



**UNIVERSIDADE ESTADUAL DE CAMPINAS**

Faculdade de Engenharia Elétrica e de Computação

SILVANIA TEIXEIRA GONÇALVES

**ARQUITETURAS DE REDES NEURAS ARTIFICIAIS APLICADAS AO PROJETO  
DE ANTENAS EM MICROFITA**

CAMPINAS  
2024

SILVANIA TEIXEIRA GONÇALVES

**ARQUITETURAS DE REDES NEURAIS ARTIFICIAIS APLICADAS AO PROJETO  
DE ANTENAS EM MICROFITA**

Dissertação apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestra em Engenharia Elétrica, na Área de Telecomunicações e Telemática.

Orientador: Prof. Dr. Gilliard Nardel Malheiros Silveira

Este exemplar corresponde à versão final da dissertação defendida pela aluna Sylvania Teixeira Gonçalves, e orientada pelo Prof. Dr. Gilliard Nardel Malheiros Silveira

---

CAMPINAS  
2024

Ficha catalográfica  
Universidade Estadual de Campinas (UNICAMP)  
Biblioteca da Área de Engenharia e Arquitetura  
Elizangela Aparecida dos Santos Souza - CRB 8/8098

G586a Gonçalves, Sylvania Teixeira, 1983-  
Arquiteturas de redes neurais artificiais aplicadas ao projeto de antenas em microfitas / Sylvania Teixeira Gonçalves. – Campinas, SP : [s.n.], 2024.

Orientador: Gilliard Nardel Malheiros Silveira.  
Dissertação (mestrado) – Universidade Estadual de Campinas (UNICAMP), Faculdade de Engenharia Elétrica e de Computação.

1. Antenas de microfita. 2. Aprendizado de máquina. 3. Redes neurais convolucionais. 4. Rede adversária generativa. I. Malheiros Silveira, Gilliard Nardel, 1980-. II. Universidade Estadual de Campinas (UNICAMP). Faculdade de Engenharia Elétrica e de Computação. III. Título.

#### Informações Complementares

**Título em outro idioma:** Architectures of artificial neural networks applied to the design of microstrip antennas

**Palavras-chave em inglês:**

Microstrip antennas

Machine learning

Convolutional neural networks

Generative adversarial networks

**Área de concentração:** Telecomunicações e Telemática

**Titulação:** Mestra em Engenharia Elétrica

**Banca examinadora:**

Gilliard Nardel Malheiros Silveira [Orientador]

Ricardo Toshinori Yoshioka

Romis Ribeiro de Faissol Attux

**Data de defesa:** 28-06-2024

**Programa de Pós-Graduação:** Engenharia Elétrica

**Identificação e informações acadêmicas do(a) aluno(a)**

- ORCID do autor: <https://orcid.org/0000-0002-0027-5448>

- Currículo Lattes do autor: <http://lattes.cnpq.br/2111191789298371>

## COMISSAO JULGADORA – DISSERTAÇÃO DE MESTRADO

**Candidato:** Sylvania Teixeira Gonçalves RA: 141005

**Data da defesa:** 28 de junho de 2024

**Título da Dissertação:** “Arquiteturas de redes neurais artificiais aplicadas ao projeto de antenas em microfita.”

Prof. Dr. Gilliard Nardel Malheiros Silveira (Presidente, FEEC/UNICAMP)

Prof. Dr. Ricardo Toshinori Yoshioka (Membro Externo, Instituto de Pesquisa Eldorado)

Prof. Dr. Romis Ribeiro de Faissol Attux (Membro Interno, FEEC/UNICAMP)

A Ata de Defesa com as respectivas assinaturas dos membros da Comissão Julgadora, encontra-se no SIGA (Sistema de Fluxo de Dissertação/Tese) e na Secretaria de Pós-Graduação da Faculdade de Engenharia Elétrica e de Computação.

## **AGRADECIMENTOS**

A Deus, por sempre me proporcionar saúde e disposição para vencer os desafios nessa fase de vida acadêmica.

Ao meu companheiro Rolando, que se fez presente em todos os momentos, compartilhando seu amor, amizade e seu apoio, na qual foram fundamentais para conclusão deste trabalho.

A minha mãe Ivani, que mesmo não estando presente hoje sempre me apoiou e torceu pelas minhas conquistas.

Ao meu irmão Orlando pelo seu apoio e suporte, que mesmo de longe sempre se fez presente em minha vida.

Ao meu orientador, Prof. Dr. Gilliard Nardel Malheiros Silveira, pelos ensinamentos, disponibilidade e confiança depositada em meu trabalho.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

## RESUMO

Neste trabalho é proposta uma metodologia para otimizar o processo de design de antenas em microfita, com o uso de diferentes arquiteturas de redes neurais artificiais. As antenas em microfita *patch* retangular apresentam vantagens significativas que as tornam uma boa escolha em aplicações de comunicação sem fio. O uso de redes neurais no projeto de antenas em microfita oferece vantagens como a otimização eficiente de múltiplos parâmetros de design. Neste trabalho foi desenvolvido dois estudos de casos com aplicação de redes neurais para modelagem de antenas. O primeiro, consistiu em projetar, treinar e analisar modelos generativos e preditivos aplicados a antena em microfita. Iniciamos com a rede adversária generativa seguida pela rede neural perceptron multicamadas, que foram treinados com um conjunto de dados gerados utilizando fórmulas CAD (*Computer-Aided Design*) para cálculo das características das antenas associados ao substrato RO3003. Após finalizado o modelamento da rede adversária generativa e da rede neural perceptron multicamadas, foi realizada uma análise comparativa entre os resultados das redes com relação aos dados calculados pelas fórmulas CAD, e ambos os modelos apresentaram resultados satisfatórios, onde a rede adversária generativa apresentou uma acurácia de 65%, a rede perceptron multicamadas obteve um excelente resultado com acurácia 95%. Na etapa seguinte desenvolvemos um terceiro modelo usando as redes adversariais generativas condicionais, onde o modelo foi treinado para gerar dados de características de antenas condicionado a uma faixa de frequência específica, com o objetivo de viabilizar o controle sobre o processo de geração desses dados. Demonstramos que a rede adversarial generativa condicional pode gerar dados sintéticos de características de antenas para uma faixa específica de frequência e substratos. O segundo estudo de caso apresenta uma metodologia para prever as características de uma antena de microfita pixelada de banda dupla baseada na rede neural convolucional que toma como entrada uma representação de toda a superfície pixelada de uma antena.

**Palavras-chaves:** Antena em microfita, Aprendizado de máquina, Redes neurais convolucionais, Redes adversárias generativas.

## ABSTRACT

This paper proposes a methodology for optimizing the design process of microstrip antennas, using different artificial neural network architectures. Rectangular patch microstrip antennas have significant advantages that make them a good choice for wireless communication applications. The use of neural networks in the design of microstrip antennas offers advantages such as the efficient optimization of multiple design parameters. In this work, two case studies were developed using neural networks to model antennas. The first consisted of designing, training, and analyzing generative and predictive models applied to microstrip antennas. We started with the generative adversarial network, followed by the multilayer perceptron network, which were trained with a set of data generated using CAD (Computer-aided design) formulas to calculate the characteristics of the antennas associated with the RO 3003 substrate. Once the modeling of the generative adversarial network and the multilayer perceptron network had been completed, a comparative analysis was carried out between the results of the networks in relation to the data calculated by the CAD formulas, and both models showed satisfactory results, where the generative adversarial network showed an accuracy of 65%, and the multilayer perceptron network obtained an excellent result with an accuracy of 95%. In the next step, we developed a third model using conditional generative adversarial networks, where the model was trained to generate antenna characteristics data conditioned to a specific frequency range, with the aim of enabling control over the process of generating this data. We demonstrated that conditional generative adversarial networks could generate synthetic antenna characteristics data for a specific frequency range and substrates. The second case study presents a methodology for predicting the characteristics of a dual-band pixelated microstrip antenna based on a convolutional neural network that takes as input a representation of the entire pixelated surface of an antenna.

**Keywords:** Microstrip antenna, Machine learning, Convolutional neural networks, Generative adversarial networks.

## LISTA DE FIGURAS

Figura 2.1. Antena em microfita com <i>patch</i> retangular .....	21
Figura 2.2. Formas básicas de elementos irradiadores. (a) Quadrada, (b) Retangular, (c) Dipolo, (d) Circular .....	22
Figura 2.3. Métodos de alimentação. (a) Alimentação de linha de Micro, (b) Alimentação por cabo coaxial, (c)Alimentação do tipo acoplamento por abertura e (d) Alimentação por proximidade .....	23
Figura 2.4. Comprimento físico e elétrico para um patch retangular (a) e Efeito do franjamento em um patch retangular (b) .....	26
Figura 2.5. Modelo um neurônio artificial .....	32
Figura 2.6. Exemplo de rede MLP .....	34
Figura 2.7. Arquitetura da CNN .....	37
Figura 2.8. Modelo de cálculo de convolução das duas primeiras posições da imagem (em azul) pelo kernel (em verde) .....	38
Figura 2.9. Exemplo de max-pooling aplicado a matriz com tamanho 2 x 2 e stride 2 .....	39
Figura 3.1. Diagrama do procedimento de treinamento da MLP .....	47
Figura 3.2. Comparação entre os dados de W esperados, os gerados pela GAN e os preditos pela MLP em diferentes faixas de frequências e para permissividades distintas. (a) Frequências entre 1 a 5GHz (b)Frequências entre 5 a 10GHz (c) Frequências entre 10 a 20GHz e (d) Frequências entre 20 a 30GHz .....	48
Figura 3.3. Comparação entre os dados de L esperados, os gerados pela GAN e os preditos pela MLP em diferentes faixas de frequências e para permissividades distintas. (a) Frequências entre 1 a 5GHz (b)Frequências entre 5 a 10GHz (c) Frequências entre 10 a 20GHz e (d) Frequências entre 20 a 30GHz .....	49

Figura 3.4. Comparação entre os dados de D esperados, os gerados pela GAN e os preditos pela MLP em diferentes faixas de frequências e para permissividades distintas. (a) Frequências entre 1 a 5GHz (b)Frequências entre 5 a 10GHz (c) Frequências entre 10 a 20GHz e (d) Frequências entre 20 a 30GHz .....	50
Figura 3.5. Comparação entre os dados de W esperados, os gerados pela GAN e os preditos pela MLP em diferentes faixas de frequências e para permissividades distintas. (a) Frequências entre 1 a 5GHz (b)Frequências entre 5 a 10GHz (c) Frequências entre 10 a 20GHz e (d) Frequências entre 20 a 30GHz .....	51
Figura 3.6. Medida de avaliação de distribuição dos dados. (a) Frequência real e gerada (b) Largura e comprimento reais e gerados (c) Diretividade real e gerada (d) Largura de banda real e gerada .....	53
Figura 3.7. Dados gerados de características de antenas em microfita para permissividades entre 2,0 a 3,0 e em função de fr para (a) W, (b) L, (c) D e (d) Qt .....	55
Figura 3.8. Dados gerados de características de antenas em microfita com 1000 amostras para os parâmetros (a) W, (b) L, (c) D e (d) Qt com permissividade entre 2,0 a 3,0 e faixa de frequências de 2 a 3GHz .....	56
Figura 3.9. Medida de avaliação de distribuição dos dados. (a) Frequência real e gerada (b) Largura e comprimento reais e gerados (c) Diretividade real e gerados (d) Fator de qualidade reais e gerados .....	57
Figura 4.1. (a) Vista superior da geometria base de uma antena em microfita pixelada. Em preto, temos a região metalizada (alguns pixels e a linha de alimentação). O plano de terra é representado pelo contorno tracejado na face oposta. (b) Imagem em perspectiva da antena pixelada desenhada no software Ansys HFSS .....	59
Figura 4.2. Parâmetro S11 para as duas frequências de ressonância .....	60
Figura 4.3. (a) Diretividade em fres1. (b) Diretividade em – fres2 .....	61
Figura 4.4. (a) Ganho em fres1. (b) Ganho de fres2 .....	61
Figura 4.5. Representação esquemática da CNN .....	65

Figura 4.6. Primeira frequência de ressonância. (a) Antenas com agrupamentos de 2 pixels adicionais. (b) Antenas com agrupamentos de 3 pixels adicionais. (c) Antenas com agrupamentos de 4 pixels adicionais .....	67
Figura 4.7. Segunda frequência de ressonância. (a) Antenas com agrupamentos de 2 pixels adicionais. (b) Antenas com agrupamentos de 3 pixels adicionais. (c) Antenas com agrupamentos de 4 pixels adicionais .....	68
Figura 4.8. Geometrias e respectivas curvas S11 de antenas com agrupamentos de 2 pixels adicionais na antena de referência considerando $\epsilon_r = 2,65$ . (a) para o menor erro (b) para o maior erro .....	70
Figura 4.9. Geometrias e respectivas curvas S11 de antenas com agrupamentos de 3 pixels adicionais na antena de referência considerando $\epsilon_r = 2,20$ . (a) para o menor erro (b) para o maior erro .....	71
Figura 4.10: Geometrias e respectivas curvas S11 de antenas com agrupamentos de 4 pixels adicionais na antena de referência considerando $\epsilon_r = 3,0$ . (a) para o menor erro (b) para o maior erro .....	72
Figura 4.11. Primeira frequência de ressonância. (a) Antenas com agrupamentos de 2 pixels adicionais. (b) Antenas com agrupamentos de 3 pixels adicionais. (c) Antenas com agrupamentos de 4 pixels adicionais .....	75
Figura 4.12. Segunda frequência de ressonância. (a) Antenas com agrupamentos de 2 pixels adicionais. (b) Antenas com agrupamentos de 3 pixels adicionais. (c) Antenas com agrupamentos de 4 pixels adicionais .....	76
Figura 4.13. Diretividade na primeira frequência de ressonância. (a) Antenas com agrupamentos de 2 pixels adicionais. (b) Antenas com agrupamentos de 3 pixels adicionais. (c) Antenas com agrupamentos de 4 pixels adicionais .....	78
Figura 4.14. Diretividade na segunda frequência de ressonância. (a) Antenas com agrupamentos de 2 pixels adicionais. (b) Antenas com agrupamentos de 3 pixels adicionais. (c) Antenas com agrupamentos de 4 pixels adicionais .....	79

Figura 4.15. Ganho na primeira frequência de ressonância. (a) Antenas com agrupamentos de 2 pixels adicionais. (b) Antenas com agrupamentos de 3 pixels adicionais. (c) Antenas com agrupamentos de 4 pixels adicionais .....	81
Figura 4.16. Ganho na segunda frequência de ressonância. (a) Antenas com agrupamentos de 2 pixels adicionais. (b) Antenas com agrupamentos de 3 pixels adicionais. (c) Antenas com agrupamentos de 4 pixels adicionais .....	82
Figura 4.17. Curvas S11 de antenas com agrupamentos de 2 pixels adicionais na antena de referência considerando $\epsilon_r = 2,65$ . (a) Para o menor erro. (b) Para o maior erro .....	84
Figura 4.18. Curvas S11 de antenas com agrupamentos de 3 pixels adicionais na antena de referência considerando $\epsilon_r = 2,20$ . (a) Para o menor erro. (b) Para o maior erro .....	85
Figura 4.19. Curvas S11 de antenas com agrupamentos de 4 pixels adicionais na antena de referência considerando $\epsilon_r = 3,00$ . (a) Para o menor erro. (b) Para o maior erro .....	86

## LISTA DE TABELAS

Tabela 3.1. Comparativa entre os erros da GAN e MLP para largura de comprimento .....	52
Tabela 3.2. Comparativa entre os erros da GAN e MLP para diretividade e largura de banda .....	52
Tabela 3.3. Erros da CGAN .....	56
Tabela 4.1. Principais características da antena de referência .....	60
Tabela 4.2. Divisão dos conjuntos de treino, validação e teste .....	64
Tabela 4.3. Erro percentual relativo médio nos dados de treino, validação e teste .....	73
Tabela 4.4. Valores de frequência de ressonância para o primeiro experimento .....	73
Tabela 4.5. Erro percentual relativo médio nos dados de treino, validação e teste .....	87

## LISTA DE ABREVIATURAS

CGAN	<i>Conditional Generative Adversarial Network</i>
CNN	<i>Convolutional Neural Network</i>
FC	<i>Fully Connected</i>
FDTD	<i>Finite-difference time-domain</i>
FEM	<i>Finite element method</i>
GAN	<i>Generative Adversarial Network</i>
GPUs	<i>Graphical Processing Unit</i>
HFSS	<i>High-frequency structure simulator</i>
MAP	<i>Maximum a Posteriori Probability</i>
MLP	<i>Multilayer Perceptron</i>
ReLU	<i>Rectified Linear Unit</i>
Tanh	Tangente Hiperbólica
VAEs	<i>Variational Autoencoders</i>

## LISTA DE SÍMBOLOS

$C_0$	Velocidade das ondas eletromagnéticas no espaço livre
$D$	Diretividade
$\varepsilon_r$	Permissividade relativa do substrato
$f_r$	Frequência de ressonância
$G$	Ganho
$G_{rad}$	Condutância de radiação
$I$	Corrente Elétrica
$L_e$	Comprimento efetivo
$Q_c$	Perdas nos condutores
$Q_d$	Perdas no dielétrico
$Q_r$	Perdas por irradiação
$Q_t$	Fator de qualidade
$R_{in}$	Resistência de entrada da ressonância
$S_{11}$	Coefficiente de reflexão
$U_{(\theta,\varphi)}$	Intensidade máxima de radiação
$Y_0$	Tamanho da inserção na antena
$w_0$	Largura do alimentador de entrada
$\mu_r$	Permeabilidade magnética relativa

# SUMÁRIO

<b>CAPÍTULO 1 .....</b>	<b>18</b>
1.1 Introdução .....	18
1.2 Estado da Arte .....	19
<b>CAPÍTULO 2 .....</b>	<b>21</b>
<b>Fundamentação Teórica .....</b>	<b>21</b>
2.1 Antenas em Microfita .....	21
2.1.1 Métodos de Alimentação .....	22
2.1.2 Antena em Microfita com Formato Retangular .....	23
2.1.3 Métodos de Análise .....	24
2.1.3.1 Modelo de Linha de Transmissão .....	24
2.1.3.2 Modelo de Cavidade .....	24
2.1.3.3 Modelos de Onda Completa .....	24
2.1.4 Projeto de Antena <i>Patch</i> Retangular .....	25
2.1.4.1 Diretividade .....	27
2.1.4.2 Largura de Banda .....	27
2.1.4.3 Fator de Qualidade .....	28
2.2 Aprendizado de máquina .....	29
2.2.1 Conceitos Básicos .....	29
2.2.1.1 Aprendizado Supervisionado .....	29
2.2.1.2 Aprendizado Não Supervisionado.....	30
2.2.1.3 Aprendizado Semi-Supervisionado.....	31
2.2.1.4 Aprendizado por Reforço .....	31
2.2.2 RNAs .....	31
2.2.3 Função de Ativação.....	33
2.2.4 A Arquitetura MLP .....	34
2.2.5 Aprendizado Profundo .....	35
2.2.6 Redes Neurais Convolucionais.....	36
2.2.6.1 Camada Convolucional .....	37

2.2.6.2 Camada Não-Linear .....	39
2.2.6.3 Camada de Agrupamento ( <i>Pooling</i> ) .....	39
2.2.6.4 Camada Totalmente Conectada .....	39
2.2.6.5 Camada de Saída .....	40
2.2.7 Redes Adversárias Generativas .....	40
2.2.7.1 Arquitetura da GAN .....	41
2.2.7.2 Espaço Latente da GAN .....	42
2.2.8 Redes Adversárias Generativas Condicionais .....	42
2.2.8.1 Rede Geradora da CGAN .....	42
2.2.8.2 Rede Discriminadora da CGAN .....	43
<b>CAPÍTULO 3 .....</b>	<b>44</b>
<b>Primeiro Estudo de Caso .....</b>	<b>44</b>
3.1 Estratégia com RNAs .....	44
3.2.1 Geração da Base de Dados .....	44
3.2.2 Arquitetura GAN .....	45
3.2.3 Arquitetura MLP .....	46
3.3 Resultados .....	47
3.3.1 Dados gerados da largura ( <i>W</i> ) .....	47
3.3.2 Dados gerados do comprimento ( <i>L</i> ) .....	48
3.3.3 Dados gerados de diretividade ( <i>D</i> ) .....	49
3.3.4 Dados gerados de largura de banda ( <i>BW</i> ) .....	52
3.4 Comparação entre os dados reais e gerados usando <i>box-plots</i> .....	52
3.5 Treinamento CGAN .....	53
3.5.1 Resultados da CGAN .....	54
3.5.1.1 Dados gerados condicionados a faixa de frequência de 1 a 5GHz .....	54
3.5.1.2 Comparação entre os dados reais e gerados da CGAN usando <i>box-plots</i> .....	57
3.6 Conclusão do Capítulo .....	58
<b>CAPÍTULO 4 .....</b>	<b>59</b>
<b>Segundo Estudo de Caso .....</b>	<b>59</b>

4.1 Modelagem de Antena em Microfita Pixelada usando CNN .....	59
4.2 Definição do Problema .....	61
4.3 Descrição da Estratégia via RNAs .....	62
4.4 Geração da Base de Dados .....	62
4.5 Treinamento da CNN .....	64
4.6 Resultados da CNN .....	65
4.7 Conclusão do Capítulo .....	87
<b>CAPÍTULO 5 .....</b>	<b>88</b>
<b>Conclusão .....</b>	<b>88</b>
<b>Referências Bibliográficas .....</b>	<b>89</b>

# CAPÍTULO 1

## 1.1 Introdução

As antenas em microfita são amplamente utilizadas em diversas aplicações que envolvem comunicação devido às suas características de baixo perfil, leveza e facilidade de fabricação. Estas características são essenciais para atender à crescente demanda por dispositivos que operem em múltiplas bandas de frequência; uma exigência impulsionada pela necessidade de maior largura de banda e flexibilidade nos sistemas de comunicações modernos. Contudo, o processo de otimização do design dessas antenas, no atendimento aos requisitos específicos de desempenho é um desafio não trivial devido à natureza complexa e às múltiplas restrições de projeto.

As metodologias comumente utilizadas na otimização de antenas geralmente envolvem processos demorados e computacionalmente intensivos. Nesse tipo de processo, um algoritmo de otimização precisa acionar, repetidamente, a execução de simulações. Uma alternativa para agilizar esse procedimento seria aplicar técnicas de aprendizado de máquina baseadas em redes neurais artificiais (RNAs). A literatura reporta algumas contribuições recentes envolvendo o emprego de RNAs no projeto de antenas [1-3].

Neste trabalho, serão apresentados dois estudos de caso que utilizam técnicas de aprendizado de máquina baseadas em RNAs para otimização do processo de síntese de antenas em microfita. As implementações envolveram o projeto, treinamento e análise de diferentes arquiteturas de RNAs.

No primeiro estudo de caso foram desenvolvidos três modelos de RNAs: uma rede adversária generativa (GAN, *Generative Adversarial Networks*) [4], uma rede perceptron multicamadas (MLP, *Multilayer Perceptron*) e redes adversariais generativas condicionais (CGAN, *Conditional Generative Adversarial Network*). Estes modelos foram treinados com um conjunto de dados gerados através de fórmulas CAD (*Computer-Aided Design*) para calcular as características das antenas, utilizando substratos distintos. Ainda neste primeiro estudo de caso, foram realizados três experimentos apresentados a seguir.

O primeiro experimento envolveu o uso de GANs para geração de novas amostras de antenas *patches* em microfita através do aprendizado das relações entre as características das antenas e suas geometrias. A GAN capturou efetivamente o conhecimento derivado de diversas iterações de design e rodadas de otimização. Escolhemos a GAN por ser uma arquitetura que

gera amostras sintéticas da mesma distribuição do conjunto de dados com a qual foi treinada. Uma vez treinadas, essas RNAs podem ser consultadas para a síntese rápida de designs de antenas que atendam às especificações desejadas. No segundo experimento, foi desenvolvido um modelo MLP para predição das características das antenas a partir dos dados de entrada. Após o treinamento, foi realizada uma análise comparativa entre os modelos GAN e MLP, utilizando os dados obtidos por meio de fórmulas CAD. O terceiro experimento explorou o uso de CGANs. Nesse modelo o gerador condicional aceita um vetor de ruído e um rótulo de classe em sua entrada e os mapeia para um vetor de parâmetros estruturais da classe indicada no rótulo de classe. As classes foram definidas por faixas de frequências, onde o modelo foi treinado para gerar dados de características das antenas condicionados a classe referente a faixa de frequência específica, permitindo um controle mais preciso sobre o processo de geração de dados.

No segundo estudo de caso, foi empregada uma CNN para modelagem das características de uma antena em microfita pixelada e de banda dupla. Esta metodologia permitiu a predição com boa precisão de suas principais características, como frequência de ressonância, ganho e diretividade.

Os resultados destes estudos demonstraram o potencial das RNAs para otimização e inovação do design de antenas em microfita. A aplicação dessas técnicas de aprendizado de máquina torna-se uma abordagem promissora para superação dos desafios durante a otimização de antenas em microfita, abrindo novas possibilidades de desenvolvimento de sistemas de comunicação de forma rápida e efetiva.

## 1.2 Estado da Arte

No trabalho publicado por [1] foi proposto o uso de GANs para síntese rápida de designs de LPFDA (*Log-Periodic Folded Dipole Array*) para duas faixas de valores do fator de qualidade,  $Q$ . Neste trabalho os autores desenvolveram modelos, usando *Vanilla GAN*, *Conditional GAN* e *Wasserstein GAN*. Para treinamento deste modelo foram criados três conjuntos de dados com base nos intervalos de interesse para os valores do fator  $Q$ . Demonstrando-se que a GAN pode ser utilizada como um banco de dados de antenas aprendido e orientado por dados e consultado para geração rápida de novos projetos semelhantes de antenas aos quais foram treinados. No entanto, os autores não apresentaram uma variação para diferentes materiais de permissividades, e não expandiram essas técnicas para outras características das antenas.

Em [2], foi proposta uma metodologia para modelar com precisão duas frequências de ressonância de uma antena em microfita pixelada de banda dupla com base na regressão de uma CNN, que toma toda a superfície pixelada da antena como entrada. A CNN apresentou excelentes resultados preditivos, com erro relativo médio dos testes de 0,13% e 0,22% para as duas frequências de ressonância. A abordagem dos autores explorou a predição das frequências de ressonância com boa precisão, entretanto, este modelo de predição não foi expandido para outras características das antenas, como a diretividade, ganho ou largura de banda; bem como, para diferentes substratos.

Em [3], uma antena em microfita retangular (RMSA, *Rectangular Microstrip Antenna*) é desenvolvida utilizando técnicas de modelagem com RNAs. O algoritmo de retropropagação é empregado para treinar a rede, visando a minimização de erros e a redução do tempo de computação. Como resultado, as dimensões geométricas do *patch* são determinadas com boa precisão e em um tempo de computação menor quando comparado a softwares de simulação tradicionais.

Em [5], foi proposto um método de *design* de padrão de antena em microfita usando o modelo GAN. Nesse caso, o projetista alimenta a RNA com apenas o padrão de radiação e o coeficiente de reflexão projetados e a RNA produz a padrão da antena desejado. Para criação dos geradores e os discriminadores foram usadas células *Long Short Term Memory* (LSTM). O software comercial Ansys HFSS foi utilizado para validação do modelo.

Em [6] é apresentada uma forma de reproduzir antenas com um  $Q$  dentro de um limite pré-determinado através do modelo GAN. O modelo proposto é treinando para gerar parâmetros de antena a partir da distribuição aprendida com um  $Q$  previsto. Os dados obtidos ofereceram nova perspectiva para compreender a correlação entre a distribuição dos parâmetros da antena e seu  $Q$ . No entanto, os autores não apresentaram uma variação para diferentes faixas de frequências, ou expandiram essas técnicas para outras características das antenas.

## CAPÍTULO 2

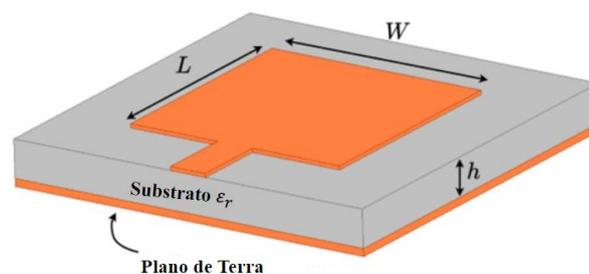
### Fundamentação Teórica

#### 2.1 Antenas em Microfita

Uma antena é comumente definida como um dispositivo capaz de radiar ou receber ondas de rádio, ou seja, é uma estrutura de transição entre o espaço livre e um meio guiado [7].

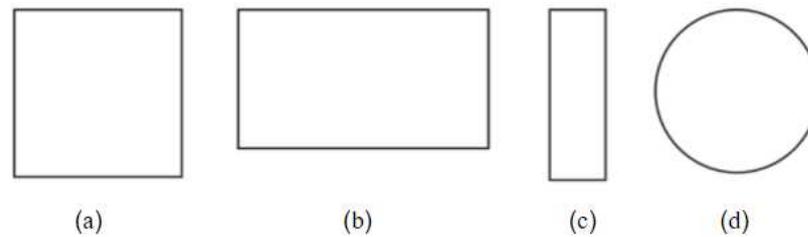
O desenvolvimento dos sistemas de micro-ondas, juntamente com os circuitos integrados e a demanda do mercado por antenas de baixo custo de fabricação, deu início ao uso das antenas em microfita. Devido às suas pequenas dimensões, essas estruturas puderam ser aplicadas em diversas tecnologias, entre essas aplicações, destacam-se o uso em aeronaves de alto desempenho, espaçonaves, satélites e mísseis, onde tamanho, peso, fácil instalação e perfil aerodinâmico são essenciais para o sucesso da implementação.

Uma antena em microfita é formada por duas placas condutoras paralelas separadas por uma fina camada de dielétrico, denominado substrato, sendo uma das placas o elemento irradiante (*patch*) e a outra o plano terra conforme representado na figura 2.1. Sua popularidade se deve ao fato de poder ser construída diretamente sobre a placa de circuito impresso de algum dispositivo, utilizando as mesmas técnicas de fabricação de circuitos impressos. O tamanho e a forma do *patch* são aspectos importantes no desempenho dessa antena.



**Figura 2.1.** Antena em microfita com *patch* retangular.

O design do *patch* pode adotar diversas formas geométricas e disposições. No entanto, os variados formatos do *patch* produzem resultados distintos, além de requererem diferentes modelos matemáticos para a determinação das dimensões da antena e suas características eletromagnéticas de funcionamento. A Figura 2.2 ilustra os formatos mais simples para a construção de uma antena em microfita.



**Figura 2.2.** Formas básicas de elementos irradiadores. (a) Quadrada, (b) Retangular, (c) Dipolo, (d) Circular.

### 2.1.1 Métodos de Alimentação

Dentre as características básicas da antena podemos destacar as diferentes configurações que podem ser usadas para alimentar antenas em microfita. Os métodos mais comuns de alimentação são: linha de microfita, cabo coaxial, acoplamento por abertura e acoplamento por proximidade.

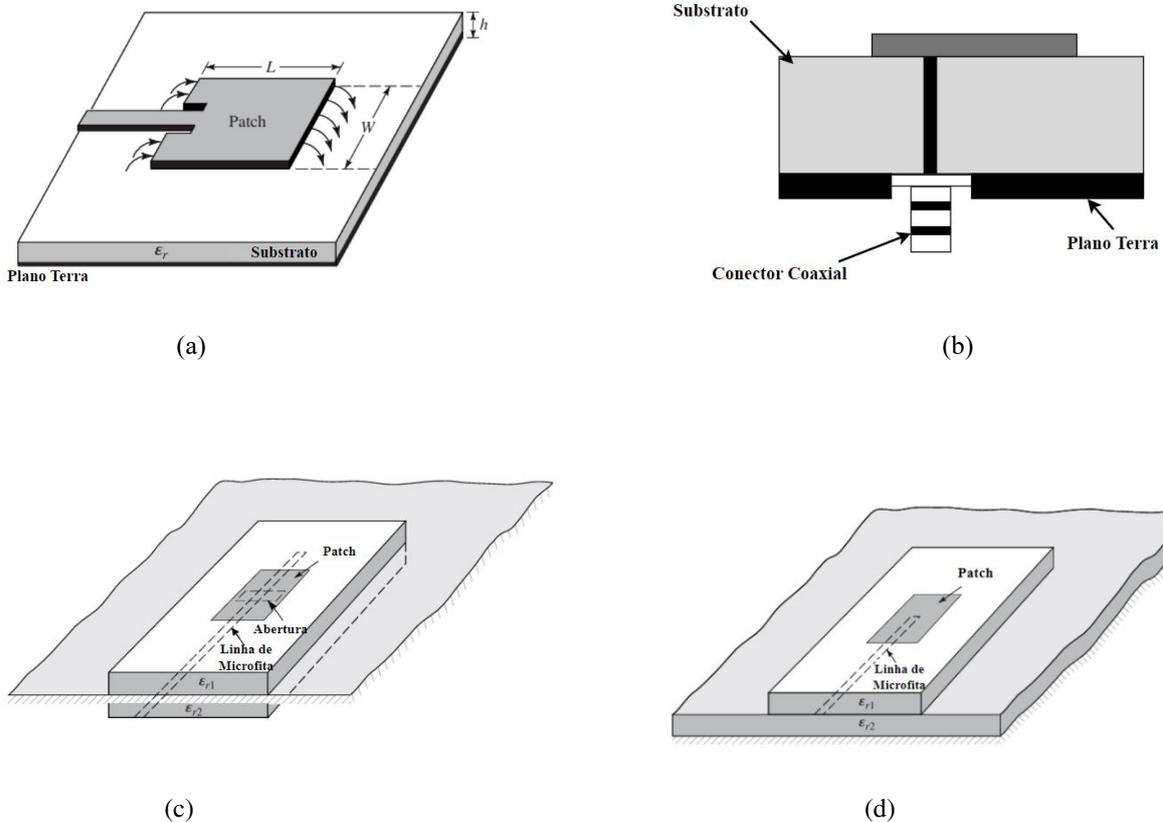
O método de linha de microfita é constituído por uma fita condutora, geralmente muito menor que a largura do *patch* (veja na figura 2.3 (a)). A linha condutora é de fácil fabricação, no entanto, à medida que a espessura do substrato aumenta, há um incremento nas ondas de superfícies e radiação espúria, limitando a sua aplicação a larguras de banda de 2% a 5% [7].

A alimentação por cabo coaxial é um tipo de alimentação onde o condutor interno do cabo coaxial é conectado ao *patch* da antena, enquanto o condutor externo é conectado ao plano de terra (veja na figura 2.3 (b)). Este tipo de alimentação é de fácil fabricação e apresenta baixa radiação espúria. No entanto, também possui largura de banda estreita.

A alimentação do tipo acoplamento por abertura reduz a radiação espúria em comparação com os métodos mencionados anteriormente (veja na figura 2.3 (c)). No entanto, ela também apresenta uma largura de banda estreita e apresenta maior complexidade na fabricação. Esse tipo de alimentação consiste em dois substratos separados por um plano de terra e uma linha de microfita abaixo do substrato inferior, responsável pela transferência de energia para o *patch* [8].

A alimentação por proximidade envolve uma linha de alimentação posicionada entre dois substratos (veja na figura 2.3 (d)). O *patch* é colocado sobre o substrato superior, enquanto o plano de terra fica sob o substrato inferior. O casamento de impedância é obtido ajustando a largura da linha de transmissão e a espessura dos substratos. Em comparação com as

alimentações por cabo coaxial e linha de microfita, esse método é o que proporciona à antena a maior largura de banda e a menor radiação espúria.



**Figura 2.3.** Métodos de alimentação. (a) Alimentação de linha por linha em microfita, (b) Alimentação por cabo coaxial, (c) Alimentação do tipo acoplamento por abertura e (d) Alimentação por proximidade.

### 2.1.2 Antena em Microfita com Formato Retangular

Uma das configurações mais simples e comumente utilizadas é a RMSA, na qual o elemento irradiador tem um formato retangular e é alimentado por meio de uma linha em microfita (veja figura 2.1).

A análise e o dimensionamento da antena podem ser inicialmente realizados por meio de dois métodos aproximados: método da linha de transmissão e o método da cavidade. No método da cavidade, são determinadas as frequências de ressonância, a distribuição de modos, o padrão de irradiação, a diretividade, o fator de qualidade e a largura de banda da antena. Já o método da linha de transmissão permite o cálculo do comprimento  $L$  e da largura  $W$ , além do estudo da impedância de entrada da antena.

### 2.1.3 Métodos de Análise

Existem diversos métodos para que podem ser utilizadas para modelagem e análise de antenas em microfita, sendo os mais frequentemente empregados os modelos de linha de transmissão, de cavidade e de onda completa. O modelo de linha de transmissão é o mais simples de todos, oferecendo uma boa aproximação física, apesar de ser menos preciso e apresentar dificuldades na modelagem do acoplamento. O modelo de cavidade, em comparação ao de linha de transmissão, é mais exato, mas apresenta maior complexidade. Os modelos de onda completa são precisos, versáteis e capazes de lidar geometrias arbitrárias.

#### 2.1.3.1 Modelo de Linha de Transmissão

O método da linha de transmissão visa modelar a antena de microfita retangular como um conjunto de duas fendas, separadas por uma linha de transmissão de microfita de baixa impedância e comprimento  $L$ . Essas duas fendas são idênticas, com largura  $W$  e altura  $h$ .

O franjamento é identificado como um aumento das dimensões do *patch* em relação às suas medidas físicas. Esse efeito das ondas eletromagnéticas no *patch* está diretamente relacionado à razão entre o comprimento ( $L$ ) e a espessura do substrato ( $h$ ), bem como à permissividade do dielétrico ( $\epsilon_r$ ). Uma permissividade relativa efetiva  $\epsilon_{\text{reff}}$  é introduzida para levar em conta a franja e a propagação da onda na linha. A permissividade relativa efetiva é considerada como uma constante dielétrica uniforme e que apresenta características de uma antena ideal sem os efeitos de borda.

#### 2.1.3.2 Modelo de Cavidade

O método da cavidade considera a RMSA como uma cavidade ressonante retangular. Assim, a partir das equações de Maxwell, são determinadas as distribuições de campo dentro do substrato dielétrico. Essa região do substrato é tratada como uma cavidade limitada por condutores elétricos perfeitos (representados pelo *patch* e terra) e por condutores magnéticos perfeitos (para representação de um circuito aberto) nas laterais da cavidade. Essa abordagem é mais complexa, embora mais precisa quando comparada ao modelo de linha de transmissão. Porém, se restringe a antenas com *patches* cuja geometria seja regular.

#### 2.1.3.3 Modelos de Onda Completa

Os modelos de onda completa são obtidos através da resolução das equações de Maxwell associadas às condições de fronteira definidas pelo problema. Esses modelos são mais

precisos e mais adequados para estudar antenas com estruturas mais complexas. Sua aplicação requer utilização de métodos numéricos, e diferente de outros métodos que envolvem aproximações para solução analítica esses exigem elevada capacidade computacional. Alguns dos modelos de onda completa mais utilizados para modelagem de antenas são: método dos momentos, método dos elementos finitos (FEM, *Finite Element Method*) e método das diferenças finitas.

#### 2.1.4 Projeto de Antena *Patch* Retangular

O projeto de uma RMSA envolve o dimensionamento da largura,  $W$ , e do comprimento,  $L$ , bem como a determinação do local onde a alimentação deve ser inserida na antena. Para isso, utilizam-se como parâmetros de entrada a frequência de operação,  $f_r$ , e a permissividade dielétrica do substrato,  $\epsilon_r$ , e sua espessura. O comprimento e a largura do *patch* podem ser calculados pelas Equações 2.1 e 2.2, onde  $C_0$  representa a velocidade da luz no vácuo e  $f_r$  a frequência de ressonância desejada para a antena.

$$W = \frac{C_0}{2f_r} \sqrt{\frac{2}{\epsilon_r + 1}} \quad (2.1)$$

O comprimento  $L$  pode ser determinado a partir da frequência de ressonância desejada para o modo  $TM_{010}$ , da seguinte forma:

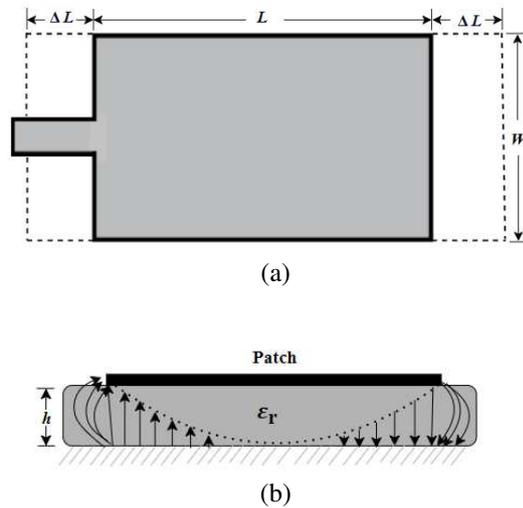
$$L = \frac{C_0}{2f_r \sqrt{\epsilon_{reff}}} - 2\Delta L \quad (2.2)$$

Devido ao efeito de franjamento dos campos de borda mencionado anteriormente, a permissividade dielétrica e a permissividade do meio externo afetam os campos, sendo necessário determinar uma constante dielétrica efetiva ( $\epsilon_{reff}$ ) através da equação 2.3.

$$\epsilon_{reff} = \frac{\epsilon_r + 1}{2} + \frac{\epsilon_r - 1}{2} \left(1 + \frac{12h}{W}\right)^{-\frac{1}{2}} \quad (2.3)$$

Esta equação é utilizada em situações em que a razão entre a largura do *patch* e a altura do substrato é superior a 1 ( $W/h > 1$ ).

O efeito de borda está relacionado a permissividade efetiva, e ao comprimento efetivo do *patch*, ou seja, eletricamente, o comprimento apresenta-se maior com relação ao comprimento físico (veja figura 2.4).



**Figura 2.4.** Comprimento físico e elétrico para um *patch* retangular (a) e Efeito do franjamento em um *patch* retangular (b).

A variação de comprimento elétrico da antena em microfita *patch* retangular pode ser calculada pela Equação 2.4.

$$\frac{\Delta L}{h} = 0,412 \frac{(\epsilon_{reff} + 0,3) \left(\frac{W}{h} + 0,264\right)}{(\epsilon_{reff} - 0,258) \left(\frac{W}{h} + 0,813\right)} \quad (2.4)$$

Sendo o comprimento efetivo, a soma do comprimento físico mais duas vezes a variação de comprimento elétrico do *patch* pelos efeitos de borda.

A condutância pode ser obtida usando a expressão de campo derivada do modelo de cavidade. Em geral, a condutância é definida como:

$$G_1 = \frac{I_1}{120\pi^2} \quad (2.5)$$

em que

$$I_1 = -2 + \cos(X) + XS_i(X) + \frac{\sin(X)}{X} \quad (2.6)$$

sendo

$$X = k_0 W \quad (2.7)$$

onde  $k_0$  é o número de onda no espaço livre e  $S_i(X)$  é a função seno integral.

A resistência de entrada para a frequência de ressonância pode ser calculada por:

$$R_{in} = \frac{1}{2(G_1 \pm G_{12})} \quad (2.8)$$

o sinal de mais (+) é utilizado para modos que apresentam uma distribuição de tensão ressonante ímpar (antissimétrica) sob o *patch* e entre os slots, enquanto o sinal de menos (-) é empregado para modos com distribuição de tensão ressonante par (simétrica). A condutância  $G_{12}$  é calculada mediante Equação 2.9 onde  $J_0(\cdot)$  é a função de Bessel de primeira espécie com base zero.

$$G_{12} = \frac{1}{120\pi^2} \int_0^\pi \left( \frac{\sin\left(\frac{k_0 W}{2} \cos(\theta)\right)}{\cos(\theta)} \right)^2 J_0(k_0 L \sin(\theta)) \sin^3(\theta) d\theta \quad (2.9)$$

Desta forma, são definidas todas as características geométricas da antena em microfita. Adicionalmente, é possível definir alguns parâmetros fundamentais da RMSA, tais como, a diretividade e a largura de banda.

### 2.1.4.1 Diretividade

A diretividade é definida como a razão entre a intensidade de máxima irradiação na direção do feixe principal e a intensidade de irradiação média em todas as direções. Assim, a diretividade da antena em microfita operando no modo  $TM_{010}$  pode ser calculada da seguinte maneira:

$$D = \left( \frac{2\pi W}{\lambda_0} \right)^2 \frac{\pi}{I_2} \quad (2.13)$$

$$I_2 = \int_0^\pi \int_0^\pi \left[ \left( \frac{\sin\left(\frac{k_0 W}{2} \cos(\theta)\right)}{\cos(\theta)} \right) \right]^2 \sin^3 \theta \cos^2 \left( \frac{k_0 L_e}{2} \sin \theta \sin \phi \right) d\theta d\phi \quad (2.14)$$

onde  $\lambda_0$  e  $k_0$  são, respectivamente, o comprimento e número de onda no espaço livre.

### 2.1.4.2 Largura de Banda

A largura de banda (*BW, bandwidth*) pode ser definida como a faixa de frequências na qual o desempenho da antena, de acordo com suas características, atende um padrão especificado. Essa característica pode ser atribuída tanto com relação aos padrões de radiação quanto para a relação entre onda estacionária de tensão e a potência refletida. A largura de banda pode ser definida em termos percentuais, pela Equação 2.15.

$$BW(\%) = 2 \frac{f_{sup} - f_{inf}}{f_{sup} + f_{inf}} \times 100 \quad (2.15)$$

Onde  $f_{sup}$  e  $f_{inf}$  são as componentes de frequência superior e inferior do sinal, respectivamente.

### 2.1.4.3 Fator de Qualidade

O fator de qualidade é inversamente proporcional às perdas de potência na antena. Entretanto, essas perdas não necessariamente referem-se à potência dissipada, mas sim à energia que não é armazenada na estrutura. O fator de qualidade total,  $Q_t$ , em antenas em microfita é constituído pelos seguintes fatores de qualidade: as perdas nos condutores,  $Q_c$ , as perdas no dielétrico,  $Q_d$ , e as perdas por irradiação  $Q_r$ . As primeiras componentes estão relacionadas à dissipação de energia, enquanto  $Q_r$  se refere à porção da energia irradiada. Assim,  $Q_t$  é afetado por todas essas perdas e, usualmente, é determinado pela equação 2.16.

$$\frac{1}{Q_{dt}} = \frac{1}{Q_d} + \frac{1}{Q_c} + \frac{1}{Q_r} \quad (2.16)$$

$$Q_d = \frac{1}{\tan(\delta)} \quad (2.17)$$

$$Q_c = h \sqrt{\pi f_r \mu \sigma} \quad (2.18)$$

$$G_{rad} = \frac{I_2}{120\pi^2} \quad (2.19)$$

$$Q_r = \frac{\pi f_r \epsilon_r \epsilon_0 LW}{h G_{rad}} \quad (2.20)$$

Sendo  $\mu$  a permeabilidade magnética do espaço livre e  $\epsilon_0$  a permissividade do espaço livre.

## 2.2 Aprendizado de máquina

Este capítulo explora conceitos básicos relacionados a área de aprendizado de máquina. Neste trabalho será abordado o uso de diferentes arquiteturas de RNAs para projetos de antenas em microfita. Assim, é fornecida uma visão geral sobre aprendizado de máquina e algumas RNAs com o objetivo de trazer ao leitor os conhecimentos básicos que serão utilizados ao longo deste trabalho.

### 2.2.1 Conceitos Básicos

Aprendizado de máquina é um campo da inteligência artificial que se concentra no desenvolvimento de algoritmos e técnicas que permitem aos computadores aprenderem a partir de dados e fazerem previsões ou tomarem decisões com base nesse aprendizado [9].

O aprendizado de máquina compreende métodos computacionais que dependem de informações anteriores disponíveis para melhorar o desempenho ou obter previsões mais precisas. No campo do Aprendizado de Máquina, há quatro abordagens de aprendizagem frequentemente empregadas, cada uma sendo valiosa para resolver diferentes categorias de problemas: aprendizagem supervisionada, não supervisionada, semi-supervisionada e por reforço.

#### 2.2.1.1 Aprendizado Supervisionado

O aprendizado supervisionado é um procedimento no qual um algoritmo recebe uma coleção de dados rotulados como entrada e gera um modelo como saída. Os dados de treinamento consistem em pares  $(x(n), y(n))$ , onde  $x(n)$  é o vetor de características de entrada e  $y(n)$  é o vetor com as saídas de referência, com  $n$  variando de 1 a  $N$  amostras de treinamento [10]. No aprendizado supervisionado tem-se o conhecimento das respostas desejadas para um conjunto de amostras, e esse conhecimento é utilizado para ajustar o modelo. Em outras palavras, durante o treinamento, é possível calcular uma medida de erro entre a resposta gerada pelo modelo e a desejada, e, em seguida, usar essa medida como critério para ajustar os parâmetros.

O problema para o aprendizado supervisionado pode ser caracterizado de acordo com a natureza de sua saída, podendo ser classificado em duas categorias, como classificação e regressão. Se o modelo realiza um mapeamento para resultados contínuos (reais) e não categóricos, caracteriza-se como um problema de regressão. Quando a saída é categórica,

expressa por diferentes classes e normalmente de maneira discreta, caracteriza-se como um problema de classificação.

A elaboração de um modelo para regressão ou classificação inicia-se com a seleção da estrutura do modelo, que determina o tipo de mapeamento realizado,  $y = f(x, w)$ , onde  $w$  representa o conjunto de parâmetros dessa estrutura e  $x$  os dados de entrada. Em seguida, na etapa de treinamento, os parâmetros em  $w$  são ajustados com base em um critério matemático relacionado ao problema, como, por exemplo, o cálculo do erro quadrático médio na saída. Para tal, é importante o uso de um algoritmo de otimização, por exemplo, métodos que utilizam o gradiente da função de custo.

Após treinado, esse modelo pode ser empregado para classificar ou prever o valor de saída de novas entradas, as quais não foram apresentadas durante o treinamento e que constituem o conjunto de teste. A capacidade do sistema de realizar previsões corretas em amostras diferentes daquelas usadas no treinamento é chamada de generalização. Na prática, o conjunto de treinamento geralmente representa uma pequena parte de todos os possíveis vetores de entrada, e, assim, a generalização é uma característica fundamental em sistemas de aprendizado de máquina e reconhecimento de padrões.

Um importante conceito em aprendizado supervisionado é a validação cruzada. O processo inicia-se dividindo os dados de forma aleatória em dois conjuntos, dados de treinamento e dados de teste. O conjunto de dados de treino é então segmentado em dois subconjuntos: um conjunto de estimação, utilizado para treinar e ajustar os parâmetros do modelo, e um conjunto de validação, que valida o modelo durante o treinamento. Esses conjuntos devem ser representativos e independentes. Assim, enquanto o modelo é ajustado com base no erro dos dados de treinamento, também se monitora o erro nos dados de validação. O conjunto de dados de validação permite prever o desempenho do modelo em dados novos. A validação cruzada ajuda a reduzir o risco de sobreajuste (*overfitting*), uma situação indesejável em que o modelo se adapta bem aos dados de treinamento, mas não apresenta uma boa capacidade de generalização [11].

### **2.2.1.2 Aprendizado Não Supervisionado**

No aprendizado não supervisionado, o conjunto de dados de treinamento contém as amostras de entrada, mas não as saídas correspondentes. Para este tipo de aprendizado, o modelo deve aprender a relação que existe entre os dados de entrada, identificar os padrões, ou categorias nos dados apresentados, produzindo uma codificação na saída. Os objetivos do

aprendizado não supervisionado incluem o processo de clusterização (que é a identificação de grupos de amostras semelhantes dentre os dados), a estimativa de densidade (que é a determinação da distribuição dos dados dentro do espaço de entrada) e a redução de dimensionalidade (onde os dados em um espaço de alta dimensão são transformados para uma dimensão inferior, tornando mais fácil a visualização e a análise da estrutura dos dados).

### **2.2.1.3 Aprendizado Semi-Supervisionado**

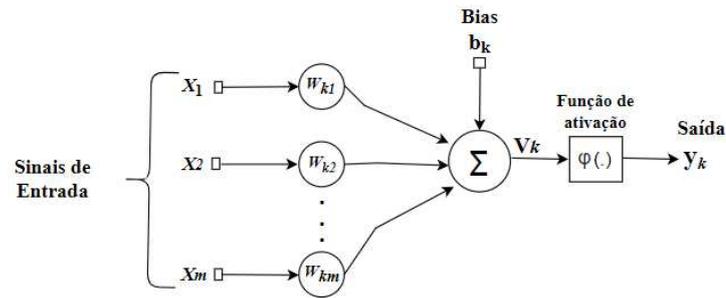
O aprendizado semi-supervisionado pode ser considerado o “meio-termo” entre o aprendizado supervisionado e não supervisionado e é particularmente útil para conjuntos de dados que contêm dados rotulados e não-rotulados (ou seja, todos os recursos estão presentes, mas nem todos os recursos têm alvos associados). Em geral, a ideia central do aprendizado semi-supervisionado é tratar um ponto de dados de forma diferente com base no fato de ele ter ou não um rótulo: para pontos rotulados, o algoritmo usará a supervisão tradicional para atualizar os pesos do modelo, e para pontos não-rotulados, o algoritmo minimiza a diferença nas previsões entre outros exemplos de treinamento semelhantes.

### **2.2.1.4 Aprendizado por Reforço**

O aprendizado por reforço é definido pela interação constante do modelo com o ambiente para selecionar ações apropriadas ao longo do tempo, com o objetivo de maximizar sua recompensa acumulada. Nesse contexto, não há uma solução perfeita para o treinamento do sistema, como ocorre no aprendizado supervisionado, em vez disso, são executadas várias ações, através de tentativa e erro, que geram recompensas diferentes. Este tipo de aprendizado é inspirado na psicologia comportamental, onde um agente aprende com base em recompensas e punições recebidas ao executar ações. Neste contexto, o objetivo do agente é sempre tomar ações que maximizem a recompensa esperada ao longo do tempo [12].

## **2.2.2 RNAs**

As RNAs são representações matemáticas inspiradas na maneira como operam o cérebro e o sistema nervoso como um todo. A unidade fundamental de uma RNA é o neurônio artificial [13]. As RNAs têm a capacidade de armazenar conhecimento com base na experiência anterior com dados, utilizando essa informação para processar novas entradas. O aprendizado ocorre por meio da aquisição de conhecimento do ambiente, ajustando os pesos das conexões entre os neurônios artificiais, conhecidos como pesos sinápticos [14].



**Figura 2.5.** Modelo um neurônio artificial.

O modelo do neurônio artificial, representado na figura 2.5, é constituído por alguns elementos básicos:

Os sinais de entrada  $\{x_1, x_2, \dots, x_m\}$ , ou informações externas são os valores assumidos pelas variáveis de uma aplicação particular. Esses sinais são frequentemente normalizados com o objetivo de aumentar a eficiência computacional dos algoritmos de aprendizado. Em seguida, temos os pesos sinápticos,  $\{w_1, w_2, \dots, w_{km}\}$ , onde os índices  $k$  e  $j$  são usados para identificar o neurônio e a sinapse que pondera a  $j$ -ésima entrada do  $k$ -ésimo neurônio. Esses valores ponderam as variáveis de entrada da rede, possibilitando quantificar suas relevâncias em relação à funcionalidade do respectivo neurônio. Depois, temos um somatório de todos os sinais de entrada, ponderados pelos respectivos pesos sinápticos e uma função de ativação que gera a saída do neurônio, e esta função comumente aplica uma transformação não-linear em sua entrada. Neste modelo também temos um *bias*,  $b_k$ , o qual tem por objetivo aumentar ou diminuir a entrada da função de ativação. O último elemento do modelo do neurônio é a função de ativação, que age sobre o potencial de ativação para produzir o sinal de saída da rede. Essa função, de maneira tradicional, confere uma característica não-linear aos neurônios e, em algumas situações, restringe a saída a determinados intervalos.

O neurônio artificial pode ser descrito pelas seguintes expressões matemáticas:

$$u_k = \sum_{j=1}^m w_{kj} x_j \quad (2.21)$$

$$y_k = \phi(u_k + b_k) \quad (2.22)$$

### 2.2.3 Função de Ativação

Como descrito anteriormente, as funções de ativação dos neurônios geralmente conferem à RNA um caráter não-linear. Isso impacta significativamente a capacidade de aproximação da rede, tornando essa escolha um aspecto fundamental do projeto. As principais funções de ativação serão apresentadas nesta seção.

#### *Função Logística*

A função logística também conhecida como função sigmoide, apresenta um comportamento de transição suave, variando entre 0 e 1. Essa função é contínua e derivável em todo o seu domínio, o que é relevante para a implementação de métodos de otimização que empregam as derivadas da função de custo. Sua saída assume valores reais entre 0 e 1 torna-a uma opção atrativa para estimar probabilidades de classificação a partir dos resultados de uma RNA [14].

#### *Função Tangente Hiperbólica*

A função tangente hiperbólica, apresenta um comportamento semelhante ao da função logística, mas sua saída está no intervalo -1 e 1, o que pode ser vantajoso para trabalhar com dados que possuem simetria significativa em relação a zero. A função é comumente utilizada em RNAs para modelar relações não lineares em tarefas de classificação e regressão.

#### *Função Rectified Linear Unit (ReLU)*

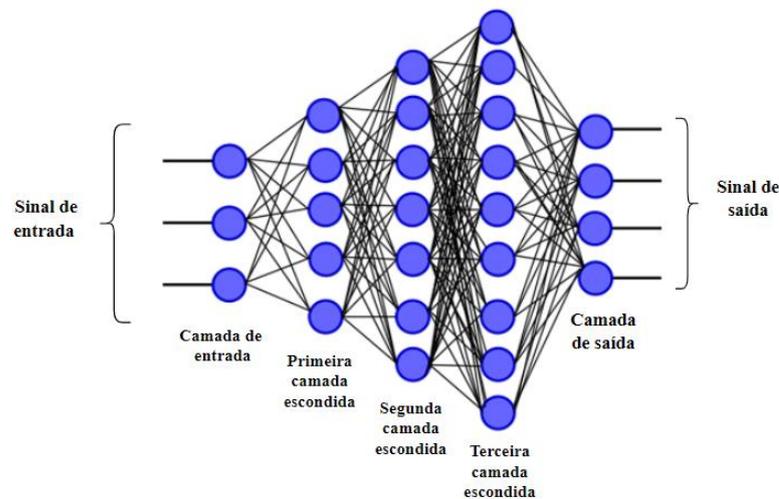
Essa função é nula no intervalo  $[-\infty, 0)$  e linear em  $[0, \infty)$ , introduzindo não-linearidade de forma simples, sem saturação [15]. Sua derivada é informativa na parte positiva, oferecendo vantagens computacionais significativas em redes neurais com muitas camadas. Ela permite que o modelo de aprendizado profundo leve em conta não-linearidades e efeitos de interação específicos.

#### *Função Leaky ReLU*

A *Leaky ReLU* é uma função variante da ReLU, mas apresenta uma leve inclinação para valores negativos em vez de ser horizontal. O coeficiente de inclinação é definido antes do treinamento, ou seja, não é ajustado durante o processo de aprendizado. Essa função de ativação é amplamente utilizada em tarefas onde podemos enfrentar gradientes esparsos, como no treinamento de GAN.

## 2.2.4 A Arquitetura MLP

As redes neurais de múltiplas camadas são compostas por agrupamentos de neurônios organizados em diferentes camadas. Ao incluir camadas adicionais, sejam elas visíveis ou ocultas, a rede é capaz de realizar mapeamentos complexos entre entradas e saídas. Isso permite que a rede tenha uma visão abrangente dos dados de entrada, graças às conexões sinápticas e à interação intensa entre os neurônios. A figura 2.6 apresenta a arquitetura de uma MLP.



**Figura 2.6.** Exemplo de rede MLP.

Os nós à extrema esquerda representam o vetor de entradas, que fornece os sinais para os neurônios da primeira camada oculta e assim sucessivamente. Geralmente, os neurônios em cada camada recebem como entrada os sinais gerados por todos os neurônios da camada anterior. Por essa razão, essa estrutura é também chamada de totalmente conectada ou densa.

Para o treinamento de uma MLP, utiliza-se comumente o algoritmo de propagação (*propagation*) e retropropagação (*backpropagation*), sendo o treinamento realizado em duas fases. Durante a fase de propagação, os pesos sinápticos da rede permanecem constantes e os sinais de entrada são propagados pela rede, camada por camada, até alcançarem a saída. Neste estágio, as alterações ocorrem apenas nos potenciais de ativação e nas saídas dos neurônios da rede. Na etapa de retropropagação, um sinal de erro é gerado ao comparar a saída da rede com a resposta esperada. A derivada da função de erro é então transmitida pela rede, camada por camada, mas de forma inversa, da última camada para a primeira. Durante essa fase, são realizados ajustes contínuos nos pesos sinápticos da rede.

O algoritmo de retropropagação fundamenta-se no gradiente de erro da camada final da rede. Uma vez definida as funções de erro ou custo da rede podemos determinar esse gradiente.

Uma alternativa tradicional em RNAs é o erro quadrático médio (MSE, *Mean-Squared Error*), que é definido pela equação 2.23:

$$\text{MSE} = \frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2 \quad (2.23)$$

Onde  $m$  representa o número de saídas da rede,  $y$  a saída desejada para o vetor de entrada e  $\hat{y}$  a saída gerada pela rede para esse mesmo vetor.

O gradiente é calculado a partir da função de erro e propagado na direção inversa da arquitetura MLP. O algoritmo de retropropagação aplica essas derivadas em todas as camadas da rede, ajustando os parâmetros conforme o algoritmo de otimização escolhido [16]. A MLP possui a capacidade de aproximar funções, conforme o teorema da aproximação universal, que afirma que uma camada escondida com um número finito de neurônios pode aproximar funções contínuas em subconjuntos compactos de  $\mathbb{R}^n$ . Isso permite que a MLP realize transformações não-lineares nos dados de entrada, aproximando-se da saída desejada em um aprendizado supervisionado. Entretanto, uma única camada não garante desempenho ótimo em termos de implementação, tempo de treinamento e generalização, sendo que arquiteturas profundas podem ser mais vantajosas dependendo da aplicação.

### 2.2.5 Aprendizado Profundo

O aprendizado profundo utiliza modelos computacionais com múltiplas camadas que representam dados em diferentes níveis de abstração. Esse método permite a descoberta de estruturas complexas em grandes bases de dados, com cada camada criando suas próprias representações internas. O treinamento de um modelo de aprendizado profundo segue o mesmo processo que uma MLP, utilizando o algoritmo de retropropagação para calcular as derivadas da função de custo em relação aos parâmetros. A partir do aprendizado profundo, muitos problemas tiveram uma melhora significativa em seu desempenho, por exemplo o caso de reconhecimento de fala, reconhecimento de imagens, processamento de linguagem natural, incluindo tarefas como classificação de tópicos e análise de sentimentos.

As técnicas tradicionais de aprendizado de máquina têm limitações no processamento de dados brutos, exigindo um extenso conhecimento do domínio para criar extratores que convertam esses dados em vetores de características úteis. Isso é essencial para que sistemas de aprendizado, como classificadores, possam classificar corretamente os padrões de entrada.

O aprendizado de representação permite que máquinas recebam dados não tratados e descubram automaticamente as representações necessárias para detecções e classificações. O aprendizado profundo utiliza esse conceito em múltiplos níveis, através de módulos simples e não lineares que transformam dados brutos em representações mais abstratas, permitindo que funções complexas sejam aprendidas por meio de sucessivas transformações [17].

Dessa forma, o ponto principal do aprendizado profundo é que as características discriminativas não são projetadas por especialistas humanos, mas sim aprendidas diretamente dos dados enquanto o modelo é treinado para executar a tarefa pretendida.

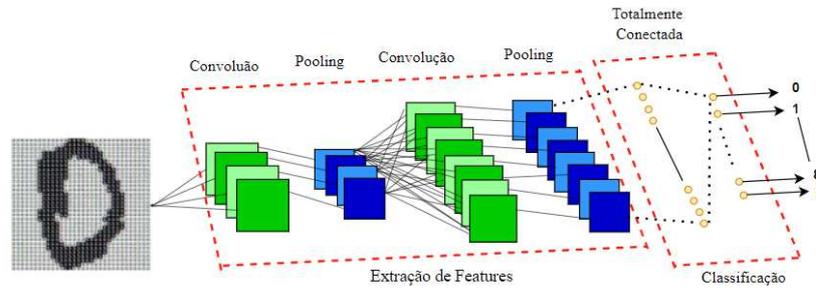
### 2.2.6 Redes Neurais Convolucionais

As redes neurais convolucionais (CNNs, *Convolutional Neural Networks*) são adequadas para o tratamento de dados que estão em formato de tensores. Um exemplo disso são imagens coloridas, que são formadas por três matrizes 2D contendo as intensidades dos pixels nos três canais distintos.

As CNNs têm inspiração biológica, baseando-se no córtex visual dos animais, que contém áreas de células sensíveis a regiões específicas do campo visual [18, 19]. Essas células neurais são ativadas por bordas particulares e estão organizadas em uma arquitetura multinível, permitindo a formação da percepção visual. Em cada etapa individual, uma CNN executa uma operação neural bastante simples, como a combinação linear das entradas ou não linearidades, como exemplo os *thresholds* de ativação ou normalização competitiva [20].

Um dos principais diferenciais das CNNs atualmente está na utilização de um algoritmo de aprendizado supervisionado de ponta a ponta, fundamentado na ideia de retropropagação, aplicado para a atualização dos conjuntos de nós. O algoritmo de retropropagação passou a ser empregado nas CNNs da mesma maneira que nas redes *feedforward* profundas, possibilitando que todos os pesos em todos os conjuntos de nós sejam treinados para um objetivo específico.

As CNNs utilizam quatro principais ideias em seu processo, sendo as primeiras conexões locais, pesos compartilhados, agrupamento ou *Pooling* e o uso de múltiplas camadas. Na figura 2.7 é apresentada a arquitetura de uma CNN.



**Figura 2.7.** Arquitetura da CNN.

### 2.2.6.1 Camada Convolutional

As CNNs utilizam a operação de convolução em vez de multiplicação matricial em pelo menos uma de suas camadas. Nesse contexto, a operação de convolução de um sinal de entrada  $f(x)$  por um kernel  $h(x)$  é expressa pela equação 2.24 [21]:

$$g(x) = (f * h)(x) = \int_{-\infty}^{+\infty} f(\tau) h(x - \tau) d\tau \quad (2.24)$$

O resultado da operação de convolução,  $g(x)$ , é frequentemente denominado mapa de características (*feature map*). Na Equação (2.24), o sinal  $f(x)$  é contínuo. Considerando agora um sinal discreto  $f(i)$ , a operação de convolução é expressa por:

$$g(i) = (f * h)(i) = \sum_{m=-\infty}^{+\infty} f(m) h(i - m) \quad (2.25)$$

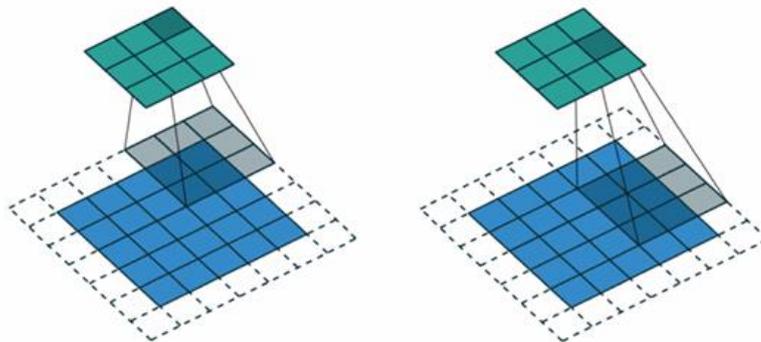
Para aplicações de aprendizado de máquina, a entrada é tipicamente representada por um vetor multidimensional de dados e o *kernel* é composto por um vetor de parâmetros que é ajustado pelo algoritmo de aprendizado. Nesse contexto, a operação de convolução pode ser realizada em espaços multidimensionais, conforme o dado de entrada. No caso de imagens, por exemplo, a operação de convolução da imagem  $f(x,y)$  pelo kernel  $h(x,y)$ , de duas dimensões, é expressa pela equação 2.26.

$$g(i, j) = (f * h)(i, j) = \sum_m \sum_n f(m, n) h(i - m, j - n) \quad (2.26)$$

No processo de convolução de imagens, o *kernel* passa por todos os pixels da imagem, realizando uma multiplicação de seus pesos com os valores desses pixels que estão na

vizinhança definida pelo kernel. Em seguida, os valores são somados e o resultado determina o valor de um elemento (pixel) no mapa de características.

Na figura 2.8 temos um exemplo do processo de convolução para as duas primeiras posições da imagem. Neste exemplo, a imagem é representada pela matriz em azul, enquanto o kernel está associado às células 3×3 em verde. Além disso, houve um preenchimento lateral da imagem original com pixels adicionais de valor nulo (*zero padding*), os quais estão representados em tracejado neste exemplo.



**Figura 2.8.** Modelo de cálculo de convolução das duas primeiras posições da imagem (em azul) pelo *kernel* (em verde).

Um aspecto interessante da operação de convolução em vez da multiplicação matricial é que ela permite lidar com dados de entrada de tamanhos variados. Enquanto RNAs tradicionais utilizam multiplicações matriciais, onde cada saída interage com todas as entradas, as redes convolucionais têm iterações esparsas, utilizando um kernel menor que o tamanho da imagem. Isso permite a detecção de características pequenas e discriminativas, como bordas, e mesmo em imagens com milhões de pixels, com um *kernel* ocupando apenas alguns pixels. Assim, menos parâmetros precisam ser armazenados, reduzindo os requisitos de memória, processamento e melhorando a eficiência estatística.

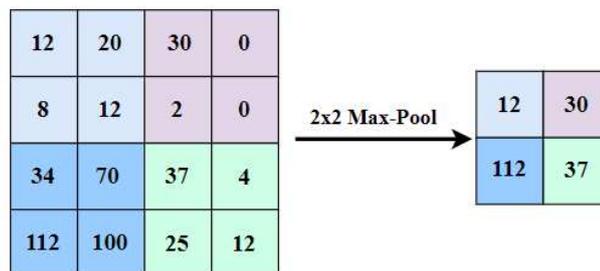
Outro ponto importante da CNN é a definição de *stride*, que é a quantidade de pixels deslocados entre o processamento de cada bloco da matriz original. Sendo assim, um grande volume de entrada (como exemplo imagens com três canais de cores) resultam em um volume de saída, onde as dimensões dependem do *stride*, da presença ou não de *pooling* e das dimensões do *kernel*.

### 2.2.6.2 Camada Não-Linear

Após a operação de convolução, os mapas de características são submetidos a uma função de ativação não-linear. Neste caso, diversas funções podem ser utilizadas, como as sigmóides aplicadas em redes totalmente conectadas. Contudo, as mais frequentemente empregadas no contexto de CNNs são as do tipo ReLU.

### 2.2.6.3 Camada de Agrupamento (*Pooling*)

A função de *pooling* combina a informação de células adjacentes em um mapa de características, geralmente resultando em uma diminuição da dimensionalidade dos dados. Por exemplo, a função de *max-pooling* determina o valor máximo dentro de uma área retangular. Outras funções de *pooling* levam em consideração a vizinhança de uma região retangular para calcular a média ou mesmo a média ponderada dos *pixels* em torno da célula central. Na Figura 2.9 é apresentado um exemplo da função *max-pooling* aplicada a uma imagem de 16 *pixels* com tamanho de passo (*stride*) igual a 2.



**Figura 2.9.** Exemplo de *max-pooling* aplicado a matriz com tamanho 2 x 2 e *stride* 2.

A função de *pooling* é importante porque torna a representação quase invariante a pequenas translações na entrada, mantendo os valores da saída inalterados. Isso é útil para problemas de classificação. Além disso, ela reduz a dimensionalidade dos dados, diminuindo a necessidade de poder computacional e memória para processamento e armazenamento.

### 2.2.6.4 Camada Totalmente Conectada

A partir de várias camadas convolucionais, a rede é capaz de extrair características discriminativas dos dados, e esta configuração pode ser aplicada em tarefas de classificação e regressão. No processamento de imagens, camadas totalmente conectadas são geralmente empregadas após as camadas convolucionais para processar essas características até a saída da rede. Em algumas arquiteturas, a rede processa em paralelo inúmeras matrizes. Para imagens coloridas, a entrada é representada por matrizes dos canais R, G e B. Cada camada

convolucional contém um conjunto de *kernels*, que são processados pela camada seguinte. Antes de entrar em uma camada densa, os atributos, representados por um tensor com múltiplos canais, devem ser achatados (*flatenned*) para formar um único vetor. Cada camada realiza um mapeamento não-linear aplicando uma função de ativação ao resultado da soma das entradas ponderadas pelos pesos até chegar à camada de saída.

### 2.2.6.5 Camada de Saída

Para tarefas de classificação, as CNNs geralmente empregam na camada final uma função que calcula a probabilidade de um dado pertencer a cada classe disponível. Nesse cenário,  $n$  saídas são utilizadas no classificador, correspondendo a  $n$  classes. Para cada entrada, é utilizada uma codificação de 0's e 1's, conhecida em inglês como *one-hot encoding*.

A partir da saída da rede  $y$ , a função de erro pode ser determinada e os gradientes propagados na direção oposta, utilizando o algoritmo de retropropagação, para atualizar os pesos das camadas totalmente conectadas, assim como os *kernels* das camadas convolucionais.

### 2.2.7 Redes Adversárias Generativas

As Redes Adversárias Generativas (GAN, *Generative Adversarial Networks*) foram propostas por Goodfellow *et al.* em 2014 [4]. As GANs são compostas por duas redes, a geradora e a discriminadora. Esses modelos generativos são capazes de representar a distribuição de dados aprendida por meio da síntese de amostras semelhantes às do conjunto de treinamento.

As GANs baseiam-se na teoria dos jogos não-cooperativo de soma zero (ou teorema minmax), onde um dos oponentes quer maximizar suas ações e o outro minimizá-las. A arquitetura é desenvolvida para um treinamento adversário entre os dois agentes: o gerador e o discriminador, ambos geralmente implementados como RNAs que competem entre si. Nesse cenário, as redes  $G$  e  $D$  competem entre-se, com sua distância mensurada pelo funcional  $F_{GAN}(G, D)$ , que é definido pela equação 2.27.

$$F_{GAN}(G, D) = E_{(x \sim p_{data})} [\log D(x)] + E_{(z \sim P_{noise})} [\log(1 - D(G(z)))] \quad 2.27$$

Onde  $x$  é um dado amostral,  $p_{data} \subset x$  é a distribuição dos dados amostrais,  $z$  é um ruído  $P_{noise} \subset z$  é a distribuição do ruído. Durante o treinamento, o gerador gradualmente aprimora sua capacidade de criar dados semelhantes aos reais, enquanto o discriminador se torna mais

eficaz em diferenciá-los. Assim, o processo atinge um equilíbrio quando o discriminador não consegue mais distinguir imagens reais de falsificações. Quando treinados adequadamente, espera-se que tanto o gerador quanto o discriminador alcancem o equilíbrio de Nash, onde nenhum dos dois consegue obter vantagem sobre o outro, e cada jogador não altera sua ação independentemente da escolha do oponente [22].

O objetivo do gerador é criar amostras que imitem tão bem as características do conjunto de dados de treinamento que pareçam indistinguíveis. O gerador, que funciona como um modelo inverso de reconhecimento de objetos, aprende a criar padrões a partir de vetores aleatórios, recebendo *feedback* do discriminador [23].

O Discriminador avalia se um exemplo é real (do conjunto de treinamento) ou falso (gerado) e melhora conforme aprende com suas classificações. Quando o discriminador é enganado, o gerador percebe que está indo bem, mas quando classifica corretamente uma amostra falsa, o gerador entende que precisa aprimorar. Assim, ambos os modelos evoluem simultaneamente, com o gerador aprimorando a geração de dados realistas e o discriminador se tornando mais eficiente na distinção entre dados reais e falsos [24].

### **2.2.7.1 Arquitetura da GAN**

Inicialmente, os modelos gerador e discriminador eram implementados como *perceptrons* multicamadas. Em 2015, Radford *et al.* [25] propuseram a (DCGAN, *Deep Convolutional GAN*), onde os modelos foram implementados como redes neurais convolucionais profundas.

A arquitetura geral da DCGAN apresenta o gerador e o discriminador como redes neurais profundas, com o conjunto de dados contendo amostras reais. O gerador recebe um vetor de ruído aleatório e gera uma imagem falsa, que é enviada ao discriminador. Este último determina se a imagem provém do ruído ou dos dados reais. O ruído passa pela primeira camada do gerador, chamada de "*project and reshape*", que expande os dados de entrada. Em seguida, blocos convolucionais duplicam o tamanho das saídas, reduzindo gradualmente o número de canais até alcançar a quantidade desejada na imagem final. O discriminador, por sua vez, tem uma estrutura similar às redes de classificação e realiza operações opostas às do gerador, resultando em uma classificação binária.

### 2.2.7.2 Espaço Latente da GAN

Ao finalizar o treinamento da GAN, um vetor de ruído aleatório  $Z$  oferecido ao gerador irá gerar uma nova imagem  $x$ , que se assemelha às amostras do conjunto de dados no qual a rede foi treinada, mas com características aleatórias, dependendo da escolha do vetor de ruído  $Z$ . Para enfrentar essa aleatoriedade na geração, que é própria do treinamento da GAN, existem várias estratégias para compreender os fatores de variabilidade do modelo e, assim, controlar as características desejadas na criação de uma imagem  $x$ . Essas estratégias incluem: geração condicional, aritmética vetorial e identificação automática de características.

### 2.2.8 Redes Adversárias Generativas Condicionais

A GAN como formulada originalmente é capaz de gerar exemplos plausíveis dentro de uma dada distribuição de um conjunto de dados, porém, não possui formas de controlar como será a instância gerada. Apenas considera que essa instância seja coerente à distribuição para a qual foi treinada.

A rede generativa adversária condicional (CGAN, *Conditional Generative Adversarial Network*) é uma extensão das GANs para um modelo condicional. Nessa abordagem tanto gerador quanto discriminador são condicionados com alguma informação chamada rótulo através de uma camada de entrada adicional [4].

A CGAN foi criada em 2014 por Mehdi Mirza e Simon Osindero [26], é uma rede generativa adversária que utiliza informações adicionais durante seu treinamento. Essas informações podem ser, por exemplo, rótulos de classe, *tags* ou descrições escritas. Para facilitar a compreensão, vamos focar nos rótulos como dados de condicionamento.

Durante o treinamento da CGAN, o gerador aprende a criar exemplos realistas para cada rótulo do conjunto de dados, e o discriminador, por sua vez, deve distinguir entre pares de rótulos e exemplos reais e falsos. Assim, para enganar o Discriminador, o Gerador não precisa apenas criar imagens realistas, essas imagens também devem corresponder aos rótulos fornecidos. Após o treinamento completo do Gerador, podemos especificar exatamente qual exemplo queremos que o CGAN crie ao fornecer o rótulo desejado.

#### 2.2.8.1 Rede Geradora da CGAN

Vamos chamar o rótulo que usamos como referência de  $y$ . O Gerador combina um vetor de ruído  $Z$  com esse rótulo  $y$  para criar um exemplo falso, representado como  $G(z, y) = x^*|y$ .

Isso significa que  $x^*$  é o exemplo gerado, condicionado ao rótulo  $y$ . O objetivo do gerador é produzir um exemplo que pareça o mais realista possível para o discriminador, de forma que ele se assemelhe a um exemplo verdadeiro que corresponda ao rótulo fornecido.

### **2.2.8.2 Rede Discriminadora da CGAN**

O discriminador analisa dois tipos de entradas: exemplos reais com rótulos, que são representados como  $(x, y)$ , e exemplos falsos, que vêm com o rótulo utilizado para gerá-los, representados como  $(x^*|y, y)$ . Nos pares reais de exemplo e rótulo, ele aprende a reconhecer dados reais e a identificar pares correspondentes. Nos exemplos gerados pelo gerador, o discriminador aprende a identificar pares imagem-rótulo falsos, e distinguindo-os dos reais. No final, o Discriminador emite uma probabilidade que expressa o quanto ele acredita que a entrada é um par verdadeiro e correspondente. O objetivo principal do Discriminador é rejeitar todos os exemplos falsos e aqueles cujos rótulos não correspondem, enquanto aceita todos os pares reais de exemplo e rótulo.

## CAPÍTULO 3

### Primeiro Estudo de Caso

No primeiro estudo de caso abordaremos três diferentes arquiteturas de redes neurais, iniciando pela GAN, uma rede capaz de gerar dados sintéticos similares aos dados de treinamento. A segunda arquitetura é uma MLP, com finalidade de prever as características da antena. E a terceira arquitetura explorada foi uma CGAN, que foi modelada para gerar dados de características de antena condicionada a uma classe referente a uma faixa de frequência específica. Para geração do conjunto de dados para o treinamento dos modelos, foram utilizadas fórmulas CAD para os cálculos das características das antenas considerando frequências de 1 a 30GHz. Este primeiro estudo de caso é subdividido em três experimentos descritos na próxima seção.

#### 3.1 Estratégia com RNAs

Na primeira abordagem (primeiro experimento) utilizamos GANs para gerar dados sintéticos das características de antena em microfitas retangulares e, para isso, considerou-se parâmetros como: frequência de ressonância ( $f_r$ ), permissividade relativa do substrato ( $\epsilon_r$ ), espessura do substrato ( $h$ ), largura e comprimento do *patch* ( $W$ ) e ( $L$ ), e as características de desempenho como a diretividade ( $D$ ), e largura de banda ( $BW$ ).

Já o segundo experimento consistiu em modelar uma MLP para prever os dados de largura e comprimento do *patch*, diretividade e a largura de banda, a partir dos dados de ( $f_r$ ), ( $h$ ) e ( $\epsilon_r$ ). Uma vez encontrada a especificação ideal do modelo, a rede foi capaz de aprender a calcular as características da antena com base em seu conjunto de dados de entrada.

No terceiro experimento foi modelada uma CGAN, que permite condicionar a rede com informações adicionais, como rótulos de classes. Para este experimento condicionamos a rede a gerar características de antenas em uma faixa de frequência de 1 a 5 GHz.

##### 3.2.1 Geração da Base de Dados

Inicialmente foi gerado um conjunto de dados de frequências numa faixa de 1 a 30 GHz que abrange bandas de frequências relacionadas à tecnologia 5G. Em seguida, escolhemos os substratos Duroid e RO3003 que apresentam os valores de permissividades relativas de 2,20 e 3,00, respectivamente, para o valor da espessura do substrato  $h = 1,575\text{mm}$ , e a partir destes

dados iniciais, foram então obtidas as quatro características referentes a largura do *patch* antena  $W$ , o comprimento do *patch* antena  $L$ , a diretividade  $D$ , a largura de banda  $BW$  calculados a partir das fórmulas CAD, equações de 2.1 a 2.20, apresentadas no Capítulo 2.

O banco de dados é composto por 500 amostras referentes a sete características das antenas, sendo estas a frequência de ressonância  $f_r$ , a espessura do substrato  $h$ , a permissividade do substrato  $\epsilon_r$ , largura do *patch* antena  $W$ , o comprimento do *patch* antena  $L$ , a diretividade  $D$ , a largura de banda  $BW$ . Antes de iniciarmos o treinamento do modelo a base de dados passou pela etapa de preparação dos dados, onde os dados são normalizados, permitindo que as diferenças possam ser analisadas de forma mais assertiva.

### 3.2.2 Arquitetura GAN

Para a GAN, o treinamento é um processo contraditório no qual uma rede discriminadora estima se um determinado exemplo é produzido pelo gerador (falso) ou é do conjunto de dados (reais). À medida que o treinamento avança, o gerador aprende aproximações cada vez melhores para a distribuição de dados e ficará mais difícil para o discriminador diferenciar entre a saída do gerador e o conjunto de dados reais.

#### *Rede Geradora*

O modelo gerador toma como entrada um ponto do espaço latente e gera uma nova amostra. O espaço latente é um espaço vetorial arbitrariamente definido de valores distribuídos gaussianos, por exemplo, 100 dimensões. O Gerador inicia com um vetor de ruído (espaço latente) definido com 100 dimensões, seguido das camadas intermediárias, onde a primeira camada configurada com tamanho de neurônios (ou nós) de 128, a segunda camada com 256 neurônios, a terceira com 512 neurônios e a camada de saída de tamanho 7 correspondendo a  $f_r$ ,  $h$ ,  $\epsilon_r$ ,  $W$ ,  $L$ ,  $D$  e  $BW$ . Foi adicionada uma camada de normalização de *batch* entre as camadas intermediárias que contribuiu para uma maior estabilidade da rede geradora durante o treinamento. Em seguida, aplicamos a normalização em lote e a função de ativação *Leaky ReLU*. Na camada final, não aplicamos normalização em lote e, em vez de *ReLU* é implementada com não linearidade sigmóide. Todas as configurações da rede foram definidas de forma empírica, onde foram realizados vários testes para chegar ao melhor modelo que nos resultasse um bom desempenho da rede.

### *Rede Discriminadora*

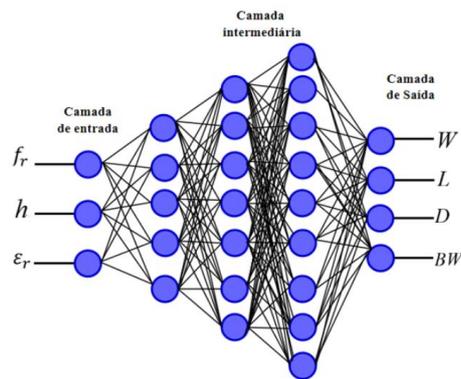
A rede discriminadora recebeu como entrada sete nós (neurônios), sendo eles:  $f_r$ ,  $h$ ,  $\varepsilon_r$ ,  $W$ ,  $L$ ,  $D$  e  $BW$ . Em seguida, foram adicionadas três camadas totalmente conectadas, sendo a primeira camada configurada com 256 neurônios, a segunda camada com 128 neurônios, a terceira com 512 neurônios. Entre cada camada foram definidas a função de ativação *LeakyReLU*, com o coeficiente de inclinação igual a 0,2. O motivo da escolha desta função de ativação é o mesmo descrito para o caso da rede geradora. Para a camada de saída temos um nó para a classificação binária usando a função de ativação linear. Após alguns testes com diferentes funções de ativação que atenderia o modelo, a linear foi a que apresentou melhor desempenho.

### *Rede GAN*

O procedimento de treinamento da GAN envolveu a atualização dos pesos da rede geradora e discriminadora ao longo de 5.000 épocas. Durante cada época, foram realizadas atualizações nos pesos de ambas as redes. O modelo empregou lotes de 2.000 amostras para o conjunto de treinamento. O modelo será atualizado em lotes, especificamente com uma coleção de amostras reais e uma coleção de amostras geradas. No treinamento, a época é definida como uma passagem por todo o conjunto de dados de treinamento. O algoritmo de otimização selecionado para o treinamento tanto da rede discriminadora quanto da geradora foi o Adam [27]. Os parâmetros utilizados no algoritmo de otimização Adam incluíram uma taxa de aprendizado de  $\alpha = 0,001$ , beta 1 = 0,9, beta 2 = 0,999 e uma taxa de decaimento de  $10^{-8}$ . O modelo GAN tem como saída a classificação binária, ou seja, a probabilidade de a amostra ser real ou falsa.

### **3.2.3 Arquitetura MLP**

O processo de treinamento de uma MLP envolve método iterativo de ajuste de pesos para minimizar o erro entre a saída da rede e a saída desejada, conforme pode ser visto no diagrama da Figura 3.1.



**Figura 3.1.** Diagrama do procedimento de treinamento da MLP.

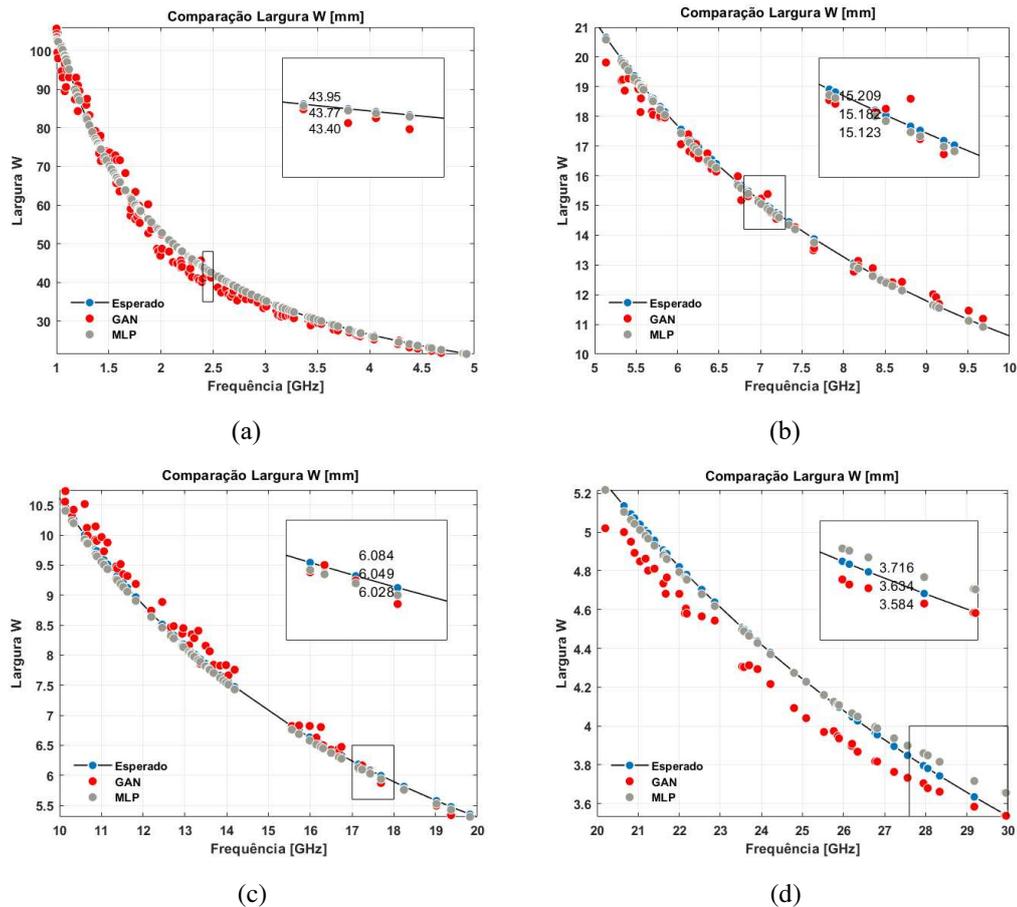
O modelo MLP é constituído por uma camada de entrada de dimensão três, que corresponde aos parâmetros da antena referente a  $f_r$ ,  $h$ ,  $\varepsilon_r$ . Apresenta três camadas intermediárias, sendo a primeira configurada com 256 neurônios, a segunda camada com 128 neurônios e a terceira com 512 neurônios. A camada de saída, de tamanho quatro, refere-se aos parâmetros  $W$ ,  $L$ ,  $D$ ,  $BW$  e com função de ativação do tipo *sigmóide*. Para o treinamento da MLP foram definidas 2.000 épocas.

### 3.3 Resultados

Este experimento teve como objetivo prever os dados usando uma MLP e gerar dados usando uma GAN. As redes foram treinadas com dados calculados a partir das fórmulas CAD, com uma faixa de frequência de 1 a 30GHz com os substratos Duroid ( $\varepsilon_r = 2,20$ ) e RO3003 ( $\varepsilon_r = 3,30$ ). Para uma análise mais precisa e apresentação dos resultados com maior clareza, foram gerados gráficos com subconjuntos de faixas de frequências de 0 a 5GHz, de 5 a 10GHz, 10 a 20GHz e 20 a 30GHz. Ademais, os gráficos estão organizados de acordo com os parâmetros  $W$ ,  $L$ ,  $D$ ,  $BW$  e sua dependência com a frequência. Nesses gráficos são apresentadas as seguintes distinções entre dados: “Esperados”: dados calculados pelas fórmulas CAD (pontos em azul); “GAN”: dados gerados pela GAN (pontos em vermelho) e “MLP”: dados preditos pela rede MLP (pontos em cinza).

#### 3.3.1 Dados gerados da largura ( $W$ )

A figura 3.2 apresenta os gráficos com os dados de  $W$  em função de  $f_r$ , para uma análise comparativa dos dados da MLP e da GAN em relação aos dados esperados.

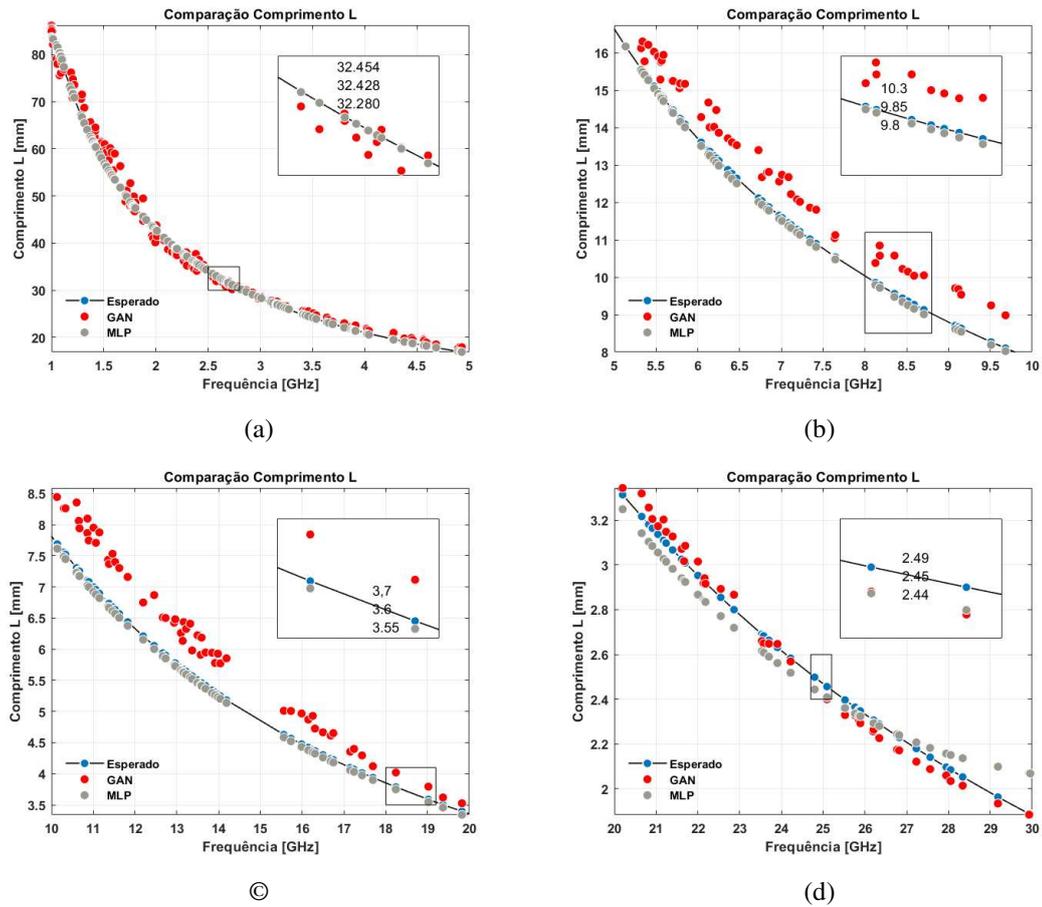


**Figura 3.2.** Comparação entre os dados de  $W$  esperados, os gerados pela GAN e os preditos pela MLP em diferentes faixas de frequências e para permissividades distintas. (a) Frequências entre 1 a 5GHz (b)Frequências entre 5 a 10GHz (c) Frequências entre 10 a 20GHz e (d) Frequências entre 20 a 30GHz.

Nas figuras 3.2(a), com frequências entre 1 a 5GHz, e na figura 3.2(b), frequências entre 5 a 10GHz, observamos que os dados preditos pela MLP e os gerados pela GAN apresentam valores similares aos dados esperados. No entanto, os dados da MLP têm uma maior aproximação quando comparados aos dados gerados pela GAN. Já as figuras 3.2 (c) e (d) com faixas de frequências de 10 a 20GHz e de 20 a 30GHz, respectivamente, observamos que os dados gerados pela GAN apresentaram valores mais aproximados aos esperados em comparação aos dados da MLP.

### 3.3.2 Dados gerados do comprimento ( $L$ )

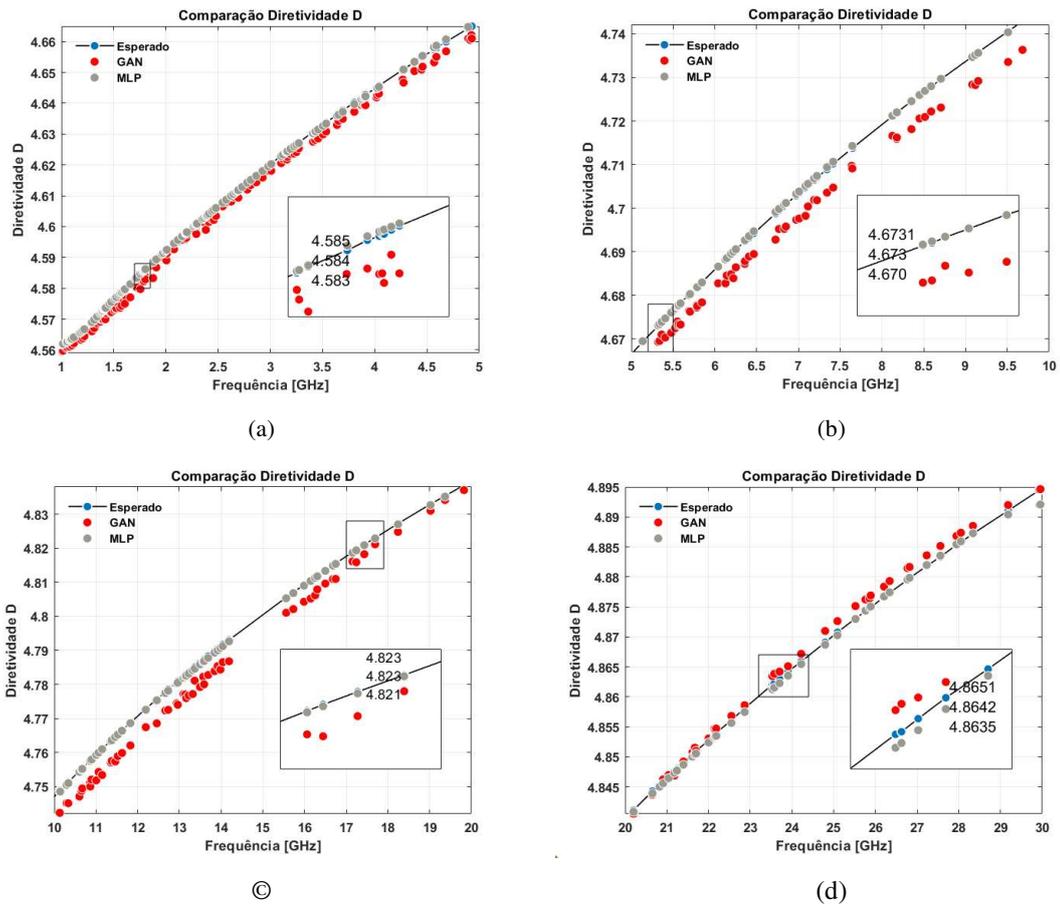
Na figura 3.3, são apresentados os gráficos com os dados de  $L$  em função da  $f_r$ . Nas figuras 3.3(a), 3.3(b) e 3.3(c), observamos que os valores da MLP têm uma maior similaridade com os dados esperados em comparação aos dados gerados pela GAN. No entanto, ambas as redes apresentam valores aproximados. Já na figura 3.3(d), observamos que os resultados da GAN tiveram uma maior similaridade com os dados esperados em comparação à MLP.



**Figura 3.3.** Comparação entre os dados de  $L$  esperados, os gerados pela GAN e os preditos pela MLP em diferentes faixas de frequências e para permissividades distintas. (a) Frequências entre 1 a 5GHz (b)Frequências entre 5 a 10GHz (c) Frequências entre 10 a 20GHz e (d) Frequências entre 20 a 30GHz.

### 3.3.3 Dados gerados de diretividade ( $D$ )

Na figura 3.4, são apresentados os gráficos com os dados de  $D$  em função da  $f_r$ . Pode-se observar uma boa similaridade entre os dados preditos pela MLP e pela GAN com os dados esperados.

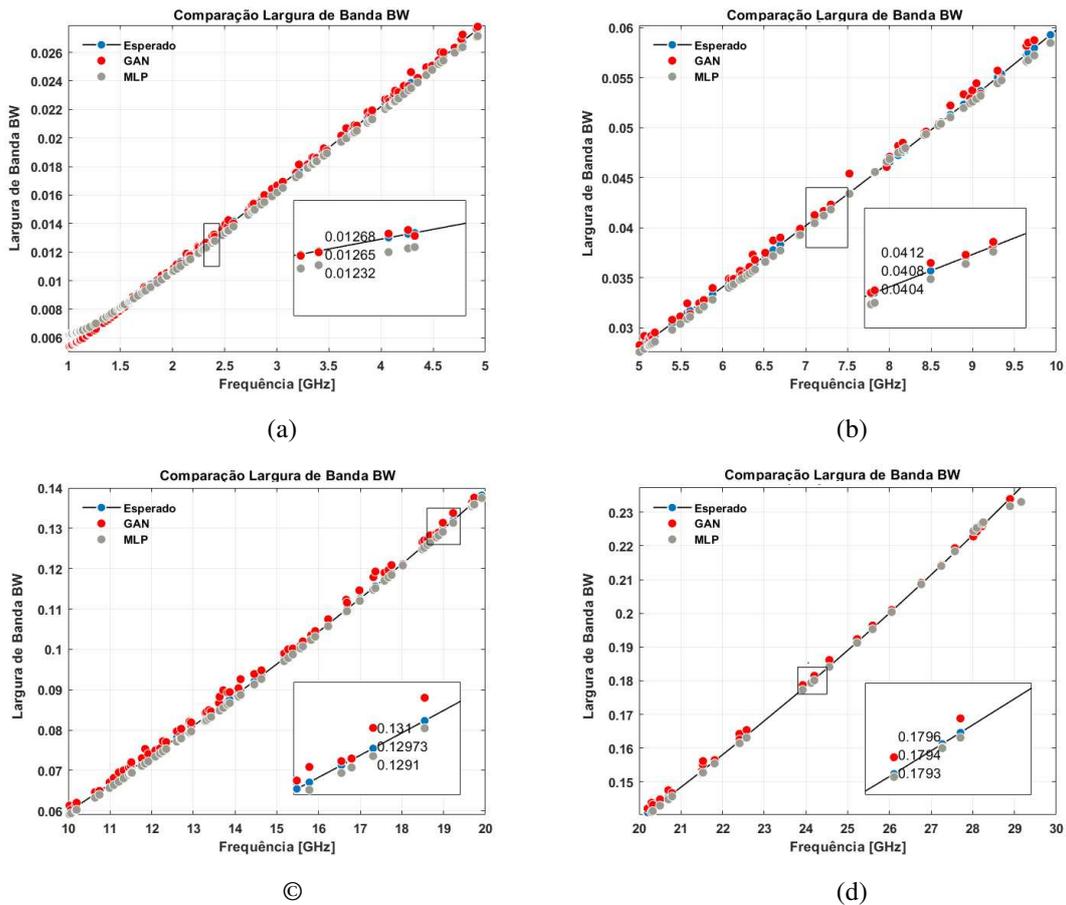


**Figura 3.4.** Comparação entre os dados de  $D$  esperados, os gerados pela GAN e os preditos pela MLP em diferentes faixas de frequências e para permissividades distintas. (a) Frequências entre 1 a 5GHz (b) Frequências entre 5 a 10GHz (c) Frequências entre 10 a 20GHz e (d) Frequências entre 20 a 30GHz.

### 3.3.4 Dados gerados de largura de banda ( $BW$ )

Na figura 3.5, são apresentados os dados da  $BW$  em função de  $f_r$ . Pode-se observar uma boa similaridade entre os dados preditos pela MLP e pela GAN em comparação aos dados esperados.

Observamos que nas figuras 3.5(a) e (b) que os valores gerados pela GAN apresentam uma maior similaridade em relação a MLP quando comparados aos dados esperados, no entanto, ambos os valores dos modelos têm uma boa aproximação com os dados esperados. Já nas figuras 3.5(c) e (d) o modelo MLP apresenta uma maior similaridade em comparação ao modelo GAN. No entanto, ambos os modelos também apresentam valores próximos aos dados esperados.



**Figura 3.5.** Comparação entre os dados de  $W$  esperados, os gerados pela GAN e os preditos pela MLP em diferentes faixas de frequências e para permissividades distintas. (a) Frequências entre 1 a 5GHz (b)Frequências entre 5 a 10GHz (c) Frequências entre 10 a 20GHz e (d) Frequências entre 20 a 30GHz.

Para os resultados apresentados nas figuras 3.2 até 3.5, foram calculados os erros para avaliação da precisão dos modelos. Este conjunto de erros são apresentados nas tabelas 3.1 e 3.2. No caso do parâmetro  $W$ , o erro relativo médio (MRE) da GAN atinge um valor de 3,26%, que é bastante elevado em comparação ao MRE obtido pela MLP, ou seja, 0,6%. Em todas as outras comparativas, a MLP apresentou um desempenho superior quando comparada a GAN. Isto pode ser explicado porque a MLP aprende a calcular o conjunto de características a partir dos dados de entrada, já no caso da GAN, ela aprende a sintetizar novos dados a partir do conjunto de treinamento.

**Tabela 3.1.** Comparativo entre os erros da GAN e MLP para  $W$  e  $L$ .

Erro	$W$		$L$	
	GAN	MLP	GAN	MLP
MAE	$1,23 \times 10^{-2}$	$2,11 \times 10^{-4}$	$9,33 \times 10^{-4}$	$1,55 \times 10^{-4}$
MRE	0,0326	$6,22 \times 10^{-3}$	$5,01 \times 10^{-2}$	$8,83 \times 10^{-3}$
MSE	$4,20 \times 10^{-6}$	$3,29 \times 10^{-7}$	$1,87 \times 10^{-6}$	$1,99 \times 10^{-7}$
RMSE	$1,18 \times 10^{-3}$	$3,31 \times 10^{-5}$	$7,91 \times 10^{-5}$	$2,58 \times 10^{-5}$

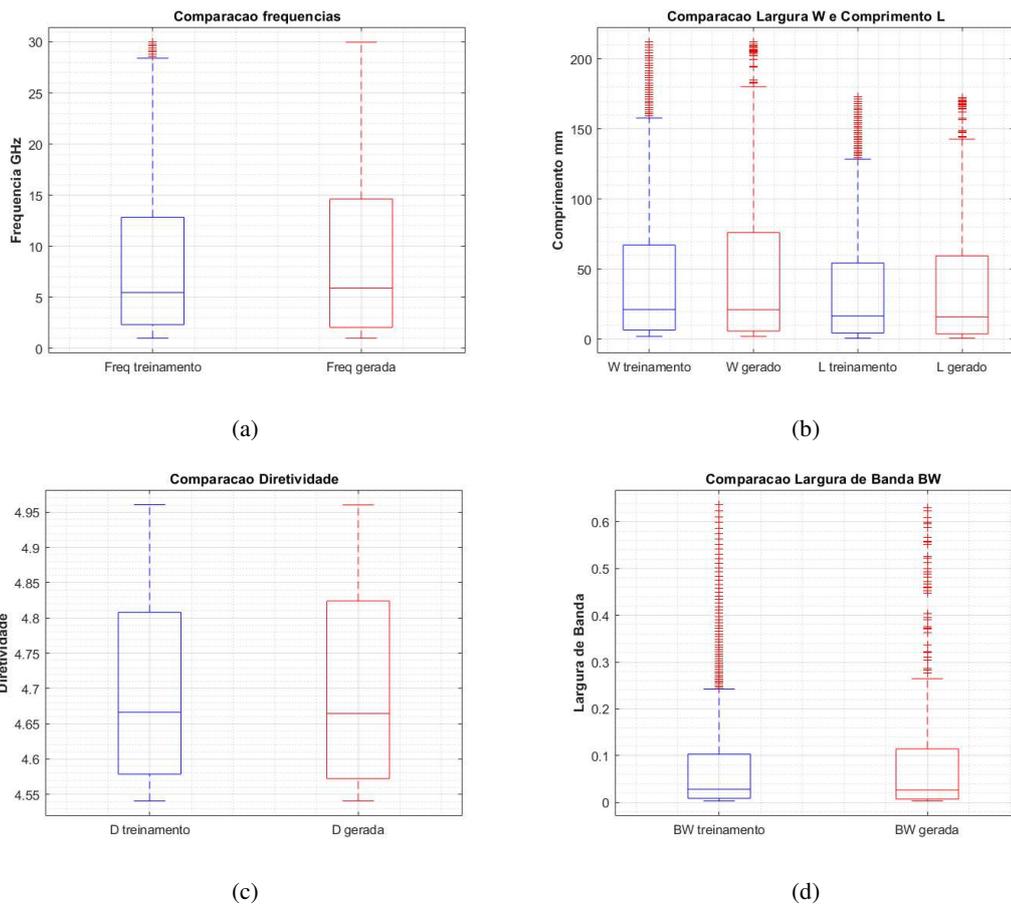
**Tabela 3.2.** Comparativo entre os erros da GAN e MLP para  $D$  e  $BW$ .

Erro	$D$		$BW$	
	GAN	MLP	GAN	MLP
MAE	$3,17 \times 10^{-3}$	$3,46 \times 10^{-4}$	$2,33 \times 10^{-3}$	$2,56 \times 10^{-4}$
MRE	$6,75 \times 10^{-4}$	$7,41 \times 10^{-5}$	$3,15 \times 10^{-2}$	$1,32 \times 10^{-2}$
MSE	$1,43 \times 10^{-5}$	$3,72 \times 10^{-7}$	$1,58 \times 10^{-5}$	$2,45 \times 10^{-7}$
RMSE	$2,18 \times 10^{-4}$	$3,52 \times 10^{-5}$	$2,30 \times 10^{-4}$	$2,86 \times 10^{-5}$

### 3.4 Comparação entre os dados reais e gerados usando *box-plots*

A figura 3.6 apresenta os *box-plots* onde são comparados os dados de treinamento (em azul) e os dados gerados pela GAN (em vermelho). A figura 3.6(a) apresenta o comparativo entre as frequências usadas para treinamento e as geradas pela GAN, onde pode ser claramente observado que a GAN aprendeu a gerar frequências na mesma faixa que os dados de treinamento.

No caso dos parâmetros  $W$  e  $L$ , apresentados na figura 3.6(b), observa-se uma boa similaridade entre os dados, o que indica que a GAN teve uma boa distribuição dos dados. De forma análoga, nas figuras 3.6(c) e (d), que apresentam os resultados para os parâmetros  $D$  e  $BW$ , respectivamente, observamos uma grande similaridade entre os dados de treinamento e os dados gerados, o que significa que o modelo apresentou uma boa capacidade de aprendizado estatístico dos parâmetros mantendo uma boa distribuição dos dados.



**Figura 3.6.** Medida de avaliação de distribuição dos dados. (a) Frequência real e gerada (b) Largura e comprimento reais e gerados (c) Diretividade real e gerada (d) Largura de banda real e gerada.

### 3.5 Treinamento CGAN

#### *Rede Geradora*

Como entrada, a rede geradora da CGAN, recebeu informações da variável latente  $Z$  e a variável condicional  $C$ . Similar a rede anterior, a variável latente recebeu um vetor de 100 posições, e a variável condicional  $C$  um vetor com 4 posições que condiciona a rede a gerar geometrias de antenas para as seguintes faixas de frequências, “0” para faixa de 1 a 2GHz, “1” para a faixa de 2 a 3GHz, “2” para faixa de 3 a 4GHz, “3” para faixa de 4GHz a 5GHz. O rótulo da classe é então passado por uma camada *Embedding* com tamanho 50. Isso significa que cada uma das 5 classes do conjunto de dados será mapeada para uma representação vetorial diferente de 50 elementos que será aprendido pelo modelo gerador. Esse valor de *Embedding* foi escolhido após a realização de testes com diferentes valores descritos na literatura. Os valores dessas quatro variáveis são então concatenados e passados para uma camada densa de 128 nós com uma função de ativação *LeakyReLU*. Em seguida, temos mais uma camada densa com 256

nós. Entre cada uma das camadas anteriores tem-se a adição da camada de normalização de *batch*. Antes da terceira camada temos uma camada de *flatten* e, em seguida, uma camada densa com 512 nós e a camada de saída com valor igual a 7, referente às características das antenas com a função de ativação sigmóide. O modelo do gerador deve ser atualizado para receber o rótulo da classe, tendo o efeito de condicionar o ponto no espaço latente ao rótulo da classe fornecida.

### *Rede Discriminadora*

A rede discriminadora recebe como entrada 7(sete) nós, que referem aos dados de características das antenas, bem como, o rótulo das classes. Assim como na rede geradora, o rótulo de classes passa por uma camada de *Embedding*. Em seguida, temos mais 2 camadas densas, sendo a primeira com 512 nós e a segunda com 256. Todas as camadas da rede discriminadora são seguidas das camadas de função de ativação *LeakyReLU* com o coeficiente de inclinação negativa igual a 0,2. Na sequência, temos uma camada de *flatten*, seguida de outra camada densa de 128 nós. Para a camada de saída temos 1 nó para a classificação binária usando a função de ativação linear.

### *Rede GAN*

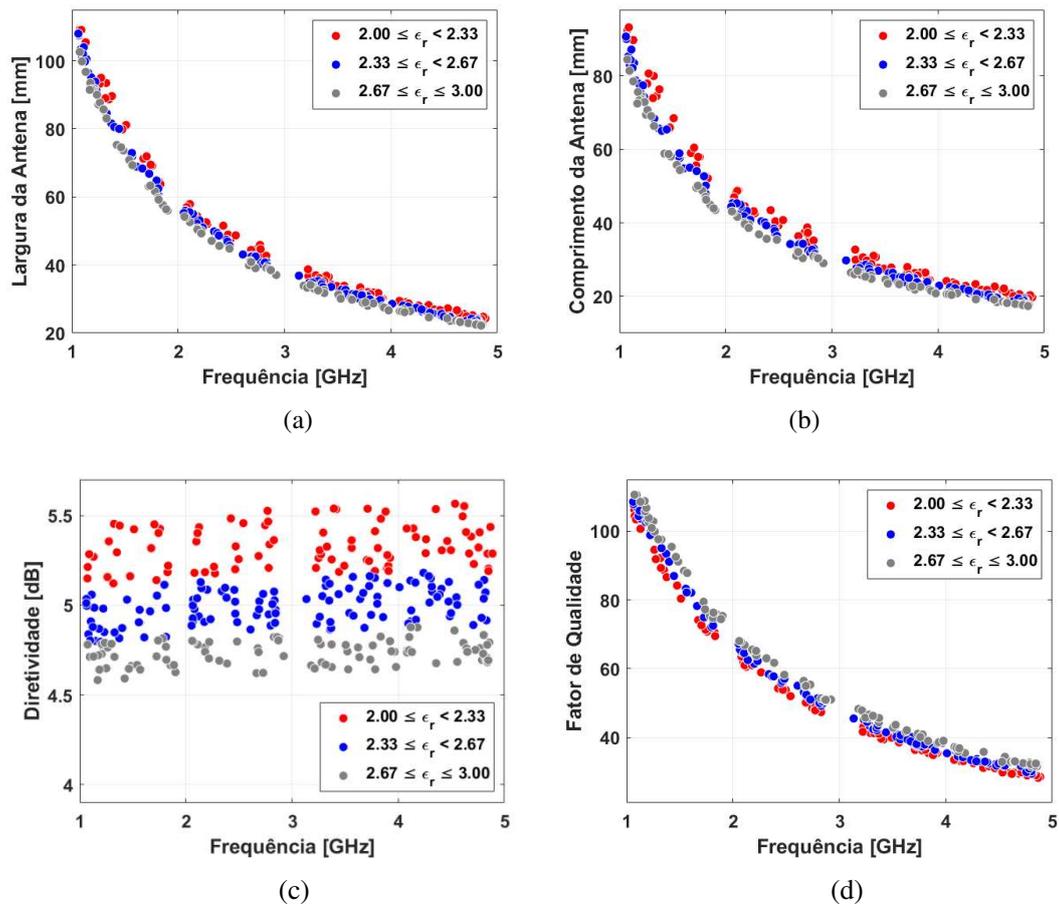
O treinamento da GAN consistiu em atualizar os pesos da rede geradora e da discriminadora utilizando algoritmo de *backpropagation* durante 10.000 épocas. Em cada uma das épocas foram feitas atualizações de pesos da rede discriminadora e da rede geradora. O modelo utilizou *batches* de 2.000 amostras para o conjunto de treinamento.

## **3.5.1 Resultados da CGAN**

### **3.5.1.1 Dados gerados condicionados a faixa de frequência de 1 a 5GHz**

Para este experimento foram geradas 300 amostras de dados referentes a classe “1”, que representa a faixa de frequência de 2 a 3GHz. A figuras 3.7 apresenta os gráficos dos dados gerados pelo modelo CGAN com permissividades relativas entre 2,0 e 3,0.

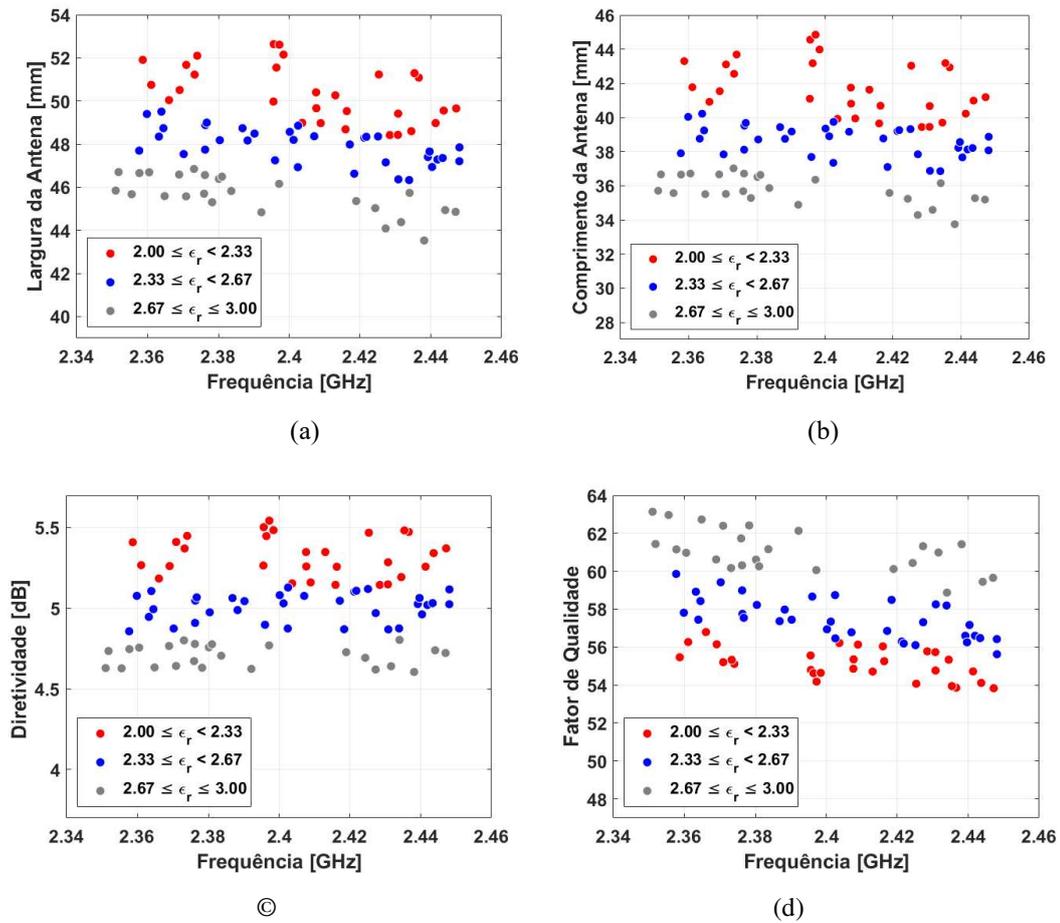
Na figura 3.7(a) temos os gráficos com os dados de  $W$  em função da  $f_r$  distribuídos em três faixas de permissividades. Em vermelho, a faixa de permissividades de 2 até 2,33; em azul, a faixa de permissividades entre 2,33 e 2,67; e em cinza, a faixa de permissividades de 2,67 até 3,0. De maneira análoga, temos nas figuras 3.7(b) o comprimento, 3.7(c) a diretividade e 3.7(d) o fator de qualidade ( $Q_r$ ).



**Figura 3.7.** Dados gerados de características de antenas em microfitas para permissividades entre 2,0 a 3,0 e em função de  $f_r$  para (a)  $W$ , (b)  $L$ , (c)  $D$  e (d)  $Q_r$ .

Em seguida realizamos um segundo experimento para filtrar os dados que foram gerados para frequências entre 2,34 a 2,46 GHz com o objetivo de identificar as características de antenas *patches* relacionados à frequência de 2,4 GHz.

Na figura 3.8, temos os dados gerados para cada faixa de permissividade, onde podemos observar que em vermelho, temos todos os dados gerados para a faixa de permissividade menor que 2,33; em azul, os dados com permissividades menores que 2,67 e em cinza, os dados com permissividade menor que 3,0. Podemos observar que para cada faixa de frequência que representa uma classe temos um conjunto de dados que foi gerado para esses intervalos de valores de permissividades, para a largura, o comprimento, a diretividade e o fator de qualidade. Das 1000 amostras geradas com o rótulo “1” (2 – 3 GHz), foram obtidas aproximadamente 98 amostras na faixa de 2,34 a 2,46 GHz.



**Figura 3.8.** Dados gerados de características de antenas em microfita com 1000 amostras para os parâmetros (a)  $W$ , (b)  $L$ , (c)  $D$  e (d)  $Q_t$  com permissividade entre 2,0 a 3,0 e faixa de frequências de 2 a 3GHz.

Para cada uma das comparações representadas nas figuras 3.7 e 3.8, foi obtido o conjunto de erros para estimar a precisão dos modelos. Esses erros são apresentados na tabela 3.3.

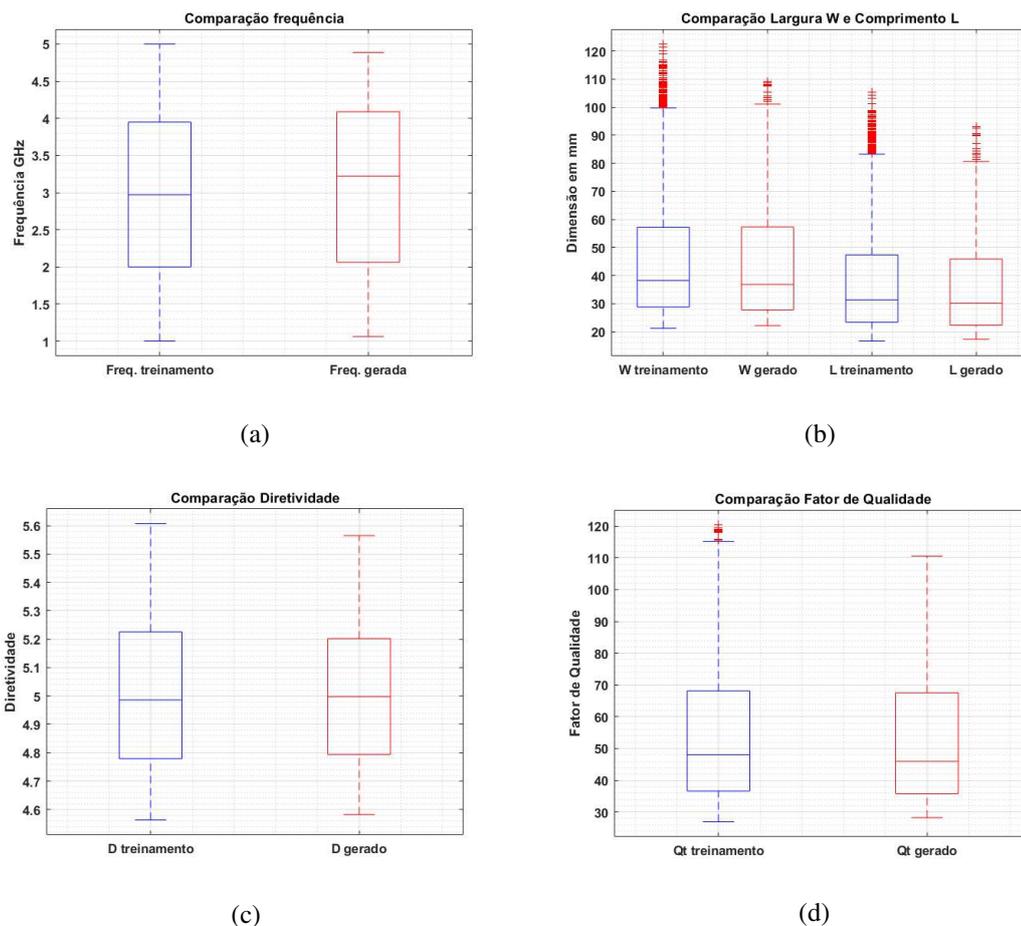
**Tabela 3.3.** Erros da CGAN

Erro	$W$	$L$	$D$	$Q_t$
MAE	$3,41 \times 10^{-4}$	$5,06 \times 10^{-4}$	$9,42 \times 10^{-3}$	$2,5894 \times 10^{-1}$
MRE	$6,997 \times 10^{-3}$	$1,345 \times 10^{-2}$	1,871	$4,79 \times 10^{-3}$
MSE	$2,16142 \times 10^{-7}$	$4,725 \times 10^{-7}$	$1,257 \times 10^{-4}$	$1,110 \times 10^{-3}$
RMSE	$2,6842 \times 10^{-5}$	$3,9686 \times 10^{-5}$	$6,475 \times 10^{-4}$	$1,923 \times 10^{-2}$

### 3.5.1.2 Comparação entre os dados reais e gerados da CGAN usando *box-plots*

A figura 3.9 apresenta os *box-plots* onde são comparados os dados de treinamento (em azul) e os dados gerados pela rede (em vermelho). A figura 3.9(a) apresenta um comparativo entre as frequências usadas para treinamento e as geradas pela rede, onde pode ser claramente observado que a rede aprendeu a gerar frequências com valores próximos às da mesma faixa que os dados de treinamento.

No caso dos parâmetros  $W$  e  $L$ , apresentados na figura e 3.9(b), observa-se uma boa similaridade entre os dados, o que indica que a rede teve uma boa distribuição entre os dados. Já nas figuras 3.9 (c) e (d) que representam a diretividade e o fator de qualidade, respectivamente, observamos uma grande similaridade entre os dados de treinamento e os dados gerados, o que significa que a rede também gerou dados mantendo a distribuição dos dados.



**Figura 3.9.** Medida de avaliação de distribuição dos dados. (a) Frequência real e gerada. (b) Largura e comprimento reais e gerados. (c) Diretividade real e gerados. (d) Fator de qualidade reais e gerados.

### 3.6 Conclusão do Capítulo

Neste primeiro estudo de caso foi proposta uma análise de três diferentes modelos de RNAs para modelagem de antenas em microfita com *patch* retangular. O primeiro modelo proposto foi uma GAN, que apresentou resultados promissores na geração de novos *designs* de antenas *patch* em microfita, similares aos utilizados durante o treinamento. A capacidade do modelo em aprender a relação entre as características das antenas e suas geometrias permitiu a geração de amostras sintéticas realistas que atendem aos requisitos de desempenho estabelecidos. Esses resultados reforçam a viabilidade e eficácia das redes GAN como ferramentas poderosas no desenvolvimento rápido e eficiente de projetos de antenas. O segundo modelo proposto foi uma MLP, que, como esperado, demonstrou sua capacidade em capturar eficientemente as características de saída a partir dos dados de entrada. Esses resultados corroboram a versatilidade e eficácia das MLPs na modelagem e na geração de soluções para problemas complexos, como o projeto de antenas *patch*. E o terceiro modelo escolhido foi uma CGAN. Ao condicionar a geração de dados sintéticos de antenas em microfita para faixas de frequência específicas, a CGAN se destacou ao produzir excelentes resultados. A capacidade do modelo em gerar antenas em microfita para faixa de frequência condicionada, ou seja, para cada rótulo atribuído, é um avanço significativo. Isso evidencia a eficácia da abordagem condicional da CGAN na geração de soluções altamente adaptáveis e personalizadas, capazes de atender às exigências de diferentes faixas de frequência. Esses resultados promissores reforçam a importância e o potencial das técnicas avançadas de aprendizado de máquina, como as CGANs, na otimização e no desenvolvimento de antenas em microfita para uma variedade de aplicações definidas por frequências de operação e demais parâmetros de antenas.

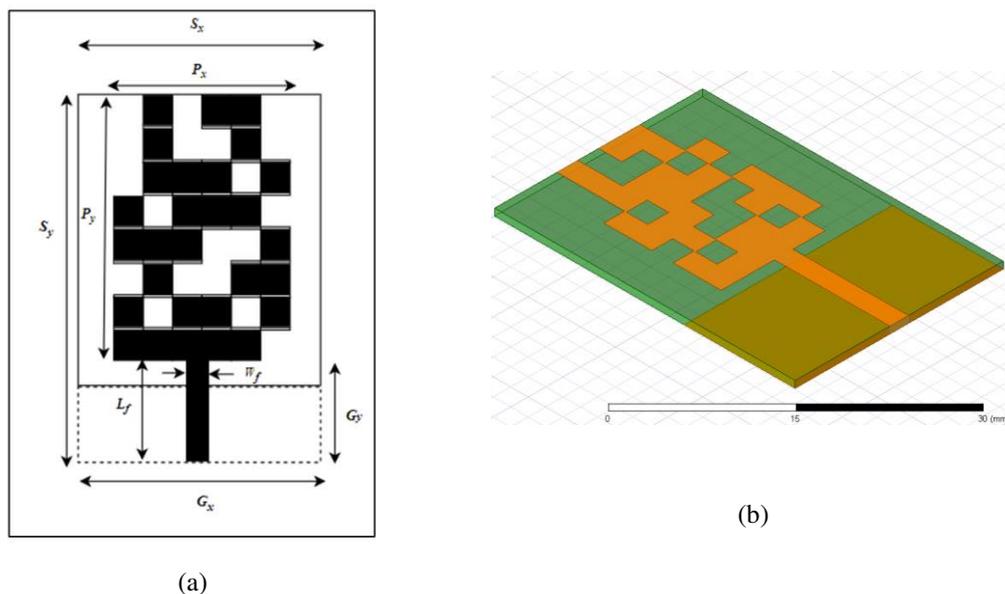
## CAPÍTULO 4

### Segundo Estudo de Caso

Neste segundo estudo de caso avaliamos o uso de uma CNN para o modelamento de alguns parâmetros fundamentais de uma antena em microfita pixelada e de banda dupla. A CNN foi treinada para prever suas frequências de ressonância e correspondentes diretividades e ganhos considerando três valores distintos de permissividade relativa para o substrato. O conjunto de dados para treinamento foi gerado a partir do software de simulação HFSS (*High-Frequency Structure Simulator*).

#### 4.1 Modelagem de Antena em Microfita Pixelada usando CNN

Neste estudo de caso, o projeto de antena com banda dupla baseia-se na antena cuja geometria inicial ou de referência está representada na figura 4.1. Ao se combinar dois, três ou quatro pixels metálicos nessa geometria de referência é possível obtermos configurações ressonantes em duas bandas de frequência.



**Figura 4.1.** (a) Vista superior da geometria base de uma antena em microfita pixelada. Em preto, temos a região metalizada (alguns pixels e a linha de alimentação). O plano de terra é representado pelo contorno tracejado na face oposta. (b) Imagem em perspectiva da antena pixelada desenhada no software Ansys HFSS.

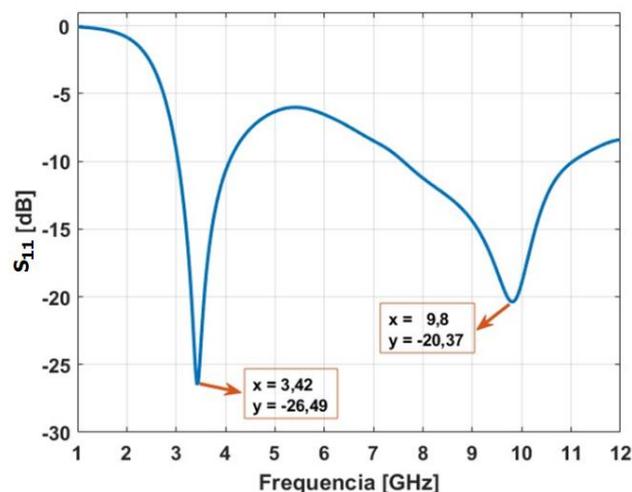
A geometria base apresentada na figura 4.1(a) tem as seguintes dimensões: o substrato S tem largura  $S_x = 23,85\text{mm}$  e comprimento  $S_y = 34,45\text{mm}$ , com constante dielétrica  $\epsilon_r = 2,26$  e altura  $h = 0,795\text{mm}$ . Já a região pixelada apresenta as seguintes configurações geométricas, largura  $P_x = 14,31\text{mm}$ , o comprimento  $P_y = 21,2\text{mm}$  e as dimensões de cada pixel com valores

de largura igual a  $\Delta_x = 2,385$  mm e comprimento  $\Delta_y = 2,65$  mm. Foram adicionados mais 0,2 mm de metalização à extremidade superior dos pixels delimitados acima dos pixels vazios, a fim de evitar conexões de ponto único (conforme figura 4.1). A antena foi alimentada por uma linha de alimentação em microfita no mesmo lado do substrato com comprimento  $L = 13,25$  mm e largura  $W = 2,12$  mm. O plano terra da microfita tem dimensões  $G_x = 23,85$  mm e  $G_y = 12,72$  mm. Na figura 4.1(b), temos a imagem da antena pixelada representada no software comercial *Ansys HFSS*. Na tabela 4.1 são apresentados alguns parâmetros dessa antena obtidos a partir da simulação no HFSS.

**Tabela 4.1.** Principais características da antena de referência.

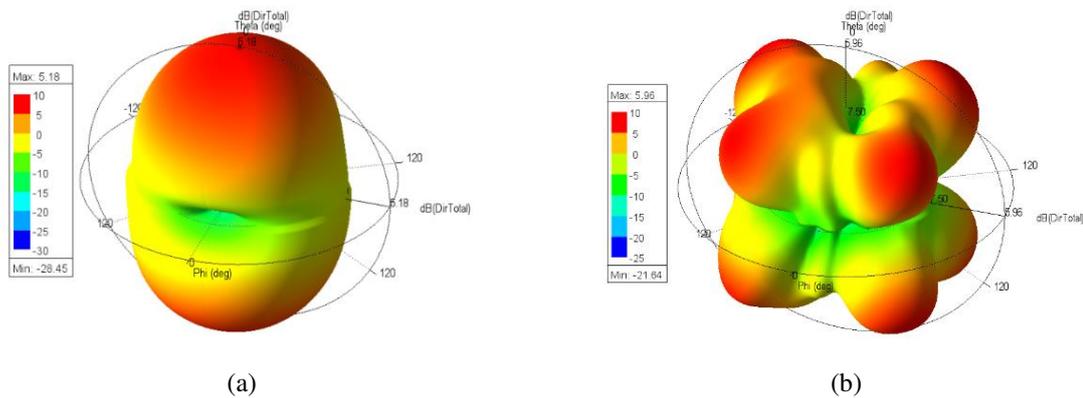
Característica	Símbolo	Valor
Primeira frequência de ressonância	$f_1$	3,42 GHz
Segunda frequência de ressonância	$f_2$	9,8 GHz
Diretividade na primeira frequência	$D_1$	5,18 dB
Diretividade na segunda frequência	$D_2$	5,96 dB
Ganho na primeira frequência	$G_1$	5,19 dB
Ganho na segunda frequência	$G_2$	6,07 dB

A curva do parâmetro  $S_{11}$  dessa antena é apresentada na figura 4.2, onde é possível identificar as frequências de ressonância. Essas frequências são os pontos onde a antena tem uma resposta ideal, mostrando a eficiência máxima na transmissão e recepção de sinais.

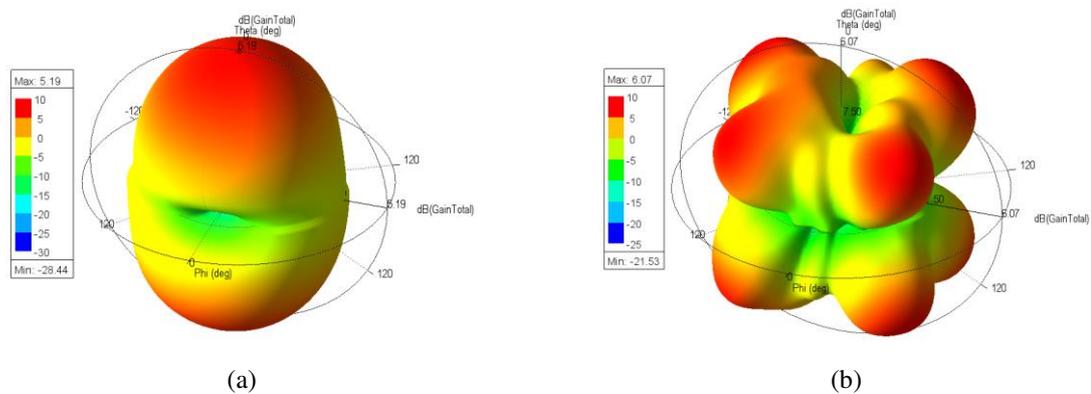


**Figura 4.2.** Parâmetro  $S_{11}$  para as duas frequências de ressonância.

Nas figuras 4.3 e 4.4, são ilustradas as diretividade e o ganhos da antena para as respectivas frequências de ressonância. A diretividade indica a capacidade da antena de concentrar a energia radiada em uma direção específica, enquanto o ganho relaciona essa diretividade com a eficiência da antena. Esses padrões são cruciais para compreender como a antena se comporta em termos de orientação e intensidade do sinal nas frequências de ressonância identificadas na curva  $S_{11}$ .



**Figura 4.3.** (a) Diretividade em  $f_{res1}$ . (b) Diretividade em  $-f_{res2}$ .



**Figura 4.4.** (a) Ganho em  $f_{res1}$ . (b) Ganho de  $f_{res2}$ .

## 4.2 Definição do Problema

A determinação da frequência de ressonância é importante para o projeto de antenas em microfita, independentemente do número e bandas. Boa parte dos estudos reportados na literatura focaram principalmente na modelagem de antenas em microfita via RNA para operarem em uma única banda de frequência. Recentemente, alguns estudos consideraram a modelagem de dessas antenas considerando operação em banda dupla. Diferentemente das antenas de patch retangular vistas nos capítulos anteriores, as antenas pixeladas apresentam um

desafio diferente de modelagem, como exemplo, não apresentam fórmulas CAD que viabilizariam o cálculo rápido e aproximado de parâmetros de antenas (como é o caso de antenas com *patches* retangulares e circulares). As instâncias geométricas de antenas pixeladas são diferenciadas por suas configurações de pixels. Essas configurações podem, por exemplo, serem obtidas pela adição e/ou remoção de pixels em relação a alguma configuração inicial, produzindo formatos de antena que podem parecer irregulares.

O projeto de antenas pixeladas geralmente envolve o uso de técnicas de busca e otimização, por exemplo, o algoritmo genético. O que torna o processo custoso por exigir muitas simulações eletromagnéticas, que por sua vez, requerem elevado custo computacional. No entanto, com o uso de técnicas de aprendizado de máquina pode ser possível reduzir o número dessas simulações a partir do emprego de modelos de RNAs; o que pode reduzir consideravelmente o tempo de projeto de uma dada antena.

Neste estudo será proposta uma metodologia capaz de modelar as duas frequências ressonantes, suas respectivas diretividades e ganhos para uma antena em microfita pixelada de banda dupla para três diferentes tipos de permissividades. A antena pixelada pode ser modelada com boa precisão usando uma estratégia baseada em RNAs.

A CNN é uma arquitetura que sugere ser adequada para essa tarefa. Nos últimos anos, as CNNs têm alcançado resultados impressionantes em tarefas de classificação e regressão de imagens. A superfície de uma antena pixelizada, quando expressa como uma matriz 2D binária, é semelhante aos dados de uma imagem em preto e branco, o que torna as CNNs especialmente adequadas para essa tarefa.

### **4.3 Descrição da Estratégia via RNAs**

Para este estudo foi feito o uso de CNNs para predição das duas frequências ressonantes e seus respectivos ganhos e diretividades a partir da representação de toda superfície pixelada e de três diferentes valores de permissividades.

### **4.4 Geração da Base de Dados**

A base de dados foi criada a partir da antena de referência (figura 4.1) e fazendo agrupamentos de 2, 3 ou 4 pixels metalizados nos espaços vazios da região pixelada. A quantidade de combinações possíveis depende da quantidade de espaços vazios disponíveis na antena base. A quantidade de combinações possíveis é descrita por:

$$C_{n,p} = \frac{n!}{p!(n-p)!} \quad 4.1$$

Onde  $n$  é o número de pixels vazios da antena e  $p$  é o número de pixels que serão preenchidos. Os números totais de combinações possíveis para dois, três ou quatro pixels são: 171, 969 e 3876, respectivamente. Para expandir o número de combinações possíveis de antenas, foram considerados três valores diferentes de permissividades relativas para o substrato da antena. Os valores das permissividades relativas correspondem aos seguintes substratos: Duroid ( $\epsilon_r = 2,2$ ), PDMS ( $\epsilon_r = 2,65$ ) e RO3003 ( $\epsilon_r = 3,00$ ).

A base de dados é composta pelas 171 combinações de antenas com dois pixels adicionais, 969 combinações com três pixels adicionais e 800 combinações das 3876 possíveis de quatro pixels adicionais. Como estamos trabalhando com três permissividades distintas, o valor total seria três vezes esse conjunto de dados para cada conjunto de antenas de dois, três e quatro pixels. Desta forma, o número total de antenas usadas como base de dados serão 5820. As 5820 diferentes antenas foram simuladas usando o software Ansys HFSS, que utiliza o FEM, para obtenção dos dados de frequências de ressonância, diretividades e ganhos. Desta forma, será completada a base de dados para o treino da rede.

### *Separação dos Dados*

A base de dados foi dividida em três conjuntos de dados: sendo o primeiro conjunto os dados de treinamento, usado para treinar a rede. O segundo conjunto chamado de conjunto de validação, será usado pela rede para avaliar seu desempenho enquanto é treinada com os dados de treino, em seguida a rede ajustará seus pesos usando os dados de validação. O terceiro conjunto é o conjunto de teste, usado para avaliar o desempenho da rede com dados que não participaram no processo de treinamento. A tabela 4.2 apresenta a divisão dos conjuntos de dados.

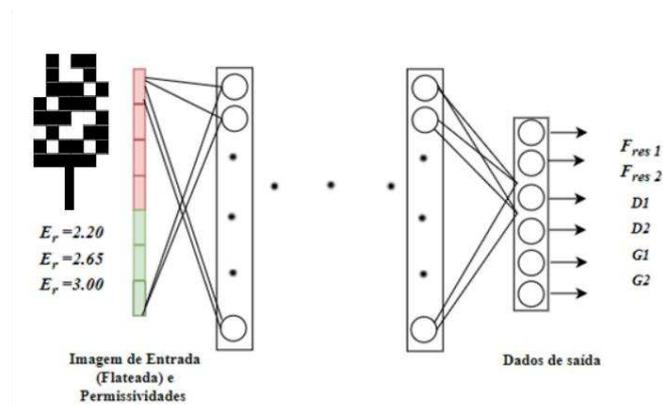
**Tabela 4.2.** Divisão dos conjuntos de treino, validação e teste.

Permissividade Relativa	Geometria	Dados de Treino	Dados de Validação	Dados de Teste
PDMS (2,20)	2-Pixels	34	34	103
	3-Pixels	194	194	581
	4-Pixels	350	150	300
Duroid (2,65)	2-Pixels	34	34	103
	3-Pixels	194	194	581
	4-Pixels	350	150	300
Rogers (3,00)	2-Pixels	34	34	103
	3-Pixels	194	194	581
	4-Pixels	350	150	300
Total		1734	1134	2952

A informação da geometria da antena é fornecida à rede através de uma matriz que representa cada pixel metalizado pelo valor “1” e, no caso, contrário pelo valor “0”. Para incorporar a informação da permissividade, cada uma das matrizes é multiplicada pelo valor da respectiva permissividade.

#### 4.5 Treinamento da CNN

A estrutura da CNN utilizada nos experimentos é apresentada na figura 4.5. Como entrada, a rede recebe informações de dois conjuntos de variáveis, a matriz 8x6 que corresponde à parte pixelada das geometrias das antenas com uns/zeros usados para indicar a presença/ausência de um pixel metalizado, respectivamente, e os dados de permissividade dos substratos que podem assumir os seguintes valores:  $\epsilon_r = 2,2$ ;  $\epsilon_r = 2,65$  e  $\epsilon_r = 3,00$ .



**Figura 4.5.** Representação esquemática da CNN.

Em seguida, são configuradas 5 camadas de convolução Conv2D, todas as camadas de convolução possuem dimensão  $3 \times 3$  (*kernel size*) e utiliza *stride*=1 com *padding*=*same*. A configuração de *padding*=*same* faz com que as dimensões dos *feature-maps* de saída sejam iguais às dimensões dos *feature-maps* de entrada. Para esta configuração foram adicionadas duas camadas de *Pooling* que mantiveram as dimensões 2-D das camadas/subcamadas. O número de nós em cada camada convolucional foi definida pela seguinte equação  $16/\sqrt{M}$ , arredondando para o número inteiro mais próximo, sendo  $M = 3$  e a quantidade de nós em cada uma das camadas igual a 9. O algoritmo de otimização escolhido para o treinamento da rede foi o *Adam* [27]. Os parâmetros utilizados no algoritmo de otimização Adam foram: taxa de aprendizado  $\alpha = 0,007$ ,  $\beta_1 = 0,9$ ,  $\beta_2 = 0,999$  e taxa de decaimento  $\epsilon = 10^{-8}$ .

Na camada de saída, a quantidade de nós é igual 6, os quais representam os dados que a rede foi treinada para prever, ou seja, as duas frequências de ressonância seus respectivos ganhos e diretividades. A rede foi configurada para ser treinada por 2.000 épocas.

## 4.6 Resultados da CNN

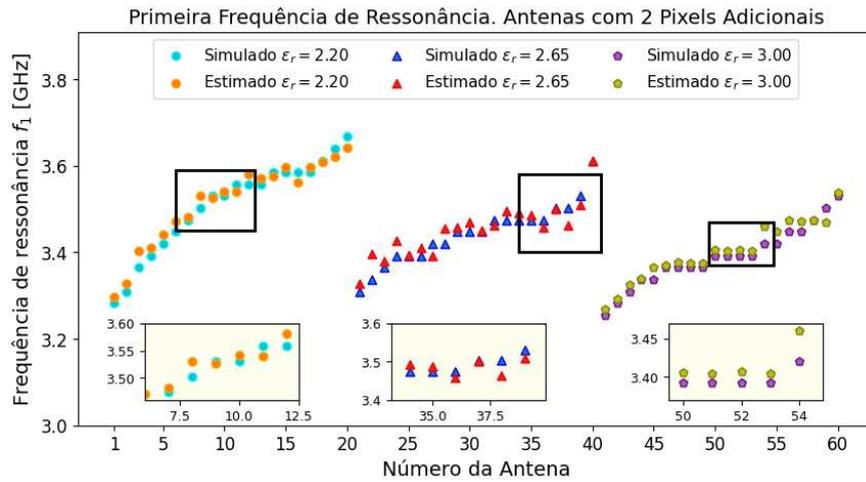
Nesta seção serão apresentados os resultados de predição da CNN para dois experimentos distintos.

### *Primeiro Experimento*

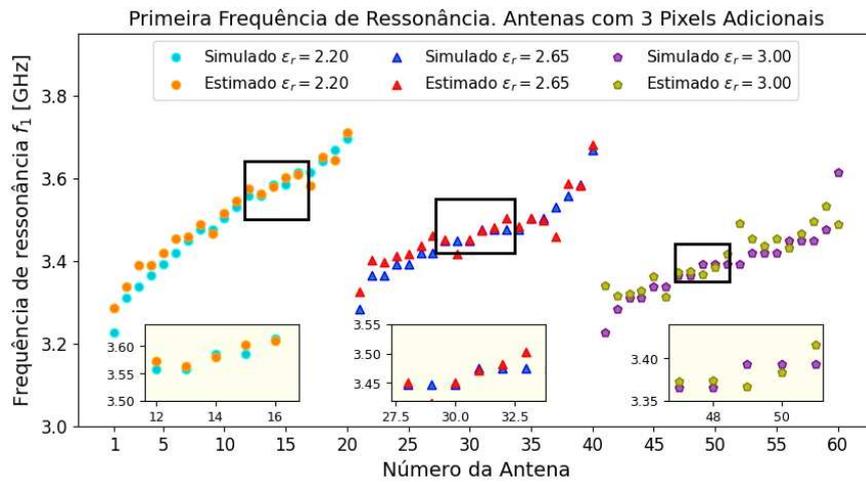
No primeiro experimento foi modelada uma CNN para prever as frequências de ressonâncias para três substratos diferentes assumindo as seguintes permissividades:  $\epsilon_r = 2,2$ ,  $\epsilon_r = 2,65$  e  $\epsilon_r = 3,00$ . As figuras 4.6 até 5.10 apresentam comparativos entre os dados reais (obtidos via simulação no Ansys HFSS) e os dados estimados pela CNN considerando as antenas com dois, três e quatro pixels, para as duas frequências ressonantes (primeira e

segunda). Essas figuras apresentam as antenas simuladas e as estimadas em função da frequência de ressonância para cada uma das três permissividades. As antenas simuladas estão alinhadas verticalmente com a sua correspondente antena predita pela CNN.

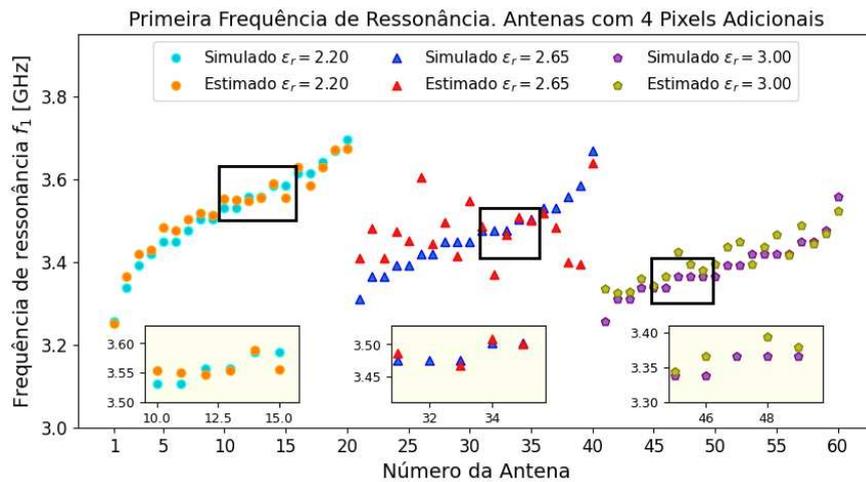
Nas figuras 4.6 e 4.7, são apresentados os resultados para a primeira e segunda frequência de ressonância, respectivamente. E, para cada uma dessas figuras são considerados os casos com agrupamentos de 2, 3, e 4 pixels adicionais na antena de referência (figura 4.1). No caso da primeira frequência de ressonância, os valores estimados estão bastante próximos dos valores reais. No caso da segunda frequência de ressonância, os valores preditos apresentam uma menor aproximação aos valores reais, entretanto, a maior taxa de erro para esses dados foi de 1,10%.



(a)

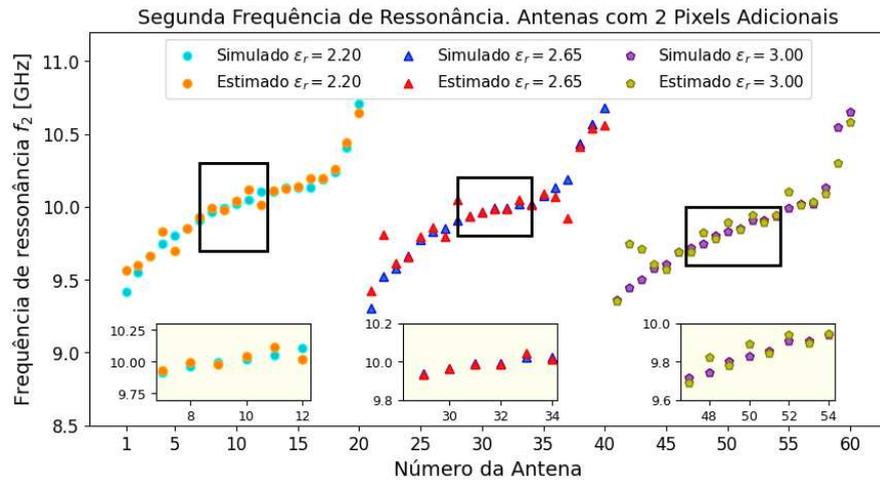


(b)

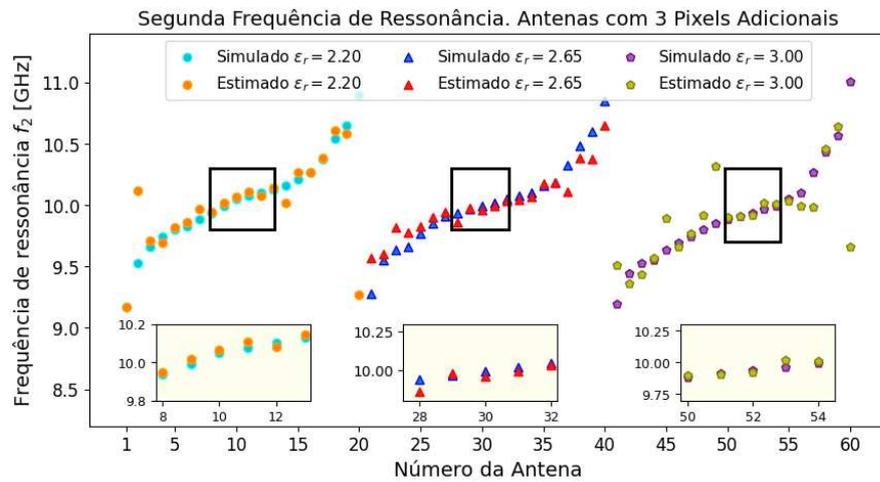


(c)

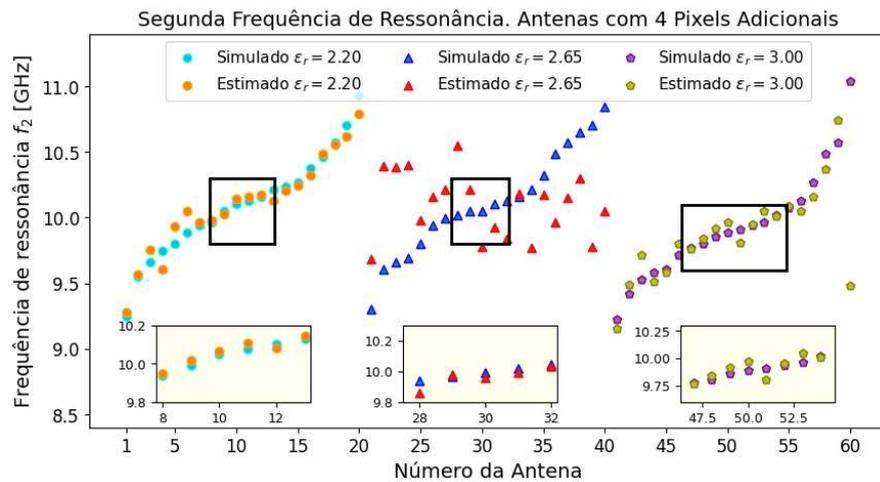
**Figura 4.6.** Primeira frequência de ressonância. (a) Antenas com agrupamentos de 2 pixels adicionais. (b) Antenas com agrupamentos de 3 pixels adicionais. (c) Antenas com agrupamentos de 4 pixels adicionais.



(a)



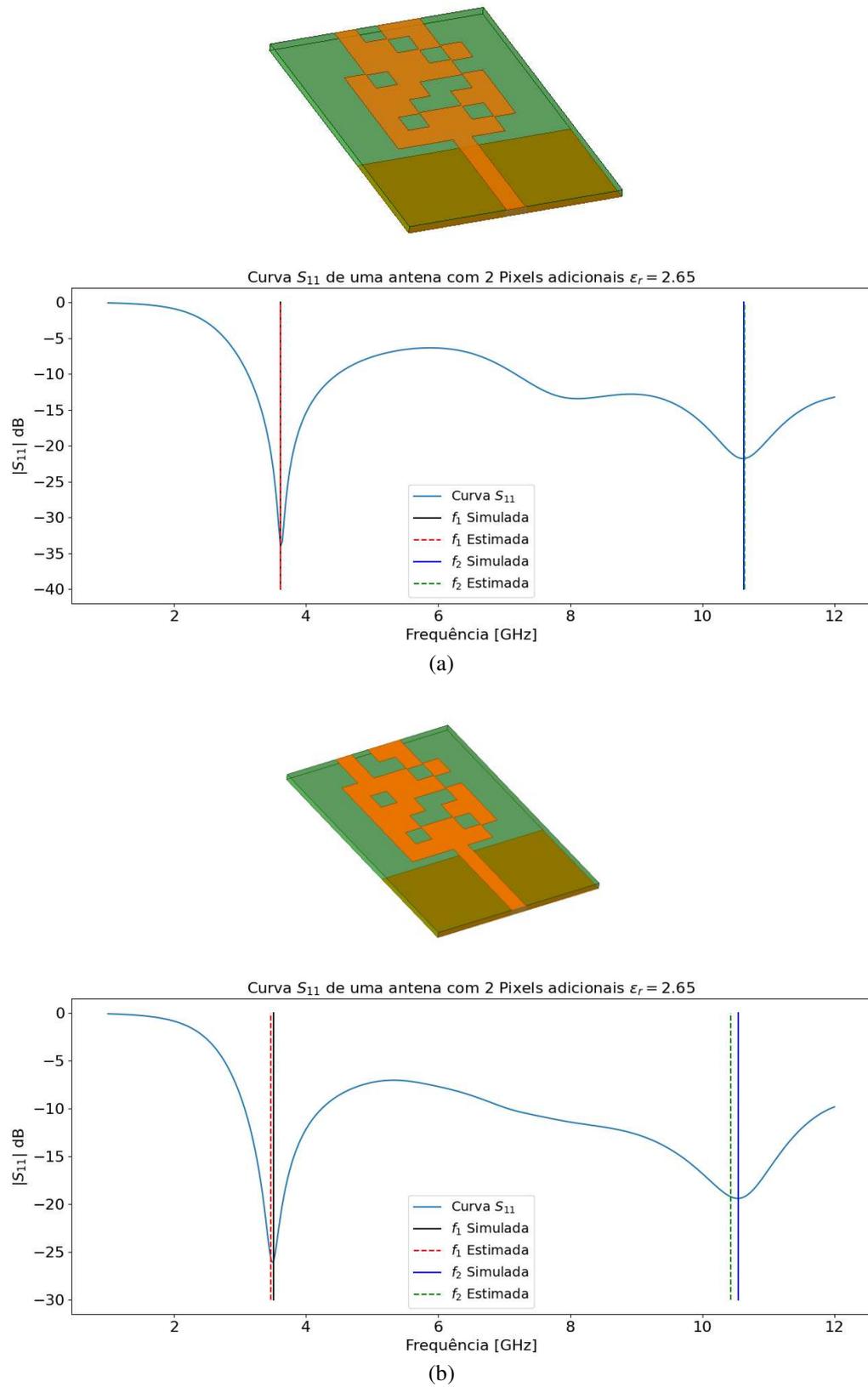
(b)



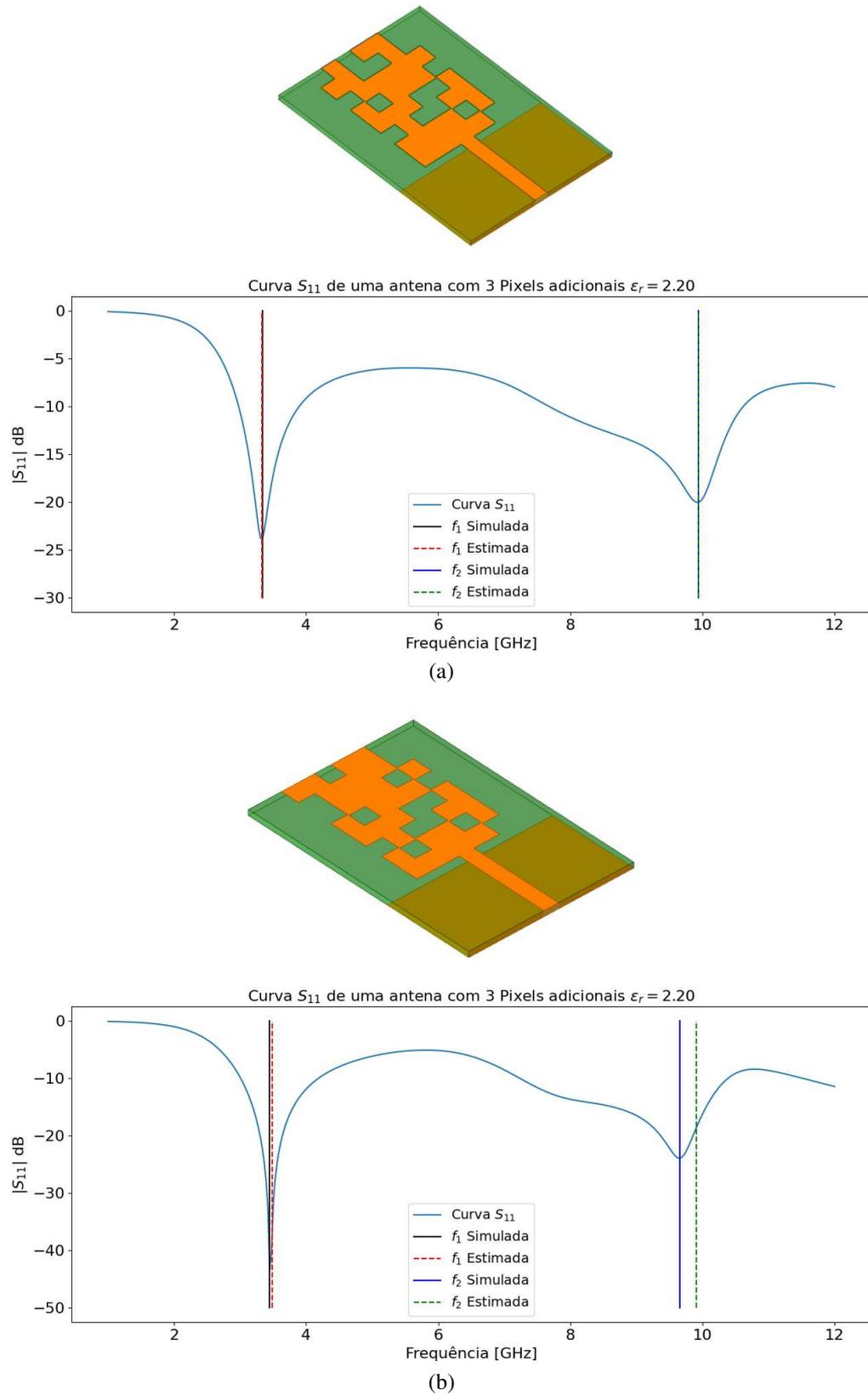
(c)

**Figura 4.7.** Segunda frequência de ressonância. (a) Antenas com agrupamentos de 2 pixels adicionais. (b) Antenas com agrupamentos de 3 pixels adicionais. (c) Antenas com agrupamentos de 4 pixels adicionais.

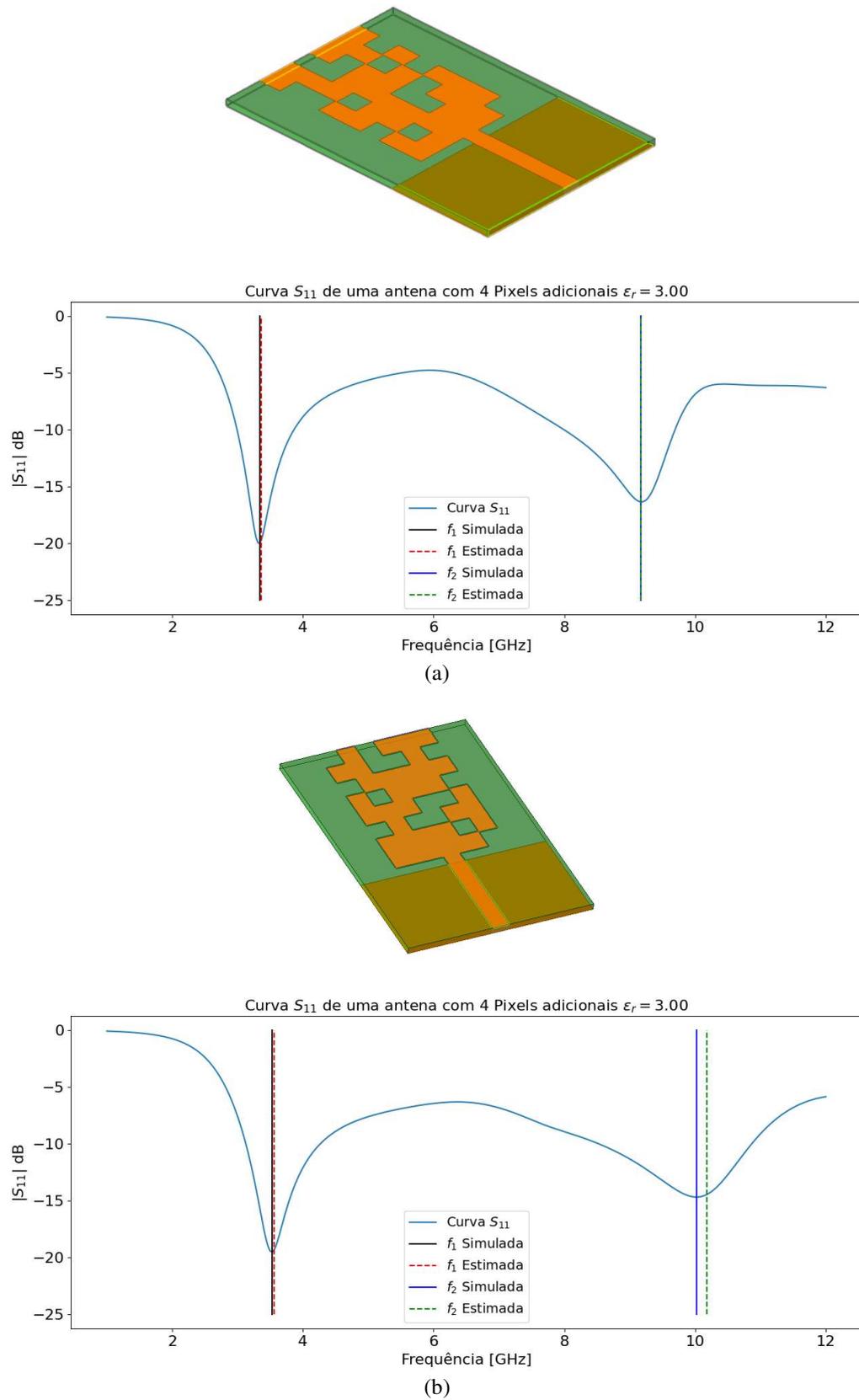
Na figura 4.8 são apresentados gráficos contendo o parâmetro  $S_{11}$  de antenas com  $\epsilon_r = 2,65$  e com agrupamentos de 2 pixels adicionais na antena de referência. Para a configuração de antena mencionada temos nas figuras 4.8(a) e (b) os casos que apresentaram a menor e maior taxa de erro, respectivamente. Na figura 4.9 são apresentados gráficos contendo o parâmetro  $S_{11}$  de antenas com  $\epsilon_r = 2,20$  e com agrupamentos de 3 pixels adicionais na antena de referência. Para a configuração de antena mencionada temos nas figuras 4.9(a) e (b) os casos que apresentaram a menor e maior taxa de erro, respectivamente. Na figura 4.10 são apresentados gráficos contendo o parâmetro  $S_{11}$  de antenas com  $\epsilon_r = 3,00$  e com agrupamentos de 4 pixels adicionais na antena de referência. Para a configuração de antena mencionada temos nas figuras 4.10(a) e (b) os casos que apresentaram a menor e maior taxa de erro, respectivamente.



**Figura 4.8.** Geometrias e respectivas curvas  $S_{11}$  de antenas com agrupamentos de 2 pixels adicionais na antena de referência considerando  $\epsilon_r = 2,65$ . (a) para o menor erro. (b) para o maior erro.



**Figura 4.9.** Geometrias e respectivas curvas  $S_{11}$  de antenas com agrupamentos de 3 pixels adicionais na antena de referência considerando  $\epsilon_r = 2,20$ . (a) para o menor erro. (b) para o maior erro.



**Figura 4.10.** Geometrias e respectivas curvas  $S_{11}$  de antenas com agrupamentos de 4 pixels adicionais na antena de referência considerando  $\epsilon_r = 3,0$ . (a) para o menor erro. (b) para o maior erro.

A partir dos resultados apresentados, observamos que a CNN apresentou um bom desempenho para prever ambas as duas frequências de ressonância. No entanto, observamos que, para todos os casos, a CNN teve um melhor resultado para predição da primeira frequência de ressonância.

Para fins comparativos da avaliação desses dados, o cálculo do erro relativo médio (MRE – *Mean Relative Error*) – equação 4.2 – foi feito separadamente para previsão dos dados de treinamento e dos dados de teste e individualmente para os dados de frequência de ressonância.

$$\text{MRE} = \frac{1}{N} \sum_{m=1}^N \frac{|f_{res, target}^{(m)} - f_{res, previsto}^{(m)}|}{f_{res, target}^{(m)}} \quad 4.2$$

A Tabela 4.3 apresenta os resultados preditivos em termos de MRE expresso em porcentagem para os dados de treino, validação e teste da primeira frequência de ressonância ( $f_1$ ) e segunda frequência de ressonância ( $f_2$ ).

**Tabela 4.3.** Erro percentual relativo médio nos dados de treino, validação e teste.

	$f_1$	$f_2$
Treino	0,458%	0,612%
Validação	0,620%	1,13%
Teste	0,589%	1,10%

Os valores das frequências de ressonância simulados e estimados, bem como seus respectivos erros são apresentados na tabela 4.4.

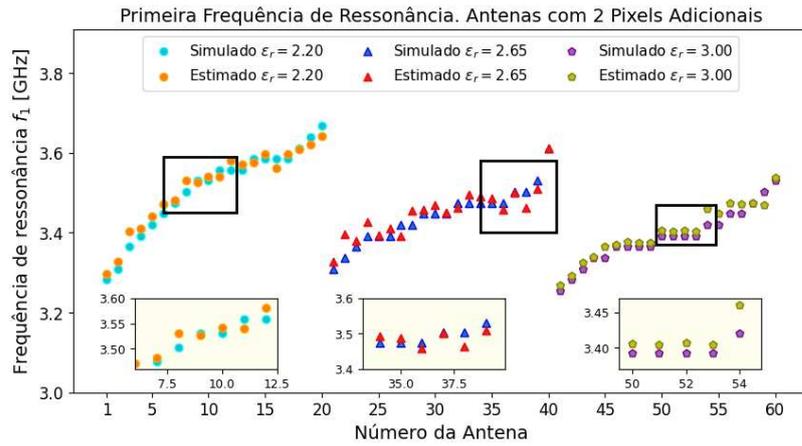
**Tabela 4.4.** Valores de frequência de ressonância para o primeiro experimento.

Num. de Pixels	Figura	$f_1$			$f_2$		
		Simulado	Estimado	Erro%	Simulado	Estimado	Erro%
2 Pixels	fig. 4.8(a)	3,6125	3,6059	0,180	10,1300	10,1200	0,099
	fig. 4.8(b)	3,3100	3,3441	1,030	9,9925	9,9021	0,904
3 Pixels	fig. 4.9(a)	3,6950	3,6894	0,152	10,6250	10,6116	0,126
	fig. 4.9(b)	3,5575	3,5759	0,517	9,4425	9,3647	0,823
4 Pixels	fig. 4.10(a)	3,3650	3,3741	0,269	10,5150	10,5073	0,073
	fig. 4.10(b)	3,3375	3,3366	0,027	10,4875	10,3722	1,099

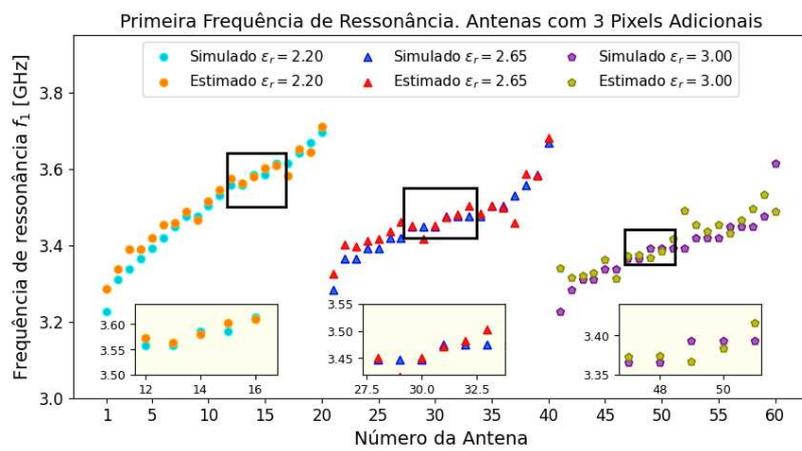
## *Segundo Experimento*

Neste segundo experimento, a CNN foi avaliada em um cenário ainda mais desafiador. A CNN terá que prever os dados de frequências de ressonância, diretividade e ganho para as antenas com adicionais de dois, três e quatro pixels para três substratos diferentes assumindo as seguintes permissividades:  $\epsilon_r = 2,2$ ,  $\epsilon_r = 2,65$  e  $\epsilon_r = 3,00$ . As figuras 4.11 até 4.19 apresentam comparativos entre os dados reais (obtidos via simulação no Ansys HFSS) e os dados estimados pela CNN considerando as antenas com dois, três e quatro pixels, para as duas frequências ressonantes (primeira e segunda). Essas figuras apresentam as antenas simuladas e as estimadas em função da frequência de ressonância para cada uma das três permissividades. As antenas simuladas estão alinhadas verticalmente com a sua correspondente antena prevista pela CNN.

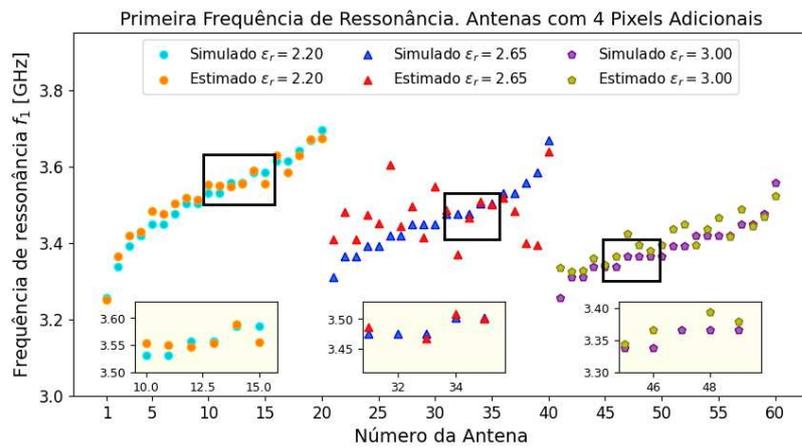
Nas figuras 4.11 e 4.12, são apresentados os resultados para a primeira e segunda frequência de ressonância, respectivamente. E, para cada uma dessas figuras são considerados os casos com agrupamentos de 2, 3, e 4 pixels adicionais na antena de referência (figura 4.1). No caso da primeira frequência de ressonância, os valores estimados estão bastante próximos dos valores reais. No caso da segunda frequência de ressonância, os valores previstos não estão tão próximos dos valores reais, no entanto apresentaram uma taxa de erro inferior a 1%.



(a)

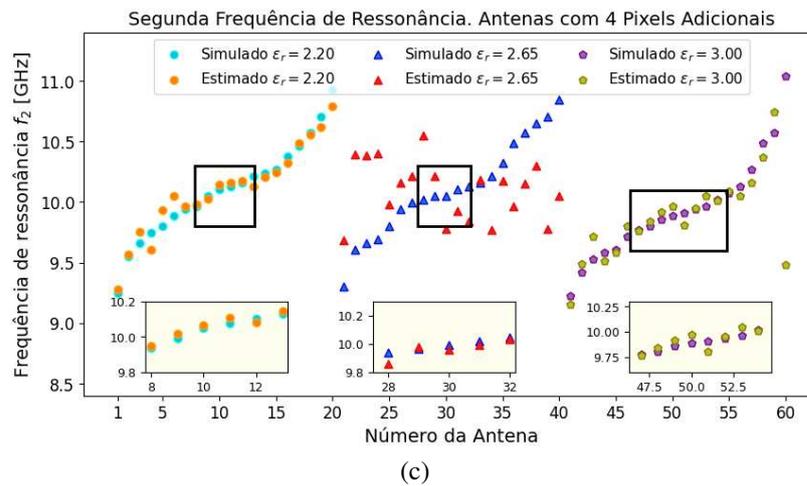
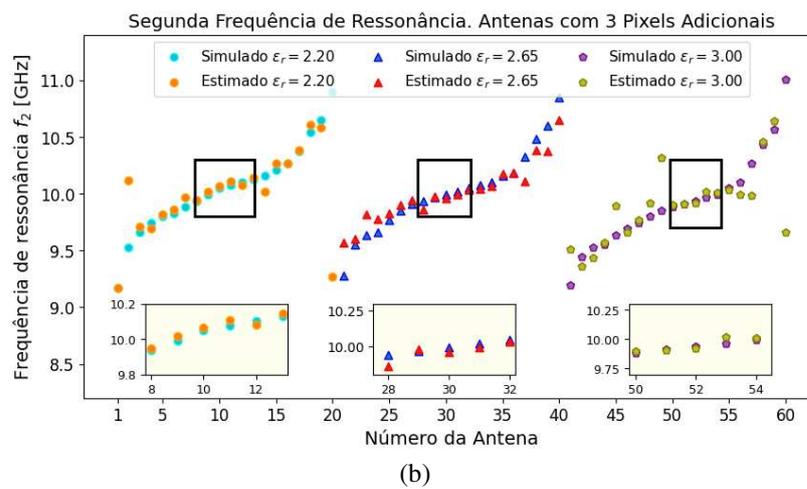
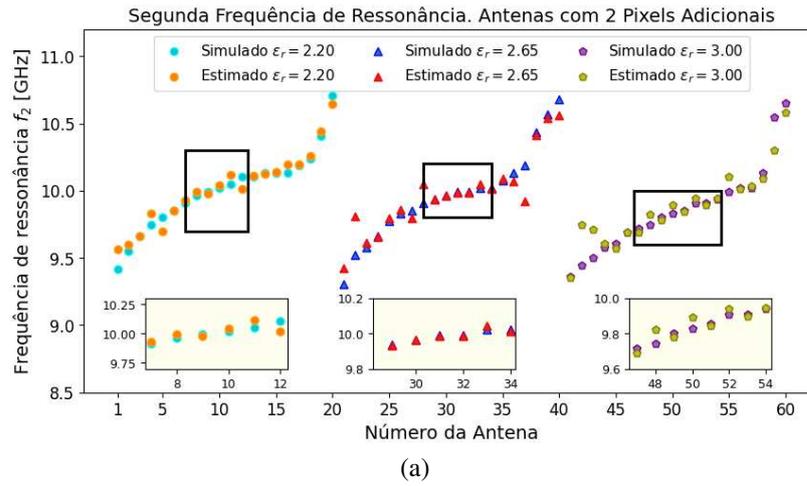


(b)



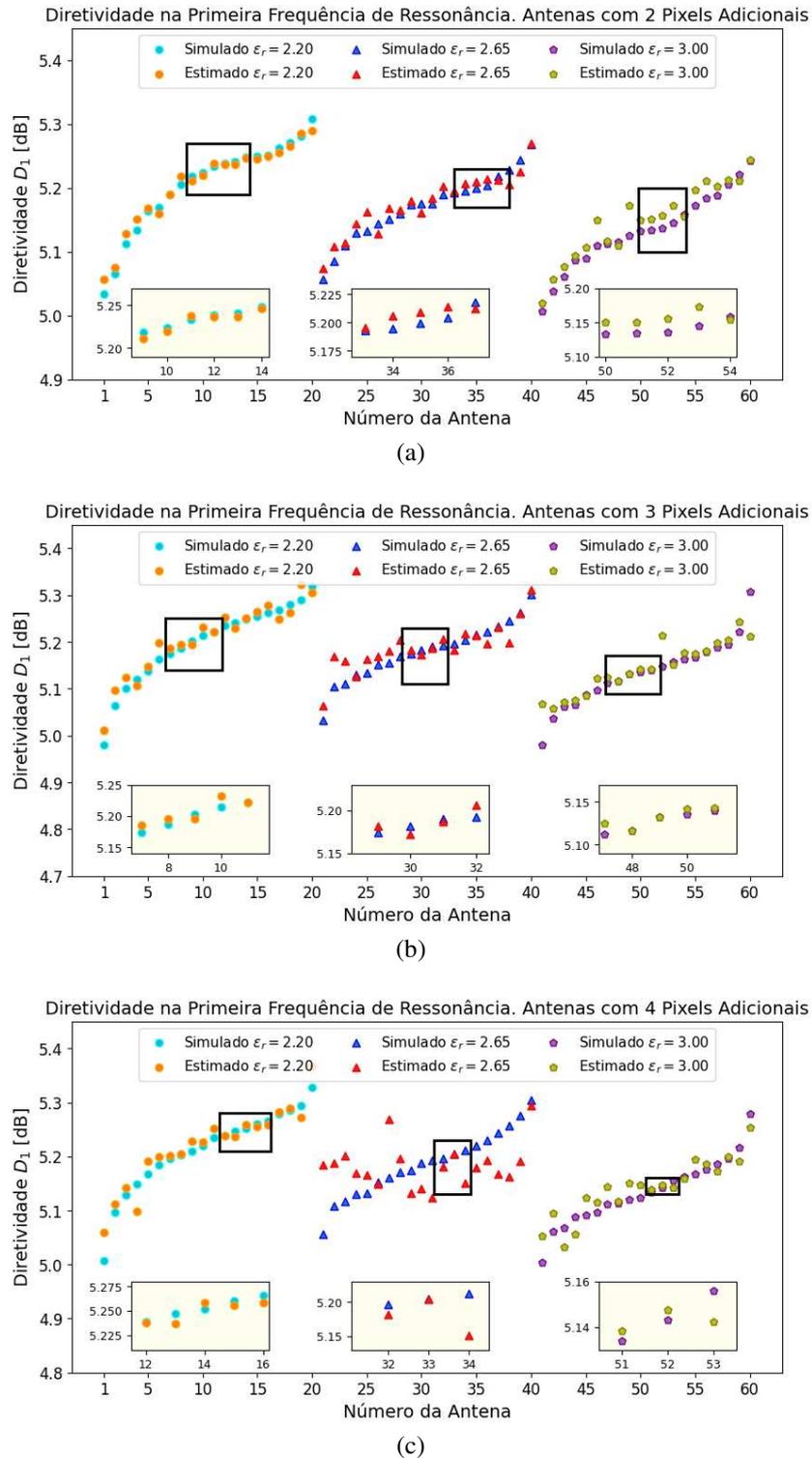
(c)

**Figura 4.11.** Primeira frequência de ressonância. (a) Antenas com agrupamentos de 2 pixels adicionais. (b) Antenas com agrupamentos de 3 pixels adicionais. (c) Antenas com agrupamentos de 4 pixels adicionais.

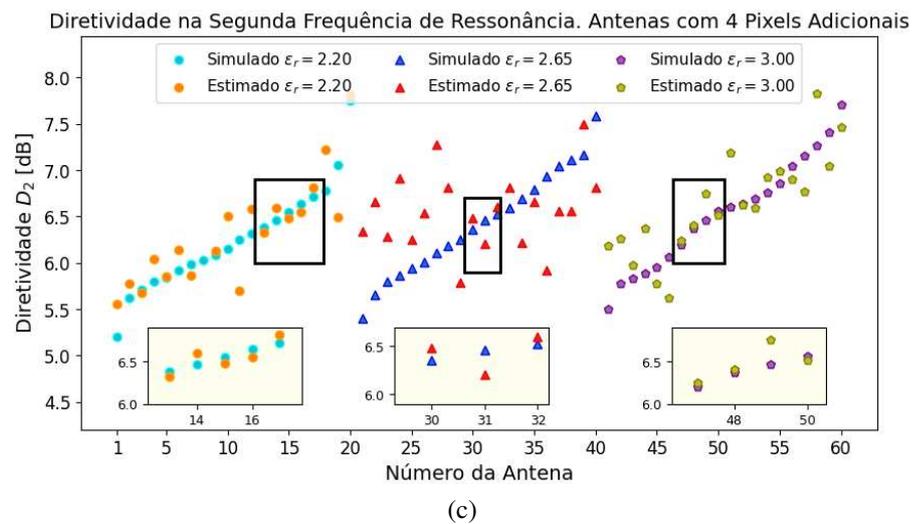
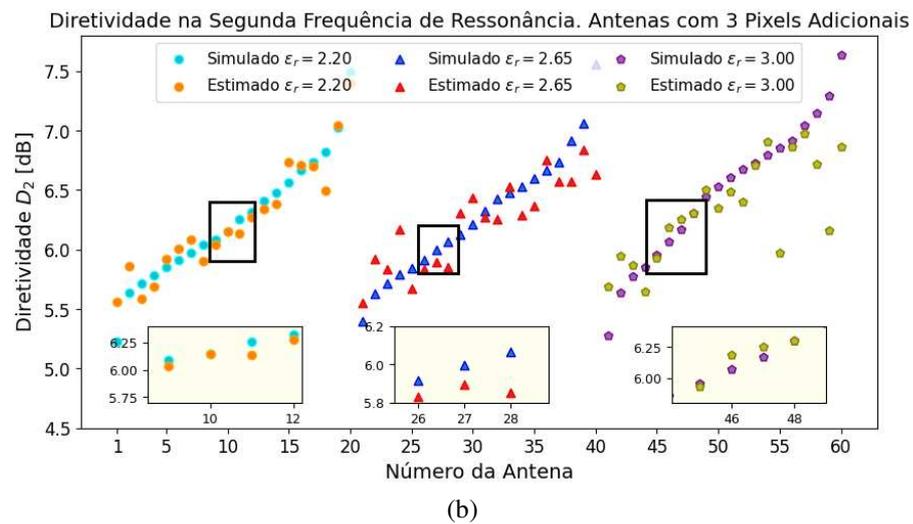
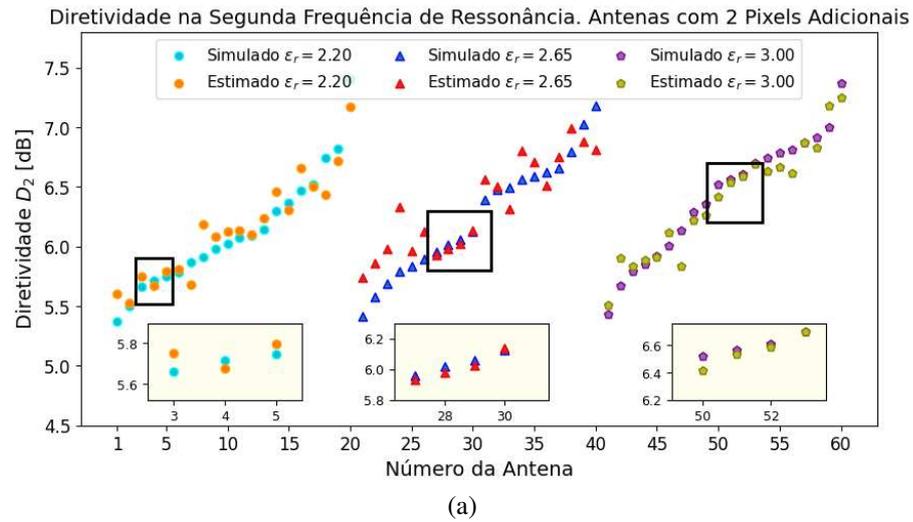


**Figura 4.12.** Segunda frequência de ressonância. (a) Antenas com agrupamentos de 2 pixels adicionais. (b) Antenas com agrupamentos de 3 pixels adicionais. (c) Antenas com agrupamentos de 4 pixels adicionais.

As figuras 4.13 e 4.14 apresentam uma comparação para os valores das diretividades da primeira e segunda frequência de ressonância, respectivamente. No geral, nota-se que a rede estima valores similares aos reais, mas pode-se observar discrepâncias para algumas antenas com  $\epsilon_r = 2,65$ ; cenário em que a rede sobrestima o valor da diretividade.

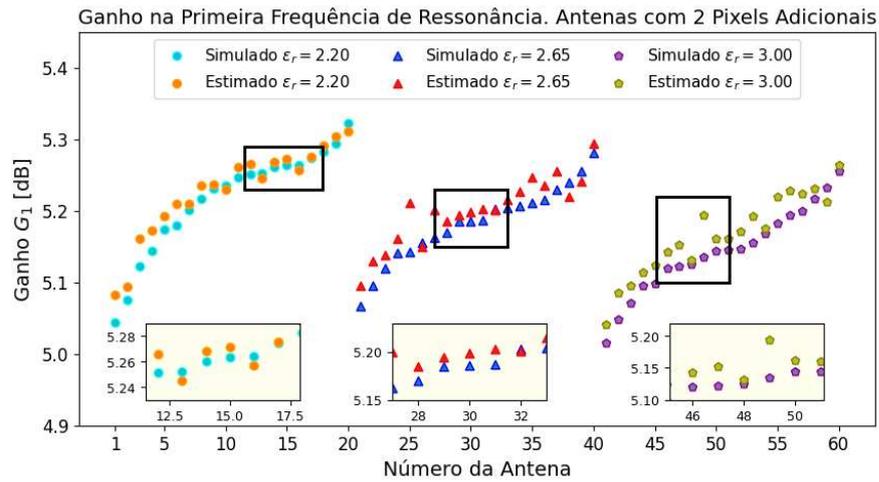


**Figura 4.13.** Diretividade na primeira frequência de ressonância. (a) Antenas com agrupamentos de 2 pixels adicionais. (b) Antenas com agrupamentos de 3 pixels adicionais. (c) Antenas com agrupamentos de 4 pixels adicionais.

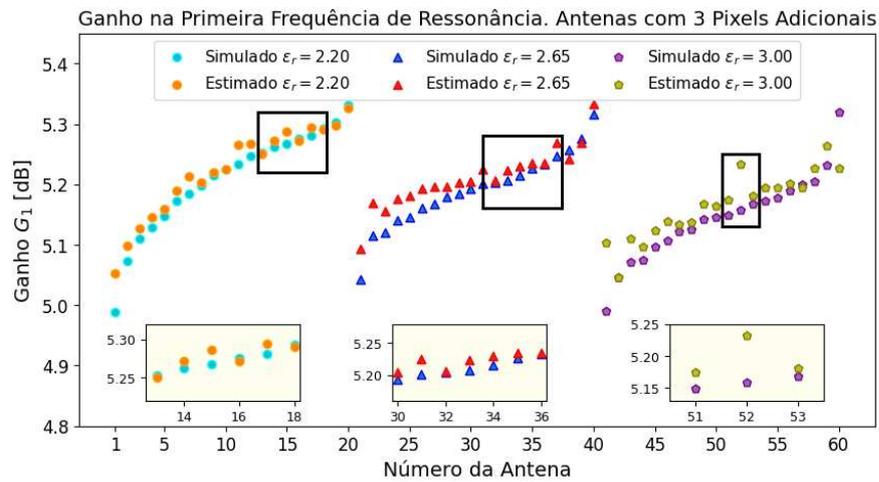


**Figura 4.14.** Diretividade na segunda frequência de ressonância. (a) Antenas com agrupamentos de 2 pixels adicionais. (b) Antenas com agrupamentos de 3 pixels adicionais. (c) Antenas com agrupamentos de 4 pixels adicionais.

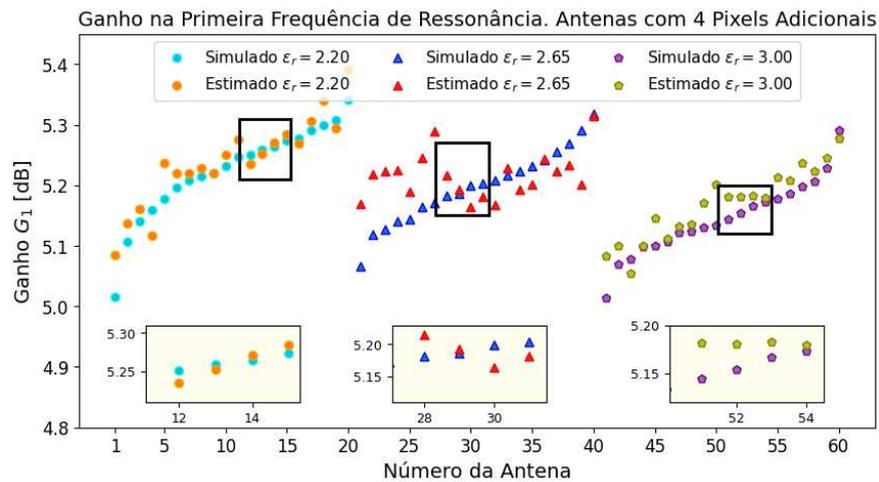
As figuras 4.15 e 4.16 apresentam uma comparação para os valores dos ganhos da primeira e segunda frequência de ressonância, respectivamente. O comportamento da rede para o caso do ganho é similar ao da diretividade, onde podemos observar maiores discrepâncias em algumas antenas com  $\epsilon_r = 2,65$ .



(a)

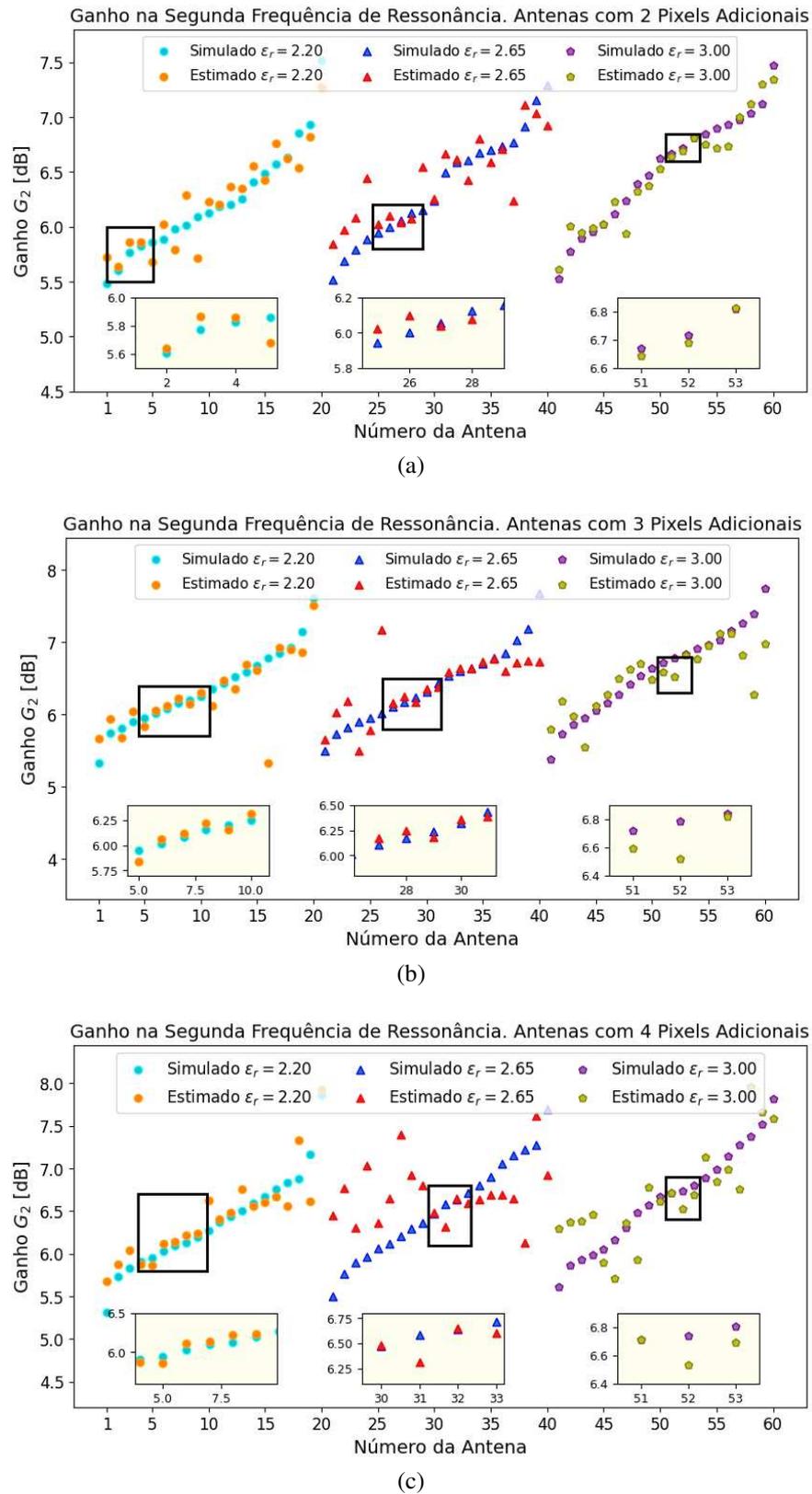


(b)



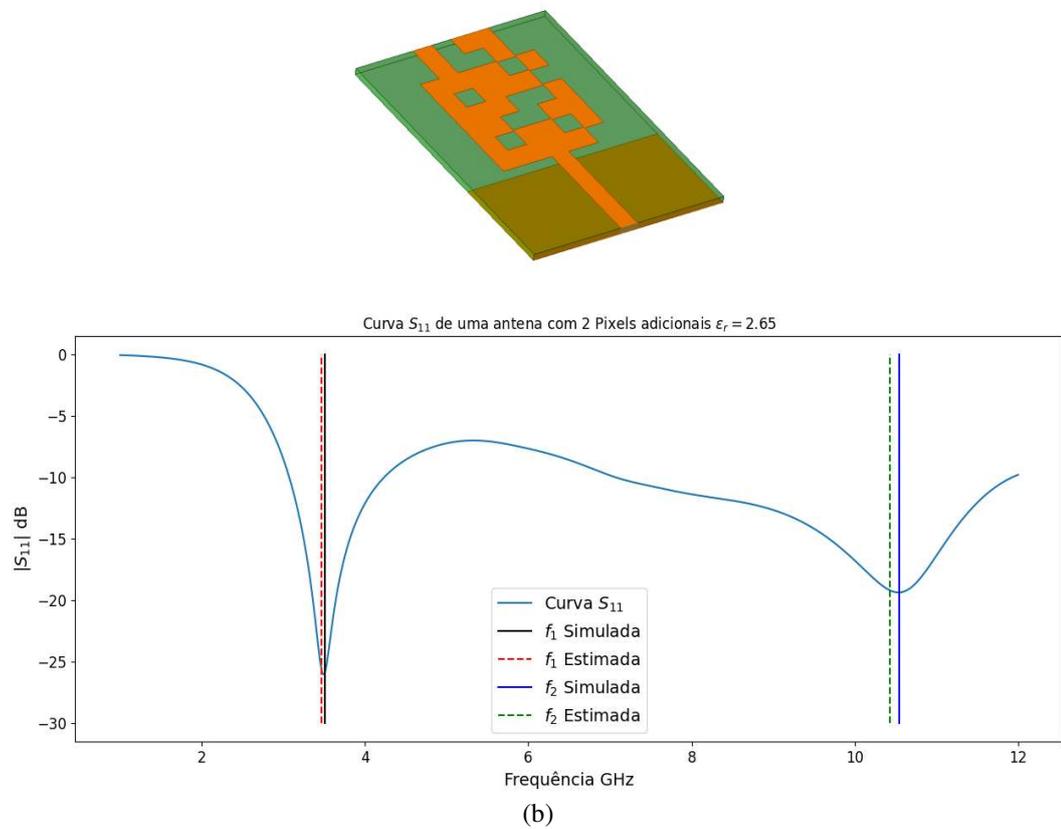
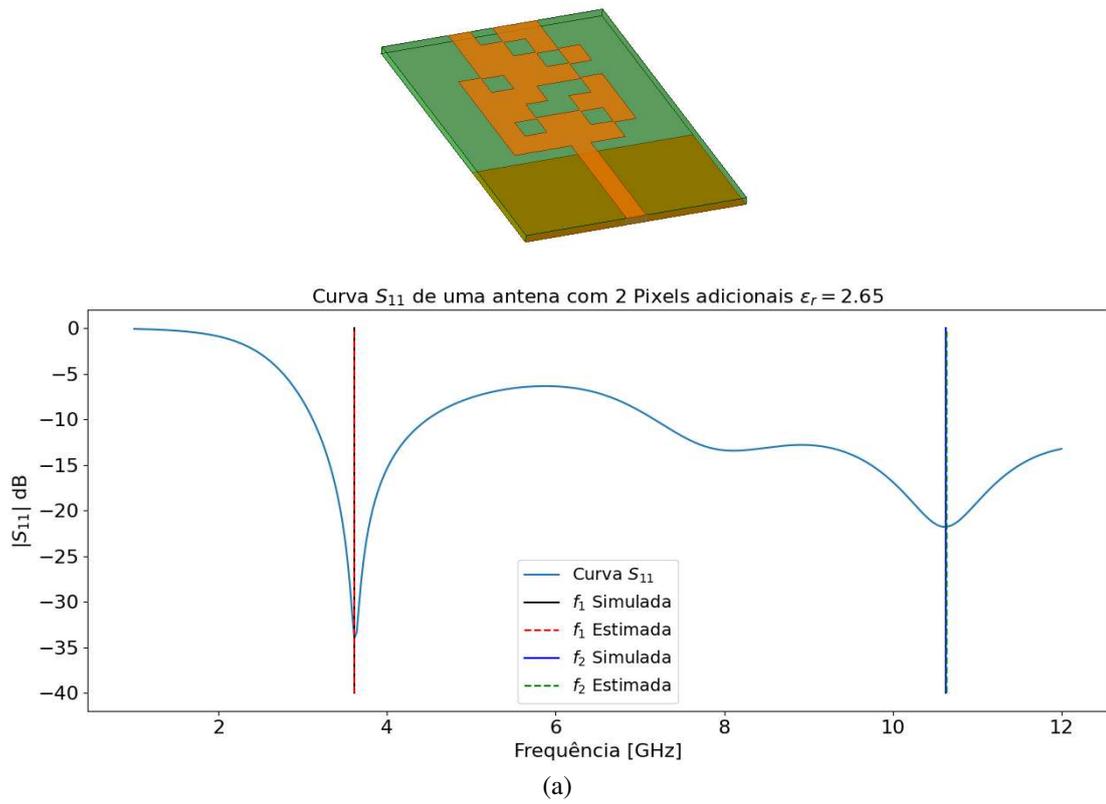
(c)

**Figura 4.15.** Ganho na primeira frequência de ressonância. (a) Antenas com agrupamentos de 2 pixels adicionais. (b) Antenas com agrupamentos de 3 pixels adicionais. (c) Antenas com agrupamentos de 4 pixels adicionais.

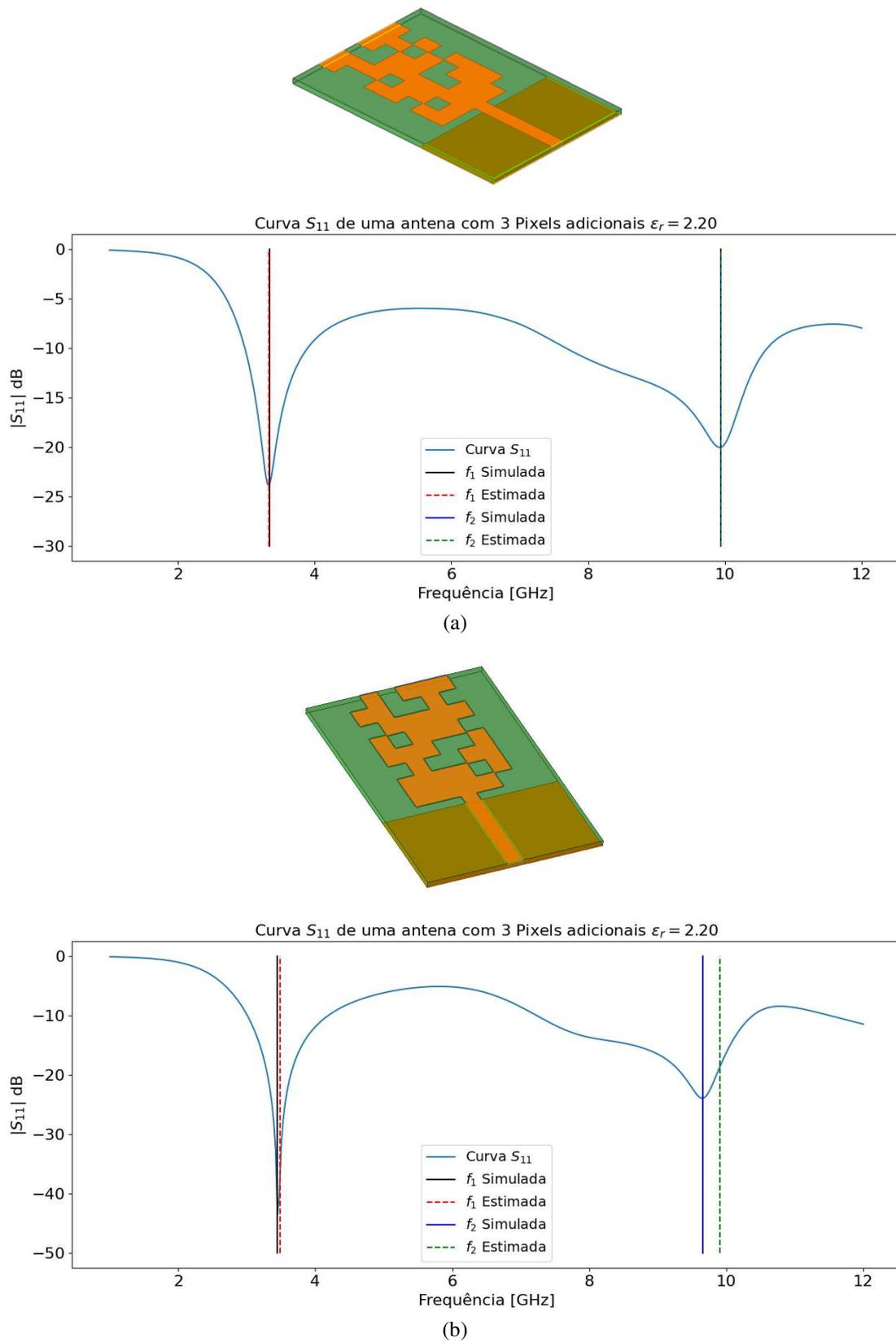


**Figura 4.16.** Ganho na segunda frequência de ressonância. (a) Antenas com agrupamentos de 2 pixels adicionais. (b) Antenas com agrupamentos de 3 pixels adicionais. (c) Antenas com agrupamentos de 4 pixels adicionais.

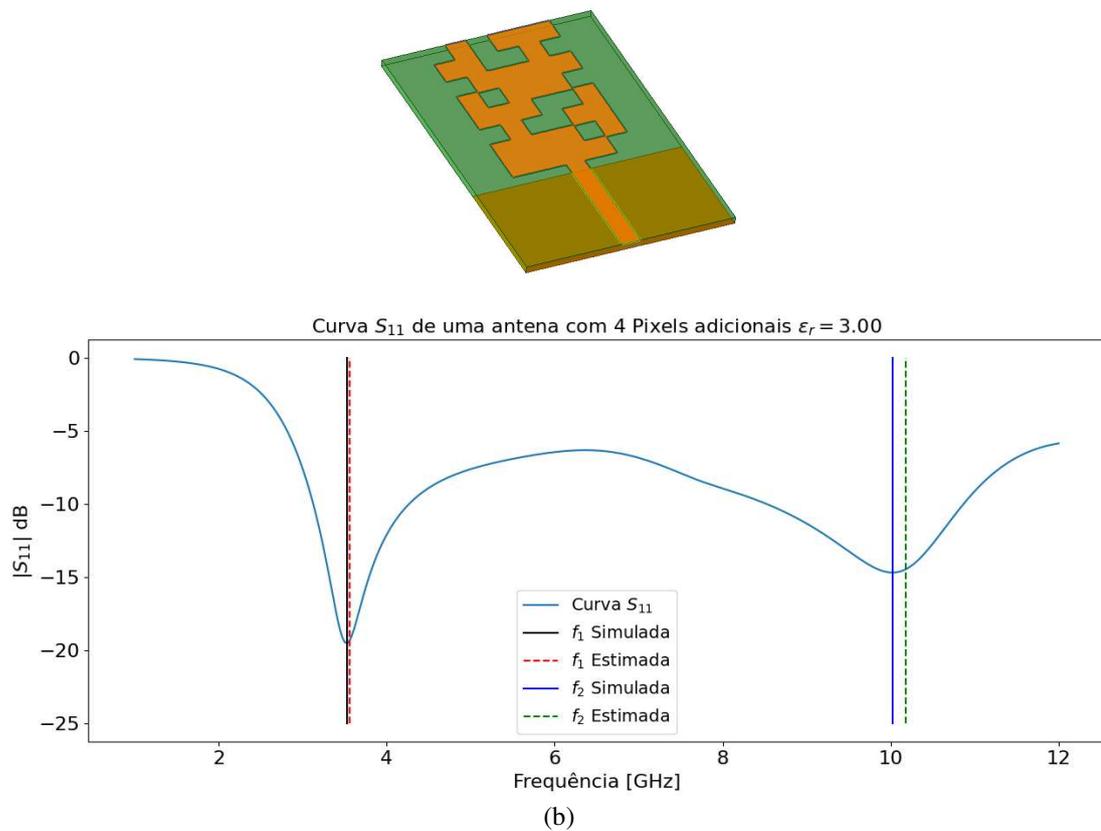
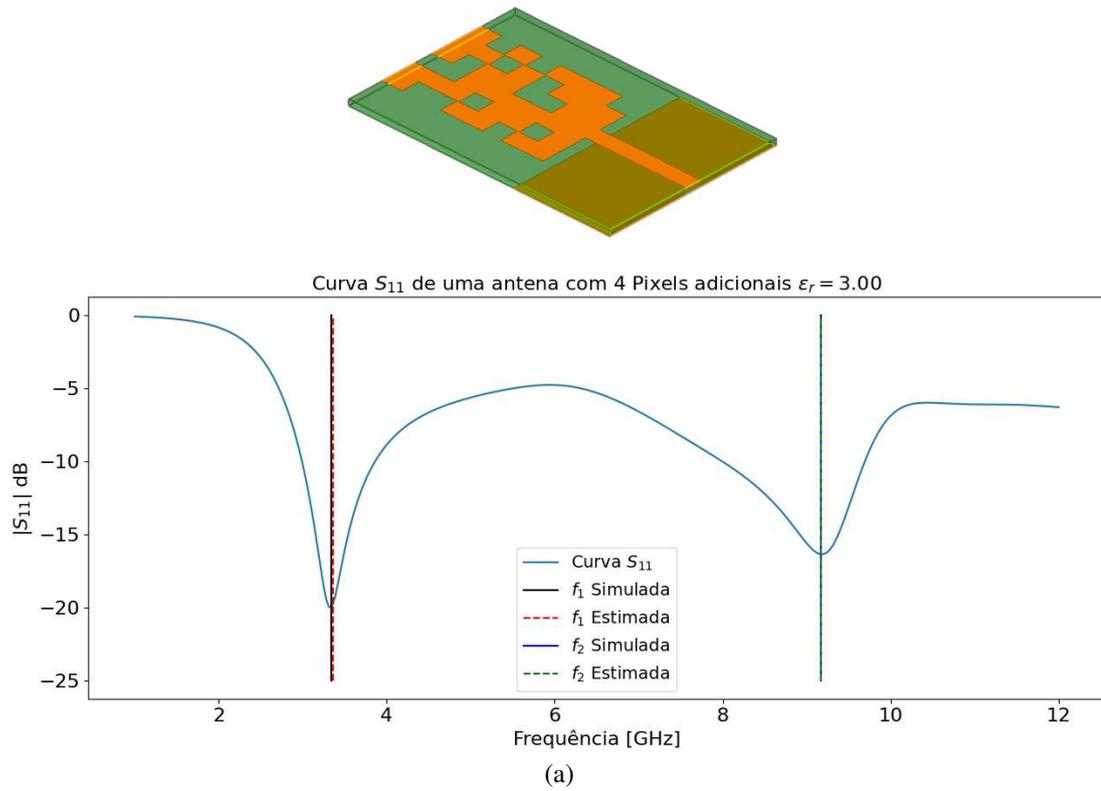
Na figura 4.17 são apresentados gráficos contendo o parâmetro  $S_{11}$  de antenas com  $\epsilon_r = 2,65$  e com agrupamentos de 2 pixels adicionais na antena de referência. Para a configuração de antena mencionada temos nas figuras 4.17(a) e (b) os casos que apresentaram a menor e maior taxa de erro, respectivamente. Na figura 4.18 são apresentados gráficos contendo o parâmetro  $S_{11}$  de antenas com  $\epsilon_r = 2,20$  e com agrupamentos de 3 pixels adicionais na antena de referência. Para a configuração de antena mencionada temos nas figuras 4.18(a) e (b) os casos que apresentaram a menor e maior taxa de erro, respectivamente. Na figura 4.19 são apresentados gráficos contendo o parâmetro  $S_{11}$  de antenas com  $\epsilon_r = 3,00$  e com agrupamentos de 4 pixels adicionais na antena de referência. Para a configuração de antena mencionada temos nas figuras 4.19(a) e (b) os casos que apresentaram a menor e maior taxa de erro, respectivamente.



**Figura 4.17.** Curvas  $S_{11}$  de antenas com agrupamentos de 2 pixels adicionais na antena de referência considerando  $\epsilon_r = 2,65$ . (a) Para o menor erro. (b) Para o maior erro.



**Figura 4.18.** Curvas  $S_{11}$  de antenas com agrupamentos de 3 pixels adicionais na antena de referência considerando  $\epsilon_r = 2,20$ . (a) Para o menor erro. (b) Para o maior erro.



**Figura 4.19.** Curvas  $S_{11}$  de antenas com agrupamentos de 4 pixels adicionais na antena de referência considerando  $\epsilon_r = 3,00$ . (a) Para o menor erro. (b) Para o maior erro.

A partir dos resultados apresentados, observamos que a rede apresentou um bom desempenho para prever ambas as frequências de ressonância, no entanto observamos que a rede teve um melhor resultado para predição da primeira frequência. A Tabela 4.5 apresenta os resultados preditivos em termos de erro relativo médio expresso em porcentagem.

**Tabela 4.5.** Erro percentual relativo médio nos dados de treino, validação e teste.

	$f_1$	$f_2$	$D_1$	$G_1$	$D_2$	$G_2$
Treino	0,629%	0,963%	0,304%	0,419%	1,240%	1,234%
Validação	0,797%	1,179%	0,396%	0,480%	3,727%	3,696%
Teste	0,690%	0,993%	0,336%	0,429%	3,111%	3,080%

## 4.7 Conclusão do Capítulo

Neste estudo de caso foi abordado o uso de CNNs para prever frequências de ressonância, diretividades e ganhos de um modelo de antena em microfita pixelada de banda dupla para três diferentes permissividades. Como entrada, a rede recebeu informação da geometria da antena através uma matriz que representa a presença ou ausência de pixels metalizados. No primeiro experimento, a rede foi treinada durante 2.000 épocas para prever as duas primeiras frequências de ressonâncias para os três valores de permissividade, e ao final apresentou um ótimo resultado com uma taxa de aprendizado de 0,86% e com erro médio de 1,10%. No segundo experimento, foram considerados mais parâmetros de antena (o ganho e a diretividade) na entrada da rede. E, mesmo assim, a rede apresentou um excelente resultado na predição dos dados com uma taxa de aprendizado de 0,82%, e com erro médio de 3%, para os dados de diretividades e ganhos para segunda faixa de frequência. Por fim, verificamos em nosso estudo que é bastante promissora a aplicação de CNNs para solução desse tipo de problema.

## CAPÍTULO 5

### Conclusão

Este trabalho foi dividido em dois estudos de caso, com a finalidade de propor uma análise para diferentes arquiteturas de RNAs para modelagem das características de antenas em microfita. No primeiro estudo de caso, foi proposto um modelo GAN, que obteve excelentes resultados na geração de novos designs de antenas *patch* retangulares semelhantes àqueles em que foi treinada. Ou seja, o modelo proposto foi capaz de aprender a relação entre as características das antenas e suas respectivas geometrias, criando assim amostras sintéticas realistas que atendem aos requisitos de desempenho pré-estabelecido, apresentando uma acurácia de 65% com taxa de erro de 0,08%. Já o segundo modelo proposto foi uma MLP, que, como esperado, também apresentou excelentes resultados, pois conseguiu obter características de saída a partir dos dados de entrada, superando até mesmo os resultados do GAN. O modelo final apresentou uma acurácia de 95% e com taxa de erro de 0,0009%. O terceiro modelo escolhido foi uma CGAN, na qual foi condicionada a gerar dados sintéticos de antena de *patch* para uma dada banda de frequências. O modelo apresentou ótimos resultados na geração dessas antenas cada faixa de frequência (*label*) que foi condicionada.

O segundo estudo de caso teve como foco a predição de alguns parâmetros fundamentais de antena para o caso de antenas pixeladas em microfita e de dupla banda, a partir da representação da superfície pixelada e considerando três diferentes valores de permissividade do substrato. Nesse segundo estudo de caso, o emprego de uma CNN mostrou boa eficácia na predição das duas frequências de ressonância e seus respectivos ganhos e diretividades.

Assim, conseguimos com êxito validar o uso de RNAs para geração e predição de novas configurações de antenas a partir dos dados de treinamento. Os modelos propostos apresentaram eficácia de aprendizado estatístico dos parâmetros mantendo boa distribuição dos dados. Uma das vantagens do uso de RNAs é que, após um treinamento apropriado, a rede pode eliminar a necessidade de realizar processos iterativos complexos repetidamente ao lidar com novos casos. A arquitetura exclusiva da RNA pode fazer previsões aproximadas para resultados de antenas em microfita, desde que os valores de entrada estejam dentro do domínio dos valores de treinamento. Por fim, os modelos avaliados, orientados por dados, podem simplificar e acelerar a modelagem de antenas em microfita.

## Referências Bibliográficas

- [1] O. Noakoosteen, J. Vijayamohanan, A. Gupta, and C. Christodoulou, “Antenna design using a gan-based synthetic data generation approach,” *IEEE Open Journal of Antennas and Propagation*, 3, 488-494, 2022.
- [2] J. P. Jacobs, “Accurate modeling by convolutional neural-network regression of resonant frequencies of dual-band pixelated microstrip antenna,” *IEEE Antennas and Wireless Propagation Letters*, 20(12), 2417-2421, 2021.
- [3] V. S. Kushwah, G. S. Tomar, “Design and analysis of microstrip patch antennas using artificial neural network,” *Trends in Research on Microstrip Antennas*, 55, 2017.
- [4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, and Y. Bengio, “Generative adversarial nets,” *Advances in neural information processing systems*, 27, 2014.
- [5] K. S. Um, S. W. Heo, “A novel antenna pattern design using generative adversarial network,” *International Conference on Consumer Electronics (ICCE)*. IEEE, 2019.
- [6] J. Vijayamohanan, O. Noakoosteen, A. Gupta, M. Martínez-Ramón, and C. G. Christodoulou, “On antenna Q-factor characterization with generative adversarial networks,” In *2020 IEEE International Symposium on Antennas and Propagation and North American Radio Science Meeting* (pp. 1643-1644), July 2020.
- [7] C. A. Balanis, *Antenna Theory: Analysis and Design*. John Wiley and Sons, 2016.
- [8] S. S. Chakravarthy, N. Sarveshwaran, S. Sriharini, and M. Shanmugapriya, “Comparative study on different feeding techniques of rectangular *patch* antenna,” *Thirteenth International Conference on Wireless and Optical Communications Networks (WOCN)* (pp. 1-6). IEEE, July 2016.
- [9] A. Géron, “*Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*,” O'Reilly Media, Inc., 2022.
- [10] C. M. Bishop, and N. M. Nasrabadi, “*Pattern recognition and machine learning* (Vol. 4, No. 4, p. 738),” New York: Springer, 2006.
- [11] S. Haykin, *Neural networks and learning machines, 3rd Edition*. Pearson Education India, 2009.

- [12] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "An introduction to reinforcement learning". The Biology and Technology of Intelligent Autonomous Agents, 90-127, 1995.
- [13] W. S. McCulloch, W. Pitts, "A logical calculus of the ideas immanent in nervous activity," The bulletin of mathematical biophysics, 5, 115-133, 1943.
- [14] I. N. D. Silva, D. H. Spatti, and R. A. Flauzino, "Redes neurais artificiais para engenharia e ciências aplicadas," ArtLiber, 2010.
- [15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in neural information processing systems, 25, 2012.
- [16] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, 521(7553), 436-444, 2015.
- [17] Y. Bengio, A. Courville, and P. Vincent, P., "Representation learning: A review and new perspectives," IEEE transactions on pattern analysis and machine intelligence, 35(8), 1798-1828, 2013.
- [18] D. H. Hubel, T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," The Journal of physiology, 160(1), 106, 1962.
- [19] D. L. Yamins, J. J. DiCarlo, "Using goal-driven deep learning models to understand sensory cortex," Nature neuroscience, 19(3), 356-365, 2016.
- [20] M. Carandini, J. B. Demb, V. Mante, D. J. Tolhurst, Y. Dan, B. A. Olshausen, and N. C. Rust, "Do we know what the early visual system does," Journal of Neuroscience, 25(46), 10577-10597, 2005.
- [21] I. Goodfellow, Y. Bengio, A. Courville, "*Deep Learning*," MIT Press, 2016.
- [22] S. Saha, "A comprehensive guide to convolutional neural networks," the ELI5 way. Towards data science, 15, 15, 2018.
- [23] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, "Unrolled generative adversarial networks". arXiv preprint arXiv:1611.02163, 2016.
- [24] V. Bok, and J. Langr. "*GANs in action: deep learning with generative adversarial networks*," Simon and Schuster, 2019.
- [25] A. Radford, L. Metz, and S. Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks," arXiv preprint arXiv:1511.06434, 2015.

[26] M. Mirza, and S. Osindero, “Conditional generative adversarial nets”. arXiv preprint arXiv:1411.1784, 2014.

[27] D. P. Kingma, “Adam: A method for stochastic optimization”. arXiv preprint arXiv:1412.6980, (2014).