DIMITRI LEANDRO DE OLIVEIRA SILVA
MATHEUS COSTA DAMASCENO

APLICAÇÃO DE REDES NEURAIS PARA EMULAÇÃO DE EFEITOS SONOROS

Santo André, SP, Brasil

2025

DIMITRI LEANDRO DE OLIVEIRA SILVA
MATHEUS COSTA DAMASCENO

APLICAÇÃO DE REDES NEURAIS PARA EMULAÇÃO DE EFEITOS SONOROS

Trabalho de Graduação apresentado ao curso de Engenharia de Informação da Universidade Federal do ABC como parte dos requisitos para obtenção do grau de Engenheiro de Informação.

Universidade Federal do ABC – UFABC

Centro de Engenharia, Modelagem e Ciências Sociais Aplicadas — CECS

Graduação em Engenharia de Informação

Orientador: RICARDO SUYAMA

Santo André, SP, Brasil

2025

Resumo

O presente documento relata o desenvolvimento do Trabalho de Graduação (TG) em Engenharia de Informação da Universidade Federal do ABC (UFABC) dos autores. O projeto, dividido em duas partes, empregou Redes Neurais Artificiais (RNA) para simular efeitos sonoros, visando proporcionar uma solução acessível para a produção de materiais musicais. Na primeira fase do projeto, seis efeitos sonoros determinísticos foram aplicados (*reverb*, distorção, compressão, filtro passa-baixas, filtro passa-altas e *chorus*) em um conjunto de dados público (*Guitar Chords v2*), com o objetivo de comparar de duas técnicas de simulação: Convolução da Resposta ao Impulso (*Impulse Response*, IR) e uma abordagem que combinou *Convolutional Neural Networks* (CNNs) e *Long Short Term Memory* (LSTM) para arquitetar uma RNA. Os resultados da primeira parte mostraram que, apesar de ter atingido resultados significativos, a RNA não conseguiu superar a emulação por IR. Na segunda etapa do projeto, os autores realizaram uma gravação de guitarra de 34 minutos, utilizando uma interface de áudio com conversor analógico-digital (*Analog-to-Digital Converter*, ADC). O sinal passou por uma cadeia de quatro efeitos em série (equalizador, compressor, simulador de amplificador e reverberação – nessa ordem), e uma nova RNA foi treinada, buscando simular os efeitos aplicados. Nesta etapa, a rede empregada conseguiu superar a emulação por IR na métrica de Similaridade do Cosseno dos Coeficientes Mel Cepstrais (MFCC-COS), mas não quando a métrica avaliada foi o Erro Médio Quadrático (*Mean Squared Error*, MSE). A redação deste documento elucida os detalhes dos efeitos aplicados em cada etapa, seus hiper-parâmetros, as metodologias adotadas no desenvolvimento das RNAs, e seus resultados.

Palavras-chave: emulação de efeitos sonoros, redes neurais artificiais, resposta ao impulso, processamento digital de sinais.

Abstract

This document reports the development of the Undergraduate Thesis (TG) in Information Engineering at the Federal University of ABC (UFABC) by the authors. The project, divided into two parts, employed Artificial Neural Networks (ANNs) to simulate sound effects, aiming to provide an accessible solution for music production. In the first phase of the project, six deterministic audio effects were applied (reverb, distortion, compression, low-pass filter, high-pass filter, and chorus) to a public dataset (Guitar Chords v2), with the goal of comparing two simulation techniques: Impulse Response (IR) convolution and an approach that combined Convolutional Neural Networks (CNNs) with Long Short-Term Memory (LSTM) to design an ANN. The results from the first phase showed that, although the ANN achieved significant outcomes, it did not outperform the IR-based emulation. In the second phase of the project, the authors recorded a 34-minute guitar session using an audio interface with an Analog-to-Digital Converter (ADC). The signal was processed through a chain of four effects in series (equalizer, compressor, amplifier simulator, and reverb — in that order), and a new ANN was trained to emulate the applied effects. In this stage, the neural network outperformed IR-based emulation in the Mel-Frequency Cepstral Coefficients Cosine Similarity (MFCC-COS) metric, but not when evaluated using the Mean Squared Error (MSE). This document details the effects applied in each stage, their hyperparameters, the methodologies adopted for ANN development, and the corresponding results.

Keywords: sound effects emulation, artificial neural networks, impulse response, digital signal processing.

Lista de abreviaturas e siglas

ADC	*Analog-to-Digital Converter*
CNN	*Convolutional Neural Network*
DAW	*Digital Audio Workstation*
DCT	*Discrete Cosine Transform*
FFT	*Fast Fourier Transform*
GRU	*Gated Recurrent Unit*
IR	*Impulse Response*
LSTM	*Long Short-Term Memory*
MFCC	*Mel-Frequency Cepstral Coefficients*
MFCC-COS	*Mean Cosine Distance of the Mel-Frequency Cepstral Coefficients*
MNA	*Modified Nodal Analysis*
MSE	*Mean Squared Error*
ReLU	*Rectified Linear Unit*
RNA	Rede Neural Artificial

RMSE	*Root Mean Squared Error*
SCNN	*Simple Convolutional Neural Network*
SLIT	Sistema Linear e Invariante no Tempo
TCN	*Temporal Convolutional Network*
WAV	*Waveform Audio File Format*
WDF	*Wave Digital Filter*

Sumário

Lista de ilustrações

Lista de tabelas

1 Introdução

A utilização de efeitos sonoros na indústria musical remonta há bastante tempo. Os efeitos são empregados para gerar uma ampla gama de texturas e sensações que aprimoram a experiência auditiva do ouvinte, acrescentando camadas de profundidade, complexidade e dinamismo às composições musicais. Eles podem assumir várias formas, desde reverberações e *delays* simples até efeitos mais complexos, como *flangers*, *phasers* e *pitch shifters*, sendo aplicáveis a vários elementos da música, como vocais, guitarras, bateria, e muitos outros instrumentos (ZÖLZER, 2011).

Nesse sentido, os equipamentos de som são essenciais para a criação e produção de material musical, seja em um estúdio de gravação profissional ou em um *home studio*. No entanto, esses equipamentos podem ser extremamente caros, dificultando o acesso de muitos músicos e produtores e impedindo a criação de material musical de alta qualidade. Os preços dos equipamentos de som podem variar bastante, desde equipamentos básicos de entrada até equipamentos sofisticados de alta qualidade. Por exemplo, uma interface de áudio – dispositivo responsável por converter sinais analógicos, como o som de uma guitarra ou microfone, em sinais digitais compreensíveis pelo computador – pode variar de poucas centenas até dezenas de milhares de reais. As Figuras 1 e 2 mostram duas capturas de tela que exemplificam essa afirmação. O mesmo ocorre com outros equipamentos de

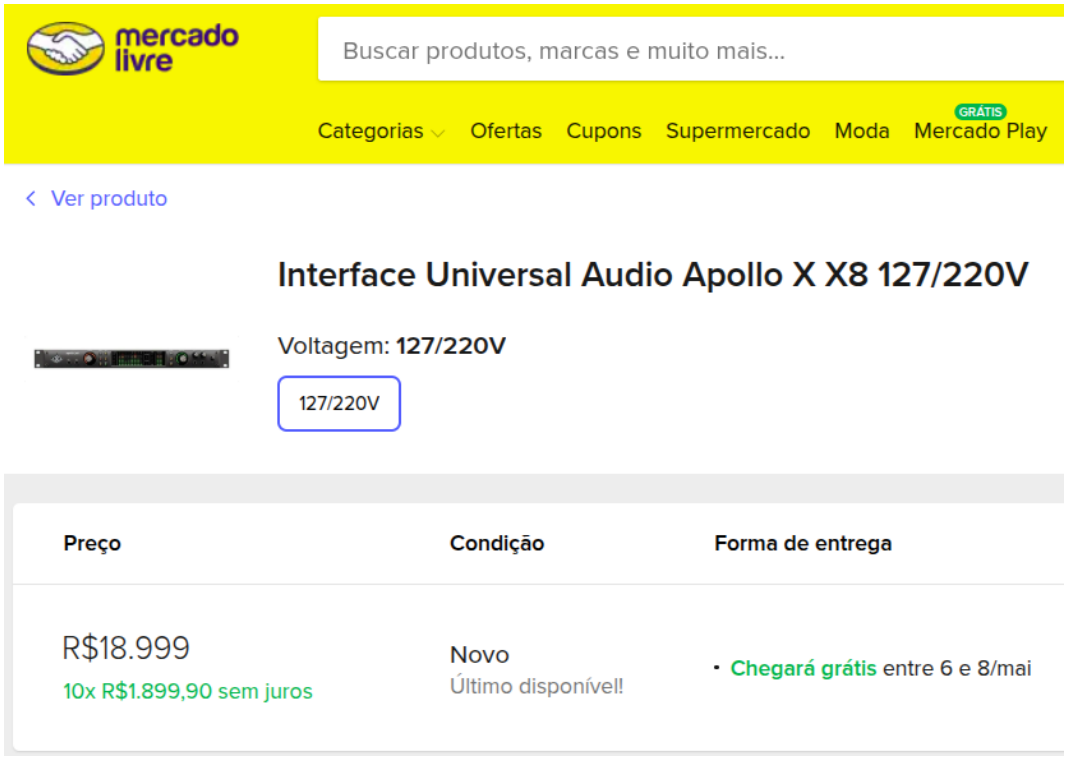
Figura 1 – Interface de áudio com conversor analógico-digital custando R$ 119,90. Acesso em abril de 2025.

Fonte: Os autores.

som, como microfones, amplificadores, processadores de efeitos e alto-falantes. Os altos preços podem ser um grande obstáculo para músicos e produtores iniciantes. Além disso, mesmo para músicos e produtores experientes, a compra de novos equipamentos pode ser uma despesa significativa, especialmente quando se trata de equipamentos de som de alta qualidade.

Visando baratear o processo de criação musical, a emulação de efeitos sonoros é uma técnica que permite reproduzir digitalmente os efeitos de equipamentos e instrumentos musicais clássicos, como

Figura 2 – Interface de áudio com conversor analógico-digital custando R$ 18.999,00. Acesso em abril de 2025.



Fonte: Os autores.

pedais de guitarra, sintetizadores e amplificadores, em um *software* de produção musical digital (*Digital Audio Workstation*, DAW). Isso pode ser uma solução econômica para os músicos que desejam utilizar efeitos sonoros em suas composições, mas não têm acesso a equipamentos ou instrumentos físicos, que podem ser muito caros. Ao emular um efeito sonoro digitalmente, é possível obter resultados muito próximos daqueles obtidos com o equipamento original, o que torna essa técnica uma opção viável para produzir música em um orçamento limitado. Além disso, a emulação de efeitos sonoros

pode economizar espaço físico no estúdio de gravação, pois não há necessidade de ter vários pedais ou equipamentos para criar diferentes efeitos sonoros. Outra vantagem significativa é a diversidade criativa proporcionada: no ambiente computacional, é possível armazenar, carregar e modificar virtualmente centenas de configurações de efeitos, algo inviável no ambiente físico tradicional, devido a limitações de espaço, custo e flexibilidade de hardware. A possibilidade de automatização e integração com fluxos de produção digital também contribui para a adoção crescente dessas soluções por músicos profissionais e amadores.

Uma das técnicas mais comuns e difundidas para a emulação de efeitos sonoros é a da Convolução da Resposta ao Impulso. Essa abordagem permite emular com precisão o som de equipamentos e espaços acústicos reais (OPSTAL, 2016). A resposta ao impulso (*Impulse Response*, IR) de um sistema é a sua saída quando submetido a um impulso unitário – ou seja, um sinal de duração muito curta e alta amplitude. Ela captura as características essenciais do sistema, como reverberação, atenuações em certas faixas de frequência e o comportamento de reflexões acústicas. A convolução, por sua vez, é uma operação matemática fundamental no processamento de sinais, utilizada para modelar como um Sistema Linear e Invariante no Tempo (SLIT) responde a uma entrada arbitrária. No domínio discreto, a convolução de dois sinais $x[n]$ (entrada) e $h[n]$ (resposta ao impulso do sistema) é definida como:

$$y[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k] \tag{1.1}$$

Essa operação representa uma soma ponderada entre os valores do sinal de entrada e uma versão invertida e deslocada da resposta ao impulso, acumulando os efeitos ao longo do tempo. Na emulação de efeitos sonoros, o sinal $x[n]$ representa tipicamente um áudio puro (sem efeito - *dry*), enquanto $h[n]$ corresponde à resposta ao impulso do equipamento ou espaço acústico real que se deseja emular. A convolução desses dois sinais gera uma simulação precisa das características do sistema original, permitindo reproduzir aspectos como reverberação, ressonâncias e resposta em frequência (OPSTAL, 2016).

Isso pode ser particularmente útil para músicos e produtores que desejam criar um som específico, como a emulação de um determinado amplificador, a acústica de uma sala de concerto ou um estúdio de gravação específico. Essa técnica requer um *software* de convolução, que combina o sinal de áudio puro com a resposta ao impulso para criar o sinal emulado. É possível encontrar pacotes de respostas ao impulso de equipamentos e espaços acústicos famosos em sites especializados e plugins de DAWs, ou até mesmo gravar as próprias respostas ao impulso para emular o som desejado. A Figura 3 mostra um exemplo para essa afirmação. Na figura, exibe-se o *plugin* gratuito *MConvolutionEZ* com uma série de arquivos de resposta em frequência que possibilitam emular diferentes reverberações.

No entanto, a emulação de efeitos sonoros pode ser uma tarefa

Figura 3 – Plugin de reverberação MConvolutionEZ exibindo arquivos *.flac* de resposta ao impulso. Disponível em: https://www.meldaproduction.com/MConvolutionEZ.

Fonte: Os autores.

complexa, uma vez que cada efeito tem suas próprias características e particularidades. Além disso, a produção de efeitos sonoros requer muitas vezes o uso de equipamentos e instrumentos específicos, o que pode tornar o processo ainda mais difícil e, principalmente, caro para quem produz a emulação. Nesse sentido, o uso de técnicas de processamento digital de sinais em conjunto com Redes Neurais Artificiais (RNAs) pode oferecer uma solução promissora para a emulação de efeitos sonoros (COVERT; LIVINGSTON, 2013). Essas técnicas permitem a criação de modelos matemáticos que representam

os sinais sonoros, possibilitando sua manipulação e emulação por meio de algoritmos computacionais. Além disso, as RNAs podem ser treinadas para identificar e reproduzir com precisão os padrões e características dos efeitos sonoros, oferecendo resultados satisfatórios para os produtores musicais, artistas e ouvintes (RAMÍREZ; REISS, 2019).

O objetivo deste projeto foi desenvolver RNAs capazes de simular diversos efeitos sonoros utilizados no ramo da produção musical. A primeira fase do projeto buscou simular seis efeitos sonoros determinísticos (*reverb*, distorção, compressão, filtro passa-baixas, filtro passa-altas e *chorus*), comparando as técnicas de convolução da resposta ao impulso e uma RNA específica para cada efeito, buscando entender as vantagens e limitações de cada abordagem nas emulações. foi conduzido um experimento prático: os autores realizaram a gravação de uma sessão de guitarra com duração de 34 minutos, utilizando uma interface de áudio equipada com um Conversor Analógico-Digital (*Analog-to-Digital Converter*, ADC). O áudio capturado passou por uma cadeia de quatro efeitos em série aplicados digitalmente no *software Cakewalk* – equalização, compressão, simulação de amplificador com distorção, e reverberação, na ordem – de forma a replicar um cenário realista encontrado em estúdios. A partir desse material, foi treinada uma RNA com o objetivo de emular digitalmente a cadeia completa de efeitos aplicada.

A justificativa para o uso de redes neurais está no fato de que,

ao contrário de métodos baseados em convolução, que são adequados principalmente para sistemas lineares e invariantes no tempo, as RNAs são capazes de aprender relações complexas e não lineares entre sinais de entrada e saída. Isso as torna especialmente promissoras para modelar efeitos sonoros com comportamento dinâmico ou dependência de contexto, como distorções e modulações, por exemplo.

Por fim, este trabalho está organizado da seguinte forma: o Capítulo 2 apresenta a revisão bibliográfica; o Capítulo 3 descreve detalhadamente a metodologia adotada; o Capítulo 4 discute os resultados obtidos; e o Capítulo 5 apresenta as considerações finais e sugestões para trabalhos futuros.

2 Revisão Bibliográfica

No que diz respeito à emulação de efeitos sonoros, a modelagem pode ser realizada de diversas formas por meio técnicas de processamento digital de sinais. Essas diferentes técnicas podem ser agrupadas de acordo com o nível de disponibilidade dos parâmetros do modelo utilizado para gerar tal emulação, podendo ser classificadas como *white-box*, *black-box* e *gray-box*.

Os métodos *white-box*, consistem em criar um modelo matemático capaz de representar o circuito do dispositivo físico que gera o efeito. Dos exemplos de métodos que tratam o problema por meio dessa abordagem cabe citar a Análise Nodal Modificada (*Modified Nodal Analysis*, MNA), e o uso de Filtros Digitais de Onda (*Wave Digital Filters*, WDF). Estes métodos exigem uma análise detalhada do circuito do amplificador e buscam resolver o sistema – que em geral é baseado em equações diferenciais – de maneira iterativa, o que costuma exigir um grande custo computacional e apresentar imprecisões, levando a distorções indesejadas no sinal de emulado (VANHATALO et al., 2022).

Já os modelos *gray-box*, como o próprio nome sugere, representam uma abordagem intermediária entre os métodos *white-box* e *black-box*. Esses modelos são baseados em uma combinação de conhecimento do circuito do dispositivo físico e análise empírica dos

sinais de entrada e saída. Dessa forma, é possível construir um modelo matemático que leve em consideração as características do circuito, mas sem a necessidade de uma análise tão detalhada quanto nos métodos *white-box* (VANHATALO et al., 2022).

Por fim, os métodos *black-box* consistem em modelar o sistema que gera o efeito sonoro apenas a partir da relação entre o sinal de entrada - sem o efeito - e o sinal de saída com o efeito aplicado. Esses modelos não exigem conhecimento prévio sobre o circuito físico do dispositivo que gera o efeito sonoro, tornando seu uso mais prático. Entre as técnicas de modelagem *black-box*, destacam-se a Convolução Dinâmica, que é útil para emular reverberações e *delays*; e as Redes Neurais Artificiais, que se mostram eficientes na emulação de diversos tipos de efeitos sonoros, como distorções, modulações e reverberações (RAMÍREZ; BENETOS; REISS, 2020).

Para emular um efeito sonoro por meio de redes neurais, uma abordagem comum envolve o uso de aprendizado supervisionado, no qual a rede é treinada com pares de exemplos de áudio com e sem o efeito desejado. Nessa configuração, a rede aprende a mapear sinais de entrada em sinais de saída que emulam o comportamento do efeito sonoro (VANHATALO et al., 2022). Por exemplo, para emular um *delay*, a rede seria treinada com exemplos de áudio de entrada sem *delay* e áudio de saída com *delay*, simulando o efeito de uma gravação com repetições acústicas. A rede neural é então capaz de aprender os padrões e características dos sinais de entrada e saída para replicar o

efeito desejado (COVERT; LIVINGSTON, 2013). No entanto, vale destacar que também existem abordagens alternativas que buscam reduzir a dependência de grandes volumes de dados, como o *Few-Shot Learning*, *One-Shot Learning* e *Zero-Shot Learning*, além de métodos baseados em aprendizado não supervisionado, semi-supervisionado ou por reforço. Cada uma dessas estratégias apresenta diferentes requisitos de dados e arquiteturas, e sua aplicabilidade depende do tipo de efeito a ser modelado e da disponibilidade de dados rotulados.

A literatura retrata diversos estudos que fazem uso de redes neurais artificiais de variados tipos para modelagem de circuitos de efeitos sonoros. (WRIGHT et al., 2019) e (PEUSSA et al., 2020), por exemplo, propõem o uso de redes recorrentes, baseadas em *Long Short-Term Memory* (LSTM) e Gated Recurrent Unit (GRU). Já outros autores abordam este problema por meio de redes convolucionais, incluindo arquiteturas tais como *Simple Convolutional Neural Networks* (SCNN) com *pooling* (SCHMITZ, 2019), *FeedForward WaveNet* (WRIGHT et al., 2020a; DAMSKÄGG et al., 2019) e *Temporal Convolutional Networks* (TCN) (STEINMETZ; REISS, 2021). Também são retratados os usos de redes híbridas, que combinam um ou mais tipos de redes, tal como o uso de redes LSTM convolucionais e redes recorrentes convolucionais (SCHMITZ, 2019).

Posterior à geração das amostras emuladas, há a necessidade da utilização de métricas comparativas que permitam avaliar a qualidade e desempenho dos sistemas emuladores desenvolvidos. Para tal, foi

adotado o uso dos coeficientes cepstrais na escala Mel (*Mel-Frequency Cepstral Coefficients*, MFCCs), que são amplamente utilizados em tarefas de reconhecimento de fala, classificação de gênero musical, identificação de instrumentos e diversas outras aplicações de processamento de áudio. Seu objetivo é representar a forma espectral de um sinal de maneira compacta, aproximando a percepção auditiva humana. O cálculo dos MFCCs envolve as seguintes etapas: inicialmente, o sinal de áudio é dividido em janelas de tempo curtas (tipicamente 20–40 ms), e em cada uma delas é aplicada a Transformada Rápida de Fourier (*Fast Fourier Transform*, FFT), produzindo o espectro de magnitude. Em seguida, aplica-se um banco de filtros triangulares espaçados de acordo com a escala Mel, que mapeia frequências lineares (Hz) para a percepção logarítmica do ouvido humano, definida aproximadamente por:

$$\text{mel}(f) = 2595 \cdot \log_{10}\left(1 + \frac{f}{700}\right) \tag{2.1}$$

Após o mapeamento, calcula-se o logaritmo da energia em cada banda filtrada. Por fim, aplica-se a Transformada Discreta do Cosseno (*Discrete Cosine Transform*, DCT) sobre os vetores log para obter os coeficientes cepstrais, resultando em uma representação compacta e descorrelacionada das características espectrais do sinal. Geralmente, apenas os primeiros 12 a 20 coeficientes são utilizados, pois carregam a maior parte da informação relevante para análise perceptual (ZHENG; ZHANG; SONG, 2001).

Nesse sentido, o projeto proposto se beneficiou da utilização dos MFCCs para obter a métrica *Mean Cosine Distance of the Mel-Frequency Cepstral Coefficients* (MFCC-COS), que é uma métrica utilizada para avaliar a qualidade de sinais de áudio, especialmente na comparação entre um sinal original e outro processado (RAMÍREZ; BENETOS; REISS, 2020). O cálculo da MFCC-COS segue duas etapas, começando na transformação dos sinais de áudio em MFCCs conforme descrito anteriormente, e terminando no cálculo da distância média do cosseno entre os coeficientes de cada sinal. Essa medida é importante para comparar sinais, pois a distância do cosseno, definida por

$$\cos(\theta) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \, \|\mathbf{b}\|} = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2} \sqrt{\sum_{i=1}^{n} b_i^2}} \tag{2.2}$$

é uma medida de similaridade que mede o grau de alinhamento entre dois vetores a e b, independente de sua magnitude. Um valor de 0 indica que os vetores são totalmente diferentes (ortogonais), enquanto 1 significa que são idênticos. Portanto, a MFCC-COS pode ser utilizada para comparar a qualidade de diferentes emulações, bastando aplicá-la entre os sinais emulados e o sinal obtido após a aplicação do efeito original.

É importante destacar a existência de outras métricas mais convencionais que são amplamente utilizadas para avaliar a qualidade de sinais de áudio (BäCKSTRöM, 2013), tais como o Erro Quadrático Médio (*Mean Squared Error*, MSE) e a Raiz do Erro Quadrático Médio

(*Root Mean Squared Error*, RMSE), definidas matematicamente por

$$\text{MSE} = \frac{1}{N} \sum_{n=1}^{N} \left(y[n] - \hat{y}[n] \right)^2 \tag{2.3}$$

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{n=1}^{N} \left(y[n] - \hat{y}[n] \right)^2} \tag{2.4}$$

onde $y[n]$ e $\hat{y}[n]$ os sinais de referência e predição, respectivamente. Essas métricas tradicionais focam na comparação direta das formas de onda entre um sinal original e um sinal processado, calculando a diferença média quadrática ou a raiz quadrada dessa diferença. No entanto, é crucial observar que o emprego dessas métricas pode demandar que a rede neural reproduza o sinal de forma precisa, o que pode não ser sempre o objetivo desejado, especialmente em tarefas de emulação de efeitos sonoros. Em contraste, métricas mais orientadas para a percepção auditiva, como a MFCC-COS, enfatizam a similaridade perceptual entre sinais de áudio, ao invés de apenas a precisão na forma de onda.

3 Metodologia

Esta seção tem como objetivo descrever a metodologia empregada durante o desenvolvimento do projeto, dividindo-se em duas partes distintas. Na primeira parte, foram aplicados seis efeitos determinísticos a um conjunto de dados disponibilizado publicamente, e as emulações da IR foram comparadas com aquelas obtidas por meio da RNA. A seção abordará detalhadamente o conjunto de dados utilizado, os efeitos aplicados, seus parâmetros e configurações, além das métricas empregadas na comparação. Também será descrito o desenvolvimento da RNA, que foi utilizada para emular cada um dos efeitos aplicados. Na segunda parte, a seção descreverá as configurações da gravação do sinal de guitarra, incluindo os efeitos aplicados a esse sinal, seus parâmetros e configurações, e também o detalhamento da elaboração e desenvolvimento da RNA utilizada para emular os efeitos aplicados.

Vale ressaltar que todos os códigos desenvolvidos encontram-se disponíveis publicamente no GitHub[1].

3.1 Parte I

A primeira parte do projeto teve como foco a emulação de efeitos sonoros isolados, aplicados sobre um conjunto de dados públicos de

[1] https://github.com/DimitriLeandro/AI-Sound-FX-Emulation

áudios de guitarra. Nessa etapa, buscou-se comparar a eficácia de duas abordagens distintas de emulação – a Convolução da Resposta ao Impulso e Redes Neurais Artificiais – por meio da aplicação de seis efeitos determinísticos.

3.1.1 Dataset

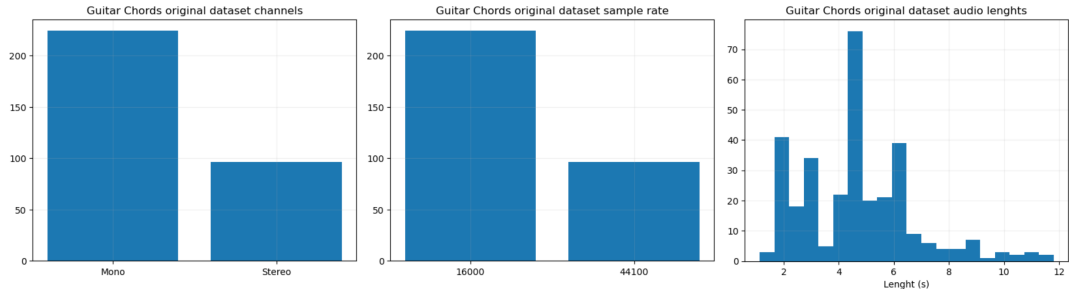
O *dataset* utilizado neste projeto é intitulado Guitar Chords Dataset v2[2]. Originalmente, este conjunto de dados foi criado para um projeto universitário de Machine Learning com o objetivo prever os acordes de guitarra tocados por um instrumento real. Os acordes no *dataset*, gravados em 320 áudios WAV, são: Am, Bb, Bdim, C, Dm, Em, F, G. Eles são tocados de quatro maneiras diferentes em relação à forma de usar a palheta: **1)** para cima e para baixo; **2)** para baixo e para cima; **3)** para cima e para baixo para baixo e para cima; e **4)** corda por corda. Os acordes são tocados por diferentes autores e em 20 guitarras e violões distintos.

A Figura 4 mostra a configuração dos áudios no *dataset*. Verifica-se uma falta de padronização em relação à quantidade de canais, amostragem e duração. Por essa razão, antes de seguir com qualquer outro passo da metodologia, os autores padronizaram os áudios com uma etapa de pré-processamento, que consistiu em:

- **Quantidade de canais:** todos os 320 áudios do *dataset* foram convertidos para mono, com apenas um canal;

[2] https://www.kaggle.com/datasets/fabianavinci/guitar-chords-v2

Figura 4 – Configurações do dataset original, apresentando a distribuição dos canais presentes no dataset, a frequência de amostragem em Hertz e a duração dos áudios em segundos.



Fonte: Os autores.

- **Taxa de amostragem:** todos os áudios foram convertidos para a taxa de 16 kHz de amostragem;
- **Profundidade de bits:** todos os áudios foram convertidos para PCM 16;
- **Fade out:** aplicou-se um fade out linear de 0.5 segundos ao final de cada áudio;
- **Adição de silêncio:** 3 segundos de silêncio foram adicionados ao final de cada áudio com o objetivo de permitir que o decaimento dos efeitos acontecesse naturalmente.

Vale ressaltar que não houve limitação de duração de nenhum áudio.

A Tabela 1 exibe os efeitos e seus parâmetros aplicados sobre os áudios pré-processados do *dataset*. Os efeitos foram aplicados utili-

Tabela 1 – Parâmetros dos efeitos determinísticos aplicados em cada áudio do dataset. Os resultados da aplicação desses efeitos serviram como base de comparação para a emulação por resposta ao impulso. Os parâmetros foram escritos exatamente como requerido pela biblioteca Pedalboard, onde a unidade de medida dos parâmetros não adimensionais é explicitada no próprio nome.

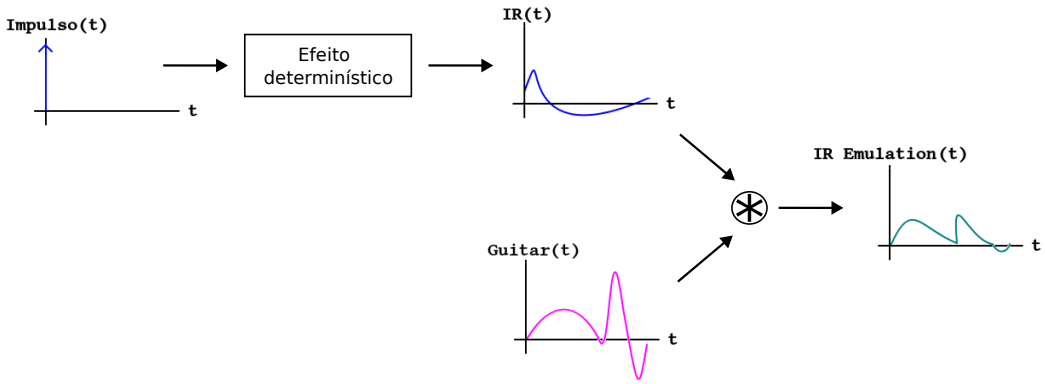
Efeito	**Parâmetro**	**Valor**	**Descrição**
Lowpass Filter	cutoff_frequency_hz	200	Frequência de corte para remover frequências altas
Highpass Filter	cutoff_frequency_hz	5000	Frequência de corte para remover frequências baixas
Distortion	drive_db	14	Nível de distorção aplicado ao sinal
Reverb	room_size	0.9	Tamanho simulado da sala para o efeito de reverb
	damping	0.3	Taxa de absorção do som, controlando o tempo de decaimento
	wet_level	0.8	Nível do sinal processado pelo efeito de reverb
	dry_level	0.2	Nível do sinal original, sem o efeito
	width	0.9	Largura estéreo do efeito
Compressor	threshold_db	-24	Nível a partir do qual a compressão começa a ser aplicada
	ratio	10	Proporção da redução de ganho quando o sinal excede o threshold
	attack_ms	20	Tempo que o compressor leva para agir após o sinal ultrapassar o threshold
	release_ms	250	Tempo que o compressor leva para parar de agir após o sinal baixar do threshold
Chorus	rate_hz	1	Taxa de modulação do efeito de chorus
	depth	0.3	Profundidade da modulação aplicada
	centre_delay_ms	7	Atraso central aplicado ao sinal duplicado
	feedback	0.05	Quantidade de sinal de saída realimentado para a entrada
	mix	0.6	Proporção entre o sinal original e o sinal modulado

Fonte: Os autores.

zando a biblioteca em *Python* denominada *Pedalboard*, desenvolvida pelo *Spotify*[3].

Os áudios com aplicação dos efeitos determinísticos formaram a base para as comparações. As emulações produzidas, e que serão descritas à seguir, buscaram transformar os áudios pré-processados em seus respectivos sinais pós aplicação dos efeitos. Nesse sentido, para garantir uma boa percepção sonora dos efeitos, exagerou-se, propositalmente, na escolha dos valores dos parâmetros de cada um.

Figura 5 – Procedimento adotado para a obtenção das respostas ao impulso correspondentes aos seis efeitos considerados na primeira parte do projeto. Ressalta-se que os desenhos dos sinais apresentados são meramente ilustrativos, com exceção do próprio sinal de impulso.



Fonte: Os autores.

3.1.2 Emulação por Convolução da Resposta ao Impulso

Com o objetivo de emular os efeitos sonoros descritos, empregou-se a técnica de Convolução da Resposta ao Impulso. Para fazer isso, em um primeiro momento, foi necessário adquirir as IRs de cada um dos efeitos. Depois disso, bastou convoluí-las com cada um dos áudios pré-processados do *dataset* para que, então, o resultado da emulação pudesse ser confrontado com o verdadeiro efeito determinístico que se desejou emular. A Figura 5 ilustra esse fluxo de desenvolvimento.

Com o objetivo de gerar o sinal de impulso necessário para a obtenção das IRs, os autores utilizaram a linguagem de programação Python, por meio da biblioteca scipy.signal. O sinal de impulso adotado foi um vetor composto por um único valor unitário seguido

[3] https://github.com/spotify/pedalboard

de zeros – caracterizando o impulso discreto ideal. A geração desse vetor foi realizada com o seguinte comando:

```
from scipy import signal
sr = 16000
imp_sec = 5
impulse_signal = signal.unit_impulse(sr * imp_sec)
```

Nesse código, *sr* representa a taxa de amostragem do áudio (em Hz), enquanto *imp_sec* define a duração desejada do sinal de impulso (em segundos). O resultado é um vetor do tipo *numpy.ndarray*, cujo primeiro valor é 1 e os demais são zeros, simulando matematicamente um impulso discreto.

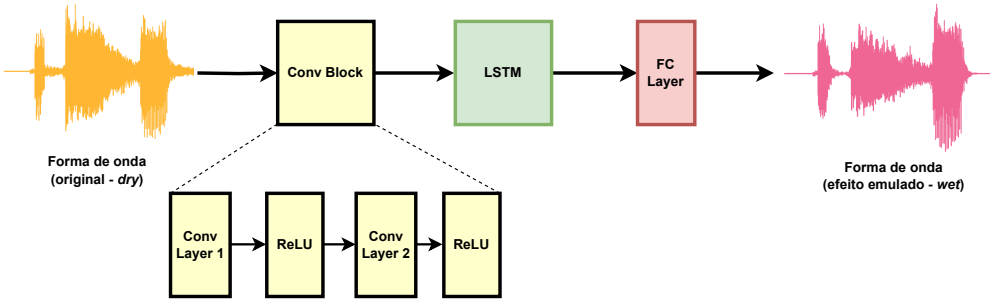
3.1.3 Emulação por Redes Neurais Artificiais

Visando emular os efeitos sonoros apresentados por meio de Redes Neurais Artificiais e compará-los com o efeito determinístico, implementou-se uma rede baseada em uma arquitetura do tipo LSTM [4], inspirada nos trabalhos de Wright et al. (WRIGHT et al., 2020b) e Bloemer (BLOEMER, 2022), adaptada para o contexto da emulação dos efeitos sonoros abordados neste projeto. Na Parte I, o objetivo foi avaliar a capacidade da rede em simular individualmente seis efeitos

[4] Além da LSTM utilizada como referência, foram feitos testes preliminares com outras arquiteturas, como uma CNN Vanilla (sem recorrência), variações da LSTM com camadas customizadas e modelos baseados em transformer, incluindo o Wav2Vec2. Como os resultados iniciais não foram satisfatórios, as análises desse estudo tiveram como foco a LSTM apresentada.

determinísticos – *reverb*, distorção, *chorus*, compressão, filtro passa-baixas e filtro passa-altas – levando em consideração as diferentes características de cada tipo de efeito.

Figura 6 – Arquitetura desenvolvida baseada em CNN+LSTM para emulação dos efeitos.



Fonte: Os autores.

A arquitetura da rede desenvolvida, apresentada na Figura 6, é composta por três blocos principais: um bloco convolucional, uma camada recorrente do tipo LSTM e, por fim, uma camada linear de saída. O bloco convolucional é formado por duas camadas convolucionais 1D com *kernel* de tamanho 3, seguidas por funções de ativação $ReLU$. Em seguida, a saída dessas camadas é reorganizada para alimentar a LSTM, que opera com a dimensão temporal como sequência. A camada LSTM possui unidades ocultas configuradas para capturar as dependências temporais do sinal. Por fim, a saída correspondente ao último passo de tempo da LSTM é conectada a uma camada linear com um único neurônio, responsável por gerar a predição da próxima amostra do sinal com efeito.

Cada uma dessas camadas cumpre um papel específico na modelagem dos efeitos sonoros. As camadas convolucionais atuam como

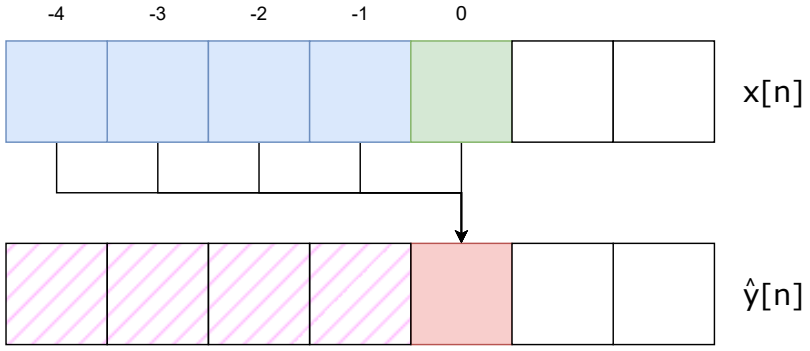
extratoras de características locais da forma de onda bruta (RAVANELLI; BENGIO, 2018; DAI et al., 2017), identificando padrões de curta duração e detalhes sutis no sinal de entrada, como picos, bordas, texturas e variações rápidas que caracterizam certos timbres e ataques. Esse processamento inicial é essencial para fornecer à LSTM uma representação mais rica e informativa do sinal. A camada LSTM (HOCHREITER; SCHMIDHUBER, 1997), por sua vez, é responsável por capturar as dependências temporais de médio e longo prazo, que são fundamentais para modelar efeitos com memória, como reverberações, compressões dinâmicas e modulações (como *chorus*), onde o valor atual do sinal depende não apenas das amostras recentes, mas de um histórico maior. Por fim, a camada linear interpreta a saída final da LSTM e mapeia o estado interno da rede para a predição da próxima amostra do sinal processado, fechando o ciclo de transformação do sinal limpo para sua versão com o efeito emulado.

Uma vez definida a arquitetura da RNA, também foi necessário considerar que, durante o processamento do áudio, a rede seria alimentada não apenas pela amostra atual de entrada, mas também um conjunto de amostras anteriores, formando uma janela de contexto deslizante. Esse mecanismo foi implementado para permitir que a rede neural capturasse a dependência temporal entre os frames do sinal, de modo que cada predição fosse gerada a partir da amostra atual e das N anteriores. Para definir o tamanho da janela de contexto ideal foram realizados testes com diversos comprimentos, variando de 5ms a 5s, e com base nessa análise preliminar, optou-se por utilizar

uma janela com um número de amostras correspondente a 300ms de áudio para todos os experimentos.

Como resultado da janela de contexto implementada, as primeiras N amostras da saída predita não podem ser geradas diretamente, uma vez que não há contexto suficiente disponível no início da sequência. Para contornar essa limitação e garantir que o áudio de saída tivesse a mesma dimensão do áudio original, foi aplicado um preenchimento (*padding*) com zeros no início do sinal de entrada de tamanho corresponde à janela de contexto. Essa abordagem permitiu manter a coerência dimensional entre os sinais ao longo de toda a sequência, possibilitando a comparação direta o sinal alvo e o predito. A Figura 7 ilustra esse processo, destacando a janela de contexto aplicada à entrada e a correspondente predição gerada para a saída.

Figura 7 – Diagrama ilustrando o mecanismo de janela deslizante aplicado ao sinal de entrada, no qual cada predição $\hat{y}[n]$ é gerada a partir da amostra atual $x[n]$ e das N amostras anteriores.



Fonte: Os autores.

O treinamento da RNA foi realizado utilizando o otimizador

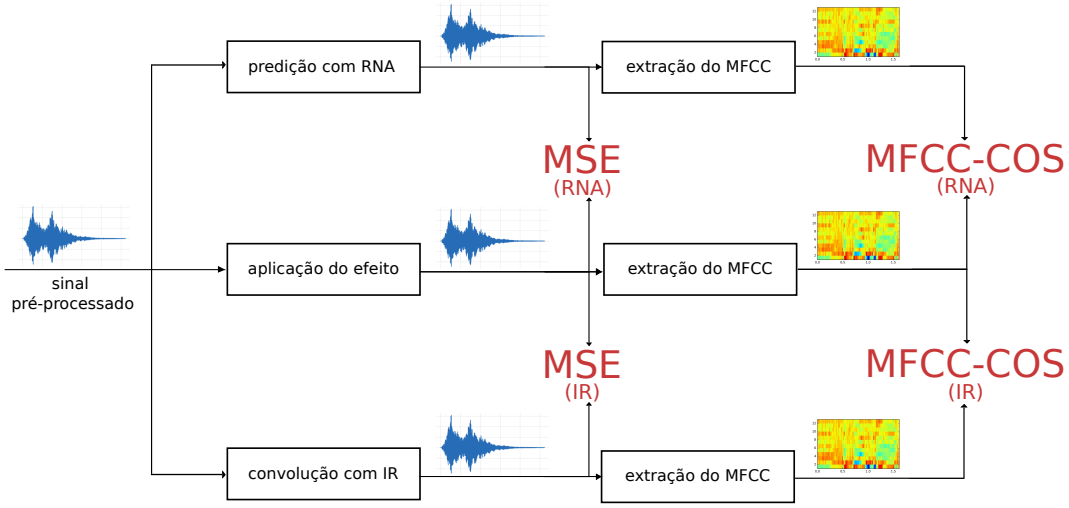
Adam, com taxa de aprendizado inicial de 0.001, escolhido por sua eficiência na adaptação de parâmetros em problemas com alta dimensionalidade. A função de perda adotada foi o MSE, apropriada para tarefas de regressão em que se deseja minimizar a diferença ponto a ponto entre as formas de onda dos sinais. Para auxiliar na convergência do modelo e evitar platôs prematuros, foi empregado um *scheduler* do tipo *StepLR*, que reduzia a taxa de aprendizado pela metade a cada duas épocas. O número total de épocas de treinamento foi fixado em 20, com base em experimentos preliminares que indicaram estabilização da função de perda e ausência de melhorias perceptuais após esse ponto.

3.1.4 Métricas de Comparação

Para comparar os resultados da emulação com o efeito determinístico, tanto na emulação por meio da IR quanto na utilização da RNA, foi necessário estabelecer uma métrica quantitativa e objetiva. Nesse contexto, o projeto se beneficiou do uso do **Erro Médio Quadrático** (MSE) e da **Similaridade do Cosseno dos Coeficientes MFCC** dos áudios.

A Figura 8 ilustra o processo de predição e avaliação das técnicas de emulação utilizadas para um determinado efeito sonoro. Inicialmente, o sinal original passa pelo pré-processamento descrito anteriormente. Em seguida, aplicam-se: 1) o efeito determinístico que se deseja emular; 2) a emulação por IR; e 3) a emulação por RNA. Em

Figura 8 – Metodologia para cálculo das métricas de comparação entre as diferentes técnicas de emulação. O sinal pré-processado é submetido à aplicação do efeito determinístico original e, em paralelo, realizam-se as predições com IR e RNA.



Fonte: Os autores.

cada um dos três sinais resultantes, calcula-se o MFCC utilizando 30 coeficientes e uma janela de 100 ms no domínio temporal. A matriz dos MFCCs é, então, convertida em um vetor unidimensional, possibilitando o cálculo da similaridade do cosseno entre a predição (y_pred) e o áudio após a aplicação do efeito determinístico (y_target). Ao final, utilizam-se as duas similaridades do cosseno calculadas para estabelecer um critério de comparação entre as duas técnicas de emulação. Além disso, os sinais resultantes das predições também são comparados com o sinal almejado através do MSE, onde, neste caso, a comparação pode ser realizada diretamente entre os sinais de áudio, sem a necessidade de conversão para os MFCCs.

3.2 Parte II

Na segunda etapa deste trabalho, os autores buscaram consolidar as ideias e o desenvolvimento apresentados na primeira parte do projeto, aplicando uma RNA em um cenário mais realista. Para isso, foi realizada uma gravação de um sinal de guitarra, da marca Cort, modelo KX5 FR, com mais de 34 minutos de duração, utilizando a interface de áudio *Scarlett Solo 4th Generation* equipada com um Conversor Analógico-Digital (*Analog-to-Digital Converter*, ADC). A Figura 9 mostra fotos dos autores durante a gravação.

Figura 9 – Montagem com três fotos dos autores realizando a gravação da Parte II do projeto. É possível enxergar a guitarra, a interface de áudio, e o DAW utilizados.

Fonte: Os autores.

Quatro efeitos comumente utilizados em produções musicais foram aplicados à gravação. O primeiro, um equalizador, teve como

objetivo remover frequências baixas ruidosas, além de realizar pequenos ajustes no espectro de frequências do instrumento. Em seguida, utilizou-se um compressor leve, também amplamente empregado em gravações musicais. O terceiro efeito foi um simulador de amplificador de guitarra, e, por fim, adicionou-se reverberação. A gravação e a aplicação dos efeitos em série tiveram como objetivo reproduzir um cenário realista, semelhante ao de um estúdio de gravação. Nesse ambiente, um músico tocaria seu instrumento, passando o sinal analógico por pedais de efeitos antes de amplificá-lo com um amplificador. O sinal, então, seria captado por um microfone posicionado diante do amplificador, resultando em uma gravação final que poderia ser mixada em uma música. Além disso, o uso da microfonação traria consigo a reverberação natural da sala, adicionando maior autenticidade ao som capturado.

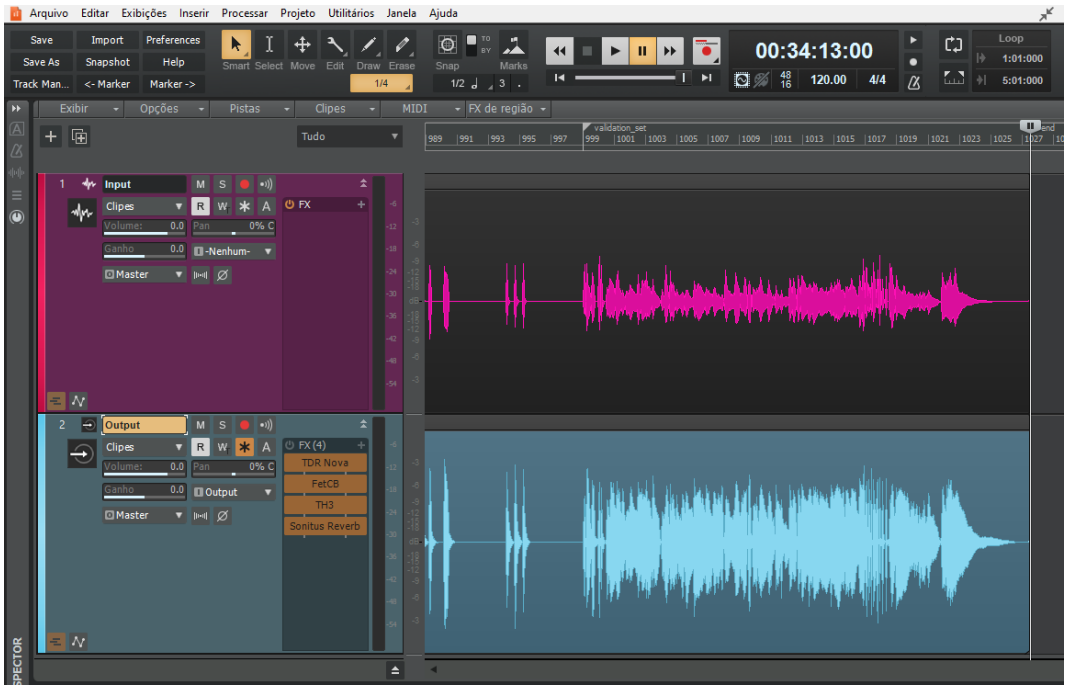
Tabela 2 – Configurações da gravação do sinal de guitarra da Parte II do projeto.

Parâmetro	**Valor**
Instrumento	Guitarra Cort KX5 FR
Interface de Áudio	Scarlett Solo 4th Generation
DAW	Cakewalk
Canais	1 (mono)
Frequência de Amostragem	44100 Hz
Profundidade de Bits	PCM 16
Tempo total de gravação	34m13s
Conjunto de treinamento	00m00s - 33m16s (33m16s)
Conjunto de teste	33m16s - 34m13s (57s)

Fonte: Os autores.

A Tabela 2 apresenta as configurações utilizadas durante a gravação. A maior parte do áudio capturado foi destinada à etapa de treinamento da rede neural, enquanto aproximadamente um minuto foi reservado para o conjunto de teste. Nessa fase, as mesmas métricas discutidas na Parte 1 foram avaliadas, incluindo a similaridade de cosseno dos coeficientes MFCC e o MSE.

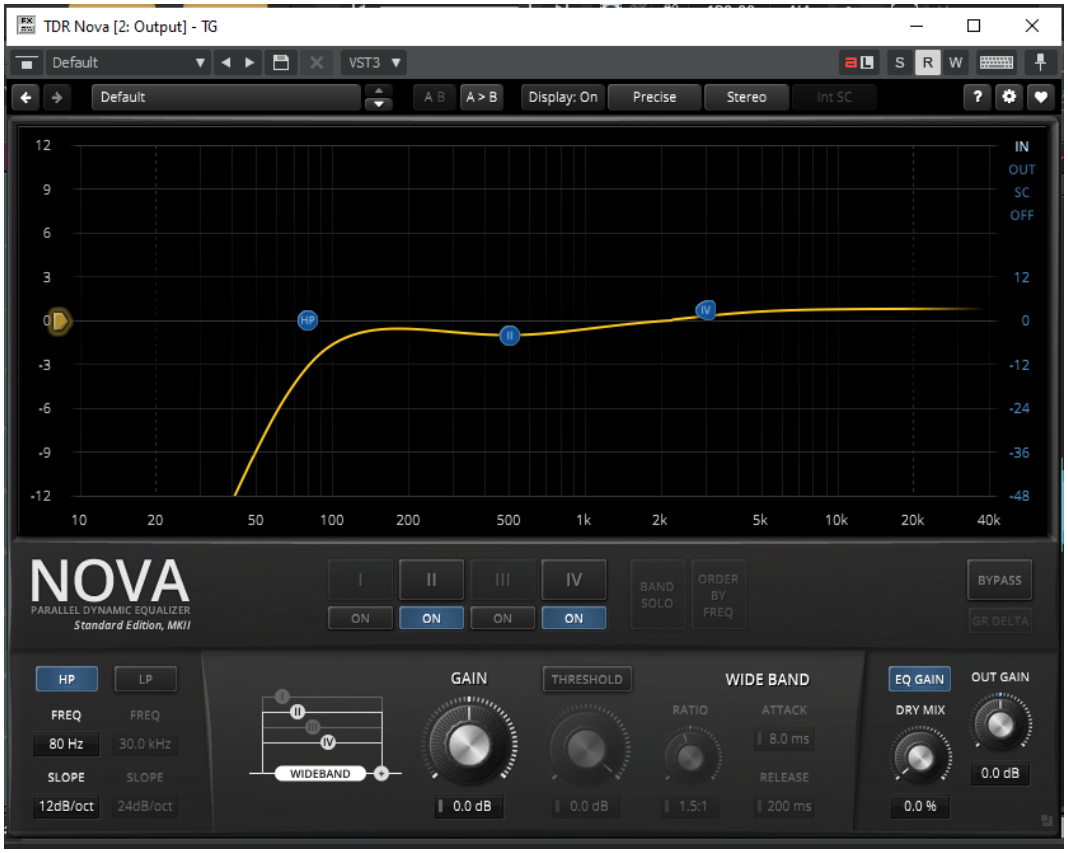
Figura 10 – Interface do *software* utilizado durante a gravação do *dataset* da segunda parte do projeto. Na parte superior, em rosa, observa-se a gravação original; na parte inferior, em azul, a gravação resultante após a aplicação dos quatro efeitos sonoros. Além disso, é possível visualizar os quatro *plugins* utilizados, bem como o tempo total de duração da gravação.



Fonte: Os autores.

1 TDR Nova
2 FetCB
3 TH3
4 Sonitus Reverb

Figura 11 – Interface do equalizador utilizado como primeiro efeito sonoro aplicado ao sinal. A descrição completa dos parâmetros utilizados pode ser consultada na Tabela 3.



Fonte: Os autores.

Com o intuito de garantir a reprodutibilidade do estudo, os autores optaram por utilizar única e exclusivamente *softwares* gratuitos e de acesso público disponíveis na *internet*. A Figura 10 mostra a tela da gravação no *software Cakewalk*, um DAW, mostrando a gravação original destacada em rosa e, logo abaixo, a faixa de saída em azul, que incorpora os quatro efeitos em série mencionados anteriormente. Também é possível observar, no topo da interface, a duração total da gravação, assim como a demarcação que separa o trecho reservado

Tabela 3 – Configurações dos efeitos utilizados na segunda parte do projeto.

Efeito	**Modelo**	**Parâmetro**	**Valor**	**Descrição**
Equalizador	TDR Nova[1]	HPF	80 Hz, 12 dB/oct	Filtro passa-altas
		Filtro de Banda	500 Hz, 0.5 Q, -1 dB	Reduz 1 dB em torno de 500 Hz
		Shelf Alta	3 kHz, 0.6 Q, +0.8 dB	Aumenta frequências acima de 3 kHz
Compressor	FetCB[2]	Ataque	30 ms	Tempo antes de o compressor atuar
		Release	80 ms	Tempo para liberar a compressão
		Ratio	2.5:1	Proporção de compressão aplicada
		Threshold	-22 dB	Nível a partir do qual a compressão inicia
		Ganho de Saída	9 dB	Ganho adicionado após compressão
Simulador de Amplificador	TH3[3]	Drive	0.25	Nível de saturação
		Bass	5	Controle de graves
		Mid	5	Controle de médios
		Treble	5	Controle de agudos
		Presence	3	Controle de presença (agudos adicionais)
		Volume	8	Nível de volume
Reverb	Sonitus Reverb[4]	HPF	500 Hz	Filtro passa-altas
		Predelay	30 ms	Tempo antes de o reverb começar
		Room Size	70	Tamanho do ambiente
		Diffusion	100%	Densidade das reflexões
		Decay Time	1.5 s	Duração do reverb
		Dry Gain	0 dB	Nível do som seco
		Reverb Gain	-9 dB	Nível do som com reverb

Fonte: Os autores.

para o conjunto de validação.

A Tabela 3 apresenta os efeitos utilizados, juntamente com seus respectivos modelos, todos eles também gratuitos e de acesso público. Além disso, são listados os parâmetros de configuração adotados para cada efeito. A tabela inclui, ainda, uma breve descrição de cada um desses parâmetros.

A Figura 11 mostra o primeiro dos efeitos aplicados, o equalizador *TDR Nova*. Conforme descrito anteriormente, esse equalizador utiliza um filtro passa-altas para eliminar os ruídos graves indesejados da gravação. Além disso, ele conta com outras duas bandas de equalização, cujas configurações detalhadas estão descritas na Tabela 3, cada uma com um papel específico no ajuste do espectro

Figura 12 – Interface do compressor utilizado como segundo efeito sonoro no processamento do sinal. A configuração detalhada de seus parâmetros está disponível na Tabela 3.

Fonte: Os autores.

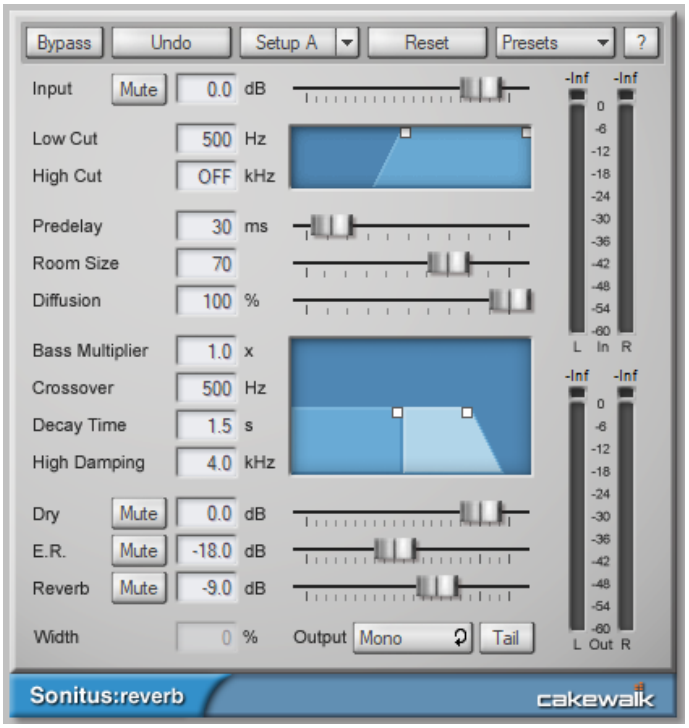
de frequências do áudio.

A Figura 12 apresenta o compressor utilizado no processo. Ajustaram-se os tempos de ataque e *release*, além dos parâmetros de *threshold* e *ratio*, com o intuito de aplicar uma compressão leve ao sinal. Na sequência, o simulador de amplificador, exibido na Figura 13, mostra os controles descritos na Tabela 3, incluindo o simulador de alto-falante. O simulador de amplificador é um recurso pré-instalado

Figura 13 – Interface do simulador de amplificador, terceiro efeito sonoro aplicado na cadeia. As configurações encontram-se na Tabela 3.

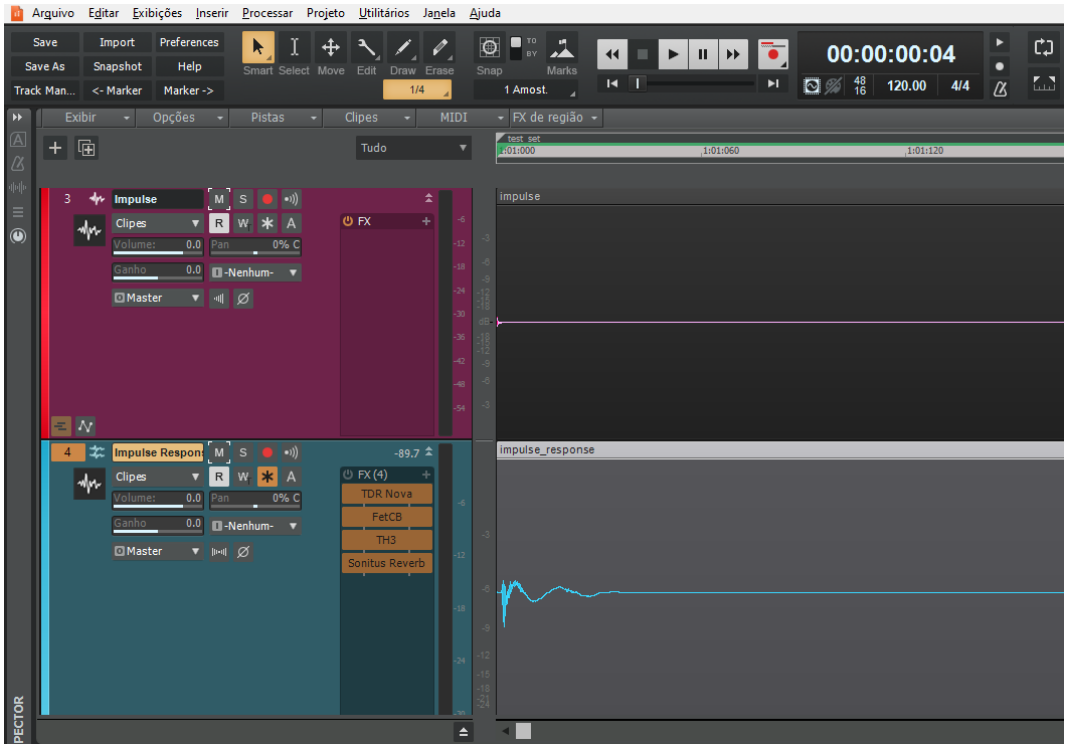
Fonte: Os autores.

Figura 14 – Interface do *plugin* de reverberação, quarto e último efeito sonoro aplicado ao sinal. As configurações encontram-se na Tabela 3.



Fonte: Os autores.

Figura 15 – Interface gráfica do *software* utilizado durante a obtenção da resposta ao impulso da cadeia de efeitos da Parte II do projeto. Acima, em rosa, verifica-se o próprio impulso. Abaixo, em azul, sua resposta após a aplicação dos quatro efeitos. A resposta ao impulso, então, foi utilizada na convolução para a obtenção da emulação por resposta ao impulso.



Fonte: Os autores.

na DAW utilizada, o que eliminou a necessidade de adquiri-lo através de *plugins* instalados externamente.

Por fim, a Figura 14 mostra o *reverb* aplicado, juntamente com suas respectivas configurações, conforme detalhado anteriormente. Assim como o simulador de amplificador, o efeito de reverberação também está pré-instalado na DAW utilizada, sem a necessidade de aquisição de *plugins* adicionais.

Após a gravação e a aplicação dos efeitos desejados, os autores avançaram para a etapa de desenvolvimento de uma RNA com a capacidade de emular os efeitos aplicados à gravação. Nesse sentido, o objetivo da RNA foi aprender a transformar o sinal de entrada da gravação – que não possuía a aplicação dos efeitos – no sinal resultante após a aplicação desses efeitos, sem ter conhecimento prévio sobre quais eram esses efeitos e seus parâmetros. Em outras palavras, a rede teve que aprender a converter o sinal representado em rosa na Figura 10 em seu equivalente em azul, que corresponde ao sinal processado.

Além disso, com o intuito de comparar a RNA desenvolvida com outra técnica de emulação, utilizou-se, novamente, a convolução da IR, seguindo o mesmo procedimento da primeira parte deste projeto.

A Figura 15 ilustra a interface gráfica do *software Cakewalk*, utilizado durante o processo de obtenção da resposta ao impulso dos quatro efeitos sonoros aplicados na segunda parte do projeto. Na parte superior da figura, em rosa, é apresentada a forma do impulso original, enquanto na parte inferior, em azul, encontra-se a resposta ao impulso após a aplicação dos efeitos sonoros.

4 Resultados e Discussões

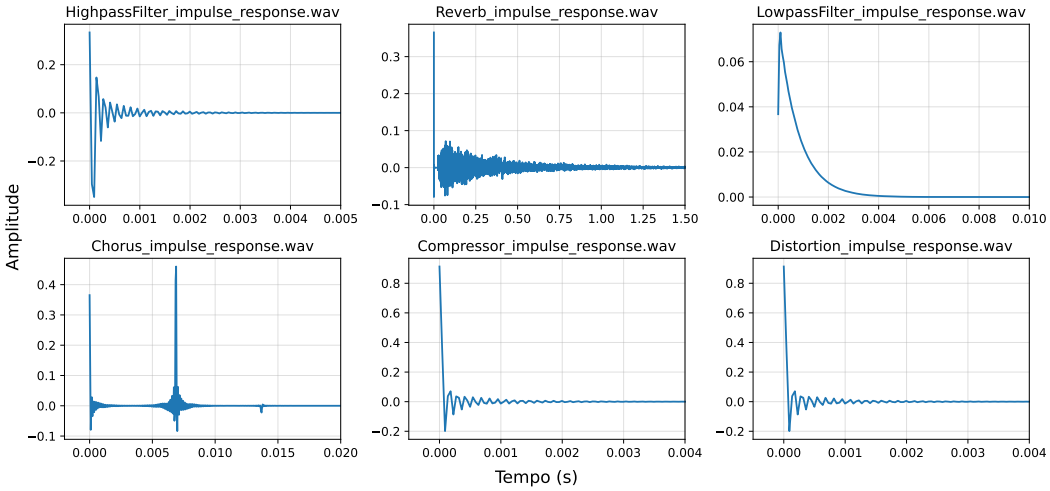
Esta seção apresenta os resultados obtidos nas duas etapas do projeto. Na Parte I, foram avaliadas as emulações de seis efeitos determinísticos aplicados individualmente, comparando-se o desempenho das redes LSTM com a técnica de Resposta ao Impulso. Já na Parte II, foi testada a capacidade da RNA em simular uma cadeia de efeitos combinados aplicada a uma gravação real de guitarra. As emulações foram avaliadas por meio do MSE e da MFCC-COS, além de análises perceptuais qualitativas. As próximas subseções detalham os principais resultados e comparações.

4.1 Parte I

A Figura 16 apresenta as respostas ao impulso utilizadas como base para a emulação dos efeitos sonoros na Parte I do projeto. Cada resposta representa a forma como um determinado efeito reagiu ao impulso descrito na Seção 3.1.2, sendo essa resposta utilizada posteriormente na convolução com os sinais originais. Vale ressaltar que para fins de facilitar a visualização, foi aplicado um zoom individual em cada gráfico, e com isso os eixos horizontal (tempo) e vertical (amplitude) não necessariamente compartilham a mesma escala entre cada um dos gráficos dos efeitos.

A fim de avaliar o desempenho da RNA em comparação com a

Figura 16 – Resposta ao impulso de cada um dos efeitos. Para facilitar a visualização, um *zoom* foi aplicado ao gráfico de cada sinal. Por esse motivo, os eixos horizontais e verticais não necessariamente coincidem.

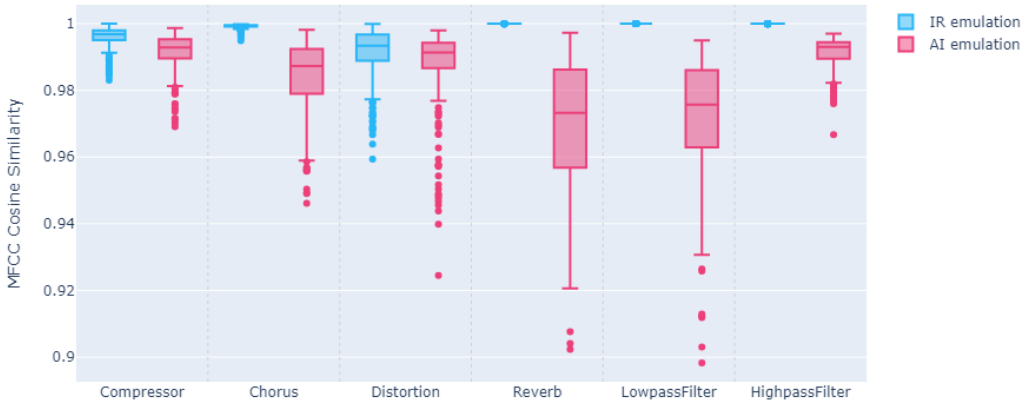


Fonte: Os autores.

técnica de resposta ao impulso, foi realizada uma análise que considerou cada áudio do conjunto de teste, calculando-se a similaridade entre o sinal original com efeito (alvo) e os sinais emulados tanto pela RNA quanto pela convolução com a resposta ao impulso. A Tabela 4 e a Figura 17 apresentam os resultados, comparando a distribuição da similaridade de cosseno dos MFCCs para cada um dos seis efeitos analisados. Cada *boxplot* representa a distribuição dos valores obtidos para os áudios individualmente, possibilitando uma visualização da variabilidade dos resultados por efeito.

Observa-se que, de modo geral, a emulação baseada em IR obteve desempenho superior à RNA, apresentando médias próximas de 1 e baixa variância. Isso é particularmente evidente para efeitos como *Reverb*, *Passa-baixa* e *Passa-alta*, em que a IR foi capaz de

Figura 17 – *Boxplot* da métrica MFCC-COS para cada um dos efeitos sonoros da Parte I do projeto. A comparação é feita entre a IR, representada em azul, e a predição obtida com a RNA, em rosa.



Fonte: Os autores.

reproduzir com alta fidelidade as características do sinal-alvo. A convolução com a resposta ao impulso, por ser uma técnica adequada para sistemas lineares e invariantes no tempo, se mostra eficaz para esses efeitos.

Por outro lado, embora a RNA tenha conseguido gerar sinais com similaridade acima de 90% em todos os casos, seus resultados foram mais dispersos, com desempenho inferior à IR em praticamente todos os efeitos. As maiores dificuldades foram observadas nos efeitos de *Reverb* e *Passa-baixa*, nos quais a variabilidade foi mais acentuada e os outliers indicam limitações do modelo em captar as características temporais prolongadas ou mesmo padrões específicos desses efeitos.

Tabela 4 – Resultados da métrica MFCC-COS por efeito e abordagem na primeira parte do projeto.

Efeito	**IR**				**RNA**			
	Mediana	Mínimo	Máximo	Desv. Padrão	Mediana	Mínimo	Máximo	Desv. Padrão
Compressor	0.9968	0.9831	0.9999	0.0032	0.9928	0.9691	0.9986	0.0051
Chorus	0.9993	0.9948	0.9998	0.0006	0.9872	0.9461	0.9981	0.0103
Distortion	0.9934	0.9594	0.9999	0.0073	0.9913	0.9245	0.9979	0.0107
Reverb	1.0000	0.9999	1.0000	0.0000	0.9733	0.9023	0.9972	0.0195
LowpassFilter	1.0000	0.9999	1.0000	0.0000	0.9757	0.8983	0.9949	0.0179
HighpassFilter	1.0000	0.9999	1.0000	0.0000	0.9931	0.9667	0.9970	0.0046

Fonte: Os autores.

Tabela 5 – Resultados da métrica MSE por efeito e abordagem na primeira parte do projeto.

Efeito	**IR**				**RNA**			
	Mediana	Mínimo	Máximo	Desv. Padrão	Mediana	Mínimo	Máximo	Desv. Padrão
Compressor	0.00013	0.00000	0.00037	0.00008	0.00061	0.00008	0.00237	0.00038
Chorus	0.00386	0.00007	0.02928	0.00593	0.00359	0.00007	0.01843	0.00398
Distortion	0.00493	0.00000	0.02643	0.00645	0.04801	0.00242	0.20927	0.04108
Reverb	0.00000	0.00000	0.00000	0.00000	0.01413	0.00039	0.11474	0.02062
LowpassFilter	0.00000	0.00000	0.00000	0.00000	0.00250	0.00000	0.01923	0.00327
HighpassFilter	0.00000	0.00000	0.00000	0.00000	0.00004	0.00000	0.00167	0.00020

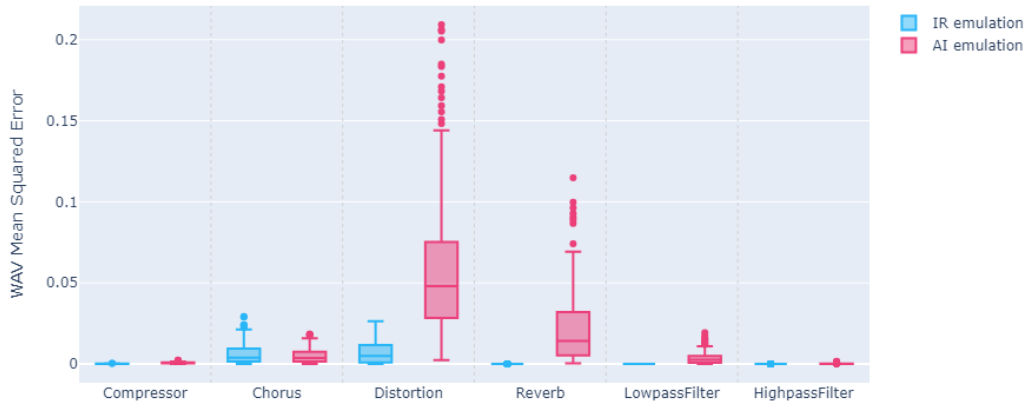
Fonte: Os autores.

No caso em específico do *Reverb*, essa degradação no resultado já era esperada, tendo em vista o tamanho da janela de contexto utilizada durante o processo de treinamento do modelo. Nos testes preliminares realizados com diferentes tamanhos de janela de contexto, foi possível observar impactos significativos na qualidade da predição. Quando utilizadas janelas muito pequenas – variando entre 5ms e 50ms –, o modelo teve dificuldade em capturar informações temporais relevantes, especialmente aquelas associadas a efeitos sonoros com maior duração ou latência. Por outro lado, ao testar janelas de contexto mais longas – entre 3 e 5 segundos – buscando justamente contemplar esses efeitos, observou-se que o modelo enfrentou sérias dificuldades de convergência. Nessas condições, o treinamento não evoluía adequadamente, as métricas de avaliação apresentavam desempenho muito inferior, e a percepção auditiva dos sinais gerados também indicava degradação significativa, que em vários cenários o áudio ficava com altos níveis de distorções indesejadas. Diante desse cenário, adotar uma janela de 300ms pareceu interessante pois, embora essa configuração não capture totalmente os efeitos de longa duração, foi possível fazer com que os modelos convergissem e apresentassem melhores resultados de similaridade em relação ao sinal *target*.

Estendendo a análise anterior, avaliou-se, também, o MSE para comparar o desempenho das duas abordagens de emulação. A Tabela 5 e a Figura 18 apresentam os valores de MSE obtidos individualmente para cada efeito sonoro, considerando as amostras

da base de testes. Um dos pontos mais relevantes observados nessa métrica está relacionado ao efeito de *Distorção*, que apresentou os maiores valores de erro, especialmente na emulação feita pela RNA. A diferença entre a forma de onda gerada pelo modelo de IA e o sinal alvo foi mais significativa nesse caso, revelando a dificuldade da rede em modelar com precisão os detalhes não lineares e abruptos típicos desse tipo de efeito. Esse resultado, inclusive, contrasta com a análise anterior baseada na similaridade dos MFCCs, na qual a discrepância observada foi menor. Essa divergência entre as métricas ressalta a importância de se utilizar múltiplos critérios de avaliação, já que cada métrica captura aspectos diferentes do sinal.

Figura 18 – *Boxplot* da métrica MSE para cada um dos efeitos sonoros da Parte I do projeto. A comparação é feita entre a IR, representada em azul, e a predição obtida com a RNA, em rosa.



Fonte: Os autores.

Enquanto o MSE compara diretamente as formas de onda no domínio temporal – sensível a pequenas variações e deslocamentos – a métrica baseada nos MFCCs considera uma perspectiva mais próxima da percepção auditiva humana, ao focar nas características espectrais do som. Assim, é esperado que os dois métodos ofereçam leituras complementares sobre a qualidade da emulação. De forma geral, observa-se que a emulação por IR obteve novamente desempenho superior à RNA na maioria dos efeitos, com erros médios mais baixos e menor dispersão. Esses resultados reforçam as conclusões anteriores de que a arquitetura proposta, embora funcional, ainda encontra dificuldades em capturar os comportamentos de certos efeitos, principalmente quando analisados com métricas de erro no domínio da forma de onda.

Esses resultados também apontam para uma limitação relevante da abordagem adotada neste projeto: todos os modelos de RNA foram treinados utilizando a mesma configuração de arquitetura e hiperparâmetros, independentemente do efeito a ser emulado. Embora tenham sido realizados diversos testes preliminares com variações de taxa de aprendizado, *scheduler* e outras configurações, definiu-se um conjunto padrão de hiperparâmetros para todos os treinamentos, conforme apresentado na seção Metodologia. A escolha dessa padronização teve como objetivo tornar a arquitetura mais generalista e alinhada ao propósito final da aplicação, em que se espera que a rede seja capaz de lidar com cadeias de efeitos combinados — como distorção seguida de reverberação ou modulação — sem que seja necessário fazer um

ajuste de parâmetros para cada cenário específico.

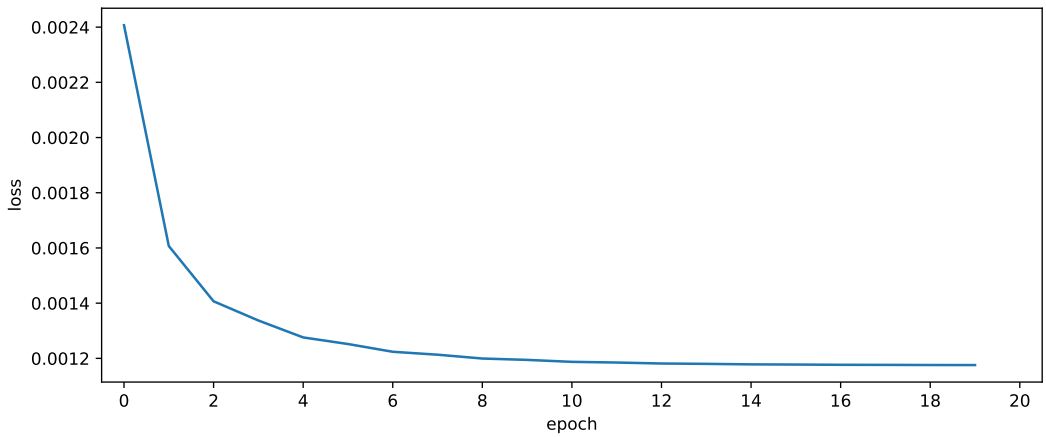
No entanto, os resultados evidenciam que essa abordagem de generalização traz limitações de desempenho, especialmente para efeitos que possuem características muito distintas entre si. Enquanto alguns efeitos como compressão e distorção foram razoavelmente bem modelados pela arquitetura proposta, outros — como o reverb e o passa-baixa — apresentaram maior dificuldade de emulação, indicando que uma única configuração de rede pode não ser ideal para capturar adequadamente os padrões de todos os efeitos. Apesar disso, essa limitação é coerente com a proposta do projeto: construir uma pipeline prática, em que uma única arquitetura possa ser utilizada como base para a emulação de diferentes efeitos ou mesmo combinações entre eles. Os resultados apresentados nesta seção servem, portanto, como uma avaliação sobre o *trade-off* entre desempenho individual por efeito e viabilidade de implementação em cenários reais, onde a flexibilidade e a generalização acaba sendo mais relevante.

4.2 Parte II

O processo de treinamento da RNA demonstrou estabilidade ao longo das épocas, conforme evidenciado pela função de perda apresentada na Figura 19. Observa-se uma queda acentuada nas primeiras épocas, seguida por uma estabilização gradual, indicando que o modelo conseguiu minimizar consistentemente o erro durante o aprendizado e convergir.

Durante o treinamento, observou-se que aproximadamente a partir das épocas 10 a 12 a função de perda passou a apresentar variações muito pequenas, e não foram mais percebidas melhorias auditivas relevantes nos áudios gerados pelo modelo, assim como não houve melhoras na métrica MFCC-COS também. Cabe ressaltar que amostras de saída foram periodicamente ouvidas ao longo do treinamento com o intuito de verificar se o modelo estava produzindo sinais coerentes.

Figura 19 – Evolução da função de perda (loss) ao longo das épocas durante o treinamento da RNA.



Fonte: Os autores.

A Tabela 6 apresenta o resultado consolidado das métricas exibidas nas Figuras 20 e 21. Os resultados obtidos na segunda parte do projeto indicam que, diante de um efeito sonoro consideravelmente mais complexo do que os utilizados na primeira parte (composto por uma cadeia de quatro efeitos distintos), a técnica de emulação baseada em RNAs apresentou uma melhora em seu desempenho no

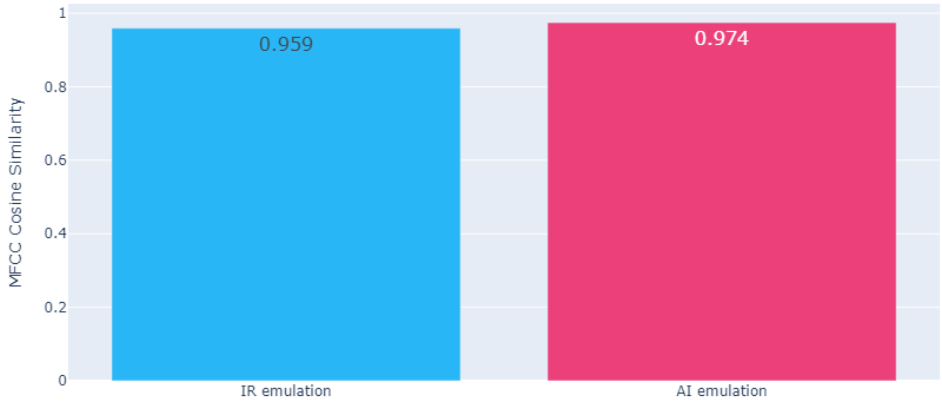
que diz respeito à comparação com a emulação por IR.

Tabela 6 – Consolidado das métricas de desempenho entre as abordagens de emulação com IR e com RNA. Os valores apresentados sintetizam os resultados já exibidos nas Figuras 20 e 21, oferecendo uma visão comparativa direta entre os métodos.

Métrica	**IR**	**RNA**
MFCC-COS	0.959	0.974
MSE	0.004	0.024

Fonte: Os autores.

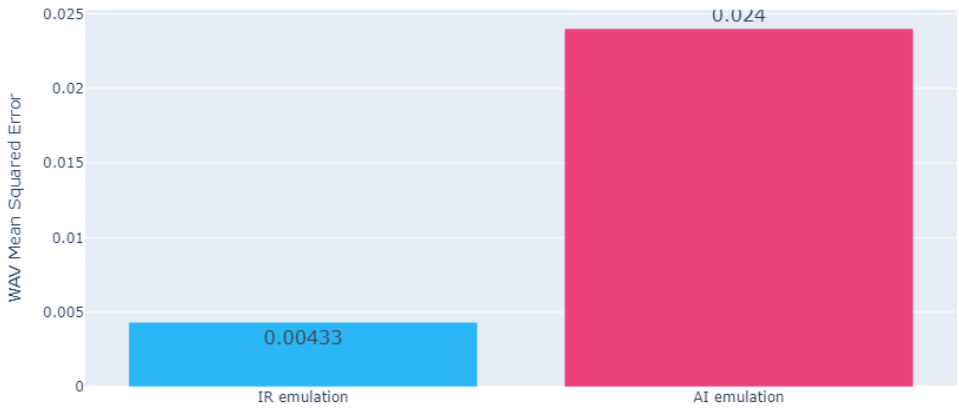
Figura 20 – Comparação entre as abordagens de emulação com IR e com RNA segundo a métrica de similaridade MFCC-COS. O modelo baseado em RNA obteve uma leve superioridade.



Fonte: Os autores.

Na métrica de MFCC-COS, a RNA apresentou uma leve superioridade em relação à técnica de emulação por IR. No entanto, em uma comparação mais direta entre os sinais de áudio, a métrica de MSE indicou que a emulação baseada na resposta ao impulso ainda

Figura 21 – Comparação entre as abordagens de emulação com IR e com RNA segundo a métrica de MSE. A abordagem com RNA apresentou desempenho inferior, com erro médio significativamente maior.



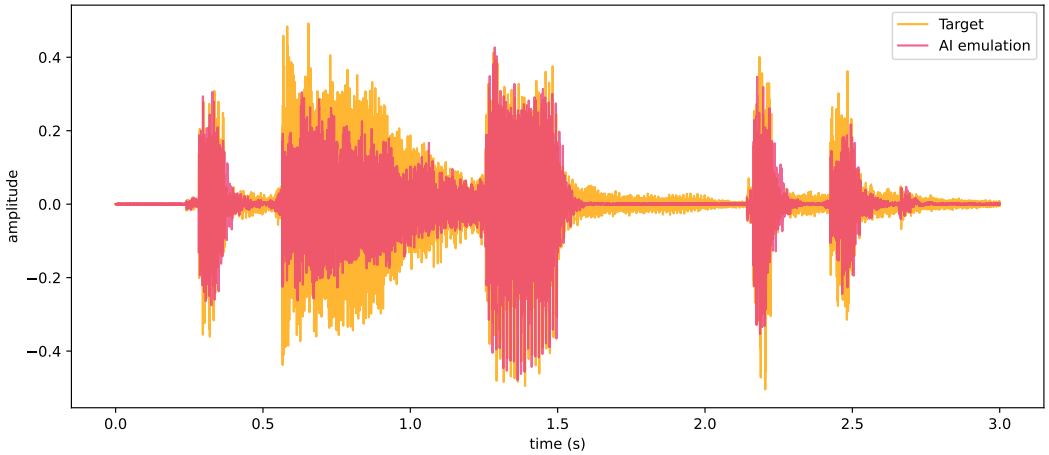
Fonte: Os autores.

pode oferecer resultados superiores. Como já discutido anteriormente, diferentes métricas são necessárias para avaliar resultados inerentemente subjetivos, como a similaridade entre áudios do ponto de vista musical.

Considerando a natureza subjetiva da avaliação proposta, foram ouvidos os áudios do conjunto de dados de validação referente à segunda parte do projeto. Em um primeiro momento, foi perceptível a presença de uma distorção sonora típica de sinais com *clipping*, o que levou à suposição inicial de que o sinal resultante estava sendo clipado. No entanto, ao analisar o sinal produzido pela RNA no domínio do tempo, conforme ilustrado na Figura 22, verificou-se que

não havia indícios de clipagem, seja por limitação na profundidade de *bits*, seja por falhas na conversão da saída predita pelo modelo para o formato de áudio WAV. Em vez disso, concluiu-se que o som gerado pela rede neural apresentava uma característica auditiva semelhante à clipagem, embora tal fenômeno não estivesse presente nas análises temporais do sinal.

Figura 22 – Comparação entre o sinal alvo e o sinal predito pela Rede Neural Artificial (RNA) no domínio do tempo.

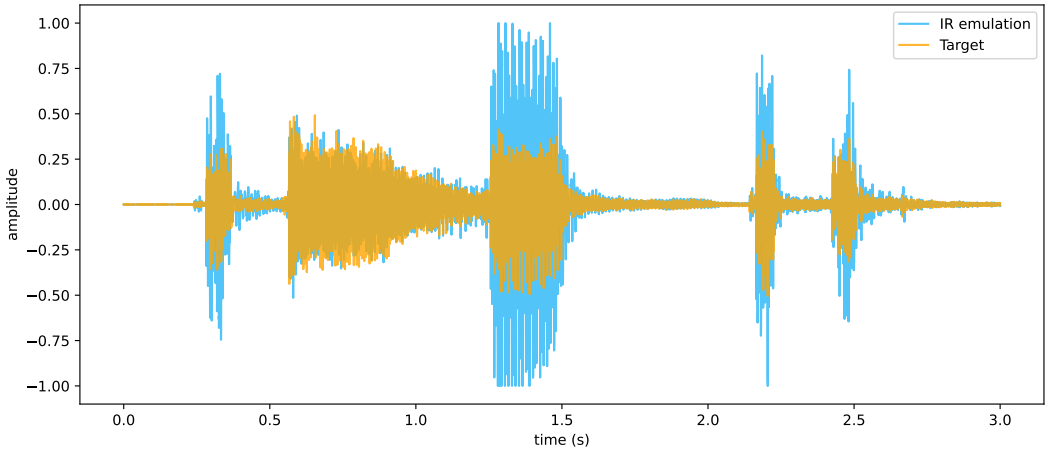


Fonte: Os autores.

Estendendo a análise para o sinal gerado pela técnica de convolução com resposta ao impulso (IR), representado na Figura 23, observa-se um pico abrupto em aproximadamente 1.25 s que se assemelha a um caso de *clipping* real. Curiosamente, essa anomalia visual não foi acompanhada de percepção auditiva correspondente ao ouvir o áudio. Uma possível explicação é que o pico excedente, embora numericamente significativo, tenha ocorrido em uma faixa de frequência ou momento do sinal de baixa relevância perceptual, sendo

atenuado pelo sistema auditivo humano. Isso reforça a importância de utilizar métricas perceptualmente alinhadas, como o MFCC-COS, além de análises no domínio do tempo.

Figura 23 – Comparação entre o sinal alvo e o sinal predito via convolução da resposta ao impulso (IR) no domínio do tempo.



Fonte: Os autores.

5 Considerações Finais

Este trabalho teve como objetivo investigar e comparar duas abordagens distintas para a emulação de efeitos sonoros aplicados à guitarra elétrica: a convolução com resposta ao impulso (Impulse Response – IR) e o uso de Redes Neurais Artificiais (RNAs), mais especificamente redes do tipo Long Short-Term Memory (LSTM). A proposta foi avaliada por meio da aplicação de diferentes efeitos — tanto lineares quanto não lineares — sobre sinais reais de guitarra, com o intuito de analisar a capacidade de cada método em reproduzir com fidelidade os resultados dos efeitos originais.

Os resultados obtidos indicaram que a abordagem baseada em convolução por IR superou, de maneira geral, as RNAs em termos de fidelidade de emulação, especialmente nos efeitos de natureza linear. Já os modelos de RNA, apesar de apresentarem desempenho razoável, apresentaram métricas consistentemente inferiores aos da IR em praticamente todos os efeitos testados. Isso evidencia a robustez e a eficácia da convolução na emulação de efeitos determinísticos e invariantes no tempo.

Entretanto, mesmo com desempenho inferior, as redes neurais demonstraram certo potencial na captura de características de efeitos mais complexos e não lineares, que representam um desafio maior para a modelagem tradicional via IR. A segunda parte do projeto re-

força essa constatação: no caso de uma cadeia de efeitos que combina operações lineares e não lineares, a emulação com RNA obteve melhor desempenho na métrica de similaridade MFCC-COS em relação à abordagem com IR. Ainda assim, o modelo LSTM mostrou limitações significativas ao tentar emular com precisão o efeito de *reverb*. A principal dificuldade observada foi a necessidade de janelas temporais longas para capturar de forma adequada a cauda e o comportamento recursivo desse tipo de efeito. Janelas pequenas comprometem a capacidade da rede em aprender as dependências temporais de longo prazo características do *reverb*, enquanto o uso de janelas maiores acarreta um aumento substancial no tempo de treinamento e dificuldade de convergência. Outra limitação a se destacar é a presença de artefatos inseridos pelo modelo no áudio exportado, que impactam significativamente na qualidade da percepção sonora.

O alto custo computacional foi um dos maiores desafios enfrentados durante o desenvolvimento. O treinamento dos modelos de RNA – especialmente os que envolviam janelas temporais extensas e conjuntos de dados maiores – frequentemente demandava várias horas de processamento, mesmo em máquinas equipadas com aceleração por GPU. Essa limitação impactou diretamente a capacidade de realizar ajustes finos de hiperparâmetros e de explorar arquiteturas mais profundas ou sofisticadas, restringindo o desempenho final das redes.

Apesar dessas dificuldades, o projeto atingiu seus objetivos

ao investigar a aplicação de técnicas de emulação de efeitos sonoros com base em inteligência artificial. Fica evidente que, enquanto a convolução por IR se mantém como uma solução altamente eficaz para efeitos lineares, há espaço para o avanço de modelos baseados em *deep learning*, especialmente com o uso de arquiteturas mais modernas, além da utilização de bases de dados maiores e mais variadas. Trabalhos futuros podem explorar com mais profundidade o uso de arquiteturas baseadas *transformers* e outros modelos com mecanismos de atenção, que podem oferecer maior capacidade de captura de dependências temporais, bem como replicar com mais fidelidade os efeitos sonoros sem inserir artefatos que comprometam a qualidade do áudio gerado.

Referências

BLOEMER, K. *GuitarLSTM*. 2022. https://github.com/GuitarML/GuitarLSTM. Citado na página 31.

BäCKSTRöM, T. Comparison of windowing in speech and audio coding. In: *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. [S.l.: s.n.], 2013. Citado na página 24.

COVERT, J.; LIVINGSTON, D. L. A vacuum-tube guitar amplifier model using a recurrent neural network. In: *2013 Proceedings of IEEE Southeastcon*. [S.l.: s.n.], 2013. p. 1–5. Citado 2 vezes nas páginas 17 e 22.

DAI, W. et al. Very deep convolutional neural networks for raw waveforms. In: IEEE. *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. [S.l.], 2017. p. 421–425. Citado na página 33.

DAMSKÄGG, E.-P. et al. Deep learning for tube amplifier emulation. In: IEEE. *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.], 2019. p. 471–475. Citado na página 22.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural computation*, MIT press, v. 9, n. 8, p. 1735–1780, 1997. Citado na página 33.

OPSTAL, J. van. Linear systems. In: *The Auditory System and Human Sound-Localization Behavior*. Elsevier, 2016. p. 51–86. Disponível em: <https://doi.org/10.1016/b978-0-12-801529-2.00003-9>. Citado 2 vezes nas páginas 15 e 16.

PEUSSA, A. et al. State-space virtual analogue modelling of audio circuits. 2020. Citado na página 22.

RAMÍREZ, M. A. M.; BENETOS, E.; REISS, J. D. Deep learning for black-box modeling of audio effects. *Applied Sciences*, v. 10, n. 2, 2020. ISSN 2076-3417. Disponível em: <https://www.mdpi-.com/2076-3417/10/2/638>. Citado 2 vezes nas páginas 21 e 24.

RAMÍREZ, M. A. M.; REISS, J. D. Modeling nonlinear audio effects with end-to-end deep neural networks. In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.: s.n.], 2019. p. 171–175. Citado na página 18.

RAVANELLI, M.; BENGIO, Y. Speaker recognition from raw waveform with sincnet. In: IEEE. *2018 IEEE spoken language technology workshop (SLT)*. [S.l.], 2018. p. 1021–1028. Citado na página 33.

SCHMITZ, T. Nonlinear modeling of the guitar signal chain enabling its real-time emulation. ULiège-Université de Liège [Applied Sciences], Liège, Belgium, 2019. Citado na página 22.

STEINMETZ, C. J.; REISS, J. D. Efficient neural networks for real-time analog audio effect modeling. *arXiv preprint arXiv:2102.06200*, 2021. Citado na página 22.

VANHATALO, T. et al. A review of neural network-based emulation of guitar amplifiers. *Applied Sciences*, v. 12, n. 12, 2022. ISSN 2076-3417. Disponível em: <https://www.mdpi-.com/2076-3417/12/12/5894>. Citado 2 vezes nas páginas 20 e 21.

WRIGHT, A. et al. Real-time guitar amplifier emulation with deep learning. *Applied Sciences*, Multidisciplinary Digital Publishing Institute, v. 10, n. 3, p. 766, 2020. Citado na página 22.

WRIGHT, A. et al. Real-time guitar amplifier emulation with deep learning. *Applied Sciences*, Multidisciplinary Digital Publishing Institute, v. 10, n. 3, p. 766, 2020. Citado na página 31.

WRIGHT, A. et al. Real-time black-box modelling with recurrent neural networks. In: *22nd international conference on digital audio effects (DAFx-19)*. [S.l.: s.n.], 2019. p. 1–8. Citado na página 22.

ZHENG, F.; ZHANG, G.; SONG, Z. Comparison of different implementations of MFCC. *Journal of Computer Science and Technology*, Springer Science and Business Media LLC, v. 16, n. 6, p. 582–589, 2001. Disponível em: <https://doi.org/10.1007-/bf02943243>. Citado na página 23.

ZÖLZER, U. (Ed.). *DAFX: Digital Audio Effects*. Wiley, 2011. Disponível em: <https://doi.org/10.1002/9781119991298>. Citado na página 12.