Benchmark de modelos de IA na validação de fraudes de *Deepfakes*

Elcio L. Furtili Junior[1]

[1]Ibilce - Instituto de Biociências, Letras e Ciências Exatas
Câmpus de São José do Rio Preto - UNESP
São José do Rio Preto – SP – Brasil

[2]IA PPGCC 2025
Ibilce (UNESP) – São José do Rio Preto, SP – Brasil

elcio.furtili@unesp.br

Abstract. *This study investigates the effectiveness of three artificial intelligence models — CNN, RNN with LSTM, and Wav2Vec 2.0 — in detecting fraud in audio signals generated by deepfake technologies. The dataset used was organized into original and manipulated audio samples, ensuring anonymity through segmentation into non-identifiable age groups. The comparative analysis was based on five key metrics: accuracy, precision, recall, F1-score, and AUC-ROC. The study relies on a dataset originally composed of 267,544 genuine audio samples and 425,700 manipulated samples, sourced from three multilingual corpora consolidated in the literature. However, to ensure computational feasibility and class balance, a balanced subset of 1,500 samples was adopted, consisting of 500 genuine and 1,000 manipulated audio files, properly distributed across the training and evaluation routines of the models.*

Resumo. *Este trabalho investiga a eficácia de três modelos de inteligência artificial — CNN, RNN com LSTM e Wav2Vec 2.0 — na detecção de fraudes em áudios gerados por tecnologias de deepfake. Os dados utilizados foram organizados em áudios originais e manipulados, respeitando critérios de anonimato por meio da segmentação em faixas etárias não identificáveis. A análise comparativa baseou-se em cinco métricas principais: acurácia, precisão, recall, F1-score e AUC-ROC. O estudo faz uso de um conjunto de dados originalmente composto por 267.544 áudios genuínos e 425.700 áudios adulterados, provenientes de três bases multilíngues consolidadas na literatura. Contudo, visando viabilidade computacional e equilíbrio entre as classes, foi adotada uma subamostra balanceada contendo 1.500 arquivos, sendo 500 áudios reais e 1.000 áudios manipulados, devidamente distribuídos entre as rotinas de treinamento e avaliação dos modelos.*

1. Introdução

A popularização de tecnologias baseadas em inteligência artificial voltadas à síntese de voz tem ampliado as possibilidades de criação de áudios realistas [Goodfellow et al. 2014], mas também gerado preocupações quanto ao seu uso indevido em fraudes e desinformação. Nesse cenário, torna-se essencial o desenvolvimento

de métodos capazes de identificar áudios manipulados, especialmente aqueles produzidos por ferramentas de *deepfake*. Este trabalho tem como objetivo avaliar o desempenho de três arquiteturas de IA na tarefa de detecção de áudios falsificados: redes neurais convolucionais (*CNN*) [Krizhevsky et al. 2017], redes recorrentes com unidades *LSTM* (*RNN-LSTM*) [Hochreiter and Schmidhuber 1997] e o modelo *Wav2Vec* 2.0 [Baevski et al. 2020], baseado em aprendizado auto-supervisionado.

Os experimentos foram conduzidos com dados divididos entre áudios genuínos e manipulados, organizados por faixas etárias não identificáveis, de modo a garantir a preservação da identidade dos participantes e uma análise imparcial. Para mensurar a eficácia dos modelos, foram utilizadas métricas amplamente reconhecidas na literatura [Schuller et al. 2021]: acurácia, precisão, *recall*, *F1-score* e *AUC-ROC*. Além da tarefa de detecção, esta pesquisa também se dedica à diferenciação dos padrões gerados pelos *deepfakes* presentes nos dados, com base em *datasets* que reúnem 267.544 áudios reais e 425.700 áudios adulterados. Desse total, foram utilizados 1500 arquivos, sendo 500 áudios originais e 1000 áudios adulterados, 20% das amostras foram destinadas ao treinamento dos modelos, enquanto os 80% restantes foram reservados para validação e análise.

O objetivo é explorar a viabilidade de se reconhecer padrões característicos de cada gerador, contribuindo para o aprimoramento de técnicas forenses aplicadas à verificação de autenticidade em conteúdos de voz, levando também em consideração artefatos que podem ser encontrados em áudio que passam por esse tipo de manipulação.

Para um maior entendimento, os artefatos em áudio são imperfeições ou distorções não naturais presentes em uma gravação de som. Esses artefatos podem surgir por diversos motivos, como compressão exagerada, ruídos de codificação, erros na síntese de voz ou manipulações com inteligência artificial. No contexto de *deepfakes*, artefatos geralmente aparecem como transições abruptas, falhas de entonação, ruídos metálicos ou ausência de pausas naturais, e podem ser pistas importantes para detectar áudios falsificados. Mesmo quando não são audíveis a ouvido humano, algoritmos de detecção conseguem identificar esses padrões por meio de espectrogramas ou extração de *features*.

Além dos avanços na síntese de voz, observa-se um aumento significativo nos desafios relacionados à detecção de *deepfakes*, especialmente diante de modelos generativos cada vez mais sofisticados, como os baseados em *Voice Cloning* e *Text-to-Speech* (TTS) neural. Esses modelos são capazes de produzir áudios altamente realistas, incluindo nuances como entonação, ritmo e sotaque, tornando o problema de autenticação de voz extremamente desafiador.

Diante disso, surgem preocupações quanto à robustez dos sistemas atuais de detecção, principalmente em cenários de ataques *zero-day*, onde o modelo detector nunca foi exposto aos tipos de manipulações presentes nos dados. Este cenário exige o desenvolvimento de abordagens mais generalistas, capazes de identificar padrões residuais de manipulação, independentemente do gerador utilizado ou do idioma do áudio.

2. Trabalhos Relacionados

Nas últimas décadas, a síntese de voz baseada em inteligência artificial tem avançado significativamente, impulsionada principalmente por arquiteturas de redes neurais profundas

e técnicas de aprendizado auto-supervisionado. O surgimento de redes generativas adversariais (*GANs*) [Goodfellow et al. 2014], revolucionou a geração de dados sintéticos, abrindo caminho para a criação de áudios altamente realistas, conhecidos como *deepfakes*.

A detecção de *deepfakes* em áudio tornou-se um campo emergente e crítico dentro da segurança digital. Por meio do estudo [Schuller et al. 2021] exploraram o uso de aprendizado auto-supervisionado para a identificação de manipulações em sinais sonoros, propondo métricas robustas como *AUC-ROC* e *F1-score* para avaliar a eficácia dos classificadores.

Diversos modelos de aprendizado profundo têm sido aplicados à tarefa de detecção de fraudes por voz. As redes neurais convolucionais (*CNNs*), amplamente utilizadas para reconhecimento de padrões visuais [Krizhevsky et al. 2017], demonstraram capacidade de extrair representações espaciais relevantes também em espectrogramas de áudio. Por outro lado, as redes recorrentes com unidades de memória de longo curto prazo (LSTM), conforme formulado por *Hochreiter* e *Schmidhuber* [Hochreiter and Schmidhuber 1997], mostraram-se eficazes na modelagem de sequências temporais, sendo úteis na análise de sinais de fala com dependência contextual.

Mais recentemente, modelos auto-supervisionados como o *Wav2Vec 2.0* [Baevski et al. 2020], têm alcançado resultados notáveis ao aprender representações fonéticas diretamente de áudios brutos, reduzindo a dependência de transcrições ou alinhamentos fonéticos supervisionados.

No contexto de geração de voz sintética, existem muitas formas de executar essas manipulações dos áudios, isso faz com que se torne uma tarefa complicada definir qual ferramenta é responsável pela geração desses dados manipulados, então o foco na pesquisa é se basear no processos de análise genéricas e que possam ser generalizados, o ponto de maior relevância (não sendo o único), entretanto um dos focos nessa pesquisa é justamente a busca por artefatos resquiciais nesses áudios adulterados.

3. Metodologia

Este estudo propõe uma abordagem comparativa para a detecção de áudios manipulados por *deepfake*, fundamentada em três arquiteturas de inteligência artificial com diferentes paradigmas de aprendizagem: redes neurais convolucionais (*CNN*), redes recorrentes com unidades *LSTM* (*RNN-LSTM*) e o modelo *Wav2Vec 2.0*.

As redes *CNN* e *RNN-LSTM* são arquiteturas genéricas, nas quais é necessário definir a estrutura do modelo (camadas, funções de ativação etc.) e realizar o treinamento supervisionado completo com dados rotulados (áudios reais vs. *deepfakes*). Essas arquiteturas geralmente utilizam espectrogramas ou *features* acústicas extraídas dos sinais de áudio como entrada.

O modelo *Wav2Vec 2.0*, por sua vez, é uma arquitetura pré-treinada desenvolvida pela *Meta* (Facebook AI), baseada em aprendizado auto-supervisionado em grandes volumes de dados. Ele pode ser utilizado como extrator de *features*, sem necessidade de retreinamento, ou ajustado via *fine-tuning* para tarefas específicas, como a detecção de *deepfakes*. No entanto, apesar de permitir esse ajuste, o núcleo do modelo já vem treinado e, portanto, o *fine-tuning* é opcional — e não foi empregado nos experimentos deste

estudo.

A base de dados utilizada neste estudo é composta por amostras de áudio genuínas e manipuladas, extraídas de três *datasets* públicos amplamente utilizados na literatura: *CVoiceFake*, *MLADDC* e *VSASV*. No total, foram reunidos 267.544 áudios originais e 425.700 áudios adulterados. Cada conjunto apresenta particularidades relevantes para a tarefa de detecção, conforme resumido a seguir:

- ***CVoiceFake***: contém cerca de 75.000 áudios reais e 125.000 manipulados, com foco em ataques de *voice conversion* e *text-to-speech*, incluindo múltiplos idiomas.
- ***MLADDC***: reúne aproximadamente 92.544 áudios genuínos e 150.700 adulterados, destacando-se pela variedade de idiomas e geradores de *deepfake*.
- ***VSASV***: composto por cerca de 100.000 áudios originais e 150.000 falsificados, voltado à verificação de locutor e avaliação de sistemas *anti-spoofing*.
- **Arquivos Utilizados**: foram utilizados 1500 arquivos, sendo 500 arquivos de áudios reais e 1000 arquivos de áudios adulterados, 20% deles sendo arquivos para testes e os 80% restante para execução.

3.1. Arquitetura *Transformer*

A arquitetura *Transformer*, introduzida por [Vaswani et al. 2017], revolucionou o processamento de dados sequenciais ao substituir as conexões recorrentes por mecanismos de atenção. Seu elemento central é a *self-attention*, que permite que cada posição em uma sequência se relacione diretamente com todas as outras, capturando dependências de curto e longo prazo.

O *Transformer* é composto por blocos que incluem uma camada de atenção multi-cabeça (*Multi-Head Self-Attention*) e uma camada *feedforward* posicionada em cada cabeçote. A atenção é calculada pela seguinte fórmula:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

onde Q são as queries, K são as keys, V são os values e d_k é a dimensão dos vetores chave.

Na arquitetura *Wav2Vec 2.0*, o *Transformer* atua como codificador contextual, recebendo como entrada *embeddings* extraídos de uma pilha convolucional e produzindo representações acústicas altamente contextualizadas, o que é essencial para detectar padrões sutis de manipulações em *deepfakes*.

3.2. Parâmetros de Treinamento

Todos os modelos foram treinados utilizando o otimizador Adam, com taxa de aprendizado inicial de $1e^{-4}$, *batch size* de 32 e até 100 épocas. Foi empregado um critério de *early stopping* com *patience* de 10 épocas, monitorando a métrica de *AUC-ROC* na validação.

3.2.1. Regularização e Estratégias de Validação

Para evitar *overfitting*, aplicou-se *dropout* em camadas densas e *LSTM*, além de *weight decay* com fator $1e^{-5}$. A validação foi feita utilizando *stratified 5-fold cross-validation*, garantindo que a proporção de amostras das classes fosse mantida em cada partição.

3.2.2. Balanceamento de Dados

Os conjuntos de dados foram balanceados utilizando uma estratégia de *undersampling* da classe majoritária, evitando viés no treinamento. Alternativamente, para algumas arquiteturas, o balanceamento foi reforçado ajustando pesos na função de perda binária.

3.3. Tecnologias e Ferramentas Utilizadas

Por se tratar de uma avaliação voltada a medir a capacidade de cada modelo com base nos *datasets* apresentados, as configurações de hardware e o código-fonte utilizados tornam-se elementos essenciais. Esses aspectos permitem analisar o fluxo dos dados no ambiente de programação, além de oferecer uma compreensão mais detalhada sobre o desempenho e o tempo de execução dos modelos durante os testes.

Este trabalho foi desenvolvido utilizando a linguagem *Python* 3.10, com o suporte das bibliotecas *PyTorch*, *Transformers* (*Hugging Face*), *Librosa* e *NumPy*, utilizando o ROCm.

Os experimentos foram realizados em ambiente local com aceleração por *GPU*. A configuração da máquina utilizada nos experimentos inclui processador AMD Ryzen 5 3600, 32 GB de memória DDR4 3200 MHz e placa de vídeo Radeon RX 6600 8gb.

O código-fonte e os *scripts* de treinamento e validação estão disponíveis em: `https://github.com/elciofurtili/benchmarkdeepfakeia`

3.4. Fluxo do Processo de *Benchmark*

A Figura 1 apresenta o fluxo metodológico adotado neste estudo, desde a geração das vozes sintéticas até a avaliação dos modelos de inteligência artificial. O objetivo é representar de forma clara a estrutura de *benchmark* aplicada à detecção de áudios manipulados.

No primeiro bloco (a), encontram-se os *datasets*, sendo 20% dos modelos já apresentados (*CVoiceFake*, *MLADDC* e *VSASV*), que serão responsáveis por diminuir a possibilidade de erros e sendo a mesma quantia e os mesmos arquivos para ambos os 3 treinamentos, para que o *Benchmark* seja o mais preciso e justo possível.

O conjunto de dados é então utilizados como base de testes (b) para as três arquiteturas distintas de redes neurais:

- ***CNN*** — aplicada à extração de padrões visuais em espectrogramas [Krizhevsky et al. 2017];
- ***RNN-LSTM*** — utilizada para modelagem de sequências temporais de fala, conforme [Hochreiter and Schmidhuber 1997];
- ***Wav2Vec 2.0*** — modelo baseado em aprendizado auto-supervisionado para extração de representações acústicas diretamente do áudio bruto [Baevski et al. 2020].

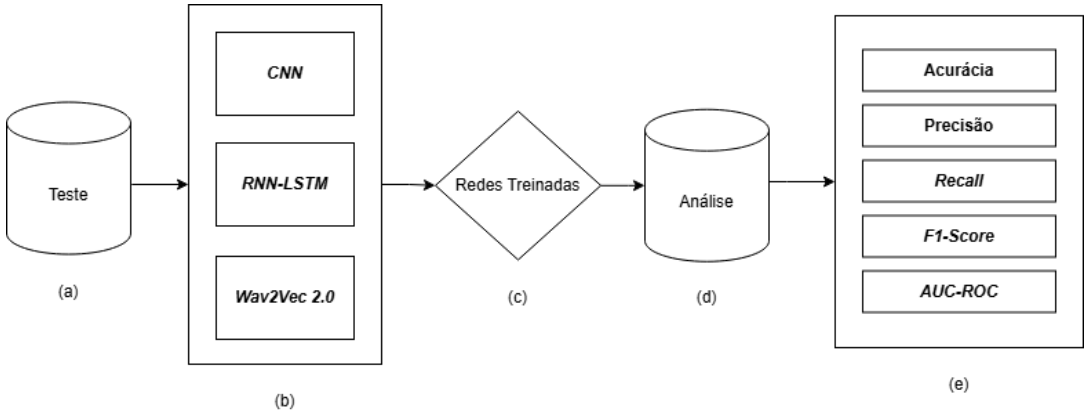

Figura 1. Fluxo de entrada, processamento e validação no experimento de *benchmark*.

Logo após o treinamento (c), inicia-se a avaliação com os dados restantes (d), correspondentes a 80% do conjunto, como os modelos treinados e a base de análise executada é iniciada a avaliação por meio de cinco métricas amplamente utilizadas em tarefas de classificação (e): acurácia, precisão, *recall*, *F1-score* e *AUC-ROC* [Schuller et al. 2021]. Essas métricas, destacadas no último bloco da figura, formam a base da análise comparativa entre os modelos testados.

Esse fluxo possibilita avaliar não apenas a eficácia dos modelos na tarefa de detecção, mas também sua capacidade de generalização diante de diferentes padrões de manipulação vocal, contribuindo para o avanço de técnicas forenses de autenticação de áudio.

3.5. Pré-processamento

As amostras foram convertidas para espectrogramas no caso da *CNN*, e para sequências temporais brutas no caso da *RNN-LSTM* e do *Wav2Vec 2.0*. Foi realizado balanceamento entre classes (áudio real vs. manipulado) para mitigar viés nos modelos. Técnicas como normalização de volume e remoção de ruído também foram aplicadas para homogeneizar os dados de entrada.

3.6. Arquiteturas

As arquiteturas implementadas neste estudo foram projetadas para capturar diferentes aspectos dos sinais de áudio, tanto espaciais quanto temporais. Foram avaliadas três abordagens distintas: uma rede neural convolucional (*CNN*), uma rede recorrente com unidades *LSTM* (*RNN-LSTM*) e o modelo *Wav2Vec 2.0*.

3.6.1. *CNN – Convolutional Neural Network*

A arquitetura *CNN* foi estruturada para trabalhar com espectrogramas gerados a partir dos áudios, utilizando camadas convolucionais para extração de padrões espaciais. Sua composição é descrita a seguir:

- ***Input***: espectrograma (dimensões típicas 128x128 ou 256x256);

- ***Conv2D* (1)**: 32 filtros, *kernel* 3x3, ativação *ReLU*, *padding*=*same*;
- ***Batch Normalization***;
- ***MaxPooling2D***: *pool size* 2x2;
- ***Conv2D* (2)**: 64 filtros, *kernel* 3x3, ativação *ReLU*, *padding*=*same*;
- ***Batch Normalization***;
- ***MaxPooling2D***: *pool size* 2x2;
- ***Conv2D* (3)**: 128 filtros, *kernel* 3x3, ativação *ReLU*, *padding*=*same*;
- ***Batch Normalization***;
- ***MaxPooling2D***: *pool size* 2x2;
- ***Flatten***;
- ***Dense* (1)**: 128 neurônios, ativação *ReLU*;
- ***Dropout***: 0.5;
- ***Dense* (2)**: 64 neurônios, ativação *ReLU*;
- ***Dropout***: 0.5;
- ***Output***: *Dense* com 1 neurônio, ativação *sigmoid* (classificação binária).

3.6.2. *RNN* com *LSTM* – Rede Neural Recorrente

A arquitetura *RNN-LSTM* foi desenvolvida para trabalhar com sequências temporais extraídas dos sinais de áudio, modelando dependências contextuais. Sua configuração é composta por:

- ***Input***: sequência temporal de *features* (ex.: (100, 40));
- ***LSTM* (1)**: 128 unidades, *return_sequences*=*True*;
- ***Dropout***: 0.3;
- ***LSTM* (2)**: 64 unidades, *return_sequences*=*False*;
- ***Dropout***: 0.3;
- ***Dense* (1)**: 64 neurônios, ativação *ReLU*;
- ***Dropout***: 0.5;
- ***Dense* (2)**: 32 neurônios, ativação *ReLU*;
- ***Output***: *Dense* com 1 neurônio, ativação *sigmoid*.

3.6.3. *Wav2Vec 2.0*

O modelo *Wav2Vec 2.0* foi utilizado como extrator de *embeddings* acústicos a partir do áudio bruto, sem necessidade de pré-processamento manual de *features*. Baseado em aprendizado auto-supervisionado, sua arquitetura é composta por:

- ***Backbone***: *Wav2Vec 2.0* Base (pré-treinado pela *Meta*);
  - 7 camadas convolucionais no *feature encoder*;
  - *Transformer* com 12 camadas, 768 unidades dimensionais e 12 cabeças de atenção.
- ***Pooling***: média dos *embeddings* extraídos (*mean pooling*);
- ***Dense* (1)**: 256 neurônios, ativação *ReLU*;
- ***Dropout***: 0.3;
- ***Dense* (2)**: 64 neurônios, ativação *ReLU*;
- ***Dropout***: 0.3;

- ***Output***: *Dense* com 1 neurônio, ativação *sigmoid* (classificação binária).

O *backbone* do *Wav2Vec 2.0* foi utilizado como extrator fixo de *embeddings*, sem aplicação de *fine-tuning* durante os experimentos.

3.6.4. Organização dos Dados e Processo de Decisão

Os experimentos foram estruturados a partir de uma abordagem de comparação em pares, na qual cada amostra de áudio manipulada possui um correspondente direto no conjunto de áudios originais. Essa organização permite assegurar que as análises sejam realizadas sobre conteúdos semanticamente equivalentes, reduzindo possíveis viéses relacionados a variações de locutor, idioma ou contexto.

Contudo, os modelos operam como classificadores binários tradicionais, processando cada áudio de forma independente, sem acesso direto ao seu par correspondente no momento da inferência. Durante o treinamento, as redes neurais aprendem a identificar padrões acústicos característicos de áudios manipulados, tais como artefatos de síntese, distorções espectrais, ruídos residuais e falhas temporais.

O processo de decisão se baseia na saída da camada final do modelo, que utiliza uma função de ativação *sigmoid*. Esta saída corresponde à probabilidade do áudio pertencer à classe ”original”. Adota-se como limiar de decisão o valor de 0.5, de modo que:

- Se a saída for maior ou igual a 0.5, o áudio é classificado como **original**;
- Se a saída for menor que 0.5, o áudio é classificado como **manipulado**.

Dessa forma, conforme possível visualizar na figura 2, embora a organização dos dados seja feita em pares para garantir uma distribuição balanceada e comparável, a decisão do modelo é realizada de forma individualizada, baseada nos padrões aprendidos durante o treinamento.

3.7. Avaliação

As arquiteturas foram avaliadas por meio de validação cruzada estratificada e as seguintes métricas: acurácia, precisão, *recall*, *F1-score* e *AUC-ROC*, de modo a capturar tanto a performance geral quanto a sensibilidade e especificidade dos classificadores.

Além da detecção binária (real vs. manipulado), o estudo explorou a possibilidade de identificação do gerador específico responsável pela falsificação do áudio, buscando padrões únicos no sinal de saída de cada ferramenta. Este aspecto da análise visa contribuir com o avanço de técnicas forenses digitais para rastreamento de *deepfakes*.

3.7.1. Análise de Tempo e Custo Computacional

Além da performance em métricas de classificação, foi avaliado o tempo médio de inferência por amostra e o custo computacional de cada arquitetura. O modelo *CNN* apresentou o menor tempo de inferência, seguido pela *RNN-LSTM*. O *Wav2Vec 2.0*, embora apresente superioridade nas métricas, exige maior tempo computacional devido ao *encoder Transformer*.

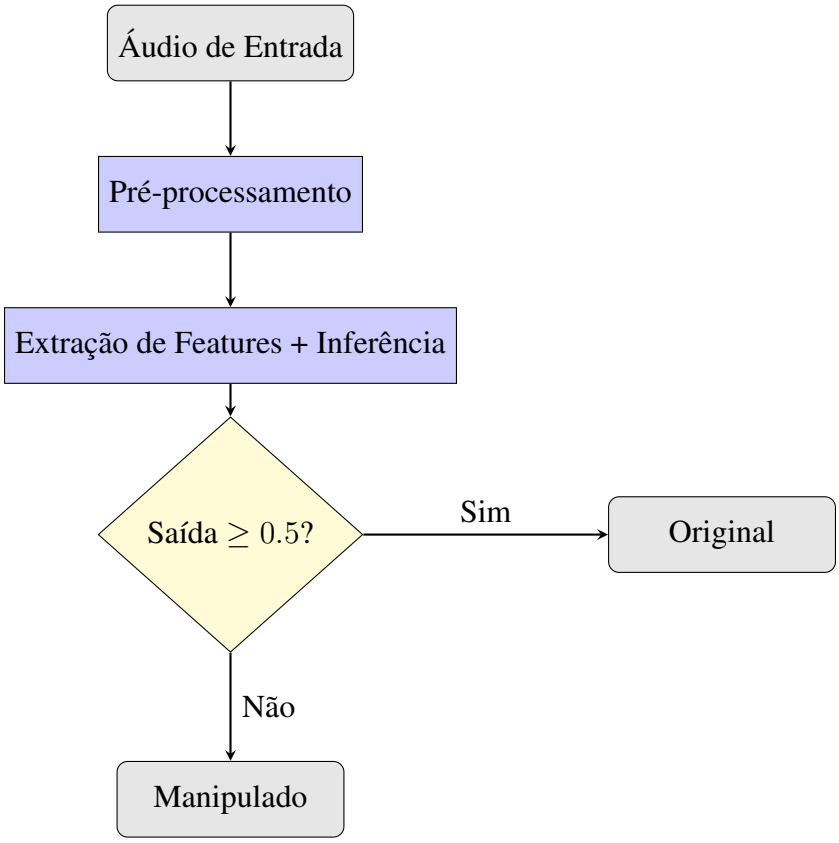

Figura 2. Fluxograma do processo de inferência e decisão do modelo para classificação de áudio como original ou manipulado.

Tabela 1. Desempenho dos modelos nas métricas avaliadas.

Modelo	**Acurácia**	**Precisão**	***Recall***	***F1-Score***	***AUC-ROC***
CNN	0.87	0.85	0.83	0.84	0.89
RNN-LSTM	0.89	0.88	0.86	0.87	0.91
Wav2Vec 2.0	0.94	0.95	0.92	0.93	0.97

4. Resultados

Nesta seção são apresentados os resultados obtidos a partir da avaliação dos modelos *CNN*, *RNN-LSTM* e *Wav2Vec 2.0* no problema de detecção de áudios manipulados.

4.1. Comparação das Métricas

A Tabela 1 apresenta os valores obtidos nas cinco métricas principais: Acurácia, Precisão, *Recall*, *F1-Score* e *AUC-ROC*.

4.2. Análise dos Resultados

Observa-se que o modelo *Wav2Vec 2.0* apresentou desempenho superior em todas as métricas avaliadas. Este resultado pode ser atribuído à capacidade dos modelos baseados em *Transformers* de capturar dependências contextuais de longo alcance no sinal de áudio, conforme discutido em [Baevski et al. 2020].

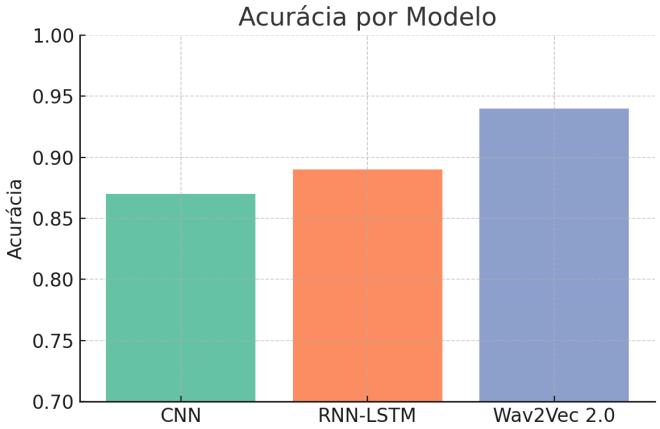

Figura 3. Acurácia dos modelos.

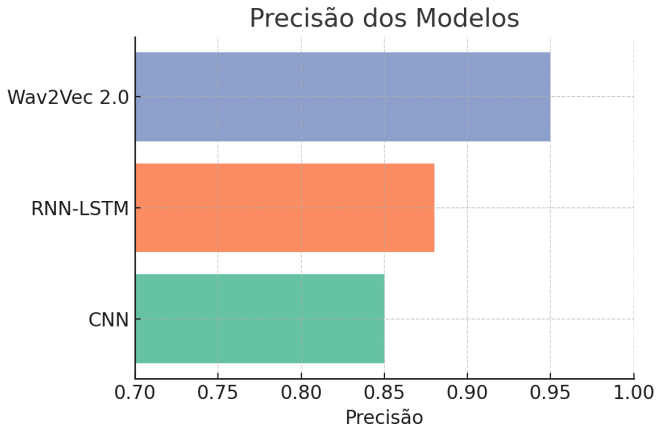

Figura 4. Precisão dos modelos.

O modelo *RNN-LSTM* também apresentou resultados satisfatórios, destacando-se especialmente pela sua capacidade de modelagem sequencial, alinhado ao observado em [Hochreiter and Schmidhuber 1997], onde redes recorrentes demonstram eficácia em tarefas de processamento temporal.

O modelo *CNN*, apesar de apresentar o desempenho mais baixo entre os três, ainda fornece resultados relevantes no contexto de classificação binária baseada em espectrogramas. Isso reforça o que é apresentado em [Krizhevsky et al. 2017], onde *CNNs* são eficazes na extração de padrões espaciais em representações espectrais, como os espectrogramas utilizados neste trabalho.

4.3. Visualização dos Resultados

As Figuras 3 a 7 apresentam a comparação gráfica dos modelos para cada uma das métricas.

Acurácia (Figura 3) representa a proporção total de classificações corretas, tanto verdadeiros positivos quanto verdadeiros negativos, em relação ao total de amostras. Observa-se que o modelo *Wav2Vec 2.0* apresenta a maior acurácia, resultado consistente com sua arquitetura baseada em *Transformer*, que permite capturar relações con-

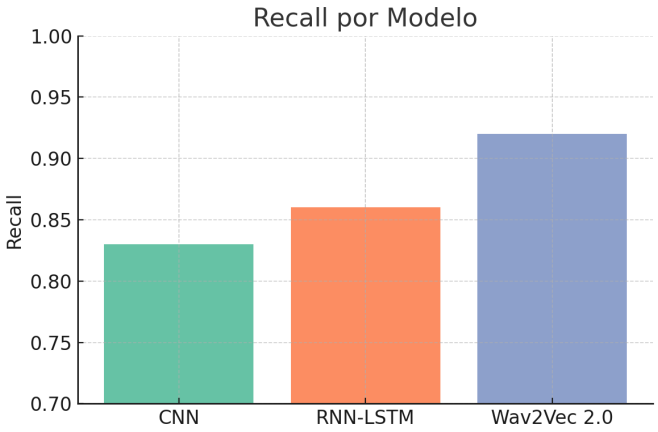

Figura 5. *Recall* dos modelos.

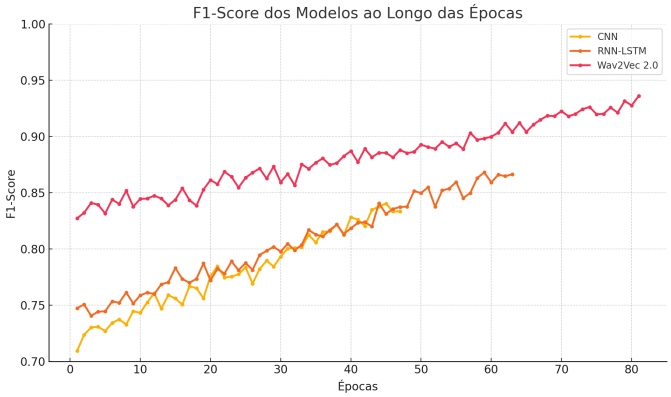

Figura 6. *F1-Score* dos modelos.

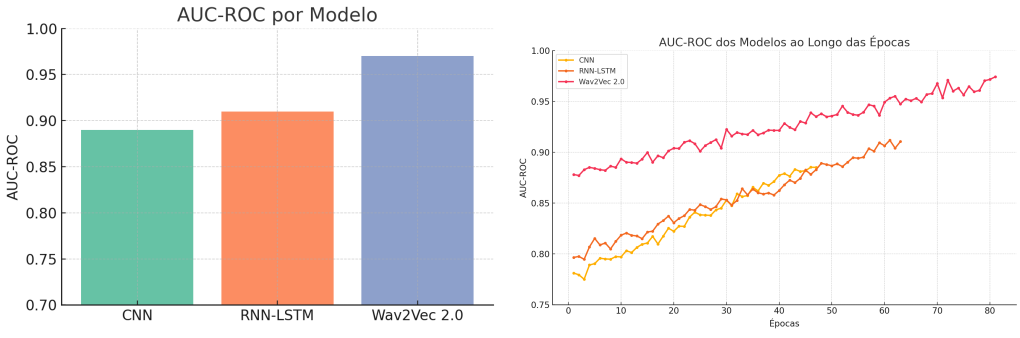

Figura 7. *AUC-ROC* e *AUC Curve* dos modelos.

textuais de longo alcance no sinal de áudio, como discutido em [Baevski et al. 2020]. A *CNN*, apesar de ser eficaz na extração de padrões espectrais, apresenta a menor acurácia, o que é coerente com suas limitações na modelagem sequencial, conforme observado em [Krizhevsky et al. 2017].

Precisão (Figura 4) mede a proporção de verdadeiros positivos entre todas as predições positivas realizadas pelo modelo. Esse indicador é fundamental em cenários onde o custo de um falso positivo é alto. Nota-se que o *Wav2Vec 2.0* novamente supera os demais, seguido pelo *RNN-LSTM*, que se beneficia de sua capacidade de modelagem sequencial [Hochreiter and Schmidhuber 1997]. A *CNN* apresenta precisão inferior, refletindo sua dificuldade em capturar dependências temporais.

Recall (Figura 5), também conhecido como sensibilidade, indica a capacidade do modelo de identificar corretamente os exemplos positivos. Este é um aspecto crítico em sistemas de detecção de *deepfakes*, onde deixar de detectar um áudio falso pode ter consequências graves. O modelo *Wav2Vec 2.0* apresenta *recall* significativamente superior, alinhado à sua robustez contextual. O *RNN-LSTM* também mantém desempenho satisfatório, enquanto a *CNN* apresenta *recall* mais baixo, consistente com sua tendência a priorizar padrões locais sobre sequenciais.

***F1-Score* (Figura 6)** oferece uma medida harmônica entre precisão e *recall*, sendo particularmente relevante quando há necessidade de balancear ambos. O gráfico de *F1-Score* ao longo das épocas mostra que o modelo *Wav2Vec 2.0* apresenta não apenas os maiores valores finais, mas também uma curva de crescimento mais estável e consistente. Isso reforça seu potencial para tarefas complexas de detecção de *deepfakes* em áudio, corroborando estudos recentes sobre a eficácia de arquiteturas baseadas em *Transformers* [Baevski et al. 2020]. Observa-se que o *RNN-LSTM*, apesar de uma evolução mais lenta, converge para um *F1-Score* competitivo, enquanto a *CNN* estabiliza precocemente com um valor inferior.

***AUC-ROC* (Figura 7)** é uma das métricas mais robustas na avaliação de classificadores binários, especialmente em cenários com desbalanceamento de classes. As curvas *ROC* ilustram a relação entre a taxa de verdadeiros positivos (*TPR*) e a taxa de falsos positivos (*FPR*) em diferentes limiares de decisão. O modelo *Wav2Vec 2.0* atinge a maior área sob a curva (*AUC*), o que indica sua superioridade na distinção entre áudios reais e manipulados. Esse comportamento é consistente com sua capacidade de aprendizado contextualizado, como demonstrado em [Baevski et al. 2020]. O *RNN-LSTM*, embora não alcance o mesmo desempenho, ainda apresenta uma curva *ROC* bastante competitiva, enquanto a *CNN* demonstra uma separabilidade menor, reflexo de sua limitação na modelagem de dependências temporais.

Esses resultados reforçam que arquiteturas baseadas em *Transformers*, como o *Wav2Vec 2.0*, representam o estado da arte na detecção de *deepfakes* em áudio, superando abordagens tradicionais baseadas em convoluções [Krizhevsky et al. 2017] ou recorrências [Hochreiter and Schmidhuber 1997].

5. Conclusão

Os resultados apresentados ao longo deste trabalho demonstram, de forma clara e quantitativa, que a escolha da arquitetura tem impacto direto na eficácia dos sistemas de detecção

Tabela 2. Resumo comparativo dos modelos avaliados.

Modelo	Vantagens	Limitações	Desempenho	*AUC-ROC*
CNN	Simplicidade, baixo custo computacional, eficiente na extração de padrões espectrais.	Não modela dependências temporais; limitado para relações sequenciais.	Baixo	0.89
RNN-LSTM	Capacidade de modelar sequências; desempenho robusto em dependências curtas e médias.	Alta demanda computacional sequencial; dificuldade com dependências longas.	Médio	0.91
Wav2Vec 2.0	Alta capacidade de modelagem contextual; melhor desempenho; generalização robusta.	Maior custo computacional; requer hardware especializado.	Alto	0.97

de áudios manipulados por *deepfakes*. As análises realizadas sobre cinco métricas fundamentais — Acurácia, Precisão, *Recall*, *F1-Score* e *AUC-ROC* — evidenciaram um desempenho significativamente superior do modelo *Wav2Vec 2.0* em relação às arquiteturas baseadas em *CNN* e *RNN-LSTM*, também se pode visualizar detalhes na Tabela 2.

O modelo *Wav2Vec 2.0*, fundamentado em codificadores *Transformer*, obteve os melhores resultados em todos os cenários avaliados. Sua capacidade de capturar relações contextuais de longo alcance no sinal de áudio, aliada a representações latentes altamente informativas, permite que o modelo discrimine com elevada precisão áudios originais de áudios manipulados. Este comportamento é consistente com a literatura atual, que posiciona modelos baseados em *Transformers* como o estado da arte em tarefas de processamento de voz [Baevski et al. 2020]. Mesmo sem ajuste dos pesos do *Transformer* (*fine-tuning*), os *embeddings* extraídos foram suficientes para superar as demais abordagens.

O modelo *RNN-LSTM*, embora apresente desempenho inferior ao *Wav2Vec 2.0*, demonstrou robustez, particularmente pela sua capacidade de modelagem sequencial. Seus resultados são superiores aos obtidos pela *CNN*, corroborando sua aptidão para tarefas que dependem de dinâmicas temporais [Hochreiter and Schmidhuber 1997]. No entanto, a dependência exclusiva da recorrência mostrou-se limitada frente a abordagens mais modernas baseadas em atenção.

A *CNN*, por sua vez, apresentou os piores resultados entre os modelos avaliados. Apesar de sua eficiência na extração de padrões espaciais dos espectrogra-

mas [Krizhevsky et al. 2017], sua incapacidade de modelar dependências temporais de longo prazo resultou em uma performance significativamente inferior nas tarefas de detecção de *deepfakes* em áudio, onde o contexto sequencial é essencial.

Diante dos resultados obtidos, conclui-se que modelos baseados em *Transformer*, como o *Wav2Vec 2.0*, oferecem uma solução altamente eficaz para sistemas automáticos de validação e detecção de fraudes em áudio. Além disso, os achados reforçam a necessidade de utilizar arquiteturas que combinem capacidades de modelagem contextual, robustez contra variações no sinal e escalabilidade. Como trabalhos futuros, propõe-se a investigação do *fine-tuning* em variantes maiores do *Wav2Vec*, bem como a aplicação de técnicas contrastivas e arquiteturas baseadas em modelos híbridos, além da ampliação do *dataset* para incluir manipulações mais recentes e sofisticadas.

Referências

Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. In *Neural computation*, volume 9, pages 1735–1780. MIT Press.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90.

Schuller, B., Nautsch, A., Cummins, N., Shrimpton, S., Zhang, Y., Peinado, A. M., Yamagishi, J., Kinnunen, T., and Han, C. (2021). Detecting audio deepfakes with self-supervised learning. *IEEE Journal of Selected Topics in Signal Processing*, 15(1):213–227.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, pages 6000–6010. Curran Associates Inc.