



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Detecção de áudios sintéticos com análise de robustez a ruído utilizando espectrogramas Mel e redes neurais convolucionais

Gustavo Vieira Rocha Rabelo

Dissertação apresentada como requisito parcial para
conclusão do Mestrado em Informática

Orientador

Prof. Dr. Flavio de Barros Vidal

Brasília
2026

Ficha catalográfica elaborada automaticamente,
com os dados fornecidos pelo(a) autor(a)

V114d VIEIRA ROCHA RABELO, GUSTAVO
Detecção de áudios sintéticos com análise de robustez a ruído utilizando espectrogramas Mel e redes neurais convolucionais / GUSTAVO VIEIRA ROCHA RABELO; orientador Flavio de Barros Vidal. Brasília, 2026.
130 p.

Dissertação(Mestrado em Informática) Universidade de Brasília, 2026.

1. redes neurais convolucionais. 2. detecção de áudio sintético. 3. espectrograma Mel. 4. forense. I. de Barros Vidal, Flavio, orient. II. Título.



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Detecção de áudios sintéticos com análise de robustez a ruído utilizando espectrogramas Mel e redes neurais convolucionais

Gustavo Vieira Rocha Rabelo

Dissertação apresentada como requisito parcial para
conclusão do Mestrado em Informática

Prof. Dr. Flavio de Barros Vidal (Orientador)
CIC/UnB

Prof. Dr. David Menotti Gomes Prof. Dr. Francisco Assis de Oliveira Nascimento
Universidade Federal do Paraná Universidade de Brasília

Prof. Dr. Camilo Chang Dorea (Suplente)
Universidade de Brasília

Prof.a Dr.a Cláudia Nalon
Coordenadora do Programa de Pós-graduação em Informática

Brasília, 11 de agosto de 2026

Dedicatória

Dedico este trabalho aos meus pais, pelo apoio, pelos ensinamentos e pelos valores que me trouxeram até aqui. Sem a formação que me proporcionaram, esta conquista não seria possível.

Dedico à minha esposa, pelo amor, companheirismo e compreensão. Seu apoio constante foi fundamental nos momentos mais desafiadores.

Dedico aos meus filhos, que são fonte de motivação e inspiração. São a razão do meu empenho contínuo em evoluir.

Agradecimentos

Agradeço ao meu orientador, Prof. Dr. Flavio de Barros Vidal, pela orientação, disponibilidade e contribuições ao longo do desenvolvimento deste trabalho. Sua experiência e direcionamento foram fundamentais para a condução desta pesquisa.

Agradeço também ao Laboratório de Imagens, Sinais e Acústica (LISA), pelo ambiente de pesquisa e pelo suporte que contribuíram para a realização deste estudo.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), por meio do Acesso ao Portal de Periódicos.

Resumo

Os rápidos avanços em modelos de síntese de fala baseados em redes neurais têm introduzido desafios para a identificação de áudios sintéticos gerados por essas tecnologias. Entre as ferramentas utilizadas nesse contexto, está o *vocoder* (do inglês *voice encoder*), um modelo capaz de analisar e sintetizar sinais de voz. Ele permite não apenas a compactação e a modificação do áudio, mas também a geração de fala artificial, incluindo a recriação do timbre e das características específicas de uma voz humana. Este trabalho implementa uma abordagem para detectar o uso de *vocoders* neurais por meio da análise de imagens de espectrogramas Mel, utilizando redes neurais convolucionais (CNNs) para identificar características distintivas presentes em áudios gerados por *vocoders*. Começando com uma visão geral abrangente dos *vocoders*, seu desenvolvimento histórico e as tecnologias que impulsionaram os avanços dos *vocoders* neurais, destacamos a transição das técnicas tradicionais de processamento de sinal para a síntese impulsionada por aprendizado profundo. Nossa abordagem envolve testar várias arquiteturas de CNNs e estressá-las sob diferentes condições de ruído para avaliar a robustez e a eficácia de cada modelo na distinção entre imagens de espectrogramas de áudios sintéticos e reais. Os resultados demonstram que esse método baseado em imagens é eficaz para a detecção de *vocoders*, embora testes adicionais sejam necessários para melhorar o desempenho em diversos ambientes de áudio e tipos de *vocoders*.

Palavras-chave: redes neurais convolucionais, detecção de áudio sintético, espectrograma Mel, forense

Abstract

Recent advances in neural network-based speech synthesis have created significant challenges for the identification of synthetic audio generated by these technologies. Among the tools used in this context is the *vocoder* (a term derived from the contraction of *voice encoder*), a model capable of analyzing and synthesizing speech signals. It enables not only the compression and modification of audio but also the generation of artificial speech, including the recreation of a human voice's timbre and specific characteristics. This work implements an approach to detect neural *vocoders* through the analysis of Mel spectrogram images, using convolutional neural networks (CNNs) to identify distinctive features present in audio generated by *vocoders*. Starting with a comprehensive overview of *vocoders*, their historical development, and the technologies that have driven the advances in neural *vocoders*, we highlight the transition from traditional signal processing techniques to deep learning-driven synthesis. Our approach involves testing multiple CNN architectures and subjecting them to different noise conditions to assess each model's robustness and effectiveness in distinguishing between spectrogram images of synthetic and real audio. The results demonstrate that this image-based method is effective for *vocoder* detection, although further testing is required to improve performance across diverse audio environments and *vocoder* types.

Keywords: convolutional neural networks, synthetic audio detection, Mel spectrogram, forensics

Sumário

1	Introdução	1
1.1	Motivação	1
1.2	Objetivo	3
1.2.1	Objetivo Geral	3
1.2.2	Objetivos Específicos	3
1.3	Justificativa	4
1.4	Organização do Trabalho	5
2	Fundamentação Teórica	7
2.1	Espectrogramas Mel	7
2.2	<i>Vocoders</i> Analógicos	10
2.2.1	O <i>Vocoder</i> de Dudley	10
2.2.2	Outros <i>Vocoders</i> Analógicos	12
2.3	<i>Vocoders</i> Digitais	12
2.4	<i>Vocoders</i> Neurais	14
2.4.1	WaveNet	15
2.4.2	WaveGlow	15
2.4.3	WaveRNN	17
2.4.4	MelGAN	17
2.4.5	Parallel WaveGAN	19
2.4.6	DiffWave	19
2.4.7	HiFi-GAN	20
2.4.8	StyleMelGAN	21
2.4.9	Multi-Band MelGAN	22
2.4.10	BigVGAN	23
2.4.11	Vocos	23
3	Trabalhos Relacionados	27
3.1	Detectores de <i>deepfake</i> de áudio	27

3.2	Competições de Detecção de <i>Deepfakes</i>	37
3.2.1	ASVspoof Challenge	38
3.2.2	ADD Challenge	40
4	Metodologia Proposta	42
4.1	Dataset	44
4.1.1	Base de Áudios Originais	44
4.1.2	Geração de Áudio Sintético	44
4.1.3	Adição de ruído	47
4.1.4	Geração do Conjunto de Imagens de Espectrograma Mel	48
4.2	Detector Usando Rede Neural Convolucional	50
4.2.1	Definição da Arquitetura	50
4.2.2	Configuração Experimental Proposta	51
5	Resultados	56
5.1	Dataset	56
5.2	Detector Usando Rede Neural Convolucional	58
5.2.1	Escolha da arquitetura	58
5.2.2	Análise do Processo de Decisão dos Modelos	60
5.2.3	Testes de Generalização	69
6	Conclusão	83
	Referências	88
	Anexo	100
I	Curvas de Loss e de aEER de treinamento dos modelos usados em testes de generalização	101
II	Curvas ROC da saída bonafide dos modelos usados em testes de generalização	105
III	Desempenho dos modelos por <i>vocoder</i> em função do nível de ruído (BER)	109

Lista de Figuras

2.1	Espectrograma Mel mostrando a variação de frequências e potências ao longo do tempo, com cores mais brilhantes indicando maior potência do sinal de áudio.	8
2.2	Exemplo de banco de filtros triangulares usado para gerar um espectrograma Mel. Reproduzido de [1].	10
2.3	Circuito esquemático do voder de Dudley. Reproduzido de [2].	12
2.4	WaveNet: à esquerda, o codificador, que cria representações latentes discretas; à direita, o decodificador, que recebe o conteúdo latente e as condições para gerar o áudio sintético. Adaptado de [3].	16
2.5	WaveGlow: à direita, o fluxo de treinamento e inferência; à esquerda, o detalhe da <i>affine coupling layer</i> condicionada pelo espectrograma Mel. Adaptado de [4].	17
2.6	MelGAN: à esquerda, o gerador recebe espectrograma Mel para sintetizar a onda sonora; à direita, discriminadores são responsáveis por decidir se a onda sonora é real ou falsa. Adaptado de [5].	18
2.7	Parallel WaveGAN: o discriminador recebe falas reais e falas geradas pelo gerador e deve decidir se são verdadeiras ou falsas. Adaptado de [6].	20
2.8	DiffWave: mostra o processo de difusão direto, em direção à direita, e reverso, em direção à esquerda. Adaptado de [7].	21
2.9	StyleMelGAN: à esquerda, arquitetura do gerador; à direita, arquitetura dos discriminadores. Adaptado de [8].	22
2.10	Multi-Band MelGAN: no bloco de cima, o gerador; no bloco de baixo, três discriminadores. Adaptado de [9].	24
2.11	BigVGAN: à esquerda, arquitetura do gerador; à direita, arquitetura dos discriminadores. Adaptado de [10].	25

2.12	Arquitetura do Vocos. Os <i>features</i> de entrada (espectrogramas Mel) são processados por uma pilha de blocos convolucionais. A rede produz parâmetros espectrais que são transformados em magnitude ($\exp$) e fase (φ), e então convertidos em áudio no domínio do tempo por meio da Transformada Inversa de Fourier de Tempo Curto (<i>Inverse Short-Time Fourier Transform</i> — ISTFT), resultando no sinal final. Adaptado de [11].	26
3.1	Arquitetura de detecção de <i>deepfake</i> de áudio proposta por <i>The Vicomtech Audio Deepfake Detection System based on Wav2Vec2 for the 2022 ADD Challenge</i> . Múltiplas camadas de wav2vec2 são combinadas antes de ser entregues ao classificador neural. Reproduzido de [12].	32
4.1	Histograma de duração dos áudios	45
4.2	Exemplos de espectrogramas Mel constantes do dataset, com crop 80x80.	50
4.3	Arquitetura básica do detector de <i>vocoder</i> baseado em CNN: a rede é alimentada com imagens bitmap de espectrogramas Mel para prever qual <i>vocoder</i> gerou o áudio correspondente. Adaptado de [13].	51
5.1	Curvas de Loss e de aEER de treinamento para escolha da arquitetura CNN.	59
5.2	Matriz de confusão normalizada - AlexNet	61
5.3	Matriz de confusão normalizada - ResNet34	62
5.4	Matriz de confusão normalizada - ResNet50	63
5.5	Matriz de confusão normalizada - VGG16	64
5.6	Matriz de confusão normalizada - VGG19	65
5.7	Exemplos de espectrogramas Mel com mapas de ativação obtidos por Grad-CAM, destacando as regiões mais relevantes para a decisão das redes neurais.	67
5.8	(a) Exemplo de espectrograma Mel gerado a partir de uma amostra de áudio sintetizada pelo <i>vocoder</i> v_11 sem adição de ruído. (b) Visualização dos 512 <i>feature maps</i> 7x7 extraídos da última camada convolucional da arquitetura VGG19 para essa amostra.	69
5.9	Projeção bidimensional obtida por UMAP (<i>Uniform Manifold Approximation and Projection</i>) a partir dos vetores de importância gerados pelo teste de ablação dos 512 <i>feature maps</i> da última camada convolucional da arquitetura VGG19.	70
5.10	Curvas de Loss e de aEER dos treinamentos 1 e 2 para criação dos modelos a serem usados em testes de generalização. Para todos os 21 treinamentos, veja Anexo I.	73

5.11	Curvas ROC da saída <i>bonafide</i> dos modelos gerados a partir dos treinamentos 1 e 2 a serem usados em testes de generalização. Para todos os 21 treinamentos, veja Anexo II.	74
5.12	Ilustração de distribuições de (a) probabilidades (após sigmoide) e (b) logits (antes da sigmoide) para amostras de validação <i>bonafide</i> e sintéticas (dados colhidos e inferidos da partição de validação do Treinamento 19).	77
5.13	Curvas da taxa de erro balanceado (<i>Balanced Error Rate</i> – BER) em função do nível de ruído (SNR), para diferentes estratégias de definição de <i>threshold</i>	79
5.14	Performance dos modelos para <i>vocoder</i> específico	80
I.1	Curvas de Loss e de aEER de treinamento dos modelos usados em testes de generalização	102
II.1	Curvas ROC da saída <i>bonafide</i> dos modelos a serem usados em testes de generalização	106
III.1	Performance dos modelos para <i>vocoder</i> específico	110

Lista de Tabelas

4.1	Estatísticas de duração por locutor	54
4.2	Resumo dos <i>vocoders</i> utilizados na geração de áudio sintético	55
4.3	Duração do áudio para largura mínima de 80 pixels	55
5.1	Distribuição do número de áudios por <i>vocoder</i> , frequência de amostragem e nível de ruído (SNR).	57
5.2	Desempenho das arquiteturas de redes neurais convolucionais em termos de aEER.	60
5.3	Configuração dos treinamentos com <i>vocoders</i> , microfones e locutores externos.	72
5.4	Comparação da taxa média de erro balanceado entre níveis de ruído e estratégias de definição de <i>threshold</i>	78
5.5	Desempenho (BER) por <i>vocoder</i> e diferentes configurações de treinamento.	81

Lista de Abreviaturas e Siglas

AASIST Additive Angular-margin Softmax Integrated Spoofing-resistant Transformer.

ACC Accuracy (Acurácia).

ADD Audio Deepfake Detection (Detecção de Deepfakes de Áudio).

aEER average Equal Error Rate (Taxa Média de Erro Igual).

AI Artificial Intelligence (Inteligência Artificial).

AlexNet AlexNet (Arquitetura Clássica de Rede Neural Convolucional de 8 Camadas).

API Application Programming Interface (Interface de Programação de Aplicações).

ASR Automatic Speech Recognition (Reconhecimento Automático de Fala).

ASV Automatic Speaker Verification (Verificação Automática de Locutor).

ASVspoof Automatic Speaker Verification Spoofing and Countermeasures Challenge.

AttM Attentive Merging.

AUC Area Under the Curve (Área sob a Curva).

BER Balanced Error Rate (Taxa de Erro Balanceada).

BigVGAN Big Generative Adversarial Network for Vocoder (Grande Rede Geradora Adversarial para Vocoder).

CELP Code-Excited Linear Prediction (Predição Linear Excitada por Codebook).

CFCC Cochlear Filter Cepstral Coefficients (Coeficientes Cepstrais de Filtro Coclear).

CIC Departamento de Ciência da Computação.

CM Countermeasure (Contramedida).

CMU Carnegie Mellon University.

CNN Convolutional Neural Network (Rede Neural Convolucional).

CPU Central Processing Unit (Unidade Central de Processamento).

CQCC Constant Q Cepstral Coefficients (Coeficientes Cepstrais de Q Constante).

CQT Constant-Q Transform (Transformada Constante-Q).

CUDA Compute Unified Device Architecture (Arquitetura de Cálculo Unificado da NVIDIA).

DCNN Deep Convolutional Neural Network (Rede Neural Convolucional Profunda).

DCT Discrete Cosine Transform (Transformada Discreta de Cosseno).

DFT Discrete Fourier Transform (Transformada Discreta de Fourier).

DL Deep Learning (Aprendizado Profundo).

DNN Deep Neural Network (Rede Neural Profunda).

DSP Digital Signal Processing (Processamento Digital de Sinais).

EER Equal Error Rate (Taxa de Erro Igual).

F1 F1-Score (Medida F1).

FAD Fake Audio Detection (Detecção de Áudio Falso).

FAR False Acceptance Rate (Taxa de Falsa Aceitação).

FastDiff Fast Diffusion Model (Modelo de Difusão Rápida para Síntese de Áudio).

FFT Fast Fourier Transform (Transformada Rápida de Fourier).

FPR False Positive Rate (Taxa de Falsos Positivos).

FRR False Rejection Rate (Taxa de Falsa Rejeição).

GAN Generative Adversarial Network (Rede Geradora Adversarial).

GAN-TTS GAN-based Text-to-Speech System.

GitHub Plataforma de hospedagem e versionamento de código.

GMM Gaussian Mixture Model (Modelo de Mistura Gaussiana).

GPU Graphics Processing Unit (Unidade de Processamento Gráfico).

Grad-CAM Gradient-weighted Class Activation Mapping (Mapeamento de Ativação de Classe Ponderado por Gradiente).

HiFi-GAN High-Fidelity Generative Adversarial Network.

ISTFT Inverse Short-Time Fourier Transform (Transformada Inversa de Fourier de Tempo Curto).

kNN k-Nearest Neighbors (k-Vizinhos Mais Próximos).

Librosa Biblioteca Python para análise de áudio.

LJSpeech Dataset LJSpeech.

LPC Linear Predictive Coding (Codificação Preditiva Linear).

LSTM Long Short-Term Memory (Memória de Curto e Longo Prazo).

MEL Mel-Spectrogram (Espectrograma Mel).

MelGAN Mel Generative Adversarial Network.

MFCC Mel-Frequency Cepstral Coefficients (Coeficientes Cepstrais na Escala Mel).

ML Machine Learning (Aprendizado de Máquina).

MLAAD Multi-Language Audio Anti-Spoofing Dataset.

MOS Mean Opinion Score (Pontuação Média de Opinião).

NumPy Biblioteca Python para computação numérica.

OOD Out-of-Distribution (Fora da Distribuição).

PPGI Programa de Pós-Graduação em Informática.

PSTN Public Switched Telephone Network (Rede Telefônica Pública Comutada).

PyTorch Biblioteca de aprendizado profundo PyTorch.

ReLU Rectified Linear Unit (Unidade Linear Retificada).

ResNet34 Residual Network com 34 Camadas (Rede Neural Convolutacional com Conexões Residuais).

ResNet50 Residual Network com 50 Camadas (Rede Neural Convolutacional com Conexões Residuais).

RNN Recurrent Neural Network (Rede Neural Recorrente).

ROC Receiver Operating Characteristic (Curva ROC).

SAD Speech Activity Detection (Detecção de Atividade de Fala).

SciPy Biblioteca Python para computação científica.

SNR Signal-to-Noise Ratio (Relação Sinal-Ruído).

SSL Self-Supervised Learning (Aprendizado Auto-Supervisionado).

STFT Short-Time Fourier Transform (Transformada de Fourier de Curto Prazo).

t-DCF Tandem Detection Cost Function (Função de Custo de Detecção em Tandem).

TPR True Positive Rate (Taxa de Verdadeiros Positivos).

TTS Text-to-Speech (Texto para Fala).

UMAP Uniform Manifold Approximation and Projection (Aproximação e Projeção Uniforme de Variedades).

UnB Universidade de Brasília.

VAD Voice Activity Detection (Detecção de Atividade de Voz).

VAE Variational Autoencoder (Autoencoder Variacional).

VCTK Voice Cloning Toolkit Dataset.

VGG16 Visual Geometry Group 16-layer Network (Arquitetura de Rede Neural Convolutacional Profunda com 16 Camadas).

VGG19 Visual Geometry Group 19-layer Network (Arquitetura de Rede Neural Convolutacional Profunda com 19 Camadas).

Vocoder Voice Encoder (Codificador de Voz).

Vocos Vocoder based on spectral coefficients (Vocoder baseado em coeficientes espectrais).

VoIP Voice over Internet Protocol (Voz sobre IP).

VQ Vector Quantization (Quantização Vetorial).

WaveGAN Waveform Generative Adversarial Network.

WaveGlow Waveform Flow-Based Generative Model.

WaveNet Waveform Neural Network.

WaveRNN Waveform Recurrent Neural Network.

WavLM Waveform Language Model (Modelo de Linguagem de Forma de Onda).

WPT Wavelet Packet Transform (Transformada de Pacotes Wavelet).

Capítulo 1

Introdução

1.1 Motivação

Em 2019, criminosos utilizaram *deepfake* de voz — tecnologia baseada em inteligência artificial capaz de sintetizar ou manipular áudios, imagens e vídeos para imitar de forma realista a aparência ou voz de uma pessoa — para aplicar um golpe sofisticado, simulando com precisão a voz do diretor de uma empresa multinacional alemã do setor de energia. Durante uma chamada telefônica, eles ordenaram que o CEO de uma subsidiária britânica transferisse US\$ 243 mil para uma conta fraudulenta, sob a alegação de uma operação sigilosa. A tecnologia de inteligência artificial convenceu o interlocutor, reproduzindo não apenas o tom de voz, mas também o sotaque e o ritmo de fala do executivo, tornando a fraude extremamente convincente [14].

Mais recentemente, em 2024, um funcionário financeiro de uma multinacional em Hong Kong foi enganado por um golpe de *deepfake* de vídeo e áudio, no qual criminosos simularam uma videoconferência com o diretor financeiro da empresa, sediada no Reino Unido. Utilizando inteligência artificial, os golpistas imitaram as vozes e aparências dos diretores e instruíram o funcionário a transferir US\$ 25 milhões para uma conta fraudulenta. A transação só foi descoberta posteriormente, quando o funcionário percebeu que os verdadeiros diretores nunca haviam solicitado o pagamento [15].

Esses dois casos exemplificam o uso de *deepfakes* para fraudes financeiras sofisticadas, demonstrando como a inteligência artificial pode ser explorada para imitar vozes de pessoas em posições de autoridade. Ademais, *deepfakes* têm sido usados para fins políticos, com candidatos ou figuras públicas sendo falsamente retratados dizendo coisas que nunca disseram, com o intuito de prejudicá-los ou manipular a opinião pública: áudios falsificados distorcem as declarações desses indivíduos, comprometendo sua credibilidade e alterando o curso de campanhas eleitorais [16, 17].

Há também registros de uso de inteligência artificial para produzir conteúdo justificando ações militares, seja para incitar apoio público a uma guerra, seja para criar pretextos para intervenções em outros países [18].

Entre os danos causados pelo mau uso da inteligência artificial, [19] destaca o *spoofing* — ataque no qual um fraudador tenta manipular sistemas biométricos para se passar por outra pessoa —, [20] aponta riscos de difamação, sabotagem e desconfiança na ciência e [21] alerta para a incitação de atos de violência contra minorias, a distribuição de desinformação, a manipulação de evidências forenses digitais e a prática de extorsão.

A síntese de áudio artificial tem avançado significativamente, evoluindo dos primeiros sistemas de *vocoders* analógicos para os *vocoders* neurais modernos. O trabalho pioneiro de [2] descreve o primeiro *vocoder* analógico, originalmente desenvolvido para comprimir e transformar sinais de voz. Desde então, os *vocoders* passaram a ter aplicações diversas: na produção musical [22], na criação de efeitos sonoros [23], bem como em sistemas de *Text-to-Speech* (TTS), compactação de áudio e anonimização de locutores [24].

No entanto, com o surgimento recente do aprendizado profundo, houve uma transição para *vocoders* neurais, que geram áudio com uma fidelidade sem precedentes, imitando de forma muito próxima a voz humana real. A capacidade de gerar áudios sintéticos altamente realistas, indistinguíveis de gravações reais, oferece vários benefícios, como avanços na produção de conteúdo, treinamento de assistentes virtuais e aprimoramento de tecnologias de acessibilidade [25].

A criação desse tipo de conteúdo está ao alcance de todos, devido a aplicativos intuitivos que produzem áudios sintéticos, realizam troca de rosto e realizam sincronização labial com o áudio fornecido [26]. No entanto, essa mesma tecnologia tem sido utilizada de forma maliciosa. A ameaça dos *deepfakes* é preocupante em um cenário global onde a autenticidade da informação é fundamental, especialmente devido às aplicações mal-intencionadas que foram mencionadas anteriormente [27].

Diante desse contexto, torna-se necessário o desenvolvimento de técnicas eficazes para a detecção de áudios sintéticos gerados por *vocoders* neurais. Este estudo propõe uma abordagem baseada na análise de espectrogramas e no uso de redes neurais convolucionais (CNNs), que processam os espectrogramas dos áudios para extrair características únicas capazes de indicar o uso de *vocoders* neurais. As CNNs são particularmente adequadas para essa tarefa devido à sua capacidade de identificar padrões hierárquicos em dados de imagem, tornando-as eficazes na captura de características sutis e recorrentes presentes em espectrogramas gerados por *vocoders* [28], contribuindo assim para o avanço de soluções que possam identificar *deepfakes* de áudio com precisão, mesmo em condições de ruído.

1.2 Objetivo

Nesta seção são apresentados os objetivos geral e específicos deste trabalho, os quais definem o direcionamento da pesquisa e as etapas necessárias para o alcance dos resultados propostos.

1.2.1 Objetivo Geral

Investigar, desenvolver e avaliar uma abordagem para a diferenciação entre áudios reais e áudios sintéticos gerados por *vocoders* neurais, utilizando espectrogramas Mel (representação de frequência inspirada na percepção auditiva humana) e modelos de redes neurais convolucionais (*Convolutional Neural Networks* — CNNs). O trabalho enfatiza a análise da robustez e da capacidade de generalização dos modelos por meio de sua avaliação sob condições controladas de adição de ruído, buscando verificar seu desempenho em cenários mais próximos de aplicações reais. Dessa forma, pretende-se contribuir para o fortalecimento da segurança cibernética e para a verificação da autenticidade de conteúdos de áudio diante do crescimento das ameaças associadas a *deepfakes* de voz.

1.2.2 Objetivos Específicos

Para atingir o objetivo geral proposto, foram definidos os seguintes objetivos específicos:

1. Revisar o estado da arte: investigar a evolução dos *vocoders* e das técnicas de detecção de áudios sintéticos.
2. Propor uma metodologia baseada em imagens de espectrogramas Mel: explorar a utilização dessa representação para evidenciar padrões distintivos entre áudios reais e sintéticos.
3. Implementar e avaliar diferentes arquiteturas de redes neurais convolucionais (CNNs): verificar sua eficácia na tarefa de reconhecimento de áudio sintético.
4. Analisar a influência do ruído: estudar de forma sistemática como a variação da razão sinal-ruído (SNR) impacta o desempenho das CNNs, avaliando sua robustez frente a diferentes níveis de degradação.
5. Comparar o desempenho entre cenários: avaliar como diferentes *vocoders* e diferentes intensidades de ruído afetam métricas de acurácia e taxa de erro, identificando padrões de robustez ou fragilidade.

1.3 Justificativa

O contínuo aumento da qualidade dos áudios sintéticos produzidos por *vocoders* neurais, aliado ao crescente volume de conteúdos gerados por essas tecnologias, faz com que a detecção desses áudios demande soluções que sejam ao mesmo tempo eficazes e escaláveis. A escolha do uso de imagens de espectrogramas Mel como representação do sinal de áudio justifica-se pela sua capacidade de condensar, em forma visual, as informações espectrais perceptualmente mais relevantes para o ouvido humano. Essa representação é amplamente empregada em sistemas de síntese de fala baseados em redes neurais. Além disso, imagens de espectrogramas Mel permitem transformar o problema em um domínio visual, preservando a identidade do sinal de áudio e possibilitando a aplicação de redes neurais convolucionais, amplamente consolidadas na identificação de padrões em imagens. No contexto da detecção de áudios sintéticos, os espectrogramas Mel favorecem a identificação de artefatos característicos introduzidos por *vocoders* neurais.

Adicionalmente, a redução de informações durante o processo de transformação, como a perda de fase, pode contribuir para a preservação da identidade do locutor, sendo potencialmente interessante em cenários que envolvem privacidade e compartilhamento seguro de dados. Por outro lado, essa mesma simplificação pode resultar na perda de detalhes finos do sinal, o que reforça a necessidade de uma avaliação cuidadosa da robustez dos modelos, especialmente em condições adversas de ruído.

Dessa forma, pode-se listar as seguintes vantagens do uso de imagens de espectrograma Mel:

- Transformação do problema para visão computacional: a transformação do áudio em espectrogramas possibilita o uso de arquiteturas já consolidadas de redes neurais convolucionais.
- Captura de padrões espaço-temporais: os espectrogramas reúnem informações de tempo, frequência e intensidade em uma única representação, permitindo que redes convolucionais detectem padrões estruturais relevantes.
- Redução de dimensionalidade e redundância: a aplicação do banco de filtros Mel reduz a quantidade de informação redundante presente no sinal bruto, resultando em uma representação mais compacta.
- Preservação de privacidade (não-invertibilidade): a transformação unidirecional por funções não-lineares dificulta a reconstrução fiel do áudio original. Essa característica é útil em cenários que envolvem privacidade, soberania de dados e aprendizado federado.

Algumas desvantagens também emergem quando se usa imagens de espectrogramas Mel:

- Perda de informação: a conversão para espectrograma Mel implica na remoção da fase e na compressão da resolução espectral, o que pode eliminar detalhes finos usados na tarefa de detecção de *deepfake*.
- Dependência de hiperparâmetros: a representação depende de parâmetros como tamanho da janela, *hop length*, número de bandas Mel e função de janela, impactando os resultados caso não se entregue para a rede imagens geradas com os mesmos parâmetros usados nos treinamentos.
- Distorções na conversão para imagem: o processo de normalização, quantização (por exemplo, de *float* para *uint8*) e escolha de *colormap* pode introduzir artefatos visuais que não correspondem ao sinal original, dificultando o aprendizado.
- Custo computacional adicional: a geração do espectrograma envolve etapas como STFT e aplicação de filtros Mel, adicionando complexidade ao pipeline de processamento.

Do ponto de vista do modelo de aprendizado, as CNNs foram selecionadas por sua reconhecida eficiência na extração automática de padrões hierárquicos em dados de imagem. Aplicadas às imagens dos espectrogramas Mel, as CNNs são capazes de identificar características locais e globais que diferenciam áudios reais de áudios artificiais. Além disso, CNNs apresentam vantagens como robustez à variação de entrada, alta capacidade de generalização e desempenho consolidado em diversas aplicações de reconhecimento visual.

A combinação desses dois elementos — espectrogramas Mel como representação e CNNs como mecanismo de classificação — oferece, portanto, uma abordagem adequada para lidar com o problema em estudo. Essa escolha metodológica se alinha à necessidade de desenvolver sistemas forenses capazes de operar de forma confiável e em condições adversas de ruído e variabilidade de locutores.

1.4 Organização do Trabalho

Este trabalho está organizado em capítulos que seguem uma sequência lógica e progressiva.

O **Capítulo 2 – Fundamentação Teórica** apresenta a evolução histórica dos *vocoders*, desde os modelos analógicos e digitais até os *vocoders* neurais, com ênfase em suas arquiteturas e características principais.

O **Capítulo 3 – Trabalhos Relacionados** revisa pesquisas relevantes e as principais competições associadas à detecção de *deepfakes* de áudio, incluindo iniciativas como as competições ASVspoof e ADD.

Em seguida, o **Capítulo 4 – Metodologia Proposta** detalha os conjuntos de dados utilizados, o processo de geração dos espectrogramas, as arquiteturas de CNN implementadas e os experimentos conduzidos.

No **Capítulo 5 – Resultados** são apresentados os testes realizados com diferentes arquiteturas de CNNs, avaliando seu desempenho e robustez frente a ruídos e à variabilidade de *vocoders*, microfones e locutores.

Por fim, o **Capítulo 6 – Conclusão** apresenta as principais contribuições deste trabalho, discute os resultados obtidos e destaca limitações e possíveis direções para pesquisas futuras.

Essa estrutura visa assegurar uma progressão coerente: da contextualização teórica e análise do estado da arte, passando pela descrição da metodologia e pela apresentação dos resultados obtidos, até chegar à discussão das contribuições do trabalho, suas limitações e as perspectivas para pesquisas futuras.

Capítulo 2

Fundamentação Teórica

Este capítulo apresenta os fundamentos teóricos que sustentam o trabalho, abordando inicialmente os espectrogramas Mel, uma das representações mais utilizadas para análise e síntese de sinais de áudio. Em seguida, discute-se a evolução histórica dos *vocoders*, desde os modelos analógicos pioneiros até as abordagens digitais, destacando suas aplicações em telecomunicações, música e síntese de fala. Por fim, são explorados os avanços recentes dos *vocoders* neurais, que se consolidaram como ferramentas de alta fidelidade na geração de áudio.

2.1 Espectrogramas Mel

O áudio é um sinal de natureza complexa que abrange uma ampla variedade de sons, incluindo música, fala e ruídos ambientais. Ele é caracterizado por propriedades como frequência, amplitude e duração, que, juntas, contribuem para a percepção do som [29].

Para visualizar e analisar dados de áudio, técnicas como a Transformada Discreta de Fourier (*Discrete Fourier Transform* — DFT) são empregadas, convertendo sinais do domínio do tempo em representações no domínio da frequência [30]. Essa representação é chamada de espectrograma. O espectrograma fornece uma representação visual que revela como o conteúdo de frequência de um sinal de áudio varia ao longo do tempo [22].

Para criar um espectrograma, de acordo com [22], o sinal de áudio digital é dividido em pequenas janelas de tempo, chamadas de quadros, tipicamente de dezenas de milissegundos. Esses quadros podem ou não se sobrepor. Para cada quadro, a Transformada Discreta de Fourier é aplicada, resultando em um espectro de frequências, chamado, também, de *Short-Term Spectrum*, devido ao pequeno tamanho da janela. A coleção de todos esses espectros ao longo do tempo forma o espectrograma. O espectrograma possui dois eixos principais: o eixo horizontal representa o tempo, enquanto o eixo vertical representa

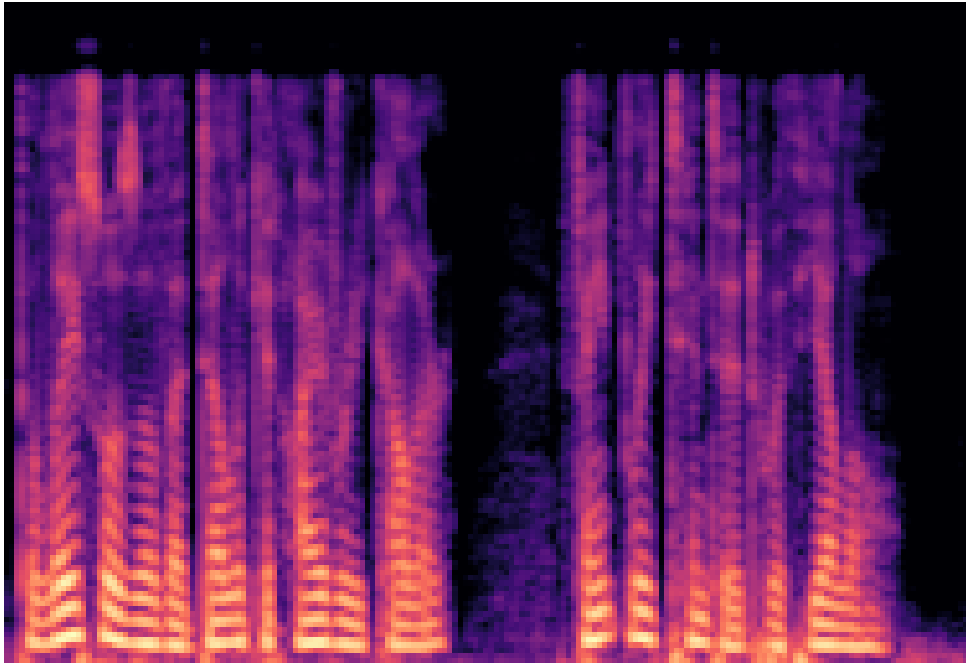


Figura 2.1: Espectrograma Mel mostrando a variação de frequências e potências ao longo do tempo, com cores mais brilhantes indicando maior potência do sinal de áudio.

as frequências. A intensidade ou energia de cada frequência em cada momento é indicada por uma escala de brilho ou cores.

O espectrograma Mel, segundo [29], é uma representação visual da intensidade das frequências de um sinal de áudio ao longo do tempo, projetada para se aproximar da percepção humana de som. Diferentemente do espectrograma descrito anteriormente, o espectrograma Mel utiliza a escala Mel, que comprime as frequências mais altas e expande as mais baixas, imitando a sensibilidade do ouvido humano. O espectrograma Mel também reduz a resolução espectral, agrupando frequências próximas em uma representação mais compacta. Conforme [31], a escala Mel tem esse nome pelo fato de ela estar relacionada à sensação humana de melodia. A Figura 2.1 exemplifica a imagem de um espectrograma Mel.

A origem do espectrograma Mel remonta a estudos psicoacústicos desenvolvidos no século XX, principalmente a partir da década de 1930, quando pesquisadores buscavam entender como os humanos percebem o som, gerando uma escala perceptiva para medir a altura tonal (*pitch*) de sons conforme percebida pelo ouvido. Foi formalizada a partir de experimentos, que ajustavam tons até que eles parecessem igualmente espaçados em termos de altura percebida pelos ouvintes. Ela foi construída de forma que em 1000 hertz fosse calculada como 1000 mels (a não ser em certas variantes). Percebeu-se que para frequências baixas a percepção de alteração do *pitch* é aproximadamente linear, enquanto que para frequências mais altas, a percepção de alteração do *pitch* é aproximadamente

logarítmica. Isso motivou a formalização de um mapeamento matemático entre escala Mel e escala em hertz [29, 32].

De acordo com [1], a montagem do espectrograma Mel segue algumas etapas básicas. Primeiro, o sinal de áudio é dividido em quadros de tempo, e em cada um é aplicada a Transformada Discreta de Fourier para calcular o espectro de frequência, como feito para calcular o espectrograma convencional. Em seguida, esse espectro é mapeado para a escala Mel, usando uma função de conversão que transforma a frequência em hertz para a escala perceptiva. Isso envolve dois elementos principais: a escolha da variante da escala e a aplicação de um banco de filtros triangulares sobre essa escala. Esses filtros são sobrepostos (ver Figura 2.2) e simulam como o ouvido humano responde a diferentes bandas de frequência, produzindo uma matriz com a energia em cada banda Mel ao longo do tempo [29].

Matematicamente, o espectrograma Mel pode ser definido, conforme [1], pela Equação 2.1:

$$X_{\text{mel}}(t, s) = \sum_{k=0}^{K-1} |X(t, k)| \cdot H_{\text{mel}}(s, k), \quad (2.1)$$

onde:

- $X(t, k) \in \mathbb{C}$: coeficiente da Transformada Discreta de Fourier do sinal no quadro t , associado à frequência discreta k ;
- $H_{\text{mel}}(s, k) \in \mathbb{R}_{\geq 0}$: resposta do filtro triangular Mel de índice s na frequência k ;
- $t \in \{1, \dots, T\}$: índice do quadro (segmentos temporais do sinal);
- $k \in \{0, \dots, K - 1\}$: índice de frequência discreta na DFT;
- $s \in \{1, \dots, S\}$: índice do filtro triangular na escala Mel.

Diversas escalas Mel foram propostas, sendo que o mapeamento mais comum, proposto por [32] é dado por

$$m = 2595 \cdot \log \left(1 + \frac{f}{700} \right), \quad (2.2)$$

onde f é a frequência em hertz.

O espectrograma Mel é amplamente utilizado em aplicações de reconhecimento automático de fala [33], identificação de locutor [34], análise musical [35, 36], e até mesmo em tarefas de detecção de emoção ou saúde mental a partir da voz [37]. Seu sucesso se deve à sua capacidade de capturar aspectos perceptualmente relevantes do som com relativa simplicidade computacional [38]. No entanto, há críticas. Uma delas é que, por

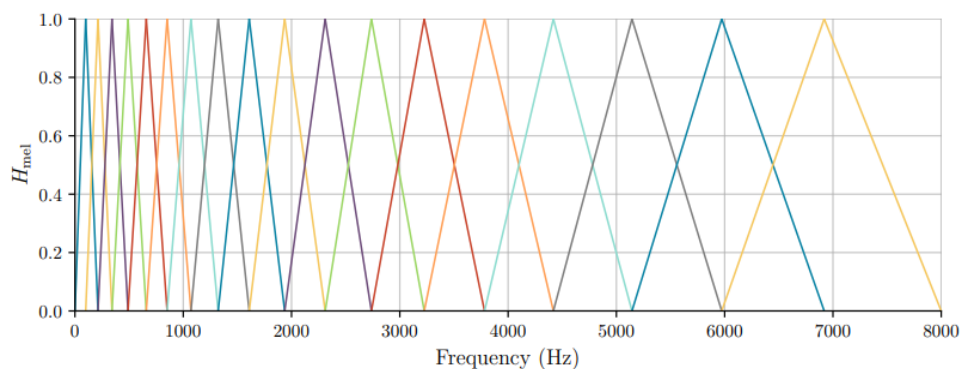


Figura 2.2: Exemplo de banco de filtros triangulares usado para gerar um espectrograma Mel. Reproduzido de [1].

ser baseado em suposições sobre a audição humana média, ele pode não representar bem indivíduos com padrões auditivos diferentes [39]. Além disso, por simplificar e agrupar informações espectrais, o espectrograma Mel pode descartar detalhes finos do som que são importantes em algumas tarefas específicas, como síntese de voz de alta fidelidade ou identificação de ruídos específicos [5].

2.2 Vcoders Analógicos

2.2.1 O Vocoder de Dudley

O primeiro sistema que conseguiu sintetizar a voz com sucesso foi inventado por Homer Dudley, engenheiro dos Laboratórios Bell Telephone, nos Estados Unidos, no final da década de 1930 [40]. Dudley demonstrou seu sistema usando um equipamento, semelhante a um órgão, conhecido como *voder*, acrônimo em inglês *voice operation demonstrator*. Era operado por um humano que controlava dez atuadores (um para cada dedo das mãos), uma barra de pulso e um pedal. Os dedos movimentavam alavancas que modulavam sinais em dez faixas de frequência diferentes. Esses sinais podiam corresponder a um de dois componentes: tom vocálico (*buzzer-like tone*) ou ruído parecido com um chiado (*hiss-like noise*). Essencialmente, o tom vocálico era utilizado para produzir vogais, enquanto o ruído ajudava a gerar consoantes. A escolha entre esses dois componentes era feita por meio da posição da barra de pulso do operador. Além disso, o operador podia controlar a altura da voz sintética com o pedal de *pitch* (Figura 2.3) [2].

O que seria efetivamente transmitido a longas distâncias seriam apenas os envelopes que modulavam os tons vocálicos — isto é, a posição dos dez atuadores —, além da posição da barra de pulso e da posição do pedal de *pitch*. Assim, o *voder* implementava,

na prática, um método de compressão de sinal, pois havia uma economia no uso da largura de banda uma vez que a posição dos controles variava em taxas silábicas [2].

Foi demonstrado, diante do público, que era possível gerar vozes sintéticas inteligíveis de forma inteiramente analógica utilizando apenas esses controles. Ainda que a voz soasse robotizada — ou, como descreveu um correspondente do New York Times, com um “sotaque elétrico” —, o feito causou grande impacto. O *voder* foi apresentado na Feira Mundial de Nova York, em 1939: enquanto um locutor discursava e comentava sobre o circuito, o operador respondia às perguntas formando sons no *voder*, conectando-os em palavras e frases. Muitos espectadores expressaram surpresa e admiração diante da capacidade da máquina de “falar” [41].

O *voder*, por si só, não encontrou aplicação comercial, já que sua operação exigia um alto nível de habilidade por parte do operador. No entanto, serviu como prova de conceito para o *vocoder* de Dudley [22].

O *vocoder* de Dudley foi, na verdade, desenvolvido anteriormente ao *voder* [2]. O termo *vocoder*, cunhado para essa ocasião, deriva de “*voice codes*”, porque ele opera com base no princípio de derivar códigos de voz para recriar a fala que analisa [42]. Embora seu circuito seja semelhante ao do *voder*, o *vocoder* não exige a atuação de um operador humano, eliminando, assim, a dependência de usuários treinados e habilidosos para sua operação. Em vez disso, filtros dispostos na entrada do sinal de voz assumem a função do operador, permitindo a codificação automática por meio de correntes elétricas geradas nas saídas desses filtros [2].

O *vocoder* de Dudley encontrou aplicações práticas imediatas durante a Segunda Guerra Mundial. As forças armadas norte-americanas desconfiavam que seus sistemas de comunicação segura haviam sido comprometidos — uma suspeita que se confirmou posteriormente quando se descobriu que os alemães haviam decifrado o principal deles. Esse receio levou à falha na comunicação com as tropas norte-americanas estacionadas no Pacífico, que não foram alertadas a tempo sobre o ataque surpresa japonês ao Havaí [43, 44].

Esses acontecimentos motivaram o desenvolvimento do SIGSALY, um sistema de comunicação de voz seguro, criado pelos Laboratórios Bell Telephone com o apoio de cientistas proeminentes, como Alan Turing. O SIGSALY foi concebido para garantir comunicações seguras em escala global entre líderes militares e políticos dos Aliados. O desenvolvimento do SIGSALY constituiu um importante avanço nas técnicas modernas de telecomunicações, estabelecendo fundamentos que influenciaram diretamente tecnologias subsequentes na área. No seu núcleo, o sistema incorporava o *vocoder* de Dudley [45].

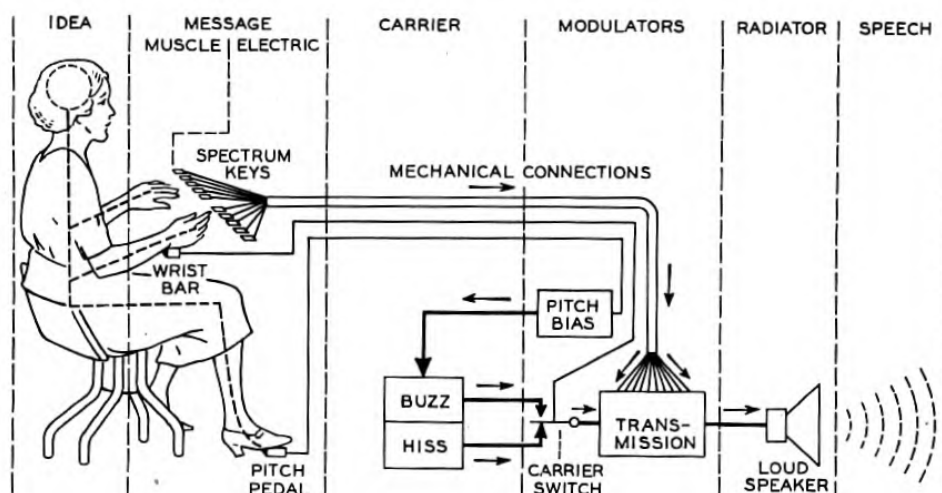


Figura 2.3: Circuito esquemático do voder de Dudley. Reproduzido de [2].

2.2.2 Outros *Vocoders* Analógicos

Apesar de o avanço tecnológico ter direcionado o mundo para soluções digitais, os *vocoders* analógicos continuaram a ser utilizados nas décadas que se seguiram à invenção de Homer Dudley, especialmente na produção musical [46]. Os progressos na eletrônica analógica forneceram a base necessária para o aprimoramento desses dispositivos.

Durante a década de 1960, os engenheiros exploraram novas formas de manipular o som da voz, transformando-a em paisagens sonoras analógicas. Já nos anos 1970 e nos anos seguintes, *vocoders* comerciais passaram a ser utilizados por bandas influentes, marcando presença em diversos gêneros musicais [47].

Os *vocoders* analógicos atuavam na extração do envelope das ondas sonoras em diferentes faixas de frequências, manipulação e posterior síntese, permitindo diversos efeitos sonoros [48].

2.3 *Vocoders* Digitais

No início da transição para o digital, engenheiros, também da Bell Telephone Laboratories, apresentaram em 1962 o primeiro modelo que produzia voz sintética em um computador. Esse modelo usava filtros digitais matemáticos que simulavam o som gerado pela passagem do ar pelas cordas vocais e pela ressonância nas cavidades bucal e nasal [49].

Os filtros representavam tubos acústicos interconectados com propriedades específicas, como comprimento ou área, por onde as ondas sonoras eram propagadas de diversos modos. A simulação era computacional e calculava como as ondas sonoras se transmitiam e refletiam dentro do trato vocal [50].

Ainda no mesmo laboratório, nesse mesmo período, outros modelos digitais matemáticos para a produção de voz foram desenvolvidos. Nesse ambiente o *phase vocoder* foi idealizado. No *phase vocoder*, os sinais de voz são decompostos usando Transformada de Fourier de Curto Termo (STFT), o resultado é modificado conforme necessidade, como, por exemplo, para alteração de *pitch*, e depois a voz é reconstruída usando STFT inversa [51].

Em 1967, uma nova abordagem foi criada: a síntese de fala por Codificação Preditiva Linear (*Linear Predictive Coding* — LPC). O LPC modela o sinal de fala assumindo que cada amostra pode ser prevista a partir de amostras passadas, segundo a Equação 2.3:

$$x[n] = \sum_{k=1}^p a_k x[n-k] + e[n], \quad (2.3)$$

onde $x[n]$ representa o sinal de fala discreto no instante n , a_k são os coeficientes de predição linear, k é o índice dos coeficientes, p é a ordem do modelo LPC e $e[n]$ representa o erro de predição, também denominado sinal residual. Durante a etapa de análise, amostras de voz são coletadas para estimar os coeficientes do modelo de forma que a energia do resíduo seja minimizada [52].

No processo de codificação, o sinal $x[n]$ é passado por um filtro que produz o resíduo $e[n]$. Esse filtro pode ser descrito pela Equação 2.4:

$$H_c(z) = 1 - \sum_{k=1}^p a_k z^{-k}, \quad (2.4)$$

onde $H_c(z)$ representa a função de transferência do filtro de análise e z é a variável complexa da Transformada Z.

Os coeficientes do modelo e uma representação compacta do resíduo (ou de uma aproximação deste) são então codificados e transmitidos, podendo essa representação ser obtida por técnicas como a codificação por *codebook* (CELP) [53].

No receptor, o processo de decodificação é realizado por um filtro que reconstrói o sinal $x[n]$ a partir de $e[n]$. Esse filtro é dado pela Equação 2.5:

$$H_d(z) = \frac{1}{1 - \sum_{k=1}^p a_k z^{-k}}. \quad (2.5)$$

Nessa expressão, $H_d(z)$ representa a função de transferência do filtro de síntese responsável pela reconstrução do sinal de fala.

Pelo fato de o LPC ser uma técnica capaz de analisar, codificar, transmitir e reconstruir a voz, ele é, essencialmente, um *vocoder* [54]. O LPC e suas variantes tornaram-se padrão nas telecomunicações, sendo incorporados a diversos codecs de compressão de voz utilizados em sistemas informáticos e teleinformáticos, como *voice-over-IP* (VoIP) e telefonia

celular [55, 56]. A tecnologia também encontrou espaço no mercado de consumo, integrando sintetizadores musicais, brinquedos falantes e outros dispositivos com capacidade de síntese de voz [57].

Outros *vocoders* digitais incluem o *vocoder* World e o *vocoder* Straight. Desenvolvidos por cientistas da Wakayama University, no início dos anos 2000, ambos os *vocoders* usam características extraídas do espectrograma Mel, incluindo frequência fundamental (F0) e aperiodicidades (ruídos, que representam as fricativas) para codificar e, posteriormente, sintetizar o áudio [58, 59].

As técnicas usadas na construção dos *vocoders* digitais não neurais desempenharam um papel importante no desenvolvimento histórico dos *vocoders*, ao introduzir conceitos de análise espectral e manipulação de frequência, que permitiram a compressão eficiente da voz e ajudaram a moldar a síntese de voz [60]. Embora os *vocoders* neurais atuais utilizem e superem essas abordagens, as inovações dos *vocoders* tradicionais abriram o caminho para o controle criativo da voz, tanto na comunicação segura quanto na música e telecomunicações.

2.4 *Vocoders* Neurais

A evolução dos *vocoders* neurais trouxe inovações na síntese de áudio, permitindo a reprodução da voz humana com qualidade e realismo [60]. Inicialmente, abordagens baseadas em redes neurais autoregressivas enfrentavam desafios como alto custo computacional e geração lenta de áudio [61, 62]. No entanto, novos paradigmas, como arquiteturas generativas adversariais e outras formas não-autoregressivas, melhoraram a qualidade e a velocidade da síntese, atendendo à demanda por soluções em assistentes virtuais e aplicativos móveis [5, 63]. Essa busca por eficiência consolidou os *vocoders* neurais como essenciais na tecnologia de síntese de voz [60].

De acordo com [64], os *vocoders* neurais podem ser classificados, didaticamente, quanto ao tipo de entrada, em acústicos ou linguísticos. Os *vocoders* linguísticos são aqueles que recebem entradas linguísticas diretamente, como fonemas, prosódia, pausa etc. Estão nessa categoria *vocoders* como WaveNet ou WaveRNN. Já os *vocoders* acústicos são *vocoders* que recebem entradas acústicas, como espectrogramas Mel. Nessa categoria encontramos, por exemplo, os *vocoders* WaveGlow, Parallel WaveGAN, MelGAN e variantes, DiffWave e HiFi-GAN.

2.4.1 WaveNet

Desenvolvido pela Google DeepMind¹, o WaveNet [61] foi o primeiro *vocoder* neural a demonstrar grande sucesso na síntese de áudio e é reconhecido por sua capacidade de gerar vozes com alta qualidade, nuances e realismo [61]. Lançado em 2016, o WaveNet representa um avanço em relação às técnicas anteriores de síntese de voz, utilizando uma arquitetura de rede neural autoregressiva.

Nesse modelo, esquematizado na Figura 2.4, a rede processa uma amostra de áudio por vez, a partir dos dados brutos do sinal. À esquerda da figura, observa-se o codificador, que extrai representações latentes contínuas do áudio. Essas representações são então quantizadas por meio de uma codificação vetorial, mapeadas para um Vector Quantization (VQ). À direita da mesma imagem, encontra-se o decodificador. Na etapa de decodificação, é utilizada uma rede autorregressiva e convolucional, que recebe o conteúdo latente e condições — isto é, informações adicionais que orientam o processo de reconstrução (como a identidade do locutor, sotaque, entre outras). Essa abordagem permite a geração de sequências sonoras que reproduzem a fluência e a expressividade da fala humana [61].

O WaveNet introduziu uma nova abordagem para a síntese de voz, empregando uma combinação de camadas convolucionais e técnicas de aprendizado profundo para capturar e reproduzir o som. Como resultado, o modelo é capaz de produzir padrões de áudio excelentes da fala humana, conforme pesquisa de opinião conduzida em [61].

No entanto, a natureza autoregressiva do modelo implica que a geração de áudio deve ser feita sequencialmente, o que pode ser bastante lento. Isso cria obstáculos para sua aplicação em tempo real em dispositivos com recursos limitados, como smartphones ou sistemas embarcados. As demandas computacionais necessárias para executar o WaveNet dificultaram a implementação de suas capacidades em cenários onde a velocidade de resposta é crítica, limitando sua adoção [62].

2.4.2 WaveGlow

O WaveGlow [4] é um modelo de *vocoder* desenvolvido pela NVIDIA Corporation², introduzido em 2018, que usa técnicas de normalização de fluxo para síntese de áudio de alta qualidade, sem necessidade de autorregressão.

O desenvolvimento do WaveGlow surgiu no contexto da demanda por *vocoders* capazes de produzir áudio de alta qualidade, superando as limitações de latência de modelos anteriores, como o WaveNet. Antes do WaveGlow, modelos como o WaveNet se destaca-

¹<https://deepmind.google/>

²<https://www.nvidia.com/>

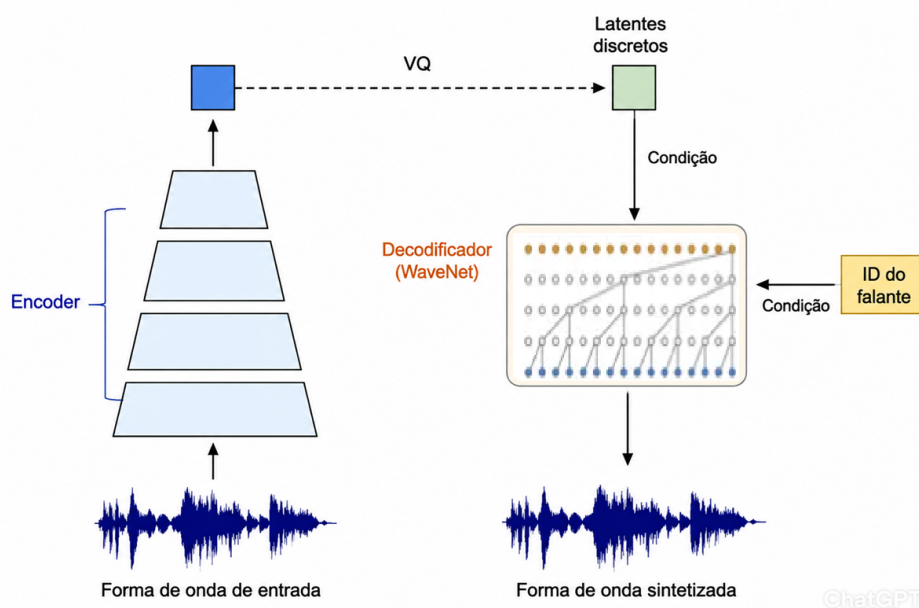


Figura 2.4: WaveNet: à esquerda, o codificador, que cria representações latentes discretas; à direita, o decodificador, que recebe o conteúdo latente e as condições para gerar o áudio sintético. Adaptado de [3].

vam pela qualidade do áudio, mas eram lentos, pois geravam amostras sequencialmente, tornando-os inadequados para aplicações em tempo real [65].

O WaveGlow, por sua vez, introduziu uma nova abordagem que utiliza uma arquitetura de normalização de fluxo. Isso permite a amostragem paralela de áudio, resultando em uma geração de áudio mais rápida e eficiente, apesar de qualidade ligeiramente inferior [65].

A arquitetura do WaveGlow pode ser observada na Figura 2.5. No bloco à direita, durante o treinamento, uma amostra de áudio é transformada em um fluxo normalizado — isto é, um vetor latente que segue uma distribuição normal. Já durante a inferência, o processo é invertido: um vetor amostrado dessa distribuição é convertido em áudio. Na parte esquerda, são mostrados os detalhes da *affine coupling layer*, na qual o espectrograma Mel é utilizado como entrada condicional para guiar a transformação. Nessa etapa, as características acústicas do sinal — como variações de amplitude, timbre, energia e estrutura harmônica — são associadas às características espectrais representadas pelo espectrograma Mel. Após o treinamento, o WaveGlow é capaz de gerar amostras de áudio de alta qualidade a partir de novos espectrogramas Mel de forma rápida e eficiente, graças à sua arquitetura totalmente paralela de amostragem [4].

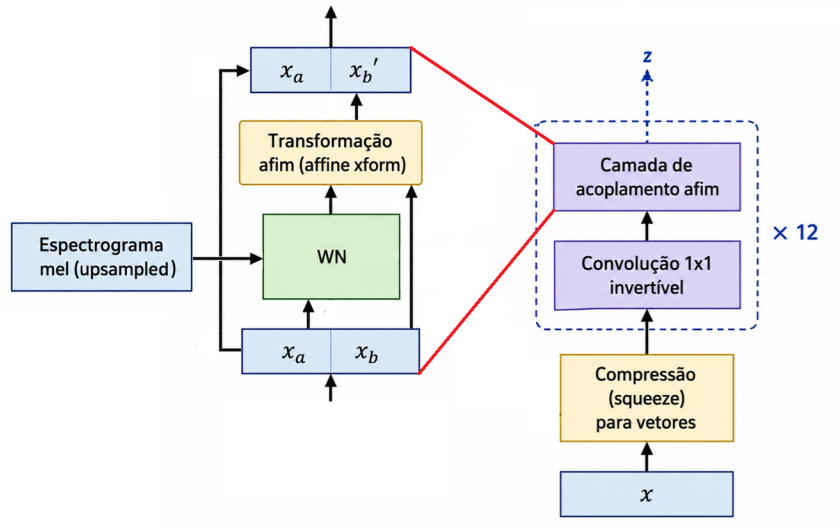


Figura 2.5: WaveGlow: à direita, o fluxo de treinamento e inferência; à esquerda, o detalhe da *affine coupling layer* condicionada pelo espectrograma Mel. Adaptado de [4].

2.4.3 WaveRNN

O WaveRNN [62], desenvolvido pela Google DeepMind³, representou uma evolução em relação ao seu predecessor, WaveNet, também da DeepMind, oferecendo uma solução mais eficiente para a geração de áudio.

Introduzido em 2018, o WaveRNN utiliza uma arquitetura de rede neural recorrente (RNN) que permite a geração sequencial de amostras de áudio, mas com uma abordagem que reduz o custo computacional associado, com uma qualidade que tange a do WaveNet, conforme pesquisa de opinião [62].

Essa inovação tornou o WaveRNN uma opção viável para aplicações práticas, superando uma das principais limitações do WaveNet, uma vez que consegue gerar amostras sequenciais até cinco vezes mais rápido que o tempo real, permitindo que ele fosse usado em dispositivos com recursos limitados, como smartphones e assistentes virtuais, expandindo sua aplicabilidade no mercado [66].

2.4.4 MelGAN

O MelGAN [5] é uma abordagem para síntese de áudio que utiliza redes generativas adversariais (GANs) e foi especificamente projetado para converter espectrogramas Mel em formas de onda realistas. Introduzido em 2019, por pesquisadores da Universidade de Montreal e pela Lyrebird AI⁴, o MelGAN se destaca por seu *design* leve e eficiente, per-

³<https://deepmind.google/>

⁴<https://www.descript.com/lyrebird>

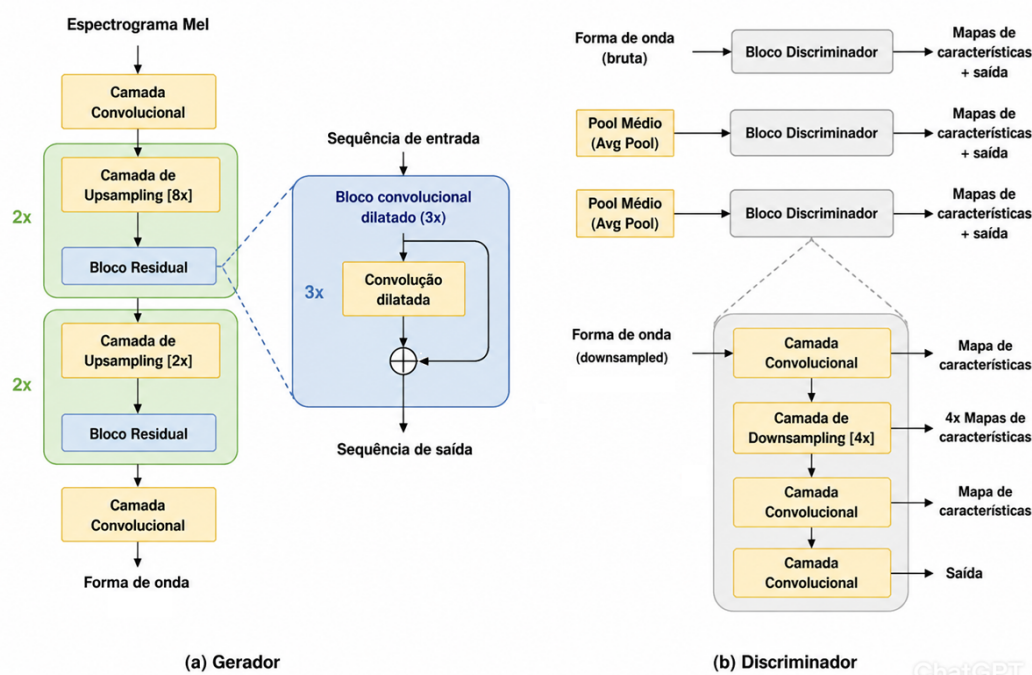


Figura 2.6: MelGAN: à esquerda, o gerador recebe espectrograma Mel para sintetizar a onda sonora; à direita, discriminadores são responsáveis por decidir se a onda sonora é real ou falsa. Adaptado de [5].

mitindo implementações em dispositivos móveis e computação de borda, onde os recursos computacionais são limitados.

Conforme se pode observar na Figura 2.6, à esquerda, o gerador recebe o espectrograma Mel como entrada para sintetizar a onda sonora bruta. À direita, três discriminadores operam em diferentes níveis de granularidade temporal: um analisa janelas curtas, outro avalia intervalos intermediários, e o terceiro observa segmentos longos. Durante o treinamento, estabelece-se um processo adversarial, no qual o gerador é penalizado sempre que falha em enganar os discriminadores, enquanto os discriminadores são penalizados quando não conseguem distinguir corretamente entre áudios reais e gerados. Essa interação incentiva o modelo a produzir amostras cada vez mais realistas.

Conforme explicado por [5], graças à sua arquitetura de processamento paralelizável, o MelGAN é capaz de sintetizar áudio rapidamente, cem vezes mais rápido que o tempo real, alcançando excelente qualidade sonora enquanto reduz a latência. Essas características o tornam adequado em aplicações que exigem resposta em tempo real, como sistemas *Text-to-Speech* (TTS) ou avatares de voz virtuais, onde a fluência e a naturalidade da fala são importantes.

2.4.5 Parallel WaveGAN

A Parallel WaveGAN [6] é uma abordagem na síntese de áudio que utiliza GANs de forma não autoregressiva. Introduzido em 2019 pela LINE Corp. e NAVER Corp⁵, a Parallel WaveGAN foi projetada para superar algumas das limitações associadas a modelos anteriores, como o WaveNet, que gera amostras sequencialmente.

Conforme [6], a Parallel WaveGAN adota uma arquitetura que permite a geração paralela de áudio, aumentando a velocidade de processamento, gerando amostras cerca de trinta vezes mais rápido que o tempo real. Quando comparada com a rede WaveNet, a Parallel WaveGAN é treinada em menos tempo e seus áudios possuem qualidade melhor, conforme pesquisa de opinião.

A arquitetura da Parallel WaveGAN baseia-se na interação entre duas redes neurais: uma geradora, inspirada no modelo WaveNet, e uma discriminadora, que são treinadas de forma adversarial (ver Figura 2.7). A rede geradora recebe como entrada um espectrograma Mel e um sinal de ruído aleatório (*random noise*), que fornece variabilidade e permite a síntese de amostras de áudio com características naturais e não determinísticas. A partir dessas entradas, o gerador produz amostras de fala sintética, enquanto o discriminador recebe tanto essas amostras geradas quanto amostras reais e deve decidir se cada uma é verdadeira ou falsa. As funções de *loss* são usadas para penalizar o discriminador caso ele não consiga distinguir corretamente entre sons reais e artificiais, ou para penalizar o gerador quando este não consegue enganar o discriminador. Essa interação adversarial, aliada ao uso do ruído aleatório na entrada, possibilita ajustes dinâmicos e promove uma síntese de áudio mais eficiente, estável e realista [6].

2.4.6 DiffWave

O DiffWave [7] é um *vocoder* neural baseado em modelos de difusão, desenvolvido em 2020, conjuntamente por pesquisadores da Universidade da Califórnia em San Diego, Baidu Research⁶ e NVIDIA⁷, para sintetizar áudio a partir de espectrogramas Mel com alta qualidade. Inspirado em técnicas de difusão popularizadas na geração de imagens, o DiffWave realiza a síntese de áudio de forma gradual e reversa, começando com ruído e refinando-o progressivamente para criar uma forma de onda natural.

De acordo com [7], a vantagem do DiffWave está na qualidade do áudio gerado, que supera muitos *vocoders* convencionais, como o WaveNet e outros modelos baseados em GAN. O modelo utiliza uma rede neural convolucional para prever e remover ruídos em diferentes escalas temporais, permitindo uma síntese flexível. Além disso, o treinamento

⁵<https://www.navercorp.com/>

⁶<https://research.baidu.com/>

⁷<https://www.nvidia.com/>

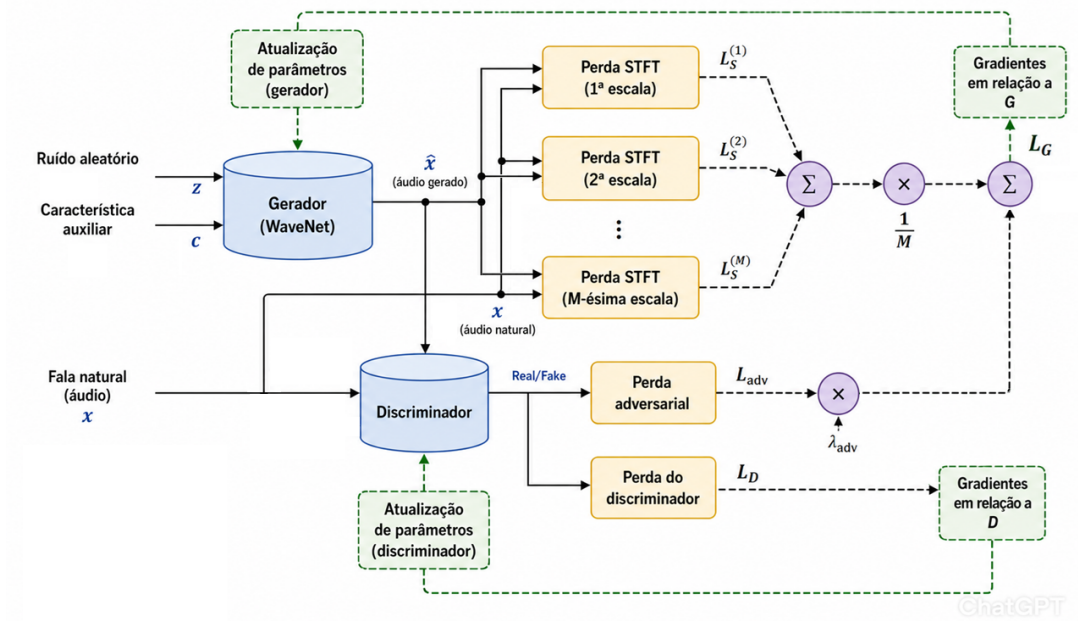


Figura 2.7: Parallel WaveGAN: o discriminador recebe falas reais e falas geradas pelo gerador e deve decidir se são verdadeiras ou falsas. Adaptado de [6].

do DiffWave é relativamente simples, pois depende de uma função de perda que mede o erro de previsão do ruído.

A Figura 2.8 apresenta o modelo DiffWave, que realiza um processo de difusão composto por duas etapas complementares: a direta e a reversa. Na fase direta, correspondente ao treinamento, adiciona-se ruído ao sinal de áudio de forma progressiva até que ele se torne indistinguível. Já na fase reversa, que ocorre durante a inferência, o modelo executa o processo inverso, removendo gradualmente o ruído para reconstruir um áudio inteligível a partir de um sinal inicialmente ruidoso.

O DiffWave ainda enfrenta alguns desafios, especialmente em relação aos tempos de treinamento e de inferência. Como o processo de difusão é iterativo, ele é mais lento do que *vocoders* não iterativos, pois requer várias etapas para gerar uma amostra de áudio completa [67]. No entanto, esforços de pesquisa, como a variante FastDiff, foram desenvolvidos para reduzir o número de etapas e otimizar a eficiência do modelo [68].

2.4.7 HiFi-GAN

O HiFi-GAN [69] é um modelo que melhora a qualidade e eficiência dos *vocoders* baseados em GANs. Desenvolvido em 2020, pela Kakao Enterprise⁸, este modelo fornece síntese de áudio de alta fidelidade a partir de espectrogramas Mel, em uma taxa quase 170 vezes

⁸<https://kakaointerprise.com/>

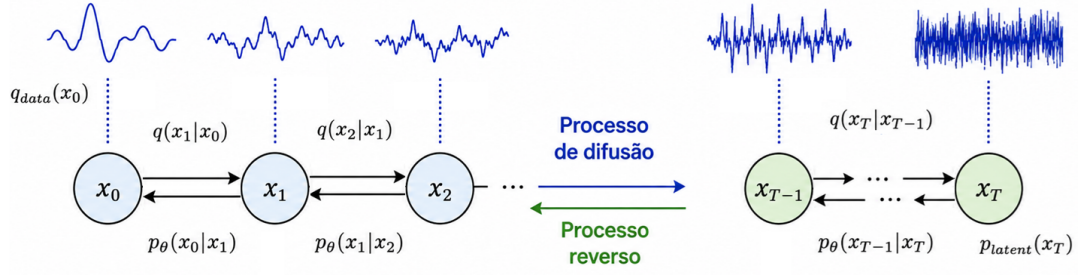


Figura 2.8: DiffWave: mostra o processo de difusão direto, em direção à direita, e reverso, em direção à esquerda. Adaptado de [7].

maior que a do tempo real. Essa combinação de alta qualidade de áudio e eficiência é um dos aspectos que o torna uma escolha em muitas aplicações de síntese de fala [70].

Apesar de tanto o MelGAN quanto o HiFi-GAN utilizarem múltiplos discriminadores para melhorar a qualidade do áudio sintetizado, o HiFi-GAN se destaca pela diversidade desses discriminadores. Ele acrescenta mais dois tipos de discriminadores: o Multi-Period Discriminator (MPD) e o Multi-Scale Discriminator (MSD), que avaliam o sinal sob diferentes perspectivas, incluindo padrões periódicos e resoluções temporais variadas. Essa combinação permite ao HiFi-GAN capturar de forma mais precisa os detalhes acústicos do áudio natural, alcançando uma fidelidade perceptual maior, conforme pesquisa de opinião conduzida pelos proponentes desse modelo [69].

2.4.8 StyleMelGAN

O StyleMelGAN [8] é um *vocoder* de pouca complexidade computacional, condicionado em espectrograma Mel, que se destaca na síntese de áudio devido à sua capacidade de gerar sons realistas e expressivos, em taxas muito maiores que a do tempo real, utilizando uma arquitetura de GANs. Lançado em 2021, pelo Instituto Fraunhofer de Circuitos Integrados⁹, o StyleMelGAN foi desenvolvido como uma extensão do MelGAN, com foco na incorporação de diversos estilos vocais na síntese de áudio, tornando-o particularmente útil para aplicações onde a expressividade e a personalização da voz são desejadas, como em sistemas Text-to-Speech (TTS) e avatares virtuais.

De acordo com [8], o StyleMelGAN consegue aprender representações de estilo a partir de dados de treinamento, permitindo que o modelo capture variações vocais que ocorrem devido a diferentes emoções ou traços pessoais. Essa flexibilidade é alcançada por meio da manipulação de entradas que definem os estilos de áudio.

Assim como seus predecessores, o StyleMelGAN utiliza uma abordagem não autoregressiva, o que significa que ele pode gerar áudio rapidamente ao processar amostras em

⁹<https://www.iis.fraunhofer.de/>

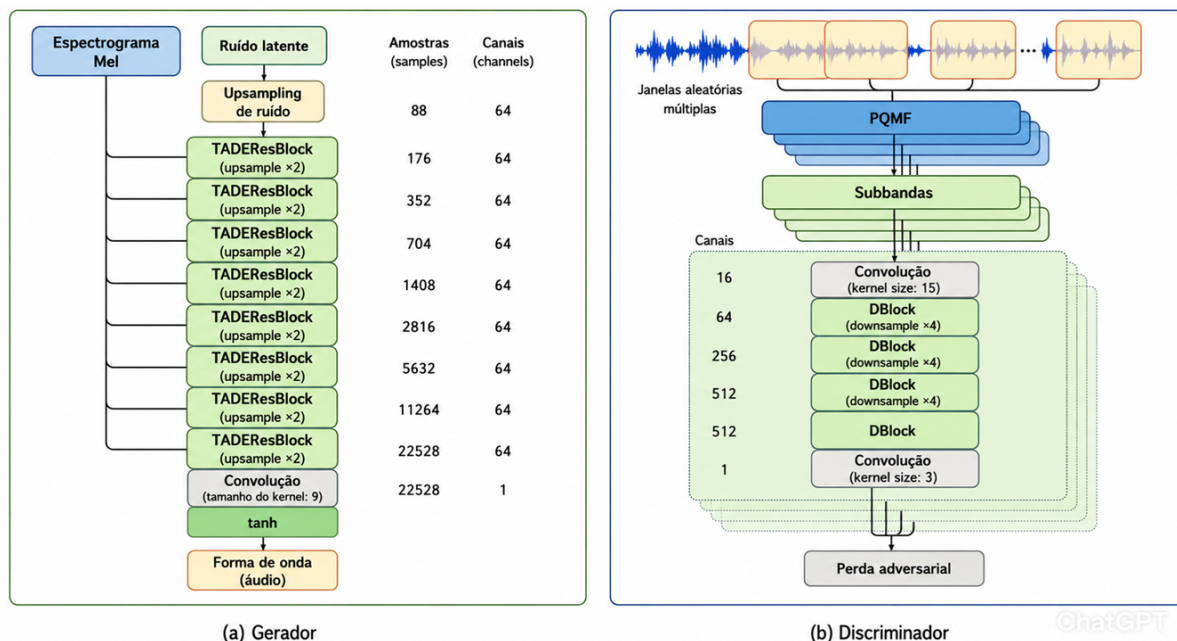


Figura 2.9: StyleMelGAN: à esquerda, arquitetura do gerador; à direita, arquitetura dos discriminadores. Adaptado de [8].

paralelo [8]. Isso contribui para uma baixa latência, tornando o modelo adequado para aplicações em tempo real.

A Figura 2.9 apresenta a arquitetura do StyleMelGAN. À esquerda, observa-se o gerador, responsável pela síntese da fala. Ele recebe como entrada um vetor de ruído aleatório e, simultaneamente, os espectrogramas Mel, que fornecem informações de conteúdo e prosódia para o condicionamento do sinal gerado. À direita, encontra-se a arquitetura dos discriminadores, nos quais o sinal de áudio é dividido em múltiplas janelas aleatórias e, em seguida, decomposto em diversas sub-bandas. Essa decomposição reduz a complexidade do processamento e permite uma análise mais detalhada por faixa de frequência. Por fim, o resultado é processado por uma rede convolucional, que, com base na função de perda adversarial, determina se os áudios apresentados são reais ou sintetizados [8].

2.4.9 Multi-Band MelGAN

Proposto em 2021, pela Universidade Politécnica do Noroeste (China), o Multi-Band MelGAN [9] foi projetado para melhorar a qualidade da síntese de áudio em aplicações de Text-to-Speech (TTS) e outras áreas que exigem som realista e expressivo. Este modelo utiliza uma abordagem multibandas que separa o espectro de áudio em várias bandas de frequência, permitindo um melhor controle sobre a geração de som.

A ideia de dividir o espectro de áudio em múltiplas bandas não é nova, mas o Multi-Band MelGAN ao integrá-las em um framework de GANs, permitiu que cada banda de

frequência fosse processada separadamente. O modelo utiliza uma rede geradora composta por várias camadas operando em diferentes bandas, resultando em uma síntese de áudio que não apenas é sete vezes mais rápida, mas também mantém uma qualidade superior, quando comparada à predecessora MelGAN, conforme pesquisado por [71].

A Figura 2.10 apresenta a arquitetura do Multi-Band MelGAN. No bloco superior, o gerador recebe o espectrograma Mel e gera sinais em múltiplas sub-bandas, que em seguida são recombinaados por um banco de filtros de síntese para formar a onda no tempo. Em paralelo, o áudio real é decomposto por um banco de filtros de análise nas mesmas sub-bandas. O sinal (real ou sintetizado) é então encaminhado aos múltiplos discriminadores do bloco inferior, que operam em diferentes resoluções temporais e, conjuntamente, determinam se o áudio é real ou sintético [9].

2.4.10 BigVGAN

O BigVGAN é um *vocoder* neural baseado em redes adversariais generativas, projetado para sintetizar sinais de áudio a partir de representações tempo-frequência, como espectrogramas Mel. Foi proposto em 2023, pela Seoul National University (SNU) e pela NVIDIA. O modelo aplica filtragem anti-aliasing durante a reamostragem para reduzir artefatos e ruídos artificiais no áudio sintetizado. Além disso, utiliza funções de ativação com componentes periódicas, o que facilita a representação de padrões harmônicos no sinal de voz e contribui para a reconstrução do conteúdo espectral e temporal [10].

O treinamento segue o paradigma adversarial, com um gerador e discriminadores que operam em múltiplas escalas temporais e padrões periódicos. Em particular, o modelo emprega discriminadores do tipo multi-resolution discriminator (MRD) e multi-period discriminator (MPD) - já utilizado no HiFi-GAN. O MRD analisa o sinal no domínio tempo-frequência, a partir de espectrogramas com diferentes resoluções. Já o MPD reorganiza o sinal no domínio temporal em uma matriz 2D para capturar padrões periódicos associados ao *pitch* e à estrutura harmônica. A rede usa ambos os discriminadores para decidir se o áudio é verdadeiro ou falso [10].

2.4.11 Vocos

O *vocoder* neural Vocos é um modelo de síntese de áudio, proposto em 2024, pela Gemelo AI, projetado para gerar formas de onda a partir de espectrogramas Mel. Embora diversos *vocoders* contemporâneos também utilizem representações de frequência como entrada e sejam treinados com redes adversariais, o principal diferencial do Vocos está no fato de que seu gerador neural estima diretamente coeficientes espectrais complexos no domínio de Fourier, em vez de estimar a forma de onda no domínio temporal. O sinal de áudio é

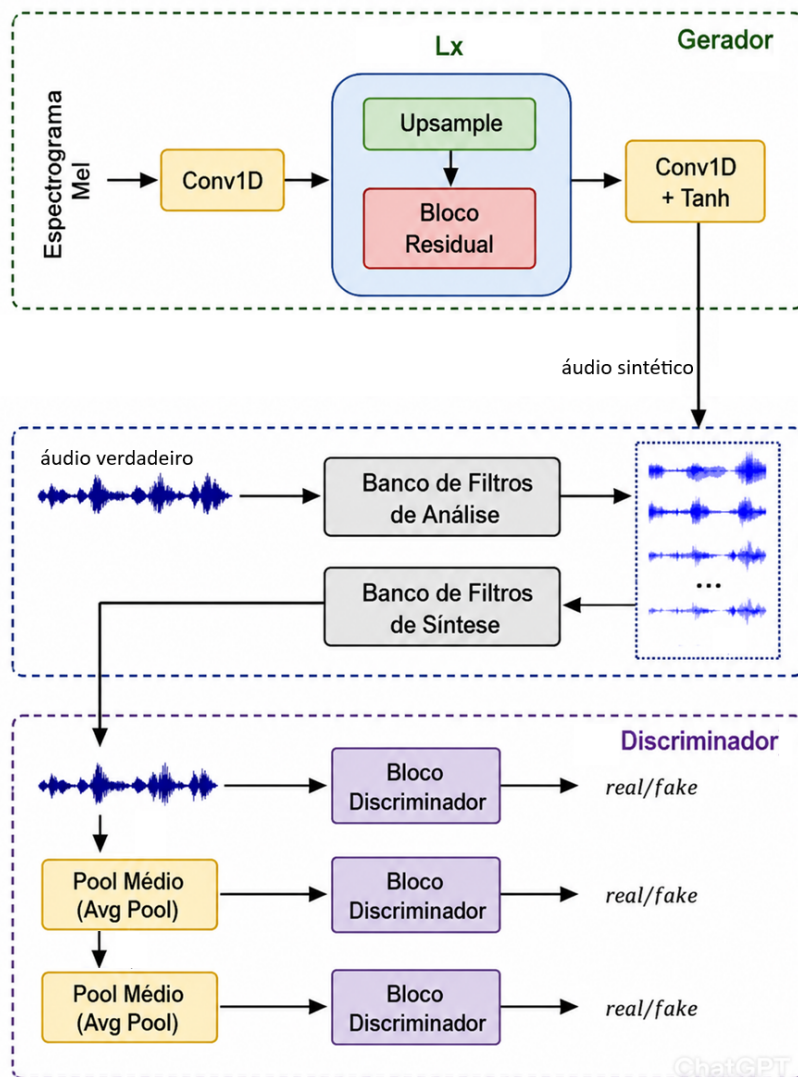


Figura 2.10: Multi-Band MelGAN: no bloco de cima, o gerador; no bloco de baixo, três discriminadores. Adaptado de [9].

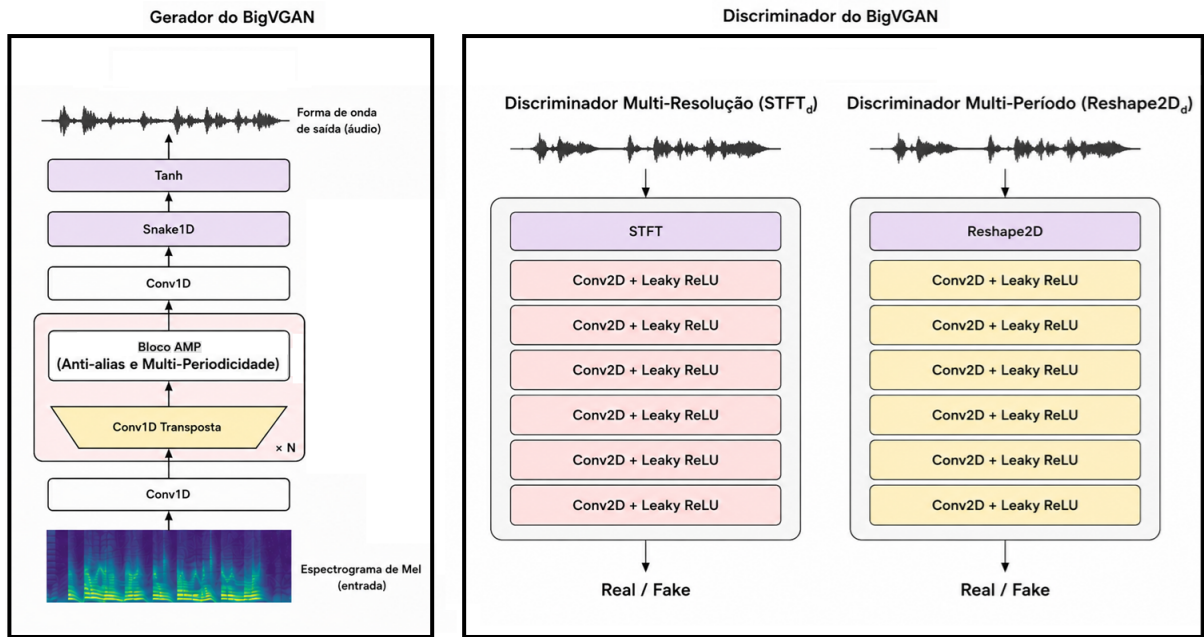


Figura 2.11: BigVGAN: à esquerda, arquitetura do gerador; à direita, arquitetura dos discriminadores. Adaptado de [10].

então reconstruído por meio de transformada inversa de Fourier, explorando propriedades estruturais do espectro [11].

Do ponto de vista metodológico, essa formulação permite resolução temporal constante ao longo de toda a rede - em contraste com outras que dependem de camadas de *upsampling* para converter representações comprimidas em sinais de alta frequência de amostragem - e simplifica o processo de síntese, resultando em menor custo computacional em comparação a arquiteturas baseadas na geração direta de amostras. O treinamento adversarial é empregado, alinhando-se a práticas já consolidadas em *vocoders* neurais recentes, porém aplicado a um espaço de saída espectral em vez de temporal. Avaliações experimentais relatam que o modelo atinge qualidade perceptual equivalente ao estado da arte, tornando-o adequado para aplicações em tempo real e sistemas embarcados [11].

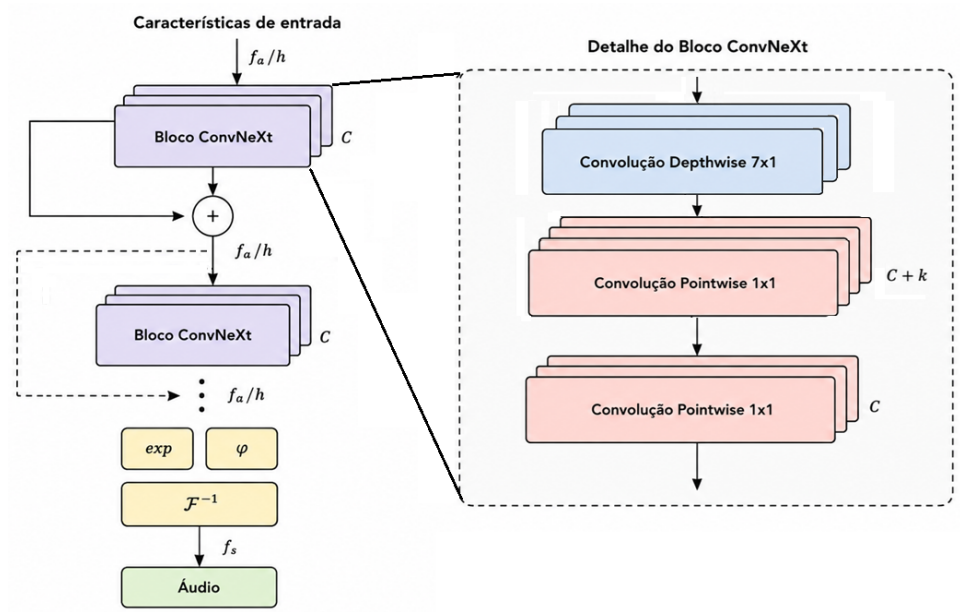


Figura 2.12: Arquitetura do Vocos. Os *features* de entrada (espectrogramas Mel) são processados por uma pilha de blocos convolucionais. A rede produz parâmetros espectrais que são transformados em magnitude (exp) e fase (φ), e então convertidos em áudio no domínio do tempo por meio da Transformada Inversa de Fourier de Tempo Curto (*Inverse Short-Time Fourier Transform* — ISTFT), resultando no sinal final. Adaptado de [11].

Capítulo 3

Trabalhos Relacionados

Nos últimos anos, detectores de *deepfake* de áudio têm explorado tanto representações tradicionais do sinal — como espectrogramas e coeficientes cepstrais — quanto abordagens baseadas em redes neurais profundas. Além disso, o avanço da área tem sido fortemente impulsionado por competições (*challenges*) internacionais, que fornecem *benchmarks* padronizados, conjuntos de dados amplos e métricas de avaliação padronizadas.

Essas competições incluem diferentes cenários, como detecção de áudios totalmente sintéticos, detecção de manipulações parciais e identificação de *vocoders*, permitindo comparar métodos de forma consistente e fomentar o desenvolvimento de técnicas cada vez mais robustas.

A seguir, apresentamos as principais técnicas e estudos recentes sobre a detecção de áudios sintéticos, bem como serão mostradas as principais competições da área de detecção de *deepfake* de áudio.

3.1 Detectores de *deepfake* de áudio

Em 2015, [72] propôs um sistema, não baseado em rede neural para a detecção de fala natural em oposição à fala falsificada, onde faz uma fusão de escores baseados nos já tradicionais *Mel-Frequency Cepstral Coefficients* (MFCC) e nos *Cochlear Filter Cepstral Coefficients* combinados com *Instantaneous Frequency* (CFCC-IF), que são utilizados para classificação por meio de um modelo estatístico de *Gaussian Mixture Model* (GMM) com 128 misturas. Obteve um *Equal Error Rate* (EER) médio de 1.211% no conjunto de testes do desafio ASVspoof 2015.

Em [73], publicado em 2018, foi proposta uma abordagem baseada em uma CNN rasa (*Shallow CNN*) para distinguir fala humana de fala sintética. A entrada do modelo é um

espectrograma 2D (clássico ou Mel), gerado a partir de amostras dos datasets LibriSpeech ASR [74] e CMU ARCTIC [75]. O trabalho também utilizou vozes sintéticas obtidas a partir de serviços TTS comerciais (Google Cloud TTS, Amazon Polly, Microsoft Azure TTS e IBM Watson TTS). Os autores reportaram que espectrogramas Mel apresentam desempenho superior aos clássicos, alcançando acurácia acima de 90% para *vocoders* já vistos e 70% para *vocoders* inéditos.

Em 2019, o Speech Technology Center (STC) apresentou o sistema *STC Antispoofing* [76] para o ASVspoof 2019 Challenge [77]. A tecnologia submetida utiliza uma arquitetura *Light Convolutional Neural Network* (*Light CNN*), que emprega a função de ativação *Max-Feature-Map* (MFM), baseada na função *Max-Out activation* [78], além de utilizar *Batch Normalization*. Este sistema representa uma versão aprimorada da arquitetura *Enhanced Light CNN*, originalmente proposta pelos mesmos autores no ASVspoof 2017 Challenge (ASV Spoofing and Countermeasures Initiative) [79]. O modelo foi testado com diferentes entradas acústicas, incluindo Transformada *Constant-Q* (CQT), *Fast Fourier Transform* (FFT) e *Discrete Cosine Transform* (DCT), diferindo do trabalho anterior dessa equipe, em que o sistema proposto utilizava extratores de características de alto nível, como *embeddings*. A CQT apresenta características semelhantes à escala Mel, porém com um mapeamento totalmente logarítmico, ao invés de perceptual [80]. O treinamento foi realizado com a função de perda *Angular Margin-based Softmax* (A-Softmax). O conjunto de dados empregado foi o ASVspoof 2019, e, no cenário *Logical Access* (LA), o melhor sistema (uma fusão de modelos com diferentes entradas acústicas) obteve uma *Equal Error Rate* (EER) de 1,86%. Além disso, os experimentos indicaram que o uso de *Speech Activity Detection* (SAD) poderia prejudicar o desempenho do sistema, uma vez que partes não faladas do sinal também contêm informações úteis. Por essa razão, o SAD não foi empregado.

Em 2020, [81] apresentou a RawNet2, uma arquitetura de rede neural profunda projetada para reconhecimento automático de locutor e detecção de falsificações de voz, tendo como principal inovação o uso direto da forma de onda bruta como entrada, sem necessidade de extração prévia de características acústicas como MFCC ou espectrogramas. A rede é composta por blocos convolucionais residuais que permitem a aprendizagem *end-to-end* de representações robustas e discriminativas do sinal de áudio. No desafio ASVspoof 2019, a RawNet2 alcançou uma EER de 1,06% no cenário *Logical Access*, superando abordagens baseadas em espectros tradicionais.

Em 2021, [82] empregou uma versão modificada da RawNet2 [83] em um sistema de

detecção de *spoofing* de ponta a ponta, com o objetivo de classificar sinais de áudio como autênticos ou falsificados. A RawNet2 é uma rede neural profunda que opera diretamente sobre formas de onda de áudio (*raw audio*) e foi originalmente proposta para a tarefa de verificação de locutor independente de texto, que não exige que o conteúdo verificado do falante seja igual ao previamente cadastrado. A arquitetura modificada obteve o segundo lugar no cenário *Logical Access* da competição ASVspoof 2019.

O trabalho apresentado em [26] propõe um classificador binário para detecção de fala sintética, cuja função é distinguir entre amostras *bonafide*, isto é, legítimas, e sintéticas. O estudo explora dois cenários distintos:

- *Closed-set*: o classificador detecta se o áudio é *bonafide* ou sintético, assumindo que todos os algoritmos de síntese de áudio pertencem a um conjunto conhecido.
- *Open-set*: além de identificar se o áudio é *bonafide*, o sistema tenta identificar se foi gerado por um algoritmo de síntese conhecido ou por um método nunca antes visto.

Os experimentos foram realizados com o conjunto de dados ASVspoof 2019 [77]. O detector proposto neste trabalho opera diretamente no domínio temporal, com base na premissa de que a fala contém dependências de longo prazo (memória), e que tal característica é essencial para distinguir sinais falsos de verdadeiros. Os autores lembram que abordagens tradicionais geralmente extraem características acústicas (*acoustic features*) significativas de forma explícita, como espectrogramas Mel. Contudo, sustentam que características de longo prazo, como *Constant Q Cepstral Coefficients* (CQCC) [84], espectro de magnitude logarítmica ou características baseadas em fase (como *group delay*), devem ser preferidas em relação às abordagens de curto prazo. Além disso, é argumentado que os traços introduzidos pelos algoritmos de síntese estão distribuídos de forma desigual entre diferentes bandas de frequência. Por esse motivo, a análise por sub-bandas foi explorada utilizando características acústicas como os *Linear-Frequency Cepstral Coefficients* (LFCC) e os *Mel-Frequency Cepstral Coefficients* (MFCC). Tais características alimentam classificadores supervisionados, usualmente em GMMs. Na metodologia proposta, são definidos conjuntos de descritores de áudio extraídos a partir da análise combinada de curto e longo prazo, permitindo a extração de informações relevantes, como padrões estatísticos e temporais, ao estudar a relação entre o passado e o presente do sinal de fala. A entrada da rede consiste em características estatísticas extraídas do áudio, obtidas a partir da filtragem do sinal com preditores de curto e longo prazo. O treinamento da rede é realizado de forma supervisionada.

O trabalho apresentado em [85] propõe o *Artifact-Aware Spoofing Detection System* (AASIST), um sistema para detecção de *spoofing* baseado em redes neurais de atenção em grafos. O trabalho em questão argumenta que os artefatos introduzidos por algoritmos de síntese e conversão de voz podem se manifestar tanto no domínio temporal quanto no espectral. Sistemas antigos tendiam a depender de múltiplos subsistemas especializados para cada tipo específico de artefato, o que, segundo os autores, limitava a generalização. Nesse contexto, o AASIST é proposto como uma única arquitetura capaz de capturar artefatos de diferentes naturezas, tornando-se, assim, um modelo unificado para detecção de falsificações em voz.

A arquitetura é baseada em *Graph Attention Networks* (GATs) e foca na tarefa de detecção de ataques de *Logical Access* (LA) da competição ASVspoof 2019. O sistema utiliza uma variante do RawNet2 para extração de representações de alto nível a partir do sinal de áudio. Essas representações são tratadas como nós de um grafo, sobre os quais são aplicadas operações de atenção com o objetivo de capturar relações estruturais relevantes para a detecção de falsificações.

Um dos principais componentes introduzidos é a camada *Heterogeneous Stacking Graph Attention Layer* (HS-GAL), que realiza operações de atenção heterogênea sobre os nós. A heterogeneidade se refere à incorporação de características provenientes tanto do domínio do tempo quanto do domínio da frequência, ampliando assim a capacidade de modelar diferentes tipos de artefatos.

O AASIST obteve uma EER de apenas 0,83% no conjunto de testes do ASVspoof 2019 (cenário *Logical Access*), enquanto uma versão mais leve do modelo apresentou EER de 0,99%.

O trabalho apresentado em [86] no ano de 2022 propõe uma abordagem tempo-frequência para a detecção de fala sintética, com foco na captura de artefatos que ocorrem em relações de longo prazo e em componentes de alta frequência.

Os autores de [86] argumentam que tais relações são essenciais para identificar sinais gerados artificialmente, mas que transformadas baseadas DFT são ineficazes nesse aspecto, por não permitirem uma resolução adequada em escalas de tempo mais longas. Ainda, observam que a CQT, apesar de possibilitar o uso de janelas maiores, tem baixa sensibilidade a componentes de alta frequência, o que limita sua aplicação nesse contexto.

Como solução, é proposta a transformada *Long-term Variable Q Transform* (L-VQT), uma técnica de análise tempo-frequência que adapta o tamanho da janela de análise conforme a frequência: janelas mais longas para baixas frequências e mais curtas para altas frequências. Essa variação permite capturar tanto padrões de longo prazo quanto detalhes em regiões espectrais elevadas, com a finalidade de detectar artefatos introduzidos

por sistemas de síntese de voz.

As representações geradas pela L-VQT são utilizadas como entrada para uma rede neural convolucional denominada L-DenseNet, uma versão simplificada da DenseNet original, projetada especificamente para a tarefa de classificação binária entre fala *bonafide* e sintética.

Os experimentos foram conduzidos sobre o conjunto de dados ASVspoof 2019 (cenário *Logical Access*) e os autores relatam desempenho robusto mesmo em condições de ruído, sugerindo a eficácia da abordagem proposta em cenários realistas.

A competição Audio Deepfake Detection 2022 (ADD 2022) introduziu a tarefa de detectar áudios parcialmente falsos, nos quais a maior parte do áudio é genuína, mas determinados trechos são sintéticos. Para lidar com este problema, foi proposto um método em [87], onde os autores propõem um modelo baseado em uma estratégia de Perguntas e Respostas (*Question-Answering* — QA), utilizando um mecanismo de *self-attention* para prever o início e o fim do trecho falso, respondendo à pergunta: “*Onde está o trecho falso?*”. Como entrada, o modelo utiliza espectrograma Mel combinado com outras características acústicas, como MFCC e LFCC, para aumentar a diversidade no estágio de fusão. Este sistema obteve o segundo lugar na competição ADD 2022, com uma EER de 7.9% na Trilha 2 (*partially fake audio detection*).

Ainda em 2022, foi apresentado em [88] um método para detecção de *deepfake* de áudio que emprega o wav2vec 2.0 como extrator pré-treinado de características, em vez das tradicionais características acústicas, proporcionando uma representação de alto nível da fala. O wav2vec 2.0 [89] é um extrator de características desenvolvido pelo Facebook AI (atual Meta AI¹), que produz vetores numéricos (*embeddings*), utilizado originalmente para tarefas de extração de reconhecimento automático de fala (*Automatic Speech Recognition* — ASR).

Para aprimorar o processo de aprendizado, o trabalho propõe o Light-DARTS, uma versão simplificada da técnica *Differentiable Architecture Search* (DARTS) [90], que, por sua vez, é um método de escolha de melhor arquitetura de rede neural, com o objetivo de aprender automaticamente as melhores representações das características wav2vec. O Light-DARTS incorpora a técnica *Max Feature Map* (MFM) [78] como função de ativação. A MFM atua como um seletor das características mais relevantes entre diferentes canais da rede neural convolucional, permitindo a redução do número de canais na camada subsequente. O método utilizou o conjunto de dados de ASVspoof 2019 no cenário *Logical Access* e alcançou uma EER de 1.08% [88].

¹<https://ai.meta.com/>

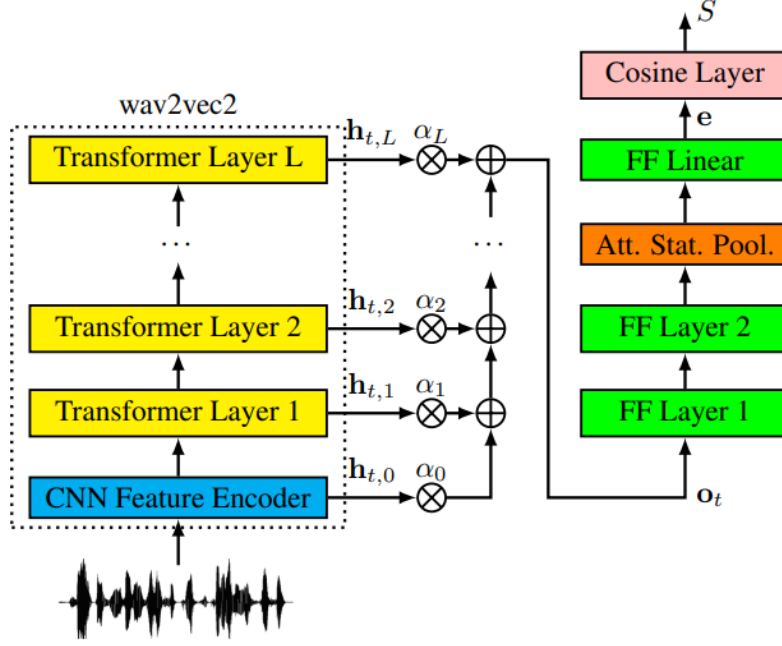


Figura 3.1: Arquitetura de detecção de *deepfake* de áudio proposta por *The Vicomtech Audio Deepfake Detection System based on Wav2Vec2 for the 2022 ADD Challenge*. Múltiplas camadas de wav2vec2 são combinadas antes de ser entregues ao classificador neural. Reproduzido de [12].

No trabalho de [12], o sistema proposto utiliza camadas pré-treinadas do extrator de características wav2vec2, combinadas a um modelo de classificação subsequente para determinar se um sinal de áudio é sintético. Nesta abordagem, múltiplas camadas do wav2vec2 são integradas por meio de pesos treináveis, que são otimizados juntamente com o classificador. A etapa de classificação é treinada com a função de perda *one-class softmax* [91], visando atribuir escores mais elevados para fala genuína (Figura 3.1). O sistema alcançou o primeiro lugar na Trilha 1 da competição ADD 2022 [92], que utilizou um conjunto de dados baseado no corpus AISHELL [93], obtendo uma EER de 21,7%.

Em 2023 foi publicado o trabalho de [94]. Este trabalho propõe um método para a detecção de vozes humanas sintéticas, explorando artefatos introduzidos pelos *vocoders* neurais no processo de geração da fala. O modelo utiliza o RawNet2 como extrator de características, argumentando que essa abordagem supera métodos que realizam pré-processamento de características acústicas, como espectrogramas Mel ou LFCC.

Para os experimentos, foi utilizado o conjunto de dados LibriSeVoc [94], derivado do corpus LibriTTS [95], criado especialmente para este estudo, com gravações sintéticas geradas por seis *vocoders* diferentes, além do WaveFake [1]. Os autores destacam que,

embora tecnologias de síntese de voz sejam legítimas e úteis — por exemplo, em assistentes virtuais ou para auxiliar pessoas que perderam a capacidade de fala — elas também podem ser usadas de forma maliciosa, como por golpistas que imitam vozes.

A arquitetura proposta adota uma estratégia multi-tarefa, em que o RawNet2 atua como *front-end*, enquanto um *backbone* realiza simultaneamente:

1. a classificação do áudio em genuíno ou falso; e
2. a identificação do *vocoder* responsável pela geração da voz sintética (quando falso).

Nos experimentos, o modelo obteve uma EER de 0.13% no LibriSeVoc e 0.19% no WaveFake, embora os autores ressaltem que ainda há espaço para melhorias futuras.

O trabalho de [96] discute a necessidade de que algoritmos de detecção de *deepfake* possuam a capacidade não apenas de identificar o algoritmo utilizado na síntese de fala, mas também de reconhecer algoritmos *Out-Of-Distribution* (OOD), ou seja, que não participaram do processo de treinamento. Essa habilidade é essencial para ampliar a robustez dos sistemas de detecção frente a novos *vocoders* que possam surgir após o desenvolvimento do modelo.

Esse estudo menciona que algoritmos como o *k-Nearest Neighbors* (kNN), que agrupam *embeddings*, já foram empregados nessa tarefa, mas, neste trabalho, a distância kNN é explorada para construir um módulo de calibração de classe. Nesse contexto, calcula-se a distância entre *embeddings* para determinar se dois áudios foram gerados pelo mesmo *vocoder*. Além disso, o método incorpora diferentes estratégias de fusão, visando melhorar a precisão e a generalização do sistema. Entre os extratores de *embeddings* avaliados, o melhor desempenho foi obtido com o extrator ECAPA-TDNN, uma arquitetura concebida originalmente para verificação de locutor [97]. Com essa abordagem, o sistema alcançou o primeiro lugar na Trilha 3 da competição ADD 2023 [98].

O trabalho de [99], desenvolvido em 2024, propõe um novo método para a detecção de *spoofing* em sistemas de verificação de locutor, explorando representações ocultas de modelos de fala auto-supervisionados. O método, denominado *Attentive Merging* (AttM), realiza a fusão de *embeddings* extraídos de múltiplas camadas do modelo WavLM [100] — modelo de representação de fala pré-treinado de forma auto-supervisionada desenvolvido pela Microsoft Research² —, utilizando mecanismos de atenção para combinar informações provenientes de diferentes níveis de abstração.

O WavLM é um modelo treinado diretamente sobre formas de onda de áudio bruto [100]. Esse tipo de modelo tem se mostrado eficaz na extração de representações foneticamente

²<https://www.microsoft.com/en-us/research/>

e semanticamente ricas. Em contraste com abordagens tradicionais baseadas em características acústicas como MFCC ou LFCC, que podem falhar em capturar informações discriminativas suficientes e deixam os sistemas mais vulneráveis a novos métodos de síntese de voz, os modelos SSL têm demonstrado maior robustez frente a ataques sofisticados.

A abordagem AttM utiliza atenção para ponderar e integrar as representações de diferentes camadas dos codificadores do *transformer* no WavLM, resultando em um vetor de características mais expressivo. Esse vetor é, então, utilizado como entrada para uma rede classificadora treinada para a tarefa de detecção de *spoofing*.

Os autores demonstram que o método apresenta forte capacidade de generalização e alcança resultados de estado da arte no conjunto de dados ASVspoof 2021 (subconjunto *Speech Deepfake*), atingindo uma EER de 3,19%.

Com foco na tarefa de detecção de *vocoders* fora da distribuição (*Out-of-Distribution* -OOD), [101] propõe um novo método baseado em reconstrução de características acústicas latentes. Diferentemente das técnicas tradicionais, que geralmente se apoiam em *probability scores* ou medidas de distância no espaço do classificador, o método evita essas limitações ao explorar diretamente a capacidade de reconstrução de um conjunto de *autoencoders* especializados.

A extração de características é realizada por meio do modelo pré-treinado WavLM [100]. Os autores consideram que WavLM é capaz de realizar várias tarefas associadas a fala melhor que outras técnicas tradicionais baseadas nos cálculos de características acústicas, como espectrograma Mel. As representações acústicas extraídas são comprimidas por um *autoencoder* compartilhado e reconstruídas por decodificadores dedicados, um para cada *vocoder* conhecido. Se nenhum dos decodificadores conseguir reconstruir as *features* com erro abaixo de um limiar pré-definido, o sinal é classificado como gerado por um *vocoder* OOD. A métrica utilizada para essa avaliação é o *Mean Squared Error* (MSE). Os experimentos analisaram diferentes camadas do WavLM e indicaram que a sexta camada fornece as representações mais eficazes individualmente. No entanto, combinações de múltiplas camadas mostraram desempenho ainda melhor. A abordagem foi comparada com diversos métodos de referência, como o RawNet2 [82], e demonstrou resultados superiores na tarefa de detecção OOD.

O trabalho de [19] propõe o uso do *Mean Opinion Score* (MOS) — tradicionalmente utilizado para avaliação da qualidade perceptual de áudio [102] — como ferramenta auxiliar na detecção de áudios sintéticos (*fake audio detection* - FAD). A pontuação MOS é baseada em avaliações humanas, variando de 1 (ruim) a 5 (excelente). O estudo explora a hipótese de que modelos preditores de qualidade de áudio, como o *Self-Supervised*

Learning Mean Opinion Score predictor (SSL-MOS) [103], podem distinguir gravações genuínas de falsificações. A arquitetura proposta combina três componentes: (i) SSL-FAD, que emprega modelos de aprendizado auto-supervisionado (como wav2vec ou WavLM) para classificar os áudios como reais ou falsos; (ii) Fused SSL-MOS, uma fusão de múltiplos preditores SSL-MOS; e (iii) MOS-FAD Fusion, uma camada de fusão que agrega as saídas dos dois módulos anteriores e gera a classificação final. A solução alcançou uma melhoria de 13,6% na EER em relação ao melhor sistema reportado até então.

O trabalho de [104] argumenta que abordagens anteriores apresentaram capacidades limitadas de generalização na detecção de áudios falsos. Os autores desafiam a suposição de que detectores baseados em redes neurais profundas não conseguem generalizar bem para geradores de áudio não vistos durante o treinamento. Para isso, propuseram um modelo baseado em convoluções dilatadas (*dilated convolution-based model*), treinado em representações de áudio nos domínios da Transformada de Pacotes Wavelet (*Wavelet Packet Transform – WPT*) e da Transformada de Fourier de Curto Termo (*Short-Time Fourier Transform – STFT*), cobrindo dois domínios de representação em duas dimensões. O estudo utilizou o conjunto de dados WaveFake [1], ampliado com geradores de áudio mais recentes para testar a capacidade de generalização dos modelos. Os resultados mostraram que redes convolucionais profundas (*Deep Convolutional Neural Networks – DCNNs*) apresentam robusta capacidade de generalização, sendo promissoras para mitigar o *overfitting* quando se trabalha com um número limitado de geradores conhecidos. Adicionalmente, o estudo evidenciou a relevância das componentes de alta frequência para a detecção eficaz de falsificações de áudio.

O estudo de [105] utilizou os conjuntos de dados ASVspoof 2019 Logical Access e Multi-Language Audio Anti-Spoofing Dataset (MLAAD) [106] para investigar a origem de áudios falsificados. Foram propostas duas estratégias para identificar se um áudio é falso e qual *vocoder* o gerou. A primeira, chamada *end-to-end learning*, consiste em utilizar uma arquitetura de contramedida (CM) existente e treiná-la desde o início para que ela classifique diretamente qual *vocoder* foi responsável pela geração do áudio. A segunda estratégia, denominada *two-stage*, consiste inicialmente em treinar uma CM para a tarefa de detecção binária (real ou falso). Em seguida, congela-se essa rede e adiciona-se uma nova camada de saída (*head*) voltada para a identificação do *vocoder*.

Três arquiteturas de CM conhecidas foram avaliadas. A primeira é baseada em ResNet [107] e utiliza LFCC como entrada. A segunda adota o modelo wav2vec, operando diretamente sobre o áudio bruto. A terceira utiliza o modelo Whisper [33], que também recebe o áudio bruto como entrada. Os melhores resultados foram obtidos com a arquitetura

baseada em wav2vec na abordagem *end-to-end*, que alcançou uma acurácia de 99,7% na tarefa de identificação do *vocoder*.

Já em [108], é sustentado que, para além de detectar se um áudio é verdadeiro ou falso, é igualmente relevante identificar qual algoritmo de *vocoder* foi utilizado em sua geração. Para isso, os autores propõem a arquitetura Vocoder Finger Printer Detection Net (VFD-Net), baseada em *patch-wise supervised contrastive learning* (PCL). Durante o treinamento, são selecionados aleatoriamente *patches*, i.e., pequenos retângulos, de imagens de espectrogramas Mel. Um desses *patches* é escolhido como âncora. Caso outro *patch* tenha origem no mesmo *vocoder*, ele é considerado uma amostra positiva; caso contrário, é tratado como amostra negativa. Simultaneamente, os rótulos dos *patches* são utilizados para treinar um classificador de *vocoders*. A função de perda (*loss*) combinada do modelo contempla tanto o aprendizado contrastivo quanto a classificação supervisionada.

O modelo foi avaliado nos conjuntos de dados VCTK [109] e LibriTTS [95], com testes em três cenários distintos. No cenário *closed-set*, usando dataset VCTK, obteve acurácia de 100%. No cenário de *cross-finetuning*, onde amostras do LibriTTS são utilizadas para ajuste fino sobre uma rede previamente treinada com VCTK, atingiu 97,99% de acurácia. Já no cenário *cross-dataset* a acurácia foi de 92,45%.

Além disso, o trabalho introduz um método baseado em janela deslizante para avaliar o modelo em situações de detecção parcial de falsificação, ou seja, quando apenas segmentos do áudio são manipulados. Os autores relatam que o VFD-Net superou abordagens anteriores com margens de 30% a 45% em termos de acurácia.

Em [110], foi investigada a contribuição das regiões vocalizadas (*voiced*) e não vocalizadas (*unvoiced*) para a detecção de áudios *deepfake*, abordando a falta de interpretabilidade dos modelos atuais nessa tarefa. O estudo utilizou o conjunto de dados MLAAD [106]. Para identificar se um quadro de áudio é vocalizado, os autores empregaram o algoritmo YIN [111], disponível na biblioteca librosa [112], e a detecção de atividade de fala (*Speech Activity Detection* — SAD), implementada pelo WebRTC *Voice Activity Detector* (VAD) [113]. A arquitetura AASIST [85] foi utilizada como método principal para a detecção de *deepfake*. Os resultados indicaram que as regiões não vocalizadas apresentam maior eficácia na distinção entre áudios sintéticos e reais. Enquanto a EER obtida usando o áudio completo foi de 11,40%, ao considerar apenas as partes não vocalizadas, a EER reduziu para 6,62%. Além disso, a fusão em nível de *score* (*score-level fusion*) entre as regiões vocalizadas e não vocalizadas resultou em uma EER ainda menor, de 5,82%.

Conclui-se que sistemas de detecção de *deepfake* treinados especificamente em regiões

não vocalizadas tendem a apresentar melhor desempenho do que aqueles treinados utilizando o áudio completo.

Em 2025, foi proposto um método, em [114], para aprimorar a generalização na detecção de vozes sintéticas geradas por inteligência artificial, enfrentando limitações observadas em abordagens anteriores, como a dependência de *vocoders* específicos, a sensibilidade a ruídos e as características individuais dos locutores. Essas limitações, segundo os autores, comprometem significativamente o desempenho em testes *cross-domain*. A proposta centra-se na extração de artefatos agnósticos ao domínio, ou seja, padrões compartilhados por diferentes *vocoders*, com o objetivo de construir uma representação mais robusta e independente do domínio de origem. Para isso, o modelo separa explicitamente artefatos agnósticos ao domínio e artefatos específicos ao domínio, o que permite uma melhor adaptação a diferentes condições. Os autores de [114] observam que arquiteturas como o RawNet2 tendem a memorizar padrões do conjunto de treinamento, resultando em superfícies de perda acentuadas (*sharp loss landscapes*) e múltiplos mínimos locais. Com base nessa análise, argumenta-se que o achatamento da superfície de perda (*loss landscape*) é essencial para promover trajetórias de otimização que favoreçam a generalização. O modelo proposto usa uma arquitetura de *autoencoder*, onde o codificador extrai representações distintas para o conteúdo, para os artefatos agnósticos ao domínio e para os artefatos específicos ao domínio. Essas representações, que conjuntamente formam o espaço latente, são posteriormente utilizadas por um decodificador para reconstrução do áudio. Durante o treinamento, duas estratégias de aprendizado contrastivo são aplicadas de forma simultânea. A primeira aproxima representações de artefatos agnósticos quando os áudios pertencem à mesma classe (ambos reais ou ambos sintéticos) e os separa quando pertencem a classes diferentes. A segunda estratégia aproxima artefatos específicos ao domínio quando os áudios são sintéticos e gerados pela mesma tecnologia, separando-os nos demais casos. Adicionalmente, é empregada uma função de perda baseada em informação mútua entre as representações de conteúdo e as agnósticas ao domínio. Os autores argumentam que esse mecanismo auxilia no achatamento da superfície de perda, o que, por sua vez, favorece a generalização do modelo. Em termos de desempenho, o método proposto superou o melhor modelo anterior em 5,12% no cenário *intra-domain* e em 7,59% no cenário *cross-domain*.

3.2 Competições de Detecção de *Deepfakes*

Como apresentado na subseção anterior, diversos métodos têm sido propostos para a tarefa de detecção de *deepfakes*. Embora os autores geralmente reportem o desempenho de

seus algoritmos, a comparação direta entre os sistemas é dificultada pelo uso de diferentes conjuntos de datasets e condições experimentais. Diante disso, surgem as chamadas competições (*challenges*), que fornecem um ambiente padronizado para avaliação comparativa, permitindo que os pesquisadores testem seus métodos sob as mesmas condições e compartilhem resultados de forma transparente.

3.2.1 ASVspoof Challenge

A ASVspoof 2015 foi a primeira edição da competição ASVspoof³, cujos resultados foram apresentados durante a conferência INTERSPEECH 2015, em Dresden, Alemanha [115]. O principal objetivo dessa competição foi padronizar os testes para a detecção de ataques *spoof* em sistemas biométricos de reconhecimento de fala, permitindo a comparação direta entre diferentes algoritmos. *Spoofing* é definido como um ataque no qual um fraudador tenta manipular sistemas biométricos para se passar por outra pessoa.

A competição concentrou-se exclusivamente na detecção de fala falsificada (*spoofed*), utilizando um conjunto de dados composto por amostras geradas a partir de dez diferentes algoritmos de síntese e conversão de voz. O dataset foi dividido em três partes: treinamento (*training*), desenvolvimento (*development*) e avaliação (*evaluation*). Os áudios genuínos foram produzidos por 106 locutores, enquanto os áudios falsos resultaram de modificações desses originais; ao todo, o conjunto continha 263.151 áudios, entre genuínos e sintéticos.

Dois cenários de ataque foram propostos: ataques conhecidos (*known attacks*), que estavam presentes durante o treinamento e desenvolvimento, e ataques desconhecidos (*unknown attacks*), que simulam situações mais realistas, com ataques inéditos para o sistema, que estavam presentes na avaliação, momento em que foi possível avaliar os sistemas de detecção em relação a sua capacidade de generalização.

O melhor algoritmo submetido à competição — *Combining Evidences from Mel Cepstral, Cochlear Filter Cepstral and Instantaneous Frequency Features for Detection of Natural vs. Spoofed Speech* [72] — alcançou uma EER, no cenário de avaliação, de 0,408% para ataques conhecidos e de 2,013% para ataques desconhecidos, resultando em uma média geral de 1,211% [72]. Este método consistiu em uma combinação de evidências extraídas de características acústicas distintas, incluindo MFCC, coeficientes *Cochlear Filter Cepstrais* (CFCC) e características de frequência instantânea (*Instantaneous Frequency Features*), que foram integradas para distinguir fala natural de fala sintética.

³<https://www.asvspoof.org/>

A edição de 2017 tratou apenas de *replay attack* [79], que ocorre quando um atacante grava a fala de um usuário autorizado e a reproduz junto ao sistema de acesso, por isso não será tratada aqui.

Em 2019, uma nova edição da competição ocorreu, o ASVspoof 2019 [77], que considerou cenários de *Logical Access* (LA) e *Physical Access* (PA). O cenário LA refere-se a ataques realizados diretamente no sinal de áudio digital, antes do acesso físico ao dispositivo, sendo este o foco principal deste estudo.

Para avaliação, o ASVspoof 2019 introduziu a *tandem detection cost function* (t-DCF) [116], uma métrica que avalia conjuntamente a capacidade do sistema em identificar o locutor e detectar ataques *spoof*. Contudo, neste trabalho a métrica t-DCF não será utilizada, uma vez que a identificação do locutor não é o foco.

O dataset do cenário LA é composto por falas genuínas (*bonafide speech*) provenientes de 78 locutores, além de sinais sintéticos gerados por 17 técnicas diferentes a partir desses áudios genuínos, totalizando aproximadamente 130.000 arquivos [116]. Assim como anteriormente, o dataset está particionado em três subconjuntos: treinamento, desenvolvimento e avaliação, contendo ataques conhecidos (*known attacks*) e desconhecidos (*unknown attacks*).

O sistema com melhor desempenho no cenário LA, segundo o ASVspoof 2019, foi desenvolvido pela empresa ID R&D [117]. Embora a técnica exata não tenha sido divulgada, sabe-se que o sistema utiliza redes neurais em um *ensemble* de classificadores, alcançando uma EER igual a 0,22%.

A competição ASVSpooF 2021 [118] introduziu, além dos cenários *Logical Access* (LA) e *Physical Access* (PA), um novo cenário denominado *Deepfake* (DF).

No cenário PA desta edição, os ataques contêm artefatos gerados por codecs e compressões associadas a canais VoIP e PSTN, que são comuns em sistemas telefônicos reais. Já o objetivo no cenário DF não é enganar um sistema de Verificação Automática de Locutor (*Automatic Speaker Verification* — ASV), mas sim gerar voz sintética a partir da voz de uma vítima para disseminar desinformação ou *fake news* e enganar a sociedade.

Nesse cenário, simula-se uma situação em que o atacante possui acesso à voz da vítima, por exemplo, por meio de redes sociais, realiza a adulteração da voz e reintroduz o material manipulado em redes sociais, *call centers* ou outras aplicações.

Para o presente trabalho, o foco está restrito ao cenário DF. Cabe destacar que o conjunto de dados utilizado é uma modificação do dataset de 2019, com alterações que dificultam o processo [118].

Quanto à avaliação, o dataset DF é composto por áudios *bonafides* e *spoofed*, comprimidos ou não, utilizando codecs comuns em armazenamentos de mídia. Os áudios são construídos, comprimidos e, quando necessário, descomprimidos antes da avaliação. O conjunto de avaliação para DF contém aproximadamente 500 mil amostras. [119]

O melhor sistema (não divulgado) para o cenário DF obteve uma EER de 15,64%, evidenciando a complexidade e o desafio dessa tarefa [119].

A competição ASVspoof 5 [120] foi construída a partir de uma base de dados de fala obtida por meio de coleta massiva (*crowdsourcing*) de amostras de cerca de 2.000 locutores e conta com duas trilhas. A Trilha 1 reflete um cenário no qual o atacante possui acesso à voz da vítima, por exemplo, a partir de publicações em redes sociais, e, com base nesse material, gera um áudio falsificado (*spoofed*) semelhante à voz original. A Trilha 2 mantém o mesmo objetivo do cenário de *Logical Access* (LA) da competição de 2021, simulando um ambiente de telefonia em que a fala, autêntica ou falsificada, é submetida às codificações e compressões típicas de sistemas de telecomunicação.

O presente trabalho concentra-se na Trilha 1. O conjunto de dados foi extraído do corpus Multilingual LibriSpeech (MLS) [121] e, somando as partições de treinamento, desenvolvimento e avaliação, atinge aproximadamente um milhão de amostras de áudio. O melhor sistema na *closed condition*, isto é, treinado exclusivamente com os dados oficiais da competição, alcançou uma EER de 8,61%. Embora poucas informações tenham sido divulgadas sobre este modelo, sabe-se que ele emprega um *ensemble* de subsistemas. Já na *open condition*, na qual é permitido o uso de dados externos para o treinamento, o melhor sistema registrou uma EER de 2,59%, utilizando um modelo pré-treinado baseado em aprendizado auto-supervisionado (*self-supervised learning* — SSL).

3.2.2 ADD Challenge

A Audio Deep Synthesis Detection⁴ (ADD) 2022 Challenge [92] foi proposta para preencher lacunas deixadas pelas edições dos anos anteriores da competição ASVspoof, especialmente a ausência de ruídos de fundo e distúrbios diversos nos áudios falsos. A competição contou com três trilhas: a Trilha 1, *Low-Quality Fake Audio Detection* (LF), composta por áudios *bonafide* e adulterações completas, com modificação de todo o áudio e adição de ruídos e sons ambientes do mundo real; a Trilha 2, *Partially Fake Audio Detection* (PF), formada por áudios *bonafide* e gravações com apenas trechos delimitados alterados, mantendo o restante autêntico; e a Trilha 3, *Audio Fake Game* (FG), voltada para um cenário adversarial envolvendo simultaneamente a geração e a detecção de áudios falsos.

⁴<http://addchallenge.cn/>

Neste trabalho, consideramos apenas a Trilha 1 (LF). O conjunto de dados de desenvolvimento e treinamento para esta trilha foi derivado do corpus AISHELL-3 [122], que contém gravações de 218 falantes de mandarim, totalizando aproximadamente 85 horas de áudio. Também foi fornecido um dataset de adaptação, com cerca de mil gravações, onde os participantes podem ajustar seus modelos antes de enviar para avaliação.

O melhor desempenho reportado na competição para esta trilha alcançou uma EER de 21,7%, obtido por um sistema baseado no wav2vec2 [12].

Em 2023 uma nova edição foi organizada. A competição ADD 2023 [98] teve como foco não apenas decidir se um áudio pertence à classificação binária verdadeiro ou falso, mas também identificar em que trecho do áudio ocorreu a alteração e qual técnica foi utilizada para tal. A competição foi dividida em três trilhas: Trilha 1, *Audio Fake Games* (FG), com as mesmas regras da Trilha FG da edição anterior; Trilha 2, Localização da Região Manipulada (*Region Location*, RL), voltada para a identificação precisa da região manipulada em áudios parcialmente falsos; e Trilha 3, Reconhecimento do Algoritmo Usado para Criação de *Deepfake* (*Deepfake Algorithm Recognition*, AR), cujo objetivo é reconhecer se o áudio inteiro é falso e, em caso afirmativo, determinar qual algoritmo foi responsável pela síntese.

Neste trabalho, o interesse recai sobre a Trilha 3. Para essa tarefa, o conjunto de dados inclui mais de 30 mil áudios entre *bonafides* e sintéticos, sendo estes últimos fabricados utilizando sete métodos diferentes. Em ambos os casos, o artigo não detalha a fonte dos áudios *bonafides*, o que limita a rastreabilidade e análise detalhada da origem das amostras. A diversidade de métodos de síntese empregados visa aumentar a robustez e a generalização dos sistemas participantes.

A métrica adotada para a Trilha 3 foi o F1-score. O vencedor dessa trilha obteve um score de 89.63, demonstrando desempenho superior na identificação correta das classes conhecidas e na detecção de algoritmos desconhecidos. O método vencedor é baseado no extrator de *embeddings* ECAPA-TDNN [97] e no uso do algoritmo *k-th Nearest Neighbors* (kNN) para avaliar a distância entre esses *embeddings*. Nesse método, dois áudios são considerados como criados pelo mesmo *vocoder* se seus *embeddings* são próximos o suficiente segundo essa distância [96].

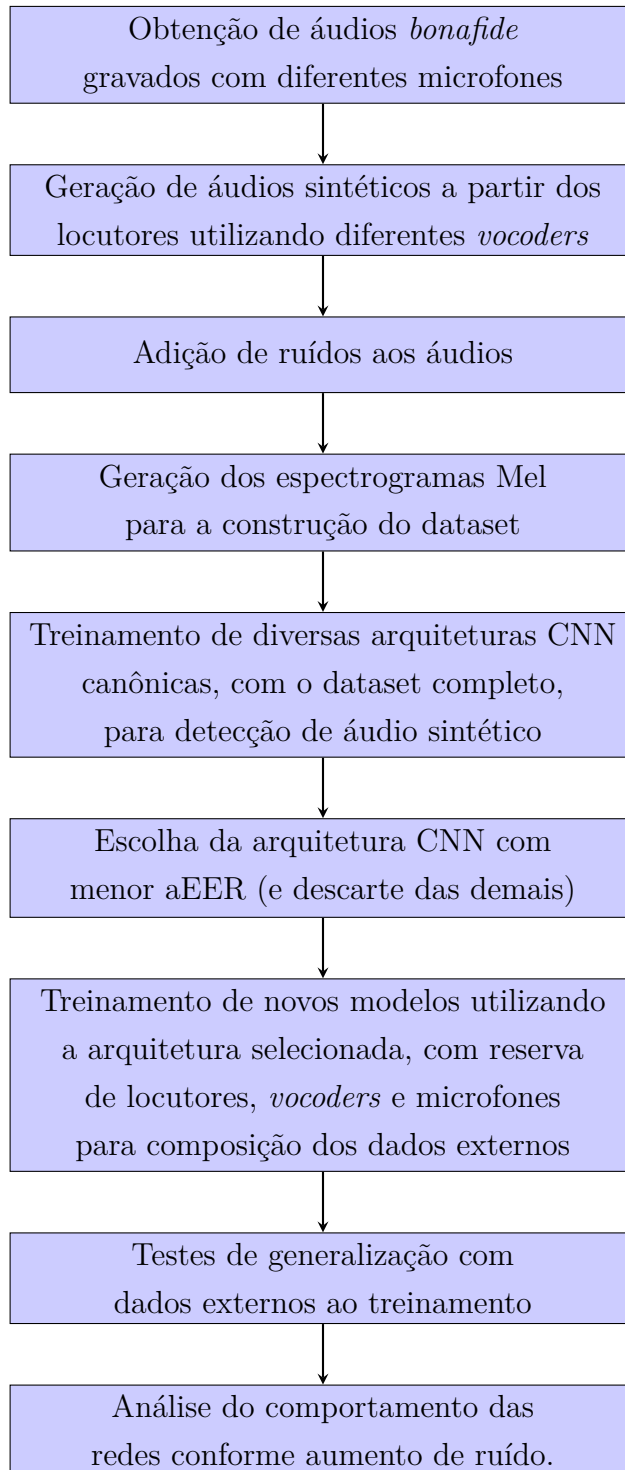
Capítulo 4

Metodologia Proposta

Neste Capítulo é apresentado um método de análise de espectrogramas Mel por meio de redes neurais convolucionais (CNNs), com o objetivo de determinar se os áudios que os originaram são legítimos ou sintéticos. Inicialmente, são coletados áudios genuínos de diferentes locutores, gravados com distintos dispositivos de captura. Em seguida, são geradas versões sintéticas dessas amostras utilizando diferentes técnicas de síntese de voz. Posteriormente, os áudios são submetidos à adição de ruído e convertidos em espectrogramas Mel, formando o conjunto de dados utilizado nos experimentos.

Com esse conjunto de dados, são treinadas e avaliadas diferentes arquiteturas canônicas de CNNs para a tarefa de detecção de áudio sintético, sendo selecionada a arquitetura que apresentou o menor aEER nas condições experimentais avaliadas. A partir dessa arquitetura, são realizados novos treinamentos com diferentes configurações de exclusão de dados, de modo que determinados locutores, dispositivos de captura e métodos de síntese não estejam presentes durante o treinamento. Os modelos resultantes são então submetidos a testes de generalização, nos quais a inferência é realizada exclusivamente sobre dados não utilizados no treinamento.

Por fim, é conduzida uma análise do comportamento dos modelos diante de diferentes níveis de ruído nos áudios. O fluxograma apresentado a seguir ilustra, de forma esquemática, todas as etapas desse processo.



4.1 Dataset

4.1.1 Base de Áudios Originais

Para cumprir com os objetivos deste trabalho, montamos nosso próprio dataset, uma vez que precisávamos de um dataset que pudesse ser particionado por tipo de *vocoder*, por locutor, por microfone usado, e por nível de ruído. Foram utilizados os áudios originais de locutores associados ao VCTK Corpus, corpus de fala em inglês criado pelo Centre for Speech Technology Research (CSTR) da Universidade de Edimburgo [109]. O VCTK Corpus foi construído com 110 locutores que, ao todo, ajudaram a construir uma base com 88.328 arquivos de áudio, sendo que metade foi gravada com um microfone (mic1) e a outra, com outro microfone (mic2).

O VCTK Corpus apresenta uma diversidade relevante de sotaques do inglês, embora com predominância de falantes do Reino Unido, refletindo sua origem na University of Edinburgh. A maioria dos locutores utiliza variações do inglês britânico, sotaques escoceses, irlandeses e regionais do norte e sul da Inglaterra. Além disso, o corpus inclui, em menor proporção, locutores com sotaques de outras regiões, como Estados Unidos, Canadá, Índia e Austrália. Cada locutor foi incumbido de ditar aproximadamente 400 frases, derivadas de uma lista padronizada de prompts. As gravações originais foram capturadas com uma frequência de amostragem de 48 kHz. No entanto, para compatibilidade com os *vocoders* escolhidos para este estudo, os dados de áudio foram reamostrados para as frequências de 16.000 Hz, 22.050 Hz e 24.000 Hz.

Os arquivos de áudio apresentam uma duração média de aproximadamente 3.4 segundos por enunciado. Para fornecer uma visão mais detalhada sobre o conjunto de dados, é mostrada a Tabela 4.1 de duração de média dos áudios, incluindo o desvio padrão. O histograma de durações da totalidade dos áudios é representado na Figura 4.1. Essa distribuição garante um conjunto de dados diversificado para a avaliação dos sistemas de síntese e detecção.

4.1.2 Geração de Áudio Sintético

A geração de áudio sintético neste trabalho foi realizada por meio da utilização de múltiplos *vocoders* neurais provenientes de diferentes implementações e grupos de pesquisa. Essa estratégia teve como objetivo ampliar a diversidade tecnológica dos sinais sintetizados, permitindo uma avaliação mais abrangente e robusta dos detectores de *vocoders*.

De maneira geral, os processos seguiram pipelines parecidos. Inicialmente, os áudios *bonafide* foram utilizados como entrada para a extração de espectrogramas Mel, configurados de acordo com os requisitos experimentais. Essas representações foram então

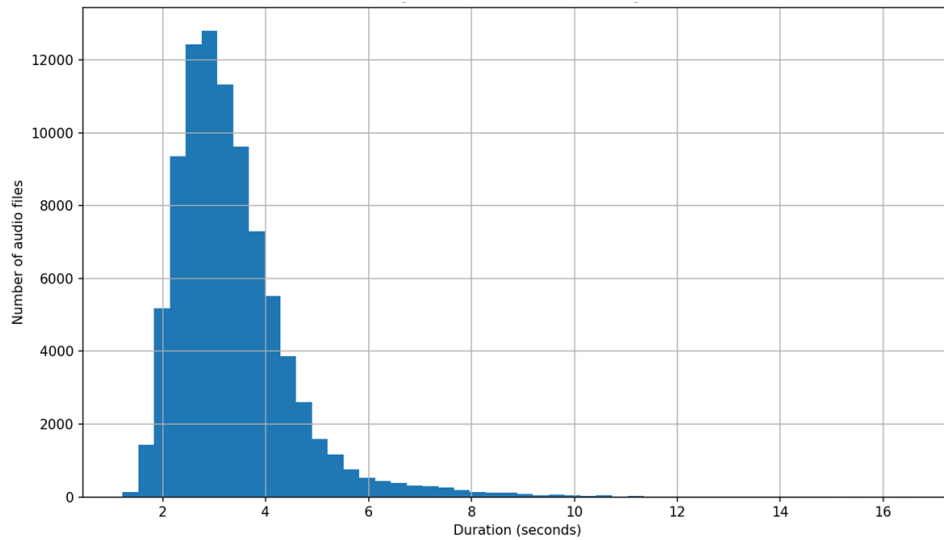


Figura 4.1: Histograma de duração dos áudios

utilizadas como entrada para os diferentes *vocoders*, responsáveis pela reconstrução dos sinais no domínio do tempo.

Cabe destacar que todos os modelos utilizados neste trabalho operam a partir de espectrogramas Mel como representação intermediária, realizando a reconstrução do sinal no domínio do tempo a partir dessas representações.

Cada conjunto de modelos foi executado em ambientes isolados, com versões específicas de dependências, assegurando reprodutibilidade e compatibilidade com as implementações originais.

Síntese de áudio usando modelos treinados por Kan-Bayashi

Nesta etapa, foram empregados *vocoders* disponibilizados na implementação proposta por Kan-Bayashi, acessível em <https://github.com/kan-bayashi/ParallelWaveGAN>.

Foram utilizados os seguintes modelos pré-treinados:

- `ljspeech_parallel_wavegan.v3`
- `ljspeech_melgan.v3`
- `ljspeech_full_band_melgan.v2`
- `ljspeech_multi_band_melgan.v2`
- `ljspeech_hifigan.v1`
- `ljspeech_style_melgan.v1`
- `arctic_slt_parallel_wavegan.v1`

- `libritts_parallel_wavegan.v1`
- `libritts_multi_band_melgan.v2`
- `libritts_hifigan.v1`
- `libritts_style_melgan.v1`

Esses modelos foram treinados sobre diferentes bases de dados, como LJSpeech [123], LibriTTS [95] e Arctic [75].

Síntese de áudio usando modelos treinados por lmnt-com

Para complementar o conjunto de dados, foi utilizado um modelo baseado na arquitetura DiffWave, disponibilizado pela LMNT¹. Foi empregado o modelo pré-treinado `diffwave-ljspeech-22kHz-1000578.pt`, treinado sobre o conjunto de dados LJSpeech.

Síntese de áudio usando modelos treinados por Rongjie Huang

Também foram utilizados modelos baseados na arquitetura FastDiff, disponibilizados por seus autores². Especificamente, foram empregados os modelos `model\_ckpt\_steps\_500000.ckpt`, treinado sobre o conjunto de dados LJSpeech, e `model\_ckpt\_steps\_1000000.ckpt`, treinado sobre o conjunto de dados LibriTTS.

Síntese de áudio usando modelos treinados por Gemelo AI

Nesta etapa, foi utilizado o *vocoder* Vocos, disponibilizado pela Gemelo AI³, por meio do modelo pré-treinado `charactr/vocos-mel-24khz`.

Síntese de áudio usando modelos treinados por SpeechBrain

Foi também utilizado um modelo HiFi-GAN disponibilizado pela biblioteca SpeechBrain⁴, especificamente o modelo `speechbrain/tts-hifigan-libritts-16kHz`, treinado sobre o conjunto LibriTTS.

Síntese de áudio usando modelos treinados por SpeechT5

Adicionalmente, foi empregado um modelo HiFi-GAN integrado ao ecossistema SpeechT5, disponibilizado pela Microsoft⁵, utilizando o modelo `microsoft/speecht5_hifigan`.

¹<https://lmnt.com/assets/diffwave/>

²<https://github.com/Rongjiehuang/FastDiff>

³<https://github.com/gemelo-ai/vocos>

⁴<https://speechbrain.github.io/>

⁵https://huggingface.co/microsoft/speecht5_hifigan

Síntese de áudio usando modelos treinados por Guan-Ting

Foi utilizado um modelo baseado na arquitetura MelGAN, com pesos disponibilizados por seus autores⁶, especificamente os arquivos `best_netG.pt` e `args.yml`. A implementação de referência foi obtida do repositório oficial do projeto⁷.

Síntese de áudio usando modelos treinados pela NVIDIA

Por fim, foram utilizados modelos disponibilizados pela NVIDIA. Entre eles, destaca-se o modelo `nvidia_waverglow`, obtido por meio do Torch Hub⁸. Também foram empregados modelos BigVGAN disponibilizados pela NVIDIA⁹, incluindo `nvidia/bigvgan_v2_24khz_100band_256x` e `nvidia/bigvgan_v2_22khz_80band_256x`, obtidos por meio da plataforma Hugging Face.

Organização do conjunto de dados

Os áudios sintéticos gerados por todos os *vocoders* foram organizados em uma estrutura padronizada, que preserva a identidade do locutor, a frequência de amostragem e o modelo gerador. Cada *vocoder* foi associado a um código específico (`v_01`, `v_02`, ... , `v_21`) — veja Tabela 4.2 —, permitindo sua rastreabilidade ao longo dos experimentos.

Para fins de comparação, os áudios originais, que não passaram por nenhum processo de síntese, foram rotulados como pertencentes à classe *bona*, representando a ausência de processamento por *vocoders*.

Alguns arquivos de áudio (por motivos não investigados) não passaram com sucesso por todos os pipelines de geração de áudio sintético e foram excluídos do dataset.

4.1.3 Adição de ruído

Com o objetivo de avaliar a robustez dos detectores de *vocoders* em condições mais adversas, realizou-se a adição controlada de ruído branco gaussiano aos sinais de áudio.

O procedimento foi aplicado de forma uniforme a todo o conjunto de dados, abrangendo tanto os áudios *bonafide* quanto os áudios sintetizados.

A inserção de ruído foi conduzida de modo a obter diferentes níveis de relação sinal-ruído (*Signal-to-Noise Ratio* – SNR), expressos em decibéis. Foram considerados os seguintes valores de SNR_{dB} : 30 dB, 20 dB, 10 dB, 0 dB, -10 dB, -20 dB e -30 dB, além da condição original sem adição de ruído. Essa faixa foi escolhida por abranger diferentes

⁶<https://huggingface.co/Guan-Ting/StyleSpeech-MelGAN-vocoder-16kHz>

⁷<https://github.com/descriptinc/melgan-neurips>

⁸https://pytorch.org/hub/nvidia_deeplearningexamples_waverglow/

⁹<https://github.com/NVIDIA/BigVGAN>

níveis de degradação acústica, desde condições em que o sinal de fala permanece praticamente inalterado (30 dB e 20 dB) até cenários severos de mascaramento pelo ruído (-20 dB e -30 dB). Estudos de inteligibilidade de fala mostram que a compreensão humana decresce progressivamente à medida que a SNR diminui, tornando-se bastante reduzida para valores inferiores a aproximadamente -10 dB e próxima de zero em torno de -15 dB ou menos [124].

Antes do cálculo da potência, cada sinal foi centralizado por meio da subtração de sua média amostral, de modo a garantir média zero.

A potência de cada sinal foi então calculada como a média do quadrado de suas amostras, conforme definido na Equação 4.1 [30, 125]:

$$P_{\text{sinal}} = \frac{1}{N} \sum_{n=0}^{N-1} x[n]^2, \quad (4.1)$$

onde $x[n]$ representa a n -ésima amostra do sinal e N o número total de amostras.

A relação sinal-ruído em decibéis é dada pela Equação 4.2 [30]:

$$\text{SNR}_{\text{dB}} = 10 \log_{10} \left(\frac{P_{\text{sinal}}}{P_{\text{ruído}}} \right). \quad (4.2)$$

A partir da Equação 4.2, a potência do ruído foi obtida conforme a Equação 4.3:

$$P_{\text{ruído}} = \frac{P_{\text{sinal}}}{10^{\text{SNR}_{\text{dB}}/10}}. \quad (4.3)$$

O ruído foi então gerado a partir de uma distribuição gaussiana centrada em zero, com variância igual à potência [30, 125] calculada na Equação 4.3. Consequentemente, o desvio padrão do ruído foi determinado pela Equação 4.4:

$$\sigma = \sqrt{P_{\text{ruído}}}. \quad (4.4)$$

O ruído foi adicionado diretamente às amostras do sinal no domínio do tempo para cada arquivo de áudio. Após essa etapa, quando necessário, os sinais foram reescalados para evitar saturação (*clipping*), preservando sua integridade numérica.

Esse procedimento foi aplicado individualmente a cada arquivo de áudio.

4.1.4 Geração do Conjunto de Imagens de Espectrograma Mel

Conforme mencionado anteriormente, os detectores neurais não recebem diretamente os arquivos de áudio brutos, mas sim imagens que representam os espectrogramas Mel desses sinais. Especificamente, foram utilizadas imagens no formato PNG, por se tratar de um formato com compressão sem perdas [126], evitando a introdução de artefatos decorrentes

da compactação. Para a representação visual dos espectrogramas, foi utilizado o mapa de cor *magma* [127], embora outros mapas de cor também pudessem ser empregados. No processo de geração das imagens, foram utilizados os seguintes parâmetros: tamanho da janela FFT de 1024 pontos, janela de análise de 1024 amostras, janela de Hann, 80 filtros Mel e *hop length* de 256 amostras. Foram usadas as frequências de amostragem de 16.000, 22.050 e 24.000 Hz, a depender da frequência de amostragem do áudio que gerou a imagem. Assim, cada imagem de espectrograma gerada apresenta altura fixa de 80 pixels e largura variável, conforme a duração do áudio.

A largura da imagem é função da duração do sinal de áudio e do parâmetro *hop length*, sendo dada por:

$$w = \left\lfloor \frac{n}{h} \right\rfloor, \quad (4.5)$$

onde w representa a largura da imagem em pixels (número de frames no tempo), n é o número total de amostras do sinal de áudio e h corresponde ao *hop length*. Considerando que $n = f_s \cdot T$, onde f_s é a frequência de amostragem e T a duração do áudio em segundos, a Equação 4.5 pode ser reescrita como:

$$w = \left\lfloor \frac{f_s \cdot T}{h} \right\rfloor. \quad (4.6)$$

Dessa forma, a largura do espectrograma cresce linearmente com a duração do áudio, sendo também influenciada pela frequência de amostragem utilizada.

A largura mínima da imagem deve ser de 80 pixels, uma vez que a rede realiza recortes (*crop*) de dimensão 80×80 . Assim, para cada frequência de amostragem considerada, existe uma duração mínima de áudio necessária para garantir esse tamanho. Essa relação pode ser obtida diretamente da Equação 4.5.

Considerando os valores de frequência de amostragem utilizados neste trabalho e um *hop length* de 256 amostras, a Tabela 4.3 apresenta a duração mínima necessária para que o espectrograma atinja largura igual ou superior a 80 pixels.

Algumas imagens geradas apresentaram largura inferior a esse limite e, portanto, foram excluídas do conjunto de dados. Ao final do processo, obteve-se 16.759.488 arquivos de imagens em um dataset de espectrogramas Mel organizado por *vocoder*, locutor, microfone, frequência de amostragem e nível de ruído. A Figura 4.2 apresenta alguns exemplos dos espectrogramas gerados.

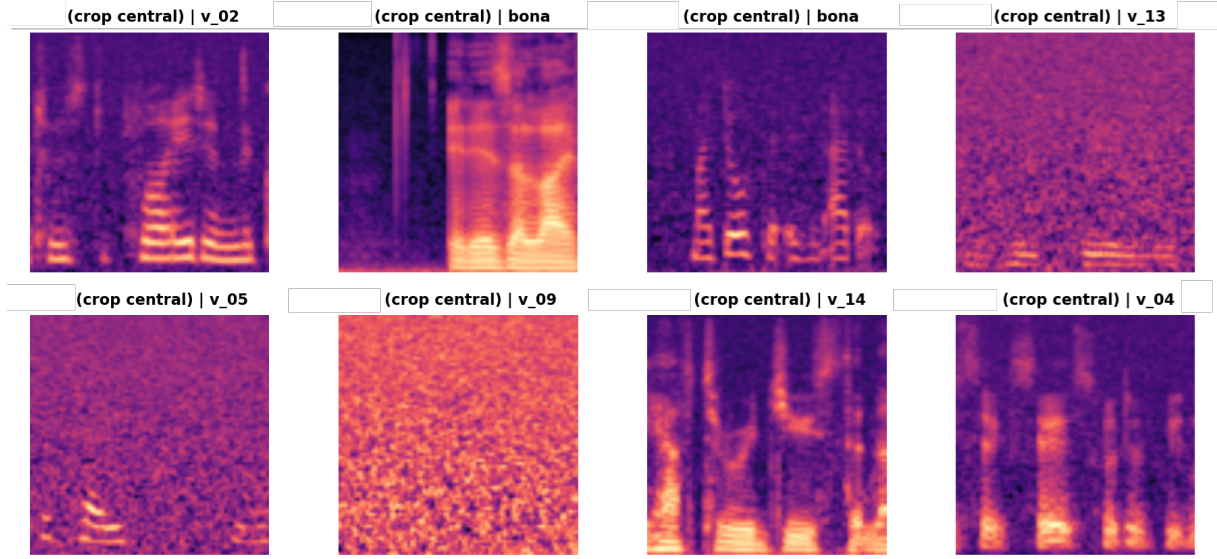


Figura 4.2: Exemplos de espectrogramas Mel constantes do dataset, com crop 80x80.

4.2 Detector Usando Rede Neural Convolutacional

4.2.1 Definição da Arquitetura

Neste estudo, uma arquitetura de detector de *vocoder* foi desenvolvida e treinada. A arquitetura é baseada em uma configuração padrão de rede neural convolutacional (CNN), conforme ilustrado na Figura 4.3.

A entrada da rede é uma imagem *bitmap* que representa o espectrograma Mel do sinal de áudio de entrada. Essa representação, amplamente adotada em tarefas de processamento de fala, encapsula as características espectrais e temporais do áudio em um formato compacto para redes, facilitando o aprendizado por redes neurais.

Para avaliar o desempenho da rede treinada, várias arquiteturas bem estabelecidas de aprendizado profundo foram utilizadas. Especificamente, as seguintes arquiteturas clássicas foram empregadas: AlexNet [128], ResNet34 [107], ResNet50 [107], VGG16 [129] e VGG19 [129]. Essas arquiteturas foram selecionadas em função de suas características distintas, tais como diferentes níveis de profundidade, eficiência computacional e capacidade de generalização. Todas foram adaptadas para um problema de classificação com 22 classes, correspondentes aos 21 *vocoders* e à classe *bonafide*. Adicionalmente, em experimentos voltados à avaliação de generalização, foram consideradas configurações com um número reduzido de classes, nas quais determinados *vocoders* foram deliberadamente excluídos da fase de treinamento e utilizados apenas na etapa de teste de generalização. Essa abordagem caracteriza um cenário de generalização para classes não vistas, frequentemente descrito na literatura como avaliação *out-of-distribution* [96, 101].

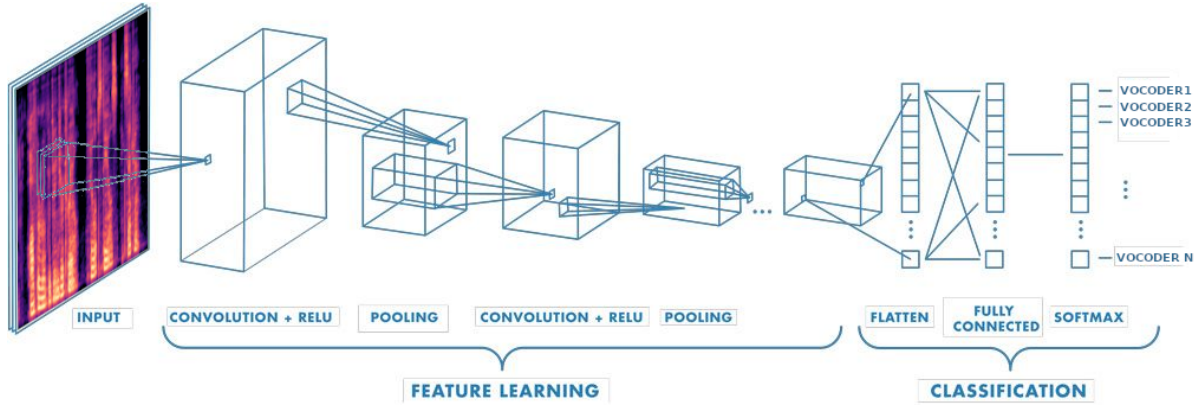


Figura 4.3: Arquitetura básica do detector de *vocoder* baseado em CNN: a rede é alimentada com imagens bitmap de espectrogramas Mel para prever qual *vocoder* gerou o áudio correspondente. Adaptado de [13].

4.2.2 Configuração Experimental Proposta

Para os experimentos, utiliza-se uma GPU Tesla V100. Inicialmente, gera-se um arquivo *manifest.csv*, no qual os arquivos são organizados segundo os seguintes atributos: *vocoder*, locutor, microfone, frequência de amostragem e nível de ruído.

São considerados 22 valores possíveis para *vocoder* (21 *vocoders* distintos além da classe *bonafide*), 110 locutores, dois tipos de microfone, três frequências de amostragem e oito níveis de ruído.

Na primeira etapa, todas as imagens são utilizadas para treinar e validar diferentes arquiteturas de redes neurais convolucionais, com o objetivo de selecionar o modelo com melhor desempenho. Para isso, utiliza-se o campo *vocoder* como rótulo, caracterizando o problema como uma tarefa de classificação multiclasse. Os dados são divididos em conjuntos de treinamento (70%) e validação (30%).

São avaliadas diferentes arquiteturas de redes neurais convolucionais, incluindo modelos clássicos amplamente utilizados na literatura. As redes são estruturadas com uma camada final do tipo *softmax*, permitindo a estimativa de uma distribuição de probabilidades entre as classes. O treinamento é conduzido utilizando a função de perda *cross-entropy*, enquanto a otimização dos parâmetros é realizada por meio do algoritmo Adam.

A seleção da arquitetura é conduzida com base no desempenho no conjunto de validação, adotando-se como critério a métrica *average Equal Error Rate* (aEER), empregada em cenários multiclasse, especialmente em desafios como ASVspoof e ADD, por constituir uma extensão natural do *Equal Error Rate* (EER) para esse contexto. No caso binário, o EER é definido como o ponto de operação em que as taxas de falso positivo (*False Acceptance Rate* – FAR) e falso negativo (*False Rejection Rate* – FRR) se igualam [130]. Para problemas multiclasse, a aEER é obtida a partir de uma estratégia *one-vs-rest*, na

qual cada classe é, alternadamente, tratada como classe positiva em relação às demais, sendo calculado um valor de EER específico para cada classe. Formalmente, considerando K classes, a aEER é definida como

$$\text{aEER} = \frac{1}{K} \sum_{k=1}^K \text{EER}_k, \quad (4.7)$$

em que EER_k representa o EER obtido para a k -ésima classe em relação às demais, e K corresponde ao número total de classes. Dessa forma, a aEER corresponde à média aritmética dos EERs individuais das classes.

Após a definição da arquitetura com melhor desempenho, inicia-se uma segunda etapa. Nessa fase, a rede selecionada é utilizada como base para o treinamento de um modelo detector voltado à distinção entre amostras *bonafide* e sintéticas, apesar da rede ainda possuir múltiplas saídas correspondendo à classe *bonafide* e às classes relativas aos *vocoders*. A camada final é substituída por uma função de ativação do tipo *sigmoid*, permitindo a estimativa independente de probabilidades para cada classe.

Nessa etapa os dados são novamente divididos em conjuntos de treinamento e validação, mantendo a mesma estratégia adotada na etapa anterior. Adicionalmente, são aplicados critérios de exclusão sobre subconjuntos específicos de *vocoders*, locutores e microfone, os quais são removidos do processo de treinamento e validação, sendo reservados para etapas posteriores de avaliação de generalização.

O treinamento é conduzido utilizando a função de perda *binary cross-entropy*, adequada à modelagem independente das saídas. O desempenho do modelo é acompanhado no conjunto de validação, sendo mantida a utilização da métrica *average Equal Error Rate* (aEER) como critério principal de avaliação, permitindo uma análise do comportamento do modelo em relação às diferentes classes.

A escolha dessa função de perda se justifica em cenários de generalização, nos quais o áudio de teste pode ser gerado por *vocoders* não observados durante o treinamento. Nessas situações, espera-se que a rede não atribua alta probabilidade a nenhuma classe associada a *vocoder* conhecido, tampouco à classe *bonafide*, resultando em todas as saídas próximas de zero, o que indica que o áudio pode ter sido gerado sinteticamente por um *vocoder* não contemplado no treinamento.

Por fim, realiza-se a etapa de teste de generalização, na qual são utilizados para inferência áudios gerados por *vocoders*, locutores e microfone previamente excluídos do processo de treinamento e validação. Essa estratégia garante que tais amostras não sejam observadas em nenhuma etapa anterior do aprendizado, permitindo avaliar a capacidade do detector de responder a dados verdadeiramente não vistos. Dessa forma, busca-se aproximar a análise de um cenário mais realista de aplicação, no qual o modelo pode ser

exposto a condições distintas daquelas presentes no treinamento.

Speaker	avg (s)	std (s)	Speaker	avg (s)	std (s)	Speaker	avg (s)	std (s)
p225	3.559	1.410	p226	3.736	1.328	p227	3.379	1.287
p228	3.887	1.363	p229	3.081	1.115	p230	3.513	1.237
p231	2.592	0.771	p232	2.876	1.038	p233	3.440	1.150
p234	3.365	1.030	p236	2.721	0.855	p237	3.519	1.115
p238	3.820	1.243	p239	3.049	0.832	p240	3.207	1.100
p241	3.203	1.078	p243	3.823	1.251	p244	3.055	0.941
p245	3.545	1.392	p246	3.589	1.444	p247	3.250	0.894
p248	3.576	1.426	p249	3.419	1.238	p250	2.747	0.872
p251	3.270	1.152	p252	3.414	1.151	p253	3.060	1.200
p254	2.984	1.118	p255	3.427	1.209	p256	4.084	1.246
p257	2.632	0.865	p258	3.020	1.110	p259	3.381	1.041
p260	3.073	1.113	p261	3.029	0.942	p262	3.493	1.245
p263	3.121	0.975	p264	3.702	1.192	p265	3.893	1.640
p266	2.941	1.159	p267	2.938	0.971	p268	3.850	1.083
p269	3.196	1.046	p270	3.214	1.229	p271	3.427	1.502
p272	3.255	1.158	p273	2.975	1.067	p274	3.524	1.059
p275	3.072	1.098	p276	3.482	1.053	p277	3.231	1.256
p278	3.332	1.059	p279	2.941	0.914	p280	3.691	1.135
p281	3.250	1.089	p282	3.425	1.369	p283	3.004	0.962
p284	3.011	1.060	p285	2.852	1.081	p286	3.201	1.046
p287	3.057	1.052	p288	3.337	1.144	p292	3.098	0.827
p293	4.340	1.487	p294	3.335	1.058	p295	3.832	1.103
p297	3.884	1.173	p298	4.192	1.136	p299	3.357	0.938
p300	4.380	1.161	p301	4.449	1.018	p302	3.718	0.938
p303	4.506	1.218	p304	3.347	1.017	p305	3.451	1.092
p306	4.939	1.444	p307	4.236	1.055	p308	3.690	1.037
p310	3.607	1.207	p311	3.284	0.979	p312	3.026	1.067
p313	3.648	1.445	p314	3.491	0.983	p315	3.279	1.111
p316	3.418	1.161	p317	3.324	1.010	p318	3.438	0.945
p323	3.721	1.290	p326	3.276	0.982	p329	3.662	1.146
p330	3.897	1.100	p333	2.810	0.925	p334	3.130	0.932
p335	3.850	1.321	p336	2.878	1.147	p339	2.893	1.043
p340	2.558	0.804	p341	3.950	1.229	p343	3.408	1.090
p345	3.236	1.153	p347	3.280	1.180	p351	3.086	0.907
p360	2.898	0.973	p361	2.930	1.016	p362	3.245	1.257
p363	2.776	0.879	p364	3.207	1.346	p374	3.176	1.131
p376	3.027	0.916	s5	4.252	1.241			

Tabela 4.1: Estatísticas de duração por locutor

Tabela 4.2: Resumo dos *vocoders* utilizados na geração de áudio sintético

Código	Arquitetura	Dataset	Freq. (Hz)	Treinado por
v_01	parallel wavegan	ljspeech	22050	kan-bayashi
v_02	melgan	ljspeech	22050	kan-bayashi
v_03	full-band melgan	ljspeech	22050	kan-bayashi
v_04	multi-band melgan	ljspeech	22050	kan-bayashi
v_05	hifi-gan	ljspeech	22050	kan-bayashi
v_06	stylemelgan	ljspeech	22050	kan-bayashi
v_07	parallel wavegan	arctic_slr	16000	kan-bayashi
v_08	parallel wavegan	libritts	24000	kan-bayashi
v_09	multi-band melgan	libritts	24000	kan-bayashi
v_10	hifi-gan	libritts	24000	kan-bayashi
v_11	stylemelgan	libritts	24000	kan-bayashi
v_12	diffwave	ljspeech	22050	lmnt-com
v_13	fastdiff	ljspeech	22050	rongjiehuang
v_14	fastdiff	libritts	24000	rongjiehuang
v_15	vocos	libritts	24000	gemelo-ai
v_16	hifi-gan	libritts	16000	speechbrain
v_17	hifi-gan	librispeech+libritts	16000	speecht5
v_18	melgan	libritts	16000	guan-ting
v_19	waveglow	ljspeech	22050	nvidia
v_20	bigvgan	libritts	24000	nvidia
v_21	bigvgan	large-scale compilation	22050	nvidia

Tabela 4.3: Duração do áudio para largura mínima de 80 pixels

Frequência (Hz)	Duração mínima (s)
16000	1.28
22050	0.93
24000	0.85

Capítulo 5

Resultados

Neste capítulo são apresentados os experimentos realizados para avaliar a capacidade de diferentes arquiteturas de redes neurais convolucionais em distinguir amostras de fala autênticas e sintéticas. Inicialmente, é descrito o conjunto de dados utilizado, incluindo a preparação dos áudios, o processo de geração das amostras sintéticas e a conversão dos sinais em imagens de espectrogramas Mel. Em seguida, detalha-se o treinamento das redes convolucionais, conduzido de modo a comparar o desempenho entre diferentes arquiteturas e verificar sua capacidade de generalização para locutores não vistos e condições de ruído variadas. Por fim, os resultados obtidos são analisados com base em métricas de desempenho e visualizações que evidenciam a robustez dos modelos frente a diferentes cenários de inferência.

5.1 Dataset

Os resultados preliminares apresentados nesta seção foram obtidos a partir do conjunto de dados descrito no Capítulo 4, composto por amostras de áudio *bonafide* e sintéticas. As amostras *bonafide* foram extraídas do conjunto VCTK Corpus [109], utilizado em pesquisas de síntese e reconhecimento de fala, por oferecer gravações de locutores com diferentes características vocais.

Os áudios sintéticos, por sua vez, foram gerados a partir dos mesmos arquivos de referência *bonafide*, utilizando múltiplos *vocoders* neurais, garantindo a diversidade de arquiteturas. Dessa forma, cada locutor original possui versões artificiais correspondentes, o que possibilita a análise direta de como diferentes modelos de *vocoder* afetam as propriedades acústicas do sinal.

Em seguida, foi adicionado ruído branco gaussiano aos sinais de áudio, com o objetivo de aumentar a complexidade da tarefa e aproximar as condições experimentais de cenários mais realistas.

Para a conversão dos áudios em imagens, cada amostra foi transformada em um espectrograma Mel utilizando a biblioteca librosa¹ para Python, por meio da função librosa.feature.melspectrogram. Uma vez que os parâmetros de geração são bem estabelecidos, a padronização e a reprodutibilidade dos experimentos ficam garantidos. A implementação da librosa utiliza o mapeamento entre frequência em hertz e escala Mel conforme descrito na Equação 2.2 [131]. Após a geração dos espectrogramas, as imagens foram normalizadas e armazenadas no formato bitmap PNG, que utiliza compressão sem perda de informação [126], constituindo o conjunto final empregado no treinamento e validação das redes neurais.

Ao final, foi disponibilizado um dataset com 16.759.488 arquivos, distribuídos conforme a Tabela 5.1.

Tabela 5.1: Distribuição do número de áudios por *vocoder*, frequência de amostragem e nível de ruído (SNR).

<i>vocoder</i>	Hz	no noise	30 dB	20 dB	10 dB	0 dB	-10 dB	-20 dB	-30 dB	total
bona	16000	87289	87289	87289	87289	87289	87289	87289	87289	698312
bona	22050	87289	87289	87289	87289	87289	87289	87289	87289	698312
bona	24000	87289	87289	87289	87289	87289	87289	87289	87289	698312
v_01	22050	87289	87289	87289	87289	87289	87289	87289	87289	698312
v_02	22050	87289	87289	87289	87289	87289	87289	87289	87289	698312
v_03	22050	87289	87289	87289	87289	87289	87289	87289	87289	698312
v_04	22050	87289	87289	87289	87289	87289	87289	87289	87289	698312
v_05	22050	87289	87289	87289	87289	87289	87289	87289	87289	698312
v_06	22050	87289	87289	87289	87289	87289	87289	87289	87289	698312
v_07	16000	87289	87289	87289	87289	87289	87289	87289	87289	698312
v_08	24000	87289	87289	87289	87289	87289	87289	87289	87289	698312
v_09	24000	87289	87289	87289	87289	87289	87289	87289	87289	698312
v_10	24000	87289	87289	87289	87289	87289	87289	87289	87289	698312
v_11	24000	87289	87289	87289	87289	87289	87289	87289	87289	698312
v_12	22050	87289	87289	87289	87289	87289	87289	87289	87289	698312
v_13	22050	87289	87289	87289	87289	87289	87289	87289	87289	698312
v_14	24000	87289	87289	87289	87289	87289	87289	87289	87289	698312
v_15	24000	87289	87289	87289	87289	87289	87289	87289	87289	698312
v_16	16000	87289	87289	87289	87289	87289	87289	87289	87289	698312
v_17	16000	87289	87289	87289	87289	87289	87289	87289	87289	698312
v_18	16000	87289	87289	87289	87289	87289	87289	87289	87289	698312
v_19	22050	87289	87289	87289	87289	87289	87289	87289	87289	698312
v_20	24000	87289	87289	87289	87289	87289	87289	87289	87289	698312
v_21	22050	87289	87289	87289	87289	87289	87289	87289	87289	698312
total	—	2094936	2094936	2094936	2094936	2094936	2094936	2094936	2094936	16759488

¹<https://librosa.org>

5.2 Detector Usando Rede Neural Convolucional

5.2.1 Escolha da arquitetura

Para o experimento, todas as imagens de espectrogramas Mel das amostras de áudio, independentemente do nível de ruído, dos microfones e para todos os locutores, foram inicialmente rotuladas de acordo com a tecnologia de *vocoder* que as gerou, usando os códigos da Tabela 4.2.

O ambiente de execução foi configurado com suporte para CUDA 12.0, permitindo o treinamento acelerado dos modelos em GPUs NVIDIA V100², conhecidas por seu alto desempenho em tarefas de *deep learning*. Para este projeto, foi utilizada a versão 3.13.0 do Python. O framework PyTorch³ foi escolhido para a construção e o treinamento das redes neurais.

As imagens de espectrogramas Mel alimentaram as seguintes redes neurais convolucionais: AlexNet [128], ResNet34 [107], ResNet50 [107], VGG16 [129] e VGG19 [129]. Essas arquiteturas foram selecionadas devido à sua eficácia comprovada em tarefas de classificação de imagens. A função de perda utilizada durante o treinamento foi a *Cross-Entropy Loss* [132], comumente aplicada em problemas de classificação multiclasse. O otimizador Adam [133] foi empregado com uma taxa de aprendizado de 0.0001, e o treinamento foi monitorado com parada antecipada (*early stop*), interrompendo o processo após 10 épocas sem melhoria na métrica de validação.

O desempenho de cada rede foi avaliado da seguinte forma: primeiro, verificamos se o treinamento resultou em *overfitting*, reservando 30% das imagens para validação. Observou-se que os modelos treinaram de forma estável, generalizaram bem para a validação, não sofreram com *overfitting* e pararam por não melhoria da métrica escolhida, conforme pode ser visto na Figura 5.1.

Adicionalmente, analisamos as respectivas matrizes de confusão, mostradas nas Figuras 5.2, 5.3, 5.4, 5.5 e 5.6, para identificar padrões de erro, verificar quais classes são mais frequentemente confundidas entre si e evidenciar casos em que determinadas classes não são bem separáveis. Verificou-se que, de modo geral, as redes — inclusive arquiteturas mais antigas, como a AlexNet — conseguem separar adequadamente os *vocoders*, resultando em matrizes de confusão com diagonal principal bem definida. Observa-se, contudo, maior dificuldade na distinção do *vocoder* v_21, associado à tecnologia BigVGAN, especialmente quando treinado em um dataset de larga escala.

Em seguida, os resultados do treinamento foram comparados utilizando a métrica aEER. Nas condições experimentais avaliadas, a arquitetura VGG19 apresentou o menor

²<https://www.nvidia.com/pt-br/data-center/v100/>

³<https://pytorch.org/>

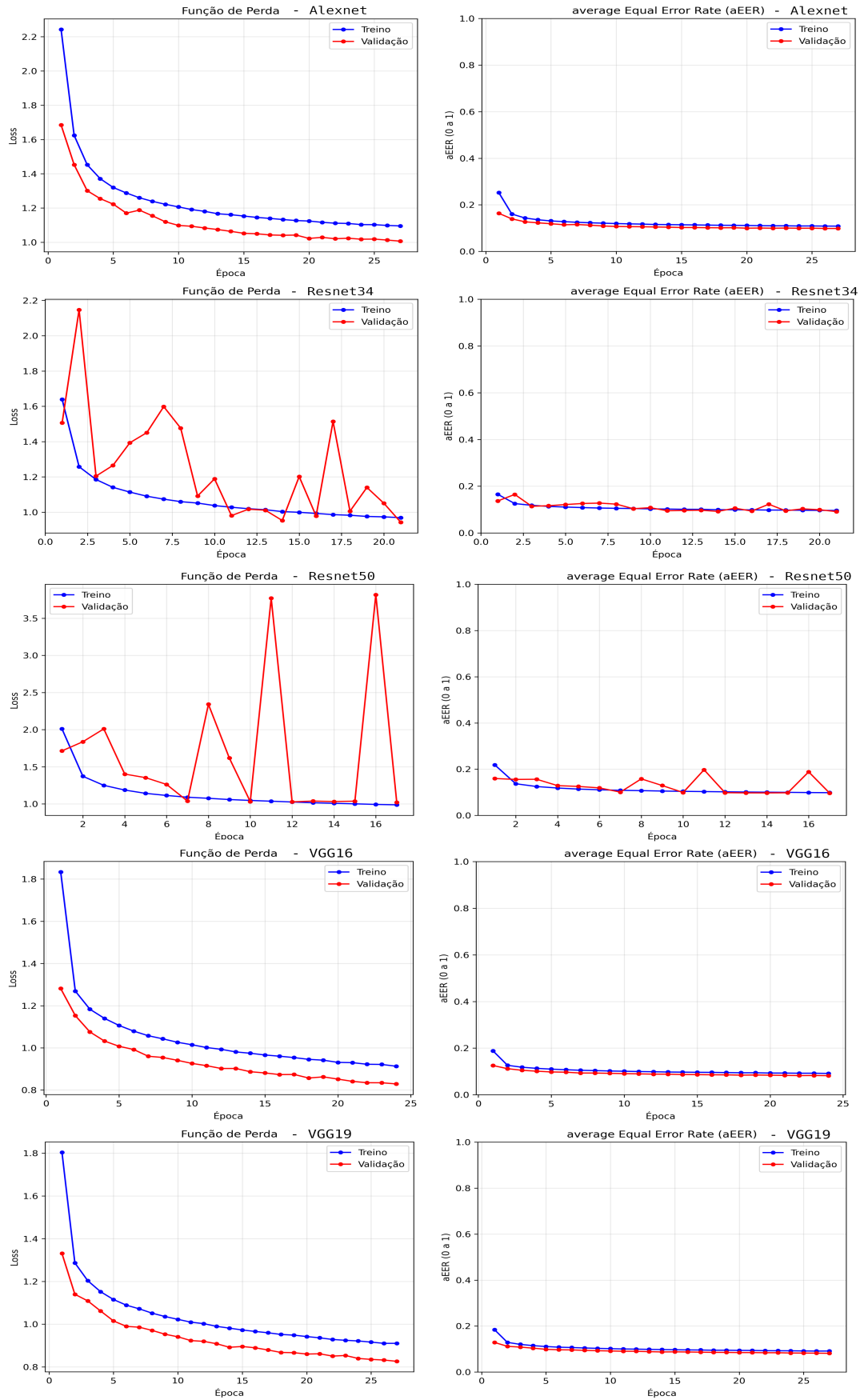


Figura 5.1: Curvas de Loss e de aEER de treinamento para escolha da arquitetura CNN.

valor de aEER (8,628%), sendo, portanto, selecionada para os experimentos subsequentes, conforme apresentado na Tabela 5.2.

Arquitetura	aEER (%)
AlexNet	10.174
ResNet34	9.504
ResNet50	10.056
VGG16	8.672
VGG19	8.628

Tabela 5.2: Desempenho das arquiteturas de redes neurais convolucionais em termos de aEER.

5.2.2 Análise do Processo de Decisão dos Modelos

Com o objetivo de compreender quais regiões dos espectrogramas Mel mais influenciam as decisões das redes neurais convolucionais, foi empregada a técnica de visualização Grad-CAM (*Gradient-weighted Class Activation Mapping*). Essa abordagem permite evidenciar, de forma espacial, quais áreas da imagem contribuem de maneira mais significativa para a classificação realizada pelo modelo [134]. Adicionalmente, foi utilizada uma técnica de oclusão de canais (*channel ablation*) [135], na qual canais da última camada convolucional são removidos individualmente, permitindo avaliar quantitativamente a contribuição de cada canal para a decisão da rede neural.

Grad-CAM

O Grad-CAM é um método que permite identificar as regiões da entrada que mais contribuíram para a predição de uma determinada classe em redes neurais convolucionais. Para isso, são calculados os gradientes da saída da classe de interesse em relação aos *feature maps* de uma camada convolucional profunda da rede. Neste trabalho, foi utilizada a última representação convolucional da arquitetura VGG19, composta por 512 *feature maps* com resolução espacial de 7×7 . Dessa forma, cada *feature map* A^k ($k \in 1, \dots, 512$) consiste em uma matriz 7×7 , cujos elementos representam a intensidade de ativação de um determinado filtro convolucional em diferentes regiões da entrada.

A importância de cada *feature map* A^k para a classe c é estimada pela média global dos gradientes da saída da rede em relação às ativações desse mapa:

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k}, \quad (5.1)$$

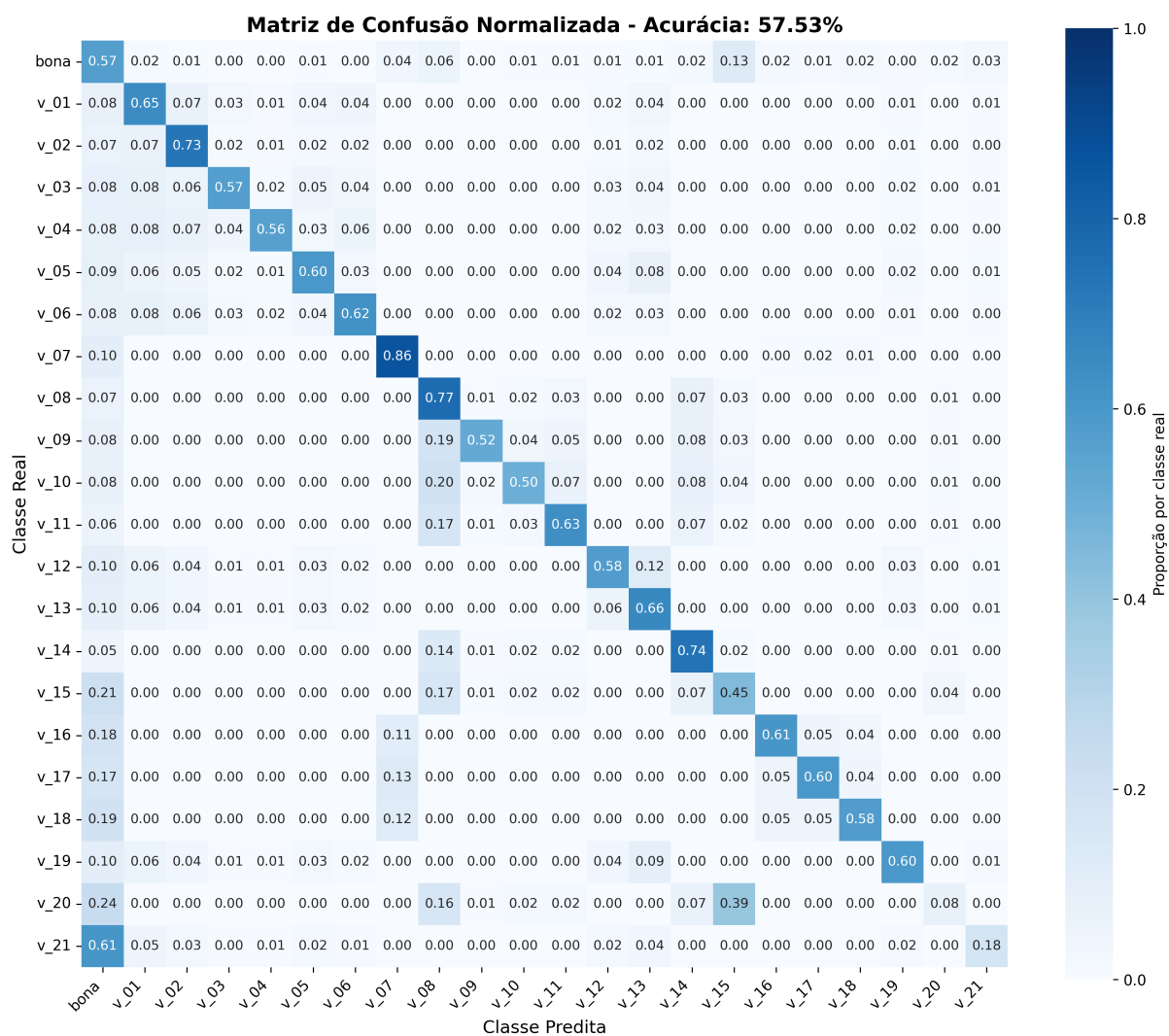


Figura 5.2: Matriz de confusão normalizada - AlexNet

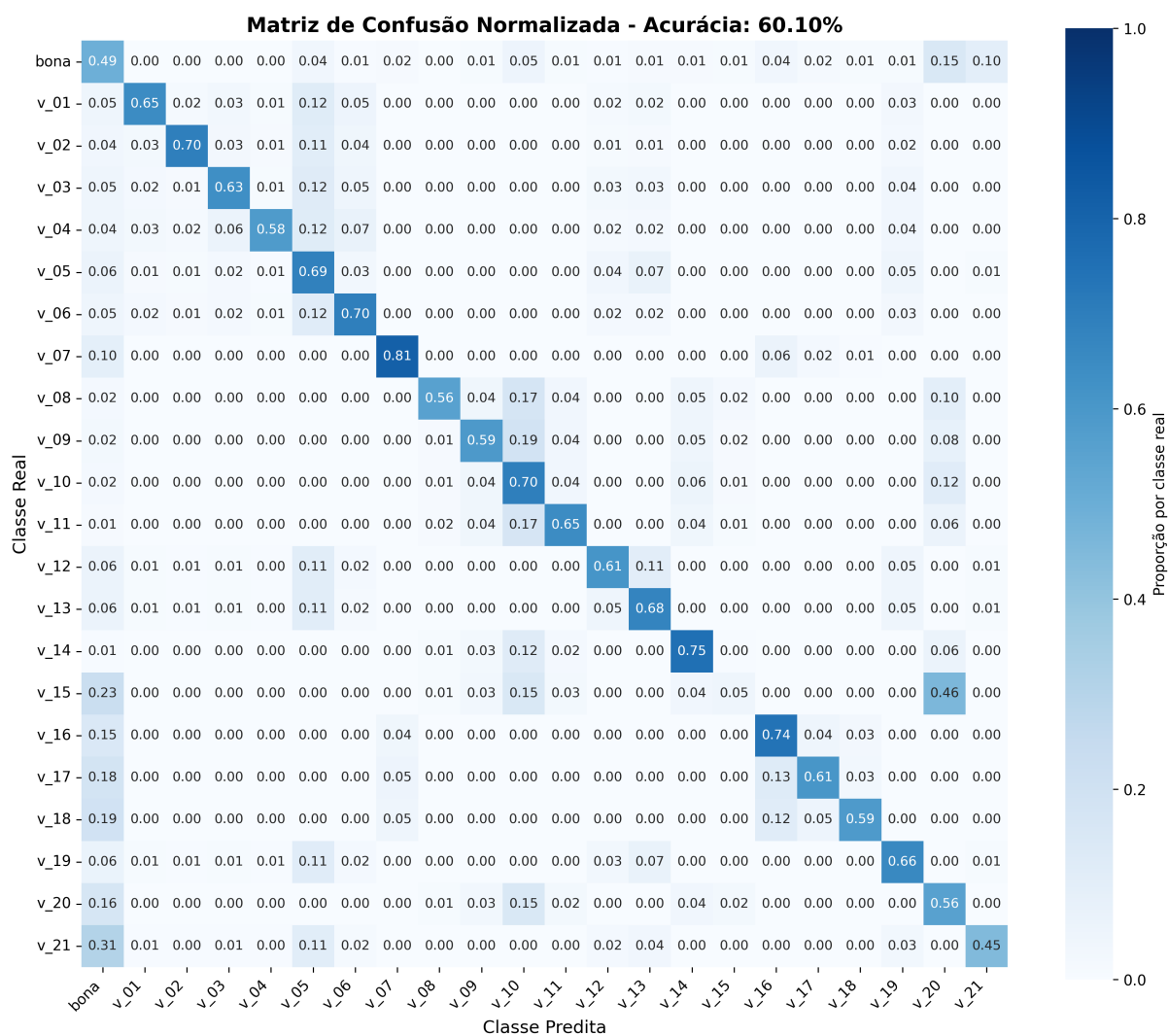


Figura 5.3: Matriz de confusão normalizada - ResNet34

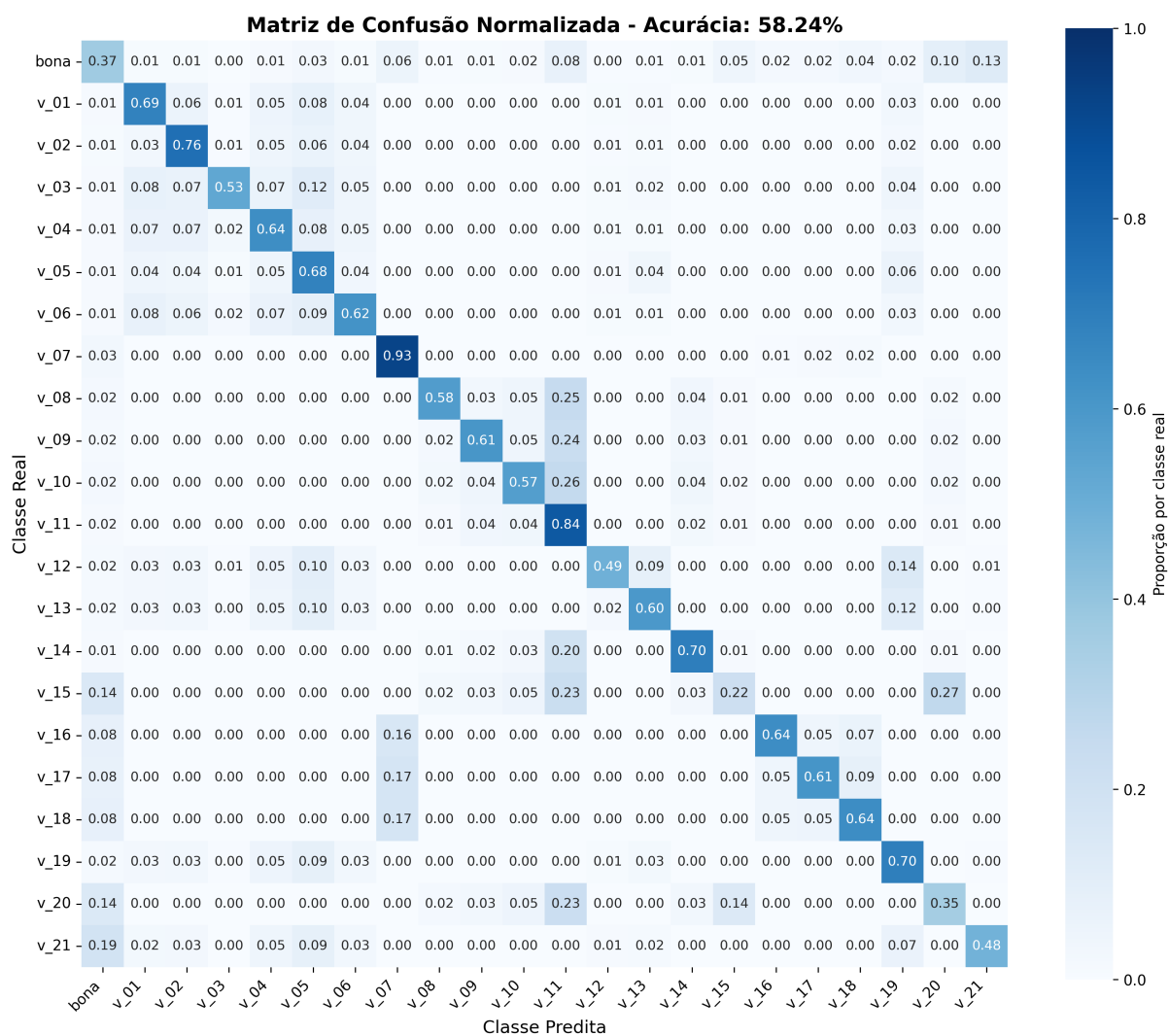


Figura 5.4: Matriz de confusão normalizada - ResNet50

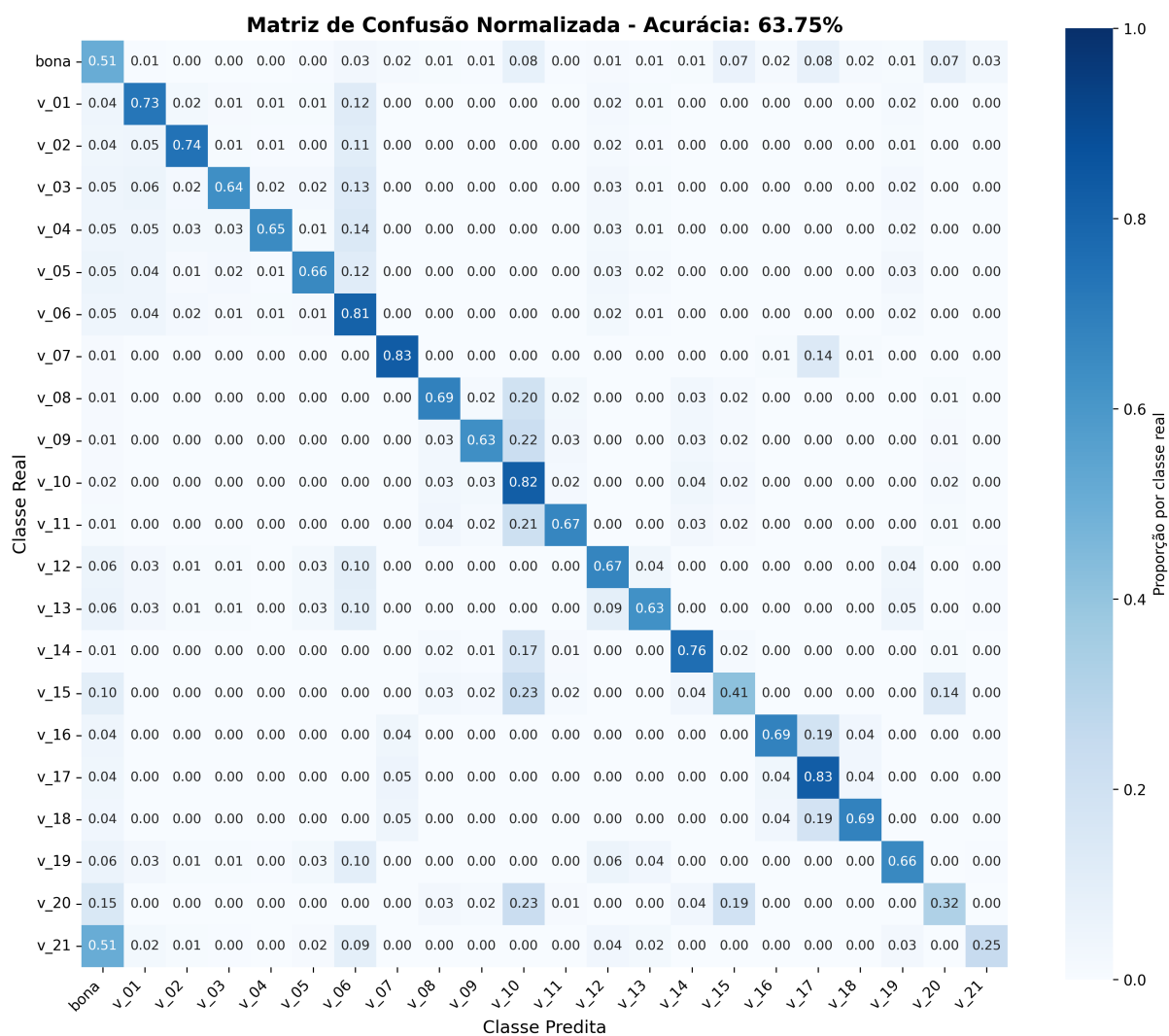


Figura 5.5: Matriz de confusão normalizada - VGG16

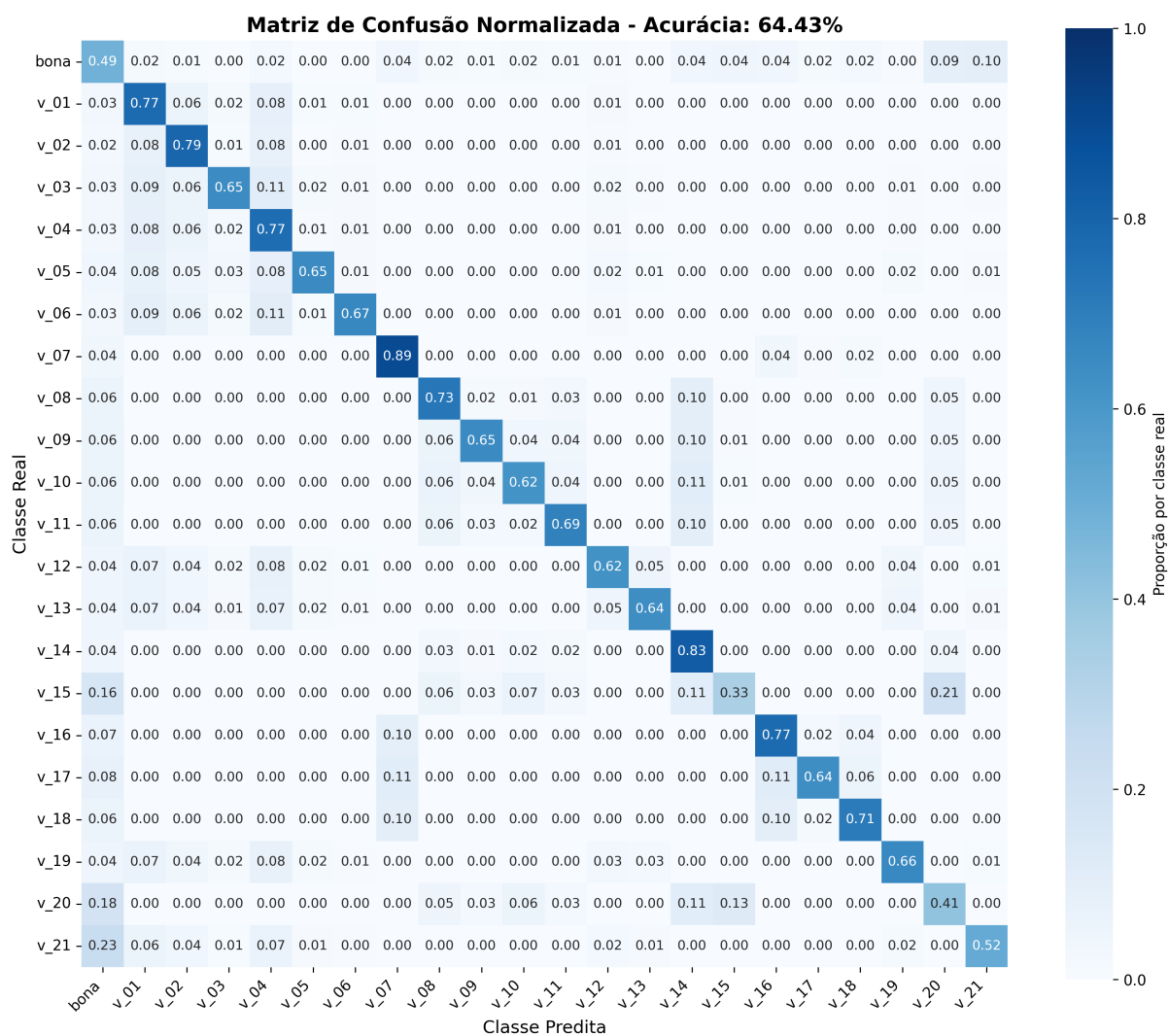


Figura 5.6: Matriz de confusão normalizada - VGG19

em que y^c representa a saída da rede para a classe analisada, A_{ij}^k corresponde à ativação da posição espacial (i, j) do k -ésimo *feature map*, e Z corresponde ao número total de posições espaciais do mapa. No caso da VGG19, cada mapa possui dimensão 7×7 , de modo que $Z = 49$.

Os pesos α_k^c são então utilizados para combinar os *feature maps*, produzindo um mapa de relevância dado por:

$$L_{\text{Grad-CAM}}^c = \text{ReLU} \left(\sum_k \alpha_k^c A^k \right). \quad (5.2)$$

O valor de cada posição de $L_{\text{Grad-CAM}}^c$ será elevado quando corresponder a regiões que apresentem simultaneamente altas ativações nos *feature maps* e forte contribuição para a classe analisada. Como cada mapa A^k possui dimensão 7×7 , o mapa de relevância $L_{\text{Grad-CAM}}^c$ também é inicialmente uma matriz 7×7 . Assim, cada uma de suas 49 posições não corresponde a um pixel específico da imagem de entrada, mas sim a uma região espacial mais ampla observada pela rede. Após seu cálculo, o mapa é redimensionado por interpolação espacial (*upsampling*) para o tamanho original da imagem e sobreposto à entrada na forma de um mapa de calor, permitindo visualizar de maneira aproximada quais regiões exerceram maior influência sobre a decisão da rede neural.

A inspeção dos mapas de ativação revelou que as regiões de maior relevância concentram-se predominantemente em áreas de baixa energia espectral, associadas a trechos de silêncio ou pausas na fala, bem como em regiões de alta frequência. Esse resultado indica que os modelos não baseiam suas decisões exclusivamente nas características fonéticas da fala, mas também em artefatos presentes fora das regiões de maior energia.

Esse achado justifica a decisão de não empregar técnicas de remoção de silêncio, baseadas em *Voice Activity Detection* (VAD), durante o pré-processamento dos dados. Embora o VAD seja amplamente utilizado para eliminar segmentos não informativos em tarefas de reconhecimento de fala, os resultados obtidos sugerem que tais regiões carregam informações relevantes para a distinção entre sinais autênticos e sintéticos. De fato, trabalhos anteriores já indicaram que a inclusão de segmentos de silêncio pode aumentar o desempenho de modelos na detecção de *deepfakes* de áudio [110]. Ademais, a relevância observada em regiões de alta frequência está alinhada com achados da literatura [1], que indicam que artefatos introduzidos por *vocoders* neurais tendem a se manifestar nessas bandas espectrais. Tais artefatos tornam-se mais evidentes por estarem menos sujeitos ao mascaramento pelas componentes espectrais dominantes do sinal de fala.

A Figura 5.7 apresenta exemplos de espectrogramas Mel com as respectivas regiões de ativação destacadas, evidenciando a concentração das áreas de atenção fora dos segmentos de fala mais energéticos.

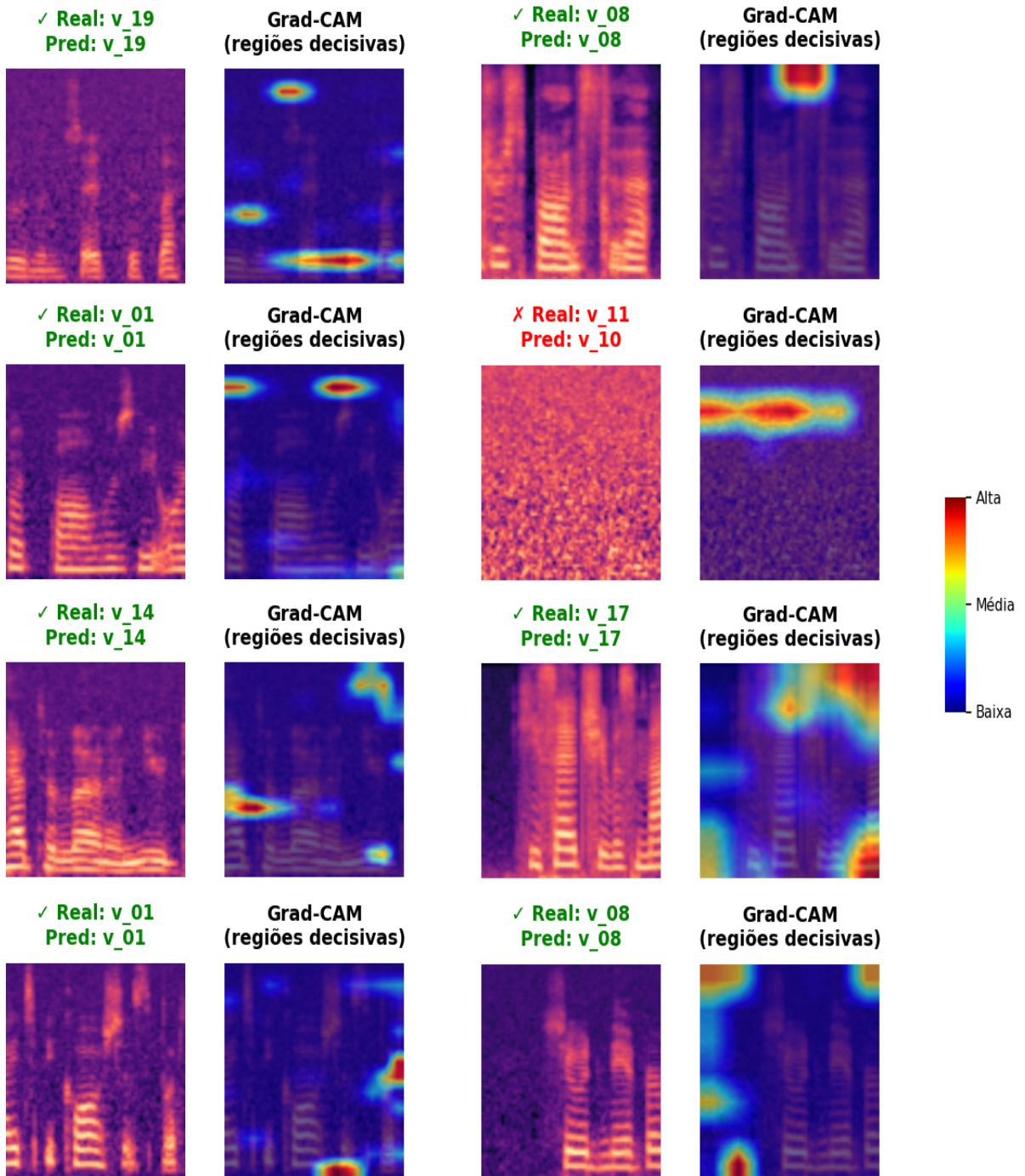


Figura 5.7: Exemplos de espectrogramas Mel com mapas de ativação obtidos por Grad-CAM, destacando as regiões mais relevantes para a decisão das redes neurais.

Oclusão de canais

Para investigar a contribuição funcional dos canais internos da rede convolucional, foi realizado um teste de ablação [135] utilizando os 512 *feature maps* da última camada convolucional da arquitetura VGG19. Inicialmente, as ativações produzidas por cada *feature map* foram extraídas individualmente, resultando em representações espaciais bidimensionais associadas às diferentes regiões de ativação da rede. Uma amostra dessas ativações é apresentada na Figura 5.8b, na qual cada quadrícula representa um dos 512 *feature maps* analisados. Esses mapas refletem padrões internos aprendidos pela rede durante o treinamento, correspondendo a diferentes características espectrais utilizadas no processo de classificação das imagens.

Para cada amostra, a importância funcional de cada *feature map* foi estimada por meio de uma estratégia de ablação individual dos canais convolucionais. Nesse procedimento, a saída original da classe correspondente (y_c) foi comparada à saída obtida após a remoção do *feature map* k ($y_c^{(-k)}$), conforme descrito na Equação 5.3. A importância do canal foi então definida como:

$$I_k = y_c - y_c^{(-k)} \quad (5.3)$$

em que I_k representa a contribuição funcional do *feature map* k para a predição da classe c . Valores elevados de I_k indicam que a remoção do canal produz uma redução significativa na confiança da predição, sugerindo maior relevância daquele *feature map* para a decisão da rede. A partir desse procedimento, cada amostra passou a ser representada por um vetor de importância contendo 512 valores, correspondentes às contribuições individuais dos *feature maps* da última camada convolucional. Esses vetores foram então utilizados como *embeddings* funcionais da CNN, permitindo investigar a organização interna das representações aprendidas pelo modelo. Em seguida, foi aplicada a técnica UMAP (*Uniform Manifold Approximation and Projection*) [136] para projeção dos vetores em um espaço bidimensional, preservando relações de similaridade entre os padrões de importância de diferentes amostras. A projeção bidimensional obtida é apresentada na Figura 5.9, na qual cada ponto representa uma amostra individual, colorida de acordo com o *vocoder* correspondente, enquanto a transparência foi utilizada para indicar diferentes níveis de ruído adicionados aos sinais.

A projeção bidimensional obtida pelo UMAP revelou a formação de agrupamentos associados aos diferentes *vocoders*, indicando que os vetores de importância derivados do teste de ablação preservaram informações discriminativas. Diversos *vocoders* apresentaram regiões compactas e relativamente separadas no espaço 2D, sugerindo que diferentes

arquitecturas de síntese de voz induzem padrões distintos de relevância funcional nos *feature maps* da última camada convolucional da VGG19.

Também foi possível observar que alguns *vocoders* apresentaram maior proximidade entre si no espaço UMAP, enquanto outros permaneceram mais isolados. Em particular, os *vocoders* v_15, v_20 e v_21 apresentaram regiões parcialmente sobrepostas ou próximas ao agrupamento correspondente aos áudios *bonafide* (*bona*), indicando maior similaridade entre os padrões de importância funcional produzidos por essas amostras. Esse comportamento sugere que as representações internas geradas pela CNN para essas *vocoders* compartilham características mais próximas daquelas observadas em áudios autênticos, dificultando a separação completa entre os grupos. Em contraste, outros *vocoders* apresentaram agrupamentos mais compactos e distantes da região ocupada pelas amostras *bonafide*, indicando maior distinção nas representações internas aprendidas pelo modelo. Além disso, observou-se que níveis crescentes de ruído contribuíram para maior dispersão espacial das amostras.

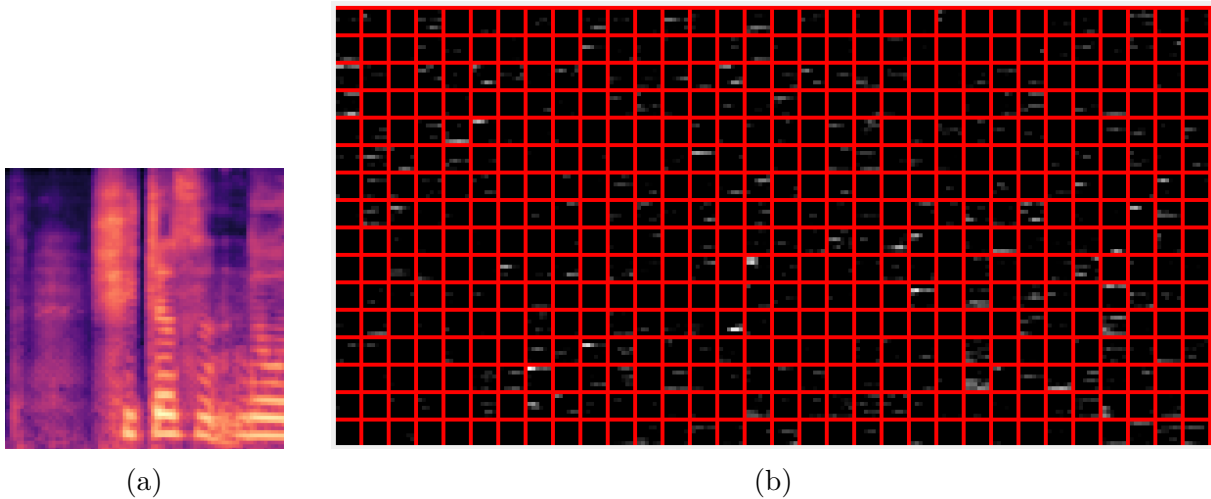


Figura 5.8: (a) Exemplo de espectrograma Mel gerado a partir de uma amostra de áudio sintetizada pelo *vocoder* v_11 sem adição de ruído. (b) Visualização dos 512 *feature maps* 7x7 extraídos da última camada convolucional da arquitetura VGG19 para essa amostra.

5.2.3 Testes de Generalização

Após a análise dos resultados obtidos nos experimentos anteriores, torna-se necessário avaliar a capacidade de generalização dos modelos frente a novos cenários. Para isso, foram realizados novos experimentos nos quais *vocoders*, locutores e microfones foram excluídos do treinamento e utilizados apenas durante os testes de generalização, permitindo verificar a capacidade das redes em detectar áudios sintéticos gerados por tecnologias e indivíduos não observados previamente.

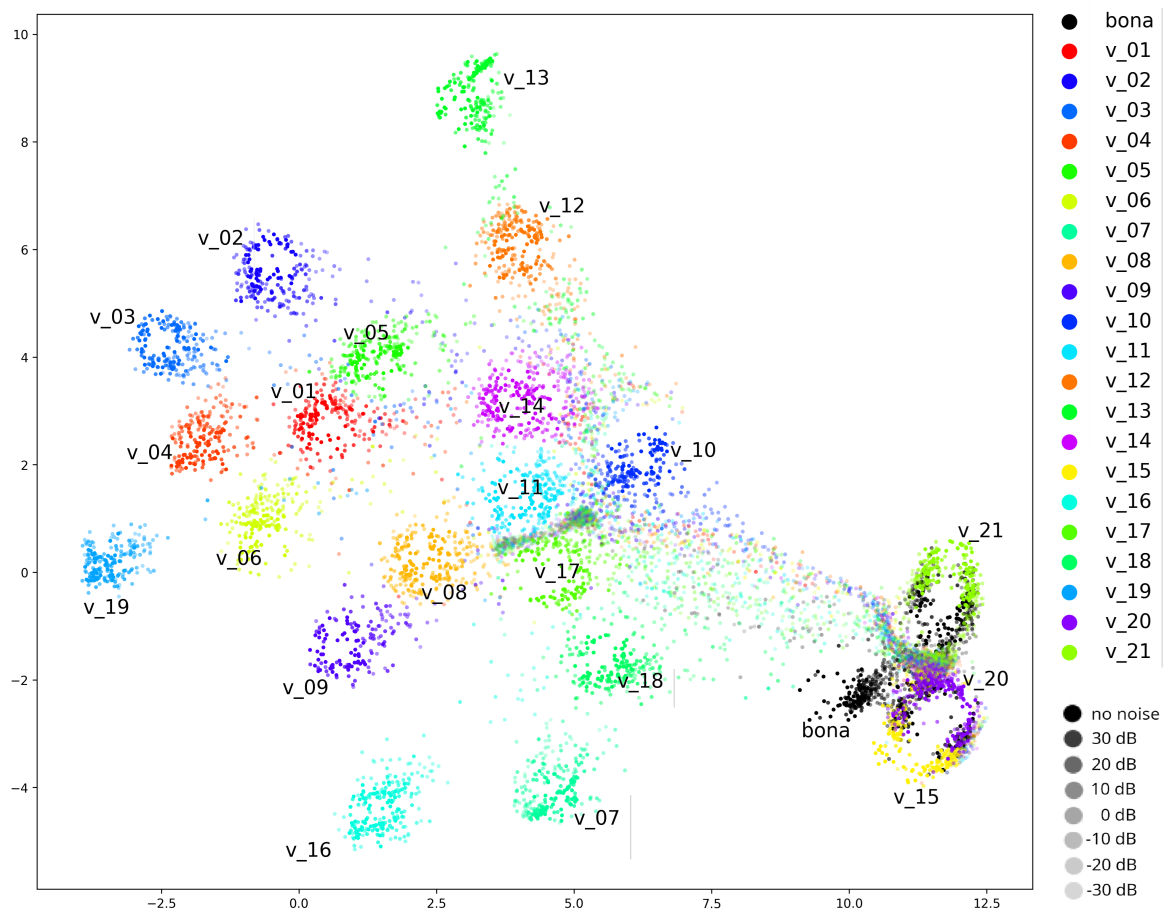


Figura 5.9: Projeção bidimensional obtida por UMAP (*Uniform Manifold Approximation and Projection*) a partir dos vetores de importância gerados pelo teste de ablação dos 512 *feature maps* da última camada convolucional da arquitetura VGG19.

Arquitetura

Para este treinamento, a camada de saída da rede VGG19 não utiliza a formulação clássica com *softmax*, sendo adotada a função sigmoide aplicada independentemente a cada neurônio de saída. Essa escolha está alinhada à modelagem do problema como uma tarefa de classificação multirrótulo, na qual cada classe é tratada como uma decisão binária independente. Essa mudança na última camada se faz necessária uma vez que existe a possibilidade de a rede entender que o espectrograma Mel foi derivado de um áudio sintético que não fez parte do treinamento, fazendo com que todas as saídas se aproximem de zero, o que não seria possível com o *softmax*.

Em relação à função de perda, é adotada uma formulação compatível com esse esquema, na qual os *logits* produzidos pela rede são utilizados diretamente em uma função do tipo *Binary Cross-Entropy with Logits* (BCEWithLogitsLoss), que incorpora internamente a aplicação da função sigmoide. No contexto dessa função de perda, foi empregada uma ponderação diferenciada entre as classes, na qual a classe bona concentra 50% da contribuição total da perda, enquanto os 50% restantes são distribuídos entre as demais classes, que, em conjunto, representam as amostras sintéticas. Dessa forma, busca-se equilibrar a importância relativa entre exemplos reais e sintéticos durante o processo de otimização.

Assim, a classe bona assume um papel central, sendo a única diretamente responsável pela distinção entre áudios reais e sintéticos. Para a decisão final, será aplicado um limiar (*threshold*) sobre a probabilidade associada à classe bona, o qual ainda será definido.

Treinamento

Após estabelecer os detalhes da arquitetura, foram realizados 21 novos treinamentos, com todos os níveis de ruídos, configurados da seguinte forma: três *vocoders*, 11 locutores e um microfone foram excluídos de cada uma das sessões de treinamento, conforme pode ser visto na Tabela 5.3.

Observou-se que, assim como nos treinamentos da etapa anterior, para decidir sobre qual modelo CNN utilizar, não houve *overfitting* e a parada se deu por não melhoria da métrica escolhida. A Figura 5.10 exemplifica, para os treinamentos 1 e 2, a evolução da função de perda e do aEER para cada final época sobre os dados de validação. Para todos os 21 treinamentos, veja o Anexo I.

A performance desses modelos quando testados com os dados de validação pode ser inferida através da observação das curvas Receiver Operating Characteristic (ROC). A Figura 5.11 exemplifica, para os treinamentos 1 e 2, as curvas ROC, calculadas sobre esses dados. Para todos os 21 treinamentos, veja o Anexo II. Observa-se que todas as

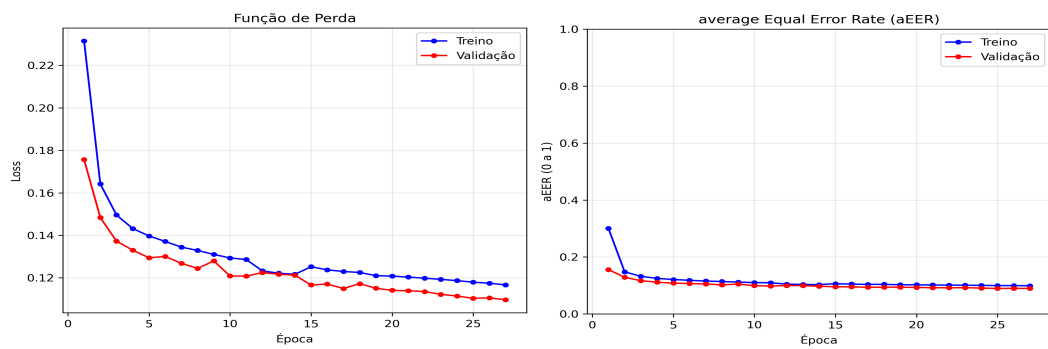
Treinamento	Vocoders externos	Mics externos	Locutores externos
1	v_01, v_04, v_08	mic1	p225, p247, p259, p272, p284, p300, p314, p333, p347, p361, s5
2	v_07, v_15, v_19	mic2	p226, p240, p251, p264, p273, p286, p301, p312, p329, p343, p374
3	v_10, v_14, v_18	mic1	p227, p231, p243, p255, p268, p279, p292, p304, p318, p336, p362
4	v_13, v_17, v_21	mic2	p228, p236, p246, p258, p271, p283, p297, p308, p323, p340, p363
5	v_05, v_09, v_12	mic1	p229, p237, p249, p262, p274, p287, p302, p315, p330, p345, p376
6	v_03, v_06, v_11	mic2	p230, p238, p250, p263, p275, p288, p303, p316, p334, p351, p360
7	v_02, v_16, v_20	mic1	p231, p241, p254, p266, p276, p293, p305, p317, p335, p339, p364
8	v_01, v_11, v_16	mic2	p232, p243, p257, p267, p278, p294, p306, p311, p326, p341, p362
9	v_04, v_07, v_20	mic1	p233, p244, p259, p270, p279, p295, p307, p314, p329, p343, p361
10	v_08, v_10, v_15	mic2	p234, p245, p260, p271, p280, p297, p308, p312, p330, p345, p374
11	v_13, v_14, v_19	mic1	p236, p247, p262, p272, p281, p298, p309, p315, p333, p347, p360
12	v_05, v_17, v_18	mic2	p237, p248, p263, p273, p282, p299, p310, p316, p334, p351, s5
13	v_03, v_09, v_21	mic1	p238, p249, p264, p274, p283, p300, p311, p317, p335, p340, p362
14	v_02, v_06, v_12	mic2	p239, p250, p265, p275, p284, p301, p312, p318, p336, p343, p363
15	v_01, v_06, v_21	mic1	p240, p251, p266, p276, p285, p302, p313, p314, p333, p347, p374
16	v_04, v_11, v_12	mic2	p241, p252, p267, p277, p286, p303, p314, p315, p334, p351, p360
17	v_07, v_08, v_16	mic1	p243, p253, p268, p278, p287, p304, p315, p316, p335, p339, p361
18	v_10, v_13, v_20	mic2	p244, p254, p269, p279, p288, p305, p316, p317, p336, p340, p362
19	v_05, v_14, v_15	mic1	p245, p255, p270, p280, p292, p306, p317, p318, p329, p343, p363
20	v_03, v_17, v_19	mic2	p246, p256, p271, p281, p293, p307, p318, p323, p330, p345, p374
21	v_02, v_09, v_18	mic1	p247, p257, p272, p282, p294, p308, p311, p326, p334, p347, p360

Tabela 5.3: Configuração dos treinamentos com *vocoders*, microfones e locutores externos.

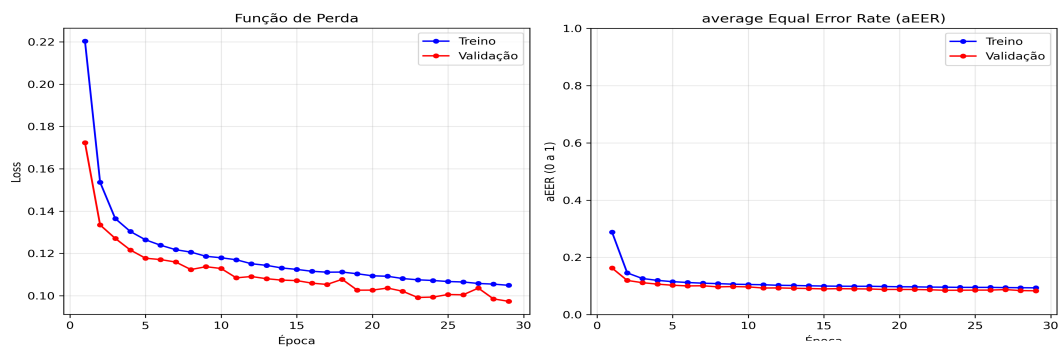
curvas apresentam comportamento consistente, com inclinação acentuada próxima à origem e tendência a se aproximar do canto superior esquerdo do gráfico, o que indica boa capacidade discriminativa dos modelos. No entanto, nota-se um padrão claro de degradação à medida que o nível de ruído aumenta: as curvas tornam-se progressivamente mais próximas da diagonal de decisão aleatória, refletindo a piora na qualidade das probabilidades associadas à saída bona. Esse comportamento é corroborado pelos valores de AUC, que diminuem conforme o ruído é intensificado. Por outro lado, os modelos treinados com menores níveis de ruído apresentam curvas mais próximas do ideal e maiores AUCs, evidenciando melhor desempenho e maior robustez nas condições de validação.

Tipos de *Threshold*

Com os modelos devidamente treinados, procede-se à avaliação de seus desempenhos em condições externas ao processo de treinamento, com o objetivo de analisar suas capacidades de generalização. Durante a etapa de inferência, a partir de uma imagem de espectrograma Mel, considera-se especificamente a saída associada à classe bona como critério para determinar se o áudio que originou a imagem é legítimo ou sintético. Idealmente, valores iguais a um indicam maior probabilidade de o áudio ser legítimo, enquanto valores iguais a zero sugerem tratar-se de um áudio sintético. No entanto, devido à natureza do modelo, essa saída corresponde a um valor contínuo no intervalo $[0,1]$, tornando necessária a definição de um limiar (*threshold*) para a tomada de decisão: valores acima desse limiar são classificados como legítimos, enquanto valores abaixo são classificados como sintéticos.

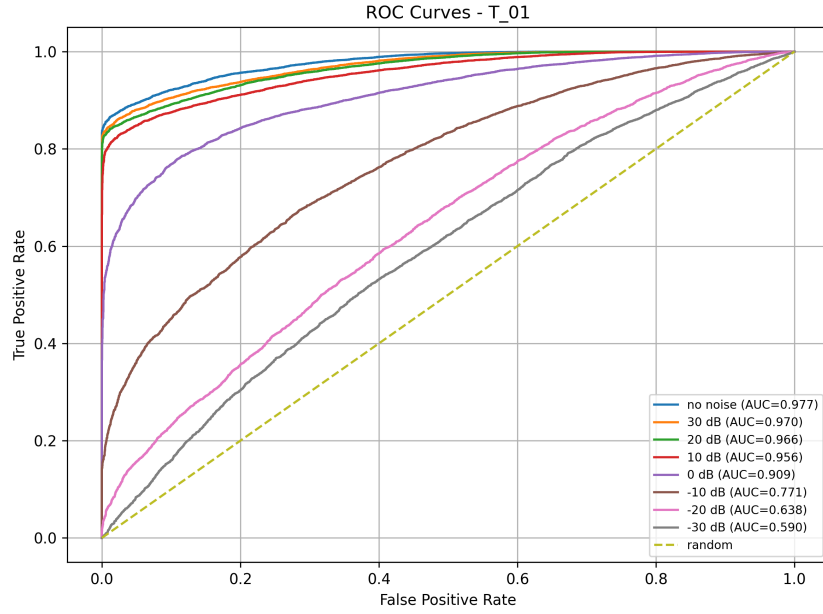


(a) Treinamento 1

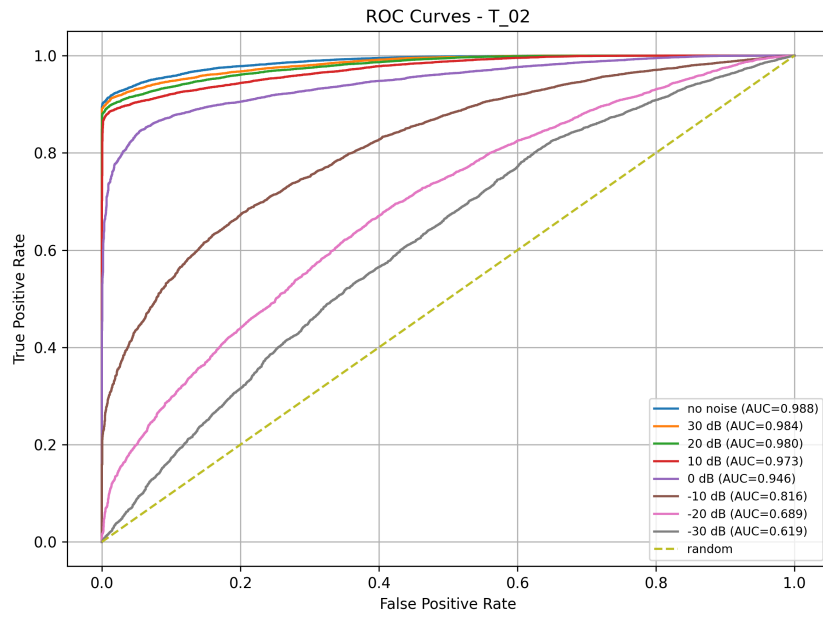


(b) Treinamento 2

Figura 5.10: Curvas de Loss e de aEER dos treinamentos 1 e 2 para criação dos modelos a serem usados em testes de generalização. Para todos os 21 treinamentos, veja Anexo I.



(a) Treinamento 1



(b) Treinamento 2

Figura 5.11: Curvas ROC da saída *bonafide* dos modelos gerados a partir dos treinamentos 1 e 2 a serem usados em testes de generalização. Para todos os 21 treinamentos, veja Anexo II.

Destaca-se que a definição desse limiar deve ser realizada de forma criteriosa, sendo inadequado utilizar, por exemplo, o valor de *threshold* correspondente ao EER obtido em um conjunto de dados externo ao de treinamento, como aquele composto por *vocoders* externos, locutores externos e microfone externo. Tal prática poderia inflar artificialmente o desempenho do modelo frente a dados desconhecidos. Assim, o limiar é tratado como um hiperparâmetro que deve ser decidido antes da entrega do modelo à produção. Nesse contexto, foram avaliadas quatro estratégias distintas para a definição do *threshold*:

- ***Threshold* em 0.5**

Nesta abordagem, utiliza-se um limiar fixo de 0.5 sobre a probabilidade da classe bona. Assim, amostras com $p_{\text{bona}} \geq 0.5$ são classificadas como legítimas, enquanto valores inferiores indicam áudios sintéticos. Trata-se de uma escolha simples e independente dos dados, porém não leva em consideração o balanceamento entre falsos positivos e falsos negativos, podendo resultar em desempenho subótimo em cenários reais.

- ***Threshold* sob EER dos dados de validação (completo)**

A definição do limiar a partir do EER dos dados de validação constitui uma escolha particularmente adequada quando se assume que os custos de falso positivo e falso negativo são equivalentes, pois, nesse ponto, as duas taxas de erro se igualam, estabelecendo um critério de decisão balanceado. Além disso, ao considerar conjuntamente todas as condições presentes na validação — incluindo diferentes níveis de ruído —, esse procedimento incorpora a variabilidade do problema e produz um *threshold* alinhado ao comportamento médio do modelo. Sob a hipótese usual de que conjuntos de dados externos apresentam estatísticas semelhantes às do conjunto utilizado no treinamento e validação, espera-se que esse limiar generalize de forma consistente, mantendo um compromisso equilibrado entre os tipos de erro também em cenários não vistos.

- ***Threshold* sob EER dos dados de validação (sem ruído)**

A definição do limiar com base no EER dos dados de validação sem ruído busca evitar que amostras muito degradadas influenciem de forma excessiva o ponto de decisão do modelo. Quando diferentes níveis de ruído são considerados simultaneamente, amostras altamente ruidosas podem deslocar o *threshold* para uma região menos adequada à classificação de sinais com melhor qualidade. Assim, o uso das amostras sem ruído permite estimar um limiar a partir de uma condição mais controlada, preservando a separação entre as classes sem que ela seja excessivamente afetada pela degradação do sinal.

- ***Threshold* média das médias (*logits*)**

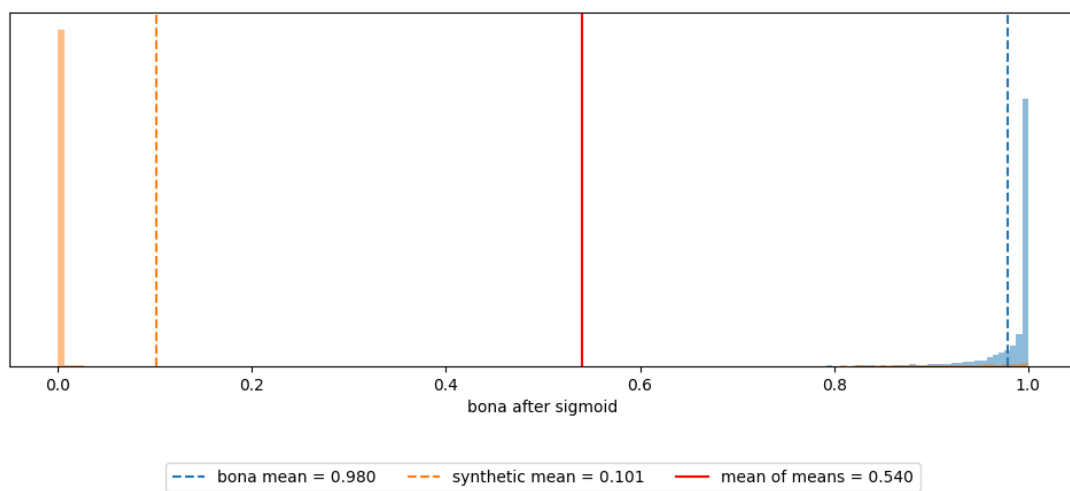
Nesta abordagem, calcula-se inicialmente a média dos valores de *logits* — valores produzidos pelo modelo antes da aplicação da função de ativação — da última camada associados a amostras *bonafides* e sintéticas, sem ruído, considerando as amostras de validação de cada um desses dois grupos. Em seguida, define-se o limiar como a média entre essas duas médias, posicionando-o em ponto intermediário entre as distribuições das classes no espaço de decisão do modelo.

Como a função sigmoide introduz uma transformação altamente não linear nas saídas do modelo, as probabilidades resultantes tendem a se concentrar nos extremos do intervalo $[0, 1]$. Ao utilizar os *logits* — isto é, as saídas do modelo antes da aplicação da função sigmoide — evita-se essa compressão, preservando uma variação mais contínua e regular entre os valores. Dessa forma, a definição do *threshold* passa a se apoiar em uma representação mais fiel da separação entre as amostras *bonafide* e sintéticas. Na Figura 5.12, são apresentadas distribuições das saídas de um modelo para os dados *bonafides* e sintéticos, construídas de forma balanceada. Observa-se, na Figura 5.12(a), que os valores se concentram próximos aos limites do intervalo $[0, 1]$, uma vez que correspondem às probabilidades (isto é, às saídas após a aplicação da função sigmoide). Como consequência, a média das médias das distribuições fica sempre próxima a 0.5, aproximando um eventual uso deste valor ao primeiro tipo de *threshold* apresentado nesta seção. Em contraste, a Figura 5.12(b), baseada nos *logits*, apresenta distribuições mais regulares, sem efeito de saturação, o que fundamenta a adoção de *logits* em detrimento das probabilidades para a definição do cálculo do limiar.

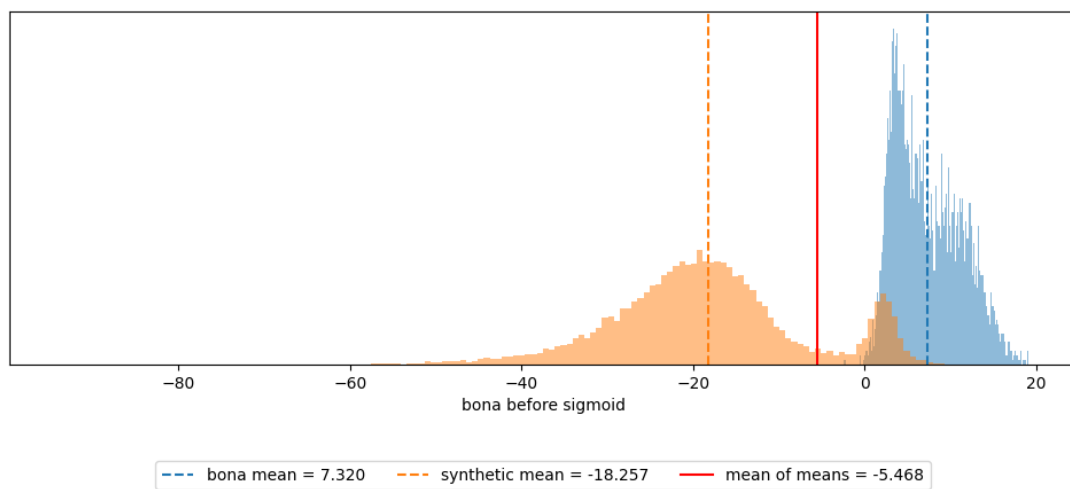
Desempenho

Agora que dispomos dos 21 modelos devidamente treinados, bem como das quatro estratégias estabelecidas para a definição dos *thresholds*, torna-se possível avaliar a capacidade de generalização das redes frente a espectrogramas Mel não contemplados durante o treinamento. Para tanto, os modelos são avaliados em um cenário de generalização composto por *vocoders* externos, além de variações de microfone e locutores não observados previamente. Para cada configuração de treinamento, são geradas três curvas, cada uma representando o desempenho do modelo diante de um *vocoder* externo distinto, considerando todos os locutores externos, sob diferentes condições de ruído.

Para avaliar a performance dos modelos foi considerado o *Balanced Error Rate* (BER) [137], que consiste em uma métrica de avaliação que leva em conta, de forma equilibrada, os erros cometidos em cada classe. Em problemas de classificação binária, o BER é



(a)



(b)

Figura 5.12: Ilustração de distribuições de (a) probabilidades (após sigmoide) e (b) logits (antes da sigmoide) para amostras de validação *bonafide* e sintéticas (dados colhidos e inferidos da partição de validação do Treinamento 19).

definido como a média entre a taxa de falsos positivos (*False Positive Rate*) e a taxa de falsos negativos (*False Negative Rate*), sendo útil em cenários com desbalanceamento entre classes. No presente caso, em que há uma quantidade maior de amostras sintéticas em relação às amostras *bonafide* no dataset, o uso do BER torna-se necessário. Repare que nos casos em que a taxa de falsos positivos é igual à taxa de falsos negativos, o BER se reduz ao EER. A métrica BER evita que o desempenho seja superestimado ou subestimado por uma classe majoritária, fornecendo uma visão mais justa da capacidade do modelo em distinguir corretamente entre as diferentes categorias.

De posse dos dados provenientes das inferências, foi possível construir a Tabela 5.4, que mostra o desempenho médio dos modelos considerando diferentes critérios de definição de limiar (*threshold*): valor fixo de 0,5; limiar determinado a partir do *Equal Error Rate* (EER) nos dados de validação completos; limiar baseado no EER dos dados de validação sem ruído; e limiar obtido pela média das médias dos *logits*.

Tabela 5.4: Comparação da taxa média de erro balanceado entre níveis de ruído e estratégias de definição de *threshold*.

Noise	Thres. (p=0.5)		EER val (all)		EER val (clean)		Mean of means	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
sem ruído	0.197	0.161	0.262	0.151	0.293	0.141	0.162	0.171
30 dB	0.197	0.166	0.263	0.167	0.299	0.165	0.170	0.174
20 dB	0.182	0.170	0.232	0.169	0.271	0.172	0.174	0.183
10 dB	0.185	0.175	0.204	0.168	0.237	0.168	0.192	0.192
0 dB	0.235	0.169	0.237	0.157	0.261	0.150	0.275	0.185
-10 dB	0.365	0.149	0.353	0.140	0.389	0.111	0.417	0.107
-20 dB	0.471	0.060	0.437	0.173	0.471	0.132	0.494	0.012
-30 dB	0.500	0.002	0.467	0.191	0.496	0.147	0.500	0.000

Para percepção visual, construiu-se a Figura 5.13, que mostra os dados da Tabela 5.4. Neste gráfico o eixo das abscissas representa o aumento de ruído e o das ordenadas, o BER para cada estratégia de definição de *threshold*.

A partir dos resultados, observa-se que o aumento do nível de ruído impacta diretamente o desempenho dos modelos, elevando progressivamente o BER em todas as estratégias avaliadas, como esperado em cenários de degradação do sinal. Em condições de baixo ruído (sem ruído até 10 dB), nota-se que as estratégias baseadas em *threshold* fixo (0,5) e na média das médias dos *logits* apresentam melhor desempenho competitivo, com ligeira vantagem para esta última no cenário sem ruído. Por outro lado, as abordagens baseadas no EER dos dados de validação tendem a apresentar desempenho inferior de forma consistente, sugerindo menor capacidade de generalização quando o limiar é definido desta forma.

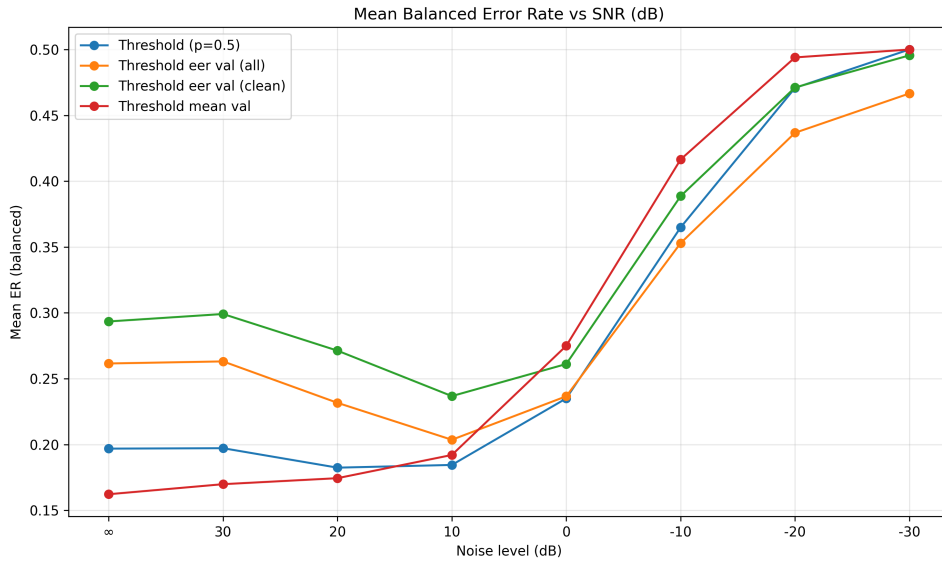


Figura 5.13: Curvas da taxa de erro balanceado (*Balanced Error Rate* – BER) em função do nível de ruído (SNR), para diferentes estratégias de definição de *threshold*.

À medida que o ruído se intensifica (0 dB para mais ruidosos), todas as estratégias convergem para desempenhos mais elevados de erro, com diferenças menos pronunciadas entre si. Ainda assim, é possível observar que a estratégia baseada no EER calculado sobre todos os dados de validação tem uma certa robustez relativa em níveis intermediários de ruído, enquanto a estratégia da média das médias passa a apresentar maior variabilidade, refletida em desvios padrão mais elevados em alguns cenários.

Ao final, a estratégia baseada na média das médias dos *logits* apresenta vantagem nos cenários de maior interesse prático, atingindo os menores valores de BER nas condições de baixo ruído, incluindo o cenário sem ruído. Esse comportamento se mantém consistente também em níveis moderados de ruído, nos quais permanece competitiva em relação ao limiar fixo e superior às estratégias baseadas em EER. Em níveis mais elevados de ruído, observa-se uma convergência entre as abordagens, com aumento generalizado do erro, reduzindo a relevância das diferenças entre os métodos. Dessa forma, ao considerar o desempenho global ao longo das diferentes condições avaliadas, a estratégia da média das médias se mostra a mais eficaz, apresentando o melhor compromisso entre desempenho e estabilidade nos cenários em que os modelos operam com maior confiabilidade.

Estabelecida a estratégia de *threshold*, passamos a analisar a performance individual dos modelos para *vocoders* específicos. A Figura 5.14 apresenta, para os treinamentos de 1 a 4, três gráficos de BER por nível de ruído, por treinamento, um para cada *vocoder* externo. Para todos os 21 treinamentos, veja Anexo III.

A partir da análise das figuras, observa-se que, de modo geral, os modelos são capazes de detectar o uso de *vocoders*. No entanto, para determinados *vocoders* específicos,

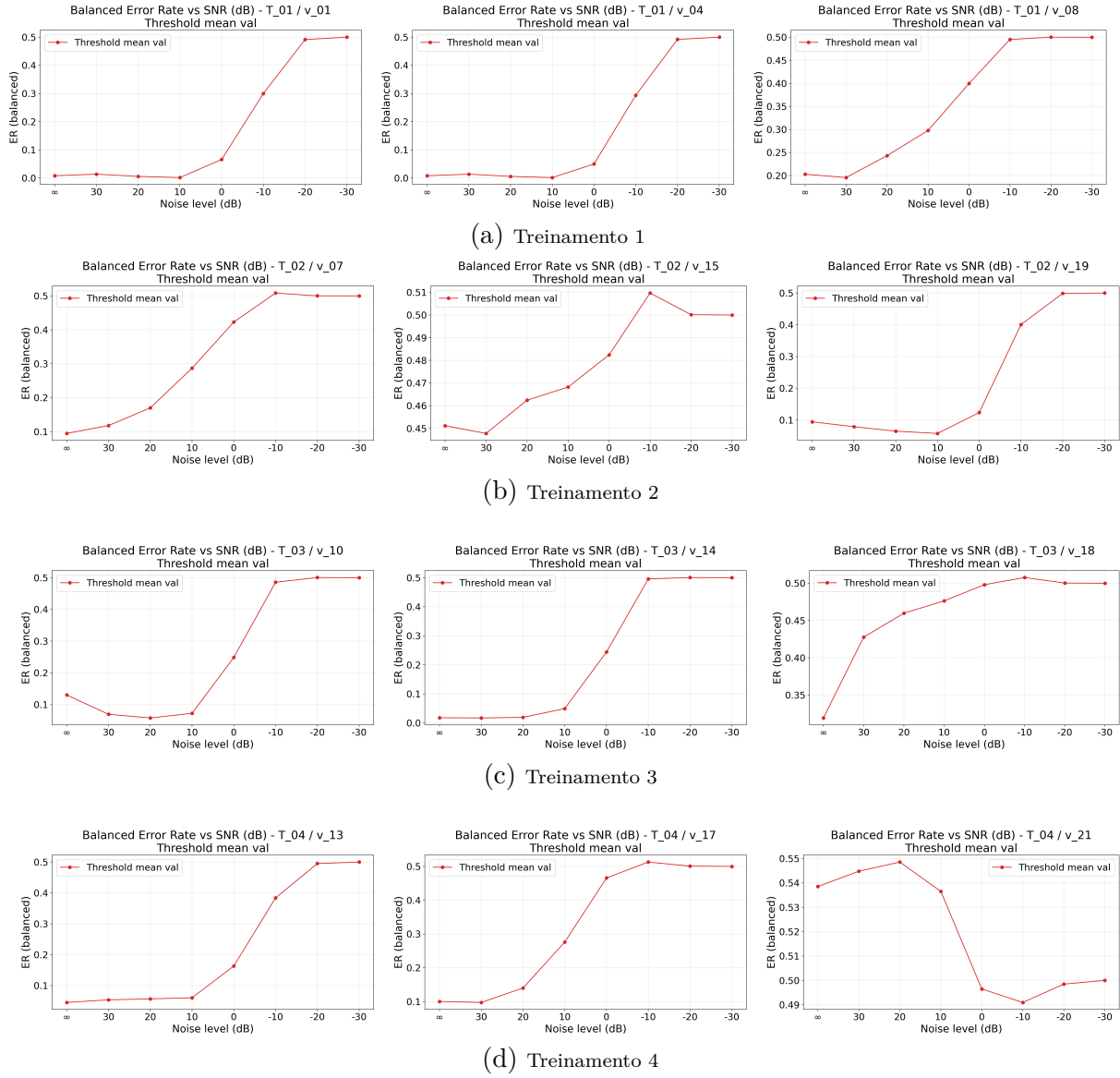


Figura 5.14: Performance dos modelos para *vocoder* específico

verificam-se falhas de detecção por parte das redes. A Tabela 5.5 sintetiza, para amostras sem ruído, a capacidade dos modelos em identificar o uso de *vocoders*, utilizando como métrica o BER obtido pela estratégia da média das médias. Esta tabela possui 3 valores de BER por linha, uma vez que cada *vocoder* foi considerado *vocoder* externo em 3 treinamentos diferentes. Para melhor visualização, os valores menores que 0.1 foram destacados de verde e os valores maiores que 0.4 foram destacados de vermelho.

Tabela 5.5: Desempenho (BER) por *vocoder* e diferentes configurações de treinamento.

<i>Vocoder</i>	Trein.	BER	Trein.	BER	Trein.	BER
v_01	1	0.0079	8	0.1002	15	0.0166
v_02	7	0.0034	14	0.1506	21	0.0076
v_03	6	0.0844	13	0.0129	20	0.0286
v_04	1	0.0079	9	0.0032	16	0.0878
v_05	5	0.0114	12	0.0328	19	0.0033
v_06	6	0.0844	14	0.1506	15	0.0166
v_07	2	0.0948	9	0.0806	17	0.0513
v_08	1	0.2028	10	0.2060	17	0.1500
v_09	5	0.0633	13	0.0631	21	0.1237
v_10	3	0.1302	10	0.1388	18	0.0758
v_11	6	0.0844	8	0.1008	16	0.0880
v_12	5	0.0122	14	0.1512	16	0.0899
v_13	4	0.0454	11	0.0061	18	0.0569
v_14	3	0.0179	11	0.0100	19	0.0163
v_15	2	0.4512	10	0.4788	19	0.4986
v_16	7	0.4584	8	0.2528	17	0.4780
v_17	4	0.1004	12	0.2406	20	0.2173
v_18	3	0.3193	12	0.3408	21	0.2514
v_19	2	0.0945	11	0.0087	20	0.1219
v_20	7	0.5004	9	0.5019	18	0.4753
v_21	4	0.5385	13	0.5117	15	0.5129

A análise da Tabela 5.5 evidencia que, embora haja uma tendência geral de bom desempenho dos modelos — refletida pela grande quantidade de valores destacados em verde — esse comportamento não é uniforme entre todos os *vocoders*. Observa-se que diversos *vocoders*, como v_01 a v_07, v_11 a v_14 e v_19, apresentam BER consistentemente baixos em pelo menos uma, e frequentemente em múltiplas configurações de treinamento, indicando boa capacidade de generalização dos modelos para esses casos.

Por outro lado, há *vocoders* que se destacam negativamente, como v_15, v_20 e v_21, cujos valores de BER permanecem elevados em todas as configurações (destacados em vermelho), indicando que os modelos não generalizaram bem para esses *vocoders* quando considerados como externos ao treinamento.

Além disso, alguns *vocoders*, como v_16, v_17 e v_18, apresentam comportamento intermediário, com forte dependência da configuração de treinamento, o que indica sensibilidade à escolha dos dados utilizados no treinamento.

Observa-se ainda que os *vocoders* associados aos maiores valores de BER correspondem, em geral, a abordagens mais recentes, que possivelmente se beneficiaram de maior disponibilidade de dados, tempo de desenvolvimento e recursos computacionais durante seu treinamento. Esse fator pode ter contribuído para a geração de sinais mais difíceis de distinguir, impactando negativamente a capacidade de generalização dos modelos quando expostos a esses cenários.

Isso não implica que tais *vocoders* sejam indistinguíveis. Pelo contrário, como evidenciado pelo desempenho dos modelos nos conjuntos de validação — quando esses *vocoders* estavam presentes no treinamento —, observa-se que sua detecção é viável, conforme ilustrado pela matriz de confusão da Figura 5.6.

Capítulo 6

Conclusão

Avanços na tecnologia de *vocoders* neurais tornaram cada vez mais desafiador distinguir entre áudio sintético e gravações autênticas. Neste estudo, implementamos um método baseado em redes neurais convolucionais (CNNs) para detectar o uso de *vocoders* neurais na produção de áudio por meio da análise de imagens de espectrogramas Mel.

Este trabalho abordou o problema da detecção de áudios sintéticos gerados por *vocoders* neurais, em um cenário de crescente preocupação com o uso malicioso de *deepfakes*, especialmente no campo forense, onde a autenticidade de evidências digitais é essencial. A abordagem proposta, baseada na utilização de espectrogramas Mel combinados com redes neurais convolucionais, mostrou-se eficaz na identificação de padrões característicos de áudios artificiais.

Uma das contribuições desta pesquisa foi a construção de um dataset controlado, composto por áudios reais e sintéticos gerados por diferentes *vocoders*, com inserção sistemática de ruído em múltiplos níveis de relação sinal-ruído (SNR). Essa estratégia permitiu não apenas o treinamento dos modelos, mas também uma avaliação estruturada de sua robustez em condições progressivamente mais desafiadoras, aproximando o estudo de cenários reais.

Os resultados indicam que os modelos baseados em CNN apresentam bom desempenho na distinção entre áudios *bonafide* e sintéticos, sendo observada superioridade na arquitetura VGG19 em relação às demais aqui estudadas. A análise da influência do ruído evidenciou uma degradação gradual do desempenho com a redução do SNR, porém mantendo níveis relevantes de capacidade de classificação mesmo em condições adversas.

Adicionalmente, a utilização da técnica de explicabilidade Grad-CAM permitiu compreender melhor o comportamento interno dos modelos. Observou-se que as redes concentram sua atenção em regiões específicas dos espectrogramas, particularmente em áreas de baixa energia espectral, como momentos de silêncio e regiões de altas frequências. Esse comportamento sugere que os artefatos explorados pelos modelos tendem a se manifestar

justamente em componentes menos dominantes do sinal, que não são mascarados por regiões de alta energia. Por esse motivo, não é adequado restringir a análise a segmentos detectados por algoritmos de detecção de silêncio (*Voice Activity Detection* — VAD), uma vez que tais abordagens poderiam descartar partes relevantes para a identificação de sinais sintéticos.

A análise baseada em teste de ablação dos 512 *feature maps* da última camada convolucional da arquitetura VGG19 permitiu investigar aspectos funcionais das representações internas aprendidas pela rede. Nesse procedimento, cada *feature map* foi removido individualmente, e o impacto dessa remoção sobre a saída da classe correspondente foi quantificado, permitindo a construção de vetores de importância funcional para cada amostra. A projeção desses vetores por meio da técnica UMAP revelou agrupamentos associados aos diferentes *vocoders*, indicando que a rede organiza internamente padrões distintos de relevância entre os canais convolucionais para diferentes métodos de síntese de voz. Observou-se ainda que alguns *vocoders*, como *v_15*, *v_20* e *v_21*, apresentaram maior proximidade em relação aos áudios *bonafide*, sugerindo maior similaridade entre as representações internas utilizadas pela CNN para esses grupos. Esse comportamento oferece uma interpretação para a maior dificuldade de classificação observada nesses *vocoders*, uma vez que representações funcionais mais próximas das amostras autênticas tendem a reduzir a separabilidade entre as classes no espaço interno da rede.

Outro ponto relevante foi a análise das estratégias de definição de *threshold* aplicado à saída do modelo para a decisão de legitimidade do áudio. A definição desse limiar deve ser realizada exclusivamente com base em dados pertencentes ao conjunto de validação, isto é, dentro do mesmo domínio utilizado no processo de desenvolvimento do modelo. O uso de dados provenientes de cenários externos — como aqueles empregados nos testes de generalização, envolvendo *vocoders*, locutores ou condições acústicas não observadas durante o treinamento — implica a incorporação indevida de informações fora do escopo de validação. Tal prática compromete a avaliação do modelo, uma vez que o *threshold* passa a ser ajustado com conhecimento implícito do comportamento de dados externos, podendo inflar artificialmente métricas de desempenho e resultar em estimativas excessivamente otimistas. Além disso, em aplicações reais, o modelo não terá acesso prévio às distribuições de dados futuros, o que torna essa estratégia inviável do ponto de vista prático. Portanto, o *threshold* deve ser tratado como um hiperparâmetro definido antes da exposição do modelo a dados externos, garantindo uma avaliação mais fiel de sua capacidade de generalização.

Nesse sentido, os resultados indicam que abordagens baseadas nos *logits* da última camada oferecem uma separação mais informativa entre áudios legítimos e sintéticos, em comparação ao uso direto de probabilidades (i.e., valores provenientes da função de ativação da última camada), contribuindo para decisões mais estáveis. Essa diferença

torna-se evidente ao analisar as distribuições das saídas do modelo, nas quais os *logits* apresentam maior dispersão, ao contrário das probabilidades, que tendem à saturação nas extremidades do intervalo $[0,1]$. Optou-se, então, pela abordagem baseada na média das médias, calculada a partir das saídas do modelo para as classes *bonafide* e sintética no conjunto de validação. Essa escolha mostrou-se adequada por reduzir a influência de variações locais e de valores extremos nas distribuições, proporcionando uma estimativa mais robusta do ponto de separação entre as classes. Do ponto de vista quantitativo, tal estratégia impactou diretamente nas análises de *Balanced Error Rate* (BER) em função do SNR, uma vez que o uso de *thresholds* baseados nos *logits*, aliado à média das médias, resultou em curvas mais consistentes ao longo dos diferentes níveis de ruído.

Os testes de generalização, envolvendo *vocoders*, locutores e microfone não observados durante o treinamento evidenciaram desafios na transferência dos modelos para domínios externos. Embora o desempenho geral permaneça satisfatório, as variações observadas indicam sensibilidade a mudanças de domínio, o que é esperado em problemas dessa natureza. De forma mais específica, verificou-se que o desempenho tende a variar entre diferentes *vocoders*. A variação do desempenho torna-se mais evidente quando analisada em conjunto com a redução do SNR, na qual se observa um impacto gradual nas curvas de *Balanced Error Rate* (BER), sem, contudo, comprometer completamente a capacidade discriminativa do modelo.

A comparação direta entre os resultados obtidos neste trabalho e *benchmarks* amplamente utilizados na literatura, como ASVspoof e ADD, deve ser interpretada com cautela devido a diferenças metodológicas substanciais. No caso do ASVspoof 5 Trilha 1, embora represente uma referência importante para detecção de *spoofing/deepfakes* em fala, seu protocolo experimental difere significativamente do adotado nesta pesquisa. O dataset foi construído em 2023 e incorpora diferentes codecs, tornando o cenário experimental distinto. O melhor desempenho reportado na condição closed foi de 8,61% EER. Em contraste, o presente trabalho obteve BER global de 16,2% para dados sem ruído até 19,2% para níveis intermediários de ruído. Estes resultados são numericamente piores em uma análise superficial, mas a comparação direta é inadequada. Quando se restringe a análise aos *vocoders* anteriores a 2023, observa-se desempenho melhor no presente estudo, inclusive com BER inferior a 1% em determinados *vocoders*. Soma-se a isso o fato de o ASVspoof 5 reportar EER calculada sobre o conjunto de avaliação (o que seria equivalente aos dados de generalização neste trabalho), prática que, conforme discutido anteriormente, não é metodologicamente apropriada para definição ou ajuste de limiares de decisão.

Situação semelhante ocorre na comparação com o ADD 2023, cuja equivalência experimental também é limitada. Além de empregar F1-score como métrica principal, a trilha mais próxima deste trabalho (Track 3) é orientada à identificação do algoritmo gerador

de *deepfake*, e não à classificação binária entre amostras genuínas e falsificadas, impossibilitando comparação direta com o problema investigado aqui. Uma referência mais adequada encontra-se no ADD 2022 Low-Quality Fake Audio Detection, tarefa de classificação verdadeiro/falso em áudios degradados por adulterações, ruídos e sons ambientes, aproximando-se mais do objetivo desta pesquisa. Nessa trilha, o melhor desempenho reportado foi de 21,7% EER, resultado numericamente pior do que o obtido neste trabalho quando considerado áudios expostos a, no máximo, níveis intermediários de ruído, especialmente ao se considerar o subconjunto de *vocoders* anteriores a 2022, para os quais os erros observados foram substancialmente menores. Ainda assim, permanece a limitação fundamental de comparabilidade, uma vez que os datasets diferem em composição, distribuição estatística, tecnologias de síntese empregadas e métricas de avaliação.

De maneira geral, os resultados indicam que os modelos desenvolvidos apresentam boa capacidade de adaptação, ainda que com espaço para aprimoramento em cenários mais desafiadores. Esse comportamento reforça a relevância da abordagem proposta e aponta para a importância de estratégias adicionais que promovam maior robustez e generalização, especialmente em aplicações práticas e forenses.

Este trabalho demonstra que a combinação entre espectrogramas Mel e redes neurais convolucionais constitui uma estratégia viável para a detecção de *deepfakes* de áudio. No contexto forense, os resultados reforçam o potencial dessas abordagens como ferramentas auxiliares na verificação da autenticidade de áudios, embora ainda não substituam outras análises especializadas.

Como perspectivas futuras, destacam-se a ampliação do dataset com inclusão de áudios processados por *codecs*, maior diversidade de *vocoders* e condições acústicas, incluindo a presença de sons ambientais, e o desenvolvimento de modelos com maior capacidade de generalização para diferentes *vocoders*, locutores e condições de gravação não observados durante o treinamento. Propõe-se, ainda, investigar outras formas de representação visual do sinal de áudio, como escalogramas obtidos por meio de transformadas *wavelet*, bem como comparar as redes convolucionais empregadas neste trabalho com arquiteturas baseadas em *Vision Transformers* (ViTs). Destaca-se, também, a necessidade de aprofundamento em métodos de explicabilidade, visando ampliar a transparência e a confiabilidade dos modelos desenvolvidos.

Por fim, os resultados obtidos reforçam a necessidade do desenvolvimento contínuo de métodos robustos para detecção de áudios sintéticos, especialmente diante da constante evolução dos sistemas de síntese de voz. O avanço de arquiteturas neurais capazes de produzir sinais cada vez mais realistas tende a reduzir progressivamente as diferenças perceptíveis entre áudios genuínos e artificiais, tornando essencial a investigação de características sutis associadas aos processos de geração. Nesse contexto, abordagens ba-

seadas em aprendizado profundo e análise espectral apresentam potencial relevante para aplicações forenses, autenticação de conteúdo digital e combate à desinformação.

Referências

- [1] Frank, Joel e Lea Schönherr: *Wavefake: A data set to facilitate audio deepfake detection*. 2021. <https://arxiv.org/abs/2111.02813>. x, 9, 10, 32, 35, 66
- [2] Dudley, Homer: *The carrier nature of speech*. Bell System Technical Journal, 19(4):495–515, 1940. x, 2, 10, 11, 12
- [3] Oord, Aaron van den: *Vq-vae: Vector quantized variational autoencoder*. <https://avdnoord.github.io/homepage/vqvae/>, 2023. Acesso em: 2 de junho de 2025. x, 16
- [4] Prenger, R., M. Sim e M. Tschannen: *Waveglow: A flow-based generative network for speech synthesis*. Em *Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, páginas 1–5. IEEE, 2018. x, 15, 16, 17
- [5] Kumar, Kundan, Rithesh Kumar, Thibault de Boissiere, Lucas Gestein, Wei Zhen Teoh, Jose M. R. Sotelo, Alexandre de Brébisson, Yoshua Bengio e Aaron C. Courville: *Melgan: Generative adversarial networks for conditional waveform synthesis*. Em *Neural Information Processing Systems*, 2019. <https://api.semanticscholar.org/CorpusID:202777813>. x, 10, 14, 17, 18
- [6] Yamamoto, Ryuichi, Eunwoo Song e Jae Min Kim: *Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram*. Em *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, páginas 6199–6203, 2020. x, 19, 20
- [7] Kong, Zhifeng, Wei Ping, Jiaji Huang, Kexin Zhao e Bryan Catanzaro: *Diffwave: A versatile diffusion model for audio synthesis*. ArXiv, abs/2009.09761, 2020. <https://api.semanticscholar.org/CorpusID:221818900>. x, 19, 21
- [8] Mustafa, Ahmed, Nicola Pia e Guillaume Fuchs: *Stylemelgan: An efficient high-fidelity adversarial vocoder with temporal adaptive normalization*. Em *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, páginas 6034–6038, 2021. x, 21, 22
- [9] Yang, Geng, Shan Yang, Kai Liu, Peng Fang, Wei Chen e Lei Xie: *Multi-band melgan: Faster waveform generation for high-quality text-to-speech*. Em *2021 IEEE Spoken Language Technology Workshop (SLT)*, páginas 492–498, 2021. x, 22, 23, 24

- [10] Lee, Sang gil, Wei Ping, Boris Ginsburg, Bryan Catanzaro e Sungroh Yoon: *Bigvgan: A universal neural vocoder with large-scale training*. Em *International Conference on Learning Representations (ICLR)*, 2023. x, 23, 25
- [11] Siuzdak, Hubert: *Vocos: Closing the gap between time-domain and fourier-based neural vocoders for high-quality audio synthesis*. arXiv preprint arXiv:2306.00814, 2023. xi, 25, 26
- [12] Martín-Doñas, Juan M. e Aitor Álvarez: *The vicomtech audio deepfake detection system based on wav2vec2 for the 2022 add challenge*. Em *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, páginas 9241–9245, 2022. xi, 32, 41
- [13] MathWorks: *What is a convolutional neural network?* <https://www.mathworks.com/discovery/convolutional-neural-network.html>, 2025. Accessed: 2025-11-04. xi, 51
- [14] Damiani, Jesse: *A voice deepfake was used to scam a ceo out of \$243,000*. Forbes, September 2019. Accessed from Forbes, published under the Technology section. 1
- [15] Brasil, CNN: *Funcionário de multinacional paga us\$ 25 mi a golpista que usou 'deepfake' para simular reunião*. CNN Brasil, 2024. <https://www.cnnbrasil.com.br/internacional/funcionario-de-multinacional-paga-us-25-mi-a-golpista-que-usou-deepfake-para-simular-reuniao/>, Acessado em 1º de abril de 2025. 1
- [16] BBC News: *Fake Biden robocall tells voters to skip New Hampshire primary election*. January 2024. <https://www.bbc.com/news/world-us-canada-68064247>, Acesso em 9 de agosto de 2025. 1
- [17] Times, Financial: *How deepfakes could impact politics in 2024: A look at the election*. Financial Times, 2024. <https://www.ft.com/content/62d81e6c-eec0-4d09-a71f-6aba579912dd>, Accessed: 2024-04-01. 1
- [18] Reuters: *Russia using generative ai to ramp up disinformation, says ukraine minister*. Reuters, 2024. <https://www.reuters.com/technology/artificial-intelligence/russia-using-generative-ai-ramp-up-disinformation-says-ukraine-minister-2024-10-16/>, Accessed: 2025-04-01. 2
- [19] Zhou, Wangjin, Zhengdong Yang, Chenhui Chu, Sheng Li, Raj Dabre, Yi Zhao e Kawahara Tatsuya: *Mos-fad: Improving fake audio detection via automatic mean opinion score prediction*. Em *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, páginas 876–880, 2024. 2, 34
- [20] Unidas, Nações: *Ameaças da inteligência artificial em guerras de informação são discutidas na onu*, 2023. <https://news.un.org/pt/audio/2023/12/1825167>, Acesso em: 01 abr. 2025. 2

- [21] Europol: *Malicious uses and abuses of artificial intelligence*, 2023. https://www.europol.europa.eu/cms/sites/default/files/documents/malicious_uses_and_abuses_of_artificial_intelligence_europol.pdf, Acessado em 1 de abril de 2025. 2
- [22] Flanagan, James L.: *Speech Analysis, Synthesis and Perception*. Springer-Verlag, Berlin, Heidelberg, 2nd edição, 1972, ISBN 978-3-642-88007-5. <https://doi.org/10.1007/978-3-642-88005-1>. 2, 7, 11
- [23] Laroche, Jean e Mark Dolson: *New phase-vocoder techniques for pitch-shifting, harmonizing and other exotic effects*. Proceedings of the 1999 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics. WASPAA'99 (Cat. No.99TH8452), páginas 91–94, 1999. <https://api.semanticscholar.org/CorpusID:1984485>. 2
- [24] Li, Menglu, Yasaman Ahmadiadli e Xiao Ping Zhang: *A survey on speech deepfake detection*. ACM Computing Surveys, 57(7):1–38, July 2025. Online publication date: 24 January 2025. 2
- [25] Vinotha, R, LD Vijay Anand, Deepak John Reji *et al.*: *Advancing accessibility: Voice cloning and speech synthesis for individuals with speech disorders*. 2024. 2
- [26] Borrelli, Clara, Paolo Bestagini, Fabio Antonacci, Augusto Sarti e Stefano Tubaro: *Synthetic speech detection through short-term and long-term prediction traces*. Eurasip Journal on Information Security, 2021, dezembro 2021, ISSN 2510523X. 2, 29
- [27] Martínez-Serrano, Alba, Claudia Montero-Ramírez e Carmen Peláez-Moreno: *Authenticity at risk: Key factors in the generation and detection of audio deepfakes*. Applied Sciences, 15(2), 2025, ISSN 2076-3417. <https://www.mdpi.com/2076-3417/15/2/558>. 2
- [28] Thornton, Andrew, Chengkai Song, Hao Wu e Zhenyu Zhou: *Audio recognition using mel spectrograms and convolution neural networks*. Relatório Técnico, University of California, San Diego, NoiseLab, 2019. https://noiselab.ucsd.edu/ECE228_2019/Reports/Report38.pdf, Course project report, ECE228. 2
- [29] Moore, Brian C. J.: *An Introduction to the Psychology of Hearing*. Brill, Leiden, The Netherlands, 6th edição, 2012, ISBN 978-9004232037. Fundamental reference on psychoacoustics, covering frequency, amplitude, duration, and sound perception. 7, 8, 9
- [30] Carlson, A. Bruce, Paul B. Crilly e Janet C. Rutledge: *Communication Systems: An Introduction to Signals and Noise in Electrical Communication*. McGraw-Hill Higher Education, New York, 4ª edição, 2002, ISBN 0-07-011127-8. McGraw-Hill series in electrical engineering. 7, 48
- [31] Fastl, Hugo e Eberhard Zwicker: *Psychoacoustics: Facts and Models*. Springer Science & Business Media, 2007, ISBN 9783540683405. <https://books.google.com.br/books?id=eGcfn9ddRhcC>. 8

- [32] Volkmann, J., S. Stevens e E. Newman: *A scale for the measurement of the psychological magnitude pitch*. The Journal of the Acoustical Society of America, 8:208–208, junho 2005. 9
- [33] Radford, Alec, Jong Wook Kim, Tao Xu, Greg Brockman, Christine Mcleavey e Ilya Sutskever: *Robust speech recognition via large-scale weak supervision*. Em Krause, Andreas, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato e Jonathan Scarlett (editores): *Proceedings of the 40th International Conference on Machine Learning*, volume 202 de *Proceedings of Machine Learning Research*, páginas 28492–28518. PMLR, 23–29 Jul 2023. <https://proceedings.mlr.press/v202/radford23a.html>. 9, 35
- [34] Soleymani, Sobhan, Ali Dabouei, Seyed Mehdi Iranmanesh, Hadi Kazemi, Jeremy Dawson e Nasser M. Nasrabadi: *Prosodic-enhanced siamese convolutional neural networks for cross-device text-independent speaker verification*. Em *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, páginas 1–7, 2018. 9
- [35] Choi, Keunwoo, György Fazekas, Mark Sandler e Kyunghyun Cho: *Convolutional recurrent neural networks for music classification*. Em *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, páginas 2392–2396, 2017. 9
- [36] Bhattacharjee, Mrinmoy, S. R. M. Prasanna e Prithwijit Guha: *Time-frequency audio features for speech-music classification*, 2018. <https://arxiv.org/abs/1811.01222>. 9
- [37] Hanai, Tuka, Mohammad Ghassemi e James Glass: *Detecting depression with audio/text sequence modeling of interviews*. páginas 1716–1720, setembro 2018. 9
- [38] Davis, Steven B. e Paul Mermelstein: *Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences*. IEEE Transactions on Acoustics, Speech, and Signal Processing, 28(4):357–366, 1980. 9
- [39] Stevens, Kenneth N.: *Acoustic Phonetics*. MIT Press, Cambridge, MA, 1998, ISBN 978-0-262-19404-0. 10
- [40] Schroeder, Manfred R.: *Vocoders: Analysis and synthesis of speech (a review of 30 years of applied speech research)*. Proceedings of the IEEE, 54:720–734, maio 1966. Reprinted in [243, pp. 352–366]. 10
- [41] Davies, Lawrence E.: *Speaking machine heard in tryout*. The New York Times, janeiro 1939. <https://www.nytimes.com>, Page 1, coverage of the Voder voice synthesizer demonstration at the 1939 New York World’s Fair. 11
- [42] Dudley, Homer: *The vocoder—electrical re-creation of speech*. Bell Laboratories Record, 18(4):122–126, December 1939. Descreve o princípio de funcionamento do vocoder e explica que o nome deriva da expressão “voice codes”, originando o termo “Vocoder”. 11

- [43] Wohlstetter, Roberta: *Pearl Harbor: Warning and Decision*. Stanford University Press, Stanford, CA, 1962, ISBN 9780804705981. 11
- [44] Kahn, David: *The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet*. Scribner, New York, 1996, ISBN 978-0684831305. 11
- [45] Campbell Jr, Joseph P e Richard A Dean: *A history of secure voice coding: Insights drawn from the career of one of the earliest practitioners of the art of speech coding*. Digital Signal Processing, An expanded pdf version is available at http://www.nsa.gov/about/files/cryptologic_heritage/publications/wwii/sigsaly_history.pdf, 1993. 11
- [46] Vail, Mark: *Vintage Synthesizers: Pioneering Designers, Groundbreaking Instruments, Collecting Tips, Mutants of Technology*. Backbeat Books, 2ª edição, 2000, ISBN 978-0879302757. 12
- [47] iZotope: *A brief history of the vocoder*, 2018. <https://www.izotope.com/en/learn/a-brief-history-of-the-vocoder.html>, Acessado em 29 de maio de 2025. 12
- [48] Loopop: *Moog spectravox – detailed overview, and how vocoders work*, 2019. <https://loopopmusic.com/moog-spectravox-detailed-overview-and-how-vocoders-work>, Acessado em 29 de maio de 2025. 12
- [49] Sterne, Jonathan: *MP3: The Meaning of a Format*. Duke University Press, Durham, 2012, ISBN 978-0822352873. 12
- [50] Kelly, John L. e Carol C. Lochbaum: *Speech synthesis*. Em *Proceedings of the Fourth International Congress on Acoustics*, Copenhagen, Denmark, 1962. 12
- [51] Flanagan, J. L. e R. M. Golden: *Phase vocoder*. The Bell System Technical Journal, 45(9):1493–1509, 1966. 13
- [52] Rabiner, Lawrence R. e Ronald W. Schafer: *Digital Processing of Speech Signals*. Prentice-Hall, Englewood Cliffs, NJ, 1978. Bell Telephone Laboratories. 13
- [53] Schroeder, Manfred R. e Bishnu S. Atal: *Code-excited linear prediction (celp): High-quality speech at very low bit rates*. IEEE Transactions on Acoustics, Speech, and Signal Processing, 33(4):937–945, 1985. 13
- [54] Bradbury, Jeremy: *Linear predictive coding*. Mc G. Hill, 2000. 13
- [55] Kar, Aditi e Kishore Sahoo: *A brief review on lpc and velpc techniques for speech compression*. 2018. <https://api.semanticscholar.org/CorpusID:53369999>. 14
- [56] Ghule, Mr. Arjun e Dr. Prabhu G. Benakop: *Lpc models and different speech enhancement techniques-a review*. 2015. <https://api.semanticscholar.org/CorpusID:250615738>. 14

- [57] Stern, Rebecca Slayton: *Speak & spell (1978)*. Em *Voice Synthesizer: A History*, páginas 108–131. MIT Press, 2023. <https://doi.org/10.7551/mitpress/14208.001.0001>. 14
- [58] Morise, Masanori, Fumiya Yokomori e Kenji Ozawa: *WORLD: A vocoder-based high-quality speech synthesis system for real-time applications*. IEICE Transactions on Information and Systems, E99-D(7):1877–1884, jul 2016. 14
- [59] Kawahara, Hideki: *Straight, exploitation of the other aspect of vocoder: Perceptually isomorphic decomposition of speech sounds*. Acoustical Science and Technology, 27(6):349–353, 2006. 14
- [60] O’Shaughnessy, Douglas: *Speech coding: History and current trends*. IEEE Communications Magazine, 61(1):12–18, 2023. 14
- [61] Oord, Aäron van den, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior e Koray Kavukcuoglu: *Wavenet: A generative model for raw audio*. Em *Speech Synthesis Workshop*, 2016. <https://api.semanticscholar.org/CorpusID:6254678>. 14, 15
- [62] Kalchbrenner, Nal, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aaron van den Oord, Sander Dieleman e Koray Kavukcuoglu: *Efficient neural audio synthesis*. Em Dy, Jennifer e Andreas Krause (editores): *Proceedings of the 35th International Conference on Machine Learning*, volume 80 de *Proceedings of Machine Learning Research*, páginas 2410–2419. PMLR, 10–15 Jul 2018. <https://proceedings.mlr.press/v80/kalchbrenner18a.html>. 14, 15, 17
- [63] Popov, Vadim, Alexey Tikhonov, Mikhail Kudinov, Ilya Mazur e Alexey Zubkov: *Fast and lightweight on-device TTS with Tacotron2 and LPCNet*. Em *Proceedings of Interspeech 2020*, páginas 3575–3579, Shanghai, China, 2020. ISCA. https://www.isca-archive.org/interspeech_2020/popov20_interspeech.pdf, Demonstrates a real-time, low-resource text-to-speech system optimized for mobile and embedded devices. 14
- [64] Tan, Xu, Tao Qin, Frank Soong e Tie Yan Liu: *A survey on neural speech synthesis*. arXiv preprint arXiv:2106.15561, 2021. <https://arxiv.org/abs/2106.15561>, Preprint. Under review. 14
- [65] Maiti, Soumi e Michael I. Mandel: *Parametric resynthesis with neural vocoders*. Em *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, páginas 229–233, New Paltz, NY, USA, 2019. IEEE. 16
- [66] Achanta, Sivanand, Albert Antony, Ladan Golipour, Jiangchuan Li, Tuomo Raitio, Ramya Rasipuram, Francesco Rossi, Jennifer Shi, Jaimin Upadhyay, David Winarsky e Hepeng Zhang: *On-device neural speech synthesis*. Em *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, páginas 6699–6703. IEEE, 2020. 17

- [67] Okamoto, Takuma, Tomoki Toda, Yoshinori Shiga e Hisashi Kawai: *Noise level limited sub-modeling for diffusion probabilistic vocoders*. Em *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, páginas 6079–6083, Toronto, Canada, 2021. IEEE. 20
- [68] Huang, Rongjie, Max W. Y. Lam, J. Wang, Dan Su, Dong Yu, Yi Ren e Zhou Zhao: *Fstdiff: A fast conditional diffusion model for high-quality speech synthesis*. Em *International Joint Conference on Artificial Intelligence*, 2022. <https://api.semanticscholar.org/CorpusID:248300058>. 20
- [69] Kong, Jungil, Jaehyeon Kim e Jaekyoung Bae: *Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis*. Em *NeurIPS*, 2020. 20, 21
- [70] Qiu, Zeyu, Jun Tang, Yaxin Zhang, Jiaxin Li e Xishan Bai: *A voice cloning method based on the improved hifi-gan model*. *Computational Intelligence and Neuroscience*, 2022:1–12, outubro 2022. <https://doi.org/10.1155/2022/6707304>. 21
- [71] Hwang, Seung Yeon e Jeong Joon Kim: *Efficient deep learning-based speech synthesis model study for tts*. *Journal of Harbin Engineering University*, 45(7):451–455, julho 2024, ISSN 1006-7043. 23
- [72] Patel, Tanvina B. e Hemant A. Patil: *Combining evidences from mel cepstral, cochlear filter cepstral and instantaneous frequency features for detection of natural vs. spoofed speech*. Em *Proceedings of the ASVspoof 2015 Challenge*, 2015. Available at: <http://www.asvspoof.org/>. 27, 38
- [73] Lieto, A., D. Moro, F. Devoti, C. Parera, Vincenzo Lipari, Paolo Bestagini e Stefano Tubaro: *HELLO? WHO AM I TALKING TO? A SHALLOW CNN APPROACH FOR HUMAN VS. BOT SPEECH CLASSIFICATION*. Em *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, páginas 2577–2581, Calgary, AB, Canada, 2018. IEEE, ISBN 978-1-5386-4658-8. 27
- [74] Panayotov, Vassil, Guoguo Chen, Daniel Povey e Sanjeev Khudanpur: *Librispeech: An asr corpus based on public domain audio books*. Em *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, páginas 5206–5210. IEEE, 2015. 28
- [75] Black, Alan W. e Kevin A. Lenzo: *Festvox: Building synthetic voices*. <http://www.festvox.org/>, 2024. Accessed: 2024-11-17. 28, 46
- [76] Lavrentyeva, Galina, Sergey Novoselov, Andzhukaev Tseren, Marina Volkova, Artem Gorlanov e Alexander Kozlov: *Stc antispoofing systems for the asvspoof2019 challenge*. Em *Interspeech*, 2019. <https://api.semanticscholar.org/CorpusID:118674374>. 28
- [77] Todisco, Massimiliano, Xin Wang, Ville Vestman, Md Sahidullah, Héctor Delgado, Nicholas Evans, Junichi Yamagishi e Tomi Kinnunen: *ASVspoof 2019: Future horizons in spoofed and fake audio detection*. Em *Proc. Interspeech*, páginas 1008–1012, 2019. 28, 29, 39

- [78] Wu, Xiang, Ran He, Zhenan Sun e Tieniu Tan: *A light cnn for deep face representation with noisy labels*. IEEE Transactions on Information Forensics and Security, 13(11):2884–2896, 2018. 28, 31
- [79] Kinnunen, Tomi, Md Sahidullah, Héctor Delgado, Massimiliano Todisco, Nicholas Evans, Junichi Yamagishi e Kong Aik Lee: *The asvspoof 2017 challenge: Assessing the limits of replay spoofing attack detection*. Em *Proceedings of Interspeech 2017*, páginas 2–6, 2017. https://www.isca-archive.org/interspeech_2017/kinnunen17_interspeech.html. 28, 39
- [80] Brown, Judith C.: *Calculation of a constant q spectral transform*. The Journal of the Acoustical Society of America, 89(1):425–434, 1991. 28
- [81] Jung, Jee weon, Hee Soo Heo e Ha Jin Yu: *Improved rawnet with feature map scaling for text-independent speaker verification*. Em *Proc. Interspeech*, páginas 4343–4347, 2020. 28
- [82] Tak, Hemlata, Jose Patino, Massimiliano Todisco, Andreas Nautsch, Nicholas Evans e Anthony Larcher: *End-to-end anti-spoofing with rawnet2*. Em *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, páginas 6369–6373, 2021. 28, 34
- [83] Jung, Jee weon, Hee Soo Kim, Hye jin Kim, Heuseok Lee e Ha Jin Kim: *Improved rawnet with feature map scaling for text-independent speaker verification*. Em *Proc. Interspeech*, páginas 4343–4347. ISCA, 2020. 28
- [84] Todisco, Massimiliano, Héctor Delgado e Nicholas W. D. Evans: *Constant q cepstral coefficients: A spoofing countermeasure for automatic speaker verification*. Em *Proc. Odyssey 2016*, páginas 283–290, 2016. 29
- [85] Jung, Jee weon, Hee Soo Heo, Hemlata Tak, Hye jin Shim, Joon Son Chung, Bong Jin Lee, Ha Jin Yu e Nicholas Evans: *Aasist: Audio anti-spoofing using integrated spectro-temporal graph attention networks*. Em *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, páginas 6367–6371, 2022. 30, 36
- [86] Li, Jialong, Hongxia Wang, Peisong He, Sani M. Abdullahi e Bin Li: *Long-term variable q transform: A novel time-frequency transform algorithm for synthetic speech detection*. Digital Signal Processing, 120:103256, janeiro 2022, ISSN 1051-2004. <https://www.sciencedirect.com/science/article/abs/pii/S1051200421002955>. 30
- [87] Wu, Haibin, Heng Cheng Kuo, Naijun Zheng, Kuo Hsuan Hung, Hung Yi Lee, Yu Tsao, Hsin Min Wang e Helen Meng: *Partially fake audio detection by self-attention-based fake span discovery*. Em *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, páginas 9236–9240, 2022. 31
- [88] Wang, Chenglong, Jiangyan Yi, Jianhua Tao, Haiyang Sun, Xun Chen, Zhengkun Tian, Haoxin Ma, Cunhang Fan e Ruibo Fu: *Fully automated end-to-end fake audio detection*. Em *DDAM 2022 - Proceedings of the 1st International Workshop on*

- Deepfake Detection for Audio Multimedia*, páginas 27–33. Association for Computing Machinery, Inc, outubro 2022, ISBN 9781450394963. 31
- [89] Baevski, Alexei, Henry Zhou, Abdelrahman Mohamed e Michael Auli: *wav2vec 2.0: A framework for self-supervised learning of speech representations*. Em *Advances in Neural Information Processing Systems*, volume 33, páginas 12449–12460, 2020. 31
 - [90] Liu, Hanxiao, Karen Simonyan e Yiming Yang: *Darts: Differentiable architecture search*. Em *International Conference on Learning Representations (ICLR)*, 2019. 31
 - [91] Zhang, You, Fei Jiang e Zhiyao Duan: *One-class learning towards synthetic voice spoofing detection*. *IEEE Signal Processing Letters*, 28:937–941, 2021. 32
 - [92] Yi, Jiangyan, Ruibo Fu, Jianhua Tao, Shuai Nie, Haoxin Ma, Chenglong Wang, Tao Wang, Zhengkun Tian, Xiaohui Zhang, Ye Bai, Cunhang Fan, Shan Liang, Shiming Wang, Shuai Zhang, Xinrui Yan, Le Xu, Zhengqi Wen, Haizhou Li, Zheng Lian e Bin Liu: *Add 2022: the first audio deep synthesis detection challenge*. 32, 40
 - [93] Zhang, Binbin, Peidong Wang, Jian Wu, Haizhou Li, Xiangyu Zhang, Xuejing Chen e Zhihong Zhu: *Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline*. Em *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, páginas 5200–5204. IEEE, 2017. 32
 - [94] Sun, Chengzhe, Shan Jia, Shuwei Hou e Siwei Lyu: *Ai-synthesized voice detection using neural vocoder artifacts*, 2023. <https://github.com/csun22/Synthetic->. 32
 - [95] Zen, Heiga, Viet Dang, Rob Clark, Yu Zhang, Ron J. Weiss, Ye Jia e Zhifeng Chen: *Libritts: A corpus derived from librispeech for text-to-speech*. Em *Proc. Interspeech*, páginas 1526–1530, 2019. 32, 36, 46
 - [96] Li, Xin, Hao Zhao, Weicheng Liu, Meng Zhong, Zhiyao Chen, Yang Gao, Kai Yu e Shanshan Zhang: *Detecting unknown speech spoofing algorithms with nearest neighbors*. Em *Proceedings of the 2nd Audio Deepfake Detection Challenge Workshop (ADD 2023)*, volume 3597, páginas 1–8. CEUR Workshop Proceedings, 2023. <https://ceur-ws.org/Vol-3597/paper15.pdf>. 33, 41, 50
 - [97] Desplanques, Brecht, Jenthe Thienpondt e Kris Demuynck: *Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdn based speaker verification*. Em *Proceedings of Interspeech 2020*, páginas 3830–3834, 2020. 33, 41
 - [98] Yi, Jiangyan, Jianhua Tao, Ruibo Fu, Xinrui Yan, Chenglong Wang, Tao Wang, Chu Yuan Zhang, Xiaohui Zhang, Yan Zhao, Yong Ren, Le Xu, Junzuo Zhou, Hao Gu, Zhengqi Wen, Shan Liang, Zheng Lian, Shuai Nie e Haizhou Li: *Add 2023: the second audio deepfake detection challenge*. arXiv preprint arXiv:2305.13774, 2023. <https://arxiv.org/abs/2305.13774>. 33, 41
 - [99] Pan, Zihan, Tianchi Liu, Hardik B. Sailor e Qiongqiong Wang: *Attentive merging of hidden embeddings from pre-trained speech model for anti-spoofing detection*. Em *Proc. Interspeech 2024*, páginas 2090–2094, 2024. 33

- [100] Chen, Sanyuan, Chengyi Wang, Yu Wu, Zhuo Chen, Jinyu Chen, Shujie Zhao, Yao Qian Liu, Shuai Wei, Jiayu Li, Ming Zhou, Yujun Xiao, Li Deng, Yifan Gong e Furu Wei: *Wavlm: Large-scale self-supervised pre-training for full stack speech processing*. Em *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, páginas 24556–24569, 2022. 33, 34
- [101] Du, Renmingyue, Jixun Yao, Qiuqiang Kong e Yin Cao: *Towards out-of-distribution detection in vocoder recognition via latent feature reconstruction*. junho 2024. <http://arxiv.org/abs/2406.02233>. 34, 50
- [102] Streijl, Robert C., Stefan Winkler e David S. Hands: *Mean opinion score (MOS) revisited: methods and applications, limitations and alternatives*. *Multimedia Systems*, 22(2):213–227, 2016. <https://link.springer.com/article/10.1007/s00530-014-0446-1>. 34
- [103] Cooper, Erica, Wen Chin Huang, Tomoki Toda e Junichi Yamagishi: *Generalization Ability of MOS Prediction Networks*. Em *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, páginas 7967–7971. IEEE, 2022. 35
- [104] Gasenzer, Konstantin e Moritz Wolter: *Towards generalizing deep-audio fake detection networks*. *Transactions on Machine Learning Research*, 2024, ISSN 2835-8856. <https://openreview.net/forum?id=RGewtLtvHz>. 35
- [105] Klein, Nicholas, Tianxiang Chen, Hemlata Tak, Ricardo Casal e Elie Khoury: *Source tracing of audio deepfake systems*. julho 2024. <http://arxiv.org/abs/2407.08016><http://dx.doi.org/10.21437/Interspeech.2024-1283>. 35
- [106] Müller, Nicolas M., Piotr Kawa, Wei Herng Choong, Edresson Casanova, Eren Gölge, Thorsten Müller, Piotr Syga, Philip Sperl e Konstantin Böttinger: *Mlaad: The multi-language audio anti-spoofing dataset*. Em *2024 International Joint Conference on Neural Networks (IJCNN)*, páginas 1–7, 2024. 35, 36
- [107] He, Kaiming, Xiangyu Zhang, Shaoqing Ren e Jian Sun: *Deep residual learning for image recognition*. Em *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, páginas 770–778, 2016. 35, 50, 58
- [108] Deng, Junlong, Yanzhen Ren, Tong Zhang, Hongcheng Zhu e Zongkun Sun: *Vfd-net: Vocoder fingerprints detection for fake audio*. Em *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, páginas 12151–12155. Institute of Electrical and Electronics Engineers Inc., 2024. 36
- [109] Yamagishi, Junichi, Christophe Veaux e Kirsten MacDonald: *Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit*. <https://datashare.ed.ac.uk/handle/10283/3443>, 2019. Accessed: 2025-08-01. 36, 44, 56
- [110] Sivaraman, Ganesh, Hemlata Tak e Elie Khoury: *Investigating voiced and unvoiced regions of speech for audio deepfake detection*. Em *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. Institute of Electrical and Electronics Engineers Inc., 2025, ISBN 9798350368741. 36, 66

- [111] De Cheveigné, Alain e Hideki Kawahara: *Yin, a fundamental frequency estimator for speech and music*. The Journal of the Acoustical Society of America, 111(4):1917–1930, 2002. 36
- [112] McFee, Brian, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg e Oriol Nieto: *librosa: Audio and music signal analysis in python*. Em *Proceedings of the 14th Python in Science Conference*, páginas 18–25, Austin, TX, 2015. Python Software Foundation. 36
- [113] Google: *Webrtc voice activity detector (vad)*. <https://github.com/wiseman/py-webrtcvad>. Acesso em: 2025-08-01. 36
- [114] Ren, Hainan, Li Lin, Chun Hao Liu, Xin Wang e Shu Hu: *Improving generalization for ai-synthesized voice detection*. Proceedings of the AAAI Conference on Artificial Intelligence, 39:20165–20173, abril 2025, ISSN 2374-3468. <https://ojs.aaai.org/index.php/AAAI/article/view/34221>. 37
- [115] Sahidullah, Md, Tomi Kinnunen e Humberto Delgado: *Asvspoof 2015: The first automatic speaker verification spoofing and countermeasures challenge*. Em *INTER-SPEECH*, Dresden, Germany, 2015. 38
- [116] *ASVspoof 2019: The Automatic Speaker Verification Spoofing and Countermeasures Challenge Evaluation Plan*. <http://www.asvspoof.org/asvspoof2019/asvspoof2019evaluationplan.pdf>, 2019. Online; accessed 2025-08-11. 39
- [117] ID R&D: *Id r&d ranks first in detecting synthetic speech in global asvspoof challenge*, 2019. <https://www.idrnd.ai/id-rd-ranks-first-in-detecting-synthetic-speech-in-global-asvspoof-challenge/>, Acessado em: 11 ago. 2025. 39
- [118] Yamagishi, Junichi, Xin Wang, Massimiliano Todisco, Md Sahidullah, Jose Patino, Andreas Nautsch, Xuechen Liu, Kong Aik Lee, Tomi Kinnunen, Nicholas Evans e Héctor Delgado: *Asvspoof 2021: accelerating progress in spoofed and deepfake speech detection*. Em *Proceedings of the 2021 Edition of the Automatic Speaker Verification and Spoofing Countermeasures Challenge*. ISCA, setembro 2021. 39
- [119] Zhang, Qishan, Shuangbing Wen e Tao Hu: *Audio deepfake detection with self-supervised XLS-R and SLS classifier*. Em *Proceedings of the 32nd ACM International Conference on Multimedia (MM '24)*, páginas 6765–6773, Melbourne, VIC, Australia, 10–11 2024. 40
- [120] Wang, Xin, Hector Delgado, Hemlata Tak, Jee weon Jung, Hye jin Shim, Massimiliano Todisco, Ivan Kukanov, Xuechen Liu, Md Sahidullah, Tomi Kinnunen, Nicholas Evans, Kong Aik Lee e Junichi Yamagishi: *Asvspoof 5: Crowdsourced speech data, deepfakes, and adversarial attacks at scale*. Em *Proc. ASVspoof Challenge Workshop*, Kos, Greece, August 2024. <http://www.asvspoof.org/>. 40
- [121] Pratap, Vineel, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve e Ronan Collobert: *Mls: A large-scale multilingual dataset for speech research*. Em *Proceedings of Interspeech*, páginas 2757–2761, 2020. 40

- [122] Shi, Yao, Hui Bu, Xin Xu, Shaoji Zhang e Ming Li: *Aishell-3: A multi-speaker mandarin tts corpus*. páginas 2756–2760, agosto 2021. 41
- [123] Ito, Keith e Linda Johnson: *The lj speech dataset*. <https://keithito.com/LJ-Speech-Dataset/>, 2017. 46
- [124] Leclère, Thibaud, David Théry, Mathieu Lavandier e John F. Culling: *Speech intelligibility for target and masker with different spectra*. Em *Speech, Noise and Hearing: Innovations in Signal Processing and Hearing Protection*, páginas 257–266. Springer, 2015. 48
- [125] Oppenheim, Alan V. e Ronald W. Schafer: *Discrete-Time Signal Processing*. Pearson, Upper Saddle River, NJ, 3rd edição, 2010. 48
- [126] Boutell, Thomas: *Png (portable network graphics) specification version 1.0*. W3C Recommendation, 1997. <https://www.w3.org/TR/REC-png>, Accessed: 2025-10-23. 48, 57
- [127] Hunter, John D.: *Matplotlib: A 2d graphics environment*. Computing in Science & Engineering, 9(3):90–95, 2007. <https://matplotlib.org/stable/users/explain/colors/colormaps.html>, Acessado em 22 de outubro de 2025. 49
- [128] Krizhevsky, Alex, Ilya Sutskever e Geoffrey E. Hinton: *Imagenet classification with deep convolutional neural networks*. Em *Proceedings of the 25th International Conference on Neural Information Processing Systems (NeurIPS)*, volume 1, páginas 1097–1105, 2012. 50, 58
- [129] Simonyan, Karen e Andrew Zisserman: *Very deep convolutional networks for large-scale image recognition*. CoRR, abs/1409.1556, 2014. <https://api.semanticscholar.org/CorpusID:14124313>. 50, 58
- [130] Reynolds, Douglas A.: *An overview of automatic speaker recognition technology*. Em *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, páginas IV–4072–IV–4075, Orlando, FL, USA, 2002. IEEE. 51
- [131] Team, Librosa Development: *librosa.mel_frequencies — librosa 0.11.0 documentation*. https://librosa.org/doc/main/generated/librosa.mel_frequencies.html, 2025. Acessado em 3 de outubro de 2025. 57
- [132] Rumelhart, David E, Geoffrey E Hinton e Ronald J Williams: *Learning representations by back-propagating errors*. Nature, 323(6088):533–536, 1986. <https://doi.org/10.1038/323533a0>. 58
- [133] Kingma, Diederik P. e Jimmy Ba: *Adam: A method for stochastic optimization*. Em *International Conference on Learning Representations (ICLR)*, 2015. 58
- [134] Selvaraju, Ramprasaath R., Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh e Dhruv Batra: *Grad-cam: Visual explanations from deep networks via gradient-based localization*. Em *2017 IEEE International Conference on Computer Vision (ICCV)*, páginas 618–626, 2017. 60

- [135] Zhou, Bolei, Yiyou Sun, David Bau e Antonio Torralba: *Revisiting the importance of individual units in CNNs via ablation*. arXiv preprint arXiv:1806.02891, 2018. <https://arxiv.org/abs/1806.02891>. 60, 68
- [136] McInnes, Leland, John Healy, Nathaniel Saul e Lukas Großberger: *Umap: Uniform manifold approximation and projection*. Journal of Open Source Software, 3(29):861, 2018. <https://doi.org/10.21105/joss.00861>. 68
- [137] Zhao, Ming Jie, Narayanan Edakunni, Adam Pocock e Gavin Brown: *Beyond fano’s inequality: Bounds on the optimal f-score, BER, and cost-sensitive risk and their implications*. Journal of Machine Learning Research, 14:1033–1090, abril 2013. Submitted 8/12; revised 1/13; published 4/13. 76

Anexo I

Curvas de Loss e de aEER de treinamento dos modelos usados em testes de generalização

Este anexo apresenta as curvas de evolução da função de perda (*Loss*) e da métrica *average Equal Error Rate* (aEER) obtidas durante o treinamento dos 21 modelos utilizados nos testes de generalização. Essas curvas permitem analisar a convergência dos modelos, verificar a estabilidade do processo de treinamento e identificar possíveis indícios de *overfitting*. Apesar das diferentes configurações de treinamento adotadas, observa-se elevada similaridade entre as curvas dos 21 experimentos, sugerindo que o processo de treinamento apresentou comportamento consistente e convergência em todos os casos. Os resultados complementam a discussão apresentada na Seção 5.2.3, fornecendo uma visão detalhada do comportamento de cada treinamento ao longo das épocas.

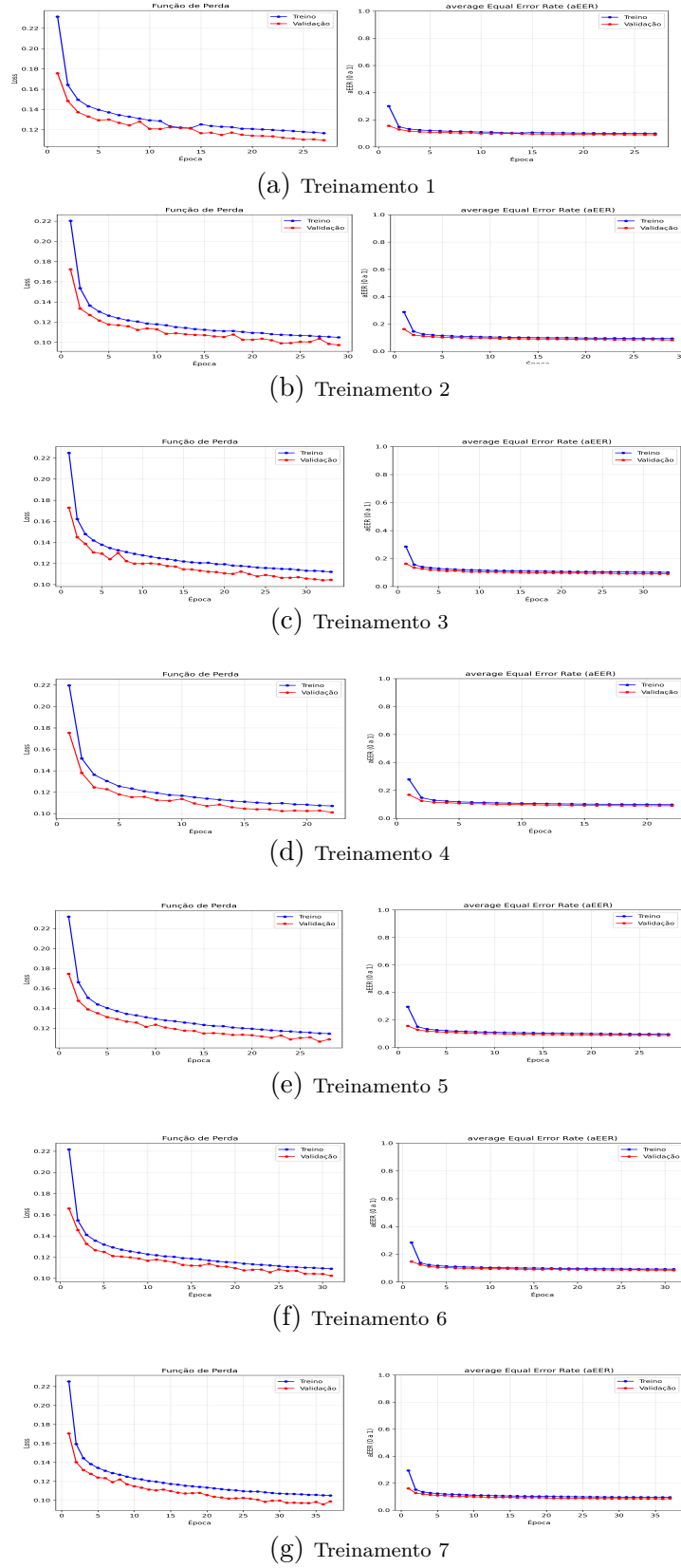


Figura I.1: Curvas de Loss e de aEER de treinamento dos modelos usados em testes de generalização

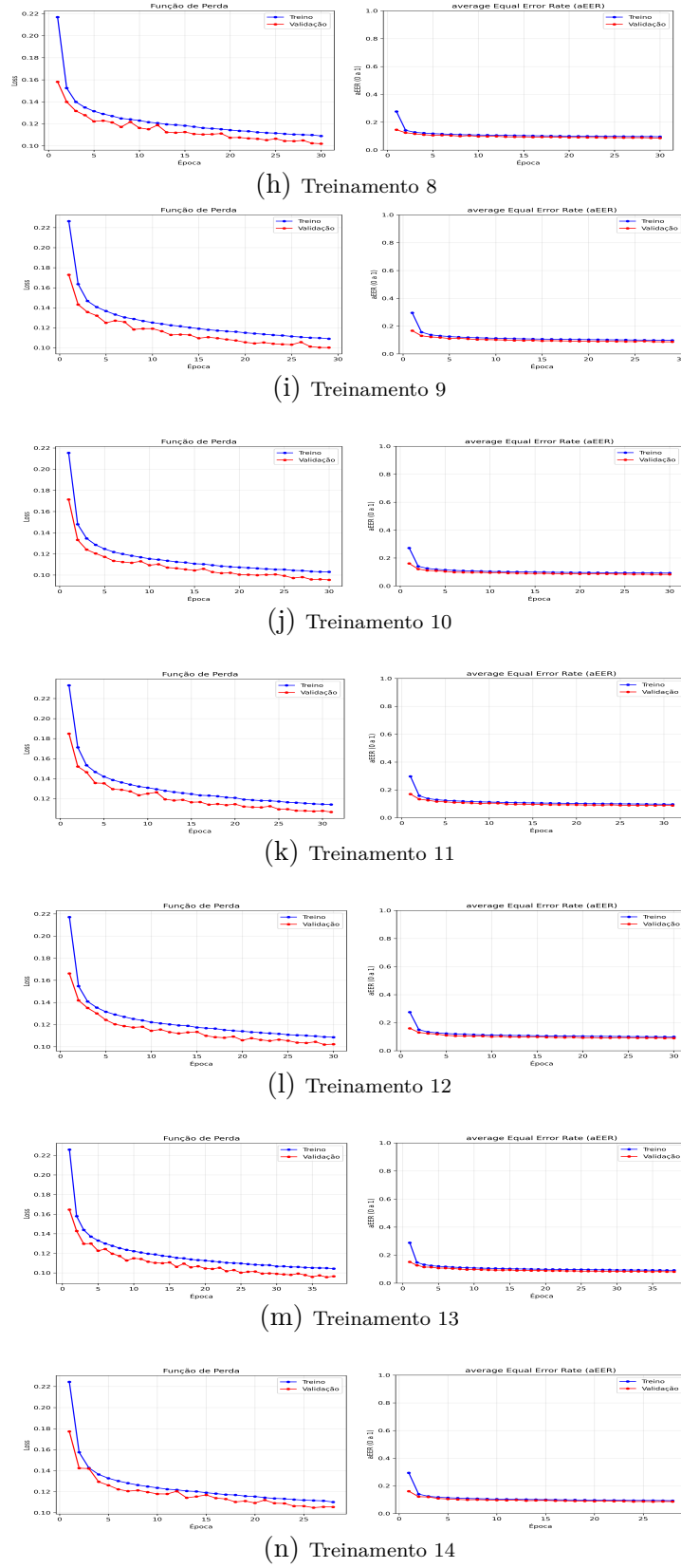
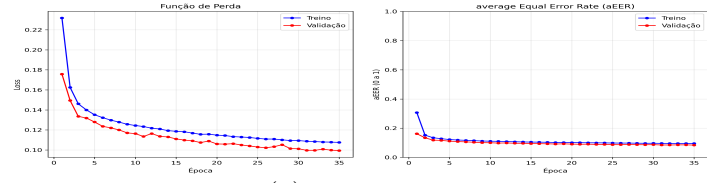
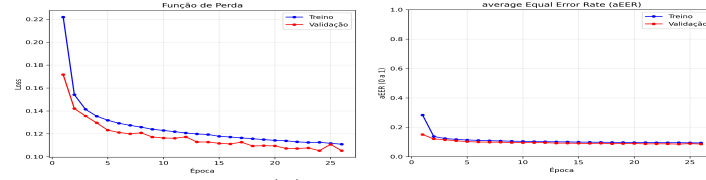


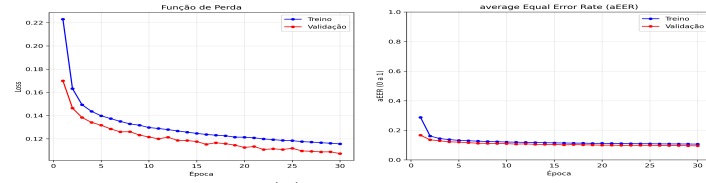
Figura I.1: Curvas de Loss e de aEER de treinamento dos usados em testes de generalização (continuação)



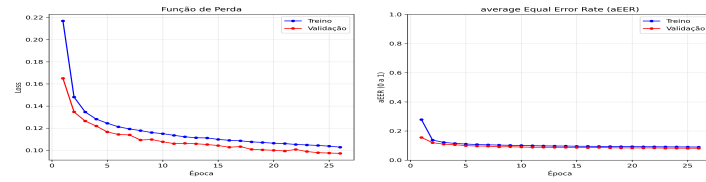
(o) Treinamento 15



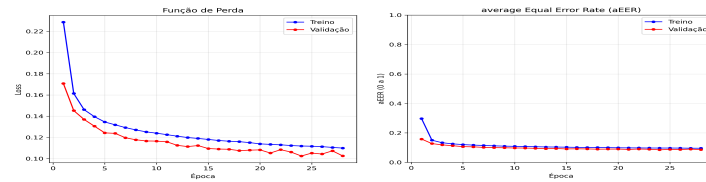
(p) Treinamento 16



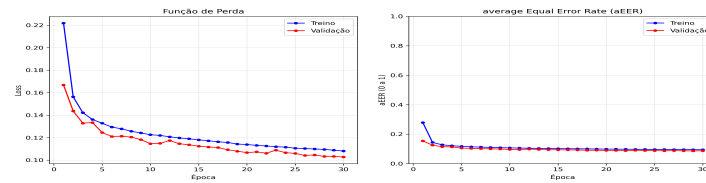
(q) Treinamento 17



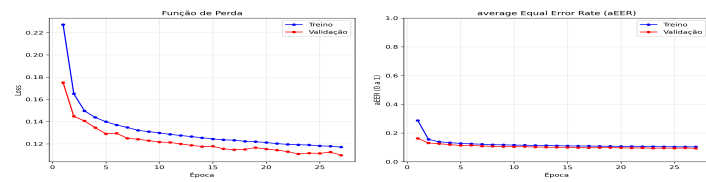
(r) Treinamento 18



(s) Treinamento 19



(t) Treinamento 20



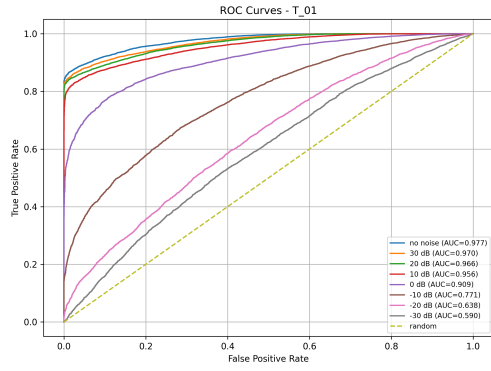
(u) Treinamento 21

Figura I.1: Curvas de Loss e de aEER de treinamento dos modelos usados em testes de generalização (continuação)

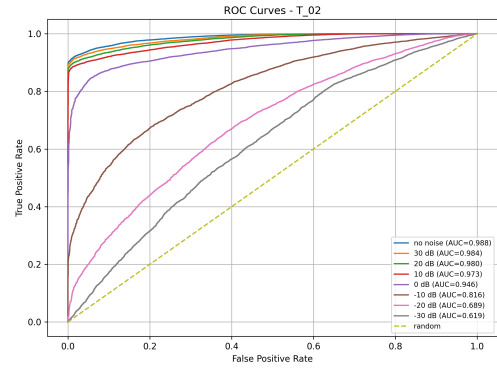
Anexo II

Curvas ROC da saída bonafide dos modelos usados em testes de generalização

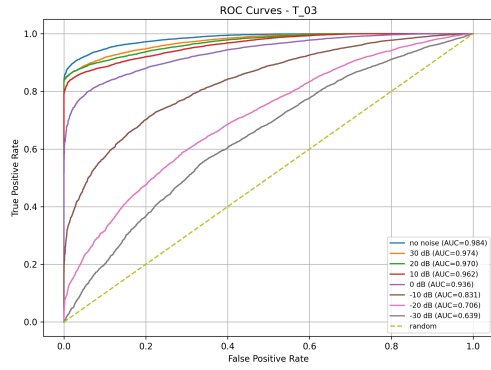
Este anexo reúne as curvas *Receiver Operating Characteristic* (ROC) calculadas a partir da saída associada à classe *bonafide* para os 21 modelos utilizados nos testes de generalização. Essas curvas permitem avaliar a capacidade discriminativa dos modelos em diferentes configurações de treinamento, evidenciando a relação entre taxa de verdadeiros positivos e taxa de falsos positivos para diferentes limiares de decisão. De modo geral, observa-se grande semelhança entre as curvas obtidas nos diferentes experimentos, sugerindo que os modelos alcançaram níveis comparáveis de capacidade discriminativa. Os gráficos complementam a análise apresentada na Seção 5.2.3, permitindo a inspeção individual do desempenho de cada modelo.



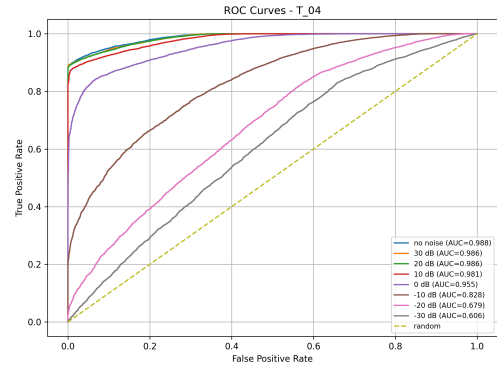
(a) Treinamento 1



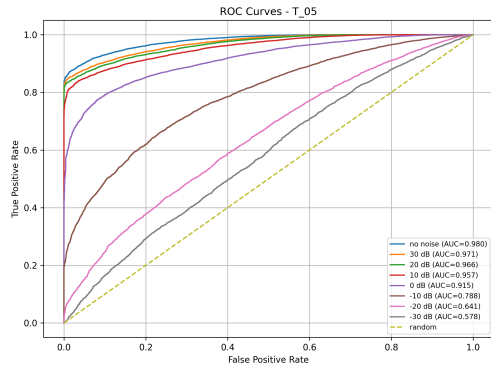
(b) Treinamento 2



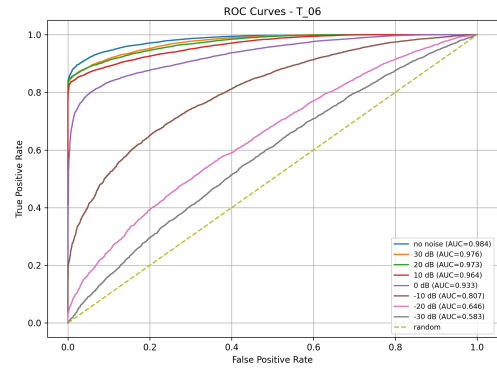
(c) Treinamento 3



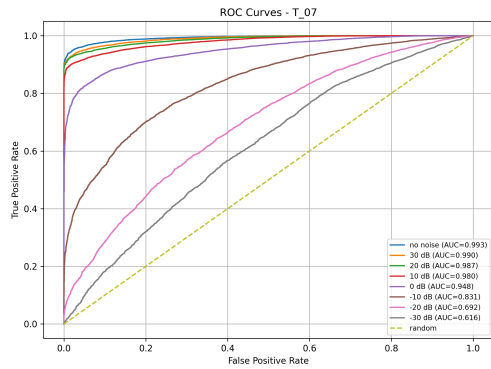
(d) Treinamento 4



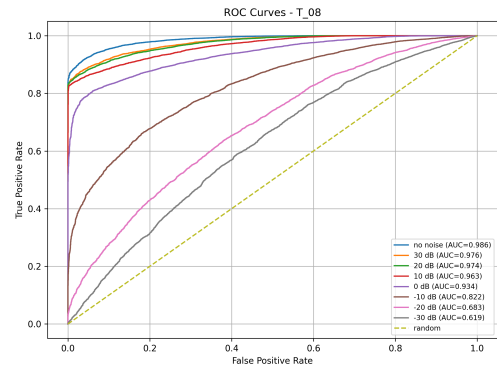
(e) Treinamento 5



(f) Treinamento 6

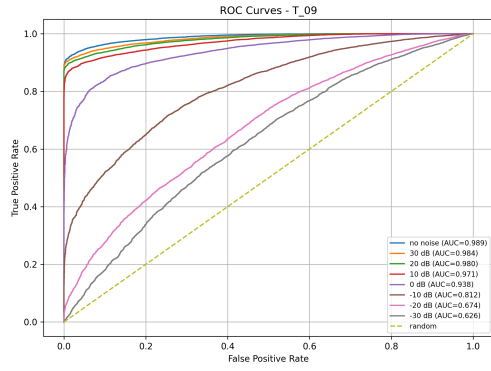


(g) Treinamento 7

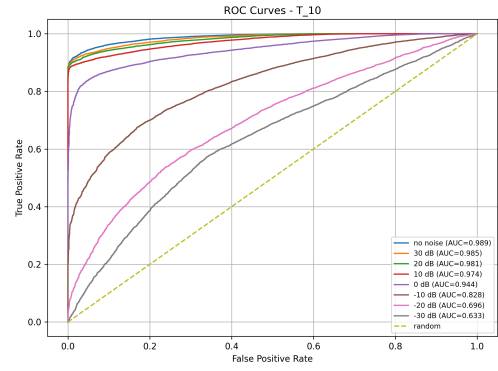


(h) Treinamento 8

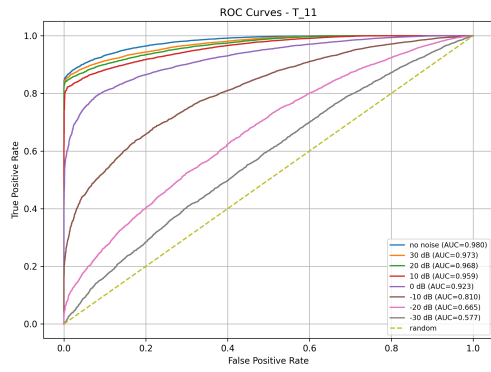
Figura II.1: Curvas ROC da saída *bonafide* dos modelos a serem usados em testes de generalização



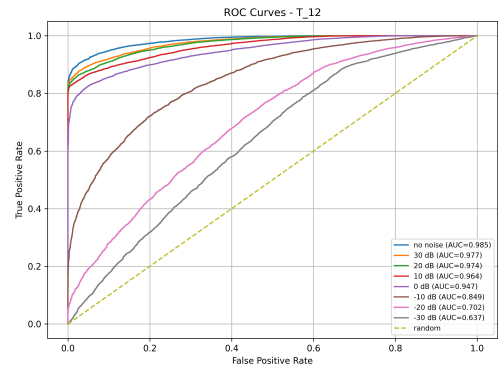
(i) Treinamento 9



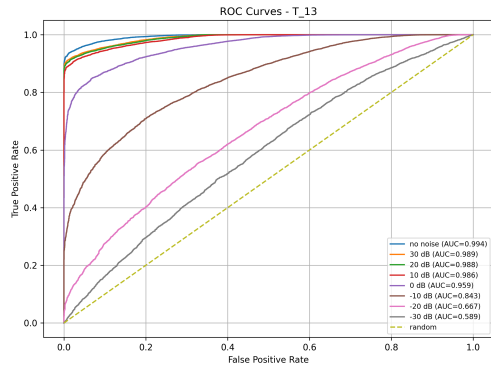
(j) Treinamento 10



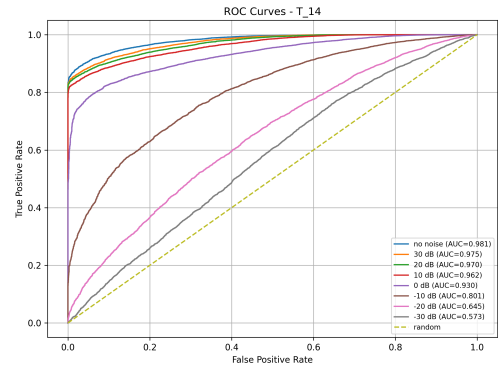
(k) Treinamento 11



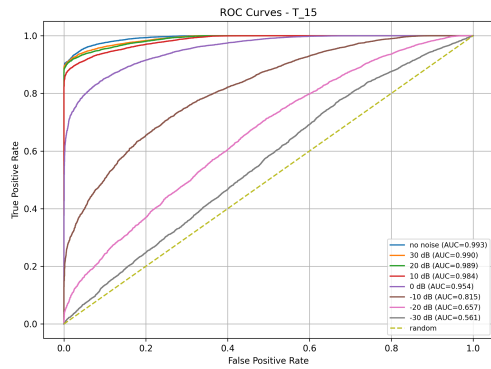
(l) Treinamento 12



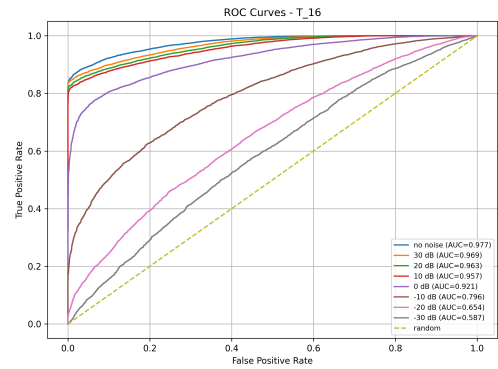
(m) Treinamento 13



(n) Treinamento 14

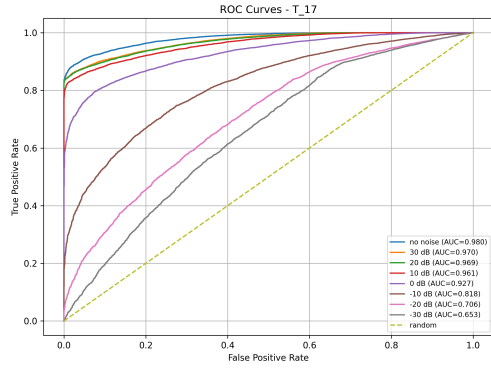


(o) Treinamento 15

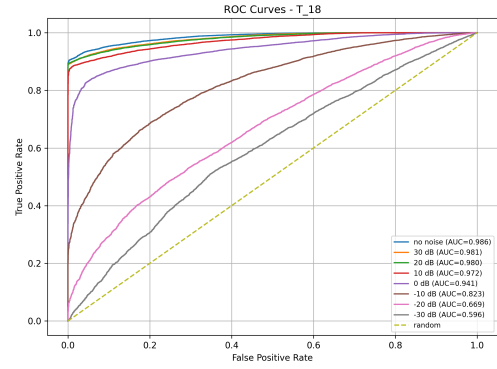


(p) Treinamento 16

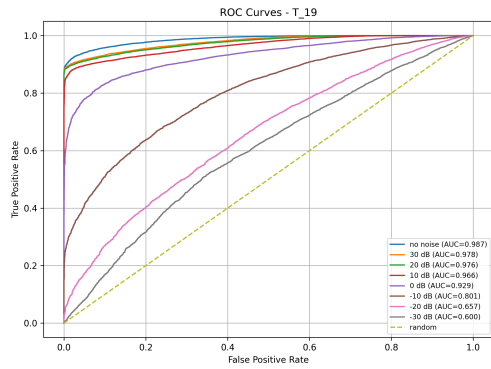
Figura II.1: Curvas ROC da saída bonafide dos modelos a serem usados em testes de generalização (continuação)



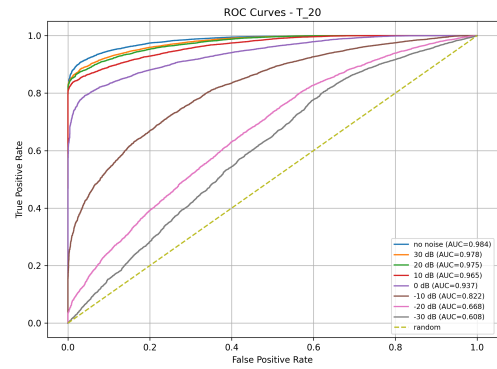
(q) Treinamento 17



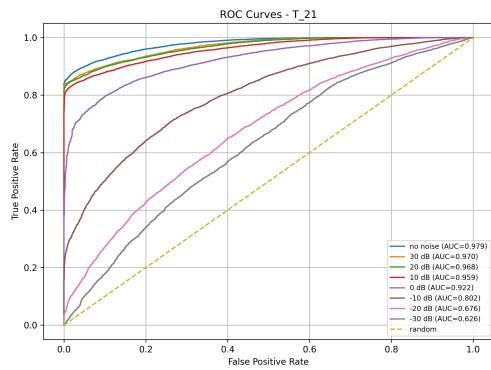
(r) Treinamento 18



(s) Treinamento 19



(t) Treinamento 20



(u) Treinamento 21

Figura II.1: Curvas ROC da saída *bonafide* dos modelos a serem usados em testes de generalização (continuação)

Anexo III

Desempenho dos modelos por *vocoder* em função do nível de ruído (BER)

Este anexo apresenta os resultados da análise de desempenho dos modelos em função do nível de ruído para cada *vocoder* externo. Para cada um dos 21 treinamentos são mostradas as curvas da taxa de erro balanceado (Balanced Error Rate — BER) em diferentes condições de relação sinal-ruído (SNR), permitindo observar como a degradação do sinal afeta a capacidade de detecção de áudios sintéticos. Os gráficos complementam os resultados resumidos apresentados no Capítulo 5, possibilitando uma análise detalhada da robustez dos modelos para cada *vocoder* individualmente.

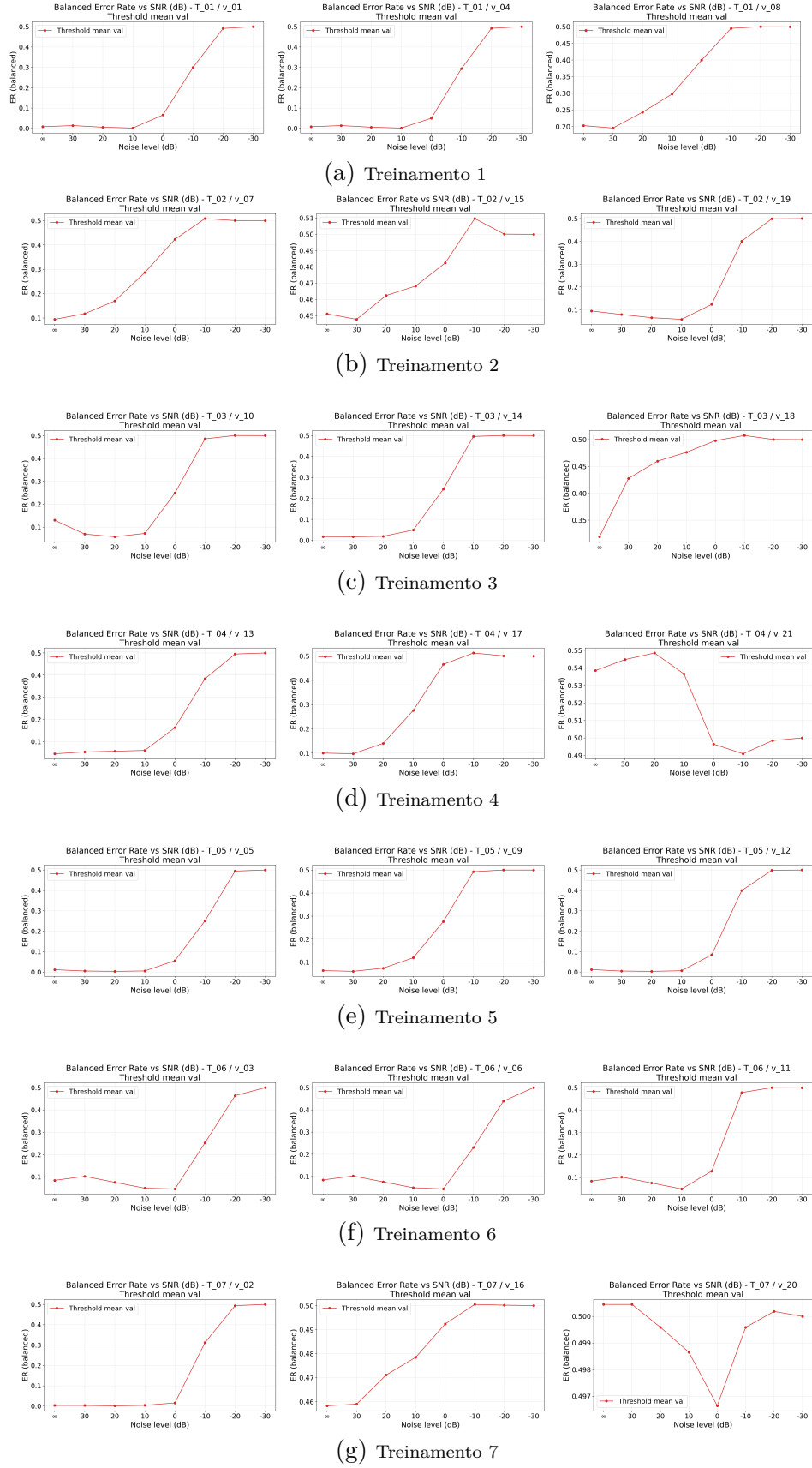


Figura III.1: Performance dos modelos para *vocoder* específico

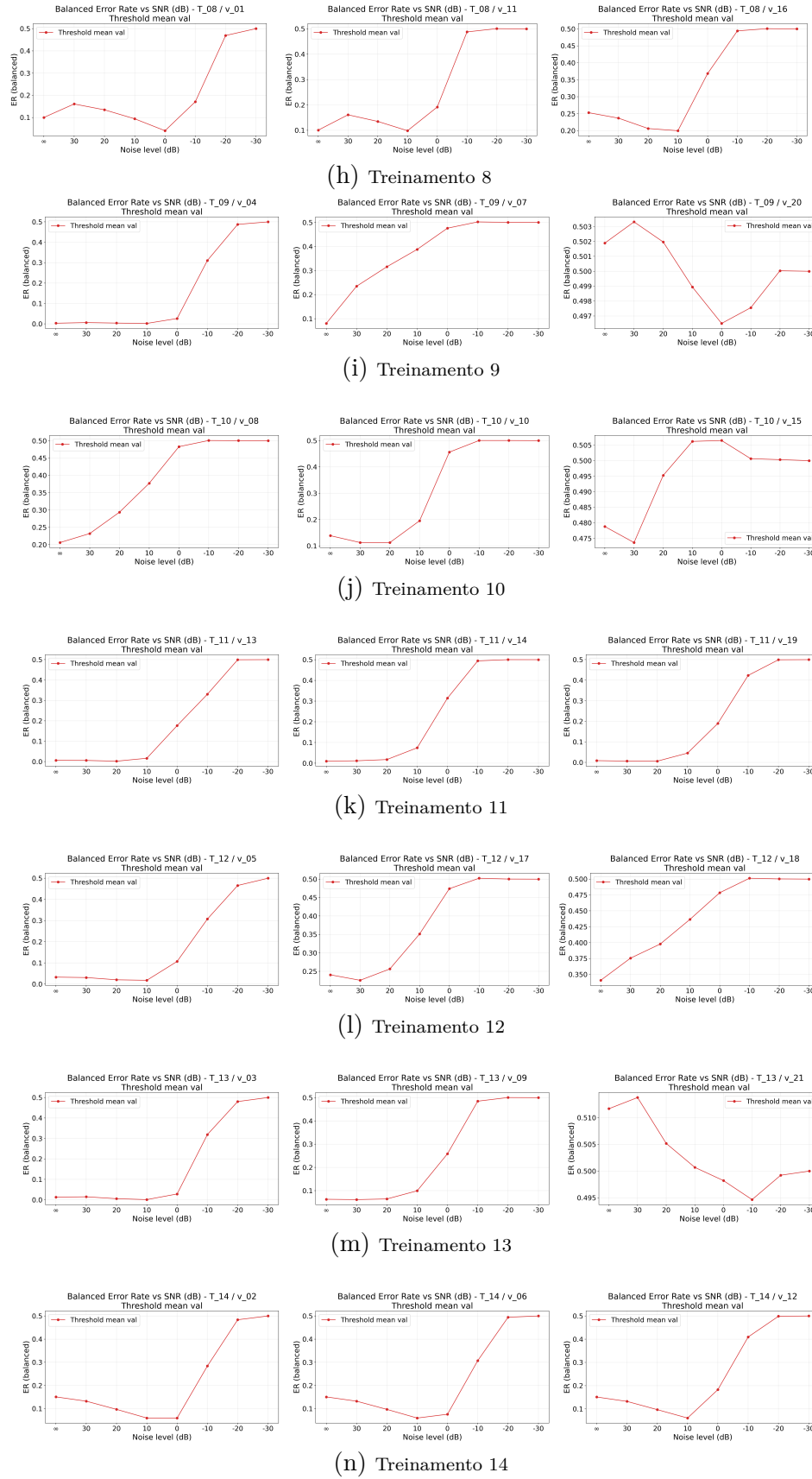
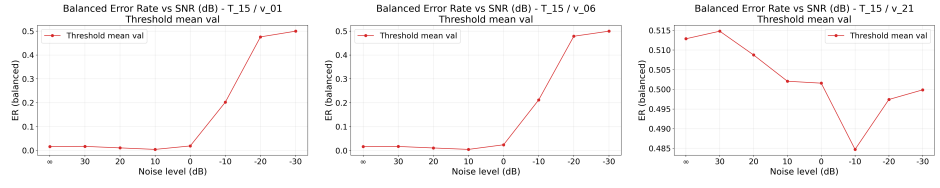
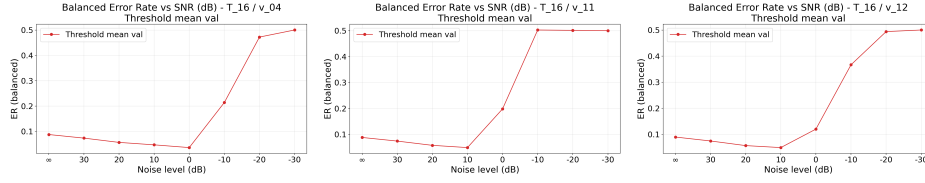


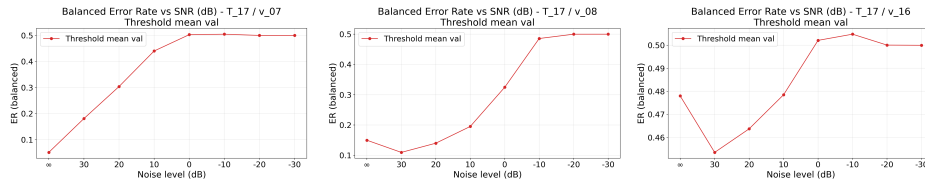
Figura III.1: Performance dos modelos para *vocoder* específico (continuação)



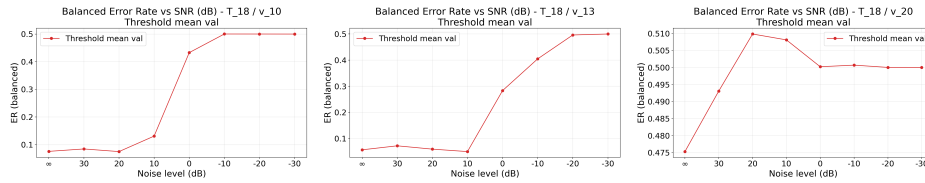
(o) Treinamento 15



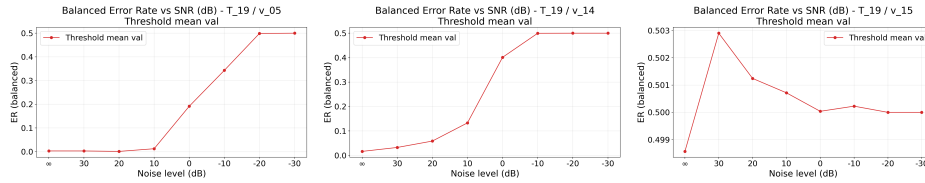
(p) Treinamento 16



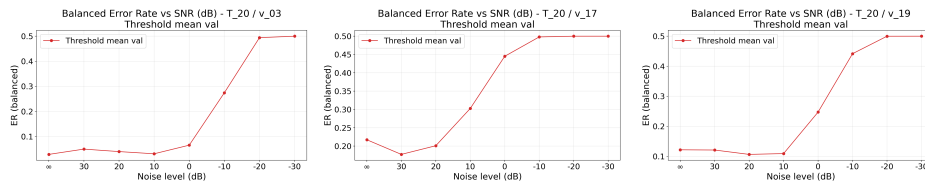
(q) Treinamento 17



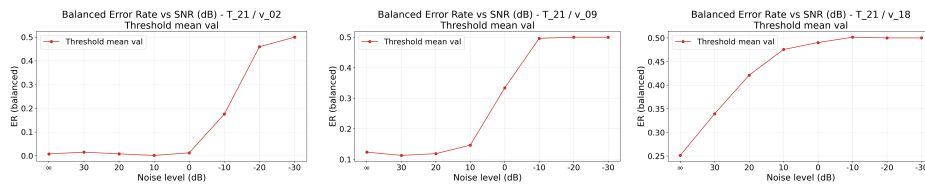
(r) Treinamento 18



(s) Treinamento 19



(t) Treinamento 20



(u) Treinamento 21

Figura III.1: Performance dos modelos para *vocoder* específico (continuação)