UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Campus de São José do Rio Preto

Pedro Henrique de Medeiros Gatti

Estudo Comparativo entre duas abordagens para Remoção de Ruído em sinais de fala: Filtro de Wiener e Autoencoders

São José do Rio Preto

2024

Pedro Henrique de Medeiros Gatti

Estudo Comparativo entre duas abordagens para Remoção de Ruído em sinais de fala: Filtro de Wiener e Autoencoders

Monografia apresentada como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação, junto ao Departamento de Ciências de Computação e Estatística do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista "Júlio de Mesquita Filho", Câmpus de São José do Rio Preto.

Universidade Estadual Paulista "Júlio de Mesquita Filho" - (UNESP)

Instituto de Biociências, Letras e Ciências Exatas - (IBILCE)

Departamento de Ciências de Computação e Estatística

Bacharelado em Ciência da Computação

Orientador: Rodrigo Capobianco Guido

São José do Rio Preto

2024

G263e

Gatti, Pedro Henrique de Medeiros

Estudo comparativo entre duas abordagens para remoção de ruído em sinais de fala : filtro de Wiener e autoencoders / Pedro Henrique de Medeiros Gatti. -- São José do Rio Preto, 2024

40 p.

Trabalho de conclusão de curso (Bacharelado - Ciência da Computação) - Universidade Estadual Paulista (UNESP), Instituto de Biociências Letras e Ciências Exatas, São José do Rio Preto

Orientador: Rodrigo Capobianco Guido

1. Processamento de sinais. 2. Speech Enhancement. 3. Filtro de Wiener. 4. Autoencoders. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Dados fornecidos pelo autor(a).

Pedro Henrique de Medeiros Gatti

Estudo Comparativo entre duas abordagens para Remoção de Ruído em sinais de fala: Filtro de Wiener e Autoencoders

Monografia apresentada como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação, junto ao Departamento de Ciências de Computação e Estatística do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista "Júlio de Mesquita Filho", Câmpus de São José do Rio Preto.

Banca Examinadora:

Prof. Dr. Rodrigo Capobianco Guido
UNESP - Câmpus de São José do Rio Preto
Orientador

Profa. Dra. Adriana Barbosa Santos
UNESP - Câmpus de São José do Rio Preto

Prof. Dr. Lucas Correia Ribas
UNESP - Câmpus de São José do Rio Preto

São José do Rio Preto
2024

Dedico esse trabalho a minha família.

Agradecimentos

Agradeço primeiramente aos meus pais, Antônio Rogério Gatti e Luciana Christina de Medeiros Gatti, por todo amor e paciência que me ajudaram a chegar até aqui. Agradeço também a minha irmã Ana Beatriz de Medeiros Gatti, por me incentivar e me apoiar e fazer de mim a pessoa que eu sou hoje.

Aos meus amigos, que fizeram dessa jornada menos solitária e com quem pude compartilhar experiências e conquistas ao longo da graduação.

Ao meu Orientador Rodrigo Capobianco Guido, por sua dedicação e disponibilidade que foram fundamentais para realização deste trabalho.

"Só se pode alcançar um grande êxito quando nos mantemos fiéis a nós mesmos"
(Friedrich Nietzsche)

Resumo

A comunicação por voz é uma parte fundamental da vida cotidiana, desempenhando um papel crucial em diversas interações sociais e profissionais. Nesse contexto, o pré-processamento de sinais de voz se destaca como uma tarefa de grande importância, visando melhorar a clareza e a inteligibilidade da fala. Para esse fim, as técnicas de *Speech Enhancement* (SE) são aplicadas por meio de algoritmos que removem ruídos dos sinais de voz. Este trabalho investiga duas técnicas de remoção de ruído em sinais de voz: o Filtro de Wiener e Autoencoder, com um foco central na comparação dessas abordagens. Foram utilizadas métricas objetivas para avaliar cada técnica, incluindo a Distância Euclidiana e o desempenho computacional. O objetivo é entender qual das técnicas se mostra mais eficaz para os cenários analisados, contribuindo assim para o avanço das tecnologias de processamento de áudio e para a melhoria da qualidade da comunicação verbal.

Palavras-chave: Processamento de sinais. *Speech Enhancement*. Filtro de Wiener. Autoencoders.

Abstract

Voice communication is a fundamental part of everyday life, playing a crucial role in various social and professional interactions. In this context, the preprocessing of voice signals stands out as a task of great importance, aiming to enhance the clarity and intelligibility of speech. For this purpose, Speech Enhancement (SE) techniques are applied through algorithms that remove noise from voice signals. This work investigates two noise reduction techniques for voice signals: the Wiener Filter and Autoencoder, with a central focus on comparing these approaches. Objective metrics were used to evaluate each technique, including Euclidean Distance and computational performance. The goal is to understand which technique proves more effective for the analyzed scenarios, thereby contributing to advancements in audio processing technologies and improving the quality of verbal communication.

Keywords: *Signal processing. Speech Enhancement. Wiener Filter. Autoencoders.*

Sumário

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

DL	*Deep Learning*
BCE	Entropia Cruzada Binária
MSE	Erro Quadrático Médio
PDF	Função de Densidade de Probabilidade
IA	Inteligência Artificial
MMSE	Mínimo Erro Quadrático Médio
SE	*Speech Enhancement*

1 Introdução

Os sinais de fala englobam um conjunto rico de propriedades acústicas e linguísticas, abrangendo desde as unidades lexicais individuais, como fonemas e palavras, até as características dos falantes, suas intenções, ou até mesmo seu estado mental (CHUNG et al., 2019).

Por isso, a comunicação por voz se tornou uma parte onipresente da vida moderna, permeando diversos dispositivos e aplicações, desde smartphones e assistentes virtuais até sistemas de segurança e aparelhos auditivos. À medida que a tecnologia avança, métodos modernos de codificação de fala têm se concentrado em maximizar a qualidade do sinal transmitido, levando em conta tanto a inteligibilidade quanto a naturalidade, com novas abordagens que incluem o uso de redes neurais para melhorar a eficiência na transmissão (O'SHAUGHNESSY, 2023).

No entanto, a presença de ruído em ambientes reais é um desafio persistente no processamento de sinais de fala. Comumente os ruídos são divididos em quatro categorias básicas (BENESTY et al., 2009). O presente trabalho irá comtemplar técnicas de processamento de sinais que cobrem especificamente o ruído aditivo. Os sinais serão submetidos a técnica de *Speech Enhancement* (SE). O *SE* é um processo que visa aprimorar a qualidade da fala, reduzindo ruídos e outras perturbações indesejadas no sinal de áudio (CHHETRI et al., 2023).

Diversas técnicas de remoção de ruído em sinais de fala têm sido desenvolvidas ao longo das últimas décadas, buscando atenuar os efeitos do ruído e aprimorar a qualidade da fala. Uma dessas técnicas é o Filtro de Wiener, uma abordagem clássica baseada em estatística que realiza a remoção de ruído introduzindo alguma distorção no sinal de fala (CHEN et al., 2006). Outra técnica para remoção de ruído em sinais de fala é o Autoencoder, uma técnica de aprendizado de representação, especialmente utilizada em algoritmos de aprendizado não supervisionado (TSCHANNEN; BACHEM; LUCIC, 2018). A ideia central do autoencoding é aprender uma transformação que mapeia observações de alta dimensão para um espaço de representação de menor dimensão, permitindo que as observações originais sejam (aproximadamente) reconstruídas a partir dessa representação reduzida (TSCHANNEN; BACHEM; LUCIC, 2018). Para esse trabalho será usado um algoritmo de *denoising autoencoder*, uma simples variação do Autoencoder descrito acima.

1.1 Objetivos

Este trabalho propõe um estudo comparativo entre o Filtro de Wiener e o Autoencoder para a remoção de ruído em sinais de fala. O foco é comparar e avaliar a performance e a

capacidade de generalização de cada técnica para diferentes sinais contaminados com ruído, além de analisar a distorção introduzida por cada método e comparar seu custo computacional.

Os objetivos específicos deste trabalho são:

- Comparar a performance do Filtro de Wiener e do Autoencoder na remoção de ruído em sinais de fala, utilizando a Distância Euclidiana como métrica de comparação.
- Comparar o custo computacional de cada técnica, considerando o tempo de processamento e os recursos computacionais necessários.
- Identificar as vantagens e desvantagens de cada abordagem, elucidando os cenários em que cada técnica se mostra mais eficaz.

Através da realização desses objetivos, este trabalho busca contribuir para o avanço do processamento de sinais de fala, destacando os pontos positivos e negativos de cada técnica através de sua performance e aplicabilidade.

A hipótese central deste trabalho é que o Autoencoder apresenta uma performance superior à do Filtro de Wiener na remoção de ruído em sinais de fala, devido à sua capacidade de generalização e de lidar melhor com padrões de ruído. Com base nessa premissa, busca-se investigar os fatores que contribuem para essa diferença de desempenho, bem como realizar uma análise detalhada da performance e identificar em quais cenários cada técnica se destacaria.

1.2 Organização do trabalho

O texto vindouro do presente trabalho está organizado da seguinte forma:

- No Capítulo 2 apresenta-se uma série de trabalhos publicados envolvendo a utilização das técnicas de remoção de ruído em sinais de fala abordadas no trabalho. Além disso, estão presentes os principais conceitos e teorias que estão relacionados com o trabalho desenvolvido.
- No Capítulo 3 apresenta-se, com detalhes, todo o desenvolvimento do trabalho proposto e de que forma os conceitos discutidos no capítulo anterior foram utilizados.
- No Capítulo 4 relatam-se todos os resultados obtidos no trabalho, a partir dos testes de reconhecimento que foram realizados.
- No Capítulo 5 apresentam-se as conclusões sobre o trabalho.

2 Revisão Bibliográfica

2.1 Conceitos Fundamentais

Esta seção aborda os fundamentos teóricos que sustentam as técnicas utilizadas no presente estudo, apresentando uma análise detalhada das metodologias aplicadas para remoção de ruído em sinais de voz. As abordagens contempladas incluem o Filtro de Wiener e o Autoencoder, técnicas que foram escolhidas por sua capacidade de restaurar sinais contaminados. Além disso, serão detalhados conceitos matemáticos que serão fundamentais para a construção deste trabalho: o Ruído Gaussiano e a Distância Euclidiana. Cada conceito citado é introduzido nesta seção em termos de sua formulação matemática, arquitetura e áreas de aplicabilidade, buscando proporcionar uma compreensão sólida para as decisões metodológicas adotadas ao longo do trabalho.

2.1.1 Filtro de Wiener

O Filtro de Wiener é um método clássico de filtragem que busca minimizar o Erro Quadrático Médio (MSE) entre o sinal filtrado e o sinal original, considerando conhecimento prévio das características do ruído e do sinal desejado (ROUX; VINCENT, 2012). O Filtro de Wiener é invariante ao tempo, ou seja, suas características de resposta não mudam ao longo do tempo, dessa forma o filtro de Wiener trata todas as partes do sinal de maneira uniforme, independentemente do tempo em que cada amostra foi coletada ou processada (KAY, 1993). O objetivo do filtro de Wiener é calcular uma estimativa estatística de um sinal desconhecido usando um sinal relacionado como entrada e filtrando esse sinal conhecido para produzir a estimativa como saída.

Por exemplo, o sinal conhecido pode consistir em um sinal de interesse desconhecido que foi corrompido por ruído aditivo. O filtro de Wiener pode ser utilizado para filtrar o ruído do sinal corrompido, fornecendo uma estimativa do sinal subjacente de interesse. Sua eficácia decorre do fato de que ele é projetado para ser um estimador de Mínimo Erro Quadrático Médio (MMSE), reduzindo a discrepância média entre o sinal de saída e o sinal original (RIBAS et al., 2022).

A função de transferência do filtro é projetada para balancear a redução do ruído com a preservação das componentes frequenciais relevantes do sinal, adaptando-se de acordo com a densidade espectral de potência do ruído e do sinal em cada frequência (ROUX; VINCENT, 2012).

$$H(f) = \frac{S_{XX}(f)}{S_{XX}(f) + S_{NN}(f)} \tag{2.1.1}$$

onde $S_{XX}(f)$ é a densidade espectral de potência do sinal original e $S_{NN}(f)$ representa a densidade espectral de potência do ruído. Este modelo permite que o filtro ajuste a supressão do ruído de acordo com a relação de energia entre o sinal e o ruído em cada frequência, preservando as componentes mais importantes do sinal original enquanto reduz as do ruído.

Embora muito eficaz em ambientes com ruído estacionário, o filtro de Wiener enfrenta limitações quando o ruído varia ao longo do tempo (ruído não estacionário). Em cenários onde as propriedades do ruído mudam, a suposição de estacionariedade não se mantém, o que pode reduzir a precisão do filtro ao tentar capturar apenas o sinal desejado (OPPENHEIM; VERGHESE, 2017).

2.1.2 Autoencoder

Autoencoder é uma rede neural projetada para aprender uma representação comprimida e informativa dos dados, com o objetivo de reconstruir as entradas de maneira precisa, minimizando o erro entre a entrada e a saída (MICHELUCCI, 2022). Por ser um modelo não supervisionado, o Autoencoder se destaca em aplicações como redução de dimensionalidade, detecção de anomalias e, especialmente, remoção de ruído.

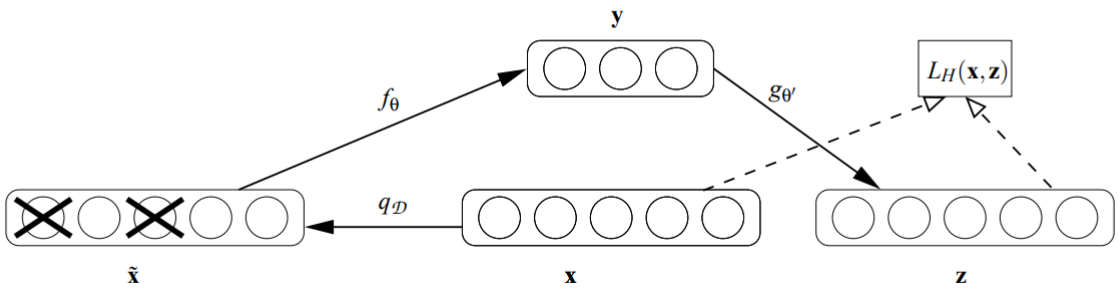
Neste trabalho será explorado o Autoencoder de remoção de ruído (*denoising autoencoder*). São redes neurais que utilizam um critério de reconstrução para aprender representações robustas a partir de dados ruidosos. Diferentemente do Autoencoder tradicional, que minimiza o erro de reconstrução entre a entrada e a saída para uma entrada intacta, o Autoencoder de remoção de ruído introduz um nível de corrupção nos dados de entrada, exigindo que o modelo reconstrua a versão "limpa"ou original da entrada a partir de uma representação corrompida (VINCENT et al., 2010).

A ideia é pegar cada vetor de entrada x_n e corrompê-lo com ruído para obter um vetor modificado x_{en}, que é então inserido em um autoencoder para produzir uma saída $y(x_{en}, w)$. A rede é treinada para reconstruir o vetor de entrada original, sem ruído, minimizando uma função de erro, como a soma dos quadrados, dada por (BISHOP; BISHOP, 2023):

$$E(w) = \sum_{n=1}^{N} \|y(x_{en}, w) - x_n\|^2 . \tag{2.1.2}$$

A arquitetura de um Autoencoder de remoção de ruído inclui três componentes principais: o **codificador**, que transforma a entrada corrompida x em uma representação latente h; a **representação latente**, que capta as características mais relevantes do sinal de entrada

Figura 2.1.1 – Modelo de um Autoencoder.



Fonte: Adaptado de (VINCENT et al., 2010)

mesmo em presença de ruído; e o **decodificador**, que reconstrói o sinal original a partir de h. Ao treinar o modelo com entradas corrompidas, ele aprende a distinguir entre ruído e características estruturais dos dados, fortalecendo a robustez da representação latente e facilitando a remoção de ruído em novos dados (VINCENT et al., 2010).

A figura 2.1.1 evidencia um modelo de Autoencoder para remoção de ruído, onde x é corrompido de forma estocástica (via q_D) para $\tilde{x}$. O Autoencoder, então, mapeia-o para y (através do codificador f_θ) e tenta reconstruir x através do decodificador g'_θ, produzindo a reconstrução z. O erro de reconstrução é medido pela perda $L_H(x, z)$.

Para descrever matematicamente o funcionamento de um Autoencoder, considera-se as funções de codificação e decodificação g e f:

1. **Codificação:** A função g mapeia a entrada $x \in \mathbb{R}^n$ para uma representação latente $h \in \mathbb{R}^q$, onde $q < n$, reduzindo a dimensionalidade e retendo informações essenciais:

$$h = g(x) \tag{2.1.3}$$

2. **Decodificação:** A função f reconstrói a entrada x a partir da representação latente h, gerando uma saída $\tilde{x}$ que se aproxima de x:

$$\tilde{x} = f(h) = f(g(x)) \tag{2.1.4}$$

3. **Função Objetivo:** O objetivo do Autoencoder é minimizar o erro de reconstrução entre x e $\tilde{x}$, comumente usando o MSE ou Entropia Cruzada Binária (BCE), conforme o tipo de dado:

$$\arg\min_{f,g} \mathbb{E}\left[\Delta(x, f(g(x)))\right] \tag{2.1.5}$$

A função de perda para o treinamento do Autoencoder de remoção de ruído é dada por:

$$\arg\min_{f,g} \mathbb{E}\left[\Delta(x, f(g(\tilde{x})))\right] \tag{2.1.6}$$

onde $\tilde{x}$ representa a entrada corrompida e $f(g(\tilde{x}))$ é a reconstrução da entrada limpa x gerada a partir da representação latente. A minimização desta função objetiva a redução do erro de reconstrução para que o Autoencoder seja capaz de remover o ruído de maneira eficaz.

Esse tipo de Autoencoder é especialmente eficaz em cenários onde se busca preservar a integridade do sinal, mesmo em ambientes ruidosos, destacando-se em tarefas de pré-processamento e compressão de dados. Ele também pode ser empilhado (stacked) para formar redes profundas, criando representações de alto nível cada vez mais robustas a partir de sucessivas camadas de Autoencoder de remoção de ruído. Esse empilhamento é particularmente útil para inicialização de redes profundas, uma vez que permite que o modelo alcance um ponto inicial vantajoso, melhorando seu desempenho em tarefas supervisionadas posteriores (VINCENT et al., 2010).

2.1.3 Ruído Gaussiano

O ruído gaussiano é frequentemente modelado como um processo estocástico em que suas amostras seguem uma distribuição normal. Nesse modelo, valores próximos à média são mais recorrentes, refletindo a natureza da interferência aleatória causada por diversos fatores em sistemas de comunicação, como componentes eletrônicos e vibrações ambientais (OTUNG, 2021). A Figura 2.1.2 ilustra uma distribuição gaussiana típica. Devido a essas características, optou-se por contaminar os sinais com ruído gaussiano para simular tais condições de interferência.

Matematicamente, o ruído gaussiano é modelado por uma Função de Densidade de Probabilidade (PDF) que descreve a amplitude do ruído em relação ao tempo, sendo definido por dois parâmetros principais: a média (geralmente zero) e a variância. A média determina o valor central em torno do qual as amplitudes do ruído se distribuem, enquanto a variância controla a dispersão dos valores, ou seja, a intensidade do ruído em relação ao sinal. Devido a essas características, o ruído gaussiano representa um modelo eficaz para testar a robustez de técnicas de filtragem e reconstrução de sinais, uma vez que permite uma simulação controlada de interferências.

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{2.1.7}$$

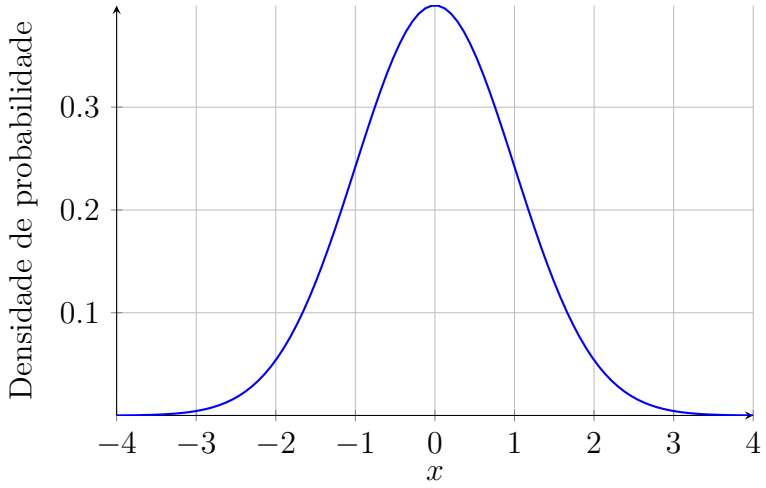
onde:

- x é a variável aleatória,

- μ é a média da distribuição,
- σ^2 é a variância da distribuição.

A presença do ruído adiciona desafios de processamento, pois exige que os métodos reconheçam e removam variações de amplitude distribuídas aleatoriamente ao longo do sinal, com o objetivo de recuperar a qualidade sonora original.

Figura 2.1.2 – Gráfico ilustrando a distribuição de um ruído gaussiano com média zero e variância unitária.



Fonte: Elaborado pelo autor.

2.1.4 Distância Euclidiana

Dada a sua simplicidade e interpretabilidade, a Distância Euclidiana será utilizada neste trabalho como métrica de comparação entre as duas técnicas de remoção de ruído.

Essa métrica matemática calcula a distância entre dois pontos em um espaço multidimensional. A Distância Euclidiana representa a menor distância reta entre dois pontos, sendo particularmente útil para avaliar a similaridade entre vetores ou sinais em espaços de alta dimensão. A fórmula da Distância Euclidiana entre dois vetores $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ e $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ em um espaço n-dimensional é expressa como:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{2.1.8}$$

onde:

- $\mathbf{x}$ e $\mathbf{y}$ representam os vetores (ou sinais) sendo comparados,
- x_i e y_i são as componentes individuais dos vetores $\mathbf{x}$ e $\mathbf{y}$,

- n é o número de dimensões do espaço (ou pontos no tempo para sinais).

Essa métrica permitirá avaliar o quanto cada técnica se aproxima do sinal de voz original após a remoção do ruído. Quanto menor a Distância Euclidiana entre o sinal processado e o sinal original, maior a precisão da técnica em preservar as características do sinal sem ruído.

2.2 Revisão dos trabalhos correlatos

A presente pesquisa teve início com uma busca por trabalhos correlatos na base de dados Web of Science. Utilizando as palavras-chave "Filtro de Wiener"e "Autoencoder"em combinação com "Speech Enhancement", foram encontrados diversos artigos relevantes para o tema deste trabalho. Após a leitura dos abstracts de 25 artigos, sete trabalhos foram selecionados para aprofundamento e incorporação na revisão da literatura, por apresentarem maior aderência aos objetivos e ao escopo da pesquisa. Os artigos selecionados são apresentados a seguir, com um resumo de suas principais contribuições e sua relevância para o presente estudo.

Um dos artigos selecionados, de Chhetri et al., aborda uma ampla gama de técnicas de aprimoramento de fala, incluindo métodos no domínio do tempo, métodos estatísticos como a filtragem de Wiener, e técnicas de domínio de transformação. O trabalho também discute as aplicações do aprimoramento de fala em diversas áreas, como telecomunicações e aparelhos auditivos, destacando os desafios enfrentados, como ruído não estacionário e fala sobreposta (CHHETRI et al., 2023). Ribas et al. propõem uma abordagem híbrida que combina o Filtro de Wiener com redes neurais profundas. Os autores argumentam que a estimativa direta da função do filtro por uma rede neural pode levar a um algoritmo mais robusto e simplificar o processo de treinamento, apresentando resultados experimentais que evidenciam sua eficácia (RIBAS et al., 2022).

Le Roux e Vincent discutem a importância da consistência entre os coeficientes da STFT na aplicação do Filtro de Wiener para separação de fontes de áudio. Os autores propõem algoritmos que impõem essa consistência como uma restrição, minimizando o efeito de artefatos sonoros e mostrando melhorias significativas na performance em comparação com o filtro de Wiener clássico (ROUX; VINCENT, 2012). Por sua vez, Borgstrom et al. apresentam um novo framework de Autoencoder multiescala, que captura informações mais ricas do sinal de fala ao processar diferentes bandas de frequência, resultando em um aprimoramento mais preciso (BORGSTRÖM; BRANDSTEIN, 2024).

Garg propõe um modelo que combina Long Short-Term Memory (LSTM) e um Filtro de Wiener adaptativo, utilizando a decomposição de modo empírico para extrair características relevantes do sinal de fala. Os resultados demonstram a eficácia na redução de ruído e na

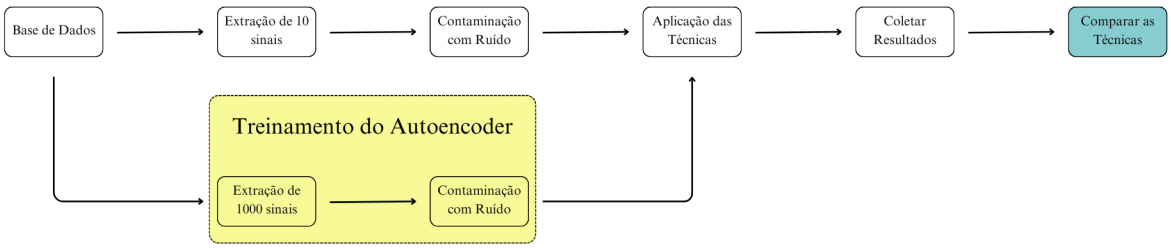
melhoria da qualidade perceptual (GARG, 2023). Narayanan e Wang propõem o uso de redes neurais para estimar a máscara de razão ideal, isolando o sinal de fala do ruído, com resultados que indicam uma alta precisão na estimativa (NARAYANAN; WANG, 2013). Por fim, Cui et al. investigam o impacto da augmentation de dados, apresentando técnicas que aumentam a robustez do modelo, especialmente em condições adversas (CUI; GOEL; KINGSBURY, 2015).

O presente trabalho se assemelha aos trabalhos revisados ao investigar a eficácia de diferentes técnicas para a remoção de ruído em sinais de fala. Assim como nos artigos mencionados, este trabalho visa avaliar o desempenho de diferentes abordagens, utilizando métricas objetivas para quantificar a qualidade e a inteligibilidade da fala aprimorada.

3 Metodologia

Este capítulo apresenta a metodologia empregada neste trabalho, detalhando os processos e as decisões fundamentais tomadas ao longo de sua execução, salientando o foco do trabalho. A Figura 3.0.1 ilustra a estrutura das escolhas metodológicas adotadas.

Figura 3.0.1 – Diagrama do processo metodológico



Fonte: Elaborado pelo autor.

Inicialmente, foi selecionada uma base de dados contendo os áudios utilizados no estudo. Dessa base, foram escolhidos dez sinais de voz para servirem como referência na comparação entre as técnicas de remoção de ruído. Esses sinais foram contaminados com ruído Gaussiano, de modo que tanto o filtro de Wiener quanto o Autoencoder fossem aplicados à mesma tarefa e aos mesmos sinais contaminados. Após a remoção do ruído, as duas técnicas foram avaliadas com base na Distância Euclidiana entre os sinais processados e seus respectivos sinais originais. Para o treinamento do Autoencoder, foram selecionados 1000 sinais de voz adicionais da base. Esses sinais também foram contaminados com ruído gaussiano para compor os conjuntos de treinamento e teste, com o objetivo de aumentar a capacidade de generalização do modelo.

A análise comparativa dos resultados obtidos com cada técnica visa avaliar as vantagens e limitações das abordagens, contribuindo para uma discussão fundamentada sobre o uso de métodos clássicos e modernos para remoção de ruído em sinais de voz.

Para comparar a reconstrução dos sinais realizadas pelas técnicas, Filtro de Wiener e Autoencoder, as distâncias Euclidianas dos sinais serão colocadas frente a frente, além do desvio padrão dos valores das distâncias Euclidianas, para entender a consistência dos valores gerados. Ademais, foi comparado o desempenho computacional de cada técnica, que avaliou o comportamento individual de cada algoritmo enquanto executa uma técnica.

3.1 Definição da Base de Dados

Para a realização deste trabalho, é essencial utilizar uma base de sinais de voz robusta, especialmente considerando o treinamento de um modelo de DL como o Autoencoder. Assim, foi selecionado o dataset *LibriSpeech*, uma base originalmente desenvolvida para tarefas de reconhecimento de fala (*Speech Recognition*), mas que oferece características adequadas para o presente estudo.

Dentre essas características, destacam-se os áudios de alta qualidade, com taxa de amostragem de 16 kHz, o que garante a preservação dos detalhes essenciais do sinal de voz. Além disso, os áudios são originalmente limpos, sem ruídos externos, proporcionando uma referência confiável para a contaminação artificial com ruídos e a análise comparativa de técnicas de remoção. Outro diferencial do dataset é a diversidade de locutores, abrangendo diferentes gêneros, idades e sotaques, o que contribui para a generalização dos modelos treinados, ampliando sua aplicabilidade em diversos contextos.

O LibriSpeech é uma coleção de aproximadamente 1.000 horas de *audiobooks* lidos em língua inglesa que fazem parte do projeto LibriVox, com a maioria das gravações provenientes do Project Gutenberg. A duração consistente e predominantemente curta dos áudios, geralmente 5 segundos, facilita o processamento, o treinamento e permite uma análise segmentada do desempenho das técnicas, minimizando o risco de sobreajuste e garantindo maior confiabilidade nos resultados obtidos.

3.2 Introdução de Ruído

Para avaliar a eficácia das técnicas de remoção de ruído, os sinais de voz selecionados serão intencionalmente contaminados com ruído gaussiano. Esse procedimento permite testar a capacidade do filtro de Wiener e do Autoencoder em recuperar a qualidade dos sinais originais ao remover o ruído introduzido, fornecendo assim uma base comparativa objetiva entre as duas abordagens.

3.2.1 Ruído Gaussiano

Como descrito no capítulo 2, o ruído introduzido nos sinais de voz foi o Ruído Gaussiano, dado a natureza aleatória que existe em sua distribuição. Ele foi introduzido nos 10 sinais avaliados por este trabalho e nos 1.000 sinais de treinamento do Autoencoder.

O processo de adição de ruído baseia-se na geração de valores aleatórios seguindo uma distribuição normal com média zero e desvio padrão unitário. O ruído gerado foi então escalonado por meio de um fator multiplicador, configurado para 0.05 por padrão. Por fim, o ruído escalonado foi somado ponto a ponto ao áudio original, produzindo um novo sinal de

áudio contaminado, mas ainda preservando características do sinal original. Esse áudio ruidoso foi utilizado para avaliar e comparar o desempenho do Filtro de Wiener e do Autoencoder.

3.3 Métricas de Desempenho

3.3.1 Distância Euclidiana

Conforme detalhado no capítulo 2, a métrica de Distância Euclidiana será adotada para avaliar a recuperação dos sinais processados pelo Filtro de Wiener e Autoencoder. A comparação será realizada entre o sinal original e o sinal recuperado por cada modelo, possibilitando uma análise direta da fidelidade de recuperação. Cada modelo será aplicado em 10 sinais de voz distintos, totalizando o cálculo da Distância Euclidiana para 20 sinais de áudio, permitindo uma avaliação consistente do desempenho de cada abordagem. Além disso, no algoritmo do Filtro de Wiener, a Distância Euclidiana será uasada para encontrar o valor ótimo da janela para o filtro.

3.3.2 Desempenho Computacional

Para comparar o desempenho de cada técnica, serão avaliados dois fatores principais: o **tempo de execução empírico** de cada algoritmo e o **uso de recursos do sistema**. O tempo de execução será calculado como a média aritmética dos tempos obtidos em cinco execuções consecutivas de cada técnica, a fim de reduzir a influência de eventuais variações de desempenho. Já o uso de recursos do sistema será avaliado considerando o desempenho da máquina enquanto executa cada algoritmo avaliando o uso de CPU e memória, fornecendo uma visão sobre o impacto de cada técnica no desempenho geral da máquina.

Para assegurar uma comparação justa entre o Filtro de Wiener e o Autoencoder, todas as medições serão realizadas sob as mesmas condições e na mesma máquina, garantindo consistência nos resultados e minimizando interferências externas.

3.4 Implementação das Técnicas

3.4.1 Implementação do Filtro de Wiener no *scipy*

Serão apresentados primeiramente os parâmetros de entrada. O parâmetro **im** é o sinal que será filtrado. Ele é passado como um array N-dimensional, permitindo que o código funcione com dados de diferentes dimensões, como imagens em 2D ou até volumes em 3D (VIRTANEN et al., 2020).O segundo parâmetro é o **mysize** que define o tamanho da janela do filtro em cada dimensão. Se mysize for um número único, esse valor será usado para todas as dimensões; se for uma lista, deve ter um valor para cada dimensão. A janela define a área em torno de cada pixel onde serão calculadas a média e a variância (VIRTANEN et al., 2020).

A seguir será descrito o passo a passo realizado pelo algoritmo do Filtro de Wiener:

1. Conversão do Sinal para Array:

   O código garante que o sinal é convertido para um array ndarray do NumPy, caso ainda não seja. Isso permite que operações numéricas sejam realizadas de forma mais eficiente.

2. Definição do Tamanho da Janela do Filtro:

   Se o tamanho da janela não for especificado, ele é definido como 3 em todas as dimensões (uma janela 3x3 para imagens 2D, por exemplo). Se mysize for um valor único, o código o expande para todas as dimensões.

3. Cálculo da Média Local:

   Nesta etapa, o código calcula a média local para cada ponto do sinal. Isso é feito aplicando uma operação de correlação com uma janela de 1s e dividindo pelo tamanho total da janela (size). A média local ajuda a suavizar as variações do sinal e identifica a intensidade média de cada região.

4. Cálculo da Variância Local:

   Para obter a variância local, o código calcula a média dos quadrados dos valores dentro da janela e subtrai o quadrado da média local (Média Local). Esse cálculo é importante para entender a variabilidade de cada região: uma variância alta indica mais ruído, enquanto uma variância baixa indica uma região mais homogênea.

5. Estimativa da Potência do Ruído:

   O valor da Potência é estimado como a média da Variância Local, assumindo que essa variância média representa bem o ruído presente no sinal.

6. Aplicação do Filtro de Wiener:

   Esta é a etapa central do filtro de Wiener. A operação (Sinal Ruidoso - Média Local) desloca os valores do sinal em torno da média local. Em seguida, multiplicamos pelo fator (1 - Potência do Ruído / Variância Local), o que reduz a intensidade do sinal nas áreas onde o ruído é comparativamente alto. Adicionamos de volta a média local para retornar o valor ajustado ao seu intervalo original.

   Para áreas onde a Variância Local é menor que o ruído, o código define o valor filtrado como a Média Local, já que nessas áreas a informação relevante é pequena e o ruído pode ser dominante.

3.4.2 Implementação do Autoencoder no *Tensorflow*

Para construir o modelo de Autoencoder será utilizada uma API de construção e treinamento de redes neurais do *Tensorflow*, o *Keras*. Para criar o modelo de forma simples

e direta, adicionando uma camada por vez, em sequência, será utilizada a classe *Sequential* do *Keras*. Para a construção do modelo serão usadas funções da classe *Sequential*. A função **add** adiciona uma camada à pilha de camadas do modelo sequencial, permitindo construir as redes neurais do modelo (ABADI et al., 2016). Além disso, a função *compile* prepara um modelo para o treinamento, definindo o otimizador, a função de perda e as métricas que serão usadas para avaliar o desempenho. Essa configuração é essencial para que o modelo saiba como ajustar seus parâmetros para minimizar o erro durante o treinamento (ABADI et al., 2016).

A seguir será descrito o passo a passo realizado pelo algoritmo do Autoencoder:

1. Definição da Camada de Entrada:

   A primeira camada adicionada é a *InputLayer*, que define a forma dos dados de entrada. Esta camada prepara o modelo para receber entradas de um tamanho específico, definindo quantos neurônios a entrada possui.

2. Adição de Camadas Ocultas:

   Três camadas densas (ou totalmente conectadas) são adicionadas em sequência. A primeira camada possui 256 neurônios e usa a função de ativação ReLU, que adiciona não-linearidade. Esta camada começa a extrair características relevantes da entrada.

   A segunda camada possui 128 neurônios e também usa ReLU, criando uma representação mais compacta dos dados. Esta é a camada que efetivamente "comprime"os dados.

   A terceira camada tem 256 neurônios e usa ReLU, expandindo a informação de volta para preparar a reconstrução na camada de saída.

3. Camada de Saída:

   A última camada do modelo também é densa e possui o mesmo número de neurônios que a entrada. Ela usa a função de ativação sigmoid, que gera valores entre 0 e 1. Esta camada tenta reconstruir os dados originais, ajustando cada neurônio de saída para ser o mais próximo possível dos valores de entrada.

4. Compilação do Modelo:

   O modelo é então compilado com o otimizador adam, que ajusta os pesos durante o treinamento de maneira adaptativa e eficiente. A função de perda usada é o MSE, que mede o erro entre a reconstrução e os dados originais. Quanto menor o mse, melhor o modelo está em reconstruir os dados.

5. Retorno do Modelo:

   Por fim, o modelo é retornado. Uma vez construído, ele pode ser treinado em dados, com os sinais de áudio com e sem ruído, para que aprenda a reconstruir versões mais limpas dos dados de entrada.

O Autoencoder possui uma arquitetura simétrica com um codificador, um ponto de estrangulamento *bottleneck* e um decodificador. A entrada do modelo aceita vetores de áudio contaminados com ruído, definidos pelo tamanho especificado em *input_shape*. O codificador reduz progressivamente a dimensionalidade dos dados, começando com uma camada oculta de 256 neurônios, seguida por uma de 128 e outra de 64, todas utilizando a função de ativação ReLU para extrair características relevantes e comprimir a representação dos dados no *bottleneck*.

A partir dessa representação comprimida, o decodificador expande novamente os dados em direção à sua dimensionalidade original, utilizando camadas ocultas de 128 e 256 neurônios, também com ativação *ReLU*. Por fim, a camada de saída possui tantos neurônios quanto a entrada e utiliza a função de ativação *linear* permitindo que o modelo produza saídas contínuas que correspondem diretamente às entradas reconstruídas, útil quando trabalhamos com dados numéricos, como sinais de áudio.

O modelo é compilado com o otimizador *Adam*, que ajusta os pesos para minimizar o erro quadrático médio (MSE), que mede a diferença entre os áudios reconstruídos e os originais. Durante o treinamento, os áudios contaminados são utilizados como entrada e os áudios limpos como rótulos, com 10% dos dados reservados para validação. O treinamento ocorre em 50 épocas, com lotes de 16 exemplos, permitindo que o autoencoder aprenda a reconstruir áudios sem ruído de suas versões contaminadas.

4 Resultados

Este capítulo apresenta os testes realizados e os resultados obtidos para avaliar o desempenho das técnicas de remoção de ruído aplicadas a sinais de voz, com foco na comparação entre o Filtro de Wiener e o Autoencoder. Os experimentos foram conduzidos em um ambiente Python, utilizando bibliotecas como *scipy* e *tensorflow*. A primeira é uma biblioteca que oferece uma ampla coleção de algoritmos e funções matemáticas, incluindo o filtro de Wiener, enquanto que a segunda facilita a criação, treinamento e implantação de redes neurais e outros modelos de machine learning, assim como o Autoencoder.

Com base nos resultados, é apresentada uma análise comparativa entre as duas técnicas, avaliando o impacto de cada método sobre a qualidade dos sinais de voz processados e o custo computacional envolvido. Finalmente, o capítulo discute as limitações de cada método e suas potenciais aplicações em cenários práticos, destacando as contribuições deste estudo para o campo de processamento de sinais de voz.

4.1 Filtro de Wiener

4.1.1 Resultados do Filtro de Wiener

Inicialmente, os dez sinais de voz usados para avaliação foram carregados e processados pelo Filtro de Wiener. Cada sinal foi ajustado para uma duração de 5 segundos e normalizado, de forma que os valores de amplitude estivessem no intervalo de -1 a 1. Em seguida, um ruído Gaussiano foi adicionado a todos os sinais para simular contaminação. Para cada um dos dez sinais de voz, foi calculado o valor ótimo do tamanho do estimador do filtro, visando minimizar a Distância Euclidiana entre o sinal filtrado e o sinal original. Com o melhor valor do tamanho do estimador identificado, o Filtro de Wiener foi aplicado a cada um dos sinais contaminados.

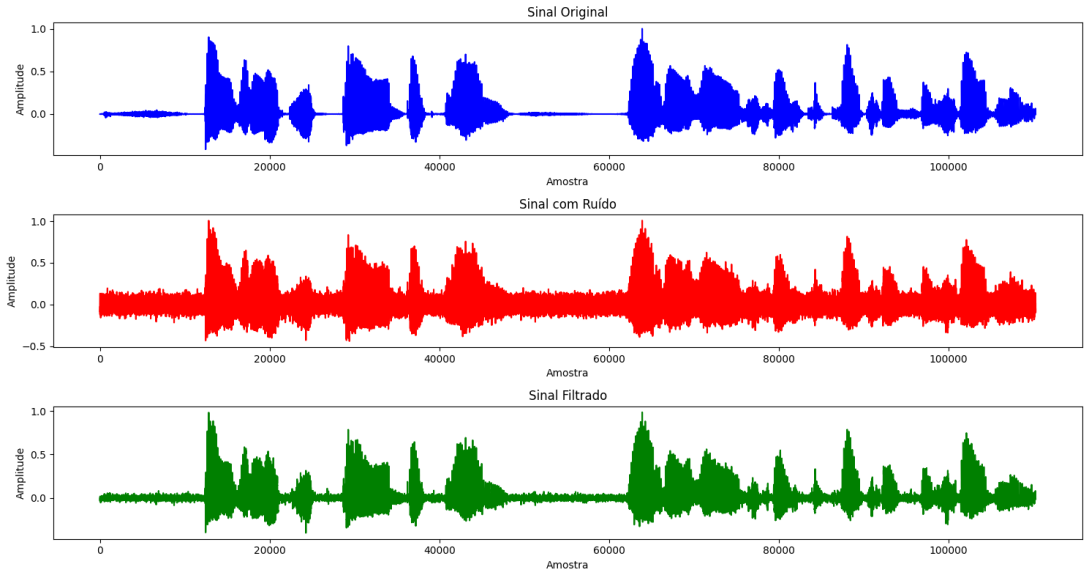
Tabela 1 – Distâncias Euclidianas e tamanhos ótimos do estimador obtidos com o Filtro de Wiener

Sinal	Distância	Tamanho do Estimador
Sinal 1	10,449	7
Sinal 2	9,036	9
Sinal 3	9,076	7
Sinal 4	9,029	7
Sinal 5	9,027	7
Sinal 6	9,507	5
Sinal 7	8,351	9
Sinal 8	9,958	5
Sinal 9	8,057	11
Sinal 10	8,140	9

Fonte: Elaborado pelo autor.

A Tabela 1 apresenta as distâncias euclidianas obtidas entre os sinais filtrados e seus respectivos sinais originais, além dos tamanhos ótimos do estimador calculados para cada áudio. Observa-se que a Distância Euclidiana se manteve relativamente próxima entre os diferentes sinais, uma vez que o desvio padrão para essas medidas foi de 0,770. Além disso, a média geral dos valores apresentados foi de 9,063. A Figura 4.1.1 ilustra a distribuição de frequência do sinal de voz filtrado com a menor Distância Euclidiana, indicando uma alta similaridade entre o sinal filtrado e o sinal original, especialmente em termos de sua estrutura espectral.

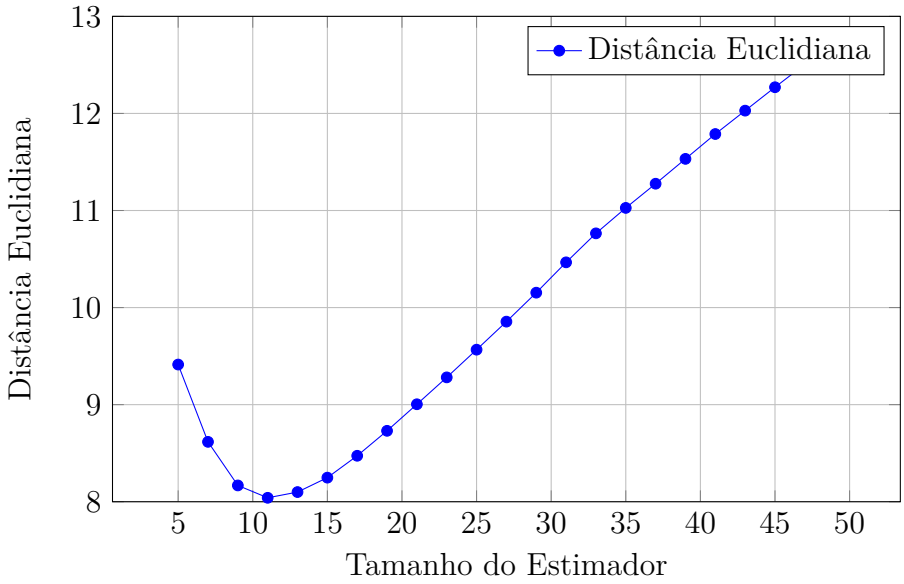
Figura 4.1.1 – Distribuição de frequência do sinal de voz filtrado com menor Distância Euclidiana



Fonte: Elaborado pelo autor.

Os resultados observados sugerem que o Filtro de Wiener foi eficaz em reduzir o ruído presente nos sinais de voz, aproximando-os de suas formas originais, como evidenciado pelas distâncias euclidianas relativamente baixas e consistentes entre os sinais processados. O valor médio das distâncias indica uma preservação satisfatória das características dos sinais originais, enquanto o uso de tamanhos ótimos do estimador contribuiu para otimizar a resposta do filtro em cada áudio individualmente. A figura 4.1.2 mostra a otimização do tamanho do estimador para cada sinal processado.

Figura 4.1.2 – Gráfico mostrando a relação entre o tamanho do estimador e a distância euclidiana.



Fonte: Elaborado pelo autor.

4.2 Autoencoder

Nos testes com o Autoencoder, foi seguido um processo similar ao do Filtro de Wiener. Os sinais foram carregados, ajustados para 5 segundos, normalizados e contaminados com ruído Gaussiano. No entanto, para o treinamento do Autoencoder, foram utilizados 1000 sinais de voz adicionais, reservando os dez sinais para a etapa de validação. O Autoencoder projetado para esta tarefa consistiu em uma rede neural com três camadas ocultas, utilizando a função de ativação ReLU. O treinamento consistiu em apresentar ao modelo os sinais originais como entrada e os sinais contaminados como saída. Após o treinamento, o modelo Autoencoder foi aplicado aos dez sinais de validação contaminados.

Durante o treinamento, o modelo foi apresentado aos sinais ruidosos como entrada e aos sinais originais como saída, buscando assim aprender a reconstrução do sinal limpo a partir da versão contaminada. Após o processo de treinamento, o Autoencoder foi então aplicado aos dez sinais de validação contaminados para verificar seu desempenho na remoção de ruído.

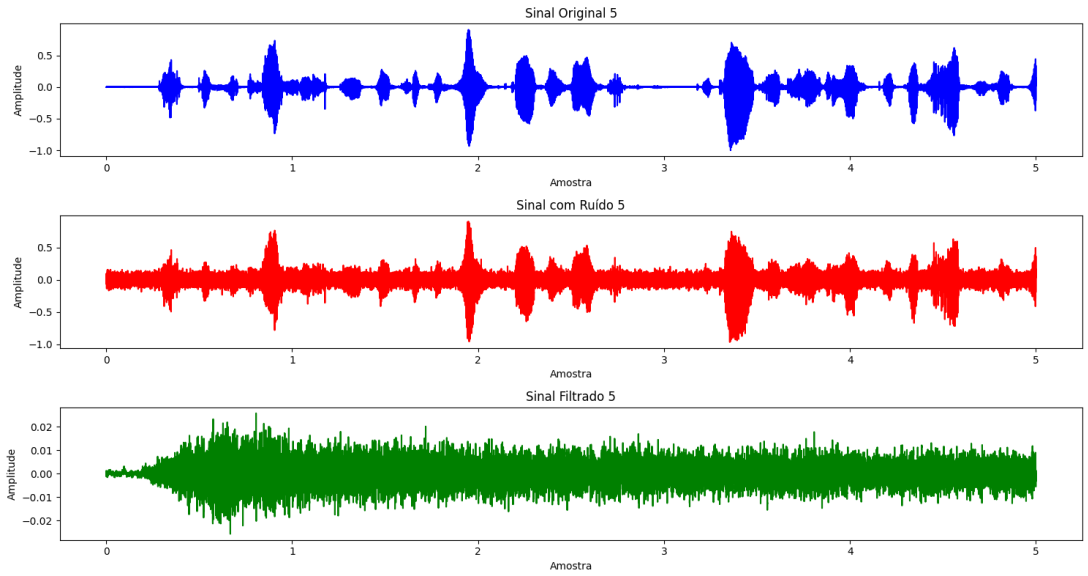
Tabela 2 – Distâncias Euclidianas de cada sinal processado pelo Autoencoder

Sinal	Distância Euclidiana
Sinal 1	47,313
Sinal 2	32,530
Sinal 3	28,551
Sinal 4	37,932
Sinal 5	24,925
Sinal 6	50,467
Sinal 7	47,219
Sinal 8	29,147
Sinal 9	49,758
Sinal 10	29,932

Fonte: Elaborado pelo autor.

A Tabela 2 apresenta as distâncias Euclidianas entre os sinais processados pelo Autoencoder e seus respectivos sinais originais. O desvio padrão dos valores apresentados na tabela é igual a 9,990 e a média equivale a 37,775. Já a Figura 4.2.1 demonstra que o sinal processado pelo Autoencoder foi completamente distorcido, indicando limitações no modelo em capturar e reproduzir adequadamente as nuances do sinal original em ambientes de ruído elevado.

Figura 4.2.1 – Distribuição de frequência do sinal de voz filtrado com menor Distância Euclidiana



Fonte: Elaborado pelo autor.

4.3 Comparação dos Resultados

Ao fim da realização dos testes, os resultados das diferentes técnicas serão comparados com base nas distâncias euclidianas entre os sinais processados e os sinais originais, além do desvio padrão entre as distâncias euclidianas e das métricas de desempenho citadas no capítulo 3.

Como as distâncias euclidianas foram calculadas para cada um dos sinais, a comparação será direta entre cada sinal, analisando as duas técnicas, além das médias das medidas de Distância Euclidiana. Já o desvio padrão foi calculado em cima das distâncias de todos os sinais. Por fim, os testes de desempenho consistiram na medição de tempo e uso da CPU durante 5 execuções seguidas de cada técnica.

A análise inclui tanto a precisão da remoção de ruído quanto a eficiência dos métodos, fornecendo uma visão detalhada sobre as vantagens e limitações de cada abordagem no contexto da filtragem de sinais de voz. A partir desses resultados, será possível entender qual técnica proporciona uma maior preservação do sinal original, atendendo ao objetivo de minimizar a distorção enquanto otimiza os recursos computacionais.

4.3.1 Comparando a Distância Euclidiana

Os valores da Distância Euclidiana podem ser observados na tabela 3, e evidentemente o filtro de Wiener apresentou desempenho muito mais satisfatório. A média das medidas das distâncias evidencia a diferença de desempenho entre as técnicas, como visto na tabela 4.

Tabela 3 – Tabela de Distâncias Euclidianas: Filtro de Wiener e Autoencoder

Sinal	Métricas Filtro de Wiener	Métricas Autoencoder
Sinal 1	10,497	47,313
Sinal 2	9,024	32,530
Sinal 3	9,046	28,552
Sinal 4	8,985	37,932
Sinal 5	9,025	24,925
Sinal 6	9,434	50,467
Sinal 7	8,385	47,198
Sinal 8	9,959	29,147
Sinal 9	8,022	49,759
Sinal 10	8,172	29,932

Fonte: Elaborado pelo autor.

Tabela 4 – Tabela de Médias de Distância Euclidiana em cada técnica

Técnicas	Médias de Distância Euclidiana
Filtro de Wiener	9,055
Autoencoder	37,775

Fonte: Elaborado pelo autor.

4.3.1.1 Diferença Absoluta entre as Distâncias

Além disso, a diferença absoluta de cada técnica pode ser observada na tabela 5, e até mesmo a menor das diferenças se mostrou um valor consideravelmente alto, o que indica que o filtro de Wiener realmente apresentou um desempenho superior em relação ao Autoencoder.

Tabela 5 – Tabela de Diferença Absoluta das Distâncias Euclidianas

Sinais	Diferença Absoluta entre as Medidas
Sinal 1	36,816
Sinal 2	23,506
Sinal 3	19,506
Sinal 4	28,947
Sinal 5	15,900
Sinal 6	41,033
Sinal 7	38,813
Sinal 8	19,188
Sinal 9	41,737
Sinal 10	21,760

Fonte: Elaborado pelo autor.

4.3.1.2 Comparando o Desvio Padrão das Medidas das Distâncias

Em relação à consistência das medidas, o desvio padrão das medidas de Distância Euclidiana foi bem menor ao utilizar o filtro de Wiener. A tabela 6 mostra que o Autoencoder apresentou um desempenho muito inconsistente.

Tabela 6 – Tabela de Desvio Padrão das Distâncias Euclidianas em cada técnica

Técnicas	Desvio Padrão
Filtro de Wiener	0,77
Autoencoder	9,99

Fonte: Elaborado pelo autor.

4.3.2 Comparando o Desempenho Computacional

Como elucidado anteriormente, os testes de desempenho computacional foram executados nas mesmas condições, sequencialmente e na mesma máquina.

4.3.2.1 Configuração da Máquina

Os testes foram realizados em um notebook com a configuração a seguir:

- Processador: Intel Core i5 12450H
- Placa mãe: LENOVO LNVNB 161216
- RAM: 8GB DDR5 5600MHz

4.3.2.2 Testes Filtro de Wiener

Os testes de desempenho do filtro de Wiener evidenciam quão bem otimizada a técnica é. A tabela 7 apresenta a consistência dos tempos medidos, com uma média de 5,639 segundos para a execução do algoritmo.

Tabela 7 – Tabela de Tempos de Execução Filtro de Wiener

Teste	Tempo de Execução
1	5,627s
2	5,628s
3	5,666s
4	5,699s
5	5,575s

Fonte: Elaborado pelo autor.

Quanto ao uso de CPU, ele se manteve estável durante todas as execuções, variando entre 30% e 40%.

4.3.2.3 Testes Autoencoder

O Autoencoder, por ser uma técnica de Deep Learning (DL), obviamente requer mais recursos do computador, além de apresentar tempos de execução mais altos, como observado na tabela 8.

Tabela 8 – Tabela de Tempos de Execução Autoencoder

Teste	Tempo de Execução
1	11 minutos e 36 segundos
2	11 minutos e 34 segundos
3	11 minutos e 24 segundos
4	11 minutos e 26 segundos
5	11 minutos e 50 segundos

Fonte: Elaborado pelo autor.

Apesar de seus tempos altos em comparação com o filtro de Wiener, os tempos foram consistentes, apresentando uma média de 11 minutos e 34 segundos.

A CPU foi utilizada em sua totalidade; em todas as execuções, o uso da CPU se manteve em 100%.

5 Conclusão

O presente trabalho teve como objetivo comparar duas abordagens de supressão de ruído: uma técnica clássica baseada em fundamentos estatísticos, representada pelo Filtro de Wiener, e uma abordagem de DL, o Autoencoder.

Os resultados obtidos indicaram que o Filtro de Wiener apresentou um desempenho consistente nas condições analisadas, demonstrando-se eficaz na remoção de ruído com um custo computacional relativamente baixo. Por outro lado, embora o Autoencoder, como técnica baseada em DL, possua potencial para alcançar resultados superiores em cenários mais complexos, o desempenho observado neste trabalho ficou aquém do esperado. Este resultado sugere que fatores como o tamanho do conjunto de dados e a complexidade do modelo podem ter limitado sua eficácia. Além disso, o custo computacional mais elevado associado ao treinamento e à implementação de modelos avançados de DL representa um desafio adicional, especialmente em cenários práticos com recursos computacionais limitados.

Cabe destacar que o desempenho inferior do Autoencoder neste estudo não invalida seu potencial, mas reforça a importância de ajustar cuidadosamente os parâmetros do modelo, ampliar o conjunto de dados e explorar diferentes configurações de arquitetura para maximizar sua eficácia. A aplicação de modelos baseados em DL exige não apenas maior investimento em recursos, mas também um planejamento criterioso para que seu potencial possa ser plenamente explorado.

Nos últimos anos, o avanço da Inteligência Artificial (IA) tem despertado grande interesse, sendo frequentemente associado a promessas de benefícios significativos para a sociedade. No entanto, os resultados deste trabalho ressaltam as complexidades e os desafios inerentes ao desenvolvimento e à implementação dessas técnicas. Além disso, vozes críticas, como a do Professor Geoffrey Hinton, um dos pioneiros da IA, alertam para os riscos potenciais associados ao uso inadequado ou irresponsável dessas tecnologias, destacando a necessidade de uma abordagem ética e responsável no desenvolvimento de aplicações baseadas em IA.

Portanto, este estudo contribui para a reflexão sobre as vantagens e limitações das abordagens clássicas e modernas em tarefas específicas, enfatizando a necessidade de avaliar cuidadosamente os recursos e requisitos antes de optar por uma ou outra técnica.

5.1 Trabalhos Futuros

Com base nos resultados e limitações identificados neste estudo, diversas direções para trabalhos futuros podem ser exploradas com o intuito de aprimorar as abordagens de supressão de ruído e expandir o conhecimento na área.

Uma linha promissora de pesquisa envolve a variação nos níveis de ruído adicionados aos sinais de áudio durante os experimentos. Como explicitado no capítulo **??**, o Ruído Gaussiano adicionado possui intensidade de 5%. Variar essa intensidade pode fornecer uma visão mais abrangente sobre a robustez e as limitações de cada abordagem.

Além disso, uma outra direção importante é a modificação da arquitetura do Autoencoder utilizado. Ajustes nos hiperparâmetros podem ser feitos afim de maximizar o desempenho do modelo, como alterar as funções de ativação e otimização, o tamanho dos lotes, a taxa de aprendizado e o número de épocas.

Por fim, a combinação de métodos clássicos e baseados em aprendizado profundo pode ser uma abordagem interessante para trabalhos futuros. Por exemplo, o Filtro de Wiener pode ser usado como uma etapa inicial de pré-processamento para reduzir ruídos grosseiros, seguido pela aplicação do Autoencoder para refinar os sinais. Essa integração pode alavancar os pontos fortes de ambas as técnicas e oferecer uma solução mais robusta.

Essas investigações poderão não apenas ampliar a aplicabilidade das técnicas de SE, mas também fornecer insights valiosos para superar os desafios associados ao uso de aprendizado profundo em tarefas específicas, contribuindo para o avanço da área.

Referências

ABADI, M. et al. {TensorFlow}: a system for {Large-Scale} machine learning. In: *12th USENIX symposium on operating systems design and implementation (OSDI 16)*. [S.l.: s.n.], 2016. p. 265–283.

BENESTY, J. et al. *Noise reduction in speech processing*. [S.l.]: Springer Science & Business Media, 2009. v. 2.

BISHOP, C. M.; BISHOP, H. *Deep learning: Foundations and concepts*. [S.l.]: Springer Nature, 2023.

BORGSTRÖM, B. J.; BRANDSTEIN, M. S. A multiscale autoencoder (msae) framework for end-to-end neural network speech enhancement. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, IEEE, 2024.

CHEN, J. et al. New insights into the noise reduction wiener filter. *IEEE Transactions on audio, speech, and language processing*, IEEE, v. 14, n. 4, p. 1218–1234, 2006.

CHHETRI, S. et al. Speech enhancement: A survey of approaches and applications. In: IEEE. *2023 2nd International Conference on Edge Computing and Applications (ICECAA)*. [S.l.], 2023. p. 848–856.

CHUNG, Y.-A. et al. An unsupervised autoregressive model for speech representation learning. *arXiv preprint arXiv:1904.03240*, 2019.

CUI, X.; GOEL, V.; KINGSBURY, B. Data augmentation for deep neural network acoustic modeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, IEEE, v. 23, n. 9, p. 1469–1477, 2015.

GARG, A. Speech enhancement using long short term memory with trained speech features and adaptive wiener filter. *Multimedia tools and applications*, Springer, v. 82, n. 3, p. 3647–3675, 2023.

KAY, S. *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*. [S.l.]: Pearson, 1993.

MICHELUCCI, U. An introduction to autoencoders. *arXiv preprint arXiv:2201.03898*, 2022.

NARAYANAN, A.; WANG, D. Ideal ratio mask estimation using deep neural networks for robust speech recognition. In: IEEE. *2013 IEEE international conference on acoustics, speech and signal processing*. [S.l.], 2013. p. 7092–7096.

OPPENHEIM, A. V.; VERGHESE, G. C. *Signals, systems & inference*. [S.l.]: Pearson London, 2017.

OTUNG, I. *Communication engineering principles*. [S.l.]: John Wiley & Sons, 2021.

O'SHAUGHNESSY, D. Review of methods for coding of speech signals. *EURASIP Journal on Audio, Speech, and Music Processing*, Springer, v. 2023, n. 1, p. 8, 2023.

RIBAS, D. et al. Wiener filter and deep neural networks: A well-balanced pair for speech enhancement. *Applied Sciences*, MDPI, v. 12, n. 18, p. 9000, 2022.

ROUX, J. L.; VINCENT, E. Consistent wiener filtering for audio source separation. *IEEE signal processing letters*, IEEE, v. 20, n. 3, p. 217–220, 2012.

TSCHANNEN, M.; BACHEM, O.; LUCIC, M. Recent advances in autoencoder-based representation learning. *arXiv preprint arXiv:1812.05069*, 2018.

VINCENT, P. et al. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, v. 11, n. 12, 2010.

VIRTANEN, P. et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, Nature Publishing Group, v. 17, n. 3, p. 261–272, 2020.