de múltiplas localizações de transmissores e o posicionamento dos receptores acima de um transmissor podem ser investigados em trabalhos futuros.

Bhuvaneshwari, Hemalatha e Satyasavithri (2016) avaliaram três redes neurais dinâmicas: *Focused Time Delay Neural Network* (FTDNN), *Distributed Time Delay Neural Network* (DTDNN) e *Layer Recurrent Neural Network* (LRNN) para a previsão de PL em uma tentativa de minimizar erros entre as medições e simulações. Cada uma das redes neurais dinâmicas foi treinada com algoritmos de treinamento LM e SCG. A avaliação do desempenho foi analisada em relação à correlação entre as medidas e simulações, o valor de erro médio (EM) e o valor de desvio padrão (σ). Os resultados mostraram que o LRNN proporcionou o melhor desempenho com o aumento do tempo de computação, enquanto o FTDNN teve um desempenho melhor que o DTDNN.

Ferreira, Matos e Silva (2016) utilizaram RNA com o propósito de melhoria do algoritmo do cálculo de PL para banda de *Ultra High Frequency* (UHF). As medições foram realizadas em 1140 MHz em ambiente urbano, enquanto a intensidade de sinal recebida foi prevista usando o método Delta-Bullington e Cascade Knife-Edge, ambos disponíveis no ITU-R. Entradas como distância transmissor/receptor, perda de difração e previsão da intensidade do sinal foram usadas na entrada na RNA, que mostrou grande eficácia na previsão de intensidade de sinal recebida. Os resultados obtidos no estudo revelaram que a RNA pode ser utilizada como ferramenta complementar para modelos de previsão de ITU.

No estudo realizado por Garah et al. (2016), dados coletados em Batna, Argélia, são usados para calcular a PL para GSM (908-957 MHz). A PL medida é comparada com a PL teórica estimada pelos modelos empíricos mais amplamente utilizados, como Cost 231, Hata, SUI e Egli. O melhor modelo para estimar a PL medida é otimizado usando AGs para prever a PL em áreas suburbanas e rurais. O Erro RMS (E_{RMS}) e outros critérios de teste entre os dados reais e previstos são calculados para vários modelos de perda de caminho. Verificou-se que o modelo ajustado Cost 231 supera os outros modelos estudados. Segundo os autores, os resultados podem ajudar os engenheiros de telecomunicações a melhorarem seu planejamento e design de sistemas microcelulares.

Ayadi, Zineb e Tabbane (2017) desenvolveram um modelo de PL na faixa UHF baseado em redes neurais para redes heterogêneas. As medições realizadas foram divididas em duas: uma para treinamento do modelo e a outra para validação do modelo. O modelo baseado em rede neural fez uso do algoritmo BP, enquanto obtém entradas do Modelo de Propagação Padrão (SPM) e uma adição de vários parâmetros, como perda de difração, tipo de ambiente, frequência e uso do solo.

Liu et al. (2017) propuseram um método de Rede Neural Avançada de Camada Oculta Única (SHLFNN) para a modelagem de canais de desvanecimento (predição de PL), incluindo variação em pequena escala e atenuação em grande escala. Para variação em pequena escala, dois algoritmos (*Extreme Learning Machine* (ELM) e LM) foram empregados no treinamento da rede neural e os resultados mostraram que o ELM foi tão eficaz quanto o algoritmo BP. Para

atenuação em larga escala, o estimador ELM necessário para o expoente de previsão de PL foi estabelecido e produziu resultados com maior velocidade quando comparado ao algoritmo BP. Esses resultados comprovaram a eficácia do algoritmo ELM no modelo de rede neural.

Eichie et al. (2017a) estabeleceram um modelo baseado em redes neurais para a determinação do nível de intensidade de sinal recebido GSM com o uso de fatores atmosféricos (por exemplo, umidade relativa, temperatura atmosférica, ponto de orvalho). Esses fatores atmosféricos foram usados como entradas para o modelo RNA e treinados com o auxílio do algoritmo LM. O desenvolvimento do modelo foi uma abordagem de três etapas. Primeiramente, foram adquiridos os dados necessários; em seguida, os dados coletados foram pré-processados; finalmente, o modelo de RNA proposto foi projetado e implementado usando os valores de peso e *bias* para formar a equação do modelo. Os resultados obtidos mostraram que o modelo desenvolvido teve um desempenho ótimo com baixo EQM e valores de precisão aceitáveis.

No artigo de Cavalcanti, d'Assunção e Mendonça (2017) são apresentados os resultados da aplicação de um GA para prever a PL em áreas urbanas para redes Long Term Evolution (LTE) e Long Term Evolution Advanced (LTE-A) em 879 MHz usando modelos de propagação empíricos. Foi realizado um teste comparativo, onde os modelos de Espaço Livre simulado e Ericsson 9999, juntamente com suas versões otimizadas, foram comparados com dados experimentais obtidos de uma campanha de medição realizada na cidade de Natal. Os resultados mostraram uma aproximação mais próxima dos modelos modificados com GA em relação às medições. Espera-se que isso permita uma estimativa do sinal mais próxima da realidade, evitando erros significativos no planejamento e implementações de redes LTE e LTE-A para este tipo de área e frequência.

Gómez-Pérez et al. (2017) recomendaram o uso de RNA para modelagem de atenuação por vegetação em bandas de frequência celular de 2G a 4G. A arquitetura MLP foi usada e treinada com medições de várias configurações da quantidade de vegetação (como espessura das árvores, densidade de vegetação, folhagem, polarização, frequência (900, 1800 e 2100MHz), densidade de tronco de espécie de vegetação e posição do receptor ao longo de um trilho linear), bem como espécies de vegetação. Após treinamento e teste, foram realizadas comparações com a atenuação deduzida pela RNA e os dados experimentais. Além disso, comparações foram feitas com a PL calculada através da Recomendação ITU P.1546. Como resultado dessas comparações, foi mostrado que o modelo foi altamente eficiente em prever atenuações precisas nas bandas de frequência especificadas, mantendo um erro mediano. Foi sugerido que o modelo seria altamente benéfico para projetistas de enlaces de rádio na previsão da atenuação em função do tipo de vegetação. No entanto, as bandas de frequência utilizadas nesse modelo podem ser estendidas para frequências mais altas (como bandas de frequência Wi-Fi, Wi-Max ou 5G), desde que seja realizado treinamento adequado.

Popoola et al. (2018) estabeleceram um modelo ótimo para a previsão de PL utilizando redes neurais *feedforward*. Os dados de entrada foram obtidos percorrendo o trajeto com um

equipamento de medição embarcado em um veículo. SHLFNN foi treinado com os dados derivados (incluindo elevação, altura do agrupamento, longitude, altitude, latitude) usando o algoritmo LM. Em seguida, avaliou-se o desempenho do modelo de RNA. Os resultados derivados do modelo foram comparados com modelos empíricos populares (Cost 231, Egli, Okumura- Hata e ECC-33) para revelar o desempenho ideal do modelo RNA na melhoria da precisão e capacidade de generalização. No futuro, mais dados podem ser obtidos de vários ambientes de propagação e outras técnicas de aprendizado de máquina também podem ser exploradas.

No trabalho de Nafea e Hamza (2020) foi otimizado o modelo Cost 231 com a técnica AG utilizando a ferramenta computacional Matlab. O estudo foi realizado na faixa de frequência da tecnologia WIMAX em torno de 3,5 GHz e com distâncias entre 3 a 7 km entre a estação transmissora e o receptor. Os resultados mostraram que a otimização apresenta melhor performance do que o algoritmo convencional.

Santos (2021) realizou um trabalho de pesquisa a partir da coleta de amostras de campo elétrico em várias regiões da cidade de Uberlândia. Um estudo minucioso de cada região foi feito por meio das amostras aferidas, utilizando análises estatísticas e comparações das amostras com os modelos de propagação espaço livre, Okumura-Hata, ITU-R P.1546 e RNA. Sua RNA é uma rede de uma camada com 100 neurônios na camada escondida, foram utilizados como parâmetros de entrada a distância e a altitude, e como saída da RNA, o campo elétrico.

2.1 CONSIDERAÇÕES FINAIS

A revisão de literatura apresentada evidencia a ampla utilização de RNAs na previsão de PL em diferentes contextos e ambientes. Ao longo das últimas décadas, estudos têm explorado diversas configurações e técnicas para aprimorar a precisão e a generalização dos modelos, demonstrando a flexibilidade e eficiência das RNAs em relação aos modelos analíticos tradicionais.

Modelos baseados em RNAs mostraram-se superiores na previsão de PL em áreas urbanas, suburbanas, rurais e até mesmo em ambientes desafiadores, como minas e vegetações densas. Pesquisas como as de Popescu et al. (2001) e Ostlin, Zepernick e Suzuki (2010) destacaram a relevância de incluir características ambientais detalhadas como entradas para as RNAs, enquanto estudos como Piacentini e Rinaldi (2011) e Gómez-Pérez et al. (2017) exploraram a combinação de técnicas de aprendizado de máquina com redução dimensional e medição experimental para melhorar o desempenho.

Os avanços no uso de RNAs foram complementados por novas abordagens, como o emprego de AGs por Garah et al. (2016) e técnicas de Evolução Diferencial por Sotiroudis et al. (2013), que contribuíram para otimizar as arquiteturas de RNAs e aumentar sua precisão. Além disso, a inclusão de dados atmosféricos por Eichie et al. (2017a) e a análise de diferentes bandas de frequência, como WiMAX, Wi-Fi e 5G, mostraram a aplicabilidade das RNAs em cenários

modernos e desafiadores.

Embora os modelos baseados em RNAs tenham alcançado resultados promissores, algumas limitações foram identificadas, como a dependência de dados topográficos detalhados por Ostlin, Zepernick e Suzuki (2010) e a necessidade de conjuntos de dados mais abrangentes para melhorar a generalização por Popoola et al. (2018). Trabalhos futuros podem se beneficiar de metodologias híbridas que combinem RNAs com outras técnicas de aprendizado de máquina, bem como da exploração de novas arquiteturas de rede, como as Redes Neurais Dinâmicas por Bhuvaneshwari, Hemalatha e Satyasavithri (2016), para enfrentar desafios específicos.

MODELOS DE PROPAGAÇÃO

Resumo do capítulo

Neste capítulo, são apresentados os modelos teóricos de propagação de sinais utilizados neste trabalho, como Perda no Espaço Livre, o Modelo Okumura-Hata, Cost 231 Hata Model, Egli, ECC-33, Recomendação ITU P-1546.

A propagação de onda eletromagnética nem sempre ocorre diretamente entre um transmissor e um receptor, uma vez que a antena transmissora nem sempre pode ser instalada em um ponto alto, como por exemplo, uma torre, um pico ou uma montanha, assim como a antena receptora pode estar pouco acima do nível do solo, como, por exemplo, em aparelhos celulares, receptores de televisão em ambientes rurais, suburbanos e urbanos. A propagação real de uma onda eletromagnética, nesses ambientes, depende da frequência, polarização, geometria das construções, estrutura dos materiais, orientação, altura (SEYBOLD, 2005).

Tratando-se de modelos de propagação, este é um assunto vasto com inúmeros artigos e modelos disponíveis. O objetivo de um modelo de propagação é determinar a PL mediana em função da distância. Neste capítulo, são apresentados os mecanismos de perda no espaço livre juntamente com os modelos existentes: Perda no Espaço Livre, Hata, Egli, ECC-33, Cost 231 e Recomendação ITU-P.1546, que foram utilizados neste estudo.

3.1 MECANISMOS DE PERDA NO ESPAÇO LIVRE

Os três principais mecanismos de perda no espaço livre que impactam na propagação das ondas em um sistema de comunicação são: reflexão, difração e espalhamento, ilustrados na Figura 3.1.

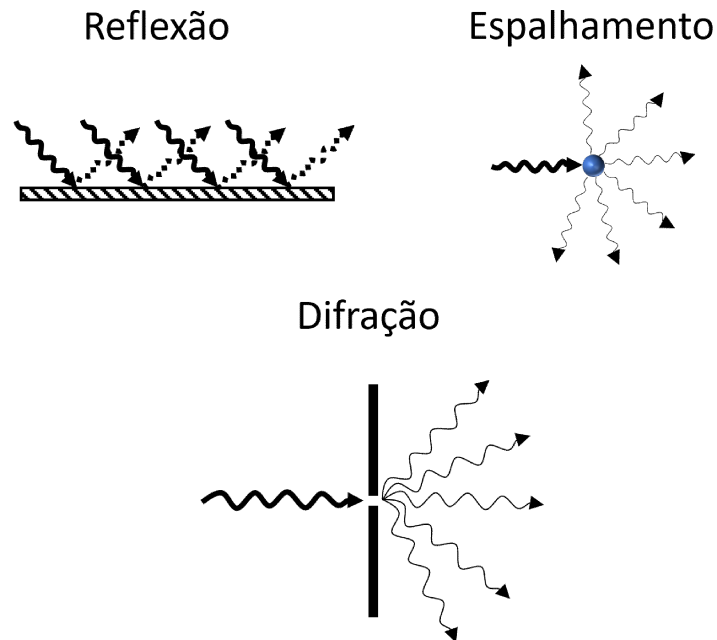
A reflexão ocorre quando uma onda eletromagnética em propagação colide com um objeto que tem dimensões muito grandes quando comparado ao comprimento de onda. As reflexões geralmente ocorrem da superfície da terra, vegetação e de edifícios (RAPPAPORT, 2002).

A difração ocorre quando o caminho, percorrido pela onda eletromagnética entre o transmissor e o receptor, está obstruído por uma superfície que apresenta irregularidades acentuadas. As ondas secundárias resultantes da superfície obstrutiva estão presentes em todo o espaço e até atrás do obstáculo, dando origem a curvatura das ondas ao redor do obstáculo, mesmo quando não existe um caminho de linha de visão entre o transmissor e o receptor. Em altas frequências, a difração, como a reflexão, depende da geometria do objeto, bem como da amplitude, fase e polarização da onda incidente no ponto de difração (RAPPAPORT, 2002).

O espalhamento ocorre quando a onda passa através de objetos cujas dimensões são pequenas em comparação ao comprimento de onda incidente e onde o número de obstáculos por volume unitário é grande. Essa situação geralmente ocorre quando o meio é constituído por pequenos objetos ou com superfícies ásperas ou rugosas, como folhagens, placas de trânsito,

fiação de poste (RAPPAPORT, 2002).

Figura 3.1 – Mecanismos de Propagação e Perdas



Fonte: autor

3.2 PERDA NO ESPAÇO LIVRE

No espaço livre, uma onda se propaga sem encontrar nenhum obstáculo. A superfície de uma frente de onda é o conjunto de todos os pontos alcançados em um determinado momento após o momento de emissão da onda dentro de um meio homogêneo. A atenuação no espaço livre resulta do espalhamento de energia que ocorre à medida que a onda se propaga para longe do transmissor (SIZUN, 2005). A atenuação do espaço livre ocorre em função da distância e frequência conforme a Equação 3.1.

$$L = 20\log(d) + 20\log(f) + 32.45 \quad (3.1)$$

Em que:

- L a perda calculada (dB);
- d a distância em quilômetros (Km);
- f a frequência em Megahertz (MHz).

3.3 MODELO OKUMURA-HATA

3.3.1 MODELO OKUMURA

O modelo de Okumura é um dos modelos mais utilizados para previsão de sinal em áreas urbanas. Esse modelo é aplicável para frequências na faixa de 150 MHz a 1920 MHz (embora seja normalmente extrapolado até 3000 MHz) e distâncias de 1 km a 100 km. Pode ser usado para alturas de antena da estação base que variam de 30 m a 1000 m. Okumura desenvolveu um conjunto de curvas que dão a atenuação mediana relativa ao espaço livre (A_{mu}), em uma área urbana sobre um terreno quase plano com uma altura efetiva da antena da estação base (h_{te}) de 200 metros (m) e uma altura da antena móvel (h_{re}) de 3 m. Essas curvas foram desenvolvidas a partir de extensas medições usando antenas verticais omnidirecionais tanto na base quanto na móvel, e são plotadas em função da frequência na faixa de 100 MHz a 1920 MHz e em função da distância da estação base na faixa 1 km a 100 km. Para determinar a PL usando o modelo de Okumura, a PL do espaço livre entre os pontos de interesse é determinada primeiro e, em seguida, o valor de $A_{mu}(f, d)$ (como lido nas curvas) é adicionado a ele junto com os fatores de correção a serem considerados para o tipo de terreno (RAPPAPORT, 2002). A perda pode ser calculada através da Equação 3.2:

$$L_{50}(dB) = L_f + A_{mu}(f, d) - G(h_{te}) - G(h_{re}) - G_{AREA} \quad (3.2)$$

Em que L_{50} é o 50º percentil, ou seja, mediana da PL, L_f é a PL no espaço livre, A_{mu} é a atenuação relativa mediana do espaço livre, $G(h_{te})$ é o ganho da antena transmissora, $G(h_{re})$ é o ganho da antena receptora e $G_{(Area)}$ é o ganho devido a o tipo de ambiente. Note que os ganhos das antenas são essencialmente em função da altura e não tem relação com ao padrão da antena (RAPPAPORT, 2002).

As curvas de A_{mu} e $G_{(Area)}$ são mostradas nas Figuras 3.2 e 3.3. Além disso, Okumura encontrou que $G(h_{te})$ varia em uma taxa de 20 dB/década (Eq 3.3) e $G(h_{re})$ varia em uma taxa de 10 dB/década para alturas menores que 3 metros (Eqs 3.4 e 3.5).

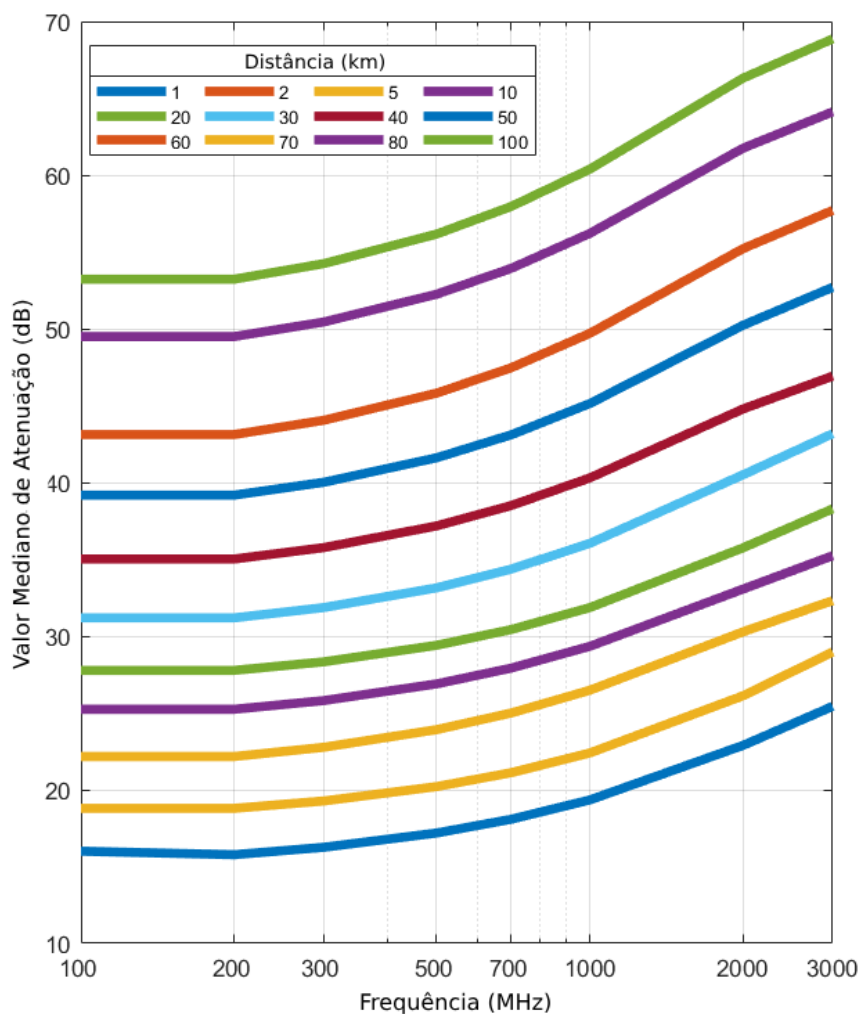
$$G(h_{te}) = 20 \log \frac{(h_{te})}{200} \text{ para } 1000m > h_{te} > 30m \quad (3.3)$$

$$G(h_{re}) = 10 \log \frac{(h_{re})}{3} \text{ para } h_{re} \leq 3m \quad (3.4)$$

$$G(h_{re}) = 20 \log \frac{(h_{re})}{3} \text{ para } 10m > h_{re} > 3m \quad (3.5)$$

Outras correções também podem ser aplicadas ao modelo de Okumura. Alguns dos parâmetros importantes relacionados ao terreno são a altura de ondulação do terreno (Δh), a altura da crista isolada, a inclinação média do terreno e o parâmetro misto terra-mar. Uma vez que os parâmetros relacionados ao terreno são calculados, os fatores de correção necessários podem ser adicionados ou subtraídos conforme necessário. Todos esses fatores de correção também estão disponíveis como curvas de Okumura.

Figura 3.2 – Atenuação média em relação ao espaço livre sobre um terreno quase-liso

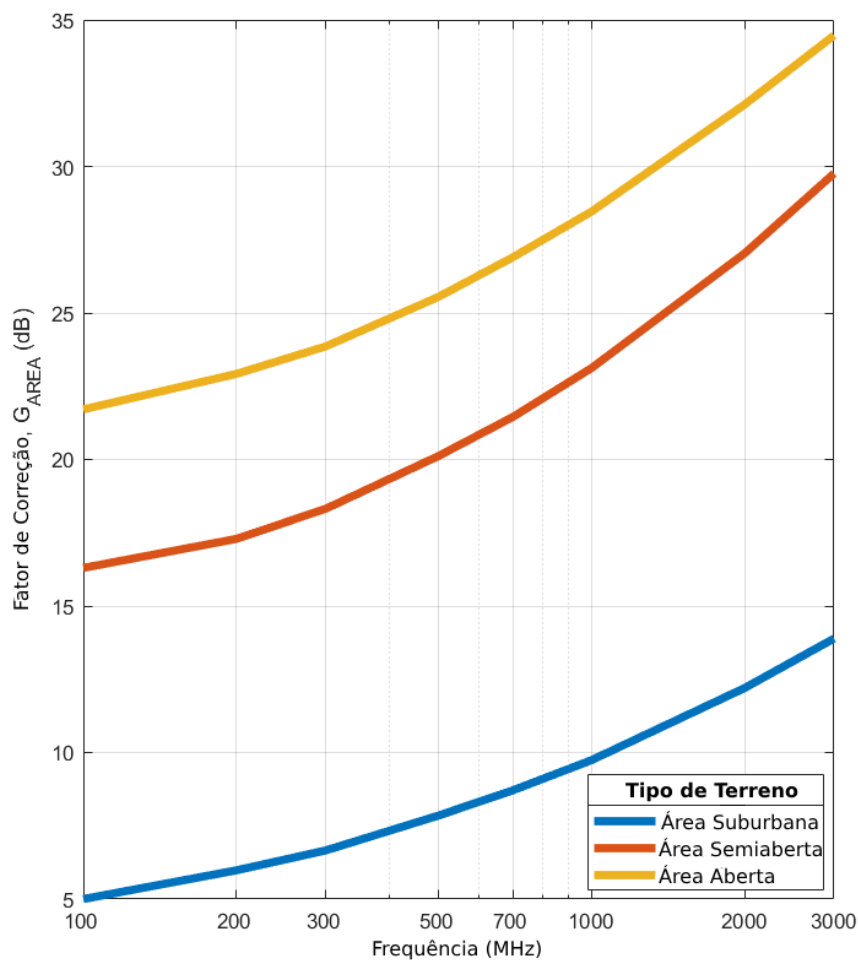


Fonte: Autor adaptado de Rappaport (2002)

O modelo de Okumura é totalmente baseado em dados medidos e não fornece nenhuma explicação analítica. Para muitas situações, extrapolações das curvas derivadas podem ser feitas para obter valores fora da faixa de medição, embora a validade de tais extrapolações dependam das circunstâncias e da suavidade da curva em questão.

Esse modelo é considerado o mais simples e o melhor em termos de precisão na previsão de PL para sistemas de rádio celular móvel terrestre em ambientes não regulares. É muito prático e tornou-se um padrão para o planejamento de sistemas em modernos sistemas de rádio móveis terrestres no Japão. A principal desvantagem do modelo é a variação (imprecisão) do PL estimado em função de variações no terreno, portanto o modelo é considerado bom em áreas urbanas e suburbanas, mas não tão bom em áreas rurais. Os desvios padrão comuns entre os valores de PL previstos e medidos são em torno de 10 dB a 14 dB.

Figura 3.3 – Fator de Correção G_{AREA} para diferentes tipos de terrenos



Fonte: Autor adaptado Rappaport (2002)

3.3.2 MODELO HATA

O modelo Hata é uma formulação empírica dos dados gráficos de PL fornecidos por Okumura, e é válido de 150 MHz a 1500 MHz (RAPPAPORT, 2002). Com intuito de tornar o modelo de Okumura mais fácil para implementação computacional, o modelo de Hata

foi desenvolvido a partir do modelo de Okumura, tendo ajustadas as curvas de Okumura com expressões analíticas. São essas expressões analíticas que tornam a implementação computacional do modelo de okumura simples. (MOLLEL; KISANGIRI, 2014). Na Equação 3.6 é apresentada a formula padrão para PL mediano em áreas urbanas (RAPPAPORT, 2002).

$$L_{50}(dB) = 69,55 + 26,16\log(f_c) - 13,82\log(h_{te}) - a(h_{re}) + (44,9 - 6,55\log(h_{te})\log(d)) \quad (3.6)$$

Em que:

- f é a frequência em MHz de 150 MHz a 1500 MHz,
- h_{te} é a altura efetiva da antena do transmissor variando de 30 metros a 200 metros,
- h_{re} é a altura da antena do receptor em metros variando de 1 metro a 10 metros,
- d é a distância de separação entre o transmissor e o receptor (km),
- $a(h_{re})$ é o fator de correção para a altura da antena móvel em função do tamanho da área de cobertura. Para cidades pequenas e médias esse fator é dado pela Equação 3.6.

$$a(h_{re}) = (1,1\log(f_c) - 0,7)h_{re} - (1,56\log(f_c) - 0,8)dB \quad (3.7)$$

e para cidade grandes é dado pelas Equações 3.8 e 3.9

$$a(h_{re}) = 8,29(\log(1,54h_{re}))^2 - 1,1dB \text{ para } f_c \leq 300MHz \quad (3.8)$$

$$a(h_{re}) = 3,2(\log(11,75h_{re}))^2 - 4,97dB \text{ para } f_c \geq 300MHz \quad (3.9)$$

Para obter a PL em uma área suburbana a formula Hata padrão é modificada pela Equação 3.10.

$$L_{50}(dB) = L_{50}(urbano) - 2[\log(f_c/28)]^2 - 5,4 \quad (3.10)$$

Embora o modelo Hata não tenha nenhuma das correções específicas de caminho como no modelo de Okumura, as equações acima têm um valor prático significativo. As previsões do

modelo Hata se comparam muito de perto com o modelo original de Okumura, desde que d seja maior que 1 km. Esse modelo é adequado para sistemas móveis de células grandes, mas não para sistemas de comunicações pessoais que possuem células na ordem de 1 km de raio.

3.4 COST 231 HATA MODEL

Um modelo amplamente utilizado para prever a PL em sistemas móveis sem fio é o modelo Cost 231 Hata. Foi concebido como uma extensão do modelo Okumura-Hata. O modelo Cost 231 Hata foi projetado para ser usado na faixa de frequência de 500 MHz a 2000 MHz. Ele também contém correções para ambientes urbanos, suburbanos e rurais. Embora sua faixa de frequência esteja fora das medições, sua simplicidade e disponibilidade de fatores de correção o tornaram amplamente utilizado para previsão de PL nessa faixa de frequência (ABHAYAWARDHANA et al., 2005). A Equação representa a PL é ,

$$L = 46,3 + 33,9\log(f) - 13,82\log(h_t) + ah_m - 44,9 - 6,55\log(h_t)\log(d) + c \quad (3.11)$$

O parâmetro c é definido como 0 dB para ambientes suburbanos ou abertos e 3 dB para ambientes urbanos. O parametro ah_m definido para ambientes urbanos é dado por:

$$a(h_r) = 3.2(\log(11.75h_r))^2 - 4.94 \text{ para } f > 400MHz \quad (3.12)$$

e para ambientes rurais(abertos) como:

$$a(h_r) = (1.1\log(f) - 0,7)h_r - (1,56\log(f) - 0,8) \quad (3.13)$$

3.5 MODELO EGLI

Para entender as características da propagação das ondas eletromagnéticas sobre um terreno irregular, John Egli desenvolveu gráficos e curvas de correção baseados em análises estatísticas de uma ampla gama de medidas, a partir de dados do mundo real em transmissões de televisão UHF e VHF, realizadas entre as frequências de 40 a 1000 MHz, em diferentes regiões dos Estados Unidos (JAWHLY; TIWARI, 2020). Egli observou uma tendência de que, em áreas pequenas, a intensidade do sinal diminua proporcionalmente ao inverso da quarta potência da distância em relação ao transmissor. Esse modelo é baseado em propagação em terra plana. A Equação 3.14 representa a formula para a PL para Egli :

$$L = \begin{cases} 20\log(f) + p_o + 76,3 & \text{se } h_r < 10 \\ 20\log(f) + p_o + 85,9 & \text{se } h_r > 10 \end{cases} \quad (3.14)$$

$$p_o = 40\log(d) - 20\log(h_t) - 10\log(h_r)$$

- h_t é a altura da antena transmissora (m);
- h_r é a altura da antena receptora (m);
- d é a distância entre o transmissor e o receptor (m);
- f é a frequência de transmissão (MHz);

3.6 ECC-33

Os dados experimentais originais de Okumura foram coletados nos subúrbios de Tóquio. Os autores referem-se a áreas urbanas subdivididas nas categorias "cidade grande" e "cidade média". Eles também fornecem fatores de correção para áreas "suburbanas" e "abertas". Uma vez que, as características de uma área altamente urbanizada como Tóquio são bastante diferentes daquelas encontradas em áreas suburbanas típicas da Europa, o uso do modelo de "cidade média" é recomendado para cidades europeias. Embora o modelo Okumura-Hata seja amplamente utilizado para bandas de UHF, sua precisão é questionável para frequências mais altas. O modelo Cost 231 estendeu seu uso até 2 GHz, mas foi proposto para sistemas móveis com antenas omnidirecionais localizadas até 3 metros acima do nível do solo. Uma abordagem diferente foi feita nesse modelo, que extrapolou as medidas originais de Okumura e modificou suas suposições para que represente mais de próximo de um sistema fixo de acesso sem fio (FWA) (ABHAYAWARDHANA et al., 2005). As Equações representam a PL prevista para este modelo.

$$L = A_{fs} + A_{bm} - G_b - G_r \quad (3.15)$$

Em que:

- A_{fs} é a atenuação do espaço livre ;
- A_{bm} PL mediano;
- G_b Ganho da antena transmissora;
- G_r Ganho da antena receptora.

$$A_{fs} = 92,4 + 20\log(d) + 20\log(f) \quad (3.16)$$

$$A_{bm} = 20,41 + 9,83\log(d) + 7,89\log(f) + 9,56(\log(f))^2 \quad (3.17)$$

$$G_b = \log\left(\frac{h_t}{200}\right) (13,958 + 5,8\log(d))^2 \quad (3.18)$$

$$A_{fs} = 92,4 + 20\log(d) + 20\log(f) \quad (3.19)$$

Para cidades de médio porte:

$$G_r = (42,57 + 13,7\log(f)(\log(h_r) - 0,585)) \quad (3.20)$$

E para cidades de grande porte:

$$G_r = 0,759h_r - 1,862 \quad (3.21)$$

Sendo f em (GHz).

O modelo de cidade média é mais apropriado para cidades europeias, enquanto o ambiente de cidade grande deve ser usado apenas para cidades com edifícios altos.

3.7 RECOMENDAÇÃO ITU P-1546

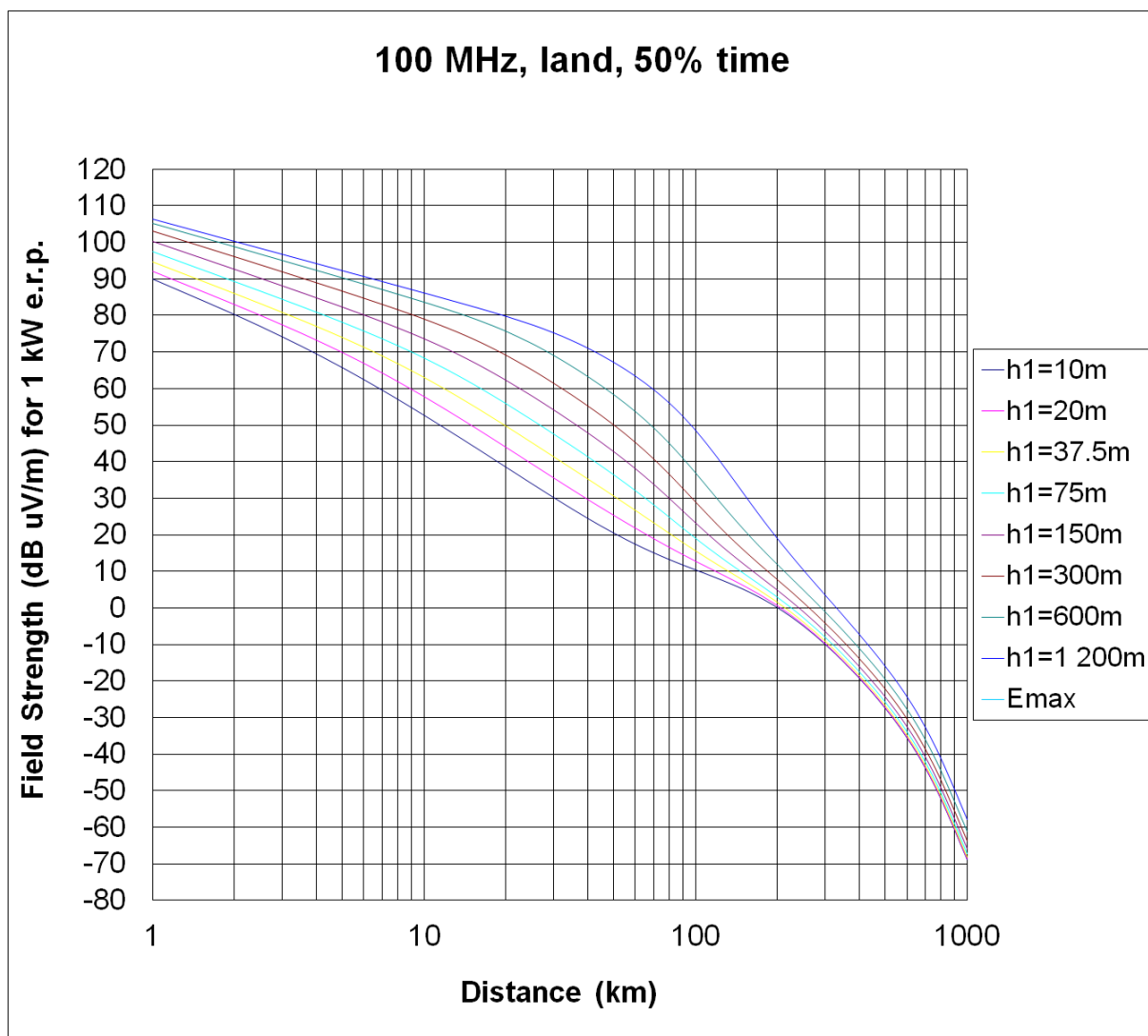
A Recomendação ITU P.1546 é um modelo desenvolvido pela União Internacional das Telecomunicações (ITU) e descreve um método para predições de propagação de rádio ponto a ponto para serviços terrestres na faixa de frequência de 30 MHz a 4.000 MHz. E tem como objetivo a utilização em enlaces de rádio troposféricos em ambientes terrestres, marítimos e/ou misto (terrestre/marítimo) até 1000 km de distância e para antenas de transmissão com alturas inferiores a 3000 m. O método é baseado na interpolação/extrapolação de curvas de intensidade de campo derivadas empiricamente em função da distância, altura da antena, frequência e tempo percentual. O procedimento de cálculo também inclui correções nos resultados obtidos a partir dessa interpolação/extrapolação para poder ser utilizado em terrenos abertos ou terminais perto de aglomerações (ITU, 2019).

Neste método, campos elétricos foram medidos nas frequências nominais de 100, 600 e

2000 MHz com uma potência no valor de 1 kW; alguns campos foram medidos em percursos terrestres em climas temperados na Europa e na América do Norte, e outros em percursos marítimos, no Mar do Norte e no Mediterrâneo. A partir dessas medições, foram traçadas curvas como ilustradas na Figura 3.4. Além disso, é possível a utilização deste método para outras frequências através de interpolação ou extrapolação dessas curvas (ITU, 2019).

Para cálculo de intensidade de potência ou campo elétrico feito computacionalmente, o ITU, através Grupo de Estudo em Radiocomunicação 3 (SG3), disponibiliza em sua *website* tabelas com os valores medidos nas curvas, além de uma função para ser utilizada no software Matlab (ITU, 2019).

Figura 3.4 – Curvas de intensidade de campo para frequência de 100 MHz e propagação em terra 50% do tempo



Fonte: ITU (2019)

3.8 CONSIDERAÇÕES FINAIS

Neste capítulo, foram apresentados diversos modelos de propagação amplamente utilizados em estudos de comunicação sem fio. Inicialmente, na Seção 3.1, foram abordados os mecanismos fundamentais de perda no espaço livre, seguidos pela análise detalhada da perda no espaço livre na Seção 3.2. Na sequência, a Seção 3.3 explorou o modelo Okumura-Hata, desdobrado em suas versões clássicas: o modelo de Okumura (Seção 3.3.1) e o modelo de Hata (Seção 3.3.2). Complementarmente, a Seção 3.4 discutiu o modelo Cost 231-Hata, enquanto a Seção 3.5 apresentou o modelo de Egli. A Seção 3.6 detalhou o modelo ECC-33, e a Seção 3.7 abordou a recomendação ITU P-1546. Por fim, na Seção 3.8, foram destacadas as conclusões gerais sobre as características e aplicabilidades desses modelos, oferecendo uma visão abrangente sobre suas vantagens e limitações em diferentes cenários de propagação.

REDES NEURAIS ARTIFICIAIS

Resumo do capítulo

O capítulo sobre RNAs explora os fundamentos dessa técnica, desde modelos de neurônios até arquiteturas de rede e estágios de treinamento. São discutidos os diferentes tipos de funções de ativação e o algoritmo de treinamento Modificação Levenberg-Marquardt.

Segundo Haykin (2009), uma RNA é utilizada como simulador do funcionamento do cérebro, ela executa tarefas em particular ou uma função de interesse. Geralmente a rede é implementada usando componentes eletrônicos ou softwares de simulação. Para alcançar um bom desempenho, a rede emprega uma interconexão massiva de células computacionais chamadas de "neurônios" ou "unidades de processamento". O sinal passa entre os neurônios através da interconexão. Cada interconexão possui um peso associado, que nas redes em geral, multiplica o sinal transmitido, esse peso representa a informação a ser usada pela rede para resolver um problema. Por fim, cada neurônio aplica uma função de ativação (geralmente não linear) para sua entrada de rede (soma dos sinais de entrada multiplicados pelos pesos) para determinar o sinal de saída (FAUSETT, 1994).

Uma rede neural é caracterizada pelo seu padrão de interconexão entre os neurônios (arquitetura), seu método para determinação dos pesos nas interconexões (treinamento, aprendizado ou algoritmo) e sua função de ativação (FAUSETT, 1994).

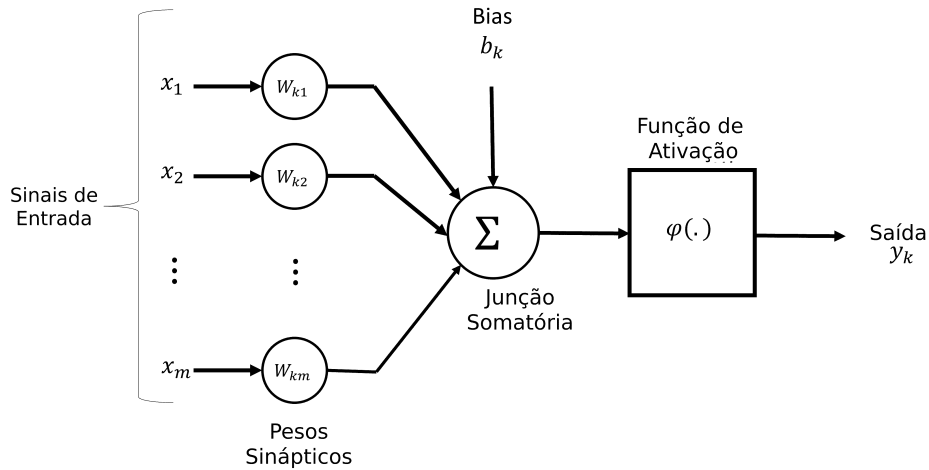
4.1 MODELOS DE NEURÔNIOS

Um neurônio é uma unidade de processamento de informação que é fundamental para a operação de uma rede neural. O diagrama de blocos de um modelo de um neurônio é ilustrado na Figura 4.1 e forma a base para o projeto de uma rede neural (artificial). Os três elementos básicos de um modelo neuronal são:

- Um conjunto de *synapses* ou interconecções, cada uma dessas é caracterizada pelo seu peso. Especificamente, um sinal x_j em uma *synapse* j conectada a um neurônio k é multiplicado pelo peso sináptico w_{kj} , em que o primeiro subscrito se refere ao neurônio em questão, e o segundo subscrito se refere à entrada da sinapse, a qual o peso se refere. Ao contrário do cérebro humano, os pesos sinápticos de um neurônio artificial podem possuir valores negativos ou positivos (HAYKIN, 2009).
- Um adicionador para somar os valores de entradas, ponderado pelas respectivas sinapses do neurônio, essa operação constitui em um combinador linear (HAYKIN, 2009).
- Uma função de ativação para limitar a amplitude da saída de um neurônio, o que faz com

que os valores de saída sejam finitos. Comumente, limitado no intervalo de $[0,1]$ ou de $[-1,1]$ (HAYKIN, 2009).

Figura 4.1 – Modelo de Neurônio não linear



Fonte: Autor adaptado de Haykin (2009)

No modelo neuronal da Figura 4.1 existe um fator externo que é aplicado, chamado de *bias*, denotado por b_k . O *bias* b_k tem o efeito de aumentar ou diminuir a entrada da função de ativação, dependendo se isto for positivo ou negativo.

Em termos matemáticos, pode-se descrever o neurônio k através do par de Equações (4.1) e (4.2)

$$u_k = \sum_{j=1}^m w_{kj} x_j \quad (4.1)$$

e

$$y_k = \varphi(u_k + b_k) \quad (4.2)$$

Em que:

- x_1, x_2, \dots, x_m são os sinais de entrada;
- $w_{k1}, w_{k2}, \dots, w_{km}$ são os pesos sinápticos do neurônio k ;
- u_k é o combinador linear devido aos sinais de entrada;
- b_k é o *bias*;
- $\varphi(.)$ é a função de ativação;

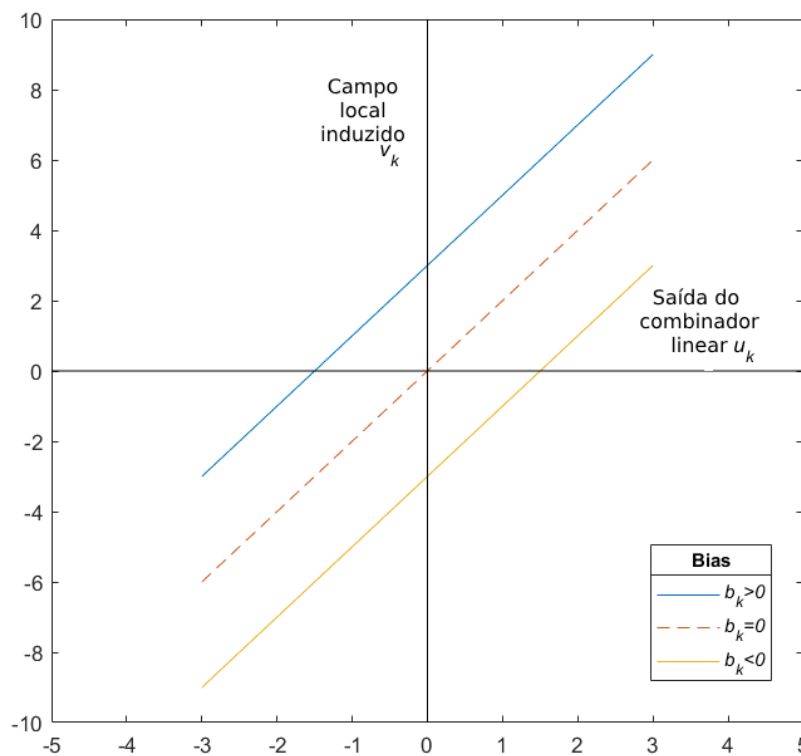
- y_k é o sinal de saída do neurônio.

O uso do *bias* b_k tem o efeito de aplicar uma transformação afim à saída u_k do combinador linear no modelo da Figura 4.2, mostrado na Equação:

$$v_k = u_k + b_k \quad (4.3)$$

Em particular, dependendo se o *bias* b_k estiver positivo ou negativo, a relação entre o campo induzido local ou potencial de ativação v_k do neurônio k e a saída do combinador linear u_k é modificada da maneira ilustrada na Figura 4.2. Observa-se que a transformação afim, no gráfico v_k versus u_k não passa mais pela origem.

Figura 4.2 – Transformação afim causado pelo *bias*



Fonte: Autor adaptado de Haykin (2009)

4.2 TIPOS DE FUNÇÃO DE ATIVAÇÃO

A função de ativação denotada por $\varphi(v)$, define a saída de um neurônio em termos do campo local induzido v . Pode-se identificar três tipos básicos de funções de ativação mostrados nas seções a seguir.

4.2.1 FUNÇÃO LIMIAR

A Função Limiar, também chamada na literatura de função de Heaviside. Para este tipo de função de ativação, ilustrada na Figura 4.3.

$$\varphi(v) = \begin{cases} 1 & \text{se } v \geq 0 \\ 0 & \text{se } v < 0 \end{cases} \quad (4.4)$$

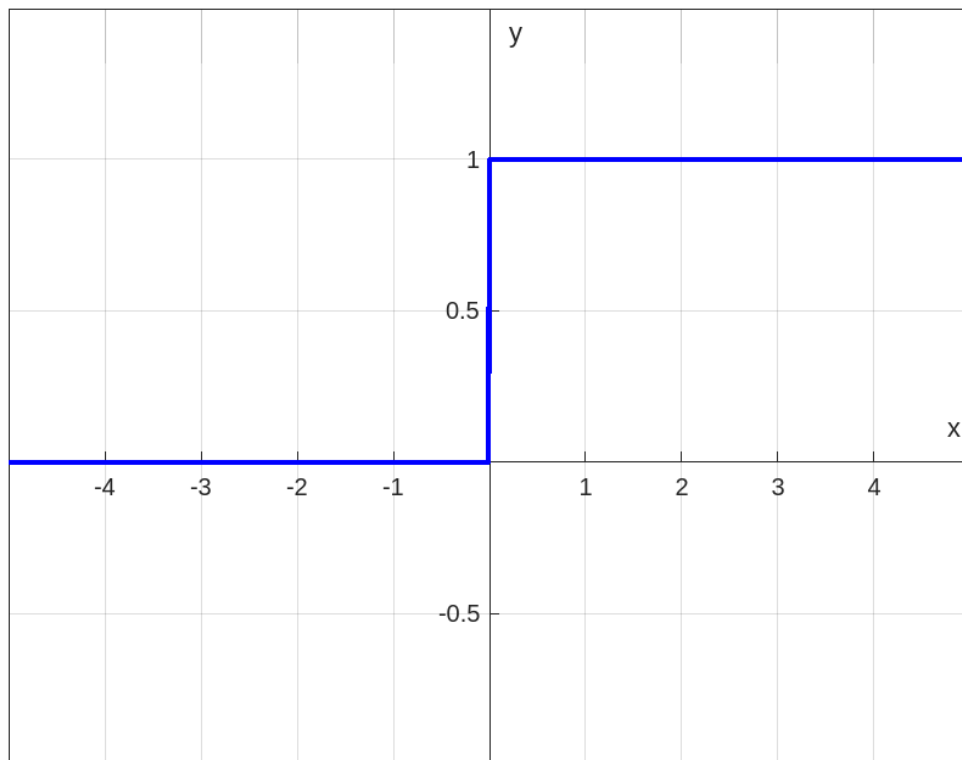
A saída do neurônio k empregando a função de limiar pode ser expressa como:

$$y(k) = \begin{cases} 1 & \text{se } v_k \geq 0 \\ 0 & \text{se } v_k < 0 \end{cases} \quad (4.5)$$

Em que v_k é o campo local induzido do neurônio.

$$v_k = \sum_{j=1}^m w_{kj} + b_k \quad (4.6)$$

Figura 4.3 – Representação gráfica da função limiar



Fonte: Autor

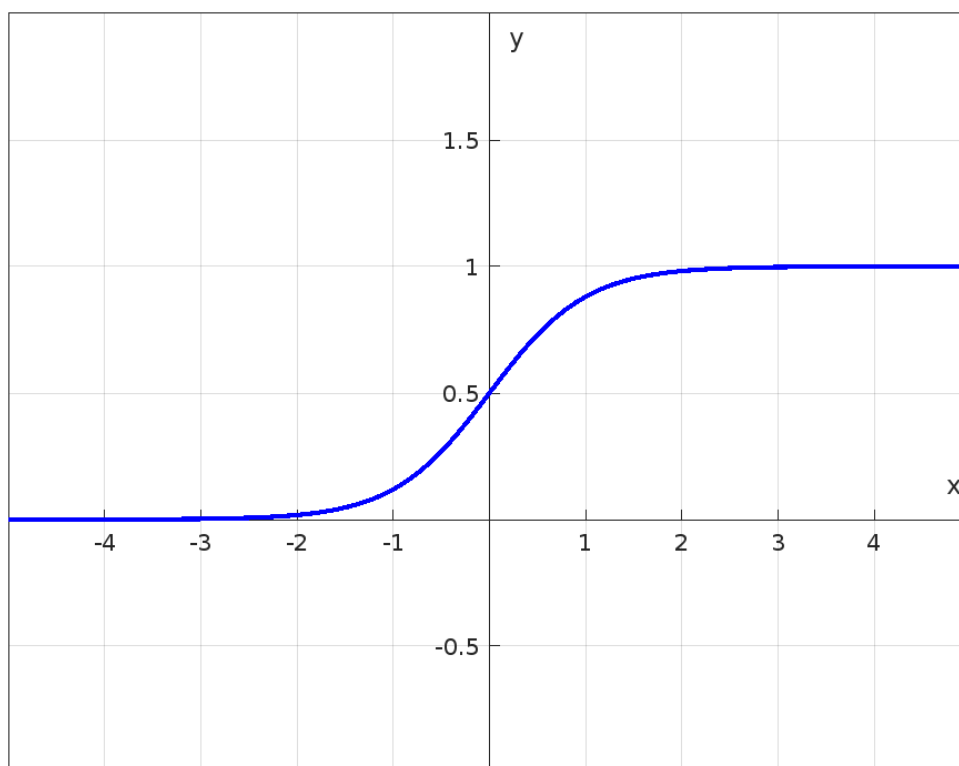
4.2.2 FUNÇÃO SIGMOIDE

A função sigmoide, cujo gráfico é em forma de "s", ilustrada na Figura 4.4, é a forma mais comum de função de ativação usada na construção de RNAs. É definida como uma função estritamente crescente que exibe um equilíbrio entre o comportamento linear e não linear. Um exemplo da função sigmoide é a função logística, definida por

$$\varphi(v) = \frac{1}{1 + \exp(-av)} \quad (4.7)$$

Em que a é o parâmetro de inclinação da função sigmoide. Variando o parâmetro a , obtêm-se funções sigmoides de diferentes inclinações. De fato, a inclinação na origem é igual a $a/4$. No limite, à medida que o parâmetro de inclinação se aproxima do infinito, a função sigmoide torna-se simplesmente uma função de limiar. Enquanto uma função de limiar assume os valores de 0 ou 1, uma função sigmoide assume uma faixa contínua de valores de 0 a 1. Pode-se observar que a função sigmoide é passível de derivação, ao contrário da função limiar, o que proporciona a sua utilização em RNAs de múltiplas camadas.

Figura 4.4 – Representação gráfica da função sigmoide



Fonte: Autor

As funções de ativação definidas nas Equações (4.4), (4.5) e (4.7) variam de 0 a +1. variam de 0 a +1. Às vezes é desejável que a função de ativação varie de -1 a +1, caso em que a função

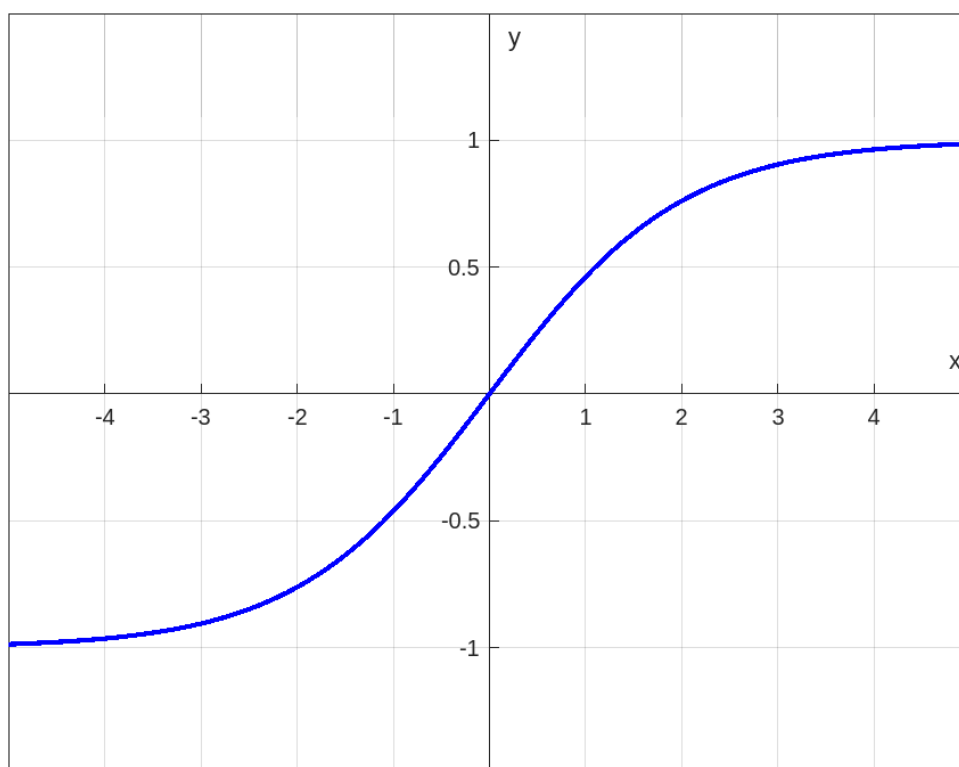
de ativação assume uma forma antissimétrica em relação à origem, ou seja, a função de ativação será uma função ímpar do campo local induzido. Nesse caso, a função é chamada de sigmoide bipolar, representada na Equação 4.8 e ilustrada na Figura 4.5.

$$\varphi(v) = \frac{2}{1 + \exp(-av)} - 1 \quad (4.8)$$

Que também pode ser escrita através da fórmula da Equação

$$\varphi(v) = \frac{1 - \exp(-av)}{1 + \exp(-av)} \quad (4.9)$$

Figura 4.5 – Representação gráfica da função sigmoide bipolar

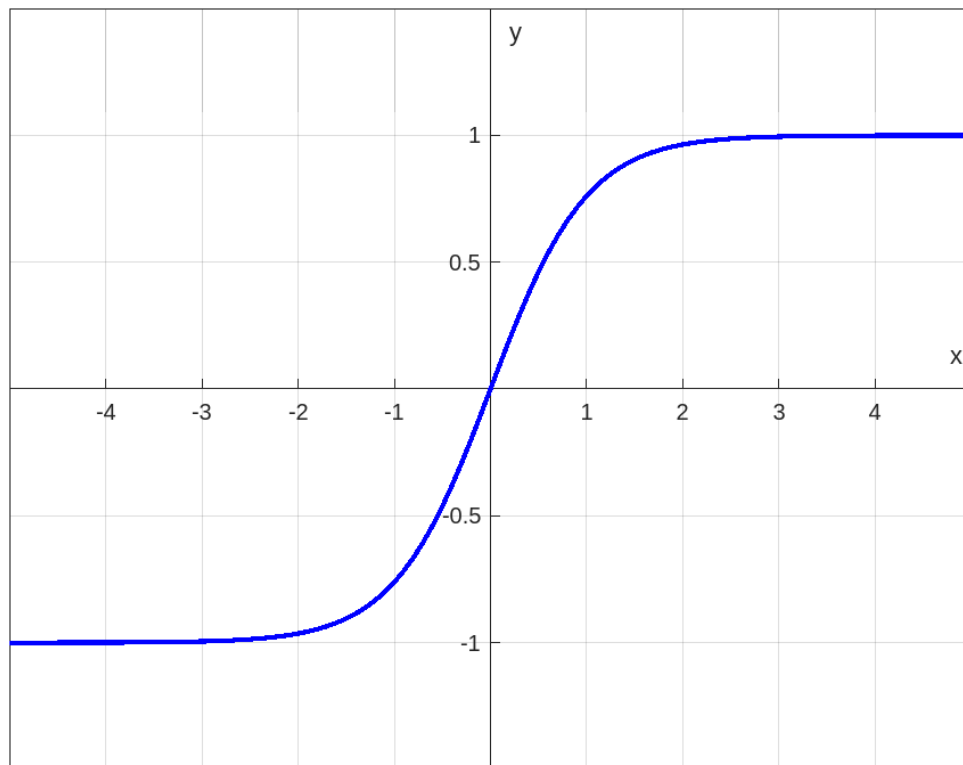


Fonte: Autor

4.2.3 FUNÇÃO TANGENTE HIPERBÓLICA

A função sigmoide bipolar está relacionada à função tangente hiperbólica, Equação 4.10, que também é frequentemente utilizada como função de ativação, quando os valores de intervalo de saída estão entre -1 e 1.

$$\varphi(v) = \tanh(v) = \frac{1 - \exp(-2v)}{1 + \exp(-2v)} \quad (4.10)$$

Figura 4.6 – Representação gráfica da função tangente hiperbólica

Fonte: Autor

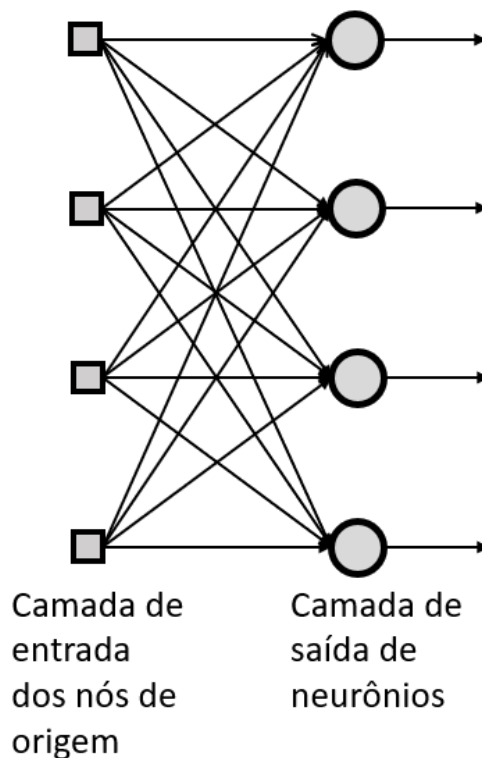
4.3 ARQUITETURA DA REDE

A forma como os neurônios de uma rede neural são estruturados está intimamente ligada ao algoritmo de aprendizado usado para treiná-la. O algoritmo de aprendizado será abordado na próxima seção.

4.3.1 REDES *Feedforward* DE CAMADA ÚNICA

Em uma rede neural, os neurônios são organizados em forma de camadas. Na mais simples forma, rede com uma camada, tem-se a camada de entrada dos nós de origem que se projeta para uma camada de saída (nós de cálculo) de neurônios, mas não vice-versa. Em outras palavras, esta rede é estritamente uma *feedforward*, ou também chamada do tipo "acíclica".

A Figura 4.7 ilustra uma RNA para o caso de quatro nós na camada de entrada e quatro nós na camada de saída. Embora na figura sejam visualizadas duas camadas de nós, a rede é denominada camada única porque somente na camada de saída ocorrem os cálculos. A primeira camada só é utilizada para a entrada dos dados não tendo nenhum procedimento matemático realizado nela.

Figura 4.7 – RNA de camada única

Fonte: auto adaptado de Haykin (2009)

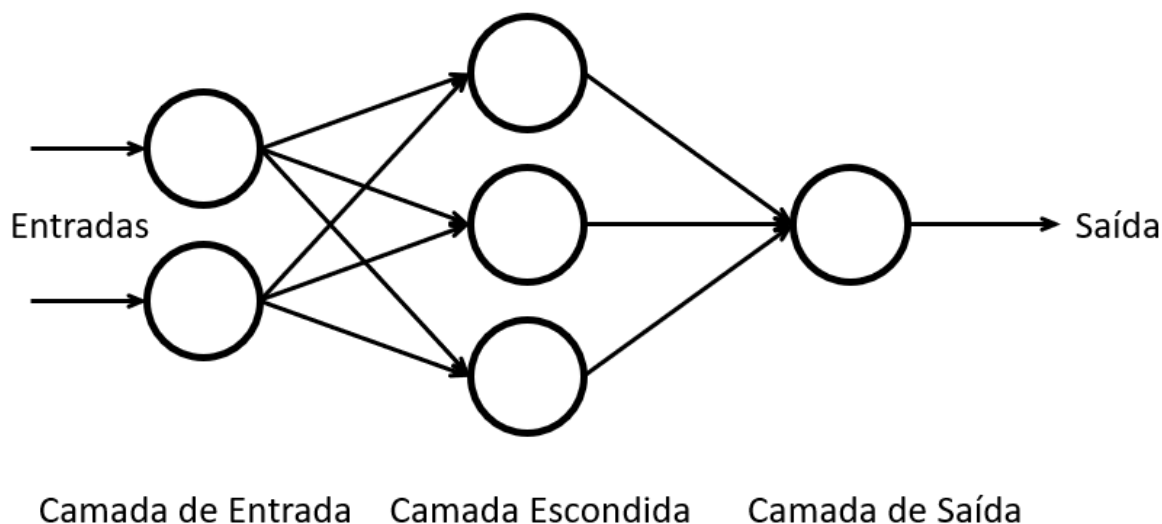
4.3.2 REDES *Feedforward* DE MÚLTIPLAS CAMADAS

A segunda classe de redes neural, *feedforward*, distingue-se da primeira pela presença de uma ou mais camadas escondidas, seus nós de cálculos são correspondentemente chamados de neurônios escondidos ou unidades escondidas. A função dos neurônios escondidos é de intervir entre a entrada externa e a saída de rede de alguma maneira útil. Ao adicionar uma ou mais camadas escondidas, a rede se torna apta a simular funções não lineares. Em um sentido amplo, a rede adquiriu uma perspectiva global, apesar de sua conectividade local, devido ao conjunto extra de conexões sinápticas e à dimensão extra das interações neurais. A capacidade dos neurônios ocultos de extrair estatísticas de ordens mais altas é particularmente valiosa quando o tamanho da camada de entrada for grande (FAUSETT, 1994).

Os nós de origem na camada de entrada da rede fornecem os respectivos elementos do padrão de ativação (vetor de entrada), que constituem os sinais de entrada aplicados aos neurônios (nós de cálculo) na segunda camada (a primeira camada escondida). Os sinais de saída da segunda camada são usados como entrada para a terceira camada e assim por diante para o restante da rede. Normalmente, os neurônios em cada camada da rede têm como entradas apenas os sinais de saída da camada anterior. O conjunto de sinais de saída dos neurônios na camada de saída (final) da rede constitui a resposta global da rede ao padrão de

ativação fornecido pelos nós de origem na camada de entrada (primeira). A Figura 4.8 ilustra uma rede neural com dois nós de origem, três neurônios escondidos e um neurônios de saída (FAUSETT, 1994).

Figura 4.8 – RNA multicamadas



Fonte: Autor adaptado de Haykin (2009)

A saída da rede neural pode ser representada pela Equação 4.11 abaixo:

$$y_k = \varphi \left(\sum_{j=1}^K w_{kj} \varphi \left(\sum_{i=1}^N w_{ji} x_i \right) \right) \quad (4.11)$$

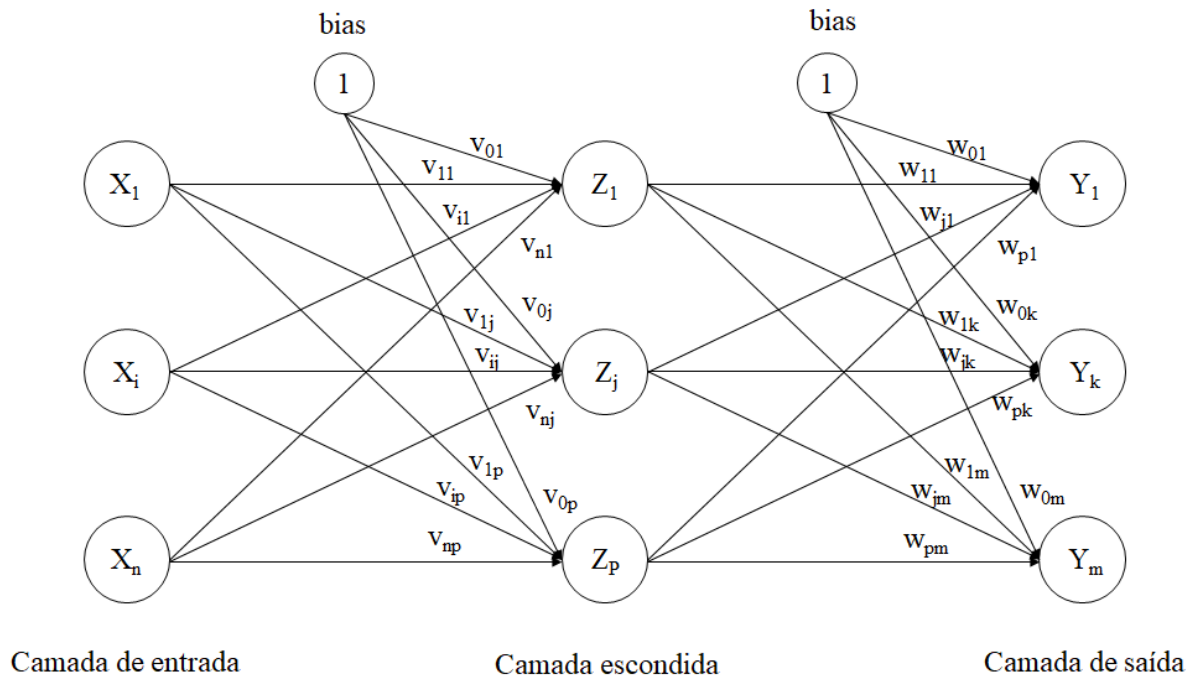
Como citado na seção anterior (4.3.1), uma rede de única camada tem a limitação de não conseguir extrair estatísticas de ordem mais altas, ao contrário da rede com múltiplas camadas. Porém, para que a rede de múltiplas camadas possa ser utilizada é necessário que ela seja treinada, o treinamento para este tipo de rede só foi possível através do método de treinamento chamado de *backpropagation (of error)*, em uma tradução livre, "retropropagação (do erro)".

O treinamento de uma rede por BP envolve 3 estágios: o estágio de propagação (*feedforward*) do padrão de entrada a ser treinado, o cálculo e a retropropagação do erro associado, e ao ajuste dos pesos. Depois do treinamento, a aplicação da rede envolve somente os cálculos da fase de propagação. Embora o treinamento às vezes seja lento, uma rede treinada pode produzir resultados de forma rápida. Muitas variações desse algoritmo têm sido desenvolvidas para melhorar a velocidade do processo de treinamento (FAUSETT, 1994).

4.4 ESTÁGIOS DO TREINAMENTO

A Figura 4.9 ilustra uma Rede Neural Multicamadas que será usada durante a explicação dos estágios de treinamento e no algoritmo de treinamento.

Figura 4.9 – RNA Multicamadas com denominações



Fonte: Autor

Neste trabalho é utilizado Fausett (1994) com referência para os estágios de treinamento. Portanto, tem-se que: X_i é o neurônio de entrada i , Z_j é o neurônio de entrada j , os neurônios da saída k são chamados de Y_k , os pesos das interconexões do neurônio da camada de entrada i até o neurônio da camada escondida j são chamados de v_{ij} e os pesos das interconexões do neurônio da camada de escondida j até o neurônio da camada de saída k são chamados de w_{jk} .

4.4.1 ESTÁGIO DE PROPAGAÇÃO

Nesta fase cada neurônio de entrada (x_i) recebe o sinal de entrada e o transmite para cada neurônio escondido Z_1, \dots, Z_p . Cada neurônio escondido calcula sua ativação e manda o seu sinal (z_j) para cada neurônio de saída. Cada neurônio de saída (Y_k) calcula sua ativação (y_j) para formar a resposta da rede para um dado padrão de entrada.

4.4.2 ESTÁGIO DE TREINAMENTO

Durante o treinamento, cada neurônio de saída compara sua ativação calculada (y_k) com o valor do objetivo (t_k) para determinar o erro associado para o padrão deste neurônio. Baseado neste erro, o fator δ_k ($k = 1, \dots, m$) é calculado. δ_k é usado para distribuir o erro no neurônio de saída Y_k de volta para todos os neurônios da camada anterior (os neurônios escondidos que são conectados a Y_k). Este fator também será usado (adiante) para atualizar os pesos entre as camadas de saída e a camada escondida. De maneira similar, o fator δ_j ($j = 1, \dots, p$) é calculado para cada neurônio escondido Z_j . Não é necessário propagar o erro para a camada de entrada, entretanto δ_j é usado para atualizar os pesos entre a camada escondida e a camada de entrada.

4.4.3 ESTÁGIO DE AJUSTE DOS PESOS

Depois de todos os fatores δ terem sido determinados, os pesos de todas as camadas são ajustados simultaneamente. O ajuste dos pesos w_{jk} (do neurônio escondido Z_j para o neurônio de saída Y_k) é baseado no fator δ_k e na ativação z_j do neurônio escondido Z_j . O ajuste dos pesos v_{ij} (do neurônio de entrada X_i para o neurônio escondido Z_j) é baseado no fator δ_j e na ativação x_i da camada de entrada.

4.5 ALGORITMO DE TREINAMENTO

O algoritmo de treinamento será descrito nesta seção através de passos.

Passo 0 - Inicializar os pesos (Escolha um conjunto de pequenos valores aleatórios).

Passo 1 - Enquanto a condição de parada for falsa, fazer do Passo 2 ao 9,

Passo 2 - Para cada par de treinamento. realizar dos passos 3 a 9,

- Propagação

Passo 3 - Cada neurônio de entrada ($X_i, i = 1, \dots, n$) recebe o sinal x_i e transmite o sinal para todos os neurônios da camada escondida.

Passo 4 - Cada neurônio escondido ($Z_j, j = 1, \dots, p$) soma com seu seu sinal de entrada ponderado de acordo com a Equação 4.12.

$$z_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (4.12)$$

Aplica-se a função de ativação (Eq.4.13) para calcular o sinal de saída e envia este sinal para a camada todas os neurônios da próxima camada (neurônios de saída).

$$z_j = f(z_in_j) \quad (4.13)$$

Passo 5 - Cada neurônio saída ($Y_k, k = 1, \dots, m$) soma com seu sinal de entrada ponderado de acordo com a Equação 4.14.

$$y_{in_k} = w_{0k} + \sum_{j=1}^n z_j w_{ij} \quad (4.14)$$

Aplica-se a função de ativação (Eq.4.15) para calcular o sinal de saída.

$$y_k = f(y_{in_k}) \quad (4.15)$$

- Retropropagação (BP) do erro:

Passo 6 - Cada neurônio de saída ($Y_k, k = 1, \dots, m$) recebe o padrão esperado correspondente ao padrão de entrada, calcula seu termo erro de informação (Eq. 4.16),

$$\delta_k = (t_k - y_k) f'(y_{in_k}) \quad (4.16)$$

é calculado o termo de correção de peso (utilizado para atualizar w_{0k} posteriormente) (Eq.4.17),

$$\Delta w_{jk} = \alpha \delta_k \quad (4.17)$$

da mesma maneira é calculada a correção para o termo *bias*.

Passo 7 - Cada neurônio escondido ($Z_j, j = 1, \dots, p$) é somado com o fator delta (das unidades da camada de saída) (Eq. 4.18),

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk} \quad (4.18)$$

é multiplicado pela derivada da função de ativação para calcular seu termo de erro de informação (Eq.4.19).

$$\delta_j = \delta_{in_j} f'(z_{in_j}) \quad (4.19)$$

Calcula-se o termo de correção de peso (usado para atualizar v_{ij} posteriormente) (Eq.4.20),

$$\delta v_{ij} = \alpha \delta_j x_i \quad (4.20)$$

Da mesma maneira é calculada a correção para o termo *bias*.

Atualização dos pesos e *bias*es.

Passo 8 - Cada neurônio de saída ($Y_k, k = 1, \dots, m$) atualiza seus pesos e *bias* ($j = 1, \dots, p$) (Eq.4.21):

$$w_{jk}(\text{novo}) = w_{jk}(\text{anterior}) + \Delta w_{jk} \quad (4.21)$$

Cada neurônio de saída ($Z_j, j = 1, \dots, p$) atualiza seus pesos e *bias* ($i = 1, \dots, n$) (Eq.4.22):

$$v_{ij}(\text{novo}) = v_{ij}(\text{anterior}) + \Delta v_{jk} \quad (4.22)$$

Passo 9 - Testar a condição de parada.

Três condições são comumente usadas como condição de parada (SHAO; TAFF; WALSH, 2011):

- Interromper o treinamento da rede quando um número pré-definido de épocas de treinamento for atingido;
- Interromper o treinamento quando uma taxa de erro pré-definida para o conjunto de treinamento for atingida;
- Interromper o treinamento quando uma taxa mínima de erro for atingida para um conjunto de validação (a ser detalhado posteriormente). Esse terceiro critério é geralmente chamado como a abordagem de validação cruzada.

Embora as duas primeiras abordagens tenham sido amplamente utilizadas na prática, um determinado número de épocas ou uma determinada taxa de erro não significa que a rede apresentou o melhor resultado (SHAO; TAFF; WALSH, 2011).

Observe que na implementação deste algoritmo existem 2 tipos de δ , para os neurônios de saída (δ_k , Passo 6) e os (δ_j , Passo 7) para os neurônios da camada escondida.

Uma época é um ciclo de todo o conjunto de vetores de treinamento. Normalmente muitas épocas são necessárias para treinar uma rede BP.

A base matemática para o algoritmo BP é a técnica de otimização conhecida como gradiente descendente. O gradiente da função (neste caso, a função é o erro e as variáveis são os pesos da rede) dá a direção na qual a função cresce mais rapidamente; o negativo do gradiente dá a direção na qual a função decresce mais rapidamente.

Apenas uma camada escondida é suficiente para uma rede BP aproximar qualquer mapeamento contínuo dos padrões de entrada para os padrões de saída com um grau de

precisão razoável. No entanto, duas camadas ocultas podem facilitar o treinamento em algumas situações.

Um problema importante que ocorre durante o treinamento da rede neural é a superadaptação, ou seja, a rede memoriza exemplos de treinamento e não aprende a generalizar novas situações. Para evitar a superadaptação e obter bons desempenhos de generalização, o conjunto de treinamento é separado em três subconjuntos (POPESCU et al., 2001):

-Treinamento: apresentado à rede durante o treinamento e os pesos são ajustados de acordo com este erro.

-Validação: usado para medir a generalização da rede e parar o treinamento quando a generalização parar de crescer.

-Teste: não tem efeito no treinamento, mas fornece uma medida independente do desempenho da rede durante e após o treinamento (MATLAB, 2010).

4.5.1 MODIFICAÇÃO LEVENBERG-MARQUARDT

Enquanto o algoritmo BP utiliza a técnica de gradiente descendente, o algoritmo de LM é uma aproximação ao método de Newton. Caso se tenha uma função $V(\underline{x})$ que se queira minimizar em relação ao vetor parâmetro \underline{x} , então o método de Newton seria:

$$\Delta \underline{x} = -[\nabla^2 V(\underline{x})]^{-1} \nabla V(\underline{x}) \quad (4.23)$$

Em que $\nabla^2 V(\underline{x})$ é uma matrix Hessiana e $\nabla V(\underline{x})$ é o gradiente. Se for assumido que $V(\underline{x})$ é a soma da função quadrática

$$V(\underline{x}) = \sum_{i=1}^N e_i^2(\underline{x}) \quad (4.24)$$

Então pode ser mostrado que

$$\nabla V(\underline{x}) = J^T(\underline{x}) e(\underline{x}) \quad (4.25)$$

$$\nabla^2 V(\underline{x}) = J^T(\underline{x}) J(\underline{x}) + S(\underline{x}) \quad (4.26)$$

Em que $J(\underline{x})$ é a matriz jacobiana

$$J(\underline{x}) = \begin{bmatrix} \frac{\partial e_1(x)}{\partial x_1} & \frac{\partial e_1(x)}{\partial x_2} & \dots & \frac{\partial e_1(x)}{\partial x_n} \\ \frac{\partial e_2(x)}{\partial x_1} & \frac{\partial e_2(x)}{\partial x_2} & \dots & \frac{\partial e_2(x)}{\partial x_3} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_N(x)}{\partial x_1} & \frac{\partial e_N(x)}{\partial x_2} & \dots & \frac{\partial e_N(x)}{\partial x_3} \end{bmatrix} \quad (4.27)$$

e

$$S(\underline{x}) = \sum_{i=1}^N e_i(\underline{x}) \nabla^2 e_i(\underline{x}) \quad (4.28)$$

No método de Gauss-Newton assume-se que $S(\underline{x}) \approx 0$, portanto a atualização fica da seguinte forma:

$$\Delta \underline{x} = -[J^T(\underline{x})J(\underline{x})]^{-1} J^T(\underline{x})e(\underline{x}) \quad (4.29)$$

A modificação de Levenberg-Marquardt no método de Gauss-Newton para encontrar a atualização $\Delta \underline{x}$ é a seguinte:

$$\Delta \underline{x} = -[J^T(\underline{x})J(\underline{x}) + \mu I]^{-1} J^T(\underline{x})e(\underline{x}) \quad (4.30)$$

O parâmetro μ é multiplicado por algum fator (β) sempre que um passo resultar em um aumento de $V(\underline{x})$. Quando um passo diminui $V(\underline{x})$, μ é dividido por β . É observado que a medida que μ cresce o algoritmo se torna gradiente descendente, por outro lado, a medida que μ diminui algoritmo se torna Gauss-Newton. Por este motivo, o algoritmo de Levenberg-Marquardt pode ser considerado uma modificação de região de confiança para Gauss-Newton.

O diferencial neste algoritmo é o cálculo da matriz Jacobiana. Para a resolução de problemas de mapeamento de rede neural, os termos da matriz Jacobiana podem ser calculados por uma simples modificação no algoritmo de BP.

O algoritmo LM é muito eficiente no treinamento de redes com até algumas centenas de pesos. Embora os requisitos computacionais sejam muito maiores para cada iteração do algoritmo, isso é mais do que compensado pelo aumento da eficiência. Isso é especialmente verdadeiro quando é necessária alta precisão (HAGAN; MENHAJ, 1994).

Em geral o algoritmo LM consegue treinar RNAs a uma taxa de 10 a 100 vezes mais rápido (dependendo do problema) do que o algoritmo padrão BP de gradiente descendente (HAGAN; MENHAJ, 1994). Ostlin, Zepernick e Suzuki (2010) observaram que o algoritmo de LM é cerca de 1.000 vezes mais rápido que o algoritmo de BP padrão, o que potencialmente permite que uma quantidade substancialmente maior de dados de treinamento seja incorporada no processo de treinamento.

4.6 CONSIDERAÇÕES FINAIS

Neste capítulo, foram apresentados os fundamentos das RNAs, abordando suas principais características e funcionamento. Iniciamos com os Modelos de Neurônios (Seção 4.1), descrevendo os elementos básicos que compõem essas redes. Em seguida, discutimos os Tipos de Função de Ativação (Seção 4.2), detalhando as características das funções Limiar (Seção 4.2.1), Sigmoide (Seção 4.2.2) e Tangente Hiperbólica (Seção 4.2.3). A Arquitetura da Rede (Seção 4.3) foi explorada, diferenciando as Redes *Feedforward* de camada única (Seção 4.3.1) das Redes *Feedforward* de múltiplas camadas (Seção 4.3.2).

Além disso, foram descritos os Estágios do Treinamento (Seção 4.4), que incluem os processos de Propagação (Seção 4.4.1), Treinamento (Seção 4.4.2) e Ajuste dos Pesos (Seção 4.4.3). Por fim, apresentamos o Algoritmo de Treinamento (Seção 4.5), com destaque para a Modificação Levenberg-Marquardt (Seção 4.5.1), que otimiza o desempenho das RNAs. Esses tópicos formam a base teórica essencial para a compreensão e aplicação das RNAs.

ALGORITMOS GENÉTICOS E EVOLUÇÃO DIFERENCIAL

Resumo do capítulo

Neste capítulo, são abordados os AGs e a ED, destacando sua aplicabilidade na otimização de modelos de propagação em redes de comunicação. São apresentados os principais conceitos, como seleção, cruzamento e mutação, além de detalhes sobre o processo de otimização utilizando esses algoritmos.

5.1 ALGORÍTIMOS GENÉTICOS

Os AG foram desenvolvidos por John Holland e seus colaboradores nas décadas de 1960 e 1970. É um modelo ou abstração da evolução biológica baseado na teoria da seleção natural de Charles Darwin. Holland foi provavelmente o primeiro a usar cruzamento e recombinação, mutação e seleção no estudo de sistemas adaptativos e artificiais. Esses operadores genéticos constituem a parte essencial do algoritmo genético como estratégia de resolução de problemas. Desde então, muitas variantes de AGs foram desenvolvidas e aplicadas a uma ampla gama de problemas de otimização, desde coloração de grafos até reconhecimento de padrões, desde sistemas discretos (como o problema do caixeiro viajante) até sistemas contínuos (por exemplo, o projeto de aerofólio em engenharia aeroespacial) e dos mercados financeiros à otimização da engenharia multiobjetivo (YANG, 2014).

Existem muitas vantagens dos AGs sobre os algoritmos de otimização tradicionais. Dois dos mais notáveis são: a capacidade de lidar com problemas complexos e paralelismo. AGs podem lidar com vários tipos de otimização, seja a função objetivo (adequação) estacionária ou não estacionária (mudanças com o tempo), linear ou não linear, contínua ou descontínua, ou com ruído aleatório. Como múltiplos descendentes em uma população agem como agentes independentes, a população (ou qualquer subgrupo) pode explorar o espaço de busca em muitas direções simultaneamente. Esse recurso torna ideal paralelizar os algoritmos para implementação. Diferentes parâmetros e até diferentes grupos de strings codificadas podem ser manipulados ao mesmo tempo (YANG, 2014).

No entanto, os AGs também apresentam algumas desvantagens. A formulação da função de aptidão, o uso do tamanho da população, a escolha de parâmetros importantes como a taxa de mutação e cruzamento e os critérios de seleção da nova população devem ser realizados com cuidado. Qualquer escolha inadequada dificultará a convergência do algoritmo ou simplesmente produzirá resultados sem sentido. Apesar dessas desvantagens, os AGs continuam sendo um dos algoritmos de otimização mais amplamente utilizados na otimização não linear moderna (YANG, 2014).

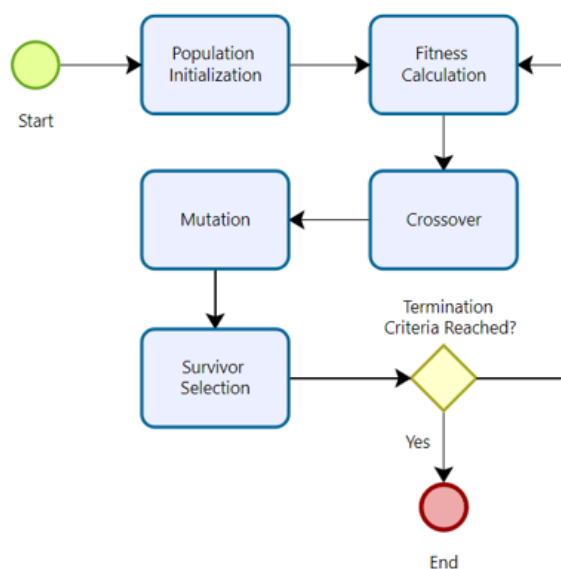
5.1.1 DESCRIÇÃO DO MÉTODO

A essência dos AGs reside na representação de uma função de otimização por meio de matrizes de bits ou cadeias de caracteres, que simulam cromossomos. O processo envolve a manipulação dessas cadeias por operadores genéticos, como cruzamento e mutação, e a seleção baseada na aptidão das soluções, com o objetivo de identificar a solução ideal para o problema proposto (KALEYBAR et al., 2023).

De maneira geral, o procedimento está ilustrado na Figura 5.1 e tendo como referência Kaleybar et al. (2023), segue as etapas descritas a seguir:

1. Codificação dos objetivos ou funções de custo;
2. Definição de uma função de aptidão ou critério de seleção;
3. Criação de uma população de indivíduos;
4. Realização do ciclo de evolução ou iterações avaliando a aptidão de todos os indivíduos da população, criando uma nova população através de cruzamento e mutação, reprodução proporcional à aptidão etc., e substituição da população antiga e iteração novamente usando a nova população;
5. Decodificação dos resultados para obtenção da solução do problema.

Figura 5.1 – Fluxograma do Algoritmo Genético



Fonte: Autor

Uma iteração destinada à formação de uma nova população é denominada geração. Na maioria dos AGs, utilizam-se cadeias de caracteres de comprimento fixo durante cada

geração, embora exista uma quantidade significativa de pesquisas focadas em cadeias de comprimento variável e em diferentes estruturas de codificação. A codificação da função objetivo é comumente realizada por meio de matrizes binárias ou matrizes com valores reais, especialmente em AGs adaptativos.

5.1.2 POPULAÇÃO INICIAL

A população no algoritmo genético consiste em várias soluções que correspondem aos cromossomos dos indivíduos, e cada cromossomo contém uma coleção de variáveis que simulam genes. Durante a etapa de inicialização, o objetivo principal é distribuir as soluções uniformemente ao longo do espaço de busca para maximizar a diversidade da população e aumentar a probabilidade de identificar áreas promissoras.

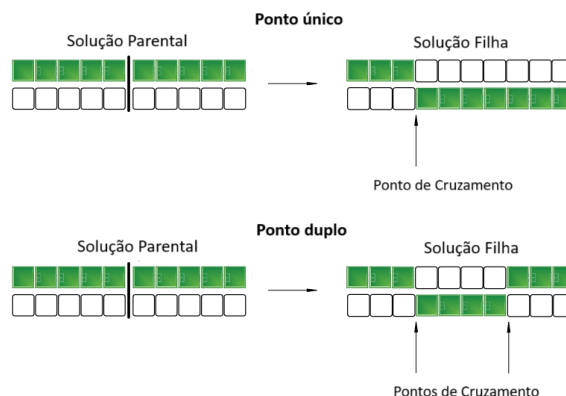
5.1.3 SELEÇÃO

A base fundamental para esta parte do algoritmo genético é derivada da seleção natural. Na natureza, os indivíduos mais fortes têm maiores chances de obter comida e se acasalar, levando a que seus genes se tornem mais prevalentes na próxima geração. Consequentemente, o algoritmo genético utiliza uma roda de roleta para atribuir probabilidades aos indivíduos com base em seus níveis de aptidão e os seleciona para criar a geração subsequente proporcionalmente aos seus valores objetivos.

Devido à natureza estocástica de uma roda de roleta, os indivíduos menos aptos têm baixas chances de contribuir para a formação da próxima geração. No entanto, se uma solução menos satisfatória for selecionada, sua composição genética ainda pode ser transmitida para a próxima geração. Portanto, é importante evitar eliminar tais soluções, pois fazê-lo diminuiria a variedade dentro da população.

5.1.4 CRUZAMENTO

Uma vez que o operador de seleção identificou os indivíduos, eles devem ser utilizados para gerar a próxima geração. Naturalmente, os cromossomos de um macho e de uma fêmea se combinam para criar um novo cromossomo. Como ilustrado na Figura 5.2 em um algoritmo genético, duas soluções selecionadas (soluções parentais) são combinadas usando técnicas como cruzamento, também chamado de *crossover*, de ponto único ou cruzamento de ponto duplo, para gerar duas novas soluções (soluções filhas) (KALEYBAR et al., 2023).

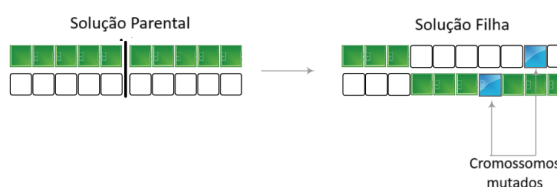
Figura 5.2 – Técnicas de Cruzamento

Fonte: Autor adaptado de Kaleybar et al. (2023)

O cruzamento de ponto único envolve a troca dos cromossomos de duas soluções parentais em um único ponto, tanto antes quanto depois desse ponto. Por outro lado, o cruzamento de ponto duplo envolve dois pontos de cruzamento, onde apenas os cromossomos entre esses pontos são alterados (KALEYBAR et al., 2023).

5.1.5 MUTAÇÃO

O operador de mutação (Figura 5.3) é usado para adicionar um nível adicional de aleatoriedade à população, o que ajuda a manter a diversidade e evita que as soluções se tornem muito semelhantes; como resultado, a probabilidade de evitar soluções locais é aumentada dentro do algoritmo genético. Para evitar que um algoritmo genético se torne uma busca aleatória básica, a taxa de mutação deve ser pequena, pois taxas elevadas de mutação podem ter esse efeito (KALEYBAR et al., 2023).

Figura 5.3 – Operador Mutação

Fonte: Autor adaptado de Kaleybar et al. (2023)

5.1.6 CRITÉRIO DE PARADA

O requisito básico de um AG, assim como de qualquer algoritmo, é convergir para uma solução satisfatória em um tempo razoável. A taxa de progressão de um AG é geralmente alta no

início, mas tende a diminuir conforme se aproxima de uma solução ótima. Por isso, a condição de término deve ser cuidadosamente definida para garantir que o algoritmo pare quando atingir um critério de parada predefinido ou quando o benefício de continuar a busca não justificar o custo computacional. Segundo Hassanat et al. (2019) as três condições de parada mais comuns são:

- Quando não há melhorias na população por X iterações.
- Quando alcançamos um número absoluto de iterações.
- Quando o valor da função objetivo atinge um determinado valor previamente definido (possível solução ótima).

5.1.7 FUNÇÃO APTIDÃO

A função de aptidão, também chamada de função de avaliação, tem como objetivo avaliar quão próxima da solução ótima ou da solução desejada o AG alcançou. De acordo com a regra de seleção natural, os cromossomos mais aptos são enviados para o próximo nível, enquanto os demais são eliminados. Os cromossomos que passam para a próxima geração atuam como cromossomos parentais, que serão posteriormente utilizados para a reprodução de novos cromossomos filhos por meio de dois métodos, a saber, cruzamento e mutação.

A identificação do cromossomo mais apto é feita atribuindo a cada cromossomo um valor de aptidão ao final de cada geração. Esse valor de aptidão é fornecido por uma função conhecida como função de aptidão. Cada cromossomo recebe uma pontuação, indicando quão próxima a solução está da ótima. Existem algumas exigências genéricas para uma função de aptidão. Elas são as seguintes:

- A função de aptidão deve ser claramente definida;
- A otimização da função de aptidão é de máxima prioridade. Se a função de aptidão se tornar um gargalo para todo o algoritmo, a eficiência do algoritmo será grandemente afetada;
- A função de aptidão deve medir quantitativamente quão adequada é a solução dada para resolver o problema em questão;
- As funções de aptidão devem fornecer resultados intuitivos. Os cromossomos melhores e piores devem receber as melhores e piores pontuações de aptidão, respectivamente.

5.2 EVOLUÇÃO DIFERENCIAL

O algoritmo clássico de ED, proposto por Storn e Price em 1997 (STORN; PRICE, 1997), é um algoritmo metaheurístico baseado em população (POLÁKOVÁ; TVRDÍK; BUJOK, 2019),

principalmente utilizado para problemas de otimização numérica. Similar a outros algoritmos metaheurísticos, como o AG citado na seção anterior (5.1), sua população contém um grupo de vetores compostos por indivíduos candidatos $x_i^G = (x_1^G, x_2^G, \dots, x_D^G, i = 1, 2, \dots, N_p$, onde G é o contador de geração, N_p é o tamanho da população e D é a dimensionalidade do problema. A ED possui quatro operadores para evoluir a população (MOHAMED; MOHAMED, 2017). A metodologia são apresentadas a seguir tendo como referencia Zhang, Cai e Zhang (2020).

5.2.1 POPULAÇÃO INICIAL

Na Geração $G = 0$, cada indivíduo na população é inicializado da seguinte forma:

$$\begin{aligned} x_{i,j} &= LB_j + randu(0, 1) \cdot (UB_j - LB_j), \\ i &= 1, 2, \dots, N_p, \\ j &= 1, 2, \dots, D \end{aligned} \quad (5.1)$$

em que $x_{i,j}$ é a j -ésima dimensão do i -ésimo indivíduo na população; LB_j e UB_j representam os limites inferior e superior na j -ésima dimensão, respectivamente; e $randu(0, 1)$ é um número aleatório distribuído uniformemente em $[0, 1]$.

5.2.2 MUTAÇÃO

Quando a população é inicializada, os indivíduos passarão pela mutação. A equação abaixo apresenta a mutação comumente utilizada:

$$v_i^G = x_{r_1}^G + F \cdot (x_{r_2}^G - x_{r_3}^G) \quad (5.2)$$

em que, r_1, r_2, r_3 são selecionados aleatoriamente de $\{1, 2, \dots, N_p\}$ e satisfazem $r_1 \neq r_2 \neq r_3 \neq i$; F é um fator de escala e v_i^G é o vetor mutante.

5.2.3 CRUZAMENTO

Após a mutação, a operação de cruzamento é introduzida com o objetivo de aumentar a diversidade da população, conforme mostrado abaixo:

$$u_i^G = \begin{cases} v_{ij}^G, & \text{se } randu(0,1) \leq CR \text{ ou } j = j_{rand} \\ x_{ij}^G, & \text{demais casos} \end{cases} \quad (5.3)$$

onde j_{rand} denota um número inteiro selecionado aleatoriamente de $\{1, D\}$; CR é a taxa de cruzamento; e x_i^G é o vetor alvo.

5.2.4 SELEÇÃO

Após o cruzamento, uma seleção por torneio "um para um" é utilizada para comparar o vetor de teste u_i^G com seu vetor alvo x_i^G . Se u_i^G for melhor ou igual a x_i^G , ele será adicionado à próxima população; caso contrário, o vetor de teste será descartado.

$$x_i^{G+1} = \begin{cases} u_i^G, & \text{se } f(u_i^G) \leq f(x_i^G) \\ x_i^G, & \text{demais casos} \end{cases} \quad (5.4)$$

em que $f(u_i^G)$ é a função de aptidão de x_i^G .

Na ED, a mutação, o cruzamento e a seleção se repetirão até que o critério de término seja atendido. Em geral, há três parâmetros (ou seja, N_p , CR e F) que precisam ser fornecidos pelo usuário.

5.3 CONSIDERAÇÕES FINAIS

Neste capítulo, foram apresentados os fundamentos teóricos e metodológicos dos AGs e da ED. Na Seção 5.1, explorou-se os AGs, abordando desde a descrição do método (Seção 5.1.1) até os principais componentes do processo, como a definição da população inicial (Seção 5.1.2), os operadores de seleção (Seção 5.1.3), cruzamento (Seção 5.1.4) e mutação (Seção 5.1.5), bem como os critérios de parada (Seção 5.1.6) e a função aptidão (Seção 5.1.7). Em seguida, a Seção 5.2 apresentou os conceitos da ED, detalhando etapas semelhantes, como a inicialização da população (Seção 5.2.1), mutação (Seção 5.2.2), cruzamento (Seção 5.2.3) e seleção (Seção 5.2.4). Esses conteúdos fornecem uma base sólida para a compreensão e aplicação das técnicas de otimização abordadas neste capítulo, consolidando os conceitos discutidos.

Resumo do capítulo

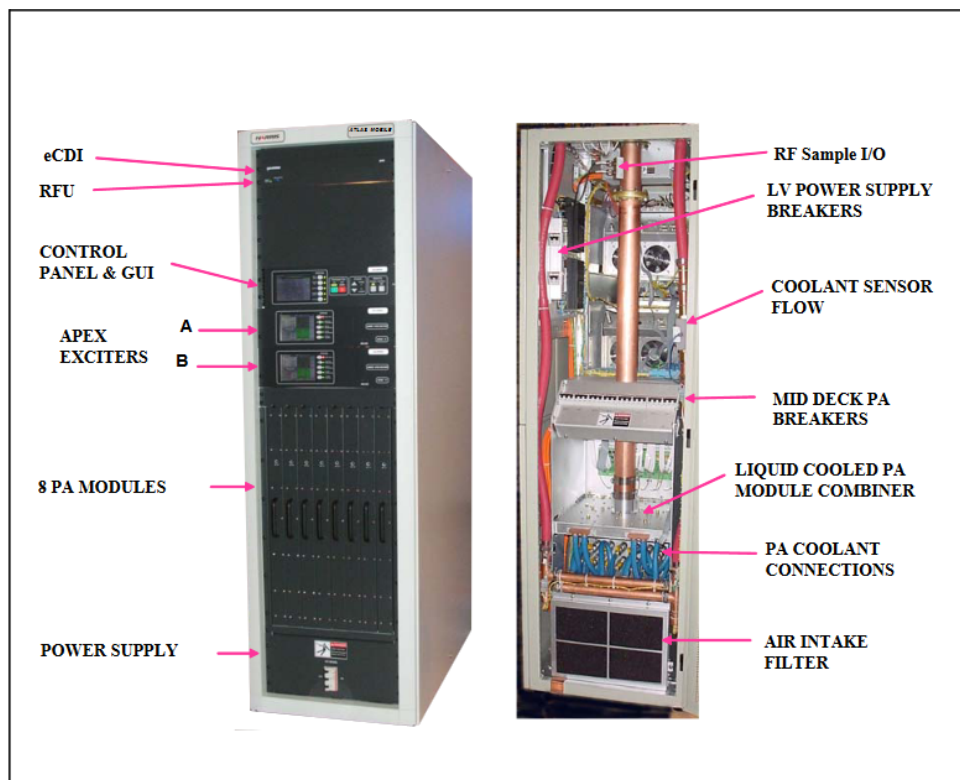
Este capítulo apresenta a metodologia adotada para o desenvolvimento e avaliação do sistema proposto, abrangendo desde os aspectos técnicos de transmissão e recepção até o processamento dos dados e a implementação das RNAs. Detalham-se as configurações dos parâmetros utilizados nos AGs e na ED, essenciais para a otimização do sistema. Cada etapa é descrita de forma sistemática, destacando-se as abordagens utilizadas e suas respectivas justificativas. Por fim, são feitas considerações sobre os desafios e avanços obtidos ao longo do processo.

A coleta de dados ocorreu em ruas, avenidas e rodovia na cidade de Uberlândia que está localizada na região oeste do Estado de Minas Gerais, chamada Triângulo Mineiro, ($18^{\circ}55'23''$ S e $48^{\circ}17'19''$ W). A cidade possui altitude média de 863 metros, possui uma área urbana de 172,85 km² e 219 km de perímetro. O clima da cidade é classificado em clima tropical com invernos secos e verões chuvosos denominados por sistemas intertropicais e polares. (SILVA et al., 2018). A vegetação natural é o Cerrado – savana brasileira, floresta, campos hidromórficos, campos de várzea e campos hidrofílicos, que cobre 17,7% do município de Uberlândia enquanto as áreas urbanas cobrem 4,6%. Excluindo os maiores centros metropolitanos, Uberlândia é uma das regiões mais populosas do país. De acordo com o último censo, a cidade tem 713.224 habitantes (IBGE, 2022). Neste capítulo será descrito o sistema de transmissão, sistema de recepção, procedimento para coleta de dados.

6.1 SISTEMA DE TRANSMISSÃO

O sistema de transmissão pertencente à emissora TV Integração está localizado nas coordenadas geográficas: latitude $18^{\circ} 53' 06''$ S longitude $48^{\circ} 15' 42''$ W, a uma altitude de 937 metros. Esse sistema é composto por um transmissor, linha de transmissão e um sistema irradiante e suas características são descritas pela Tabela 6.1.

O transmissor, ilustrado em sua parte frontal e traseira na Figura 6.1, tem como fabricante a empresa HARRIS Broadcast Communications cujo modelo DVI3400 opera no canal 30 (de acordo com a Anatel) na faixa de frequência de 566 a 572 MHz, sua frequência de operação é 569,142857 MHz, e a potência de transmissão é 2,5 kW.

Figura 6.1 – Visão dianteira e traseira do transmissor DVI3400

Fonte: Harris (2007)

A Figura 6.2 ilustra o cabo utilizado na linha de transmissão da empresa Radio Frequency Systems (RFS) e modelo HCA 158-50J, comprimento de 71 m, diâmetro de 1 5/8, perda de inserção de 1,717 dB/100 m, eficiência de 0,704, 0,25 dB de perdas acessórios e impedância de 50 ohms.

Figura 6.2 – Cabo utilizado na linha de transmissão

Fonte: RFS (2007)

A antena possui polarização elíptica de 70x30, correspondendo às componentes horizontal

e vertical, respectivamente, com ganhos de 13,3484 dBi na horizontal e 9,6684 dBi na vertical. O centro de radiação está a 64 metros de altura, e o *beam tilt* é de 4,0°. Fabricada pela empresa Ideal Antenas, a antena é do modelo ISDE082822UT. Informações técnicas detalhadas sobre o sistema irradiante estão disponíveis no Anexo A, conforme dados extraídos do *Website* da Anatel (ANATEL, 2024).

A Tabela 6.1 apresenta os dados do sistema irradiante que será utilizado neste trabalho.

Tabela 6.1 – Características do Sistema Irradiante

Localização da estação transmissora	-18,885° -48,2617°
Canal	30
Faixa de Frequência	566 a 572 MHz
Frequência da Portadora	569,142857 MHz
Potência do Transmissor	2,5 kW
Modelo da Antena	ISDE083022UT
Beam-tilt	4°
Azimuth	210°
Polarização	Elíptica (70x30)
Ganho da Antena	13,3484 dBi (Horizontal)
Transmissora	9,6684 dBi (Vertical)
Centro de Irradiação	64 m
Altura da Linha	
de Transmissão	HCA158-50J
Modelo da Linha	
Comprimento da Linha	71 m
de Transmissão	
Outras Perdas	1,27 dB
Largura do Feixe Vertical	6,5°
da Antena Transmissora	

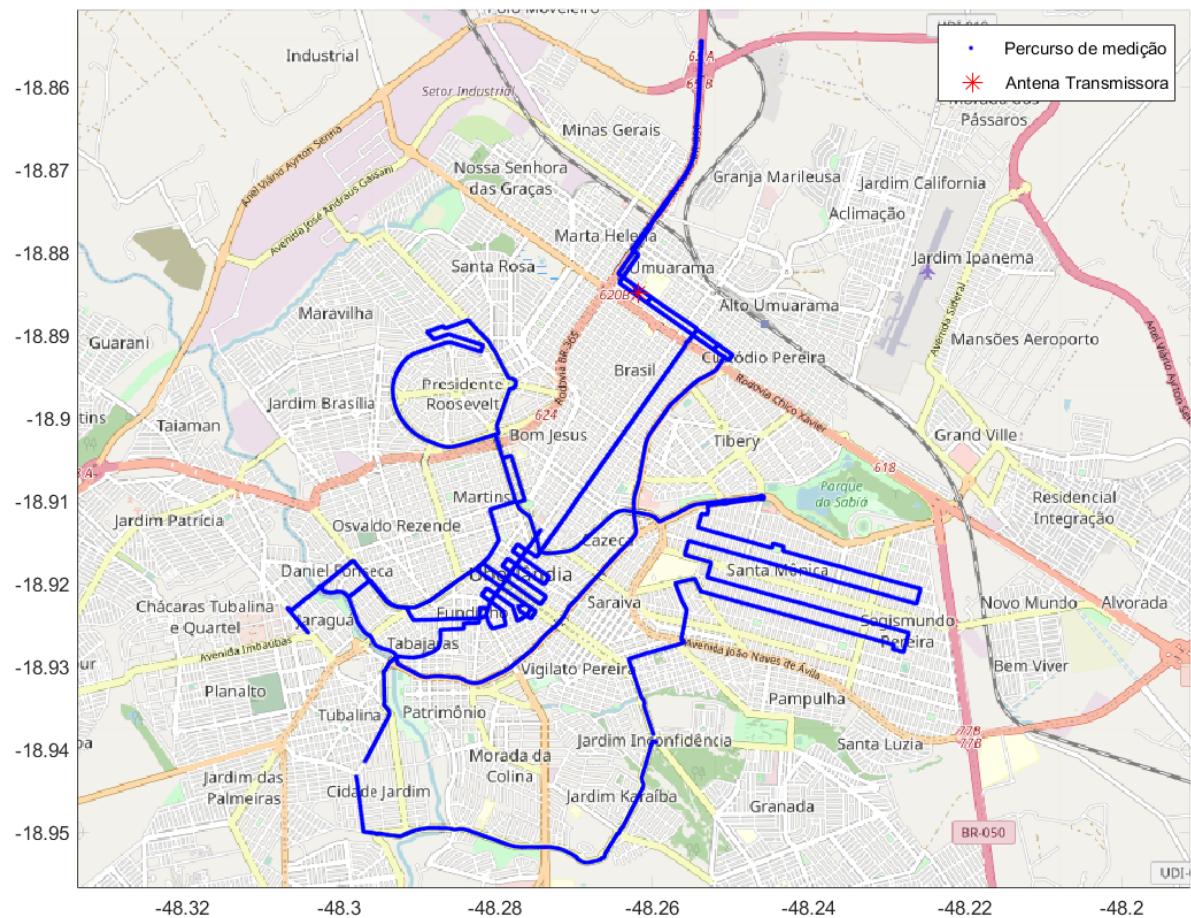
6.2 SISTEMA DE RECEPÇÃO

O sistema de recepção ilustrado na (Figura 6.3) é composto por um analisador de cobertura modelo TSMW produzido pela empresa Rhode & Schwarz, duas antenas verticais com ganho unitário, um dispositivo de Sistema de Posicionamento Global (GPS) e um laptop com o software ROMES que armazena em um arquivo no formato Keyhole Markup Language (kml), que é uma linguagem de programação baseada em XML. Com intuito de armazenar conteúdo geográfico, as informações coletadas pelo analisador e georreferenciado por um GPS.

Figura 6.3 – Sistema de Recepção

Fonte: Santos (2021)

As amostras foram coletadas e cedidas gentilmente pelo Dr. Tiago Nunes Santos que realizou um trabalho em conjunto com o Professor Dr. Edgar Silva Júnior. O sistema de recepção foi embarcado em um carro que percorre o caminho representado na Figura 6.4 abaixo. O trajeto contempla as principais vias da cidade de Uberlândia de forma que seja colhidas amostras nas regiões, norte, sul, leste e oeste. Foram percorridos 80 quilômetros e colhidas 9.319 amostras obtidas pela antena 1 e 9.319 pela antena 2 (potência, latitude, longitude e altitude). Na Figura 6.4 estão ilustrados: o trajeto percorrido com o carro, onde foram coletadas as amostras e a localização do sistema transmissor.

Figura 6.4 – Trajeto onde foram realizadas as medições

Fonte: Autor

6.3 PROCESSAMENTO DOS DADOS

Os dados coletados no arquivo de formato “kml” foram convertidos em uma planilha. Nesta planilha, cada linha representa uma amostra, ou seja, cada linha está associada a uma coordenada específica coletada pelo GPS e associado a diversos parâmetros alocados nas colunas conforme lista abaixo:

- Coluna 1 – Latitude;
- Coluna 2 – Longitude;
- Coluna 3 – Altitude;
- Coluna 4 – Potência medida pela Antena 1;
- Coluna 5 – Potência medida pela Antena 2;
- Coluna 6 – Distância entre a amostra e a antena transmissora;

- Coluna 7 – Média entre potência medida pela Antena 1 e Antena 2;
- Coluna 8 – Perda de percurso.

A partir do banco de dados foi gerada a Tabela 6.2, nela está a descrição estatística da coleta da amostra, portanto na primeira linha (média), foi calculada a média aritmética dos parâmetros que estão nas colunas do banco de dados, latitudes, longitudes, altitudes, distâncias e PL, ou seja, as colunas da tabela, na segunda linha foi calculado o desvio padrão dos parâmetros supracitados, da mesma maneira foram geradas as linhas 3 e 4, respectivamente com os valores mínimos e máximos de cada parâmetro. Essa tabela foi constituída com o intuito de situar sobre o local em que foi realizado o estudo de caso.

Tabela 6.2 – Descrição estatística do conjunto de dados medidos

	Latitude (graus)	Longitude (graus)	Altitude (m)	Distância (km)	Perda de Percurso (dB)
Média	-18,914	-48,2707	862,49	3,93	133,36
Desvio Padrão	0,0175	0,018	39,48	1,8	11,37
Mínimo	-18,9537	-48,3066	770,75	0,03	93,95
Máximo	-18,8545	-48,2257	943,91	8,12	163,51

Com os dados de latitude e longitude e os dados de latitude e longitude da antena transmissora foi calculado a distância entre a amostra e a antena transmissora através da Equação 6.1. Essa distância foi alocada na coluna 6.

$$d = 6378,137 \cos(\cos(lat_1)\cos(lat_2)\cos(lon_2 - lon_1) + \sin(lat_1)\sin(lat_2)) \quad (6.1)$$

Como cada amostra também possui dois dados de potência (coluna 4 e coluna 5), calculou-se a média entre elas e alocou-se o resultado na coluna 7, ou seja, o valor médio encontrado será utilizado no decorrer deste trabalho. Assim, a partir dos dados do sistema transmissão e as amostras colhidas foi calculado a $PL_{(dB)}$ utilizando a Equação 6.13 e alocado o resultado na coluna 8.

$$PL_{(dB)} = Pt_{(dB)} + Gt_{(dB)} - Pr_{(dB)} - Gr_{(dB)} \quad (6.2)$$

Em que: $PL_{(dB)}$ é o valor da PL;

$-Pt_{(dB)}$ é o valor da potência transmitida;

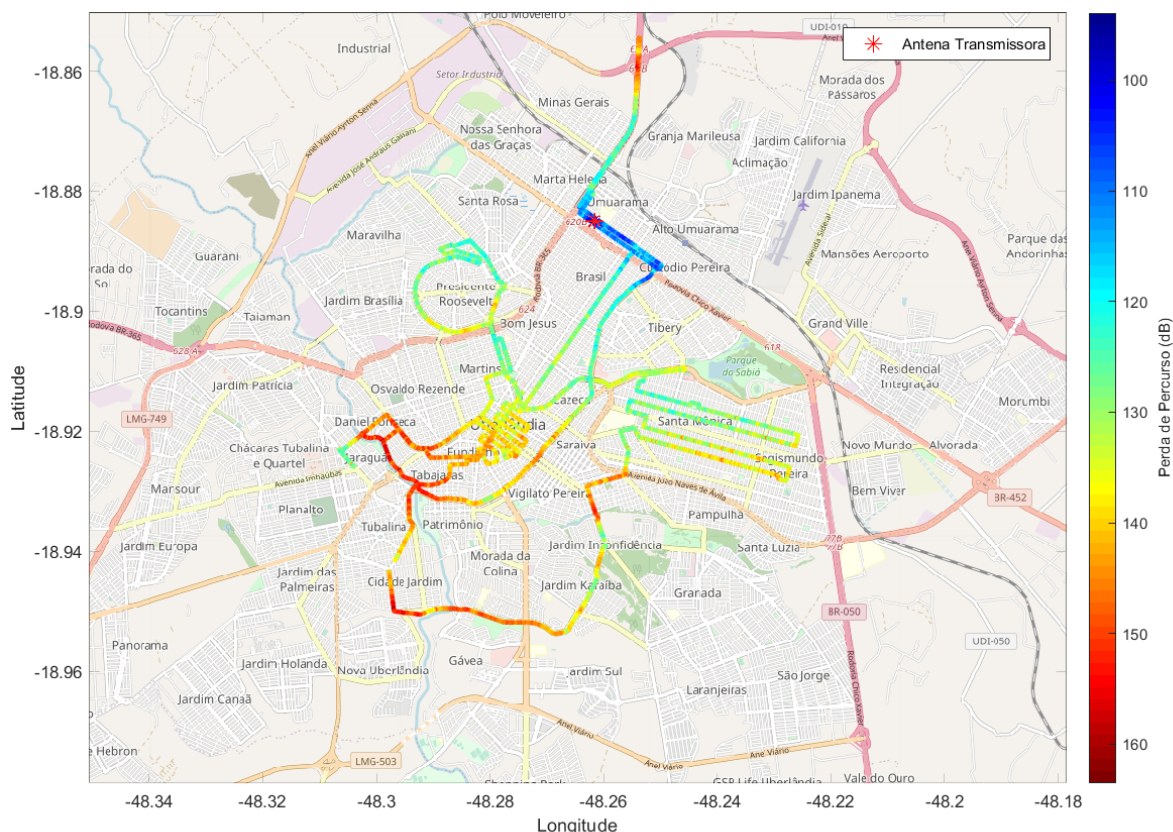
$-Gt_{(dB)}$ é o ganho da antena transmissora;

$-Pr_{(dB)}$ é a potência recebida pelo sistema de recepção;

$-Gr_{(dB)} = 0$ dB que é o ganho da antena receptora.

Com a consolidação do banco de dados produziu-se a Figura 6.5. Nela é apresentada o PL e cada localização do trajeto realizado. O *colorbar* na lateral indica a PL, com uma paleta disponível no *software* chamada *jet*. Essa paleta varia de tons de azul que indicam onde ocorreram as menores perdas de percurso, até os tons de vermelho onde ocorreram as maiores perdas de percurso.

Figura 6.5 – Mapa com a perda de percurso medida

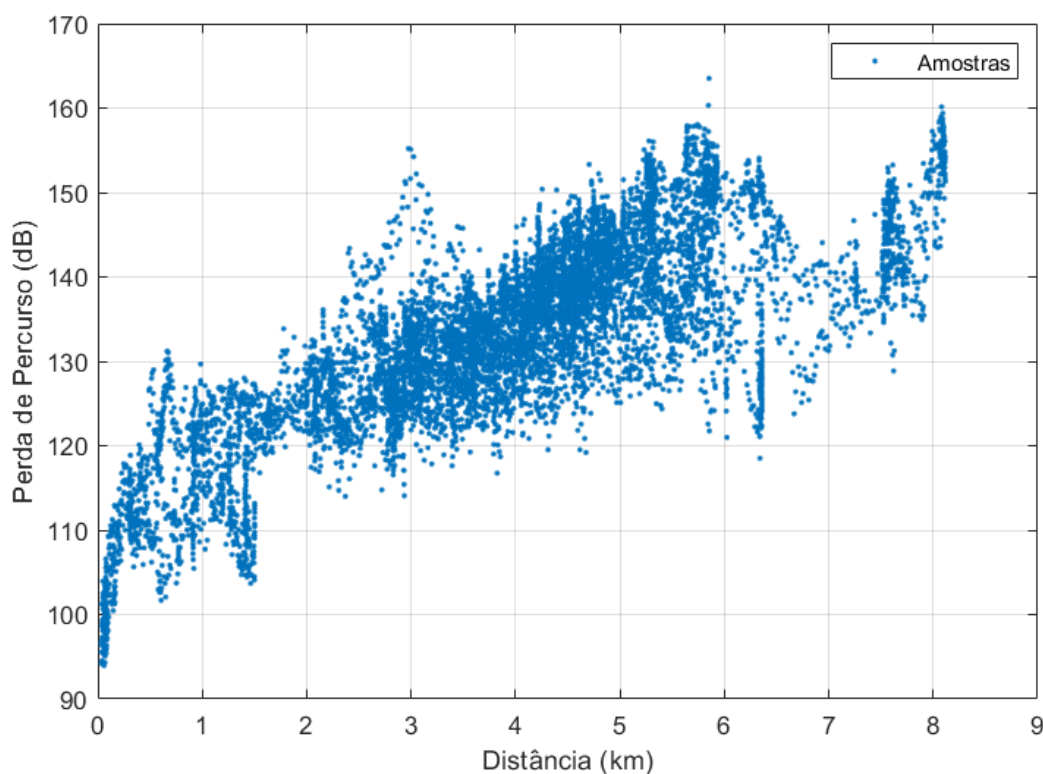


Fonte: Autor

A Figura 6.5 apresenta uma representação visual dos dados coletados e processados, onde os locais em azul indicam os menores valores para a PL, enquanto os pontos vermelhos indicam os maiores valores. A barra de cores na lateral fornece uma referência visual para interpretar a intensidade dos valores representados no mapa.

Ao observar o mapa, é possível notar que, quanto mais próximo da antena transmissora, o percurso tende a assumir uma coloração azul. Por outro lado, à medida que se afasta, o percurso tende a se tornar mais vermelho. No entanto, também é possível observar a ocorrência de perdas de percurso menores em regiões mais distantes, representadas por pontos amarelos, indicando que a PL é influenciada por outros fatores, como por exemplo altitude, multiplo percurso, além da distância.

A Figura 6.6 apresenta a relação entre a PL e a distância, nela está representada cada uma das amostras com relação a esses parâmetros.

Figura 6.6 – Perda de percurso *versus* a distância

Fonte: Autor

Na Figura 6.6, cada ponto representa um valor de PL para cada uma das 9.319 amostras. Observa-se que quanto mais distante o ponto, maior o valor do PL. Visualmente, nota-se que a maioria dos pontos está concentrada em 4 km de distância. Essa concentração é explicada pelos valores contidos na Tabela 6.2, onde a distância média dos pontos é de 3,93 km e o desvio padrão é de 1,8 km.

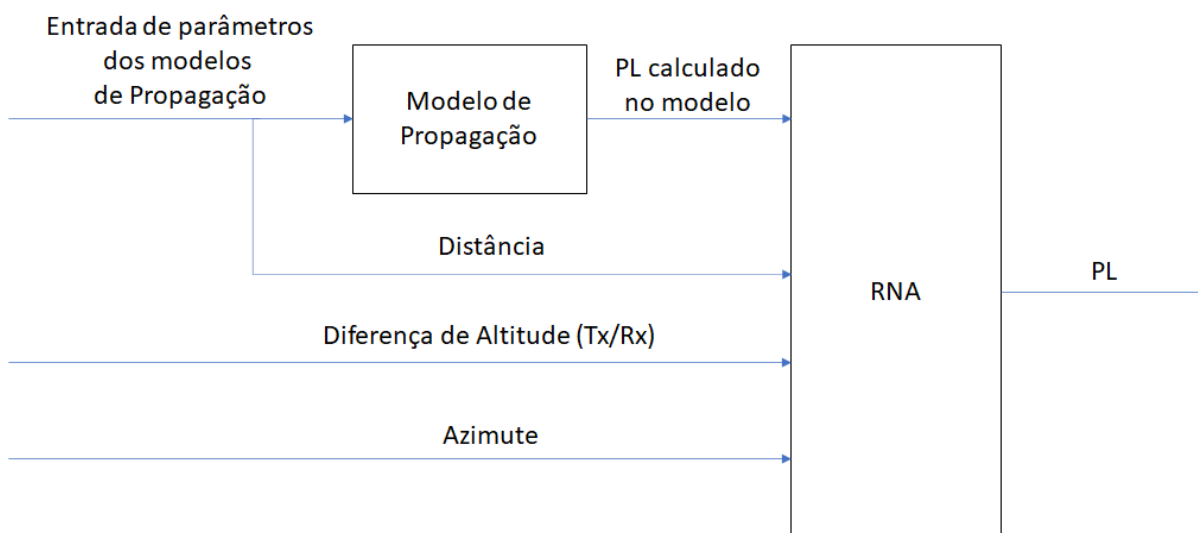
Assim como na figura anterior (Figura 6.5), há pontos mais próximos à antena transmissora com PL maior do que pontos mais distantes. Portanto, como explicado anteriormente, outros fatores impactam na PL.

6.4 DESENVOLVIMENTO DAS RNAs

A partir do conteúdo apresentado nos Capítulos 3, 4 e neste capítulo foram desenvolvidos os modelos híbridos (Modelo de Propagação - RNA). O desenvolvimento da RNA iniciou-se pela escolha dos parâmetros de entrada e saída (objetivo da rede). Sabe-se que outros fatores afetam a PL, mas neste trabalho são consideradas como entradas: perda calculada por um modelo de predição (Perda no Espaço Livre, Okumura-Hata, Ecc-33, Egli, Cost-231 e ITU P.1546-6), distância entre o sistema de transmissão e receptor, diferença de altitude entre antena e receptor e azimuth entre o sistema de transmissão e o receptor. Para cada modelo de perda de propagação

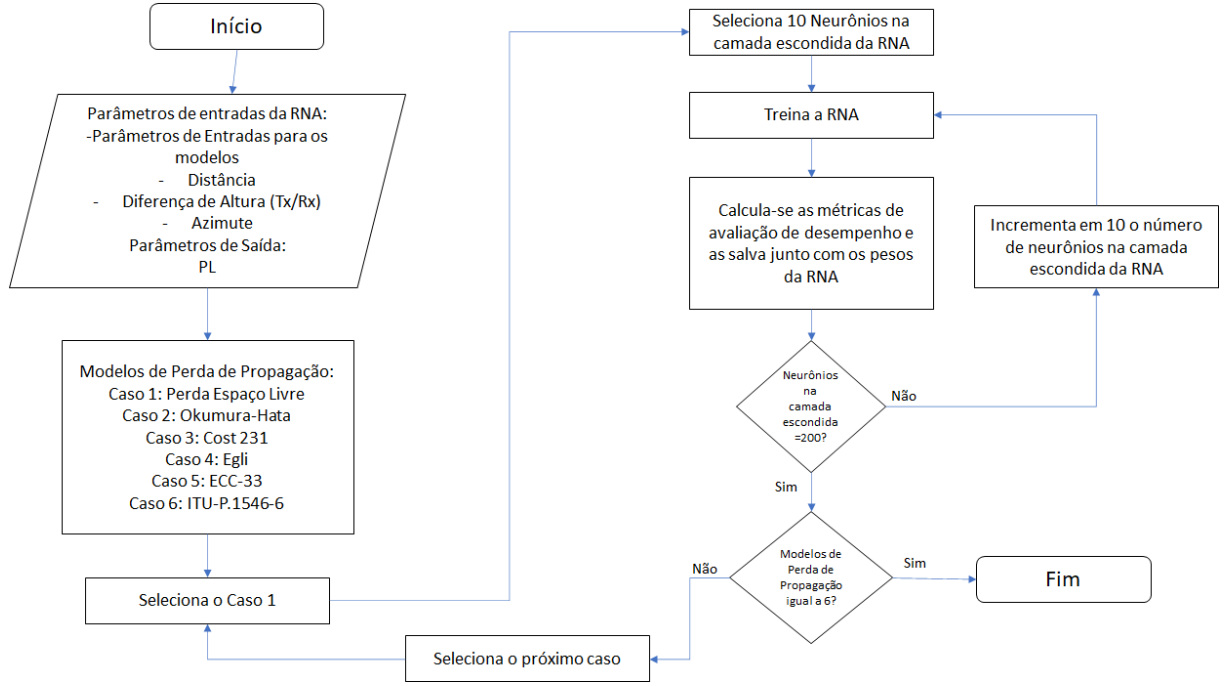
será realizado um treinamento diferente. Utiliza-se um parâmetro que representa a diferença de altitude da antena transmissora (Tx) e receptora (Rx), para que a rede se torne genérica a qualquer localidade que se desejar utilizar a RNA e não limitada somente às altitudes em que foram coletados os dados. A saída (objetivo) da rede são os dados na coluna 8 do novo banco de dados. A Figura 6.7 apresenta como foram configurados os parâmetros na RNA.

Figura 6.7 – Representação das entradas e saída da RNA



Fonte: Autor

Através do software Matlab e do seu *toolbox* Neural Net Fitting treinou-se a rede. Definiu-se a distribuição de amostras para treinamento, validação e testes, respectivamente, 50%, 25% e 25%. O próximo passo é a escolha do número de neurônios na camada escondida. A escolha foi realizada de forma empírica realizando o treinamento da RNA, com diferentes números de neurônios na camada escondida, iniciou-se com 10 neurônios e foi incrementando de 10 em 10 neurônios até chegar ao total de 200 neurônios na camada escondida. Como se tem seis modelos de propagação, esta etapa foi realizada seis vezes, uma para cada modelo de perda de propagação. O fluxograma da Figura 6.8 ilustra como foi configurada a rede.

Figura 6.8 – Fluxograma com os passos realizados

Fonte: Autor

A avaliação de desempenho da RNA em comparação com os modelos de predição foi medida usando as seguintes métricas estatísticas: EM, EQM, Erro Médio Absoluto (EMA), Erro RMS (E_{RMS}), σ e Coeficiente de Determinação (R^2), as fórmulas destes quatro últimos estão descritas abaixo através das equações (6.3)-(6.8). No Apêndice B está o código fonte utilizado para calcular a PL de cada modelo de propagação e a configuração e treinamento da rede neural conforme o fluxograma da 6.8.

$$EM = \frac{1}{N} \sum_{i=1}^N \left(PL_i^{measured} - PL_i^{predicted} \right) \quad (6.3)$$

$$EQM = \frac{1}{N} \sum_{i=1}^N \left(PL_i^{measured} - PL_i^{predicted} \right)^2 \quad (6.4)$$

$$EMA = \frac{1}{N} \sum_{i=1}^N \left| PL_i^{measured} - PL_i^{predicted} \right| \quad (6.5)$$

$$E_{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^N \left(PL_i^{measured} - PL_i^{predicted} \right)^2} \quad (6.6)$$

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N \left(|PL_i^{measured} - PL_i^{predicted}|^2 - \mu \right)} \quad (6.7)$$

Tem-se que o (μ) representa o erro de predição médio (dB).

$$R^2 = \frac{\sum_{i=1}^N (PL_i^{measured} - PL_i^{measured\ mean})^2 - \sum_{i=1}^N \left(PL_i^{predicted} - PL_i^{measured\ mean} \right)^2}{\sum_{i=1}^N (PL_i^{measured} - PL_i^{measured\ mean})^2} \quad (6.8)$$

6.5 PROCESSO DE OTIMIZAÇÃO POR ALGORITMO GENÉTICO E EVOLUÇÃO DIFERENCIAL

Com o objetivo de obter uma equação que estime o PL de forma mais precisa, propomos uma expressão analítica resultante da combinação linear das parcelas constituintes das fórmulas originais dos modelos de predição. Cada parcela será ponderada por um coeficiente x_i . Para determinar os valores ótimos desses coeficientes, utilizaremos AG, implementados na biblioteca 'geneticalgorithm' do Python, e ED, disponíveis na biblioteca 'scipy', também do Python. A otimização será realizada buscando minimizar a diferença entre os valores de PL obtidos pela equação proposta e os valores observados nos dados práticos.

Portanto a fórmula de Okumura-Hata ficou da seguinte forma:

$$PL(dB) = x_0 69.55 + x_1 26.16 \log(f_c) - x_2 13.83 \log(h_t) \\ (x_3 44.9 - x_4 6.55 \log(h_r)) \log(d) - A(h_r) \quad (6.9)$$

$$A(h_r) = (x_5 1.1 \log(f_c) - x_6 0.7) h_r \\ - (x_7 1.56 \log((f_c) - x_8 0.8)) \quad (6.10)$$

Também foi adicionado o parâmetro altitude multiplicada por uma variável a ser otimizada.

$$PL(dB) = x_0 69.55 + x_1 26.16 \log(f_c) - x_2 13.83 \log(h_t) \\ (x_3 44.9 - x_4 6.55 \log(h_r)) \log(d) - A(h_r) \quad (6.11)$$

$$A(h_r) = (x_5 1.1 \log(f_c) - x_6 0.7) h_r \\ - (x_7 1.56 \log((f_c) - x_8 0.8)) \\ + x_9 \text{altitude} \quad (6.12)$$

De forma análoga as variáveis x_i foram introduzidas nos outros modelos de perda de

propagação e suas fórmulas ficaram da seguinte maneira.

ECC-33:

$$PL(dB) = A_{fs} + A_{bm} - G_b - G_r \quad (6.13)$$

$$A_{fs} = \mathbf{x}_0 92.4 + \mathbf{x}_1 20 \log(d) + \mathbf{x}_2 20 \log(f) \quad (6.14)$$

$$A_{bm} = \mathbf{x}_3 20.41 + \mathbf{x}_4 9.83 \log(d) + \mathbf{x}_5 7.89 \log(f) + \mathbf{x}_6 9.56 (\log(f))^2 \quad (6.15)$$

$$G_b = \log\left(\frac{h_t}{200}\right) (\mathbf{x}_7 13.958 + \mathbf{x}_8 5.8 \log(d))^2 \quad (6.16)$$

$$G_r = \mathbf{x}_9 0.759 h_r - \mathbf{x}_{10} 1.862 \quad (6.17)$$

ECC-33 com altitude:

$$PL(dB) = A_{fs} + A_{bm} - G_b - G_r + \mathbf{x}_{11} \text{altitude} \quad (6.18)$$

$$A_{fs} = \mathbf{x}_0 92.4 + \mathbf{x}_1 20 \log(d) + \mathbf{x}_2 20 \log(f) \quad (6.19)$$

$$A_{bm} = \mathbf{x}_3 20.41 + \mathbf{x}_4 9.83 \log(d) + \mathbf{x}_5 7.89 \log(f) + \mathbf{x}_6 9.56 (\log(f))^2 \quad (6.20)$$

$$G_b = \log\left(\frac{h_t}{200}\right) (\mathbf{x}_7 13.958 + \mathbf{x}_8 5.8 \log(d))^2 \quad (6.21)$$

$$G_r = \mathbf{x}_9 0.759 h_r - \mathbf{x}_{10} 1.862 \quad (6.22)$$

Egli:

$$PL(dB) = \mathbf{x}_0 20 \log(f) + p_o + \mathbf{x}_4 76.3 \quad (6.23)$$

$$p_o = \mathbf{x}_1 40 \log(d) - \mathbf{x}_2 20 \log(h_t) - \mathbf{x}_3 10 \log(h_r) \quad (6.24)$$

Egli com altitude:

$$PL(dB) = \mathbf{x}_0 20 \log(f) + p_o + \mathbf{x}_4 76.3 + \mathbf{x}_5 \text{altitude} \quad (6.25)$$

$$p_o = \mathbf{x}_1 40 \log(d) - \mathbf{x}_2 20 \log(h_t) - \mathbf{x}_3 10 \log(h_r) \quad (6.26)$$

Cost 231 Hata Model:

$$PL(dB) = \mathbf{x}_0 46.3 + \mathbf{x}_1 33.9 \log(f) - \mathbf{x}_2 13.82 \log(h_t) - A(h_r) - \mathbf{x}_3 44.9 - \mathbf{x}_4 6.55 \log(h_t) \log(d) \quad (6.27)$$

$$A(h_r) = (\mathbf{x}_5 1.1 \log(f_c) - \mathbf{x}_6 0.7) h_r - (\mathbf{x}_7 1.56 \log((f_c) - \mathbf{x}_8 0.8)) \quad (6.28)$$

Cost 231 Hata Model com altitude:

$$PL(dB) = \mathbf{x}_0 46.3 + \mathbf{x}_1 33.9 \log(f) - \mathbf{x}_2 13.82 \log(h_t) - A(h_r) - \mathbf{x}_3 44.9 - \mathbf{x}_4 6.55 \log(h_t) \log(d) + \mathbf{x}_9 \text{altitude} \quad (6.29)$$

$$A(h_r) = (\mathbf{x}_5 1.1 \log(f_c) - \mathbf{x}_6 0.7) h_r - (\mathbf{x}_7 1.56 \log((f_c) - \mathbf{x}_8 0.8)) \quad (6.30)$$

6.5.1 CONFIGURAÇÃO DOS PARÂMETROS DO ALGORITMO GENÉTICO

Uma vez definidas as variáveis a serem otimizadas, o próximo passo foi especificar os parâmetros do AG descritos abaixo:

- Número Máximo de Iterações: 10,000;
- Tamanho da população: 50;
- Probabilidade de Mutação: 0.25;
- Taxa de Elitismo: 0.01;
- Probabilidade de crossover: 0.8;

- Porção parental: 0.5;
- Tipo de crossover: dois pontos;
- Número máximo de iterações sem melhoria: 3,000

6.5.2 CONFIGURAÇÃO DOS PARÂMETROS DA EVOLUÇÃO DIFERENCIAL

Da mesma maneira que o GA, estão especificados a seguir os parâmetros do ED:

- Número Máximo de Iterações: 10,000;
- Tamanho da população: 50;
- Probabilidade de mutação: 0.25;
- Probabilidade de recombinação: 0.8

Tanto para o AG quanto para ED definimos as variáveis podem ser representadas por um número real com limites: inferior com valor de -100 e superior de 100. Também se estabeleceu a função objetiva como o erro quadrático médio (EQM) entre os valores do PL (coluna 8) e o valor da previsão do PL otimizado pela geração, portanto, quanto menor o valor do EQM, melhor será o PL previsto.

6.6 CONSIDERAÇÕES FINAIS

Neste capítulo, foi apresentada a metodologia utilizada neste trabalho, abrangendo diversas etapas fundamentais para a condução do estudo. Inicialmente, a Seção 6.1 descreveu o sistema de transmissão, detalhando sua estrutura e componentes essenciais. Em seguida, a Seção 6.2 abordou o sistema de recepção, explicando as etapas envolvidas na captação e processamento inicial do sinal. Na Seção 6.3, foi discutido o processamento dos dados, enfatizando os métodos aplicados para extrair informações relevantes. A Seção 6.4 apresentou o desenvolvimento das RNAs, destacando as arquiteturas e estratégias implementadas. Por fim, a Seção 6.5 detalhou o processo de otimização realizado por meio de AGs e ED, com subseções específicas para a configuração dos parâmetros de cada abordagem (6.5.1 e 6.5.2). Essas etapas foram integradas de forma a proporcionar uma base sólida para os resultados apresentados nos capítulos subsequentes.

RESULTADOS DA PREDIÇÃO OBTIDA PELA RNA

Resumo do capítulo

Neste capítulo, são apresentados os resultados da aplicação de RNAs nos modelos de propagação. Diferentes modelos híbridos são discutidos, combinando os modelos teóricos de propagação com as RNAs. Os resultados são analisados e comparados com estudos anteriores para avaliar sua eficácia e precisão.

7.1 MODELO HÍBRIDO PERDA NO ESPAÇO LIVRE E RNA

Na Tabela 7.1, estão apresentados os testes realizados para o modelo proposto nesta seção. Pode-se notar que, para este caso, o melhor desempenho da rede ocorre quando são utilizados 140 neurônios na camada escondida. Para um número maior de neurônios, há um afastamento da melhor configuração testada comprometendo a capacidade da rede de generalização.

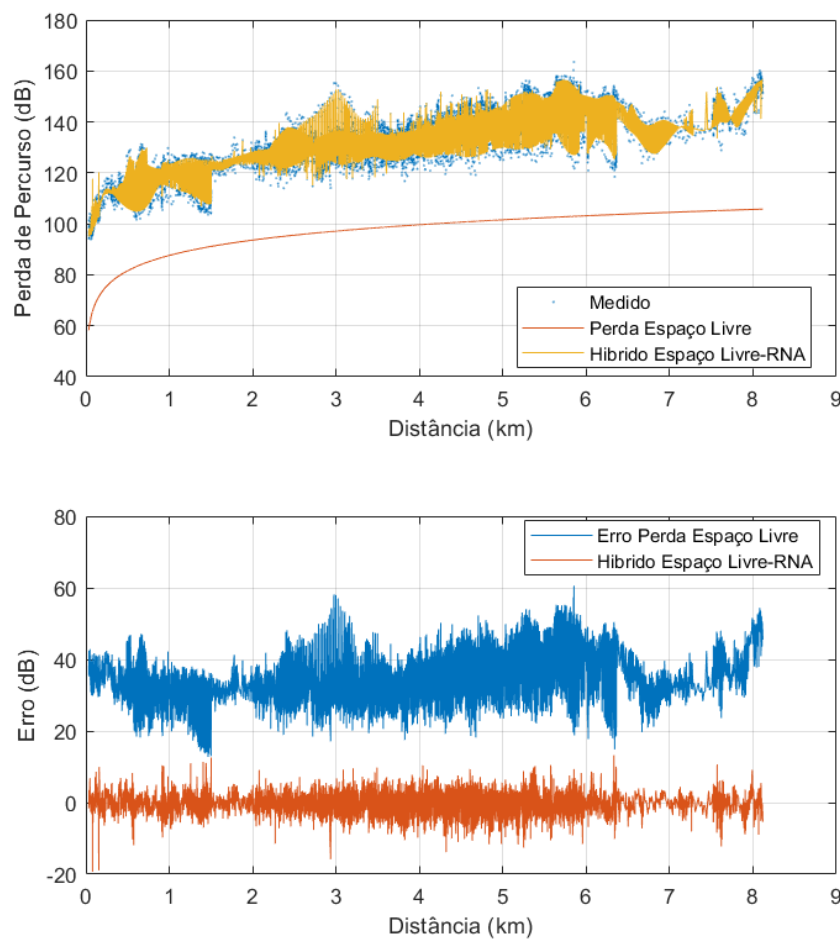
Tabela 7.1 – EQM para o treinamento variando neurônios na camada escondida para modelo híbrido Perda no Espaço Livre-RNA

Neurônios na Camada Escondida	EQM Treinamento	EQM Validação	EQM Teste	EQM Total
10	15,34407	16,88223	16,5497	16,03009
20	12,89332	13,33775	14,81522	13,48496
30	11,1578	12,36998	12,41623	11,77552
40	10,14685	10,68765	12,03288	10,75362
50	8,893215	9,977358	10,33334	9,52435
60	8,730663	11,19761	11,15365	9,953279
70	8,333343	10,387	10,87478	9,482241
80	8,409178	10,43659	10,32679	9,39554
90	8,641175	9,97127	11,76431	9,754601
100	8,411674	9,634373	9,506704	8,991168
110	9,32258	10,67899	10,23843	9,890708
120	8,117214	10,45979	10,02085	9,178882
130	8,329281	10,42387	10,11246	9,298826
140	8,115844	9,935812	9,758028	8,981475
150	9,91157	10,76413	11,5747	10,54056
160	8,172007	9,874993	10,38873	9,15204
170	8,591181	9,629043	10,6247	9,359108
180	8,322654	10,4126	11,4934	9,637969
190	9,232608	10,06922	9,885196	9,604949
200	8,623351	10,81692	10,79863	9,715682

A Figura 7.1a apresenta graficamente o comportamento da Perda no Espaço Livre e o Modelo Híbrido Perda no Espaço Livre-RNA, na Figura 7.1b mostra o erro que ocorreu e o que

foi medido e os dois tipos de perdas supracitados. Ao realizar uma análise visual, observa-se que o erro do modelo híbrido Perda Espaço Livre-RNA oscilou em torno do eixo das abscissas, enquanto o modelo Perda Espaço Livre ficou todo acima desse eixo, ou seja, o erro para este modelo é totalmente positivo. Essa análise visual pode ser comprovada através dos parâmetros estatísticos da Tabela 7.2 discutida abaixo.

Figura 7.1 – Comparação entre perda de percurso, Perda no Espaço Livre e Perda Espaço Livre-RNA



Fonte: Autor

A Tabela 7.2 mostra um comparativo entre o modelo Perda no Espaço Livre e o modelo Híbrido Perda no espaço livre-RNA através dos parâmetros estatísticos EM, EMA, EQM, E_{RMS} , σ e R^2 .

Nota-se que para todos os erros, o modelo híbrido teve um desempenho melhor que o modelo Perda no Espaço Livre, houve melhor desempenho também no desvio padrão e no coeficiente de determinação.

Tabela 7.2 – Comparação entre os erros obtidos pelos modelos Perda no Espaço Livre e o híbrido Perda no Espaço Livre-RNA

Modelo	EM	EMA	EQM	E_{RMS}	σ	R^2
Perda no Espaço Livre	35,58362	35,58362	1318,358	36,30921	7,22287	0,62622
Perda Espaço Livre-RNA	0,028813	2,323913724	8,981475	2,996911	1,892428	0,930576

7.2 MODELO HÍBRIDO COST-231 E RNA

O mesmo procedimento realizado para a Perda no Espaço Livre foi realizado com o modelo Cost-231, e os resultados encontrados estão na Tabela 7.3. Para este caso, o treinamento que apresentou melhor resultado foi com 130 neurônios na camada escondida.

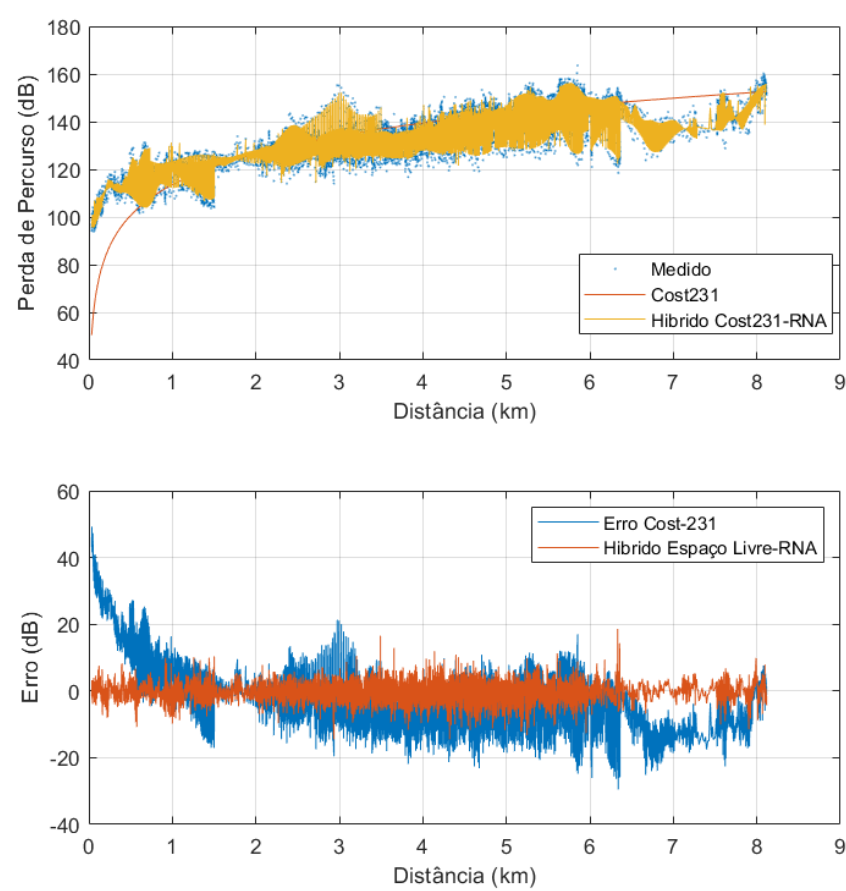
Tabela 7.3 – EQM para o treinamento variando neurônios na camada escondida para modelo híbrido Cost 231-RNA

Neurônios na Camada Escondida	EQM Treinamento	EQM Validação	EQM Teste	EQM Total
10	14,92738	14,88665	15,3438	15,02131
20	11,18032	12,56527	12,18962	11,77895
30	10,90617	11,26241	11,69448	11,19234
40	9,922013	10,73641	11,22542	10,45152
50	8,862232	9,867476	10,37482	9,491759
60	8,523511	9,933268	10,07141	9,263005
70	8,554312	9,719509	10,90358	9,433023
80	8,882344	9,984787	10,29025	9,509999
90	9,248075	10,45571	11,31071	10,06573
100	10,5714	11,77857	11,62418	11,13645
110	8,565887	10,01597	11,5222	9,667604
120	7,813282	10,02303	10,06179	8,927966
130	8,083167	9,486721	9,847066	8,875116
140	9,243947	10,69161	10,47608	9,913968
150	8,645931	10,37801	10,02969	9,424973
160	8,942468	9,879129	10,2259	9,497552
170	8,650641	11,15442	10,65853	9,778678
180	9,318368	11,45101	10,74384	10,20799
190	9,39685	10,74039	10,14142	9,918935
200	9,61982	10,87951	10,70696	10,20659

A Figura 7.2a ilustra graficamente o comportamento do modelo Cost-231 e o Modelo

Híbrido Cost 231-RNA, na Figura 7.2b mostra o erro que ocorreu e o que foi medido e os dois tipos de modelos. Observa-se que o erro do modelo híbrido Cost-231-RNA oscilou em torno das abscissas, enquanto o modelo Cost 231, nos primeiros 2 quilômetros, o erro foi positivo, ou seja, acima do eixo das abscissas, e depois do 6º quilômetro se tornou predominantemente negativo.

Figura 7.2 – Comparação entre perda de percurso, Cost 231 e Cost231-RNA



Fonte: Autor

A Tabela 7.2 mostra um comparativo entre o modelo Cost 231 e o modelo Híbrido Cost 231-RNA através dos parâmetros estatísticos EM, EMA, EQM, E_{RMS} , σ e R^2 .

Tabela 7.4 – Comparação entre os erros obtidos pelos modelos Cost-231 e o híbrido Cost 231-RNA

Modelo	EM	EMA	EQM	E_{RMS}	σ	R^2
Cost-231	-2,05304	6,870461	90,10529	9,492381	6,550317	0,62622
Cost 231-RNA	-0,01245	2,314197	8,875116	2,979113	1,876162	0,931399

Nota-se que para todos os erros, o modelo híbrido teve um desempenho melhor que o modelo Perda no Espaço Livre, houve melhor desempenho também no desvio padrão e no

coeficiente de determinação.

7.3 MODELO HÍBRIDO EGLI E RNA

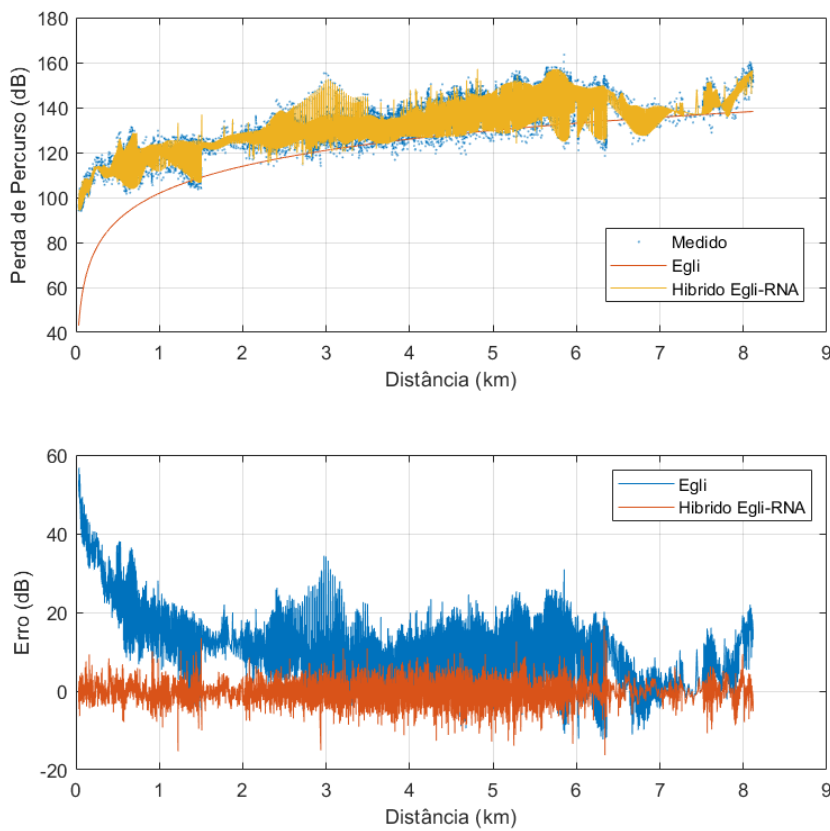
O mesmo procedimento realizado para os modelos anteriores foi realizado com o modelo Egli, e os resultados encontrados estão na Tabela 7.5. Para este caso, o treinamento que apresentou melhor resultado foi com 130 neurônios na camada escondida.

Tabela 7.5 – EQM para o treinamento variando neurônios na camada escondida para modelo híbrido Egli-RNA

Neurônios na Camada Escondida	EQM Treinamento	EQM Validação	EQM Teste	EQM Total
10	16,53764	16,00756	16,3286	16,35284
20	21,29575	23,15176	21,87189	21,90386
30	10,26233	10,93787	11,11426	10,64424
40	10,09458	11,37379	10,28592	10,46226
50	8,920313	10,65202	10,80477	9,82445
60	9,757955	11,23341	10,97174	10,43034
70	8,627413	9,715918	11,06253	9,508412
80	8,466008	10,48547	10,21051	9,4071
90	8,61215	10,61379	10,65174	9,622567
100	8,31842	10,60421	10,46739	9,427229
110	8,521847	10,33236	10,13404	9,377616
120	7,460626	10,426	10,10841	8,864067
130	7,946927	9,275981	9,708869	8,719759
140	8,249165	9,982125	10,94334	9,356068
150	7,843893	10,03267	9,437659	8,78963
160	7,969621	10,02606	9,888332	8,963514
170	7,83634	10,16693	10,31856	9,039671
180	7,494512	9,845067	10,13949	8,74353
190	8,028151	10,51112	9,460827	9,007167
200	8,551902	10,15126	10,43047	9,421479

A Figura 7.3a apresenta graficamente o comportamento do modelo Cost-231 e o Modelo Híbrido Cost 231-RNA, na Figura 7.3b mostra o erro que ocorreu, e o que foi medido e, os dois tipos de modelos. Observa-se que o erro do modelo híbrido Egli-231-RNA variou em torno das abscissas, enquanto no modelo Egli, o erro é predominantemente positivo.

Figura 7.3 – Comparação entre perda de percurso, Egli e Egli-RNA



Fonte: Autor

A Tabela 7.6 mostra um comparativo entre o modelo Egli e o modelo Híbrido Egli-RNA através dos parâmetros estatísticos EM, EMA, EQM, E_{RMS} , σ e R^2 .

Tabela 7.6 – Comparação entre os erros obtidos pelos modelos Egli e o híbrido Egli-RNA

Modelo	EM	EMA	EQM	E_{RMS}	σ	R^2
Egli	11,14331	11,60793	198,4842	14,08844	7,984167	0,62622
Egli-RNA	0,013534	2,282026	8,719759	2,952924	1,874164	0,932625

Nota-se que para todos os erros o modelo híbrido teve um desempenho melhor que o modelo Perda no Espaço Livre, houve melhor também no desvio padrão e no coeficiente de determinação.

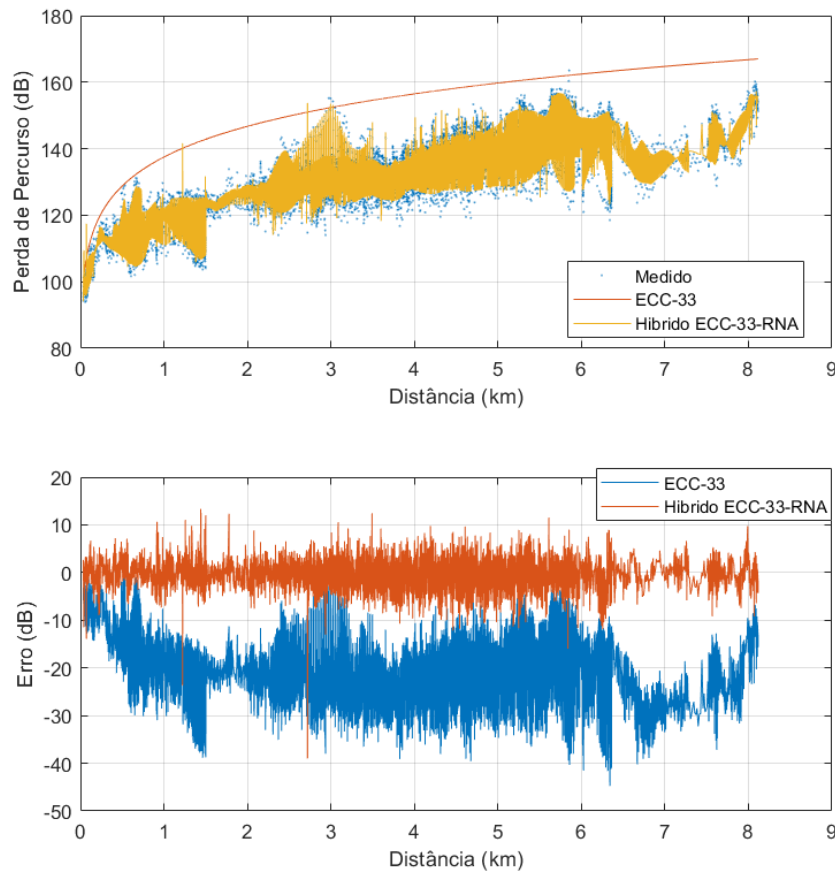
7.4 MODELO HÍBRIDO ECC-33 E RNA

Foi realizado para o modelo ECC-33 o mesmo procedimento para os modelos anteriores, conforme mostra a Tabela 7.7. Para este caso, o treinamento que apresentou melhor resultado foi com 140 neurônios na camada escondida.

Tabela 7.7 – EQM para o treinamento variando neurônios na camada escondida para modelo híbrido ECC-33-RNA

Neurônios na Camada Escondida	EQM Treinamento	EQM Validação	EQM Teste	EQM Total
10	15,14625	14,30539	15,06709	14,91622
20	11,41724	12,26792	12,80359	11,97656
30	10,08485	11,46642	11,67809	10,82863
40	10,22773	11,40138	11,55498	10,85302
50	12,60804	13,39252	15,43979	13,51219
60	10,38727	11,3132	10,76709	10,71374
70	8,991932	9,695967	10,00536	9,421344
80	8,449884	9,936924	9,587837	9,106203
90	8,7404	10,03929	10,17844	9,424705
100	8,508045	9,037868	10,37494	9,107288
110	10,02041	11,57138	10,79937	10,60295
120	9,750767	11,24864	11,49763	10,56204
130	8,137337	9,919633	10,65346	9,212056
140	8,009257	9,679145	9,891358	8,89735
150	7,890458	10,0185	9,787398	8,896811
160	9,27064	11,67163	11,27322	10,37165
170	9,003388	10,32335	11,01694	9,836856
180	8,684522	9,111726	10,43394	9,228735
190	8,026956	10,41605	9,931748	9,100542
200	8,514718	10,06092	11,37042	9,615311

A Figura 7.4a apresenta graficamente o comportamento do modelo ECC-33 e o Modelo Híbrido ECC-33-RNA, na Figura 7.4b mostra o erro que ocorreu e, o que foi medido e, os dois tipos de modelos. Observa-se que o erro do modelo híbrido ECC33-RNA ficou em variou em torno das abcissas, enquanto o modelo ECC-33 o erro é predominantemente negativo.

Figura 7.4 – Comparação entre perda de percurso, ECC-33 e ECC33-RNA

Fonte: Autor

A Tabela 7.2 mostra um comparativo entre o modelo ECC-33 e o modelo Híbrido ECC33-RNA através dos parâmetros estatísticos EM, EMA, EQM, E_{RMS} , σ e R^2 .

Tabela 7.8 – Comparação entre o ECC-33 e o híbrido ECC-33-RNA

Modelo	EM	EMA	EQM	E_{RMS}	σ	R^2
ECC-33	-20,3998	20,40204	464,5238	21,55281	6,948788	0,644492
ECC-33-RNA	-0,06102	2,287072	8,89735	2,982843	1,914953	0,931296

Nota-se que para todos os erros, o modelo híbrido teve um desempenho melhor que o modelo Perda no Espaço Livre, houve melhor desempenho também no desvio padrão e no coeficiente de determinação.

7.5 MODELO HÍBRIDO OKUMURA-HATA E RNA

A Tabela 7.9 mostra os resultados do treinamento do modelo híbrido Okumura-Hata-RNA. Neste caso, a RNA com 90 neurônios na camada escondida foi a que apresentou melhor

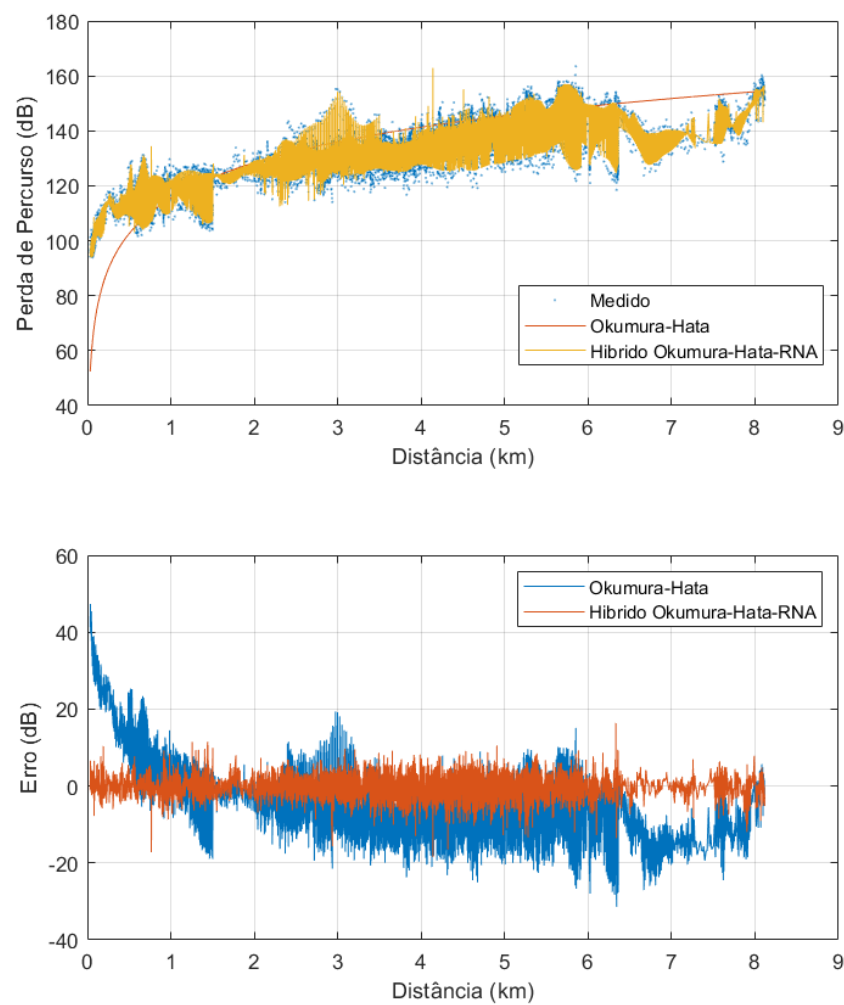
desempenho.

Tabela 7.9 – EQM para o treinamento variando neurônios na camada escondida para modelo híbrido Okumura-Hata-RNA

Neurônios na Camada Escondida	EQM Treinamento	EQM Validação	EQM Teste	EQM Total
10	14,67095	15,25992	14,66056	14,81561
20	12,1636	12,96467	12,77523	12,51681
30	10,15597	10,87239	12,33821	10,88071
40	10,36188	11,27839	11,62243	10,9062
50	24,60497	26,48245	25,62069	25,32835
60	9,493217	10,33339	10,49626	9,95407
70	10,14717	11,09814	11,375	10,69193
80	9,368313	10,56081	11,50019	10,1995
90	7,635885	9,625106	9,540335	8,609407
100	8,982949	10,32476	10,18859	9,619881
110	7,525263	9,078616	11,64845	8,94455
120	8,884113	10,50478	10,99184	9,816311
130	8,959337	10,5478	10,39466	9,715364
140	8,621957	10,11615	10,31173	9,418032
150	8,576537	9,940007	10,28178	9,343796
160	8,062229	9,560466	11,15343	9,209712
170	8,273618	10,32725	10,71131	9,39657
180	8,057165	9,478286	9,291844	8,721186
190	8,726997	11,0914	11,01326	9,889788
200	8,552151	9,966187	10,5087	9,394887

A Figura 7.5a apresenta o graficamente o comportamento do modelo Okumura-Hata e o Modelo Híbrido Okumura-Hata-RNA, na Figura 7.5b mostra o erro que ocorreu, e o que foi medido e, os dois tipos de modelos. Observa-se que o erro do modelo híbrido Okumura-Hata-RNA variou em torno do eixo das abscissas, enquanto o modelo Okumura-Hata, nos primeiros 2 quilômetros, o erro foi positivo, ou seja, acima do eixo das abscissas, e após esta distância tornou predominantemente negativo. Observa-se que o erro do modelo híbrido Okumura-Hata-RNA variou em torno do eixo das abscissas, enquanto o modelo Okumura-Hata, nos primeiros 2 quilômetros, o erro foi positivo, ou seja, acima do eixo das abscissas, e após esta distância tornou predominantemente negativo.

Figura 7.5 – Comparação entre perda de percurso, Okumura Hata e Okumura Hata-RNA



Fonte: Autor

A Tabela 7.10 mostra um comparativo entre o modelo Okumura-Hata e o modelo Híbrido Okumura-Hata-RNA através dos parâmetros estatísticos EM, EMA, EQM, E_{RMS} , σ e R^2 .

Tabela 7.10 – Comparação entre os erros obtidos pelos modelos Okumura-Hata e o híbrido Okumura-Hata-RNA

Modelo	EM	EMA	EQM	E_{RMS}	σ	R^2
Okumura-Hata	-3,95957	7,66887	101,5685	10,07812	6,539229	0,62622
Okumura-Hata-RNA	-0,00133	2,262488	8,609407	2,934179	1,868404	0,933515

7.6 MODELO HÍBRIDO ITU-P.1546-6 E RNA

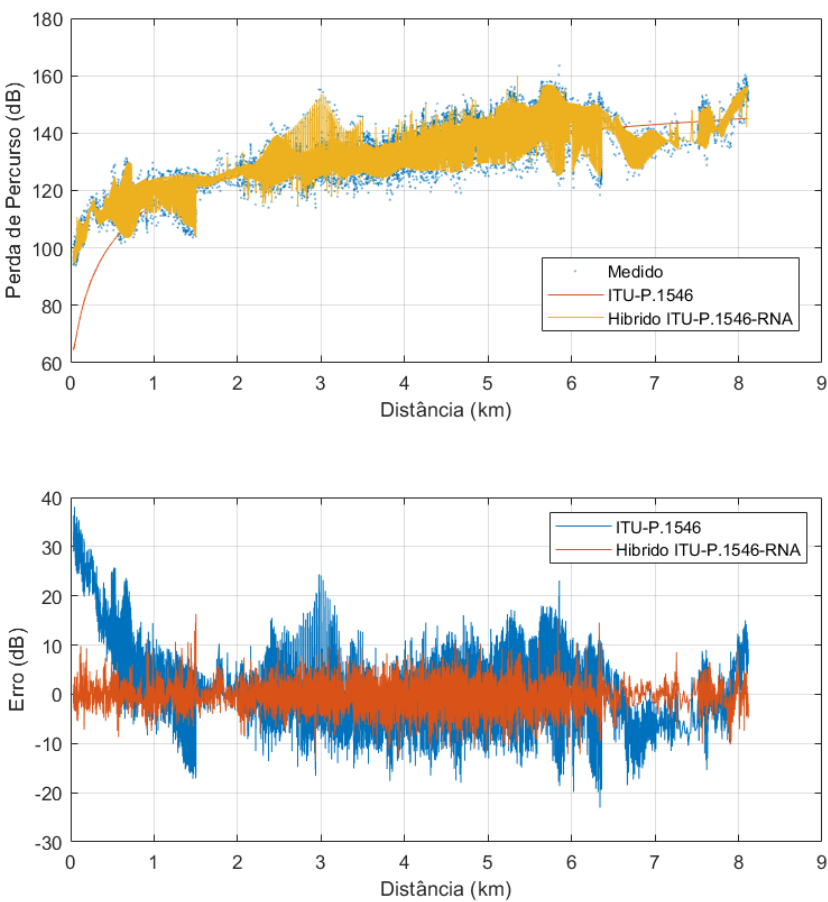
O último modelo híbrido analisado foi o da Recomendação ITU-P.1546-6-RNA, na Tabela 7.11 mostra os resultados dos treinamentos realizados variando o número de neurônios na camada escondida. Para este caso, a RNA com 170 neurônios na camada escondida foi a que apresentou melhor desempenho.

Tabela 7.11 – EQM para o treinamento variando neurônios na camada escondida para modelo híbrido ITU-P.1546-6-RNA

Neurônios na Camada Escondida	EQM Treinamento	EQM Validação	EQM Teste	EQM Total
10	15,48945	15,60146	15,25865	15,45975
20	13,2555	14,17259	14,01346	13,6743
30	11,14704	12,09943	11,71276	11,52661
40	9,717545	10,55236	11,20969	10,29935
50	9,231229	10,08348	10,82647	9,843168
60	10,21029	11,09844	11,11121	10,6576
70	9,766522	11,01094	11,37467	10,47974
80	10,11652	11,18268	11,29195	10,67698
90	8,007009	9,442617	10,61295	9,017505
100	8,734739	9,560162	10,57684	9,401691
110	9,416764	11,04438	10,64637	10,13115
120	9,046652	10,28571	10,22008	9,649838
130	8,406438	10,68089	10,10033	9,39863
140	7,621825	10,3737	9,867189	8,871269
150	10,9378	11,27765	11,27992	11,10831
160	9,183733	11,7891	10,58193	10,18473
170	7,787871	9,426699	9,532991	8,633949
180	8,388024	9,706454	11,19546	9,419602
190	8,845436	11,50812	10,09824	9,824414
200	7,960707	10,17276	10,98062	9,26884

A Figura 7.6a apresenta graficamente o comportamento do modelo ITU P.1546-6 e o Modelo Híbrido ITU P.1546-6-RNA, na Figura 7.6b mostra o erro que ocorreu e o que foi medido e os dois tipos de modelos. Observa-se que o erro do modelo híbrido P.1546-6-RNA variou em torno das abcissas. O Modelo P.1546-6 possui um erro maior no primeiro quilômetro.

Figura 7.6 – Comparação entre perda de percurso, ITU-P.1546-6 e ITU-P.1546-6-RNA



Fonte: Autor

A Tabela 7.12 mostra um comparativo entre o modelo ITU-P.1546-6 e o modelo Híbrido ITU-P.1546-6-RNA através dos parâmetros estatísticos: EM, EMA, EQM, E_{RMS} , σ e R^2 .

Tabela 7.12 – Comparação entre os erros obtidos pelos modelos ITU-P.1546-6 e o híbrido ITU-P.1546-6-RNA

Modelo	EM	EMA	EQM	E_{RMS}	σ	R^2
ITU-P.1546.6	1,732532	6,154002	69,12468	8,314125	5,590733	0,602039
ITU-P.1546.6-RNA	0,04033	2,271354	8,633949	2,938358	1,864208	0,933312

7.7 COMPARAÇÃO COM OUTROS ESTUDOS

Na Tabela 7.13 estão representadas os resultados de EM, EMA, EQM, E_{RMS} , σ , R^2 encontrados em trabalhos semelhantes. Logo abaixo, é apresentada a Tabela 7.14 com os resultados encontrados em todos os modelos híbridos desenvolvidos neste trabalho.

Tabela 7.13 – Resultados obtidos em outros estudos

Trabalho	EM	EMA	EQM	E_{RMS}	σ	R^2
Popoola et al. (2018)	-	4,21	30,99	5,56	0,86	
Piacentini et al. (2011)	-	-	53,25	-	-	-
Eichie et al. (2017b)	-	-	-	1,22	-	-
Popescu et al. (2005)	3,67	-	-	3,71	2,46	-
Cerri et al. (2004)	-	0,71	-	-	1,0	-
Sotiroudis et al. (2015)	-	2,74	-	3,54	-	-
Sotiroudis et al. (2013)	-	1,84	-	2,41	-	-
Kalakh et al. (2012)	-	-	1,04	-	-	
Popescu et al. (2001)	1,65	-	-	2,13	1,35	-
Ostlin et al. (2010)	0	-	-		7,00	-
Liu et al. (2017)	-	-	-	0,28		-
Zaarour et al. (2012)	-	-	1,76			-
Anitzine et al. (2012)	-	-	1,77	2,52	1,79	-
Zineb et al. (2015)	0,75	-	-	-	5,22	0,96
Pérez et al. (2017)	-	-	-	0,57	-	-
Eichie et al. (2017a)	-	-	0,06	-	-	-
Benmus et al. (2015)	-	-	-	3,0	-	-
Bhuvaneshwari et al. (2016)	-	-	-	-	11,37	0,99
Angeles et al. (2014)	-	-	94,58	-	-	-
Ferreira et al. (2016)	-	-	-	-	2,86	-
Ayadi et al. (2017)	-	0,24	-	-	6,85	0,85
Santos (2021)	-1,26	-	-	3,72	2,88	-

Tabela 7.14 – Resultados obtidos neste trabalho

	EM	EMA	EQM	E_{RMS}	σ	R^2
ITU-P.1546.6-RNA	0,040	2,27	8,63	2,94	1,87	0,93
Okumura-Hata-RNA	-0,01	2,26	8,61	2,93	1,87	0,93
ECC-33-RNA	-0,06	2,29	8,90	2,98	1,91	0,93
Egli-RNA	0,01	2,28	8,72	2,95	1,87	0,93
Cost 231-RNA	-0,01	2,31	8,88	2,98	1,88	0,93
Perda Espaço Livre-RNA	0,029	2,32	8,99	2,99	1,89	0,93

O intuito de colocar as duas tabelas em sequência é possibilitar a realização de um comparativo entre este e outros trabalhos. Todavia, como cada estudo foi realizado em uma localidade diferente e com características geográficas distintas, o fato de um trabalho obter uma métrica melhor que o outro pode não significar que este trabalho seja superior. Outro fator determinante é como foram feitas as coletas e se houve algum pré-processamento dos dados coletados, uma vez que é sabido que, durante o procedimento de coleta, alguns ruídos podem contaminar os dados. No caso, sabe-se que os estudos pré-processaram as amostras.

7.8 CONSIDERAÇÕES FINAIS

Neste capítulo, foram apresentados os resultados obtidos a partir da aplicação de RNA em modelos híbridos de predição. Na Seção 7.1, foi descrito o modelo híbrido baseado em Perda no Espaço Livre e RNA. A Seção 7.2 abordou o modelo híbrido Cost-231 combinado com RNA, enquanto a Seção 7.3 explorou a integração entre o modelo Egli e RNA. Na sequência, a Seção 7.4 apresentou o modelo híbrido ECC-33 e RNA, e a Seção 7.5 destacou a associação do modelo Okumura-Hata com RNA. O modelo híbrido ITU-P.1546-6 e RNA foi detalhado na Seção 7.6. Por fim, a Seção 7.7 comparou os resultados obtidos com os de outros estudos, evidenciando o desempenho dos modelos híbridos. A síntese e as considerações gerais sobre os resultados foram discutidas na Seção 7.8, consolidando as principais conclusões deste capítulo.

RESULTADOS DA OTIMIZAÇÃO POR ALGORITMOS GENÉTICOS E EVOLUÇÃO DIFERENCIAL

Resumo do capítulo

Neste capítulo são apresentados os resultados das otimização dos modelos de propagação utilizando AG e ED. Os diferentes modelos otimizados são discutidos, incluindo o processo de otimização e os resultados obtidos. O ranqueamento entre os modelos e métodos é realizado para identificar as abordagens mais eficazes.

8.1 OTIMIZAÇÃO DO MODELO OKUMURA-HATA

Na Tabela 8.1 abaixo estão os valores as variáveis x_i otimizados pelo AG e pela ED para o modelo de propagação Okumura-Hata. Além das variáveis x_i , na última linha da tabela está o valor do EQM encontrado após a otimização.

Valores das variáveis otimizadas pelo AG e Erro Quadrático Médio para o modelo Okumura-Hata

Tabela 8.1 – Valores das variáveis otimizadas pelo AG e Erro Quadrático Médio para o modelo Okumura-Hata

	AG	ED
x_0	49,5725	-38,0584
x_1	-74,2299	27,61673
x_2	-76,7922	-8,2334
x_3	-15,529	-7,22591
x_4	-61,1053	-29,5853
x_5	-53,5752	-86,2576
x_6	-25,8262	54,376
x_7	-33,6564	-3,64013
x_8	47,51516	15,06026
EQM (dB)	50,21421	48,33676

Nota-se que através dos valores encontrados pela otimização através do AG para as variáveis x_i uma melhora no modelo comprovado pelo valor do EQM que diminuiu de 56,50332 dB para 50,21421 dB, ou seja, uma melhora de 11,13% enquanto para a ED o EQM diminuiu para 48,33676 dB (14,39%), portanto, houve um ganho na predição de, respectivamente, 6,28911 dB e 8,16656 dB. Desta forma observou-se que tanto utilizando o AG quando utilizando o ED houve melhora do modelo de propagação, sendo o segundo 1,87745 dB melhor no desempenho.

Por fim, a partir equação otimizada pelo AG ficou da seguinte maneira:

$$PL(dB) = 3447.767 + 1941.854 \log(f_c) + 1062.036 \log(h_t) + (-697.252 + 400.240 \log(h_r)) \log(d) - A(h_r) \quad (8.1)$$

$$A(h_r) = (-58.933 \log(f_c) - 18.078) h_r - (-52.504 \log(f_c) - 38.012) \quad (8.2)$$

E a equação otimizada pela ED ficou da seguinte maneira:

$$PL(dB) = -2646.962 + 722.454 \log(f_c) - 113.868 \log(h_t) + (-324.443 - 193.784 \log(h_r)) \log(d) - A(h_r) \quad (8.3)$$

$$A(h_r) = (-94.883 \log(f_c) - 38.063) h_r - (5.679 \log(f_c) - 12.0482) \quad (8.4)$$

A Tabela 8.2 a seguir mostra os valores das variáveis x_i para o modelo Okumura-Hata quando é adicionado o parâmetro altitude na equação da perda de caminho, na última linha está o valor do EQM para essa otimização.

Tabela 8.2 – Valores das variáveis otimizadas pelo ED e Erro Quadrático Médio para o modelo Okumura-Hata com adição da altitude

	AG	ED
x_0	-92,9024	-43,2667
x_1	67,30366	67,94443
x_2	-70,7976	93,08952
x_3	8,942511	4,235167
x_4	32,63664	14,45348
x_5	-7,63215	-40,2798
x_6	-1,31054	37,33553
x_7	-22,4736	45,02526
x_8	-1,66257	-70,0474
x_9	0,140746	-0,09577
EQM (dB)	41,2188	39,11442

Observa-se que com a adição do parâmetro altitude, a otimização através do AG para as variáveis x_i melhorou o desempenho, que foi constatado através do EQM que em relação à otimização sem o referido parâmetro, a função objetiva diminuiu para 41,2188 dB (24,18%) enquanto para a ED diminuiu para 39,11442 dB (29,66%), respectivamente, 8,99541 dB e 9,22234 dB, com relação ao algoritmo original a melhora foi de 15,28452 dB e 17,3889 dB. Portanto a adição do parâmetro altitude contribuiu para melhora na otimização pelos dois algoritmos.

Finalmente, a partir das equações otimizadas pelo GA, tem-se o seguinte:

$$\begin{aligned} PL(dB) = & -6461.362 + 1760.664\log(f_c) \\ & + 979.131\log(h_t) + (401.519 - 213.770\log(h_r)) \\ & \log(d) - A(h_r) \end{aligned} \quad (8.5)$$

$$\begin{aligned} A(h_r) = & (-8.395\log(f_c) + 0.917)h_r \\ & - (-35.059\log(f_c) + 1.330) + 0.141altitude \end{aligned} \quad (8.6)$$

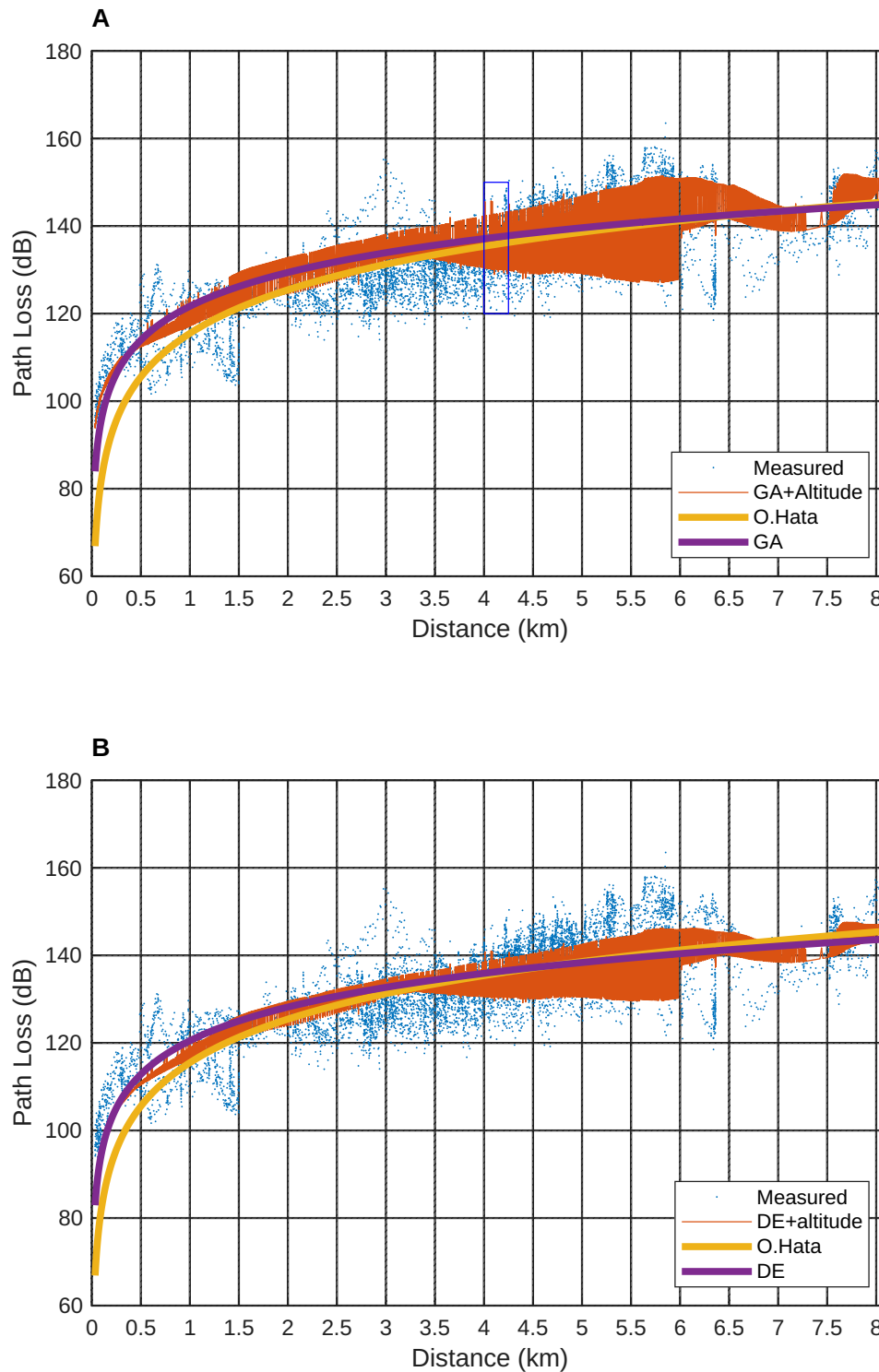
E a equação otimizada pela ED com o parâmetro altitude ficou da seguinte maneira:

$$\begin{aligned} PL(dB) = & -3009.199 + 1777.426\log(f_c) - \\ & 1287.428\log(h_t) + (190.159 - 94.670\log(h_r)) \\ & \log(d) - A(h_r) \end{aligned} \quad (8.7)$$

$$\begin{aligned} A(h_r) = & (-44.308\log(f_c) - 26.135)h_r \\ & - (70.239\log(f_c) + 56.038) - 0.096altitude \end{aligned} \quad (8.8)$$

A Figura 8.1a mostra uma comparação gráfica entre os modelos Okumura-Hata, modelo otimizado pelo AG e o modelo otimizado pelo AG com altitude. A Figura 8.1b mostra uma comparação gráfica entre os modelos Okumura-Hata, o modelo otimizado pela ED e o modelo otimizado pelo ED com altitude.

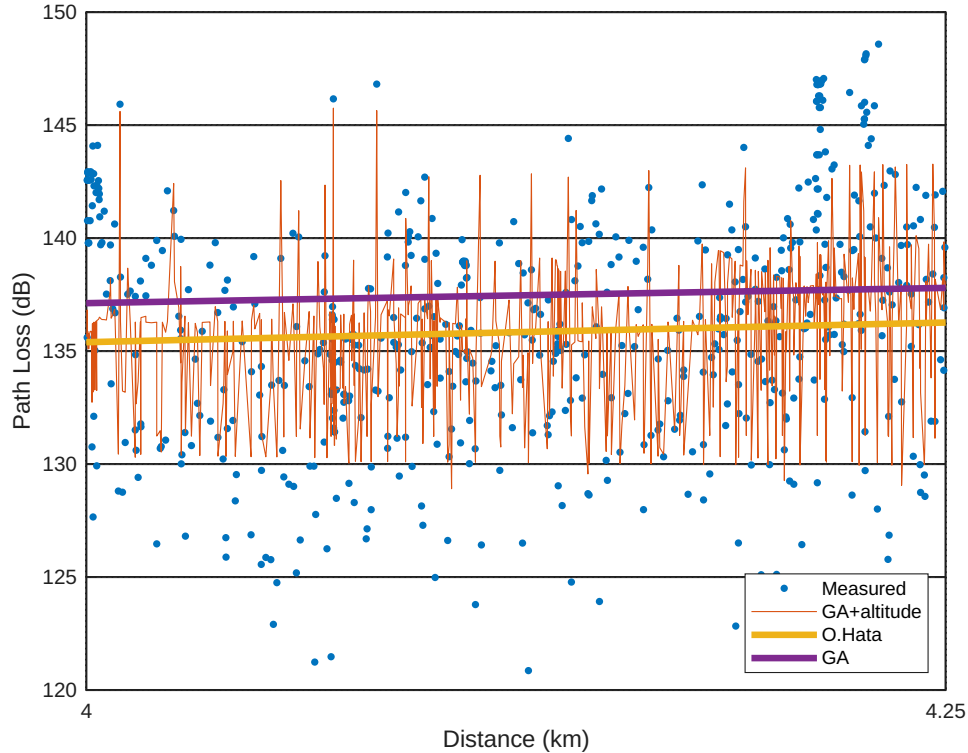
Figura 8.1 – PL com otimização por AG (A), DE (B) e pelo modelo Okumura-Hata



A Figura 8.1 é um gráfico da distância versus PL, através dele compreende-se melhor os resultados encontrados nas Tabela 8.2, observa-se que a linha roxa (modelo otimizado) está mais próxima deles do que a linha amarela (modelo sem otimização) dos pontos azuis (amostras). A linha marrom representa a otimização com o parâmetro altitude, como a Figura 8.1, possui uma escala pequena e não se conseguiu visualizar a variação desta linha, foi feito um zoom

representado pelo retângulo da Figura 8.1a e apresentado na Figura 8.2.

Figura 8.2 – Ampliação realizada na Figura 8.1 para melhorar a compreensão do efeito da adição do parâmetro altitude



Na Figura 8.2 pode-se observar que a adição do parâmetro altitude afeta o PL em distâncias muito curtas (linha marrom), isso é justificado, pois para essas distâncias pode ocorrer diferença de altitude, uma vez que os modelos tradicionais não utilizaram este parâmetro (linha amarela), esse efeito não ocorria. Portanto, a inclusão desse parâmetro resultou de forma inequívoca na melhora desempenho do modelo.

O *zoom* foi apresentado na Figura 8.1a para entendimento do que significa a linha marrom, porém o mesmo acontece na Figura 8.1b e nas outras figuras abaixo que representam o gráfico distância versus PL.

A Tabela 8.3 mostra um comparativo entre o modelo Okumura Hata e o modelo otimizado pelo AG, AG com adição do parâmetro altitude, ED e ED com adição do parâmetro altitude através dos parâmetros estatísticos: EM, EMA, EQM, E_{RMS} , σ e R^2

Tabela 8.3 – Comparação entre o Modelo Okumura-Hata e suas otimizações

Modelo	Algoritmo de Otimização	EM	EMA	EQM	E_{RMS}	σ	R^2
O. Hata	Sem Otimização	-1,0557	5,8203	56,5033	7,5169	4,7571	0,6262
O. Hata	AG	1,3643	5,7127	50,2142	7,0862	4,193	0,6262
O. Hata	AG + Altitude	1,2745	5,0348	42,8403	6,5453	4,1825	0,6827
O. Hata	ED	0,1473	5,6162	48,3731	6,9551	4,1028	0,6262
O. Hata	ED + Altitude	-0,7926	5,0225	39,7452	6,3044	3,8107	0,6976

A partir desta tabela, observa-se claramente a superioridade dos modelos otimizados pelo AG e pela ED em comparação com os modelos tradicionais. Essa superioridade é particularmente notável quando o parâmetro da altitude é incluído na modelagem.

8.2 OTIMIZAÇÃO DO MODELO ECC-33

Da mesma forma que foi feito para o modelo Okumura-Hata abaixo está a Tabela 8.4 com os valores os valores das variáveis x_i otimizados pelo AG e pela ED para o modelo de propagação ECC-33 e na última linha o valor do EQM.

Tabela 8.4 – Valores das variáveis otimizadas pelo ED e Erro Quadrático Médio para o modelo ECC-33

	AG	ED
x_0	-18,0629	13,20006
x_1	-46,2008	7,066653
x_2	-71,5693	-43,4754
x_3	70,21217	-84,1889
x_4	96,35757	-11,9329
x_5	-6,16778	-23,8923
x_6	-49,3634	-4,94365
x_7	2,976021	-20,6276
x_8	-4,39466	-3,67587
x_9	-47,6491	83,38738
x_{10}	-59,8741	-48,6027
EQM (dB)	42,85749	42,58661

Os algoritmos AG e ED também foram capaz de otimizar o modelo ECC-33 com as variáveis x_i e similarmente notou-se uma melhora significativa no desempenho constatado a partir do valor do EQM, de 453,0686 sem otimização para 42,87225 (90,54%), enquanto para a ED o EQM foi de 42,58924 (90,60%), portanto, houve uma melhora na predição de,

respectivamente, 410,19635 e 410,47936. Desta forma observou-se que tanto utilizando o AG quando utilizando o ED houve melhora do modelo de propagação.

Por fim, a partir equação otimizada pelo AG com o parâmetro altitude ficou da seguinte maneira:

$$PL(dB) = A_{fs} + A_{bm} - G_b - G_r \quad (8.9)$$

$$A_{fs} = -1669.012 - 924.016\log(d) - 1431.386\log(f) \quad (8.10)$$

$$A_{bm} = 1433.030 + 947.195\log(d) - 48.663\log(f) - 471.914(\log(f))^2 \quad (8.11)$$

$$G_b = \log\left(\frac{h_t}{200}\right) (41.539 - 25.489\log(d))^2 \quad (8.12)$$

$$G_r = -36.166h_r + 111.486 \quad (8.13)$$

E a equação otimizada pela ED com o parâmetro altitude ficou da seguinte maneira:

$$PL(dB) = A_{fs} + A_{bm} - G_b - G_r \quad (8.14)$$

$$A_{fs} = 1219.685 + 141.333\log(d) - 869.508\log(f) \quad (8.15)$$

$$A_{bm} = -1718.295 - 117.300\log(d) - 188.510\log(f) - 47.261(\log(f))^2 \quad (8.16)$$

$$G_b = \log\left(\frac{h_t}{200}\right) (-287.920 - 21.320\log(d))^2 \quad (8.17)$$

$$G_r = 63.291h_r + 90.498 \quad (8.18)$$

Da mesma forma que foi feito para o modelo Okumura-Hata a Tabela 8.5 a seguir mostra

os valores das variáveis x_i para o modelo ECC-33 quando é adicionado o parâmetro altitude na equação da perda de caminho, na última linha está o valor do EQM para essa otimização.

Tabela 8.5 – Valores das variáveis otimizadas pelo ED e Erro Quadrático Médio para o modelo ECC-33 com adição da altitude

	AG	ED
x_0	-10,1984	14,22775
x_1	-1,35957	15,55418
x_2	62,73018	-81,2575
x_3	61,80356	-79,6951
x_4	4,785868	-29,7232
x_5	-43,7177	-49,0154
x_6	-10,3014	78,84416
x_7	-12,446	29,29124
x_8	-2,73845	-2,73871
x_9	-47,0476	-28,8261
x_{10}	-4,96936	-69,6417
x_{11}	0,044779	0,082618
EQM (dB)	37,91029	36,09638

Observa-se que com a adição do parâmetro altitude, a otimização através do AG para as variáveis x_i o desempenho do modelo melhorou pois o EQM em relação à otimização sem o referido parâmetro, diminuiu para 37,91424 dB (91,63%) enquanto para a ED diminuiu para 36,09853 dB (92,03%), respectivamente, 4,95801 dB e 6,49071 dB, com relação ao algoritmo original a melhora foi de 415,15436 dB e 416,97007 dB. Portanto, a adição do parâmetro altitude contribuiu para melhora na otimização pelos dois algoritmos.

Por fim, a partir equação otimizada pelo AG com o parâmetro altitude ficou da seguinte maneira:

$$PL(dB) = A_{fs} + A_{bm} - G_b - G_r \quad (8.19)$$

$$A_{fs} = -942.332 - 27.191\log(d) + 1254.604\log(f) + 0.045altitude \quad (8.20)$$

$$A_{bm} = 1261.411 + 47.045\log(d) - 344.932\log(f) - 98.481384(\log(f))^2 \quad (8.21)$$

$$G_b = \log\left(\frac{h_t}{200}\right) (-173.721 - 15.883\log(d))^2 \quad (8.22)$$

$$G_r = -35.709h_r - 9.25295 \quad (8.23)$$

E a equação otimizada pela ED ficou da seguinte maneira:

$$PL(dB) = A_{fs} + A_{bm} - G_b - G_r + 0.087altitude \quad (8.24)$$

$$A_{fs} = 1314.644 + 311.084\log(d) - 1625.150\log(f) \quad (8.25)$$

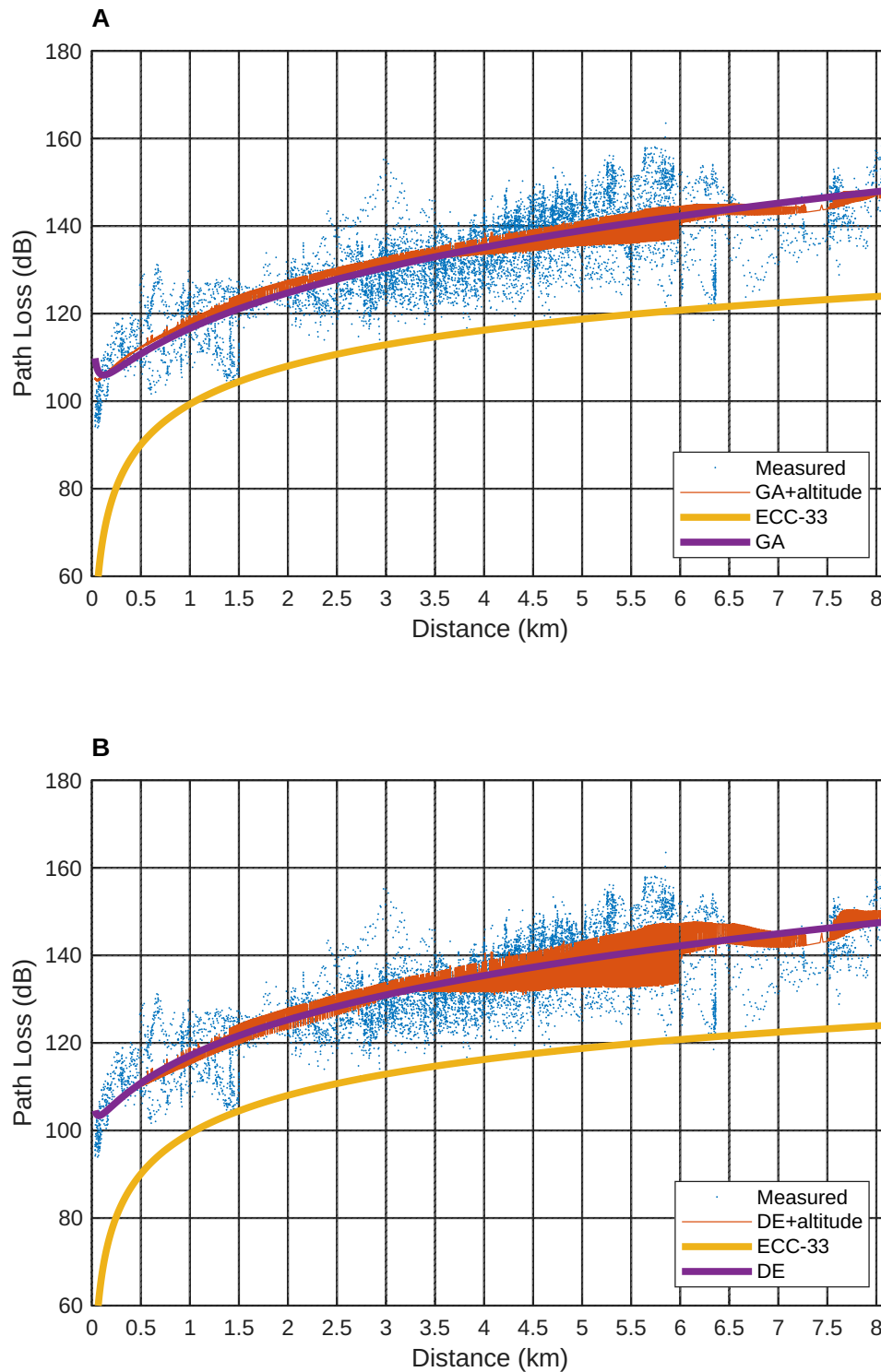
$$A_{bm} = -1626.577 - 292.179\log(d) - 386.731\log(f) + 753.750(\log(f))^2 \quad (8.26)$$

$$G_b = \log\left(\frac{h_t}{200}\right) (408.847 - 15.884\log(d))^2 \quad (8.27)$$

$$G_r = -21.879h_r + 129.673 \quad (8.28)$$

A Figura 8.3a mostra uma comparação gráfica entre os modelos ECC-33, modelo otimizado pelo AG e o modelo otimizado pelo AG com altitude. A Figura 8.3b mostra uma comparação gráfica entre os modelos ECC-33, o modelo otimizado pela ED e o modelo otimizado pelo ED com altitude.

Figura 8.3 – Predição de perda de percurso pelo AG, ED e modelo ECC-33



Através da Figura 8.3 compreende-se melhor os resultados encontrados nas Tabela, observa-se que a linha amarela (modelo sem otimização) está distante das amostras, justificando o EQM de 453,0686, enquanto a linha roxa passa por toda a região das amostras. Do mesmo modo que o modelo anterior a linha marrom (otimização com o parâmetro altitude), apresenta variações que não é possível visualizar devido a escala do gráfico, mas se comporta como a Figura 8.2,

isso ocorre, pois, como explicado anteriormente, em uma mesma distância pode haver duas amostras com altitudes diferentes, o que influencia na potência recebida e não foi levado em consideração nos modelos tradicionais.

A Tabela 8.6 mostra um comparativo entre o modelo ECC-33 e o modelo otimizado pelo AG, AG com adição do parâmetro altitude, ED e ED com adição do parâmetro altitude através dos parâmetros estatísticos: EM, EMA, EQM, E_{RMS} , σ e R^2

Tabela 8.6 – Comparação entre o ECC-33 e suas otimizações

Modelo	Algoritmo de Otimização	EM	EMA	EQM	E_{RMS}	σ	R^2
ECC-33	Sem Otimização	-19,9818	19,9827	453,0686	21,2854	7,3326	0,6029
ECC-33	AG	-0,112	5,2309	42,8722	6,5477	3,9384	0,6687
ECC-33	AG + Altitude	0,2569	4,8659	37,9142	6,1575	3,7735	0,7122
ECC-33	ED	0,0009	5,2148	42,5892	6,526	3,9238	0,6708
ECC-33	ED + Altitude	0,0008	4,6741	36,0985	6,0082	3,7753	0,7209

8.3 OTIMIZAÇÃO DO MODELO EGLI

Os valores das variáveis x_i encontrados a partir da otimização por AG e por ED para o modelo Egli estão na Tabela 8.7 abaixo:

Tabela 8.7 – Valores das variáveis otimizadas pelo ED e Erro Quadrático Médio para o modelo Egli

	AG	ED
x_0	7,48132	28,87531
x_1	0,660535	0,639113
x_2	-38,1807	36,95115
x_3	-24,9014	63,9597
x_4	-22,8899	0,741089
EQM (dB)	48,44783	48,35141

Ao realizar a otimização pelos algoritmos AG e ED, constatou-se que conforme esperado e similarmente às otimizações anteriores houve uma melhora no EQM que sem otimização possui valor de 198,4842 dB, enquanto a otimização por AG, passou para 48,33678 dB (75,59%) e com a otimização por ED 48,33676 dB (75,64%). Observa-se que para este modelo os dois algoritmos tiveram a otimização bastante próxima 0,9642 dB, ou seja, menos que a diferença da primeira casa decimal.

Por fim, a partir equação otimizada pelo AG com o parâmetro altitude ficou da seguinte maneira:

$$PL(dB) = 149.626\log(f) + p_o - 1746.499 \quad (8.29)$$

$$p_o = 26.421\log(d) + 763.614\log(h_t) + 249.014\log(h_r) \quad (8.30)$$

E a equação otimizada pela ED com o parâmetro altitude ficou da seguinte maneira:

$$PL(dB) = 577.506\log(f) + p_o - 1294.311 \quad (8.31)$$

$$p_o = 16.886\log(d) + 28216.415\log(h_t) + 15926.860\log(h_r) \quad (8.32)$$

Os valores das variáveis x_i encontrados a partir da otimização por AG e por ED para o modelo Egli com adição do parâmetro altitude estão na Tabela 8.8 abaixo:

Tabela 8.8 – Valores das variáveis otimizadas pelo ED e Erro Quadrático Médio para o modelo Egli com adição da altitude

	AG	ED
x_0	-20,5974	55,39793
x_1	0,478728	0,479187
x_2	-92,2225	-63,129
x_3	-36,7587	23,6487
x_4	-28,7346	-67,5174
x_5	0,064683	0,095766
EQM (dB)	40,63053	39,11442

Com a adição do parâmetro altitude o desempenho da otimização foi ainda melhor. O EQM com o AG foi de 40,63053 dB (79,53%), ou seja, 7,8173 dB abaixo do que com o algoritmo sem a altitude. Para o algoritmo ED o valor do EQM foi de 39,11442 dB (80,29%), portanto, 9,23438 dB abaixo do que o algoritmo sem a utilização do parâmetro altitude.

Por fim, a partir equação otimizada pelo AG com o parâmetro altitude ficou da seguinte maneira:

$$PL(dB) = -411.948\log(f) + p_o - 2192.445 + 0.065altitude \quad (8.33)$$

$$p_o = 19.149 \log(d) - 1844.450 \log(h_t) - 367.587 \log(h_r) \quad (8.34)$$

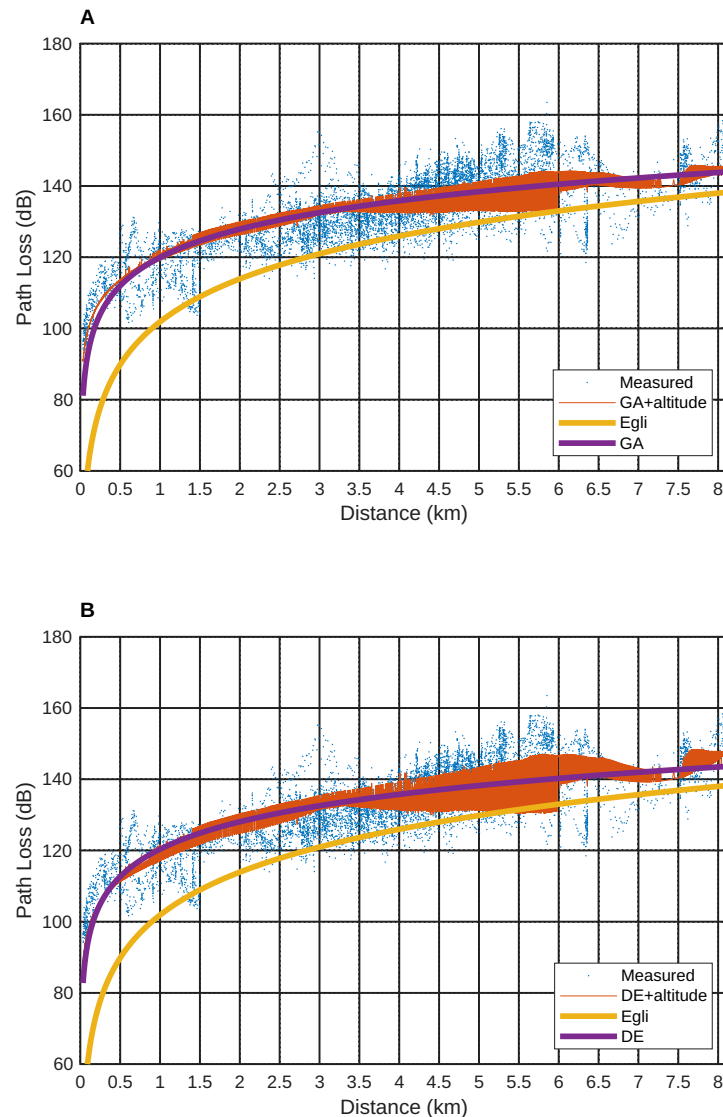
E a equação otimizada pela ED com o parâmetro altitude ficou da seguinte maneira:

$$PL(dB) = 1107.959 \log(f) + p_o + 148028.522 + 0.006 \text{altitude} \quad (8.35)$$

$$p_o = 9.176 \log(d) - 116438.284 \log(h_t) + 8692.955 \log(h_r) \quad (8.36)$$

A Figura 8.4a mostra uma comparação gráfica entre os modelos Egli, modelo otimizado pelo AG e o modelo otimizado pelo AG com altitude. A Figura 8.4b mostra uma comparação gráfica entre os modelos Egli, o modelo otimizado pela ED e o modelo otimizado pelo ED com altitude. Neste modelo também se identificou que a adição do parâmetro altitude colabora significativamente para melhor desempenho do modelo proposto.

Figura 8.4 – Predição de perda de percurso pelo modelo Egli e suas otimizações por AG, ED e AG, ED e altitude



Na Figura 8.4 observa-se em azul os valores das amostras, em amarelo o valor de predição de perda de propagação com o modelo sem otimização, em roxo a otimização do modelo pelos algoritmos AG e ED e em marrom a otimização pelos algoritmos AG e ED com o parâmetro altitude. Ao realizar uma análise visual percebe-se que a linha roxa representa melhor uma aproximação com os valores das amostras do que a linha amarela do modelo sem otimização. Além disso, é observado que a linha marrom, ocorreu o mesmo que nas figuras anteriores (8.1, 8.3), ou seja, este é motivo do EQM ao utilizar o parâmetro altitude ser melhor.

A Tabela 8.9 mostra um comparativo entre o modelo Egli e o modelo otimizado pelo AG, AG com adição do parâmetro altitude, ED e ED com adição do parâmetro altitude através dos parâmetros estatísticos: EM, EMA, EQM, E_{RMS} , σ e R^2

Por último, na Tabela 8.10 temos os valores das variáveis x_i realizadas pela otimização através de AG e DE para o modelo Cost 231.

Tabela 8.9 – Comparação entre o Egli e suas otimizações

Modelo	Algoritmo de Otimização	EM	EMA	EQM	E _{RMS}	σ	R ²
Egli	Sem Otimização	-11,1433	11,6079	198,4842	14,0884	7,9842	0,6262
Egli	AG	0,0172	5,6191	48,4478	6,9604	4,1079	0,6262
Egli	AG + Altitude	-0,0498	5,0772	40,6305	6,3742	3,8541	0,6929
Egli	ED	-0,0015	5,6156	48,3514	6,9535	4,1011	0,6262
Egli	ED + Altitude	-0,0009	4,9416	39,117	6,2544	3,834	0,6976

Tabela 8.10 – Valores das variáveis otimizadas pelo AG e Erro Quadrático Médio para o modelo Cost 231

	AG	ED
x_0	88.53048	11.53235
x_1	-40.1743	15.78508
x_2	38.5985	48.27671
x_3	19.31354	-0.29517
x_4	71.26117	-3.28114
x_5	-86.8928	83.1537
x_6	-96.5008	-22.9436
x_7	94.79704	-49.6674
x_8	76.3432	-83.4231
EQM(dB)	48.6008	48.3514

Bem como os modelos citados anteriormente, conseguiu-se uma melhora significativa para o desempenho através da utilização dos algoritmos AG e ED. O modelo sem otimização apresenta o EQM de 64,16348 dB, o valor otimizado pelo AG: 48,60082 dB (24,25%) e o valor otimizado pelo ED: 48,35141 dB (24,64%). O AG apresentou melhora de 15,56266 dB e o ED apresentou melhora de 15,81207 dB. Portanto o desempenho do ED foi 0,24941 dB melhor com relação ao AG. Outro aspecto a se destacar é que o valor encontrado pelo ED para este modelo foi o mesmo valor (48,33676 dB) encontrado para o modelo Egli.

Por fim, a partir equação otimizada pelo AG com o parâmetro altitude ficou da seguinte maneira:

$$PL(dB) = 4098.961 - 1361.909\log(f) - 533.431\log(h_t) \\ (867.178 - 466.761\log(h_t))\log(d) - A(h_r) \quad (8.37)$$

$$A(h_r) = (-95.582\log(f_c) - 67.550)h_r \\ - (147.883\log(f_c) - 61.075)) \quad (8.38)$$

E a equação otimizada pela ED com o parâmetro altitude ficou da seguinte maneira:

$$PL(dB) = 533.947805 + 535.114\log(f) - 667.184\log(h_t) \\ (-13.253 - 21.491\log(h_t))\log(d) - A(h_r) \quad (8.39)$$

$$A(h_r) = (91.469\log(f_c) + 16.061)h_r \\ - (-77.4811\log(f_c) + 66.738) \quad (8.40)$$

A Tabela 8.11 com a otimização para o modelo Cost 231 com adição do parâmetro altitude.

Tabela 8.11 – Valores das variáveis otimizadas pelo ED e Erro Quadrático Médio para o modelo Cost-231 com adição da altitude

	AG	ED
x_0	3.307912	94.59615
x_1	-11.5719	-38.6182
x_2	-36.8672	13.79075
x_3	13.17686	24.77399
x_4	48.43454	92.40409
x_5	26.80406	45.72959
x_6	-34.1109	-41.732
x_7	69.09678	13.44892
x_8	-34.1456	50.49498
x_9	0.123418	0.095765
EQM(dB)	40.1195	39.1170

Através da utilização dos algoritmos AG e ED conseguiu-se melhorar o desempenho do modelo. Sem otimização, o modelo apresenta o EQM de 64,16348 dB, o valor otimizado pelo AG com adição do parâmetro altitude: 40,11492 dB (37,47%) e o valor otimizado pelo ED: 39,11442 (39,04%). O AG apresentou melhora de 24,044 dB e o ED apresentou melhora de 25,0465 dB. Portanto, o desempenho do ED foi 1,00243 dB melhor com relação ao AG.

Por fim, a partir equação otimizada pelo AG com o parâmetro altitude ficou da seguinte maneira:

$$PL(dB) = 153.1563 - 392.287\log(f_c) + 509.505\log(h_t) \\ + (591.641 - 317.246\log(h_t))\log(d) - A(h_r) \\ + 0.123altitude \quad (8.41)$$

$$A(h_r) = (29.484\log(f_c) + 23.878)h_r - (107.791\log((f_c) + 27.316)) \quad (8.42)$$

E a equação otimizada pela ED com o parâmetro altitude ficou da seguinte maneira:

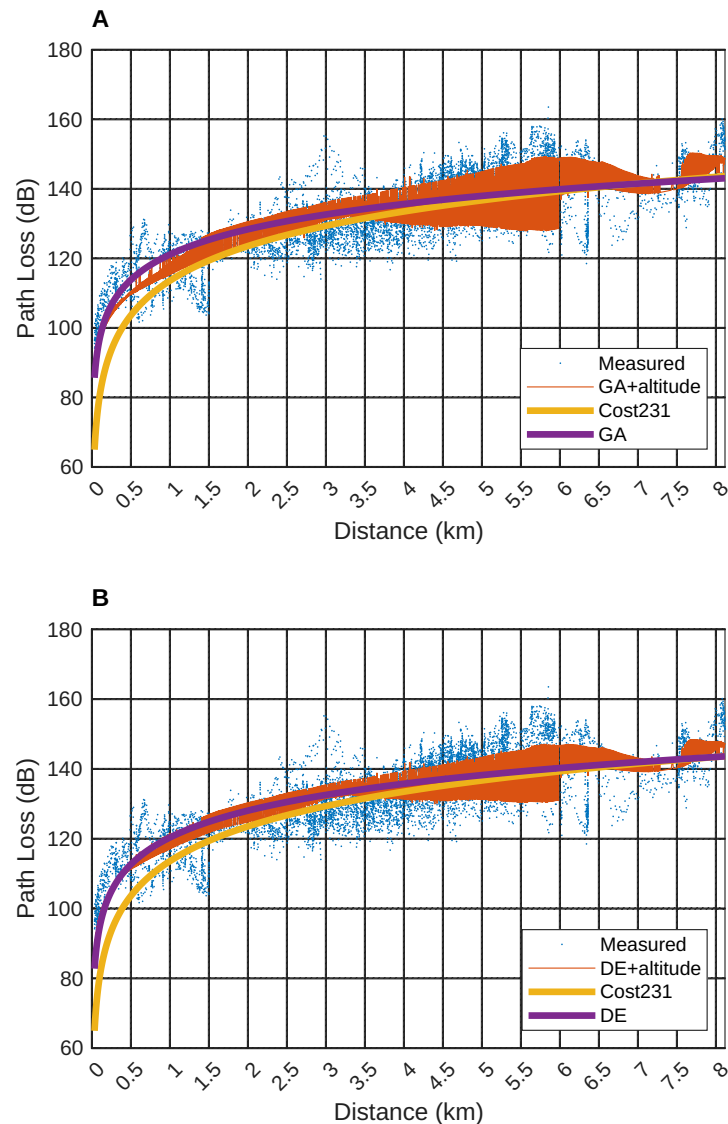
$$\begin{aligned} PL(dB) = & 4379.802 - 1309.157\log(f_c) - 190.588\log(h_t) \\ & + (1112.352 - 605.2475\log(h_t))\log(d) - A(h_r) \\ & + 0.096altitude \end{aligned} \quad (8.43)$$

$$A(h_r) = (50.302\log(f_c) + 29.212)h_r - (20.980\log((f_c) - 40.396)) \quad (8.44)$$

8.4 OTIMIZAÇÃO DO MODELO COST 231

A Figura 8.5a mostra uma comparação gráfica entre os modelos Cost 231, modelo otimizado pelo AG e o modelo otimizado pelo AG com altitude. A Figura 8.5b mostra uma comparação gráfica entre os modelos Cost 231, o modelo otimizado pela ED e o modelo otimizado pelo ED com altitude.

Figura 8.5 – Predição de perda de percurso pelo modelo Cost 231 e suas otimizações por AG, ED e AG, ED e altitude



A Figura 8.5 é similar as figuras dos outros modelos, em que se observa em amarelo o valor de predição de perda de propagação com o modelo sem otimização, em roxo a otimização do modelo pelos algoritmos AG e ED e em marrom a otimização pelos algoritmos AG e ED com o parâmetro altitude. Ao realizar uma análise visual percebe-se que a linha roxa representa melhor uma aproximação com os valores das amostras do que a linha amarela do modelo sem otimização. Além disso é observado que a linha marrom consegue também acompanhar as amostras.

A Tabela 9.1 mostra um comparativo entre o modelo Cost 231 e o modelo otimizado pelo AG, AG com adição do parâmetro altitude, ED e ED com adição do parâmetro altitude através dos parâmetros estatísticos: EM, EMA, EQM, E_{RMS} , σ e R^2

Tabela 8.12 – Comparação entre o Modelo Cost 231 e suas otimizações

Modelo	Algoritmo de Otimização	EM	EMA	EQM	E _{RMS}	σ	R ²
Cost 231	Sem Otimização	-2,9622	6,2068	64,1635	8,0102	5,0638	0,6262
Cost 231	AG	0,0124	5,6195	48,6008	6,9714	4,1259	0,6262
Cost 231	AG + Altitude	-0,131	4,9504	40,1195	6,334	3,9515	0,6945
Cost 231	ED	-0,0015	5,6156	48,3514	6,9535	4,1011	0,6262
Cost 231	ED + Altitude	-0,0009	4,9416	39,117	6,2544	3,834	0,6976

A Tabela 8.13 sintetiza a comparação entre todas as otimizações realizadas acima. Nela há o valor do EQM para cada modelo, sem otimização, otimizado pelo AG, otimizado pelo AG mais o parâmetro altitude, otimizado pelo ED e otimizado pelo ED mais o parâmetro altitude.

Tabela 8.13 – Comparação entre os modelos, e otimização por AG, ED e otimizações com adição do parâmetro altitude

Algoritmo de Otimização/EQM	O. Hata	ECC-33	Egli	Cost 231
Sem otimização (dB)	56,50332	453,0686	198,4842	64,16348
GA (dB)	50,21421	42,87225	48,44783	48,60082
GA+altitude (dB)	42,84034	37,91424	40,63053	40,11946
ED (dB)	48,37309	42,58924	48,35141	48,35141
ED+altitude (dB)	39,74517	36,09853	39,11703	39,11703

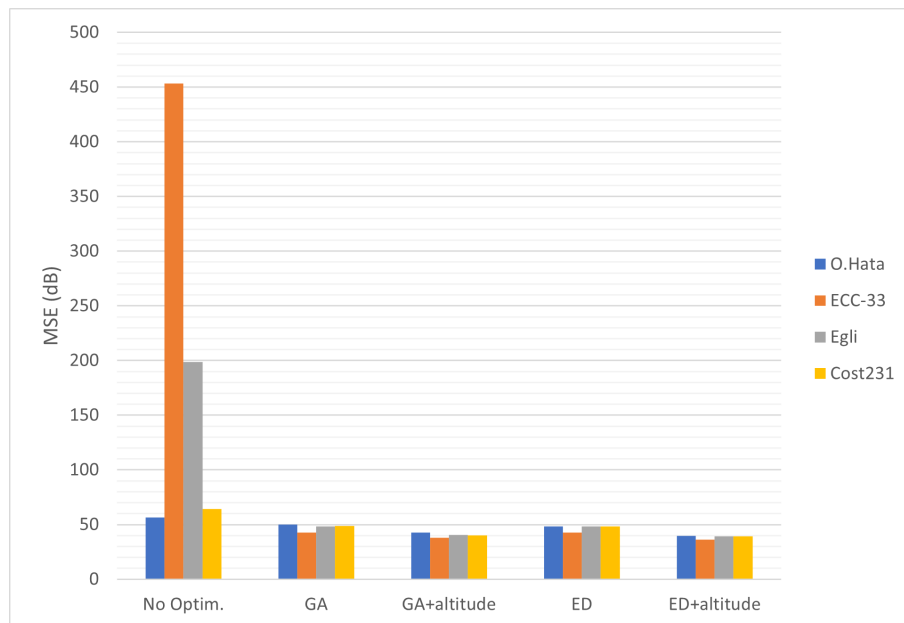
Ao analisarmos os valores contidos na Tabela 8.13 observa-se que a otimização melhorou o desempenho de todos os modelos. Comparando com os modelos originais, a otimização, o Modelo Okumura-Hata teve uma melhora no desempenho em 11,13%. Quando utilizado ED o desempenho melhorou em 14,39%. O Modelo ECC-33 teve uma melhora no desempenho em 90,54%. Quando utilizado ED o desempenho melhorou em 90,60%. O Modelo Egli teve uma melhora no desempenho em 75,59%. Quando utilizado ED o desempenho melhorou em 75,64%. E por fim, o Modelo Cost 231, teve uma melhora no desempenho em 24,25%. Quando utilizado ED o desempenho melhorou em 24,64%.

Quando se analisa o impacto da adição do parâmetro altitude, o modelo Okumura-Hata original melhorou em 24,18%. Quando utilizado ED, o desempenho melhorou em 29,66%. O Modelo ECC-33, teve uma melhora no desempenho em 91,63%. Quando utilizado ED, o desempenho melhorou em 92,03%. O Modelo Egli, teve uma melhora no desempenho em 79,53%. Quando utilizado ED, o desempenho melhorou em 80,29%. E por fim, o Modelo Cost 231, teve uma melhora no desempenho em 37,47%. Quando utilizado ED, o desempenho melhorou em 39,04

Uma outra análise possível é a comparação entre a otimização sem o parâmetro altitude e a otimização com o parâmetro altitude, neste caso para o modelo Okumura-Hata com o parâmetro

altitude o desempenho foi 14,68% melhor que sem o parâmetro altitude. Quando utilizado ED, o desempenho melhorou em 21,71% melhor do que sem altitude. O modelo ECC-33 com o parâmetro altitude o desempenho foi 11,56%. Quando utilizado ED, o desempenho com o parâmetro altitude foi em 15,24% melhor do que sem altitude. O modelo Egli com o parâmetro altitude o desempenho foi 16,14% melhor que sem o parâmetro altitude. Quando utilizado ED o desempenho com o parâmetro altitude foi em 19,10% melhor do que sem altitude. O modelo Cost 231 com o parâmetro altitude o desempenho foi 17,45% melhor que sem o parâmetro altitude. Quando utilizado ED o desempenho com o parâmetro altitude foi em 19,10% melhor do que sem altitude.

Figura 8.6 – Predição de perda de percurso pelo modelo Cost 231 e suas otimizações por AG, ED e AG, ED e altitude



A partir da Figura 8.6 ficou constata-se que com a proposição de variáveis multiplicativas para ajuste de cada parcela dos modelos tradicionais, o processo de otimização da busca dos valores ótimos das variáveis levou a uma otimização em relação aos modelos originais. Além disso, quando estes modelos otimizados são comparados entre si observou-se uma convergência para valores mínimos e próximos entre eles.

8.5 RANQUEAMENTO ENTRE OS MODELOS E MÉTODOS

A partir dos dados da Tabela 8.13, foram elaboradas quatro Tabelas (8.14 a 8.17), cada uma apresentando o ranqueamento dos métodos de otimização de acordo com o desempenho observado em diferentes modelos. Nas Tabelas 8.14- 8.17, foi realizada uma classificação de qual método apresentou o melhor resultado dentro do respectivo modelo.

Tabela 8.14 – Ranqueamento do modelo Okumura-Hata com relação aos Algoritmos de Otimização

Ranking	Algoritmo de Otimização	EQM(dB)
1º	ED+altitude (dB)	39,74517
2º	GA+altitude (dB)	42,84034
3º	ED (dB)	48,37309
4º	GA (dB)	50,21421
5º	Sem otimização (dB)	56,50332

Quanto ao modelo Okumura-Hata o melhor desempenho foi do algoritmo ED com adição do parâmetro altitude, seguido pelo GA com adição do parâmetro altitude, algoritmo ED, algoritmo GA e por último o algoritmo sem otimização.

Tabela 8.15 – Ranqueamento do modelo ECC-33 com relação aos Algoritmos de Otimização

Ranking	Algoritmo de Otimização	EQM (dB)
1º	ED+altitude	36,09853
2º	GA+altitude	37,91424
3º	ED	42,58924
4º	GA	42,87225
5º	Sem otimização	453,0686

Quanto ao modelo ECC-33 o melhor desempenho foi do algoritmo ED com adição do parâmetro altitude, seguido pelo GA com adição do parâmetro altitude, algoritmo ED, algoritmo GA e por último o algoritmo sem otimização.

Tabela 8.16 – Ranqueamento do modelo Egli com relação aos Algoritmos de Otimização

Ranking	Algoritmo de Otimização	EQM (dB)
1º	ED+altitude	39,11703
2º	GA+altitude	40,63053
3º	ED	48,35141
4º	GA	48,44783
5º	Sem otimização	198,4842

Quanto ao modelo Egli o melhor desempenho foi do algoritmo ED com adição do parâmetro altitude, seguido pelo GA com adição do parâmetro altitude, algoritmo ED, algoritmo GA e por último o algoritmo sem otimização.

Tabela 8.17 – Ranqueamento do modelo Cost 231 com relação aos Algoritmos de Otimização

Ranking	Algoritmo de Otimização	EQM (dB)
1º	ED+altitude	39,11703
2º	GA+altitude	40,11946
3º	ED	48,35141
4º	GA	48,60082
5º	Sem otimização	64,16348

Quanto ao modelo Cost 231 o melhor desempenho foi do algoritmo ED com adição do parâmetro altitude, seguido pelo GA com adição do parâmetro altitude, algoritmo ED, algoritmo GA e por último o algoritmo sem otimização.

Além disso, foram elaboradas quatro Tabelas adicionais que apresentam o ranqueamento dos algoritmos de otimização em relação aos diferentes modelos de predição. Nessas tabelas, os algoritmos foram classificados com base no desempenho observado em cada modelo, permitindo identificar quais técnicas de otimização foram mais eficazes para melhorar a precisão de cada abordagem preditiva.

Tabela 8.18 – Ranqueamento dos modelos com relação aos algoritmos sem otimização

Ranking	Sem Otimização	EQM (dB)
1º	O. Hata	56,50332
2º	Cost-231	64,46348
3º	Egli	198,4842
4º	ECC-33	453,0686

A Tabela 8.18 apresenta classificação de desempenho dos modelos sem otimização e revelou que o modelo de Okumura-Hata apresentou o menor EQM, com 56,50332 dB, seguido pelo modelo Cost 231, com 64,16348 dB. O modelo de Egli registrou um EQM significativamente maior, de 198,4842 dB, enquanto o modelo final obteve o maior valor entre os avaliados, com 453,0686 dB.

Tabela 8.19 – Ranqueamento dos modelos com relação ao AG

Ranking	Modelo	EQM (dB)
1º	ECC-33	42,87225
2º	Egli	48,44783
3º	Cost 231	48,60082
4º	O. Hata	50,21421

A Tabela 8.19 classificação dos modelos otimizados pelo Algoritmo Genético (AG), sem a adição do parâmetro altitude, indicou que o modelo ECC-33 apresentou o melhor desempenho,

com um erro quadrático médio (EQM) de 42,87225 dB. Em seguida, destacou-se o modelo Egli, com um EQM de 48,44783 dB, seguido pelo modelo Cost 231, com 48,60082 dB. O modelo Okumura-Hata obteve o maior EQM entre os avaliados, registrando 50,21421 dB.

Tabela 8.20 – Ranqueamento dos modelos com relação ao AG com a adição do Parâmetro Altitude

Ranking	Modelo	EQM (dB)
1º	ECC-33	37,91424
2º	Cost 231	40,11946
3º	Egli	40,63053
4º	O. Hata	42,84034

A classificação dos modelos otimizados pelo AG com a adição do parâmetro altitude apresentada na Tabela 8.20 mostra os seguintes resultados: o modelo ECC-33 obteve o menor EQM, com 37,91424 dB, seguido pelo modelo Cost 231, com 40,11946 dB. Em terceiro lugar ficou o modelo Egli, com 40,63053 dB, enquanto o modelo Okumura-Hata apresentou o maior valor entre os analisados, com 42,84034 dB.

Tabela 8.21 – Ranqueamento dos modelos com relação a ED

Ranking	Modelo	EQM(dB)
1º	ECC-33	42,58924
2º	Cost 231	48,35141
2º	Egli	48,35141
4º	O. Hata	48,37309

Na classificação dos modelos otimizados pelo método ED sem a adição do parâmetro altitude (Tabela 8.21, o modelo ECC-33 apresentou o EQM, com 42,58924 dB, seguido pelos modelos Cost 231 e Egli, ambos com um EQM de 48,35141 dB, indicando que convergiram para o mesmo valor. Por fim, o modelo Okumura-Hata registrou um EQM de 48,37309 dB, sendo o maior entre os modelos analisados.

Tabela 8.22 – Ranqueamento dos modelos com relação a ED com adição do parâmetro altitude

Ranking	Modelo	EQM(dB)
1º	ECC-33	36,09853
2º	Cost 231	39,11703
2º	Egli	39,11703
4º	O. Hata	39,74517

Na Tabela 8.22 está a classificação dos modelos otimizados pelo ED com a adição do parâmetro altitude. Os resultados indicaram que o modelo ECC-33 apresentou o menor erro

quadrático médio (EQM), com 36,09853 dB, seguido pelos modelos Cost 231 e Egli, ambos com um EQM de 39,11703 dB, e, por fim, o modelo Okumura-Hata, com 39,74517 dB. É importante destacar que, nesta classificação, os modelos Cost 231 e Egli apresentaram o mesmo valor de EQM, evidenciando desempenho semelhante para essas abordagens ao serem otimizadas com a inclusão do parâmetro de altitude.

Por fim, foi elaborado ranqueamento geral dos modelos, modelos otimizados e modelos otimizados com adição do parâmetro altitude. A Tabela 8.23 apresenta esta classificação.

Tabela 8.23 – Ranqueamento Geral entre os modelos de predição e Algoritmos de otimização

Ranking	Modelo/Otimização	EQM(dB)
1º	ECC-33/ED+altitude	36.09853
2º	ECC-33/GA+altitude	37.91424
3º	Egli/ED+altitude	39.11703
4º	Cost 231/ED+altitude	39.11703
5º	O. Hata/ED+altitude	39.74517
6º	Cost231/GA+altitude	40.11946
7º	Egli/GA+altitude	40.63053
8º	ECC-33/ED	42.58924
9º	O. Hata/GA+altitude	42.84034
10º	ECC-33/GA	42.87225
11º	Egli/ED	48.35141
12º	Cost 231/ED	48.35141
13º	O. Hata/ED	48.37309
14º	Egli/GA	48.44783
15º	Cost 231/GA	48.60082
16º	O. Hata/GA	50.21421
17º	O. Hata/Sem otimização	56.50332
18º	Cost 231/Sem otimização	64.16348
19º	Egli/Sem otimização	198.4842
20º	ECC-33/Sem otimização	453.0686

Os resultados apresentados na Tabela 8.23 mostram que os modelos otimizados com a adição do parâmetro altitude apresentaram um desempenho superior em comparação aos modelos sem essa adaptação. Entre os algoritmos utilizados, o ED demonstrou melhor eficiência em relação ao AG, evidenciando sua maior capacidade de reduzir o EQM. Por outro lado, os modelos sem qualquer otimização ficaram nas últimas posições do ranking, destacando que a metodologia proposta neste trabalho contribuiu significativamente para o aprimoramento do desempenho dos modelos de predição analisados.

8.6 CONSIDERAÇÕES FINAIS

Neste capítulo, foram apresentados os resultados obtidos com a aplicação de AGs e ED para a otimização de modelos de previsão de perda de propagação. Na Seção 8.1, foram abordados os ajustes realizados no Modelo Okumura-Hata, enquanto a Seção 8.2 detalhou a otimização aplicada ao Modelo ECC-33. A Seção 8.3 tratou da melhoria do Modelo Egli, seguida pela Seção 8.4, que descreveu a otimização do Modelo COST 231. Posteriormente, na Seção 8.5, foi realizado o ranqueamento entre os modelos e métodos, avaliando o desempenho de cada abordagem. Por fim, a Seção 8.6 apresentou as conclusões gerais deste capítulo, destacando a relevância dos resultados obtidos para a melhoria dos modelos de propagação.

CONCLUSÕES

Neste trabalho foram combinadas técnicas de Inteligência Computacional com métodos tradicionais reconhecidos na literatura para aprimorar a precisão das previsões de PL. Utilizando dados de potência coletados em diversos pontos da cidade de Uberlândia, foram analisadas as informações do sistema irradiante e desenvolvidos modelos preditivo mais robustos e mais precisos. Foram desenvolvidos modelos híbridos compostos pelos modelos tradicionais e RNA e também foram otimizados modelos a partir de AG e ED.

No primeiro capítulo é realizada a introdução também são apresentados os objetivos do trabalho, destacando o propósito da pesquisa. A estrutura do documento é delineada para oferecer uma visão geral do conteúdo a ser abordado.

O Capítulo 2 apresentou a revisão bibliográfica, o resumo de alguns trabalhos que foram realizados com o propósito semelhante. E em cada parágrafo do capítulo está descrito como foi desenvolvido o trabalho de cada autor.

O Capítulo 3 concentra-se na explicação dos modelos de propagação de sinal, incluindo uma análise detalhada dos mecanismos de perda no espaço livre e uma série de modelos utilizados na área, como o Modelo Okumura-Hata, Cost 231 Hata Model, Modelo Egli, ECC-33 e a recomendação ITU P-1546. Também são discutidos os procedimentos e tecnologias relacionadas à coleta de dados, abrangendo tanto o sistema de transmissão quanto o sistema de recepção, além do processamento subsequente desses dados.

O Capítulo 4 apresenta os conceitos fundamentais das RNAs, incluindo modelos de neurônios, tipos de função de ativação e diferentes arquiteturas de rede, como redes *feedforward* de camada única e múltiplas camadas. Os estágios do treinamento de uma RNA são detalhados, juntamente com o algoritmo de treinamento e o algoritmo LM.

No Capítulo 5 são abordados os AGs e a ED como métodos de otimização. São discutidos os diferentes estágios e componentes desses algoritmos, como a população inicial, seleção, cruzamento, mutação e critérios de parada. Também são exploradas as funções de aptidão e o processo de otimização.

No Capítulo 6, apresentou a metodologia aplicada neste trabalho, nele falou-se da região em que foram coletados os dados, que é a cidade de Uberlândia. Foi descrito sobre o sistema de transmissão e suas características. Depois, descreveu-se o sistema de recepção. A seguir foi mostrado como foi feita a coleta de dados e como esses dados foram tratados. Também está neste capítulo a configuração da RNA que foi utilizada nos experimentos e, por fim, foram apresentadas as métricas estatísticas utilizadas como validação.

As duas próximas seções (9.1 e 9.2) apresentam as conclusões das duas abordagens utilizadas neste trabalho.

9.1 CONCLUSÕES ACERCA DO APRIMORAMENTO DOS MODELOS DE PROPAGAÇÃO POR REDES NEURAIS ARTIFICIAIS

No Capítulo 7 observou-se que a aplicação de modelos híbridos combinando perdas no espaço livre com RNAs demonstrou resultados consistentemente superiores em comparação com os modelos tradicionais. A análise dos testes, resumidos nas tabelas e figuras apresentadas, evidencia que a configuração ideal de neurônios na camada escondida varia entre 90 e 170, proporcionando a melhor generalização e menor erro.

Nota-se que em todos os casos, o modelo híbrido melhorou a performance do modelo de propagação, principalmente pelo fato de que os modelos foram projetados para distâncias maiores que 1 km, e a RNA consegue prever para distâncias inferiores a esta. Observa-se na Tabela 7.14 e utilizando como métrica os EQM e o E_{RMS} , o modelo híbrido que apresentou melhor resultado foi o Okumura-Hata-RNA, seguido pela Recomendação ITU-P.1546-6-RNA, Egli-RNA, Cost 231-RNA, ECC-33-RNA e Perda Espaço Livre-RNA. Embora os resultados tenham ficado muito próximos, com menos de 1 dB de diferença entre o maior e o menor, observa-se que ao utilizar um modelo de PL como entrada da RNA, influencia no resultado final.

O modelo híbrido, em todos os casos analisados—seja utilizando Perda no Espaço Livre, Cost-231, Egli, ECC-33, Okumura-Hata, ou ITU-P.1546-6—superou os modelos base em termos de EM, EMA, EQM, E_{RMS} , σ , R^2 . Esses resultados confirmam a eficácia das propostas de modelos híbridos na melhora da precisão preditiva em comparação com os métodos tradicionais, demonstrando a superioridade das redes neurais na modelagem e previsão de perdas de percurso em diferentes contextos e parâmetros.

Nas Tabelas 7.13 e 7.14 foram apresentadas uma comparação com outros estudos, nota-se que tratando EM, somente o Santos (2021) teve erro menor que este estudo, dos trabalhos que calcularam essa métrica, e se tratando de EMA, este trabalho só não foi melhor que os trabalhos de Ayadi, Zineb e Tabbane (2017), Sotiroudís et al. (2013) e Cerri et al. (2004) dos trabalhos que calcularam essa métrica. Na comparação entre o EQM e o E_{RMS} este trabalho apresentou trabalho melhores que Popoola et al. (2018), Eichie et al. (2017a), Popescu, Nafornta e Constantinou (2005), Sotiroudís e Siakavara (2015), Popescu et al. (2001), Anitzine, Argota e Fontán (2012), Benmus, Abboud e Shatter (2015) e Santos (2021) em relação aos trabalhos que calcularam essa métrica. Com relação à métrica R^2 este trabalho teve desempenho melhor que o trabalho dos autores Ayadi, Zineb e Tabbane (2017) que apresentaram essa métrica.

Como cada estudo foi realizado em um local diferente e com características geográficas diferentes, o fato de um trabalho obter uma métrica melhor que o outro pode não significar que este trabalho seja melhor. Outro fator determinante é de como foram feitas as coletas e se houve algum pré-processamento com os dados coletados, uma vez que é sabido que durante o procedimento de coleta alguns ruídos podem contaminar os dados. No caso, sabe-se que

os estudos pré processaram as amostras. Porém, quando comparado ao trabalho de Santos (2021), em que foram utilizados os mesmos dados coletados, este estudo apresentou melhores resultados em todas as métricas de avaliação de desempenho. Portanto, conclui-se que a utilização da PL calculada por um modelo de propagação tradicional na entrada da RNA melhora o desempenho da RNA para cálculo de PL.

9.2 CONCLUSÕES ACERCA DA OTIMIZAÇÃO DOS MODELOS DE PROPAGAÇÃO POR ALGORÍTIMOS GENÉTICOS E EVOLUÇÃO DIFERENCIAL

A proposta apresentada por este trabalho de otimizar os modelos Okumura-Hata, Cost 231, Egli e ECC-33 a partir dos algoritmos evolucionais, através de medições realizadas na cidade de Uberlândia, foram otimizados com sucesso no sentido de fornecer desempenho superior aos modelos empíricos tradicionais. Em todos os casos o processo de otimização melhorou o erro quadrático médio, principalmente quando foi adicionado aos modelos o parâmetro altitude comprovado na Tabela 8.13 e/ou visualmente na Figura 8.6. Em média para os modelos otimizados com AGs o EQM foi 7,1568 dB menor quando adicionado o parâmetro altitude, e com a ED o EQM foi 8,3968 dB menor quando adicionado o parâmetro altitude e por último a ED em média teve EQM 1,2371 dB menor do que o AG.

O EQM da previsão de perda de caminho foi melhorado para a média dos modelos empíricos em 13,1669 dB nos dados experimentais. Os resultados deste estudo mostraram que o modelo de PL otimizado é adequado para prever a perda de propagação neste cenário com uma antena receptora altura de 1,60 m e a antena transmissora fixa na estação de televisão.

A Tabela 8.13 evidencia a superioridade dos modelos otimizados, especialmente aqueles que consideram a altitude. O algoritmo ED demonstrou uma leve vantagem em relação ao AG, apresentando os menores valores de EQM. A análise comparativa revela que a adição da altitude não apenas melhorou o desempenho geral dos modelos, mas também ampliou a diferença em relação aos modelos otimizados sem essa variável. Dentre os modelos avaliados, o ECC-33 otimizado com ED e altitude destacou-se como o de melhor desempenho. Essa combinação mostrou-se altamente eficaz em ajustar os modelos aos dados reais, proporcionando previsões mais precisas da PL.

Além do mais, como discorrido nas equações apresentadas na seção resultados e discussões chegou-se em equações possibilitando o cálculo da PL, ou seja, ao contrário dos trabalhos que utilizaram redes neurais e/ou *machine learning*, não há dependência de hardwares e a possibilidade de calcular de forma analítica a PL.

Com base nos resultados empíricos, o método proposto é adequado para melhorar a aplicabilidade de todos os modelos de perda de caminho na cidade de Uberlândia e de forma análoga este método pode ser personalizado para qualquer região. A precisão da previsão do modelo proposto pode ser melhorada com base na qualidade da informação geográfica do local onde está a antena receptora.

Finalmente esta metodologia fornece equações para cada um dos modelos que proporcionou resultados mais fidedignos em relação aos resultados fornecidos pelos modelos de propagação tradicionais.

Como cada estudo foi realizado em um local diferente e com características geográficas diferentes, o fato de um trabalho obter uma métrica melhor que o outro pode não significar que este trabalho seja melhor. Outro fator determinante é de como foram feitas as coletas e se houve algum pré-processamento com os dados coletados, uma vez que é sabido que durante o procedimento de coleta alguns ruídos podem contaminar os dados. No caso, sabe-se que os estudos pré processaram as amostras. Porém, quando comparado ao trabalho de Santos (2021), em que foram utilizados os mesmos dados coletados, este estudo apresentou melhores resultados em todas as métricas de avaliação de desempenho. Portanto, conclui-se que a utilização da PL calculada por um modelo de propagação tradicional na entrada da RNA melhora o desempenho da RNA para cálculo de PL.

9.3 COMPARAÇÃO ENTRE OS RESULTADO OBTIDOS PELA RNA E OS ALGORITMOS EVOLUTIVOS

A tabela abaixo mostra os melhores resultados encontrados pelo aprimoramento realizado pela RNA e a otimização dos modelos através do AG e ED.

Tabela 9.1 – Comparação entre o Modelo Cost 231 e suas otimizações

Modelo	Metodologia	EM	EMA	EQM	E_{RMS}	σ	R^2
O. Hata	RNA	-0,01	2,26	8,61	2,93	1,87	0,93
ECC-33	RNA	-0,06	2,29	8,90	2,98	1,91	0,93
Egli	RNA	0,01	2,28	8,72	2,95	1,87	0,93
Cost 231	RNA	-0,01	2,31	8,88	2,98	1,88	0,93
O. Hata+Altitude	ED	-0,7926	5,0225	39,7452	6,3044	3,8107	0,6976
ECC-33+Altitude	ED	0,0008	4,6741	36,0985	6,0082	3,7753	0,7209
Egli+Altitude	ED	-0,0009	4,9416	39,117	6,2544	3,834	0,6976
Cost 231+Altitude	ED	-0,0009	4,9416	39,117	6,2544	3,834	0,6976

Nesta análise, fica evidente que os resultados fornecidos pela RNA geralmente apresentaram erros menores do que aqueles obtidos pela otimização. Essa observação era esperada, uma vez que as RNAs possuem a capacidade inerente de generalizar, ou seja, produzir resultados precisos mesmo com dados contaminados por ruído, o que as coloca em vantagem em relação às equações analíticas otimizadas. Além disso, como discutido no Capítulo 4, as RNAs empregadas neste estudo são de múltiplas camadas, o que as habilita a simular funções não lineares, como é o caso da PL.

Entretanto, o desempenho dos algoritmos evolutivos tenha sido inferior ao das RNAs, ao

considerarmos o E_{RMS} como parâmetro, percebe-se o melhor resultado encontrado pela ED (6,0082 dB) em relação ao melhor resultado encontrado pela RNA (2,93 dB) foi uma diferença de apenas 3,0782 dB. Portanto, esse resultado pode ser considerado satisfatório, uma vez que, por meio de equações analíticas, é possível calcular a PL sem a necessidade de recursos computacionais, ao contrário do método baseado em RNAs, os quais são dependentes do poder computacional, o qual nem sempre está disponível. Também há de se destacar que a RNA produziu melhor resultado quando teve como entrada o PL do modelo Okumura-Hata enquanto que os algoritmos evolutivos apresentaram o melhor resultado quando foi otimizado o modelo ECC-33.

9.4 PRINCIPAIS CONTRIBUIÇÕES

As principais contribuições deste trabalho são apresentadas da seguinte forma:

1. Desenvolvimento de Modelos Híbridos: Propor modelos híbridos para a predição da PL, combinando dados de potência coletados e os modelos tradicionais de propagação (Okumura-Hata, Cost 231, Egli, ECC-33 e ITU P.1546-6) com RNAs.
2. Análise de Configurações de RNA: Investigar o impacto da configuração da RNA, particularmente o número de neurônios na camada escondida, no desempenho dos modelos híbridos, identificando a configuração mais eficiente para cada modelo e apontando aquele que apresenta os melhores resultados.
3. Otimização por Algoritmos Evolutivos: Refinar os modelos híbridos (exceto o modelo ITU P.1546-6) por meio de algoritmos evolutivos, como AGs e ED, e estabelecer um ranqueamento dos modelos otimizados com base no desempenho obtido.
4. Incorporação da Altitude como Parâmetro: Introduzir a altitude como um parâmetro adicional nos cálculos de PL, desenvolvendo modelos preditivos capazes de gerar equações analíticas mais precisas do que as oferecidas pelos modelos de propagação tradicionais.
5. Estudo da Influência da Altitude: Examinar a relevância da altitude na estimativa da PL, comparando os resultados preditivos com medições práticas, destacando a importância deste parâmetro, que é frequentemente negligenciado nos modelos tradicionais, apesar de sua fácil obtenção por GPS ou APIs de georreferenciamento.

9.5 SUGESTÕES PARA TRABALHOS FUTUROS

Com base nos resultados obtidos e nas lacunas identificadas, propõem-se as seguintes sugestões para futuras pesquisas:

1. Expansão das áreas de medição:

- Realizar medições em outras áreas da cidade para validar o estudo atual.
- Ampliar a coleta de dados para cidades vizinhas, permitindo validar e/ou ajustar as técnicas desenvolvidas às particularidades de cada localidade.

2. Diversificação de cenários e condições:

- Realizar medições em diferentes frequências para explorar a aplicabilidade das técnicas a outras bandas de operação.
- Efetuar medições em diferentes épocas do ano e sob variadas condições climáticas, avaliando o impacto dessas variáveis no cálculo da PL.

3. Aprimoramento dos modelos de RNA:

- Inserir mais parâmetros de entrada na RNA, como altura média dos prédios e densidade de construção em cada região, para refinar a modelagem.
- Investigar o impacto de adicionar mais camadas escondidas na RNA, avaliando o impacto na precisão e desempenho do modelo.
- Testar diferentes algoritmos de treinamento para RNA, comparando sua eficácia na predição de PL.

4. Estudo sazonal do campo elétrico: Realizar medições específicas do campo elétrico em cada estação do ano, para identificar possíveis variações sazonais e incorporá-las nos modelos de predição.

REFERÊNCIAS

- ABHAYAWARDHANA, V.; WASSELL, I.; CROSBY, D.; SELLARS, M.; BROWN, M. Comparison of empirical propagation path loss models for fixed wireless access systems. In: **2005 IEEE 61st Vehicular Technology Conference**. [S.l.: s.n.], 2005. v. 1, p. 73–77 Vol. 1. DOI:<<https://doi.org/10.1109/VETECS.2005.1543252>>. Citado na pág. 41, 42.
- ANATEL. **Canais de Radiodifusão**. 2024. Disponível em <<https://sistemas.anatel.gov.br/se/public/view/b/srd.php>>. Acessado em 27 de dezembro de 2024. Citado na pág. 71.
- ANGELES, J. C. D.; DADIOS, E. P. Neural network-based path loss prediction for digital TV macrocells. In: **2015 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)**. [S.l.]: IEEE, 2015. DOI:<<https://doi.org/10.1109/hnicem.2015.7393223>>. Citado na pág. 30.
- ANITZINE, I. F.; ARGOTA, J. A. R.; FONTÁN, F. P. Influence of training set selection in artificial neural network-based propagation path loss predictions. **International Journal of Antennas and Propagation**, Hindawi Limited, v. 2012, p. 1–7, 2012. DOI:<<https://doi.org/10.1155/2012/351487>>. Citado na pág. 123.
- AYADI, M.; ZINEB, A. B.; TABBANE, S. A UHF path loss model using learning machine for heterogeneous networks. **IEEE Transactions on Antennas and Propagation**, Institute of Electrical and Electronics Engineers (IEEE), v. 65, n. 7, p. 3675–3683, jul. 2017. DOI:<<https://doi.org/10.1109/tap.2017.2705112>>. Citado na pág. 23, 24, 31, 123.
- BENMUS, T. A.; ABOUD, R.; SHATTER, M. K. Neural network approach to model the propagation path loss for great tripoli area at 900, 1800, and 2100 MHz bands. In: **2015 16th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)**. [S.l.]: IEEE, 2015. DOI:<<https://doi.org/10.1109/sta.2015.7505236>>. Citado na pág. 30, 123.
- BHUVANESHWARI, A.; HEMALATHA, R.; SATYASAVITHRI, T. Performance evaluation of dynamic neural networks for mobile radio path loss prediction. In: **2016 IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics Engineering (UPCON)**. [S.l.]: IEEE, 2016. DOI:<<https://doi.org/10.1109/upcon.2016.7894698>>. Citado na pág. 31, 34.
- BOTELHO, A. L. P. **Otimização da escolha de modelo de propagação por medição de campo e inteligência artificial**. Dissertação (Mestrado) — Universidade Presbiteriana Mackenzie, São Paulo, 2019. Disponível em:<<https://dspace.mackenzie.br/handle/10899/24493>>. Acessado em 27 de dezembro de 2024. Citado na pág. 25.
- CAVALCANTI, B. J.; D'ASSUNÇÃO, A. G.; MENDONÇA, L. M. Optimizing empirical propagation models for lte and lte-a using genetic algorithms at 879 mhz. In: **2017 IEEE-APS Topical Conference on Antennas and Propagation in Wireless Communications (APWC)**. [S.l.: s.n.], 2017. p. 312–315. DOI:<<http://dx.doi.org/10.1109/APWC.2017.8062313>>. Citado na pág. 32.
- CERRI, G.; CINALLI, M.; MICHETTI, F.; RUSSO, P. Feed forward neural networks for path loss prediction in urban environment. **IEEE Transactions on Antennas and Propagation**,

Institute of Electrical and Electronics Engineers (IEEE), v. 52, n. 11, p. 3137–3139, nov. 2004. DOI:<<https://doi.org/10.1109/tap.2004.835252>>. Citado na pág. 28, 95, 123.

EICHIE, J. O.; OYEDUM, O. D.; AJEWOLE, M. O.; AIBINU, A. M. Artificial neural network model for the determination of GSM rxlevel from atmospheric parameters. **Engineering Science and Technology, an International Journal**, Elsevier BV, v. 20, n. 2, p. 795–804, abr. 2017. DOI:<<https://doi.org/10.1016/j.jestch.2016.11.002>>. Citado na pág. 32, 33, 95, 123.

_____. COMPARATIVE ANALYSIS OF BASIC MODELS AND ARTIFICIAL NEURAL NETWORK BASED MODEL FOR PATH LOSS PREDICTION. **Progress In Electromagnetics Research M**, The Electromagnetics Academy, v. 61, p. 133–146, 2017. DOI:<<https://doi.org/10.2528/pierm17060601>>. Citado na pág. 28, 95.

FARUK, N.; ABDULRASHEED, I. Y.; SURAJUDEEN-BAKINDE, N.; ADETIBA, E.; OLOYEDE, A.; ABDULKARIM, A.; SOWANDE, O.; IFIJEH, A.; ATAYERO, P. A. Large-scale radio propagation path loss measurements and predictions in the vhf and uhf bands. **Heliyon**, v. 7, p. e07298, 06 2021. DOI:<<https://doi.org/10.1016/j.heliyon.2021.e07298>>. Citado na pág. 21.

FAUSETT, L. **Fundamentals of neural networks : architectures, algorithms, and applications**. Englewood Cliffs, NJ Delhi Dorling Kindersley: Prentice-Hall, 1994. ISBN 9780133341867. Citado na pág. 46, 53, 54, 55.

FENG, C.; BOZHANG, L.; SHIJI, Y.; YING, S. A new prediction method for propagation loss in VHF. In: **2015 IEEE 6th International Symposium on Microwave, Antenna, Propagation, and EMC Technologies (MAPE)**. [S.l.]: IEEE, 2015. DOI:<<https://doi.org/10.1109/mape.2015.7510253>>. Citado na pág. 22.

FERREIRA, G. P.; MATOS, L. J.; SILVA, J. M. M. Improvement of outdoor signal strength prediction in UHF band by artificial neural network. **IEEE Transactions on Antennas and Propagation**, Institute of Electrical and Electronics Engineers (IEEE), v. 64, n. 12, p. 5404–5410, dez. 2016. DOI:<<https://doi.org/10.1109/tap.2016.2617379>>. Citado na pág. 31.

GARAH, M.; DJOUANE, L.; OUDIRA, H.; HAMDIKEN, N. Path loss models optimization for mobile communication in different areas. **Indonesian Journal of Electrical Engineering and Computer Science**, v. 3, p. 126, 06 2016. DOI:<<http://dx.doi.org/10.11591/ijeecs.v3.i1.pp126-135>>. Citado na pág. 31, 33.

GÓMEZ-PÉREZ, P.; CREGO-GARCÍA, M.; CUIÑAS, I.; CALDEIRINHA, R. F. Modeling and inferring the attenuation induced by vegetation barriers at 2g/3g/4g cellular bands using artificial neural networks. **Measurement**, Elsevier BV, v. 98, p. 262–275, fev. 2017. DOI:<<https://doi.org/10.1016/j.measurement.2016.12.014>>. Citado na pág. 32, 33.

HAGAN, M.; MENHAJ, M. Training feedforward networks with the marquardt algorithm. **IEEE Transactions on Neural Networks**, Institute of Electrical and Electronics Engineers (IEEE), v. 5, n. 6, p. 989–993, 1994. DOI:<<https://doi.org/10.1109/72.329697>>. Citado na pág. 60.

HARRIS, C. **ATLAS ISDB-T Transmitter**. [S.l.], 2007. Citado na pág. 70.

HASSANAT, A.; ALMOHAMMADI, K.; ALKAFaweEN, E.; ABUNAWAS, E.; HAMMOURI, A.; PRASATH, V. B. S. Choosing mutation and crossover ratios for genetic algorithms—a review with a new dynamic approach. **Information**, v. 10, n. 12, 2019. ISSN 2078-2489. DOI:<<https://doi.org/10.3390/info10120390>>. Citado na pág. 66.

HAYKIN, S. S. **Neural networks and learning machines**. 3^a. ed. Upper Saddle River, NJ: Pearson Education, 2009. Citado na pág. 46, 47, 48, 53, 54.

IBGE. **Censo Brasileiro de 2022**. Rio de Janeiro(cidade): IBGE, 2022. Citado na pág. 69.

IEEE, C. I. S. **What is Computational Intelligence?** 2024. Disponível em:<<https://www.citedrive.com/pt/blog/how-to-cite-a-website-in-bibtex/>> Acessado em 02 de maio de 2025. Citado na pág. 24, 25.

ITU, I. T. U. **Method for point-to-area predictions for terrestrial services in the frequency range 30 MHz to 4 000 MHz**. [S.l.], 2019. Disponível em: <<https://www.itu.int/rec/r-rec-p.1546/en>> Acessado em 27/12/2024. Citado na pág. 43, 44.

JAWHLY, T.; TIWARI, R. C. The special case of egli and hata model optimization using least-square approximation method. **SN Applied Sciences**, Springer Science and Business Media LLC, v. 2, n. 7, jun 2020. DOI:<<https://doi.org/10.1007/s42452-020-3061-0>>. Citado na pág. 41.

KALAKH, M.; KANDIL, N.; HAKEM, N. Neural networks model of an UWB channel path loss in a mine environment. In: **2012 IEEE 75th Vehicular Technology Conference (VTC Spring)**. [S.l.]: IEEE, 2012. DOI:<<https://doi.org/10.1109/vetecs.2012.6240318>>. Citado na pág. 28, 29.

KALEYBAR, H. J.; DAVOODI, M.; BRENNAN, M.; ZANINELLI, D. Applications of genetic algorithm and its variants in rail vehicle systems: A bibliometric analysis and comprehensive review. **IEEE Access**, v. 11, p. 68972–68993, 2023. DOI:<<https://doi.org/10.1109/ACCESS.2023.3292790>>. Citado na pág. 63, 64, 65.

LIU, J.; JIN, X.; DONG, F.; HE, L.; LIU, H. Fading channel modelling using single-hidden layer feedforward neural networks. **Multidimensional Systems and Signal Processing**, v. 28, 07 2017. DOI:<<https://doi.org/10.1007/s11045-015-0380-1>>. Citado na pág. 31.

MATLAB. **version 7.10.0 (R2010a)**. Natick, Massachusetts: The MathWorks Inc., 2010. Citado na pág. 59.

MOHAMED, A. W.; MOHAMED, A. K. Adaptive guided differential evolution algorithm with novel mutation for numerical optimization. **International Journal of Machine Learning and Cybernetics**, Springer Science and Business Media LLC, v. 10, n. 2, p. 253–277, ago. 2017. ISSN 1868-808X. DOI:<<https://doi.org/10.1007/s13042-017-0711-7>>. Citado na pág. 67.

MOLLEL, M.; KISANGIRI, M. Comparison of empirical propagation path loss models for mobile communication. **Computer Engineering and Intelligent Systems**, v. 5, 01 2014. Citado na pág. 40.

NAFEA, S.; HAMZA, E. Path loss optimization in wimax network using genetic algorithm. **Iraqi Journal of Computers, Communications, Control and Systems Engineering**, University of Technology, v. 20, n. 1, p. 24–30, 2020. ISSN 1811-9212. Disponível em: <https://ijccce.uotechnology.edu.iq/article_168046.html>. Citado na pág. 33.

OSTLIN, E.; ZEPERNICK, H.; SUZUKI, H. Macrocell path-loss prediction using artificial neural networks. **IEEE Transactions on Vehicular Technology**, Institute of Electrical and Electronics Engineers (IEEE), v. 59, n. 6, p. 2735–2747, jul. 2010. DOI:<<https://doi.org/10.1109/tvt.2010.2050502>>. Citado na pág. 29, 33, 34, 60.

PIACENTINI, M.; RINALDI, F. Path loss prediction in urban environment using learning machines and dimensionality reduction techniques. **Computational Management Science**, v. 8, p. 371–385, 11 2011. DOI:<<https://doi.org/10.1007/s10287-010-0121-8>>. Citado na pág. 29, 33.

POLÁKOVÁ, R.; TVRDÍK, J.; BUJOK, P. Differential evolution with adaptive mechanism of population size according to current population diversity. **Swarm and Evolutionary Computation**, v. 50, p. 100519, 2019. ISSN 2210-6502. DOI:<<https://doi.org/10.1016/j.swevo.2019.03.014>>. Citado na pág. 66.

POPESCU, I.; NAFOMITA, I.; CONSTANTINOU, P.; KANATAS, A.; MORAITIS, N. Neural networks applications for the prediction of propagation path loss in urban environments. In: **IEEE VTS 53rd Vehicular Technology Conference, Spring 2001. Proceedings (Cat. No.01CH37202)**. [S.l.: s.n.], 2001. v. 1, p. 387–391 vol.1. DOI:<<https://doi.org/10.1109/VETECS.2001.944870>>. Citado na pág. 28, 33, 59, 123.

POPESCU, I.; NAFORNITA, I.; CONSTANTINOU, P. Comparison of neural network models for path loss prediction. In: **WiMob'2005), IEEE International Conference on Wireless And Mobile Computing, Networking And Communications, 2005**. [S.l.: s.n.], 2005. v. 1, p. 44–49. DOI:<<https://doi.org/10.1109/WIMOB.2005.1512814>>. Citado na pág. 28, 123.

POPOOLA, S. I.; ADETIBA, E.; ATAYERO, A. A.; FARUK, N.; CALAFATE, C. T. Optimal model for path loss predictions using feed-forward neural networks. **Cogent Engineering**, Informa UK Limited, v. 5, n. 1, p. 1444345, jan. 2018. DOI:<<https://doi.org/10.1080/23311916.2018.1444345>>. Citado na pág. 32, 34, 95, 123.

RAPPAPORT, T. **Wireless communications : principles and practice**. Upper Saddle River, N.J: Prentice Hall PTR, 2002. Citado na pág. 35, 36, 37, 38, 39, 40.

RFS. **1-5/8 HELIFLEX® Air Dielectric Coaxial Cable**. [S.l.], 2007. Citado na pág. 70.

SANTOS, T. **Análise da variabilidade da intensidade do campo elétrico na faixa de 566 a 572 MHz na cidade de Uberlândia**. Tese (Doutorado) — Universidade Federal de Uberlândia, Uberlândia, 2021. DOI:<<https://doi.org/10.14393/ufu.te.2021.6016>>. Citado na pág. 33, 72, 123, 124, 125.

SASAKI, M.; INOMATA, M.; YAMADA, W.; ONIZAWA, T. Path loss models for wireless access network systems using high frequency bands. **NTT Technical Review**, v. 14, n. 12, 12 2016. DOI:<<https://doi.org/10.53829/ntr201612ra2>>. Citado na pág. 23.

SEYBOLD. **Intro RF Propagation**. Hoboken, New Jersey: John Wiley Sons, 2005. DOI:<<https://doi.org/https://doi.org/10.1002/0471743690>>. Citado na pág. 35.

SHAO, Y.; TAFF, G. N.; WALSH, S. J. Comparison of early stopping criteria for neural-network-based subpixel classification. **IEEE Geoscience and Remote Sensing Letters**, Institute of Electrical and Electronics Engineers (IEEE), v. 8, n. 1, p. 113–117, jan. 2011. DOI:<<https://doi.org/10.1109/lgrs.2010.2052782>>. Citado na pág. 58.

SILVA, V. J. da; SILVA, C. R. da; ALMEIDA, L. D. S.; SILVA, C. R. da; CARVALHO, H. D. P.; CAMARGO, R. de. Mobile transect for identification of intra-urban heat islands in uberlandia, brazil. **Ambiente e Agua - An Interdisciplinary Journal of Applied Science**, Instituto de Pesquisas Ambientais em Bacias Hidrograficas (IPABHi), v. 13, n. 4, p. 1, jul. 2018. DOI:<<https://doi.org/10.4136/ambi-agua.2187>>. Citado na pág. 69.

SIZUN, H. **Radio Wave Propagation for Telecommunication Applications**. Belfort, France: Springer, 2005. Citado na pág. 36.

SOTIROUDIS, S. P.; GOUDOS, S. K.; GOTSIS, K. A.; SIAKAVARA, K.; SAHALOS, J. N. Application of a composite differential evolution algorithm in optimal neural network design for propagation path-loss prediction in mobile communication systems. **IEEE Antennas and Wireless Propagation Letters**, Institute of Electrical and Electronics Engineers (IEEE), v. 12, p. 364–367, 2013. DOI:<<https://doi.org/10.1109/lawp.2013.2251994>>. Citado na pág. 30, 33, 123.

SOTIROUDIS, S. P.; SIAKAVARA, K. Mobile radio propagation path loss prediction using artificial neural networks with optimal input information for urban environments. **AEU - International Journal of Electronics and Communications**, Elsevier BV, v. 69, n. 10, p. 1453–1463, out. 2015. DOI:<<https://doi.org/10.1016/j.aeue.2015.06.014>>. Citado na pág. 28, 123.

STORN, R.; PRICE, K. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. **Journal of Global Optimization**, Springer Science and Business Media LLC, v. 11, n. 4, p. 341–359, 1997. ISSN 0925-5001. DOI:<<http://dx.doi.org/10.1023/A:1008202821328>>. Citado na pág. 66.

YANG, X.-S. Chapter 5 - genetic algorithms. In: YANG, X.-S. (Ed.). **Nature-Inspired Optimization Algorithms**. Oxford: Elsevier, 2014. p. 77–87. ISBN 978-0-12-416743-8. DOI:<<https://doi.org/10.1016/B978-0-12-416743-8.00005-1>>. Citado na pág. 62.

ZAAROUR, N.; KANDIL, N.; HAKEM, N.; DESPINS, C. Comparative experimental study on modeling the path loss of an UWB channel in a mine environment using MLP and RBF neural networks. In: **2012 International Conference on Wireless Communications in Underground and Confined Areas**. [S.l.]: IEEE, 2012. DOI:<<https://doi.org/10.1109/icwcuca.2012.6402503>>. Citado na pág. 29.

ZHANG, Z.; CAI, Y.; ZHANG, D. Solving ordinary differential equations with adaptive differential evolution. **IEEE Access**, v. 8, p. 128908–128922, 2020. DOI:<<https://doi.org/10.1109/ACCESS.2020.3008823>>. Citado na pág. 67.

ZINEB, A. B.; AYADI, M. A multi-wall and multi-frequency indoor path loss prediction model using artificial neural networks. **Arabian Journal for Science and Engineering**, Springer Science and Business Media LLC, v. 41, n. 3, p. 987–996, nov. 2015. DOI:<<https://doi.org/10.1007/s13369-015-1949-6>>. Citado na pág. 30.

APÊNDICE A

Código-fonte do script utilizado para tratar as amostras que foram coletadas na mesma posição geográfica e salvar em um novo banco de dados.

```
1
2 clear all
3 close all
4 clc
5 dados=xlsread('dados.xlsx');
6 unicos=unique(dados(:,1:2),'rows');
7
8 for i=1:length(unicos)
9 a=find((unicos(i,1)==dados(:,1))\&(unicos(i,1)==dados(:,1)));
10 unicos(i,3)=mean(dados(a,3));
11 unicos(i,4)=mean(dados(a,4));
12 unicos(i,5)=mean(dados(a,5));
13 end
14
15 xlswrite('dadosunicos.xlsx',unicos)
```

APÊNDICE B

Código fonte utilizado para o cálculo da perda de percurso em cada um dos modelos de propagação utilizado e treinamento da Rede Neural Artificial.

```

1 clear all
2 close all
3 clc
4
5 dados=xlsread("dadosunicos1.xlsx");
6
7 %dados do transmissor
8 f=569.142857; %frequencia
9 antenalocal=[-18.885 -48.2617];
10 PTx=2500*0.3;Gt=9.6684 ;
11 Ptdb=10*log10(PTx)+Gt-1.27 ;
12 local=dados(:,1:2);
13
14 for i=1:length(dados)
15     dados(i,8)=azimuth(antenalocal(1),antenalocal(2),
16     dados(i,1),dados(i,2));
17 end
18
19 dados(:,7)=Ptdb-(dados(:,7)-30);
20
21 %praparacao dos dados para predicao
22 input=[dados(:,3)-(937+64) dados(:,6) dados(:,8)];
23 output=dados(:,7);
24 %predicao dos dados
25 %PLrna=rna160ok(input); %predicao por RNA
26 PLfs= 32.5+20*log10(f)+20*log10(dados(:,6));
27 PLcost=cost231(f,dados(:,6),64,2);
28 PLoH=okumurahata(dados(:,6),f,64,2);
29 PLegli=egli(f,dados(:,6),64,2);
30 PLecc33=ecc33(f,dados(:,6),64,2);
31
32 for i=1:length(dados(:,6))
33     d_v=dados(i,6);
34     hrter=dados(i,3)+2;
35     [E, L] = P1546FieldStrMixed(f,50,(937)-852,1.5,15,'Urban',d_v,
36     'Land',0,50,[],2.5*0.3,64,1.5,[],[],937,hrter,[],[],0,0);
37     PLp1546(i,1)=L;
38 end
39 PL=[PLfs PLcost PLegli PLecc33 PLoH PLp1546];
40 PLname={'Free Space Loss','Cost-231','Egli','ECC-33'
41 ,'Okumura-Hata','ITU-P.1546'};

```

```

42
43 a=1
44 for j=1:6
45   ninput=[PL(:,j) input];
46   for i=10:10:200
47     net = feedforwardnet([i] );
48     net.divideParam.trainRatio = 0.5;
49     net.divideParam.valRatio = 0.25;
50     net.divideParam.testRatio = 0.25;
51     net.layers{1}.transferFcn = 'tansig';
52     net.layers{2}.transferFcn = 'tansig';
53     net.trainParam.max_fail=10;
54     [net,tr] = train(net,ninput',output');
55
56     y = net(ninput');
57     mse_train = tr.perf(tr.best_epoch + 1);
58     mse_val = tr.vperf(tr.best_epoch + 1);
59     mse_test = tr.tperf(tr.best_epoch + 1);
60
61     desemp(a,:)=[j i mse_train mse_val mse_test perform(net,output',y)]
62     a=a+1;
63     genFunction(net,'rna'+string(i),'MatrixOnly','yes')
64   end
65 end
66
67 function [L] = cost231(f,d,ht,hr)
68 %Funcao que calcula a perda de percurso para o modelo Cost231
69 a=(1.1*log10(f)-0.7)*hr-(1.56*log10(f)-0.8);
70 L=46.3+33.9*log10(f)-13.82*log10(ht)-a+(44.9-6.55*log10(hr))*log10(d);
71 end
72
73 function L = egli(f,d,ht,hr)
74 %Funcao que calcula a perda de percurso para o modelo Egli
75 po=40*log10(d)-20*log10(ht)-10*log10(hr);
76 L=20*log10(f)+po+85.9;
77 end
78
79 function[L] = ecc33(f,d,ht,hr)
80 %Funcao que calcula a perda de percurso para o modelo ECC33
81 Afs= 92.4+20*log10(d)+20*log10(f/1000);
82 Abm= 20.41 + 9.83*log10(d)+7.894*log10(f)+9.56*(log10(f/1000)).^2;
83 Gb=log10(ht/200)*(13.958+5.8*(log10(d)).^2);
84 %Gr=(42.57+13.7*log10(f/1000))*(log10(hr)-0.585);
85 Gr=0.759*hr-1.862;
86 L=Afs+Abm-Gb-Gr;
87 end
88

```

```

89 function [L] = okumurahata(distancia,frequencia,
90 alturatransmissor,alturareceptor)
91 %Funcao que calcula a perda de percurso para o modelo Okumura-Hata
92 a=(1.1*log10(frequencia)-0.7)*alturareceptor-
93 (1.56*log10(frequencia)-0.8);
94 L=69.55+26.16*log10(frequencia)-13.83*log10(alturatransmissor)
95 -a+(44.9-6.55*log10(alturareceptor))*log10(distancia);
96
97 end
98
99 function [emedio,mi,eqm,erms,sigma,r2] = erros(medido,previsto)
100 %funcao que calcula os erros
101 x=medido;
102 y=previsto;
103
104 N=length(x);
105 Eabs=abs(x-y);%erro medio absoluto
106 mi=mean(Eabs);
107 emedio=mean(x-y);%erro medio
108 eqm=mean((x-y).^2);%erro quadratico medio
109 erms=sqrt(eqm);%erro rms
110
111 sigma=(sum(Eabs.^2)-N*mi^2)/(N-1);
112 sigma=sqrt(sigma); % desvio padrao
113
114 r2n=(sum(x.*y)-sum(x)*sum(y)/N)^2;
115 r2d=(sum(x.^2)-sum(x)^2/N)*(sum(y.^2)-sum(y)^2/N);
116 r2=r2n/r2d; % coeficiente de determinacao
117 end

```

APÊNDICE C

Código fonte utilizado para o otimização dos modelos de PL utilizando AG.

```

1 import numpy as np
2 from geneticalgorithm import geneticalgorithm as ga
3 import pandas as pd
4
5
6
7 PTx = 2500 * 0.3
8 Gt = 9.6684
9 Ptdb = 10 * np.log10(PTx) + Gt - 1.27
10 f0 = 569.142857
11 hr = 2
12 ht = 64
13 df = pd.read_excel('dadosunicos1kmmais.xlsx')
14 df = df.values
15 PL = Ptdb - df[:, 6] + 30
16 dist = df[:, 5]
17 altitude= 1001-df[:, 2]
18 #print(dist)
19
20 def newmodel(X):
21     OF= X[0]*(altitude-X[1])
22     return np.square(np.subtract(PL,OF)).mean()
23
24
25
26 def fslajus(X): # Free Space Loss Ajusted
27     #OF = 0
28     #for i in range(100):
29         #j = random.randrange(0, len(dist))
30         #PL=32.5+20*np.log10(f0)+20*log10(dist)
31     OF= (X[0]*32.44 + X[1] * 20 * np.log10(f0) + X[2]*20* np.log10(dist))
32     #OF=X[0]*(altitude)+X[1]
33     return np.square(np.subtract(PL,OF)).mean()
34
35 def fslajush(X): # Free Space Loss Ajusted with elevation
36     #OF = 0
37     #for i in range(100):
38         #j = random.randrange(0, len(dist))
39         #PL=32.5+20*np.log10(f0)+20*log10(dist)
40     OF= X[0]*32.44 + X[1]*20 * np.log10(f0) + X[2]*20*np.log10(dist)+X[3]*(
        altitude)
41     return np.square(np.subtract(PL,OF)).mean()

```

```

42
43 def ohataajust(X):
44     a=(X[5]*1.1*np.log10(f0)-X[6]*0.7)*hr-(X[7]*1.56*np.log10(f0)-X[8]*0.8)
45     OF= (X[0]*69.55)+(X[1]*26.16)*np.log10(f0)-(X[2]*13.82)*np.log10(ht)-a+((
46         X[3]*44.9)-(X[4]*6.55)*np.log10(ht))*np.log10(dist)
47     return np.square(np.subtract(PL,OF)).mean()
48
49
50 def ohataajusth(X):
51     a=(X[5]*1.1*np.log10(f0)-X[6]*0.7)*hr-(X[7]*1.56*np.log10(f0)-X[8]*0.8)
52     OF= (X[0]*69.55)+(X[1]*26.16)*np.log10(f0)-(X[2]*13.82)*np.log10(ht)-a+((
53         X[3]*44.9)-(X[4]*6.55)*np.log10(ht))*np.log10(dist)+X[9]*(altitude)
54     return np.square(np.subtract(PL,OF)).mean()
55
56 def ecc33(X):
57     afs=X[0]*92.4+X[1]*20*np.log10(dist)+X[2]*20*np.log10(f0/1000)
58     abm=X[3]*20.41+X[4]*9.83*np.log10(dist)+X[5]*7.89*np.log10(f0/1000)+X
59         [6]*9.56*(np.log10(f0/1000))**2
60     gb=np.log10(ht/200)*(X[7]*13.958+X[8]*5.8*(np.log10(dist))**2)
61     #gr=(42.57+13.7*np.log10(f0/1000))*(np.log10(hr)-0.585)
62     gr=X[9]*0.759*hr-X[10]*1.862
63     OF=afs+abm+gb+gr
64     return np.square(np.subtract(PL,OF)).mean()
65
66 def ecc33h(X):
67     afs=X[0]*92.4+X[1]*20*np.log10(dist)+X[2]*20*np.log10(f0/1000)
68     abm=X[3]*20.41+X[4]*9.83*np.log10(dist)+X[5]*7.89*np.log10(f0/1000)+X
69         [6]*9.56*(np.log10(f0/1000))**2
70     gb=np.log10(ht/200)*(X[7]*13.958+X[8]*5.8*(np.log10(dist))**2)
71     #gr=(42.57+13.7*np.log10(f0/1000))*(np.log10(hr)-0.585)
72     gr=X[9]*0.759*hr-X[10]*1.862
73     h= X[11]*(altitude)
74     OF=afs+abm+gb+gr+h
75     return np.square(np.subtract(PL,OF)).mean()
76
77 def eglih(X):
78     OF= X[0]*20*np.log10(f0)+X[1]*40*np.log10(dist)-X[2]*20*np.log10(ht)-X
79         [3]*10*np.log10(hr)+X[4]*76.3+X[5]*(altitude)
80     return np.square(np.subtract(PL,OF)).mean()
81
82 def egli(X):
83     OF= X[0]*20*np.log10(f0)+X[1]*40*np.log10(dist)-X[2]*20*np.log10(ht)-X
84         [3]*10*np.log10(hr)+X[4]*76.3
85     return np.square(np.subtract(PL,OF)).mean()
86
87 def cost231(X):

```

```

83 a=(X[5]*1.1*np.log10(f0)-X[6]*0.7)*hr-(X[7]*1.56*np.log10(f0)-X[8]*0.8)
84 OF=X[0]*46.3+X[1]*33.9*np.log10(f0)-X[2]*13.82*np.log10(ht)-a+(X[3]*44.9-
    X[4]*6.55*np.log10(ht))*np.log10(dist)+X[9]*(altitude)
85 return np.square(np.subtract(PL,OF)).mean()
86
87 dimension1=10
88 f = ecc33
89 #varbound = np.array([[-100, 100],[-20, 20],[-20, 20],[-20, 20]] )
    #,[-100,100]])
90 #varbound = np.array([[-20, 20],[-20, 20],[-20, 20],[-20, 20],[-20, 20]])
    #,[-100,100]])
91 varbound = np.array([[-100, 100]]*dimension1) #,[-100,100]])
92
93 algorithm_param = {'max_num_iteration': 10000,\
94                    'population_size':50,\
95                    'mutation_probability':0.25,\
96                    'elit_ratio': 0.01,\
97                    'crossover_probability': 0.8,\
98                    'parents_portion': 0.5,\
99                    'crossover_type':'two_point',\
100                    'max_iteration_without_improv': 3000}
101
102 model = ga(function=f,
103            dimension=dimension1,
104            variable_type='real',
105            variable_boundaries=varbound,
106            algorithm_parameters=algorithm_param)
107
108 model.run()

```

APÊNDICE D

Código fonte utilizado para o otimização dos modelos de PL utilizando ED.

```

1 # differential evolution global optimization for the ackley multimodal
  objective function
2 from scipy.optimize import differential_evolution
3 import openpyxl
4 import numpy as np
5 #from numpy.random import rand
6 from numpy import exp
7 from numpy import sqrt
8 from numpy import cos
9 from numpy import e
10 from numpy import pi
11 import pandas as pd
12
13 #PTx = 2500 * 0.3
14 #Gt = 9.6684
15 #Ptdb = 10 * np.log10(PTx) + Gt - 1.27
16 #f0 = 569.142857
17 #hr = 2
18 #ht = 64
19 #df = pd.read_excel('ari.xlsx')
20 #df = df.values
21 #PL = Ptdb - df[:, 6] + 30
22 #dist = df[:, 5]
23 #altitude= 1001-df[:, 2]
24
25 PTx = 80 * 0.3
26 Gt = 6.4
27 Ptdb = 10 * np.log10(PTx) + Gt - 1.58
28 f0 = 575.142857
29 hr = 2
30 ht = 48
31 df = pd.read_excel('ari.xlsx')
32 df = df.values
33 PL = Ptdb - df[:, 6] + 30
34 dist = df[:, 5]
35 altitude= 1001-df[:, 2]
36 # objective function
37 #def objective(v):
38 # x, y = v
39 # return -20.0 * exp(-0.2 * sqrt(0.5 * (x**2 + y**2))) - exp(0.5 * (cos(2 *
  pi * x) + cos(2 * pi * y))) + e + 20
40

```



```

41 #def objective(v):
42     #x1, x2, x3,x4= v
43     #OF= (x1*32.44 + x2 * 20 * np.log10(f0) + x3 *20* np.log10(dist))+x4*(
         altitude)
44     #return np.square(np.subtract(PL,OF )).mean()
45
46
47 #def objective(v):
48     # x0, x1, x2, x3, x4,x5, x6, x7, x8,x9= v
49     # a=(x5*1.1*np.log10(f0)-x6*0.7)*hr-(x7*1.56*np.log10(f0)-x8*0.8)
50     # OF= (x0*69.55)+(x1*26.16)*np.log10(f0)-(x2*13.82)*np.log10(ht)-a+((x3
         *44.9)-(x4*6.55)*np.log10(ht))*np.log10(dist)+x9*altitude
51     # return np.square(np.subtract(PL,OF )).mean()
52
53 ##def objective(v):
54     ## x0,x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11 =v
55     ## afs=x0*92.4+x1*20*np.log10(dist)+x2*20*np.log10(f0/1000)
56     ## abm=x3*20.41+x4*9.83*np.log10(dist)+x5*7.89*np.log10(f0/1000)+x6*9.56*(
         np.log10(f0/1000))*2
57     ## #abm=20.41+9.83*np.log10(dist)+7.89*np.log10(f0/1000)+9.56*(np.log10(f0
         /1000))*2
58     ## gb=np.log10(ht/200)*(x7*13.958+x8*5.8*(np.log10(dist))*2)
59     ## #gb=np.log10(ht/200)*(13.958+5.8*(np.log10(dist))*2)
60     ## # gr=(42.57+13.7*np.log10(f0/1000))*(np.log10(hr)-0.585)
61     ## gr=x9*0.759*hr-x10*1.862
62     ## # gr=0.759*hr-1.862
63     ## h= x11*altitude
64     ## OF=afs+abm+gb+gr+h
65     ## return np.square(np.subtract(PL,OF)).mean()
66 # define range for input
67
68 #def objective(v):
69     # x0, x1, x2, x3, x4=v
70     #OF= x0*20*np.log10(f0)+x1*40*np.log10(dist)-x2*20*np.log10(ht)-x3*10*np.
         log10(hr)+x4*76.3#+x5*(altitude)
71     #return np.square(np.subtract(PL,OF)).mean()
72
73 def objective(v):
74     x0, x1, x2, x3, x4,x5, x6, x7, x8,x9= v
75     a=(x5*1.1*np.log10(f0)-x6*0.7)*hr-(x7*1.56*np.log10(f0)-x8*0.8)
76     OF= (x0*69.55)+(x1*26.16)*np.log10(f0)-(x2*13.82)*np.log10(ht)-a+((x3
         *44.9)-(x4*6.55)*np.log10(ht))*np.log10(dist)#+x9*altitude
77     return np.square(np.subtract(PL,OF )).mean()
78
79
80 #def objective(v):
81     # x8,x9= v

```

```

82 # a=(1.1*np.log10(f0)-0.7)*hr-(1.56*np.log10(f0)-0.8)
83 # OF=46.3+33.9*np.log10(f0)-13.82*np.log10(ht)-a+(44.9-6.55*np.log10(ht))*
    np.log10(dist)+(x9*(altitude+x8))
84 # return np.square(np.subtract(PL,OF)).mean()
85
86 r_min, r_max = -100, 100
87 # define the bounds on the search
88 bounds = [[r_min, r_max]]*10
89
90
91 # perform the differential evolution search
92 result = differential_evolution(objective, bounds, popsize=15)
93 # summarize the result
94 print('Status : %s' % result['message'])
95 print('Total Evaluations: %d' % result['nfev'])
96 # evaluate solution
97 solution = result['x']
98 evaluation = objective(solution)
99 print('Solution: f(%s) = %.5f' % (solution, evaluation))
100 print(result)

```

ANEXO A

Relatório do Canal

Id solicitação: 57dbab8aa0126

Informações da Entidade

Dados da Entidade	
Nome da Entidade: RADIO TELEVISAO DE UBERLANDIA LTDA	
Nome Fantasia: TV INTEGRAÇÃO	
Telefone: (34) 3218-3420	E-mail: fiscal@tvintegracao.com.br
CNPJ: 25.631.672/0001-26	Número do Fistel: 50405865317
Tipo Usuário: Adm Privada	Tipo Taxa: Integral
Data do contrato: 05/10/1992	Serviço: 247 - Geradora de Radiodifusão de Sons e Imagens - Digital
Carater: Primário	Local específico:
Rede:	Categoria da Estação: Principal
Observações: ATO N° 299, DE 22/01/2009, PUBLICADO NO DOU. DE 23/01/2009.	

Endereço Sede		
Logradouro: Avenida Maranhão	Complemento:	
Bairro: Umuarama	Numero: 1666	
Município: Uberlândia	UF: MG	CEP: 38405318

Endereço Correspondência		
Logradouro:	Complemento:	
Bairro:	Numero:	
Município:	UF:	CEP:

Endereço do Transmissor		
Logradouro: Rio Grande do Norte	Complemento:	
Bairro: Umuarama	Numero: 1069	
Município: Uberlândia	UF: MG	CEP: 38405321

Endereço do Estúdio Principal		
Logradouro: Rio Grande do Norte	Complemento:	
Bairro: Umuarama	Numero: 1069	
Município: Uberlândia	UF: MG	CEP: 38405321

Endereço do Estúdio Auxiliar		
Logradouro:	Complemento:	
Bairro:	Numero:	
Município:	UF:	CEP:

Informações do Plano Básico

Localização			
Município: Uberlândia		UF: MG	
Parâmetros Técnicos			
Canal: 30	Frequência: 569 MHz	Classe: A	ERP Máxima: 23.4981kW
HCI: 64 m	Pareamento: 31023	Decalagem:	Fase: 2

Informações da Estação

Informações Gerais

Relatório do Canal

Número da Estação: 691293600	Número Indicativo: ZYA724
Data Último Licenciamento: 27/04/2017	Número da Licença: 000001/2011-MG

Estação Principal		
Localização		
Latitude: 18°53'6" S	Longitude: 48°15'42" W	Cota da base: 935 m

Transmissor Principal	
Código Equipamento: 004571500352	Modelo: EC704HP (3400W)
Fabricante: Hitachi Kokusai Linear Equipamentos Eletrônicos SA	Potência de Operação: 2.5 kW

Linha de Transmissão Principal			
Modelo: LCF158-50A	Fabricante: RFS BRASIL - KMP		
Comprimento da Linha: 71.0 m	Atenuação: 1.717 dB/100m	Perdas Acessórias: 0.25 dB	Impedância: 50.00 ohms

Antena Principal					
Modelo: ISDE083022UT			Fabricante: Ideal Indústria e Comércio de Antenas Ltda.		
Ganho: 11.2 dBd	Beam-Tilt: 4.0 °	Orientação NV: 210 °	Polarização: Elíptica	HCI: 64 m	ERP Máxima: 23.5 kW

Padrão de Antena dBd											
0°: 16.64	5°: 15.73	10°: 14.92	15°: 14.52	20°: 14.28	25°: 14	30°: 13.88	35°: 14.06	40°: 14.49	45°: 15.26	50°: 16.12	55°: 16.91
60°: 17.5	65°: 17.72	70°: 17.5	75°: 16.63	80°: 15.38	85°: 13.9	90°: 12.42	95°: 11.29	100°: 10.35	105°: 9.56	110°: 8.9	115°: 8.32
120°: 7.86	125°: 7.57	130°: 7.38	135°: 7.22	140°: 7.1	145°: 7.04	150°: 7.01	155°: 6.95	160°: 6.92	165°: 6.95	170°: 7.01	175°: 7.06
180°: 7.1	185°: 7.15	190°: 7.19	195°: 7.25	200°: 7.28	205°: 7.24	210°: 7.19	215°: 7.18	220°: 7.19	225°: 7.2	230°: 7.19	235°: 7.16
240°: 7.1	245°: 7.02	250°: 6.92	255°: 6.81	260°: 6.74	265°: 6.76	270°: 6.83	275°: 6.9	280°: 7.01	285°: 7.19	290°: 7.47	295°: 7.92
300°: 8.47	305°: 9.06	310°: 9.72	315°: 10.48	320°: 11.33	325°: 12.26	330°: 13.3	335°: 14.56	340°: 15.86	345°: 17.24	350°: 18.12	355°: 17.63

Coordenadas por radial											
0°: Lat - Lon -	5°: Lat - Lon -	10°: Lat - Lon -	15°: Lat - Lon -	20°: Lat - Lon -	25°: Lat - Lon -	30°: Lat - Lon -	35°: Lat - Lon -	40°: Lat - Lon -	45°: Lat - Lon -	50°: Lat - Lon -	55°: Lat - Lon -
60°: Lat - Lon -	65°: Lat - Lon -	70°: Lat - Lon -	75°: Lat - Lon -	80°: Lat - Lon -	85°: Lat - Lon -	90°: Lat - Lon -	95°: Lat - Lon -	100°: Lat - Lon -	105°: Lat - Lon -	110°: Lat - Lon -	115°: Lat - Lon -
120°: Lat - Lon -	125°: Lat - Lon -	130°: Lat - Lon -	135°: Lat - Lon -	140°: Lat - Lon -	145°: Lat - Lon -	150°: Lat - Lon -	155°: Lat - Lon -	160°: Lat - Lon -	165°: Lat - Lon -	170°: Lat - Lon -	175°: Lat - Lon -
180°: Lat - Lon -	185°: Lat - Lon -	190°: Lat - Lon -	195°: Lat - Lon -	200°: Lat - Lon -	205°: Lat - Lon -	210°: Lat - Lon -	215°: Lat - Lon -	220°: Lat - Lon -	225°: Lat - Lon -	230°: Lat - Lon -	235°: Lat - Lon -
240°: Lat - Lon -	245°: Lat - Lon -	250°: Lat - Lon -	255°: Lat - Lon -	260°: Lat - Lon -	265°: Lat - Lon -	270°: Lat - Lon -	275°: Lat - Lon -	280°: Lat - Lon -	285°: Lat - Lon -	290°: Lat - Lon -	295°: Lat - Lon -
300°: Lat - Lon -	305°: Lat - Lon -	310°: Lat - Lon -	315°: Lat - Lon -	320°: Lat - Lon -	325°: Lat - Lon -	330°: Lat - Lon -	335°: Lat - Lon -	340°: Lat - Lon -	345°: Lat - Lon -	350°: Lat - Lon -	355°: Lat - Lon -

Distância por radial											
0°:	5°:	10°:	15°:	20°:	25°:	30°:	35°:	40°:	45°:	50°:	55°:
60°:	65°:	70°:	75°:	80°:	85°:	90°:	95°:	100°:	105°:	110°:	115°:
120°:	125°:	130°:	135°:	140°:	145°:	150°:	155°:	160°:	165°:	170°:	175°:
180°:	185°:	190°:	195°:	200°:	205°:	210°:	215°:	220°:	225°:	230°:	235°:
240°:	245°:	250°:	255°:	260°:	265°:	270°:	275°:	280°:	285°:	290°:	295°:
300°:	305°:	310°:	315°:	320°:	325°:	330°:	335°:	340°:	345°:	350°:	355°:

Estação Auxiliar	
Transmissor Auxiliar	
Código Equipamento: 011601001684	Modelo: UAX 1000IS
Fabricante: GatesAir Inc.	Potência de Operação: 1.0 kW

Transmissor Auxiliar 2	

Relatório do Canal

Código Equipamento:				Modelo: Equipamento não encontrado			
Fabricante:				Potência de Operação: kW			

Linha de Transmissão Auxiliar							
Modelo: HF3 1/8"				Fabricante: RFS Brasil - KMP			
Comprimento da Linha: 53.1 m		Atenuação: 0.92 dB/100m		Perdas Acessórias: 0.25 dB		Impedância: 50 ohms	

Antena Auxiliar							
Modelo: TTSL8-U-A-30				Fabricante: Transtel Conti & Cia. Ltda.			
Ganho: 11.99 dBd	Beam-Tilt: 4.9 °	Orientação NV: 210 °	Polarização: Horizontal	HCI: 48 m	ERP Máxima: 23.5 kW		

Informações do documento de Outorga							
Núm Processo	Núm Documento	Tipo Documento	Orgão	Data do docu	Data DOU	Razão do Doc	Natureza
530000547292007	33	Portaria	MC	13/03/2009	16/03/2009	Consignação de TVD	Jurídico

Informações do documento de Aprovação de Locais							
Núm Processo	Núm Documento	Tipo Documento	Orgão	Data do docu	Data DOU	Razão do Doc	Natureza
530000013762009	98	Portaria	MC	13/03/2009	16/03/2009	Aprovação de Local	Técnico

Histórico de Documentos Emitidos							
Núm Processo	Núm Documento	Tipo Documento	Orgão	Data do docu	Data DOU	Razão do Doc	Natureza
74201962	1127	Decreto	PR	04/06/1962	05/06/1962	Autoriza Executar Serviço	Jurídico
962771977	80933	Decreto	PR	05/12/1977	06/12/1977	Renovação	Jurídico
507100000351992	11	Decreto	PR	02/10/1996	04/10/1996	Renovação	Jurídico
507100000351992	83	Decreto Legislativo	CN	23/09/1999	24/09/1999	Deliber. do C. Nacional	Jurídico
530000013762009	1255	Ato	CMPRL	13/03/2009	16/03/2009	Autoriza o Uso de Radiofrequência	Técnico
9999	50285	Ato	ER04	09/11/2015		Autoriza a Alteração de Características Técnicas da Estação	Técnico

Horário de funcionamento							