Universidade de São Paulo
Instituto de Matemática e Estatística
Bacharelado em Ciência da Computação

Estratégias para Lidar com Ruído de Áudio em Aprendizado Automático para Insuficiência Respiratória: Uma análise da Filtragem

Luan Tavares de Andrade

Monografia Final

mac 499 — Trabalho de Formatura Supervisionado

Supervisor: Prof. Dr. Marcelo Finger

São Paulo
2025

Agradecimentos

No one knows what the future holds. That's why its potential is infinite.

— Okabe Rintarou

Agradeço à minha família e aos meus amigos pelo apoio, compreensão e incentivo ao longo de toda a graduação. Agradeço também ao meu orientador pela oportunidade e orientação durante o desenvolvimento deste trabalho.

Resumo

Luan Tavares de Andrade. **Estratégias para Lidar com Ruído de Áudio em Aprendizado Automático para Insuficiência Respiratória: Uma análise da Filtragem**. Monografia (Bacharelado). Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2025.

Este trabalho tem como objetivo investigar a eficácia da filtragem de ruído como estratégia de pré-processamento para mitigar o viés ambiental em modelos de rede neural para detecção de insuficiência respiratória por análise de áudio. Para isso foi utilizado o *dataset* do Projeto SPIRA, caracterizado por uma forte correlação entre ruído hospitalar e a classe positiva, além de duas arquiteturas de Redes Neurais pré-treinadas: a CNN10 (Rede Neural Convolucional) e o AudioMAE (Transformer), de forma que fosse possível a comparação do desempenho de dois modelos com complexidades diferentes. A metodologia consistiu em duas etapas experimentais: (1) uma análise de sensibilidade para determinar a estabilidade dos parâmetros do filtro, que revelou uma tendência ao overfitting de artefatos de filtragem; e (2) um teste de viés comparando os modelos ao serem testados com os áudios filtrados e com inserção de ruído antes do processo do filtragem. Como resultado, com a inserção de ruído prévia, ambos os modelos apresentaram um colapso das métricas avaliadas, evidenciando que o aprendizado baseou-se em atalhos (*shortcut learning*) associados ao ruído residual e aos fragmentos de filtragem. Conclui-se que a filtragem de ruído é insuficiente para a remoção de vieses ambientais, revelando-se não apenas uma abordagem ineficaz, mas um potencial intensificador do problema ao introduzir novos vieses.

Palavras-chave: Insuficiência Respiratória. Aprendizado Profundo. Processamento de Áudio. Filtragem de Ruído. Viés Ambiental. Rede Neural Convolucional. Transformer.

Abstract

Luan Tavares de Andrade. **Strategies for Handling Audio Noise in Machine Learning for Respiratory Insufficiency: *An Analysis of Filtering***. Capstone Project Report (Bachelor). Institute of Mathematics and Statistics, University of São Paulo, São Paulo, 2025.

This work aims to investigate the efficacy of noise filtering as a pre-processing strategy to mitigate environmental bias in neural network models for respiratory insufficiency detection via audio analysis. To this end, the SPIRA Project dataset was used, characterized by a strong correlation between hospital noise and the positive class, along with two pre-trained Neural Network architectures: CNN10 (Convolutional Neural Network) and AudioMAE (Transformer), allowing for the performance comparison of two models with different complexities. The methodology consisted of two experimental stages: (1) a sensitivity analysis to determine filter parameter stability, which revealed a tendency toward overfitting to filtering artifacts; and (2) a bias test comparing the models when tested with filtered audios versus noise insertion prior to the filtering process. As a result, with prior noise insertion, both models presented a collapse in the evaluated metrics, evidencing that the learning process relied on shortcuts (shortcut learning) associated with residual noise and filtering fragments. It is concluded that noise filtering is insufficient for removing environmental biases, revealing itself not only as an ineffective approach but also as a potential intensifier of the problem by introducing new biases.

Keywords: Respiratory Insufficiency. Deep Learning. Audio Processing. Noise Filtering. Environmental Bias. Convolutional Neural Network. Transformer.

Lista de abreviaturas

CNN	Rede Neural Convolucional (*Convolutional Neural Network*)
ReLU	*Rectified Linear Unit*
MAE	*Masked Autoencoders*
IME	Instituto de Matemática e Estatística
USP	Universidade de São Paulo
PANN	Rede Neural de Áudio Pré-treinada (*Pretrained audio neural network*)

Lista de figuras

Lista de tabelas

Sumário

Capítulo 1

Introdução

1.1 Contextualização

A pandemia da COVID-19, deflagrada pelo vírus SARS-CoV-2 no final de 2019, representou uma das maiores crises sanitárias da história moderna, impondo desafios monumentais aos sistemas de saúde em todo o mundo. A rápida disseminação da doença, cujas complicações mais severas são de natureza respiratória, gerou uma pressão insustentável sobre a infraestrutura hospitalar. A superlotação de prontos-socorros e a escassez de leitos de UTI tornaram-se uma realidade global (Rache *et al.*, 2020), evidenciando a necessidade urgente de soluções inovadoras que pudessem otimizar o fluxo de pacientes e recursos.

Neste cenário adverso, a tecnologia emergiu como uma aliada fundamental, catalisando projetos inovadores como o SPIRA[1] (Finger *et al.*, 2021). A ferramenta foi concebida como um sistema de triagem remota, cuja premissa era empregar modelos de inteligência artificial para analisar características acústicas em áudios da voz, tosse e respiração.

A grande vantagem dessa abordagem era a sua acessibilidade, já que os dados poderiam ser facilmente capturados por dispositivos de uso massivo, como smartphones. Ao fornecer uma análise de risco preliminar, o objetivo era ajudar a evitar a ida desnecessária de pessoas com sintomas leves aos hospitais, aliviando a sobrecarga do sistema e protegendo tanto pacientes quanto profissionais de saúde.

1.2 Motivação e Objetivo

Uma das estratégias notáveis do projeto SPIRA para garantir a robustez de seu modelo foi a inserção artificial de ruído hospitalar nos dados de áudio, forçando o modelo a se tornar invariante às condições ambientais. Contudo, essa abordagem levanta uma questão de pesquisa pertinente: seria a inserção de ruído a melhor forma de mitigar o viés ambiental? Uma estratégia alternativa, e talvez mais intuitiva, seria a de remover o ruído existente por meio de algoritmos de filtragem.

[1] (acrônimo) Sistema de detecção Precoce de Insuficiência Respiratória por meio de análise de Áudio

Essa abordagem alternativa de remoção de ruído busca, em teoria, isolar o sinal de voz em sua forma mais pura, o que poderia facilitar o aprendizado de características biomédicas relevantes. Contudo, essa estratégia não é isenta de riscos. O processo de filtragem pode, paradoxalmente, distorcer o sinal de voz, correndo o risco de remover não apenas o ruído, mas também biomarcadores acústicos sutis.

Diante disso, este trabalho se propõe a investigar a eficácia e os desafios da aplicação de filtros de ruído como método de pré-processamento para a classificação de insuficiência respiratória a partir da voz. Espera-se, com isso, não apenas avaliar comparativamente as duas estratégias, mas também oferecer subsídios práticos e teóricos para o desenvolvimento de sistemas mais confiáveis e robustos em ambientes reais de uso.

Capítulo 2

Fundamentação Teórica

O objetivo deste capítulo é apresentar os conceitos fundamentais de processamento de áudio e aprendizado de máquina que servem como alicerce para este trabalho, oferecendo o embasamento necessário para a compreensão das técnicas e metodologias aplicadas.

2.1 Redes Neurais Convolucionais (CNNs)

Uma Rede Neural Convolucional (CNN) é uma classe de rede neural profunda especializada em tratar dados em grade onde a localidade é relevante, ou seja, a vizinhança de um valor é relevante para a análise dele, como por exemplo na análise de imagens. Para realizar sua tarefa, a arquitetura de uma CNN processa os dados através de uma sequência de camadas especializadas, cujas funções serão detalhadas nas subseções a seguir.

2.1.1 Camada Convolucional

Sendo a principal camada de uma CNN, consiste em procurar por padrões específicos nos dados passados. Isso é feito com a utilização de filtros, que deslizam pela imagem (convolução) e reconhecem sempre que captam seu padrão atribuído.

No contexto de um espectrograma, esses padrões podem ser características acústicas como linhas horizontais (que representam tons estáveis, como possíveis pausas), linhas verticais (como cliques ou o início de uma tosse), ou formas mais complexas que correspondem a formantes da voz.

2.1.2 Função de Ativação - ReLU (Rectified Linear Unit)

As funções de ativação têm o papel de introduzir não-linearidade nas redes neurais. Sem elas, uma sequência de camadas lineares seria matematicamente equivalente a apenas uma única transformação linear, limitando a rede a problemas muito simples. Ao aplicar uma função de ativação, a rede passa a ser capaz de modelar fenômenos não-lineares, como os padrões complexos presentes em sinais de fala e em outros dados do mundo real.

A função de ativação mais utilizada em CNNs é a ReLU (Rectified Linear Unit), definida como:

$$f(x) = max(0, x) \tag{2.1}$$

Essa função mantém valores positivos inalterados e zera os valores negativos. Entre suas principais vantagens estão a simplicidade computacional e a eficiência no treinamento de redes profundas.

Em síntese, a introdução dessa não-linearidade é crucial para que a rede consiga extrair representações mais ricas e discriminativas, transformando os mapas de características em descrições cada vez mais adequadas à tarefa de classificação.

2.1.3 Camada de *Pooling*

Essa camada tem como função principal reduzir as dimensões da entrada, mantendo apenas as informações que atribuir como mais relevantes. Ela pode ser do tipo *Max Pooling*, que pega o valor máximo de uma região específica, atribuindo ele ao novo bloco reduzido ou do tipo *Average Pooling*, que calcula a média da região e atribui esse valor ao novo bloco reduzido.

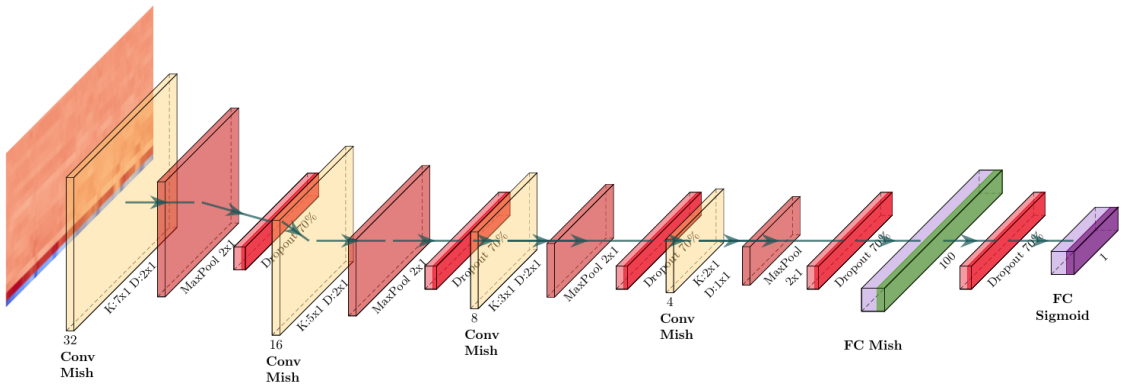
Essa etapa é importante pois ao reduzir as dimensões espaciais do mapa de características, ela diminui drasticamente o número de parâmetros e a carga computacional para as camadas seguintes da rede.

2.1.4 Camada Totalmente Conectada ou Densa

Após as camadas convolucionais e de *pooling*, a representação extraída ainda possui formato bidimensional (semelhante a uma imagem com várias "profundidades" de filtros). Para que essa informação possa ser usada em uma camada densa, é necessário primeiro realizar o achatamento (*flattening*), que transforma os mapas de características em um único vetor unidimensional.

A camada totalmente conectada, como o próprio nome indica, conecta todos os seus neurônios a todos os neurônios da camada anterior. Essa etapa funciona como a parte decisória da rede, combinando as características aprendidas para produzir a saída final. Em tarefas de classificação, por exemplo, essa camada costuma ter número de neurônios igual à quantidade de classes a serem previstas.

Por fim, os valores gerados por essa camada passam por uma função de ativação final apropriada à tarefa: Sigmoid no caso de classificação binária, ou Softmax quando há múltiplas classes. Isso converte os resultados em probabilidades normalizadas, permitindo a interpretação direta como a previsão do modelo para cada classe possível.



Fonte: Casanova, Gris *et al.*, 2021.

Figura 2.1: *Arquitetura da Rede Neural Convolucional* `SpiraConvV1`

2.2 Transformers

Enquanto as CNNs se destacam na extração hierárquica de características locais, uma classe de arquiteturas mais recente, os Transformers (Vaswani *et al.*, 2017), foi proposta para modelar dependências de longo alcance nos dados, ao mesmo tempo que permitia uma melhor paralelização do processo ao tratar dados sequenciais, fator limitante em modelos anteriores como *Recurrent Neural Networks* e *Long Short-Term Memory*.

Originada no campo de Processamento de Linguagem Natural, essa arquitetura revolucionou a área e foi subsequentemente adaptada com grande sucesso para domínios como visão computacional e análise de áudio. Os pontos cruciais desse modelo são explicados a seguir:

2.2.1 O mecanismo de *Self-Attention*

O mecanismo de *Self-Attention*, elemento central dos Transformers, permite que cada *token* de uma sequência estabeleça relações diretas com todos os demais, independentemente da distância entre eles. Dessa forma, o modelo atribui pesos de atenção que indicam o grau de relevância entre pares de *tokens*, capturando dependências globais de maneira eficiente.

Diferentemente das CNNs, que operam com janelas locais e extraem características a partir de vizinhanças fixas, o *Self-Attention* não se limita a regiões próximas da entrada. Isso possibilita que o modelo aprenda relações de longo alcance sem a necessidade de camadas adicionais para ampliar o campo receptivo, além de permitir o processamento totalmente paralelo, resultando em maior eficiência e escalabilidade.

2.2.2 A arquitetura *Encoder-Decoder*

Tendo sido proposta inicialmente em Vaswani *et al.* (2017) para a realização de tarefas de tradução automática, essa arquitetura consiste em duas partes principais, o *Encoder* e o *Decoder*. Juntos, esses módulos permitem tanto a compreensão da entrada quanto a geração de uma saída contextualizada.

Encoder

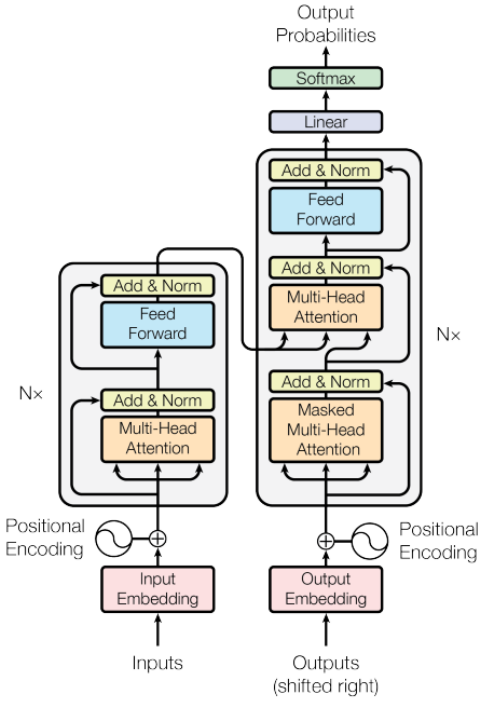
O *Encoder* tem como função principal processar os dados de entrada e extrair representações contextuais significativas. Cada elemento da entrada é transformado em um vetor de embeddings, ao qual é somada uma codificação de posição, necessária para indicar a posição de cada elemento (*token*) na sequência.

Sua estrutura consiste em uma pilha de camadas idênticas, cada uma composta por dois módulos: o *Self-Attention*, que permite a interação entre todos os elementos, e uma Rede Neural *Feed-Forward*, responsável por melhorar as representações obtidas. Entre cada módulo é acrescentada uma etapa de normalização dos dados.

Decoder

O *Decoder* tem como função principal gerar a sequência de saída de forma autoregressiva, isto é, produzindo um elemento por vez com base nos elementos previamente gerados.

Sua estrutura também é composta por uma pilha de camadas, nas quais o primeiro módulo realiza uma operação de *Masked Self-Attention*, que impede o acesso a tokens futuros, garantindo que a geração ocorra de forma autoregressiva. O segundo módulo realiza o *Cross-Attention*, que combina as informações internas do *Decoder* com as representações codificadas pelo *Encoder*, permitindo que a saída seja produzida de forma coerente com o contexto da entrada. Por fim, uma camada *Feed-Forward* refina o resultado, que é então projetado por uma camada linear seguida de uma função Softmax para gerar a distribuição de probabilidade sobre o vocabulário de saída e selecionar o próximo elemento da sequência.



Fonte: Vaswani *et al.*, 2017.

Figura 2.2: *Arquitetura proposta do Transformer*

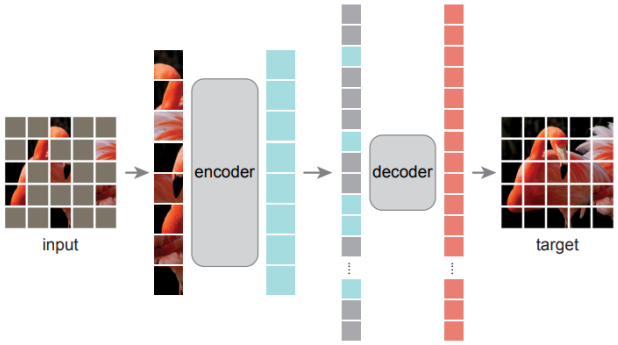
2.2.3 Masked Auto-Encoders

A eficácia das arquiteturas Transformer está diretamente ligada à sua capacidade de processar sequências longas e capturar relações contextuais. Contudo, treinar modelos muito grandes e complexos do zero exige vastos conjuntos de dados rotulados, que são caros e difíceis de obter. Para superar essa limitação, ao longo dos anos vêm sendo popularizada cada vez mais a utilização do Aprendizado Auto-Supervisionado, um paradigma de treinamento onde o modelo aprende representações úteis a partir dos próprios dados, sem a necessidade de rótulos explícitos.

Dentro do Aprendizado Auto-Supervisionado, uma das estratégias mais eficientes e bem-sucedidas para Transformers é a de Masked Auto-encoders (MAE) (He *et al.*, 2021). A ideia central é fazer com que o modelo, em vez de classificar uma imagem ou som, reconstrua partes da entrada que foram deliberadamente escondidas (mascaradas). Esse processo força o modelo a desenvolver uma compreensão profunda da estrutura intrínseca dos dados para ser capaz de "imaginar"o conteúdo ausente.

O MAE adota uma arquitetura de *Encoder-Decoder* assimétrica: o *Encoder* processa apenas os elementos não mascarados, extraindo representações compactas e eficientes, enquanto o *Decoder* recebe essas representações junto com informações sobre as posições mascaradas para reconstruir a entrada original. Essa assimetria reduz significativamente o custo computacional durante o treinamento, permitindo o uso de taxas de mascaramento elevadas — frequentemente acima de 75% — sem perda substancial de desempenho. Como resultado, os MAEs se mostram particularmente adequados para pré-treinamento de

grandes modelos, servindo como base para tarefas posteriormente supervisionadas ou de *fine-tuning*. Nesta segunda etapa, o *Decoder*, que serviu apenas como ferramenta para o pré-treinamento, é completamente descartado, e apenas o *Encoder* pré-treinado é utilizado para a tarefa final.



Fonte: He *et al.*, 2021.

Figura 2.3: *Arquitetura do Masked Autoencoder*

2.3 Trabalhos Correlacionados

Diversos estudos têm explorado o uso de modelos baseados em CNNs e Transformers para a análise de sinais de áudio. No contexto da pandemia de COVID-19, a investigação da detecção de insuficiência respiratória a partir da fala ganhou destaque. Casanova, Candido Jr *et al.*, 2021, por exemplo, demonstraram a eficácia do aprendizado por transferência a partir de modelos de voz para essa tarefa, alcançando resultados promissores com dados limitados. De forma similar, Finger *et al.* (2021) e Gauy e Finger, 2021 focaram no uso de arquiteturas de redes neurais, focando principalmente no uso de MFCCs (*Mel-Frequency Cepstral Coefficients*) como entrada para o modelo, visando aumentar a acurácia do diagnóstico. Em conjunto, esses estudos estabelecem a viabilidade da abordagem, servindo como ponto de partida fundamental para o presente trabalho.

Paralelamente, o avanço em modelos de áudio pré-treinados e auto-supervisionados tem sido um pilar para o progresso da área. Kong *et al.*, 2020 introduziram as PANNs (*Pretrained Audio Neural Networks*), uma família de CNNs pré-treinadas em larga escala no dataset *AudioSet*,[1] que se tornaram uma forte base para diversas tarefas de classificação de áudio. Mais recentemente, a abordagem de aprendizado auto-supervisionado foi explorada por Huang *et al.*, 2023 com um Transformer que aprende representações robustas ao reconstruir espectrogramas mascarados, demonstrando um desempenho superior no *fine-tuning* com poucos dados. A eficácia de tais modelos pré-treinados, como também analisado por Gauy e Finger, 2022, reforça a aplicabilidade da metodologia proposta neste estudo, que busca combinar o poder dessas representações com uma análise crítica das etapas de pré-processamento.

[1] https://research.google.com/audioset/dataset/index.html

Capítulo 3

Metodologia

Este capítulo detalha todos os procedimentos metodológicos adotados para a condução deste trabalho, desde a preparação dos dados até a descrição dos experimentos realizados para avaliar o impacto da filtragem de ruído no treinamento de modelos de rede neural.

3.1 Conjunto de Dados

Os dados utilizados nesse trabalho foram obtidos do *SPIRA Dataset*.[1] Esses dados foram coletados utilizando smartphones durante a pandemia de COVID-19. O *dataset* é composto por áudios de duas classes: vozes de pessoas saudáveis (grupo de controle), gravados majoritariamente em ambientes controlados e silenciosos devido ao isolamento social, e áudios de pessoas em situação de insuficiência respiratória (grupo de pacientes com concentração de oxigênio no sangue abaixo de 92%), frequentemente capturados em ambientes clinicamente mais complexos e ruidosos.

Para a realização da tarefa, foi utilizada uma versão balanceada do *dataset*, que incuía gravações de 423 pessoas diferentes pronunciando a mesma frase de referência: "O amor ao próximo ajuda a enfrentar o coronavírus com a força que a gente precisa". Esses dados foram divididos para treinamento (283), validação (32) e testes (108), mantendo o balanceamento entre as duas classes em todos os conjuntos de dados, conforme detalhado na Tabela 3.1.

Conjunto	Homem (Controle)	Mulher (Controle)	Homem (Paciente)	Mulher (Paciente)	Total (Controle)	Total (Paciente)	Total
Treinamento	57	84	76	66	141	142	283
Validação	8	8	8	8	16	16	32
Teste	22	26	28	32	54	54	108

Tabela 3.1: *Distribuição dos dados utilizados*

[1] https://github.com/Edresson/SPIRA-ACL2021

Conjunto	Duração Média (s) (Controle)	Duração Média (s) (Paciente)
Treinamento	8.16	13.23
Validação	7.75	10.78
Teste	8.77	9.44

Tabela 3.2: *Média de duração dos dados em segundos*

3.2 Pré-Processamento

No contexto desse trabalho, pode-se considerar essa como sendo a fase mais crítica, pois é onde os problemas intrínsecos do *dataset* são tratados, assim como adaptados para uma melhor compatibilidade e performance nos modelos utilizados.

3.2.1 Filtragem de Ruído

Como detalhado na Seção 3.1, o desafio mais crítico deste dataset é o viés ambiental, originado da diferença sistemática entre os ambientes de gravação dos grupos de controle (silenciosos) e de pacientes (ruidosos). Esse viés representa o risco de o modelo aprender a classificar os áudios com base no ruído de fundo em vez dos padrões vocais da patologia, um fenômeno conhecido como *shortcut learning*.

Para investigar e tentar neutralizar este viés, foi utilizado um filtro de ruído que atua nos domínios tempo-amplitude e tempo-frequência baseado em subtração espectral, desenvolvido durante o projeto SPIRA como um trabalho de formatura (Pereira, 2020) e cujo funcionamento será detalhado na Seção 3.3.

3.2.2 Transformação do Domínio do Áudio

Após a etapa de filtragem, o sinal de áudio ainda se encontra no domínio do tempo. Conforme discutido no Capítulo 2, essa representação não é a ideal para ser processada por arquiteturas como CNNs e Transformers. Portanto, essa etapa do pré-processamento consiste na conversão de cada arquivo de áudio para a sua representação tempo-frequência, nesse caso a representação de *Log-Mel Spectrogram*.

Nessa representação, além de se considerar o áudio de forma mais intuitiva visualmente por meio da transformação para espectrograma utilizando a STFT, também é feita uma transformação para uma escala que representa melhor a forma como o sistema auditivo humano percebe as frequências, como exemplificado na Figura 3.1.

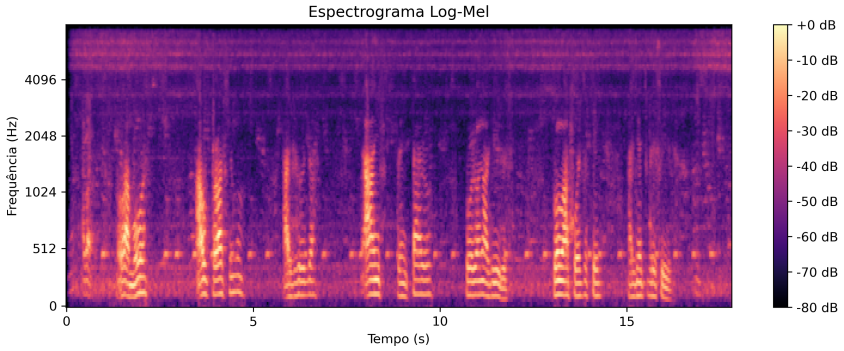


Figura 3.1: *Exemplo do Espectrograma Log-Mel de um áudio.*

3.2.3 Janelamento

Conforme apresentado na Tabela 3.2, a duração das gravações de áudio é variável entre as amostras e, notavelmente, apresenta uma média consistentemente menor para o grupo de pacientes. Uma vez que as arquiteturas de redes neurais poderiam aprender a usar essa diferença de duração como um atalho indesejado para a classificação, ou tipicamente exigem entradas de tamanho fixo, essa variabilidade precisa ser tratada.

Para resolver essa questão, foi aplicada uma técnica de janelamento (*windowing*) com sobreposição. Cada arquivo de áudio foi segmentado em janelas de 4 segundos, com uma sobreposição de 1 segundos entre janelas consecutivas. Este procedimento cumpre dois objetivos simultaneamente: normaliza o comprimento de todas as amostras de entrada para o modelo e atua como uma forma de aumentação de dados (*data augmentation*), multiplicando o número de exemplos disponíveis.

3.3 Filtro Utilizado

O filtro de ruído empregado neste trabalho, adaptado do trabalho desenvolvido no contexto do projeto SPIRA (Pereira, 2020), opera através de um processo de duas etapas principais: um classificador fala/ruído seguido por um supressor de ruído.

3.3.1 Etapa 1: Classificador Fala/Ruído

A primeira etapa do filtro consiste em um classificador que identifica e isola os segmentos do áudio que contêm apenas ruído de fundo operando no domínio do tempo da seguinte forma:

1. O sinal é dividido em janelas curtas n;
2. É calculado a energia em decibéis, $dB(i)$, para cada janela do sinal;
3. Um limiar de ruído (ϵ) é então estabelecido para classificar cada janela. Verificou-se que a ferramenta permite duas abordagens distintas para a definição deste limiar:
    - Limiar por Valor Fixo (dB): Neste modo, o limiar é definido adicionando-se um

valor fixo V_{dB} à energia mínima encontrada no sinal, $E_{dB_{min}}$:

$$\epsilon_{fixo} = \min_i\{dB(i)\} + V_{dB} \tag{3.1}$$

- Limiar por Porcentagem: Neste modo, o limiar é definido como uma porcentagem V_{pct} da faixa dinâmica de energia (a diferença entre a energia máxima e mínima do sinal):

$$\epsilon_{pct} = \min_i\{dB(i)\} + ((\max_i\{dB(i)\} - \min_i\{dB(i)\}) \times P_{limiar}) \tag{3.2}$$

4. Finalmente, qualquer janela x_i onde $dB(x_i) < \epsilon$ é classificada como ruído e as restantes são classificadas como fala.

3.3.2 Etapa 2: Supressor de Ruído

A segunda etapa, o supressor de ruído, utiliza o perfil de ruído detectado para realizar uma subtração espectral, operando no domínio tempo-frequência para atenuar as frequências associadas ao ruído no áudio original da seguinte forma:

1. O áudio original e os segmentos de ruído são convertidos para o domínio tempo-frequência usando a STFT, resultando nos respectivos espectrogramas utilizando janelas de tamanho $\mathbf{w} \times \mathbf{h}$;
2. Um perfil de ruído é criado a partir do espectrograma do ruído obtido no item 1, calculando-se, para cada frequência, a média e o desvio padrão de suas magnitudes.
3. Finalmente, o áudio original é filtrado por meio da aplicação, para cada frequência do espectrograma obtido no item 1, de um filtro que atenua por um fator de γ as componentes cuja magnitude não ultrapassa a média (μ_i) mais a desvios padrão (σ_i) das magnitudes correspondentes a essa frequência conforme a equação 3.3.

$$l'_i = \begin{cases} l_i, & \text{se } |l_i| > \mu_i + \mathbf{a}\sigma_i \\ \gamma\, l_i, & \text{caso contrário} \end{cases} \tag{3.3}$$

3.3.3 Software

O programa do filtro utilizado opera, portanto, utilizando 5 parâmetros principais a serem analisados (além do sinal de áudio):

- **n_grad_freq (h)**: Altura da janela a ser utilizada no supressor;
- **n_grad_time (w)**: Largura da janela a ser utilizada no supressor;
- **n_std_thresh (a)** : Define quantos desvios padrão acima da média uma frequência precisa estar para ser considerada sinal.
- **prop_decrease ($1 - \gamma$)**: Define, em porcentagem, quanto do ruído detectado deve ser reduzido do áudio original;

- **noise_threshold / noise_threshold_pct**: Métodos para criação do limiar do ruído, respectivamente com valor fixo e em porcentagem.

3.4 Modelos Utilizados

Para investigar o impacto do viés de pré-processamento e avaliar a eficácia das soluções propostas, duas arquiteturas de rede neural pré-treinadas distintas foram empregadas, ambas já introduzidas conceitualmente no Capítulo 2 e exemplificadas a seguir.

3.4.1 Rede Neural Convolucional: CNN10 (PANNs)

Para a realização dos experimentos iniciais, foi utilizada a arquitetura CNN10, mostrada na Figura 3.2 que faz parte da família de Redes Neurais de Áudio Pré-treinadas (PANNs) proposta por Kong *et al.* (2020).

Este modelo é pré-treinado no dataset *AudioSet* e foi utilizado alterando sua "cabeça"para realizar uma classificação binária e realizando um processo de *fine-tuning* nos nossos dados.

VGGish [1]	CNN6	CNN10	CNN14
Log-mel spectrogram 96 frames × 64 mel bins	Log-mel spectrogram 1000 frames × 64 mel bins		
3 × 3 @ 64 ReLU	5 × 5 @ 64 BN, ReLU	(3 × 3 @ 64 BN, ReLU) × 2	(3 × 3 @ 64 BN, ReLU) × 2
MP 2 × 2	Pooling 2 × 2		
3 × 3 @ 128 ReLU	5 × 5 @ 128 BN, ReLU	(3 × 3 @ 128 BN, ReLU) × 2	(3 × 3 @ 128 BN, ReLU) × 2
MP 2 × 2	Pooling 2 × 2		
(3 × 3 @ 256 ReLU) × 2	5 × 5 @ 256 BN, ReLU	(3 × 3 @ 256 BN, ReLU) × 2	(3 × 3 @ 256 BN, ReLU) × 2
MP 2 × 2	Pooling 2 × 2		
(3 × 3 @ 512 ReLU) × 2	5 × 5 @ 512 BN, ReLU	(3 × 3 @ 512 BN, ReLU) × 2	(3 × 3 @ 512 BN, ReLU) × 2
MP 2 × 2 Flatten	Global pooling		Pooling 2 × 2
FC 4096 ReLU × 2	FC 512, ReLU		(3 × 3 @ 1024 BN, ReLU) × 2
FC 527, Sigmoid	FC 527, Sigmoid		Pooling 2 × 2
			(3 × 3 @ 2048 BN, ReLU) × 2
			Global pooling
			FC 2048, ReLU
			FC 527, Sigmoid

Fonte: Kong *et al.*, 2020.

Figura 3.2: *Arquitetura das PANNs*

Para este trabalho, foi optada pela utilização da CNN10 dentre as PANNs por conta dos resultados obtidos em estudos anteriores utilizando as redes pré-treinadas objetivando a análise de voz como visto em Gauy e Finger (2022).

3.4.2 Masked Autoencoder: AudioMAE

Como arquitetura utilizada para a avaliação de uma solução de treinamento robusta, foi empregado o modelo AudioMAE, proposto por Huang *et al.* (2023). Esta arquitetura é baseada em Transformers e utiliza a estratégia de pré-treinamento auto-supervisionado Masked Autoencoder (MAE), conforme detalhado conceitualmente na Seção 2.2.3.

O modelo utilizado também foi pré-treinado no dataset *AudioSet*, porém com a tarefa auto-supervisionada de reconstruir espectrogramas mascarados. O processo de fine-tuning para a tarefa de classificação seguiu o protocolo padrão do MAE:

1. O *Decoder*, utilizado apenas durante a fase de pré-treinamento para a reconstrução dos patches, foi completamente descartado.
2. O *Encoder* pré-treinado, que aprendeu a extrair representações contextuais ricas dos dados, foi mantido como o "corpo"do modelo.
3. Uma nova "cabeça"de classificação, foi adicionada ao topo do *Encoder* para adaptá-lo à nossa tarefa de classificação binária.

A escolha desse modelo se deve à hipótese de que um modelo forçado a aprender a estrutura fundamental dos sons será mais robusto aos vieses superficiais, como o ruído de fundo, em comparação com um modelo supervisionado tradicional.

3.5 Métricas de Avaliação

Para avaliar o desempenho dos modelos e diagnosticar a presença ou não de vieses de aprendizado, a avaliação foi conduzida com base na análise da matriz de confusão, que relaciona as predições do modelo com os rótulos reais dos dados.

Em nossa classificação binária, a classe "Positiva"refere-se à presença de insuficiência respiratória (Pacientes) e a classe "Negativa"refere-se aos indivíduos saudáveis (Controle). Assim, definem-se:

- Verdadeiros Positivos (TP): Pacientes corretamente identificados com insuficiência respiratória.
- Verdadeiros Negativos (TN): Indivíduos saudáveis corretamente identificados como controle.
- Falsos Positivos (FP): Indivíduos saudáveis incorretamente classificados como pacientes.
- Falsos Negativos (FN): Pacientes incorretamente classificados como saudáveis.

A partir destes valores, foram calculadas as seguintes métricas:

3.5.1 Acurácia (*Accuracy*)

A acurácia mede a proporção global de acertos do modelo sobre o total de amostras. Embora seja uma métrica intuitiva, ela pode ser enganosa em cenários de possível

viés extremo.

$$Acurácia = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.4}$$

3.5.2 Precisão (*Precision*)

A precisão avalia a qualidade das predições positivas. Ela responde à pergunta: "De todos os indivíduos que o modelo classificou como pacientes, quantos realmente eram pacientes?". Uma precisão baixa indica uma alta taxa de falsos positivos.

$$Precisão = \frac{VP}{VP + FP} \tag{3.5}$$

3.5.3 Sensibilidade (*Recall*)

Esta métrica é crítica em aplicações médicas de triagem. Ela mede a capacidade do modelo de detectar a doença quando ela existe. Um valor baixo de Recall é indesejável, pois implica que pacientes com insuficiência respiratória estão deixando de ser diagnosticados.

$$Recall = \frac{TP}{TP + FN} \tag{3.6}$$

3.5.4 Especificidade (*Specificity*)

Complementar ao Recall, a especificidade mede a capacidade do modelo de identificar corretamente os casos negativos (saudáveis). Esta métrica é particularmente importante neste trabalho para a análise de viés.

$$Especificidade = \frac{TN}{TN + FP} \tag{3.7}$$

3.5.5 F1-Score

O F1-Score é a média harmônica entre a Precisão e o *Recall*. Ele fornece uma medida única que penaliza valores extremos. É uma métrica essencial para comparar modelos, pois exige que a rede tenha um bom desempenho tanto na detecção da doença (*Recall*) quanto na confiabilidade dessa detecção (*Precisão*).

$$F1\text{-}Score = 2 \times \frac{Precisão \times Recall}{Precisão + Recall} \tag{3.8}$$

Capítulo 4

Experimentos e Resultados

Este capítulo apresenta a sequência de experimentos realizados para investigar o impacto da filtragem de ruído no desempenho dos modelos utilizados. A estrutura segue a própria linha de experimentos, onde o resultado de um experimento justifica as decisões tomadas para conduzir o próximo.

4.1 Configuração de Treinamento

Para garantir a reprodutibilidade, todos os modelos foram treinados (*fine-tuning*) seguindo uma configuração padronizada:

- **Função de Perda (*Loss Function*)**: Para esta tarefa de classificação, foi utilizada a função *Cross-Entropy Loss*;
- **Otimizador**: Foi empregado o otimizador Adam com $betas = (0.9, 0.98)$ e com taxas de aprendizado diferentes para a base do modelo e nossa camada de classificação adicional:
  - Camadas de Base (Pré-treinadas): Foi aplicada uma taxa de aprendizado conservadora de $lr =$ 1e-5. Isso permite que os pesos da rede principal sejam ajustados sutilmente para a nova tarefa.
  - Camada de Classificação ("Cabeça"): Foi aplicada uma taxa de aprendizado de $lr = 1e - 4$. Como esta camada é nova e seus pesos são inicializados aleatoriamente, ela requer um aprendizado mais rápido e agressivo para mapear as características extraídas pela base para a nossa tarefa binária;
  - Seed = 42 fixa durante todos os experimentos;

4.1.1 Configurações de Pré-Processamento

Como os modelos utilizados são pré-treinados, os parâmetros de extração de características foram mantidos fixos para corresponder à configuração de pré-treinamento dos modelos:

Parâmetro	CNN10	AudioMAE
Sample Rate	32000	16000
Window Size	1024	1024
Hop Size	320	320
Mel Bins	64	128
Fmin	0	0
Fmax	16000	8000

Tabela 4.1: *Configurações dos modelos utlizados*

4.2 Experimentos - CNN10

A investigação sobre a eficácia da filtragem foi iniciada utilizando a arquitetura CNN10 como modelo de base.

4.2.1 Experimento 1: Análise de Instabilidade da Filtragem

O primeiro experimento teve como objetivo avaliar a estabilidade dos dois modos de operação do classificador do filtro (descritos na Seção 3.3).

Configuração de Treinamento

Para este experimento, o modelo foi treinado por 50 épocas. Os parâmetros do filtro foram mantidos fixos, utilizando os valores padrão recomendados baseados em estudos anteriores. Os valores utilizados foram:

- **n_grad_time** = 3
- **n_grad_freq** = 3
- **n_std_thresh** = 2.0

Para investigar a influência do método de limiarização, foram comparadas duas abordagens: um primeiro modelo, treinado com limiar de valor fixo ($6dB$), e outro treinado com limiar por porcentagem (34%).

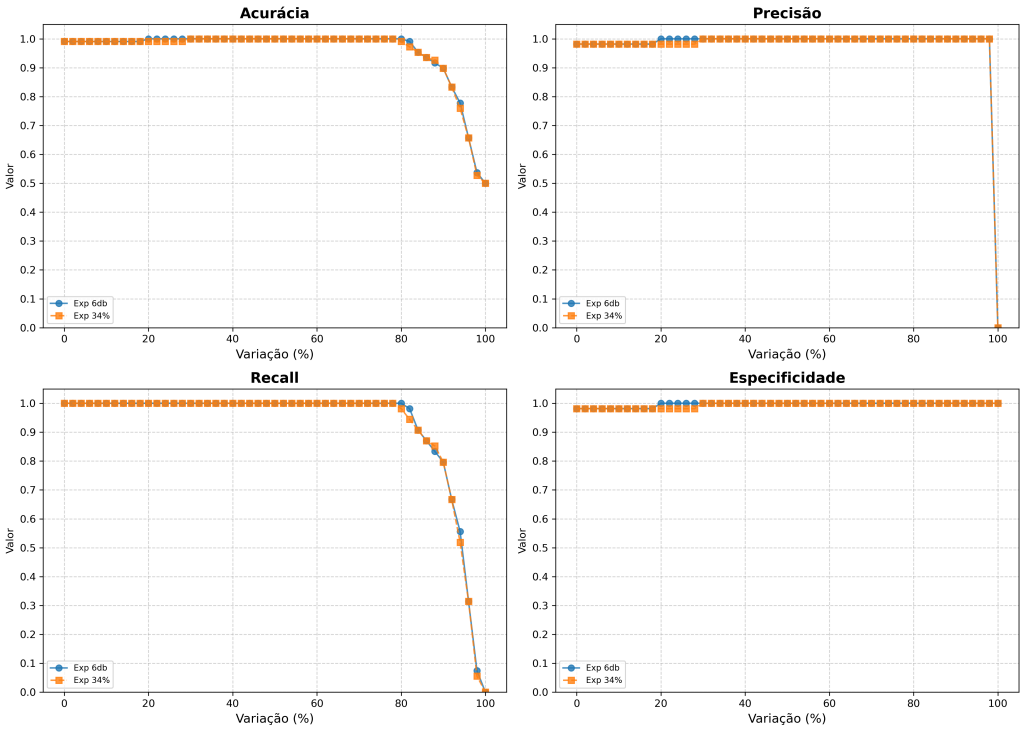
Adicionalmente, para avaliar o impacto da intensidade da remoção de ruído, ambos os modelos foram submetidos a dois cenários de supressão distintos:

1. **Moderada:** Redução parcial (50%, ou $prop\_decrease = 0.5$) do ruído detectado;
2. **Agressiva:** Remoção total (100%, ou $prop\_decrease = 1.0$) do ruído detectado.

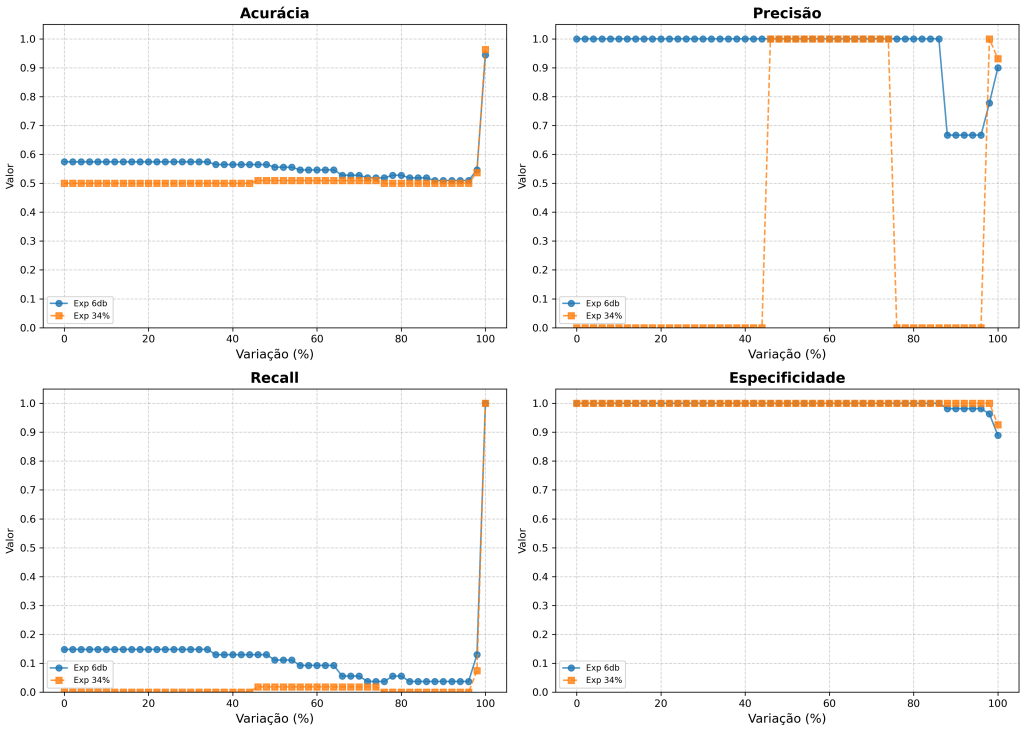
Essa distinção foi feita para verificar se a supressão parcial introduz artefatos que afetam a estabilidade do aprendizado.

Após o treinamento, o modelo salvo foi submetido a um teste de sensibilidade. O conjunto de teste foi processado 50 vezes, com o parâmetro de redução *prop_decrease* variando linearmente de 1.0 até seu valor mínimo 0.0 (sem filtragem). A performance do modelo foi registrada em cada um desses 50 pontos para gerar as curvas de sensibilidades.

Experimento 1: Resultados



(a) *Modelos Treinados com Supressão Moderada (50%)*



(b) *Modelos Treinados com Supressão Agressiva (100%)*

Figura 4.1: *Análise de Sensibilidade entre os Modos de Filtragem.*

O resultado, apresentado na Figura 4.1, demonstra uma instabilidade severa em ambos os cenários.

Visualmente, observa-se que os modelos apresentam alta performance apenas quando os parâmetros de teste coincidem com os de treino, formando 'picos' de acurácia ou 'penhascos' onde o desempenho falha catastroficamente. Isso prova que os modelos, independentemente do modo do classificador, sofreram overfitting aos artefatos específicos criados pelo filtro, em vez de aprenderem padrões da patologia.

4.2.2 Experimento 2: Validação da Robustez

Após identificar no Experimento 1 que a configuração padrão do filtro induzia o modelo ao aprendizado de atalhos (shortcut learning) devido a artefatos na subtração espectral, este experimento teve como objetivo validar uma configuração mais robusta.

A hipótese central era que as janelas de estimativa de ruído originais (3×3) eram demasiadamente pequenas, gerando um perfil de ruído muito específico e ruidoso. Para mitigar isso, buscou-se suavizar a estimativa do ruído aumentando as dimensões da janela de análise, conforme permitido pela ferramenta Pereira, 2020.

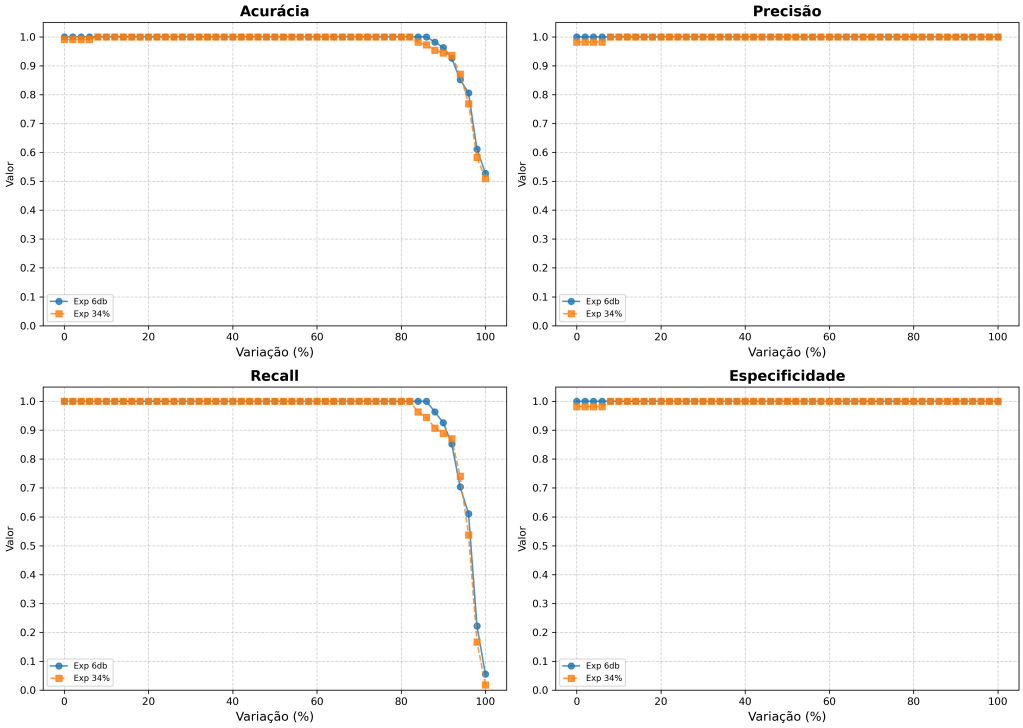
Configuração de Parâmetros

Foram definidos novos parâmetros visando uma generalização maior do perfil de ruído (suavização), mantendo-se o limiar de detecção conservador:

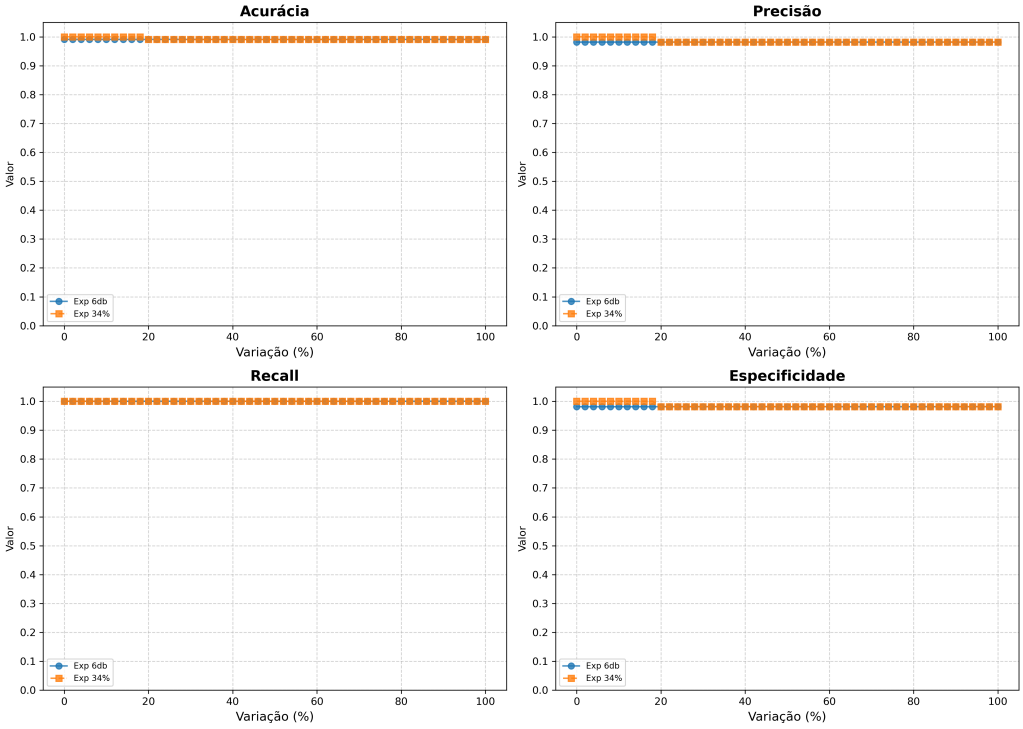
- **n_grad_time** = 8
- **n_grad_freq** = 4
- **n_std_thresh** = 1.5

Com esses novos parâmetros de base, repetiu-se integralmente o protocolo de sensibilidade do Experimento 1, treinando e testando modelos em quatro cenários distintos.

Análise dos Resultados



(a) *Modelos Treinados com Supressão Moderada (50%)*



(b) *Modelos Treinados com Supressão Agressiva (100%)*

Figura 4.2: *Análise de Estabilidade dos Parâmetros.*

Os resultados demonstraram uma mudança drástica no comportamento da CNN10.

Diferentemente da instabilidade observada anteriormente, a aplicação dos parâmetros de suavização resultou em curvas de sensibilidade lineares, indicando que o modelo parou de utilizar os artefatos de filtragem como critério de decisão.

Observou-se especificamente que:

1. **Impacto da Intensidade:** A supressão moderada ($prop\_decrease = 0.5$) demonstrou manter o viés de filtragem, mesmo após a suavização da mesma. A supressão intensa ($prop\_decrease = 1.0$), que no Experimento 1 era a mais instável, tornou-se mais robusta com a suavização. Isso valida o uso da remoção total do ruído, essencial para eliminar o viés ambiental sem inserir viés de filtragem.
2. **Fixo vs. Porcentagem:** Com o filtro estabilizado, ambos os métodos de limiar (Fixo em 6dB e Porcentagem em 34%) apresentaram desempenho e estabilidade equivalentes. A "falha" observada no experimento anterior não era intrínseca aos métodos, mas sim exarcebada pela parametrização inadequada.

Definição dos Parâmetros Finais

Diante da estabilidade alcançada, definiu-se a configuração final que será utilizada para a investigação principal. Acrescentado aos parâmetros do Experimento 2, optou-se pelo método de **Limiar por Porcentagem (**34%**)** devido à sua vantagem conceitual de adaptar-se à faixa dinâmica de diferentes dispositivos de gravação, combinado com a **Supressão Total (**100%**)** para garantir a máxima remoção do ambiente hospitalar e a estabilidade da filtragem.

4.3 Experimentos - AudioMAE

4.3.1 Transferabilidade dos Parâmetros de Filtragem

Após a estabilização bem-sucedida da CNN10 no Experimento 2, o próximo passo lógico foi aplicar a mesma configuração "ótima"de filtragem ao modelo AudioMAE. A hipótese era que, sendo uma arquitetura mais avançada e pré-treinada de forma robusta, o AudioMAE se beneficiaria igualmente (ou até mais) da estabilidade proporcionada pelos parâmetros de suavização:

- **n_grad_time** = 8
- **n_grad_freq** = 4
- **n_std_thresh** = 1.5

Resultados e Discussão

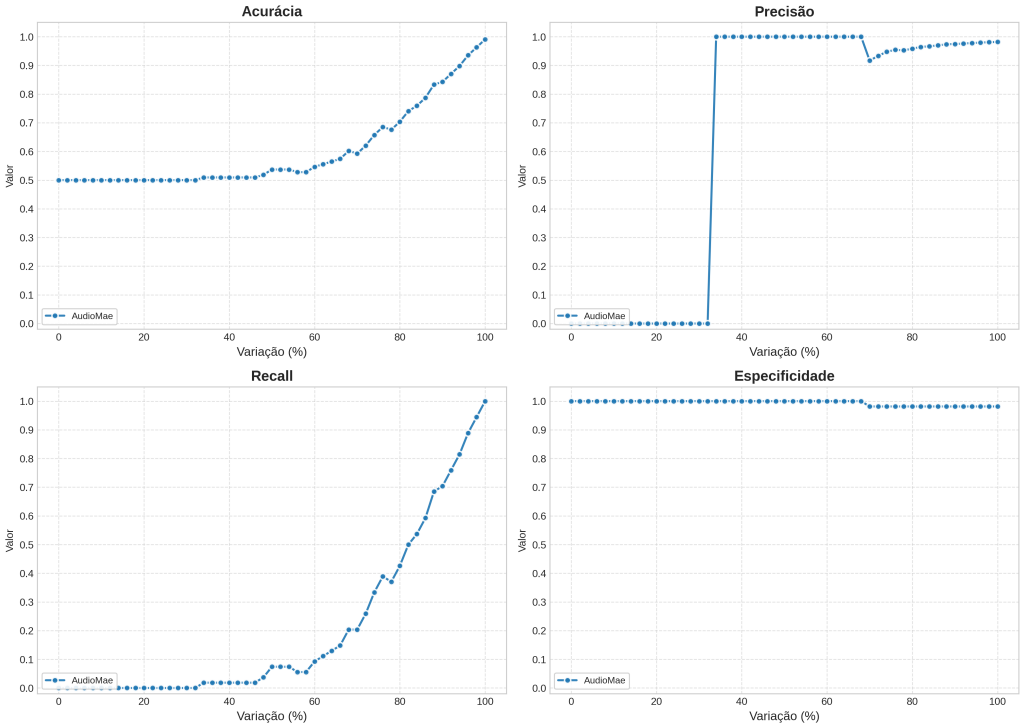


Figura 4.3: *Desempenho do AudioMAE com os parâmetros encontrados*

Contrariando a hipótese inicial, os resultados do teste de sensibilidade para o AudioMAE (Figura 4.3) revelaram que a instabilidade persistiu. Diferentemente da CNN10, que apresentou uma curva linear e estável, o AudioMAE continuou demonstrando sensibilidade às variações da intensidade do filtro, mantendo o comportamento de queda de performance fora do ponto de treino.

Este resultado sugere uma diferença fundamental na forma como a nova arquitetura lida com artefatos de pré-processamento: O AudioMAE, baseado em *Vision Transformers*, processa o espectrograma em patches e utiliza mecanismos de *self-attention* que capturam relações globais e detalhes minuciosos. Devido à sua alta capacidade, o modelo foi capaz de detectar e aprender os artefatos da filtragem mesmo com os parâmetros de suavização, resultando no *overfitting* de pré-processamento.

Conclui-se, portanto, que os parâmetros de filtragem **não são universais**.

Foi conduzida uma investigação exploratória preliminar com diferentes configurações de filtragem para o AudioMAE, No entanto, as análises demonstraram que a instabilidade ao filtro persistiu de forma consistente, independentemente das combinações testadas. A persistência desse fenômeno evidencia que a sensibilidade do AudioMAE não é meramente uma questão de ajuste de hiperparâmetros, mas sim uma consequência de sua arquitetura robusta baseada em atenção.

Apesar da sensibilidade observada aos artefatos de filtragem, para a condução do teste final de verificação de viés ambiental, optou-se por manter a configuração de filtragem padronizada, idêntica àquela validada para a CNN10. Essa decisão visa garantir a consistência metodológica e permitir uma comparação direta entre a capacidade das duas

arquiteturas de lidar com o viés do dataset sob as mesmas condições de pré-processamento, independentemente das variações de estabilidade interna de cada modelo.

4.4 Experimento Final - Verificação de Viés Ambiental

Com os parâmetros de filtragem estabilizados (conforme definido nos experimentos anteriores), foi conduzido o teste definitivo para verificar se o pré-processamento foi capaz de mitigar o viés ambiental do dataset.

A Tabela 4.2 sumariza o desempenho dos dois modelos (CNN10 e AudioMAE) em dois cenários distintos: o Cenário Original, utilizando o conjunto de teste padrão, e o Cenário com Ruído, onde ruído hospitalar foi inserido artificialmente em todas as amostras antes da filtragem.

Modelo	Cenário de Teste	Acurácia	F1	Precisão	Recall	Especificidade
CNN10	Filtragem sem Inserção	**0.9907**	0.9908	0.9818	1.0000	0.9815
	Filtragem com Inserção	**0.5278**	0.6792	0.5143	1.0000	0.0556
AudioMAE	Filtragem sem Inserção	**0.9907**	0.9908	0.9818	1.0000	0.9815
	Filtragem com Inserção	**0.5000**	0.6667	0.5000	1.0000	0.0000

Tabela 4.2: *Comparação de desempenho dos modelos (CNN10 e AudioMAE) nos conjuntos de teste com e sem inserção de ruído hospitalar antes da filtragem.*

4.4.1 Análise dos Resultados

Observa-se, primeiramente, que no Cenário Original, ambos os modelos apresentam um desempenho funcional, com o AudioMAE atingindo índices de acurácia e F1-Score superiores a 99%. Estes resultados, embora aparentemente excelente, devem ser interpretados com cautela.

O resultado crítico revela-se no Cenário com Ruído. Ao introduzir o ruído hospitalar nas amostras de controle e submetê-las ao mesmo processo de filtragem, podemos observar:

1. Colapso da Especificidade: Para ambos os modelos, a Especificidade caiu para 0.00. Isso indica que os modelos falharam em identificar qualquer indivíduo saudável corretamente quando este estava inserido em um ambiente ruidoso.

2. Precisão de 50%: Como o modelo passou a classificar indiscriminadamente todas as amostras como pertencentes à classe ’Paciente’ (devido à presença do ruído), a precisão caiu devido à alta presença de falsos positivos.

3. Recall de 100%: O Recall máximo não indica uma detecção perfeita da doença, mas sim que o modelo classificou todas as amostras (positivas e negativas) como sendo da classe "Paciente".

4. Acurácia de 50%: Em um conjunto de teste balanceado, classificar todas as amostras como pertencentes a uma única classe resulta matematicamente em uma acurácia de 50%, equivalente a um classificador aleatório ou tendencioso.

Capítulo 5

Conclusão

O objetivo central deste trabalho foi investigar a eficácia da filtragem de ruído como uma estratégia de pré-processamento para mitigar o viés ambiental presente em *datasets* como o do projeto SPIRA. A premissa inicial era que, ao remover o ruído de fundo característico dos ambientes hospitalares, seria possível forçar modelos de aprendizado profundo a aprenderem biomarcadores da insuficiência respiratória, eliminando o fenômeno de *shortcut learning*.

A investigação foi conduzida através de um protocolo experimental rigoroso, dividido em etapas de análise de estabilidade e verificação de viés. Os experimentos iniciais revelaram que a aplicação de filtros com parâmetros fixos introduz um novo problema: um "viés de filtragem". Demonstrou-se que os modelos tendem a sofrer overfitting nos artefatos espectrais gerados pelo processo de supressão, resultando em uma performance que se degrada com pequenas variações nos parâmetros de teste.

O experimento final de viés trouxe a descoberta mais significativa deste trabalho. Ao submeter os modelos a um conjunto de teste onde ruído hospitalar era inserido antes da etapa de filtragem, observou-se um colapso total das métricas analisadas. Este resultado comprova que a filtragem de ruído não resultou na remoção do viés. Tanto a CNN10 quanto o AudioMAE continuaram a utilizar o ruído residual (ou os fragmentos resultantes do processo de filtragem) como o principal discriminador para a classe "Paciente".

Conclui-se, portanto, que a filtragem de ruído mostrou-se uma estratégia insuficiente para a resolução do problema encontrado, pois não remove a correlação entre o ambiente acústico e a patologia, servindo apenas para transformar a natureza do atalho aprendido pelo modelo.

O fato de uma arquitetura avançada e pré-treinada como o AudioMAE falhar da mesma maneira que uma CNN tradicional reforça que a complexidade do modelo não é solução para dados enviesados. Pelo contrário, a maior capacidade de aprendizado destes modelos pode torná-los ainda mais sensíveis a atalhos sutis.

Diante dos resultados obtidos, sugere-se que pesquisas futuras no contexto do projeto SPIRA e similares foquem em abordagens diametralmente opostas, recomendando a investigação de outras técnicas de tratamento de áudio ou de tratamento direto do dataset

como estratégias de *Data Augmentation*.

Referências

[Casanova, Candido Jr *et al.* 2021] Edresson Casanova, Arnaldo Candido Jr *et al.* "Transfer learning and data augmentation techniques to the covid-19 identification tasks in compare 2021." In: *Interspeech*. 2021, pp. 446–450 (citado na pg. 8).

[Casanova, Gris *et al.* 2021] Edresson Casanova, Lucas Gris *et al.* "Deep learning against COVID-19: respiratory insufficiency detection in Brazilian Portuguese speech". In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Ed. por Chengqing Zong, Fei Xia, Wenjie Li e Roberto Navigli. Online: Association for Computational Linguistics, ago. de 2021, pp. 625–633. doi: 10.18653/v1/2021.findings-acl.55. url: https://aclanthology.org/2021.findings-acl.55/ (citado na pg. 5).

[Finger *et al.* 2021] Marcelo Finger *et al.* "Detecting respiratory insufficiency by voice analysis: the spira project". *Acoustic communication: an interdisciplinary approach* (2021) (citado nas pgs. 1, 8).

[Gauy e Finger 2021] Marcelo Matheus Gauy e Marcelo Finger. "Audio mfcc-gram transformers for respiratory insufficiency detection in covid-19". In: *Anais do XIII Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana (STIL 2021)*. STIL 2021. Sociedade Brasileira de Computação, nov. de 2021, pp. 143–152. doi: 10.5753/stil.2021.17793. url: http://dx.doi.org/10.5753/stil.2021.17793 (citado na pg. 8).

[Gauy e Finger 2022] Marcelo Matheus Gauy e Marcelo Finger. *Pretrained audio neural networks for Speech emotion recognition in Portuguese*. 2022. arXiv: 2210.14716 [cs.SD]. url: https://arxiv.org/abs/2210.14716 (citado nas pgs. 8, 13).

[He *et al.* 2021] Kaiming He *et al. Masked Autoencoders Are Scalable Vision Learners*. 2021. arXiv: 2111.06377 [cs.CV]. url: https://arxiv.org/abs/2111.06377 (citado nas pgs. 7, 8).

[Huang *et al.* 2023] Po-Yao Huang *et al. Masked Autoencoders that Listen*. 2023. arXiv: 2207.06405 [cs.SD]. url: https://arxiv.org/abs/2207.06405 (citado nas pgs. 8, 14).

[Kong *et al.* 2020] Qiuqiang Kong *et al. PANNs: Large-Scale Pretrained Audio Neural Networks for Audio Pattern Recognition*. 2020. arXiv: 1912.10211 [cs.SD]. url: https://arxiv.org/abs/1912.10211 (citado nas pgs. 8, 13).

[Pereira 2020] Pedro Leyton Pereira. *noise-reduce-tool: Uma ferramenta para reduzir ruído nos áudios de voz do SPIRA*. 2020. URL: https://github.com/SPIRA-COVID19/noise-reduce-tool (citado nas pgs. 10, 11, 20).

[Rache *et al.* 2020] Beatriz Rache *et al.* "Necessidades de infraestrutura do sus em preparo à covid-19: leitos de uti, respiradores e ocupação hospitalar". *São Paulo: Instituto de Estudos para Políticas de Saúde* 3 (2020), pp. 1–5 (citado na pg. 1).

[Vaswani *et al.* 2017] Ashish Vaswani *et al.* "Attention is all you need". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 6000–6010. ISBN: 9781510860964 (citado nas pgs. 5, 7).