Avaliação de Modelos para Melhoramento de Sinais de Fala Usando o Conjunto de Dados NTCD-TIMIT

Augusto Cesar Becker, Gabriel Saatkamp Lazaretti, Rafael Rodrigo Pertum,
Eduardo Vinícius Kuhn e Rui Seara

***Resumo*— Este artigo visa avaliar o desempenho de modelos obtidos a partir das arquiteturas de redes neurais artificiais desenvolvidas por Park *et al.* e Zhang *et al.* para o melhoramento de sinais de fala. Especificamente, tais arquiteturas são aqui implementadas usando linguagem Python e a biblioteca TensorFlow, treinadas com o mesmo conjunto de dados (público) usando hiperparâmetros adequadamente escolhidos. Através de resultados de simulação, os modelos obtidos são avaliados utilizando métricas padronizadas confirmando que ambos os modelos melhoraram a qualidade e inteligibilidade dos sinais de fala processados, independentemente das características do ruído.**

***Palavras-Chave*— Métricas de qualidade e de inteligibilidade, redes neurais convolucionais, redes neurais recorrentes.**

***Abstract*— This paper aims to assess the performance of models obtained from the artificial neural network architectures developed by Park *et al.* and Zhang *et al.* for speech signal enhancement. Specifically, such architectures are implemented here using Python and the TensorFlow library, trained on the same (public) dataset with properly chosen hyperparameters. Through simulation results, the obtained models are evaluated by using standardized metrics confirming that both models improve the quality and intelligibility of the processed speech signals, irrespective of the noise characteristics.**

***Keywords*— Quality and intelligibility metrics, convolutional neural networks, recurrent neural networks.**

I. Introdução

No contexto de processamento de sinais, o melhoramento de sinais de fala refere-se a um conjunto de técnicas e algoritmos projetados para aprimorar a qualidade e a inteligibilidade da fala [1]. Basicamente, o objetivo é reduzir e/ou remover ruídos e distorções (artefatos) presentes em gravações ou transmissões envolvendo sinais de fala, tornando-os mais agradáveis e inteligíveis [2]. Nesse sentido, a adoção de diferentes técnicas de processamento tem o potencial de melhorar a qualidade da comunicação em sistemas de telefonia e videoconferência, aumentar a precisão em sistemas de reconhecimento de fala, melhorar a experiência e o conforto dos usuários em aplicações de voz, assim como facilitar a compreensão da fala em ambientes ruidosos. Dessa forma, tais técnicas tornam-se relevantes em setores distintos da economia em que a qualidade e a inteligibilidade da fala é um fator crítico, tais como telecomunicações, tecnologias de voz e assistência médica (acessibilidade) [3–6]. Como consequência, diversas abordagens para melhoramento de sinais de fala vêm sendo consideradas devido às suas importantes aplicações práticas.

Dependendo dos requisitos e das características da aplicação, diferentes técnicas podem ser utilizadas para o melhoramento de sinais de fala, as quais envolvem, e.g., filtragem adaptativa, estimação de ruído, cancelamento de ruído, modelos probabilísticos baseados em atributos e redes neurais [1, 2, 7, 8]. O uso de redes neurais é particularmente interessante tendo-se em vista a capacidade dos modelos resultantes de aprenderem padrões complexos mediante treinamento com grandes volumes de dados [8]. Dentre as arquiteturas mais difundidas na literatura, destacam-se a DNN (*deep neural network*) [9, 10], DAE (*denoising autoencoder*) [11, 12], RNN (*recurrent neural network*) [13], LSTM (*long short-term memory*) [14], CNN (*convolutional neural network*) [15–17], CRNN (*convolutional recurrent neural network*) [18] e GAN (*generative adversarial network*) [19]. Entretanto, considerando a natureza experimental associada à obtenção dos modelos, faz-se necessário realizar treinamento extensivo de diferentes arquiteturas a fim de identificar qual a mais adequada para um dado cenário de operação. Nesse contexto, focando na tarefa de melhoramento de sinais de fala contaminados por ruído, o presente trabalho de pesquisa visa:

i) implementar as arquiteturas introduzidas por Park *et al.* [15] e Zhang *et al.* [18], utilizando a biblioteca Tensorflow [20];
ii) efetuar uma busca extensiva em grade, em um espaço pré-definido, para auxiliar a escolha dos valores dos hiperparâmetros;
iii) realizar o treinamento das arquiteturas implementadas usando, exclusivamente, o conjunto de dados (público) NTCD-TIMIT [21]; e
iv) avaliar, comparar e discutir o desempenho dos modelos obtidos através de métricas objetivas de qualidade e inteligibilidade.

Note que um *framework* desenvolvido cumprindo tais metas pode servir para avaliar outras arquiteturas disponíveis na literatura utilizando um mesmo conjunto de dados, ajustes adequados de hiperparâmetros e métricas padronizadas, permitindo assim comparações justas de desempenho.

Augusto Cesar Becker e Rui Seara estão vinculados ao LINSE–Laboratório de Circuitos e Processamento de Sinais do Departamento de Engenharia Elétrica e Eletrônica da Universidade Federal de Santa Catarina (UFSC), Florianópolis, SC, Brasil (e-mails: augustobecker02@gmail.com; seara@linse.ufsc.br).

Gabriel Saatkamp Lazaretti atua como consultor em Ciência de Dados para a Sensoteq, Belfast, Irlanda do Norte, Reino Unido (e-mail: gabriel-lazaretti22@hotmail.com).

Rafael Rodrigo Pertum é egresso do curso de Engenharia Eletrônica da Universidade Tecnológica Federal do Paraná (UTFPR), Toledo, PR, Brasil (e-mail: pertum@alunos.utfpr.edu.br).

Eduardo Vinícius Kuhn está vinculado ao LAPSE–Laboratório de Processamento de Sinais e Eletrônica do Departamento de Engenharia Eletrônica da Universidade Tecnológica Federal do Paraná (UTFPR), Toledo, PR, Brasil (e-mail: kuhn@utfpr.edu.br).

Este trabalho foi parcialmente financiado pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

II. Formulação do Problema

Esta seção, primeiramente, apresenta o conjunto de dados NTCD-TIMIT [21] adotado. Em seguida, revisita as arquiteturas de Park *et al.* [15] e de Zhang *et al.* [18], consideradas aqui para o aprimoramento de sinais de fala. Na sequência, são descritas as métricas de qualidade e inteligibilidade utilizadas para avaliar o desempenho dos modelos obtidos. E, por fim, é discutida a abordagem para ajuste dos hiperparâmetros das arquiteturas consideradas.

A. Conjunto de Dados

O conjunto de dados NTCD-TIMIT consiste de sinais de fala obtidos do conjunto de dados TCD-TIMIT [22] corrompidos por 6 tipos distintos de ruído para diversos valores de SNR (*signal-to-noise ratio*). Especificamente, o conjunto de dados TCD-TIMIT contém gravações de 62 voluntários falantes nativos do inglês irlandês, sendo 3 deles locutores profissionais (desconsiderados aqui). Dessa forma, restam gravações de 59 voluntários, os quais proferiram ao todo 5782 sentenças (i.e., 98 sentenças por voluntário). Tais gravações são corrompidas por ruído i) branco, ii) de balbúrdia, iii) de carro, iv) de sala de estar, v) de rua e vi) de cafeteria, considerando 6 valores de SNR na faixa de -5 dB a 20 dB, o que resulta em 208152 sinais de fala corrompidos por ruído no NTCD-TIMIT. Esses sinais do conjunto de dados, para fins de treinamento e avaliação dos modelos, são separados (aleatoriamente) em três subconjuntos com base nos 59 voluntários considerados, i.e., 70% dos voluntários são selecionados para o conjunto de treinamento, 20% para validação e 10% para teste. Contudo, dado que algumas sentenças aparecem repetidas entre os voluntários, a divisão final após a remoção de repetições é de $79,22\%$ das gravações para treinamento, $12,79\%$ para validação e $7,99\%$ para teste. Vale mencionar que os sinais amostrados originalmente em 16 kHz são reamostrados para 8 kHz (assim como em [15]).

B. Arquiteturas

Em ambas as arquiteturas consideradas aqui (assim como para os modelos resultantes), os dados de entrada consistem de espectrogramas de magnitude extraídos dos sinais de fala corrompidos por ruído. Tais espectrogramas são gerados usando a STFT (*short-time Fourier transform*), com 8 espectros de magnitude obtidos a partir de janelas de tempo consecutivas (com comprimento de 256 amostras), sobreposição de 75% e janela definida com auxílio de busca extensiva, considerando apenas componentes de frequência não negativos. Então, para o processamento dos espectrogramas de magnitude[1], adota-se:

1) Arquitetura de Park et al. [15]: Essa arquitetura de CNN, conforme ilustrada na Fig. 1, é composta por 5 blocos R-CED (*redundant convolutional encoder-decoder*) na topologia *encoder-decoder* (com total de 32935 parâmetros, consumindo 128 KB de memória). O *encoder* é responsável por extrair características relevantes do sinal de entrada, aplicando diversos filtros no espectrograma do sinal de fala a partir de

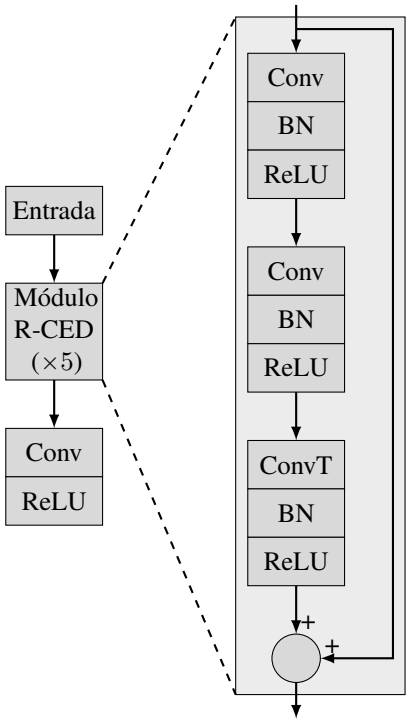

Fig. 1. Diagrama de blocos da arquitetura de Park *et al.* [15].

camadas convolucionais a fim de detectar diferentes padrões acústicos e capturar informações em várias escalas de tempo-frequência. Por sua vez, a partir dos atributos obtidos na saída do *encoder*, o *decoder* realiza a reconstrução do sinal aprimorado através de camadas convolucionais transpostas (ou camadas desconvolucionais). Vale destacar que, após cada camada convolucional ou desconvolucional, tem-se uma camada de *batch normalization* seguida pela função de ativação *leaky* ReLU (*rectified linear unit*); ainda, conexões de atalho (*skip connections*), conectando a entrada de cada módulo à entrada do próximo, são utilizadas para acelerar o treinamento. Por fim, o espectro de magnitude é então propriamente reconstruído por uma última camada de convolução, seguida da função de ativação *leaky* ReLU.

2) Arquitetura de Zhang et al. [18]: Essa arquitetura de CRNN, ilustrada na Fig. 2, segue também a topologia *encoder-decoder* (com total de 2492506 parâmetros, consumindo $9,51$ MB de memória). Todavia, na interconexão entre o *encoder* e o *decoder*, camadas GRU (*gated recurrent unit*) são utilizadas, visando combinar características de CNNs e RNNs para lidar com o processamento de sequências temporais (e.g., sinais de fala). Note que, após cada camada convolucional ou desconvolucional, tem-se uma camada de *batch normalization*, uma função de ativação *leaky* ReLU, seguida de uma camada de *dropout* (com proporção de 30%), visando prevenir *overfitting*. Por fim, uma última camada densa com função de ativação *leaky* ReLU[2] é responsável por reconstruir o espectro de magnitude do sinal processado.

Vale destacar que a variável alvo do processo de otimização, relativo ao treinamento de ambas as arquiteturas, é definida

[1]A reconstrução do sinal no domínio do tempo é realizada através da STFT inversa, utilizando o espectrograma de magnitude processado juntamente com o espectrograma de fase do sinal ruidoso.

[2]Em [18], uma função de ativação sigmóide é utilizada na saída.

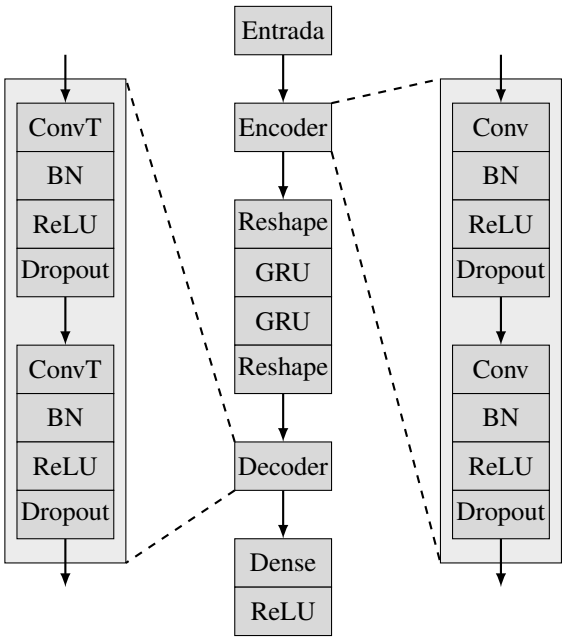

Fig. 2. Diagrama de blocos da arquitetura de Zhang *et al.* [18].

como o espectro de magnitude do sinal limpo (não corrompido por ruído) correspondente à última janela de tempo dos dados de entrada. E, como função de custo, utiliza-se MSE (*mean-squared error*), definido por

$$\text{MSE} = \frac{1}{N}\sum_{n=1}^{N} \|\widehat{\mathbf{Y}}_n - \mathbf{Y}_n\|^2 \tag{1}$$

onde $\widehat{\mathbf{Y}}_n$ denota a predição do modelo (estimativa do espectro de magnitude do sinal limpo), $\mathbf{Y}_n$ caracteriza a variável alvo (espectrograma de magnitude do sinal limpo), enquanto N é o número total de exemplos de entrada (tamanho do *batch*).

C. Métricas de Avaliação

Para a avaliação dos modelos obtidos, as seguintes métricas de qualidade e inteligibilidade são utilizadas [15]:

- SDR (*source-to-distortion ratio*) [23], definida por

$$\text{SDR} = \frac{1}{N}\sum_{n=1}^{N} 10\log_{10}\left[\frac{\| \mathbf{y}_n \|^2}{\| \widehat{\mathbf{y}}_n - \mathbf{y}_n \|^2}\right] \tag{2}$$

  avalia a razão entre a potência do sinal limpo $\mathbf{y}_n$ com respeito à potência residual entre $\mathbf{y}_n$ e o sinal aprimorado $\widehat{\mathbf{y}}_n$ (obtido na saída do modelo), visando capturar distorções introduzidas. Quanto maior a SDR, melhor o desempenho do modelo; afinal, um modelo eficaz deve ser capaz de reduzir o ruído sem introduzir artefatos e/ou distorções significativas.
- PESQ (*perceptual evaluation of speech quality*) [24] estabelece um método de avaliação objetiva de qualidade percebida da fala. Essa métrica tem a sua pontuação representada na escala MOS-LQO (*mean opinion score-listening quality objective*) de 1 (qualidade percebida mais baixa) a $4,5$ (qualidade percebida mais alta).
- STOI (*short-time objective intelligibility measure*) [25] visa estimar a degradação da inteligibilidade de um sinal de fala em decorrência do seu processamento. Essa métrica atribui, por comparação entre o sinal limpo e o sinal processado, uma pontuação na escala de 0 (quando ocorre degradação significativa de inteligibilidade) a 1 (inteligibilidade máxima).

Portanto, tais métricas possibilitam avaliar tanto a qualidade quanto a inteligibilidade dos sinais processados, fornecendo uma base consistente para avaliação dos modelos obtidos.

D. Sobre a Seleção de Hiperparâmetros

A seleção de hiperparâmetros é feita com base em uma busca extensiva em grade, a qual é realizada sobre um conjunto predefinido de valores. Esse espaço de busca contempla i) a janela de {Hamming, Hann}; ii) a taxa de aprendizado de $\{0,0001,\ 0,0005,\ 0,001\}$; iii) o parâmetro de *momentum* das camadas de *batch normalization* de $\{0,99,\ 0,995,\ 0,999\}$; e iv) a inclinação negativa da função *leaky* ReLU de $\{0,001,\ 0,01,\ 0,1\}$. Dessa forma, 54 combinações de hiperparâmetros são produzidas para testar cada arquitetura, totalizando assim 108 rodadas de treinamento. Cada rodada de treinamento é realizada por 30 épocas, sendo a taxa de aprendizado reduzida pela metade quando uma estimativa da SDR (computada entre os espectros de magnitude do sinal de fala limpo e estimado/predito) não apresentar melhora por 4 épocas consecutivas. Apenas o subconjunto de sinais de fala contaminados por ruído de balbúrdia com SNR de 0 dB é usado aqui. Após cada rodada de treinamento, o modelo obtido é avaliado utilizando a estimativa da SDR sobre o conjunto de validação, sendo o resultado armazenado em um arquivo `.csv` para análises posteriores.

III. Resultados e Discussão

Esta seção trata inicialmente de alguns detalhes pertinentes à implementação e ao treinamento das arquiteturas de Park *et al.* [15] e de Zhang *et al.* [18]. Também, apresenta os resultados obtidos a partir da busca em grade, possibilitando assim definir claramente quais valores de hiperparâmetros levam cada arquitetura ao melhor desempenho. Por fim, resultados da avaliação dos modelos frente às métricas SDR, PESQ e STOI são discutidos.

A. Implementação e Escolha de Hiperparâmetros

A implementação do código[3] necessário para o desenvolvimento do presente trabalho é realizada em linguagem Python (versão 3.11.5) com auxílio da biblioteca TensorFlow (versão 2.14.0). O treinamento das arquiteturas é executado em um computador dispondo de um processador AMD Ryzen 5800x, 64 GB de mémoria RAM, GPUs NVIDIA RTX 3080 e 3090, utilizando o sistema operacional Ubuntu 22.04. Quanto ao processo de treinamento, são utilizadas duas estratégias de *data augmentation* i.e., adição de um *padding* aleatório de até 64 amostras e aplicação de um ganho aleatório variando de -1 a 1 dB, proporcionando dados de entrada ligeiramente diferentes a cada época. Ademais, são considerados *batches* compostos por 512 amostras e otimizador Adam com

[3]Para detalhes, acesse *https://github.com/lablapse/speech_enhancement*.

TABELA I
RESULTADOS DA BUSCA EM GRADE PARA AS ARQUITETURAS DE PARK *et al.* [15] E DE ZHANG *et al.* [18], CONSIDERANDO UMA ESTIMATIVA DA MÉTRICA SDR.

			0,99			0,995			0,999		
			0,001	0,01	0,1	0,001	0,01	0,1	0,001	0,01	0,1
Park *et al.* [15]	Hamming	0,0001	6,77	6,54	6,79	6,58	6,63	6,76	6,65	6,51	6,70
		0,0005	7,63	7,65	7,60	7,49	7,60	7,50	7,51	7,61	7,65
		0,001	7,71	7,75	7,79	7,80	7,80	7,84	7,85	7,68	7,88
	Hann	0,0001	6,60	6,58	6,70	6,74	6,75	6,83	6,56	6,66	6,85
		0,0005	7,68	7,64	7,59	7,75	7,65	7,56	7,54	7,62	7,64
		0,001	7,66	7,82	7,71	7,83	7,80	7,82	7,74	7,82	7,87
Zhang *et al.* [18]	Hamming	0,0001	7,06	7,49	7,58	6,91	7,35	7,62	7,05	7,70	7,36
		0,0005	7,54	8,01	8,13	7,96	7,92	8,06	7,87	8,09	8,00
		0,001	7,70	7,94	7,90	7,80	7,81	8,17	7,61	7,67	7,71
	Hann	0,0001	6,96	7,59	7,74	7,39	7,55	7,67	7,42	7,63	7,44
		0,0005	7,65	7,98	8,17	7,64	7,60	8,05	7,53	8,15	7,97
		0,001	7,73	7,86	7,97	7,56	7,43	7,78	7,54	7,76	7,92

**Valores sublinhados indicam os melhores desempenhos alcançados.*

$\beta_1 = 0,9$, $\beta_2 = 0,999$ e $\varepsilon = 10^{-8}$ (como em [15]), sendo os demais hiperparâmetros ajustados para cada arquitetura de acordo com os resultados da busca em grade (veja a Tabela I). Em particular, para a arquitetura de Park *et al.* [15], verifica-se que o melhor desempenho é alcançado com o uso da janela de Hamming, taxa de aprendizado de $0,001$, hiperparâmetro de *momentum* igual a $0,999$ e inclinação negativa da função ReLU de $0,1$. Por sua vez, para a arquitetura de Zhang *et al.* [18], o melhor desempenho é atingido usando a janela de Hann, taxa de apredizado de $0,0005$, hiperparâmetro de *momentum* igual a $0,99$ e inclinação negativa da função ReLU de $0,1$. Portanto, levando em conta a combinação apropriada de hiperparâmetros, torna-se possível efetuar comparações justas de desempenho entre os modelos obtidos.

B. Comparações de Desempenho

As Figs. 3, 4 e 5 mostram os resultados obtidos para as métricas SDR, PESQ e STOI, respectivamente, considerando a média dos valores agregados tanto por SNR quanto por tipo de ruído; dessa forma, é possível inferir sobre o desempenho dos modelos frente às diferentes condições de operação presentes no conjunto de dados utilizado. Observa-se, dessas figuras, que ambos os modelos são capazes de aprimorar a qualidade e a inteligibilidade dos sinais de fala em comparação aos sinais não tratados (corrompidos por ruído), independentemente da SNR e/ou do tipo de ruído. Em particular, com respeito às métricas PESQ e STOI (capazes de capturar características da percepção humana), o modelo obtido da arquitetura de Zhang *et al.* [18] permite alcançar ganhos maiores de qualidade e inteligibilidade, sobretudo quando a SNR é reduzida. Esse melhor desempenho em termos de qualidade e inteligibilidade se sustenta mesmo frente aos diferentes tipos de ruído considerados. Todavia, o desempenho superior é atrelado ao número significativamente maior de parâmetros da arquitetura de Zhang *et al.* [18]. Dessa forma, o uso da arquitetura de Park *et al.* [15] pode ser vantajoso em aplicações com recursos computacionais limitados e/ou exigindo baixo consumo de energia. Vale ressaltar que comparações de desempenho com os resultados originais apresentados em Park *et al.* [15] e Zhang *et al.* [18] não são possíveis devido às diferenças metodológicas, indisponibilidade do conjunto de dados e falta de padronização quanto à escolha das métricas de desempenho adotadas.

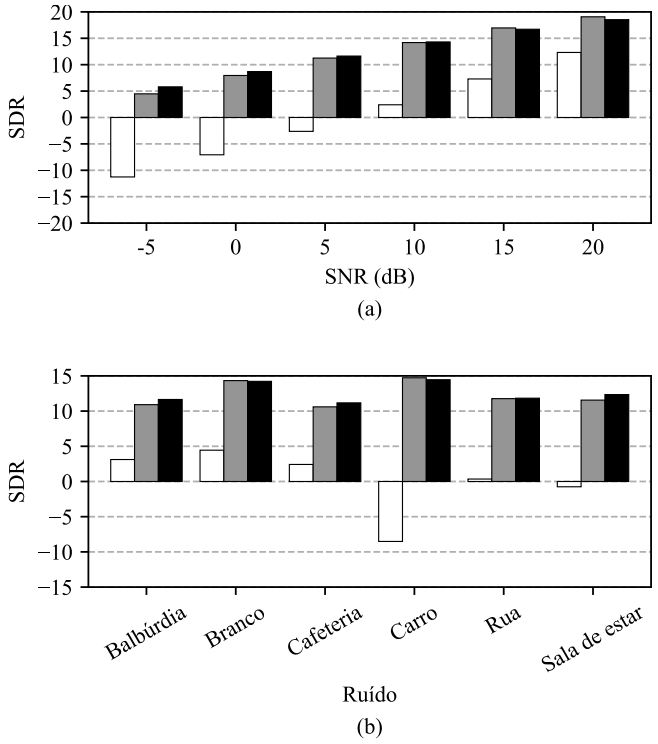

Fig. 3. Resultados da métrica SDR para sinais sem qualquer tratamento (branco) e processados pelos modelos de Park *et al.* [15] (cinza) e de Zhang *et al.* [18] (escuro), considerando a média dos valores agregados por (a) SNR e (b) tipo de ruído.

IV. CONSIDERAÇÕES FINAIS

Neste trabalho, as arquiteturas desenvolvidas por Park *et al.* [15] e Zhang *et al.* [18] para a tarefa de aprimoramento de sinais de fala foram avaliadas. Especificamente, ambas arquiteturas foram implementadas em linguagem Python usando a biblioteca TensorFlow, treinadas com o mesmo conjunto de dados e com hiperparâmetros apropriadamente escolhidos. Os modelos obtidos foram então avaliados através de métricas de desempenho padronizadas. Os resultados obtidos sugerem que ambos os modelos foram capazes de produzir melhoria de qualidade e inteligibilidade nos sinais de fala processados, quando comparados ao caso em que nenhum tratamento tenha sido realizado. Em particular, o modelo obtido da arquitetura de Zhang *et al.* [18] apresentou um desempenho superior ao obtido pela arquitetura de Park *et al.* [15]. Por fim, destaca-se que o *framework* desenvolvido aqui pode ser utilizado para avaliar outras arquiteturas disponíveis na literatura, bem como para verificar o desempenho de modelos frente a outros conjuntos de dados (e.g., sinais de fala em outros idiomas).

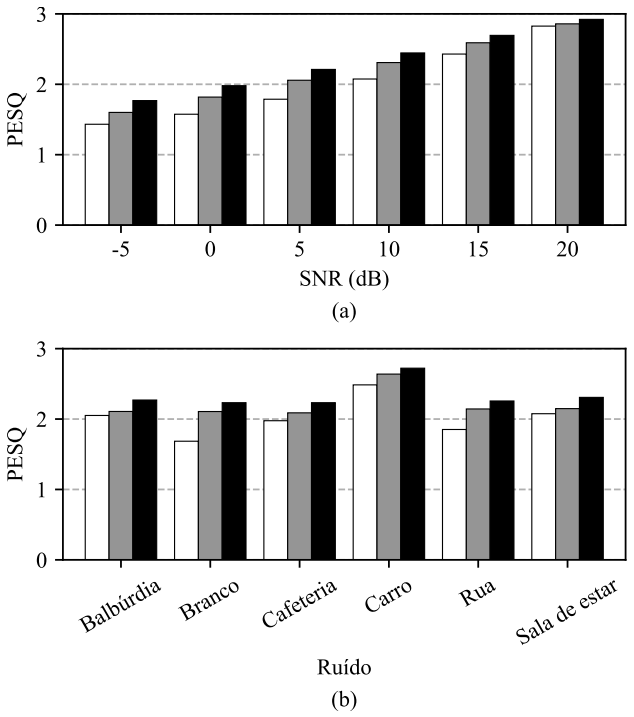

Fig. 4. Resultados da métrica PESQ para sinais sem qualquer tratamento (branco) e processados pelos modelos de Park *et al.* [15] (cinza) e de Zhang *et al.* [18] (escuro), considerando a média dos valores agregados por (a) SNR e (b) tipo de ruído.

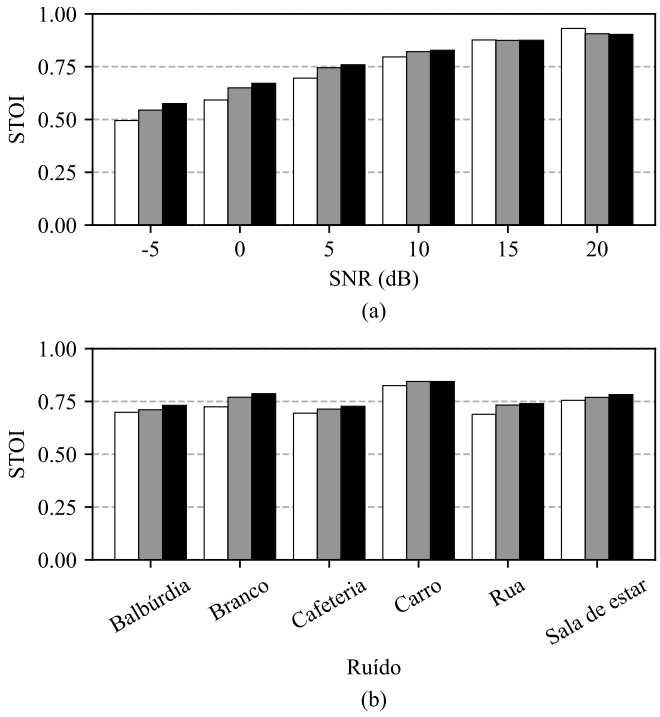

Fig. 5. Resultados da métrica STOI para sinais sem qualquer tratamento (branco) e processados pelos modelos de Park *et al.* [15] (cinza) e de Zhang *et al.* [18] (escuro), considerando a média dos valores agregados por (a) SNR e (b) tipo de ruído.

REFERÊNCIAS

[1] P. C. Loizou, *Speech Enhancement: Theory and Practice*, 2nd ed. Boca Raton, FL: CRC Press, 2013.

[2] P. Ochieng, "Deep neural network techniques for monaural speech enhancement and separation: State of the art analysis," *Artif. Intell. Rev.*, vol. 56, pp. 3651–3703, Oct. 2023.

[3] M. S. Kavalekalam *et al.*, "Model-based speech enhancement for intelligibility improvement in binaural hearing aids," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 1, pp. 99–113, Jan. 2019.

[4] R. Haeb-Umbach *et al.*, "Speech processing for digital home assistants: Combining signal processing with deep-learning techniques," *IEEE Signal Process. Mag.*, vol. 36, no. 6, pp. 111–124, Nov. 2019.

[5] Z. Sun *et al.*, "A supervised speech enhancement method for smartphone-based binaural hearing aids," *IEEE Trans. Biomed. Circuits Syst.*, vol. 14, no. 5, pp. 951–960, Oct. 2020.

[6] J. He *et al.*, "Functional split of in-network deep learning for 6G: A feasibility study," *IEEE Wireless Commun.*, vol. 29, no. 5, pp. 36–42, Oct. 2022.

[7] S. Haykin, *Adaptive Filter Theory*, 5th ed. Upper Saddle River, NJ: Prentice Hall, 2014.

[8] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, 1st ed. MIT Press, 2016.

[9] Y. Xu *et al.*, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal Process. Lett.*, vol. 21, no. 1, pp. 65–68, Jan. 2014.

[10] Y. Zhao *et al.*, "Perceptually guided speech enhancement using deep neural networks," in *Proc. Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Calgary, Canada, Sep. 2018, pp. 5074–5078.

[11] X. Lu *et al.*, "Speech enhancement based on deep denoising autoencoder," in *Proc. Int. Speech Communication Assoc. (INTERSPEECH)*, Lyon, France, Aug. 2013, pp. 436–440.

[12] X. Feng, Y. Zhang, and J. R. Glass, "Speech feature denoising and dereverberation via deep autoencoders for noisy reverberant speech recognition," in *Proc. Int. Conf. Acoust., Speech and Signal Process. (ICASSP)*, Florence, Italy, May 2014, pp. 1759–1763.

[13] A. L. Maas *et al.*, "Recurrent neural networks for noise reduction in robust asr," in *Proc. Int. Speech Communication Assoc. (INTERSPEECH)*, Portland, USA, Sep. 2012, pp. 22–25.

[14] T. Gao *et al.*, "Densely connected progressive learning for LSTM-based speech enhancement," in *Proc. Int. Conf. Acoust., Speech and Signal Process. (ICASSP)*, Calgary, Canada, Sep. 2018, pp. 5054–5058.

[15] S. R. Park and J. W. Lee, "A fully convolutional neural network for speech enhancement," in *Proc. Int. Speech Communication Assoc. (INTERSPEECH)*, Stockholm, Sweden, Aug. 2017, pp. 1993–1997.

[16] S.-W. Fu *et al.*, "Raw waveform-based speech enhancement by fully convolutional networks," in *Proc. Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, Kuala Lumpur, Malaysia, Dec. 2017, pp. 006–012.

[17] A. Pandey and D. Wang, "A new framework for CNN-based speech enhancement in the time domain," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, pp. 1179–1188, Jul. 2019.

[18] X. Zhang *et al.*, "Low-delay speech enhancement using perceptually motivated target and loss," in *Proc. Int. Speech Communication Assoc. (INTERSPEECH)*, Brno, Czechia, Sep. 2021, pp. 2826–2830.

[19] H. P. Phan *et al.*, "Improving GANs for speech enhancement," *IEEE Signal Process. Lett.*, vol. 27, pp. 1700–1704, Sep. 2020.

[20] M. Abadi *et al.*, *TensorFlow: Large-scale machine learning on heterogeneous systems*, Software available at tensorflow.org, 2015.

[21] A. H. Abdelaziz, "NTCD-TIMIT: A new database and baseline for noise-robust audio-visual speech recognition," in *Proc. Int. Speech Communication Assoc. (INTERSPEECH)*, Stockholm, Sweden, Aug. 2017, pp. 3752–3756.

[22] N. Harte and E. Gillen, "TCD-TIMIT: An audio-visual corpus of continuous speech," *IEEE Trans. Multimedia*, vol. 17, no. 5, pp. 603–615, Feb. 2015.

[23] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 14, pp. 1462–1469, Aug. 2006.

[24] International Telecommunication Union (ITU), "P.862: Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs," Recommendation no. P.862, 2001.

[25] C. H. Taal, R. C. Hendriks, and R. Heusdens, "An algorithm for intelligibility prediction of time–frequency weighted noisy speech," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 7, pp. 2125–2136, Feb. 2011.