
This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.

Wright, Alec; Välimäki, Vesa

Perceptual Loss Function for Neural Modelling of Audio Systems

Published in:

2020 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2020 - Proceedings

DOI:

[10.1109/ICASSP40776.2020.9052944](https://doi.org/10.1109/ICASSP40776.2020.9052944)

Published: 04/05/2020

Document Version

Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:

Wright, A., & Välimäki, V. (2020). Perceptual Loss Function for Neural Modelling of Audio Systems. In *2020 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2020 - Proceedings* (pp. 251-255). Article 9052944 (Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing). IEEE. <https://doi.org/10.1109/ICASSP40776.2020.9052944>

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

PERCEPTUAL LOSS FUNCTION FOR NEURAL MODELLING OF AUDIO SYSTEMS

Alec Wright and Vesa Välimäki*

Acoustics Lab, Dept. Signal Processing and Acoustics, Aalto University, FI-02150 Espoo, Finland
alec.wright@aalto.fi

ABSTRACT

This work investigates alternate pre-emphasis filters used as part of the loss function during neural network training for nonlinear audio processing. In our previous work, the error-to-signal ratio loss function was used during network training, with a first-order highpass pre-emphasis filter applied to both the target signal and neural network output. This work considers more perceptually relevant pre-emphasis filters, which include lowpass filtering at high frequencies. We conducted listening tests to determine whether they offer an improvement to the quality of a neural network model of a guitar tube amplifier. Listening test results indicate that the use of an A-weighting pre-emphasis filter offers the best improvement among the tested filters. The proposed perceptual loss function improves the sound quality of neural network models in audio processing without affecting the computational cost.

Index Terms— Audio systems, deep learning, digital filters, nonlinear distortion, psychoacoustics

1. INTRODUCTION

Virtual analog modelling is a field of research which seeks to create algorithms that emulate music hardware, such as instruments, amplifiers or audio effects [1]. Music hardware, specifically guitar amplifiers and distortion effects, exhibit highly nonlinear behaviour, which is particularly difficult to emulate accurately [2]. This study is on the topic of virtual analog modelling for highly nonlinear audio circuits, using Recurrent Neural Networks (RNNs).

Approaches to virtual analog modelling exist on a scale between “white-box” and “black-box”. In “white-box” modelling the physical characteristics of the circuit or instrument are studied and used to derive equations that describe their behaviour [3, 4, 5, 6]. In “black-box” modelling the input-output relationship of the device is emulated directly, by using data measured from the device [7, 8]. “Grey-box” models fall somewhere between these two approaches [9, 10].

In recent years a number of researchers have published work on black-box modelling of nonlinear audio circuits using neural networks. This has included a feedforward

WaveNet-style model [11, 12], as well as RNNs [13, 14] and a hybrid model, which combined a convolutional layer with an RNN [15]. In our previous work [11, 12, 14] the *error-to-signal ratio* (ESR) loss function was used during network training, with a first-order highpass *pre-emphasis* filter being used to suppress the low frequency content of both the target signal and neural network output.

In this work we investigate the use of alternate pre-emphasis filters in the training of a neural network consisting of a Long Short-Term Memory (LSTM) unit and a fully connected layer. We have conducted listening tests to determine whether the novel loss functions offer an improvement to the quality of the resulting model. This paper shows that in addition to the highpass filtering at low frequencies, it is advantageous to suppress the highest audio frequencies in the loss function, since such high frequencies are perceptually irrelevant.

The rest of this paper is structured as follows. Sec. 2 introduces the model used during this work. Sec. 3 introduces the device being modelled and the training procedure. Sec. 4 describes the pre-emphasis filters used during this work. Sec. 5 compares the resulting models, both in terms of objective error measurements and in terms of listening test scores. Finally, Sec. 6 concludes the paper.

2. NEURAL NETWORK MODEL

The model used in this work was first proposed in [14], it is depicted in Fig. 1. The model consists of a single LSTM unit, followed by a fully connected layer. It is an end-to-end model which takes a sequence of audio samples as input, and produces a sequence of audio samples as output. At each discrete time step in the sequence, the model predicts the corresponding output sample produced by the modelled device, based on the input sample, LSTM state and the model’s parameters:

$$\hat{y}[n] = f(x[n], s[n-1], \theta), \quad (1)$$

where n is the discrete time index, $\hat{y}[n]$ and $x[n]$ are the RNN’s predicted output sample and the corresponding unprocessed input sample respectively, $s[n-1]$ is the LSTM state from the previous time step, and θ are the RNN’s parameters which are learned during training.

*This research is part of the activities of the Nordic Sound and Music Computing Network—NordicSMC (NordForsk project no. 86892).

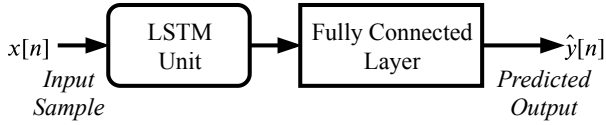


Fig. 1. Neural network architecture, where $x[n]$ is the input signal and $\hat{y}[n]$ is the network's predicted output sample.

3. MODELLED DEVICE AND TRAINING

In this study, the RNN models were trained to emulate the Blackstar HT-1 guitar amplifier [16], which was modelled previously in [14]. This is a small 1-Watt vacuum tube amplifier, with both a high gain and a low gain channel. The models were trained using guitar and bass audio processed by the amplifier. The dataset consisted of the same material used in [14], a total of approximately seven minutes of audio. The dataset was recorded with the amplifier set to the high gain channel, with the “EQ” and “volume” controls set to 10 and the “gain” control set to 5. All audio used in this study was recorded at a sampling rate of 44.1 kHz.

The training process was similar to that described in [14], with the data being split into half second segments and processed in mini-batches. For each segment, the first 1000 samples would be processed without gradient tracking, to allow the LSTM state to initialise. Then the remainder of the segment could be processed, with backpropagation and parameter updates being carried out every 2048 samples. After the complete training dataset was processed, it would be shuffled before the next training epoch began. All models were trained for a total of 750 epochs.

One issue with the comparison of pre-emphasis filters is that, due to random nature of neural network training, two networks of identical structure, once trained, will achieve different errors on the test set. When comparing two networks trained using different pre-emphasis filters, it is not clear if any differences in performance are due to the pre-emphasis filter used, or occur as a result of the training process. To compensate for this, five copies of each RNN configuration were trained, with the RNN that achieved the lowest loss, i.e. the best result, on the test set being used for evaluation.

4. LOSS FUNCTION PRE-EMPHASIS FILTERING

The neural networks were trained to minimise a loss function representing the ESR, which is the squared error divided by the energy of the target signal. The aim of this study was to investigate the application of different pre-emphasis filters to the network output and target signals, prior to calculating the loss. The aim of applying a pre-emphasis filter is to emphasise certain frequencies in the loss function, thus teaching the network to prioritise minimising certain frequencies over others. For an audio sequence of length N , the pre-emphasised

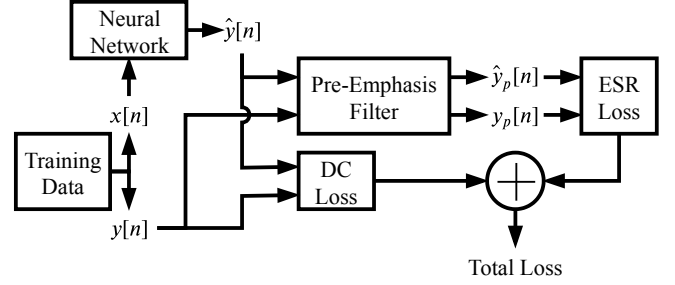


Fig. 2. Training process including the loss calculations.

ESR loss is given by

$$\mathcal{E}_{\text{ESR}} = \frac{\sum_{n=0}^{N-1} |y_p[n] - \hat{y}_p[n]|^2}{\sum_{n=0}^{N-1} |y_p[n]|^2}, \quad (2)$$

where y_p is the pre-emphasised target signal and $\hat{y}_p[n]$ is the pre-emphasised output of the neural network. As proposed in [14], a loss function measuring the difference in DC offset between the target and network output signals was also included, and is given by

$$\mathcal{E}_{\text{DC}} = \frac{|\frac{1}{N} \sum_{n=0}^{N-1} (y[n] - \hat{y}[n])|^2}{\frac{1}{N} \sum_{n=0}^{N-1} |y[n]|^2}. \quad (3)$$

The loss function used during training is simply the sum of the two loss terms:

$$\mathcal{E} = \mathcal{E}_{\text{ESR}} + \mathcal{E}_{\text{DC}}. \quad (4)$$

The process of calculating the loss is depicted in Fig. 2. The objective of this study is to investigate how the perceptual quality of the model output is affected by the choice of loss function pre-emphasis filter. The rest of this section describes the filters tested.

4.1. First-order highpass

As proposed in [11], and used in [12, 14], a first-order highpass pre-emphasis filter was the first pre-emphasis filter we considered during this study. Its transfer function is

$$H_{\text{HP}}(z) = 1 - 0.85z^{-1}. \quad (5)$$

The magnitude response of this filter is shown in the top panel of Fig. 3. This type of filter is commonly used in speech processing to boost spectral flatness [17]. In previous work this has been intended to suppress low frequencies in the signal, as the energy in the guitar amplifier output tends to be more concentrated in the lower end of the frequency spectrum.

4.2. Folded differentiator

The second pre-emphasis filter we tested during this study is the folded differentiator filter, with the transfer function

$$H_{\text{FD}}(z) = 1 - 0.85z^{-2}. \quad (6)$$

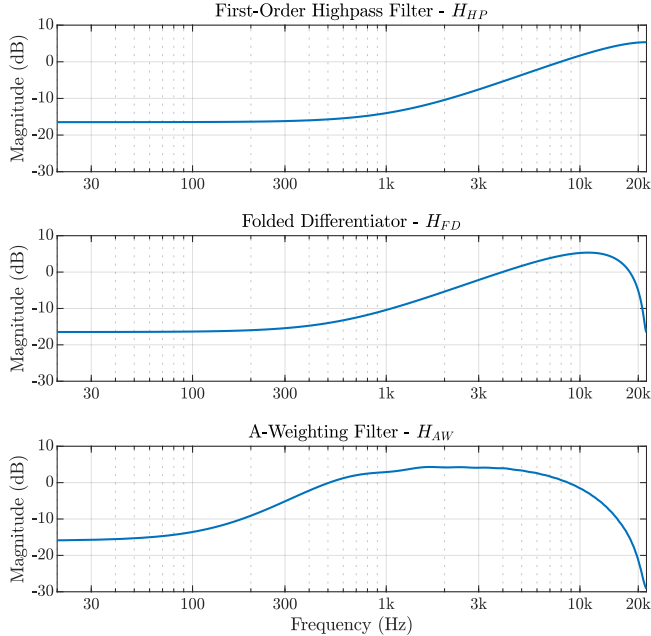


Fig. 3. Magnitude frequency response of pre-emphasis filters, at 44.1-kHz sample rate.

This was chosen as it was noted that the output of the guitar amplifier is very limited above 10 kHz. It is thought that attenuating these higher frequencies as well as the low frequencies, as shown in Fig. 3 (middle), might cause the network to concentrate more on the mid frequency range, which is perceptually significant.

4.3. A-weighting filter

The final pre-emphasis filter we tested during this study is based on the A-weighting curve, as defined in the International Electrotechnical Commission standard [18]. The A-weighting curve is intended to compensate for the relative loudness of different frequencies as perceived by the human ear. The A-weighting filter was then followed by a first order lowpass filter with the transfer function

$$H(z) = 1 + 0.85z^{-1}, \quad (7)$$

to decrease the emphasis on the high frequency region where relatively little energy is present. In this paper the low-passed A-weighting filter will be represented as H_{AW} .

Whilst an A-weighting filter can be implemented, for example, as an eleventh-order infinite impulse response filter [19], for this work we chose to implement it as a 100-tap finite impulse response (FIR) filter. As FIR filters are fully parallelisable, they can be applied very cheaply on a Graphics Processing Unit (GPU), and in practice can simply be implemented as a one-dimensional convolutional layer. As such the application of an 100-tap FIR filter had a negligible effect

on the overall time required to train the network. The filter was designed using the least-squares method, with the target magnitude frequency response calculated using the weighting function described in the IEC standard [18]. The low-passed A-weighting filter’s magnitude response is shown at the bottom of Fig. 3.

5. EVALUATION

To evaluate the pre-emphasis filters, the RNN model described in Sec. 2 was trained with an LSTM hidden size of either 32 or 64. For both of these hidden sizes, a total of four neural networks were compared, one trained with each of the three different pre-emphasis filters described in Sec. 4, as well as one which was trained using no pre-emphasis filter.

In our previous work we measured the running speed of a C++ implementation of this model, on an Apple iMac with a 2.8 GHz Intel Core i5 processor [14]. The time required to process a second of audio for the models with LSTM hidden size of 32 and 64 was 0.12 s and 0.24 s respectively. It should be noted that as the pre-emphasis filters are only applied during training, they have no influence on the running speed of the models.

5.1. Objective evaluation

For the objective evaluation, the loss was calculated for an unseen test set of electric guitar and bass guitar input signals. For each network evaluated, the test loss was calculated with the loss function with which that network was trained, as well as with the other three loss functions tested. The resulting test losses are shown in Table 1.

The results show that as you would expect, the RNN model with an LSTM hidden size of 64 achieved a lower test loss than the model with hidden size of 32. It might also be expected that for each loss function, the network that achieved the lowest test loss would be the network that was trained to minimise that same loss function. However, the results show that the networks trained using the H_{FD} pre-emphasised loss function achieved a higher H_{FD} pre-emphasised test loss, i.e. they performed worse, than the networks trained using the H_{HP} pre-emphasised loss function. Because of this, the networks trained with the H_{FD} pre-emphasised loss function were not used in the listening tests.

The test set error signal, obtained by subtracting the network’s predicted output signal from the target signal, is plotted in the frequency domain in Fig. 4. It can be observed that the error for each network varies according to frequency. For the network trained using the H_{AW} pre-emphasised loss function, the error is approximately 5 dB less than the other networks over the 1-2 kHz range, which is a region where human hearing is particularly sensitive.

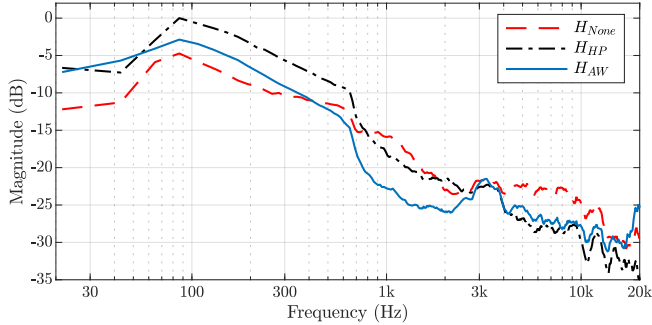


Fig. 4. Test set error in the frequency domain

Table 1. Test loss for the RNN models

Hidden Size	Training Pre-Emph.	Test Loss Pre-Emphasis			
		H_{None}	H_{HP}	H_{FD}	H_{AW}
32	H_{None}	1.56%	10.9%	7.68%	3.16%
	H_{HP}	2.49%	6.78%	4.54%	3.10%
	H_{FD}	2.92%	10.1%	5.57%	3.33%
	H_{AW}	3.20%	23.1%	7.73%	2.66%
64	H_{None}	0.77%	5.65%	3.56%	1.41%
	H_{HP}	1.08%	3.79%	2.61%	1.66%
	H_{FD}	1.31%	5.92%	2.87%	1.63%
	H_{AW}	1.01%	12.3%	4.15%	1.28%

5.2. Listening test

To assess how well each of the neural networks emulate the guitar amplifier, a multiple stimuli with hidden reference and anchor (MUSHRA) listening test was conducted [20]. The webMUSHRA interface was used to conduct the tests [21].

For each MUSHRA trial, the participant was presented with an audio clip that had been processed through the guitar amplifier, as well as clips that had been processed by neural network models trained by each of the three loss functions being tested. Additionally an anchor, created by processing the input through a \tanh nonlinearity, was included. The users were asked to assign a score out of 100 to each of the test items, based on how accurately they emulated the reference.

Four different input sounds were used during the test. The sounds were clips of bass guitar, taken from the test set used during the objective evaluation. Bass guitar sounds were chosen over guitar sounds, as preliminary listening tests indicated that they resulted in differences between the RNN models and the reference that were much more audible.

Trials were conducted separately for both of the LSTM hidden sizes being tested. This resulted in a total of eight MUSHRA trials in each test. In total, 14 people, with no reported hearing impairments, participated in the listening tests. The tests were conducted in sound-proof booths using Sennheiser HD-650 headphones.

The results of the listening test are shown in Fig. 5, with the mean score for each method and the 95% confidence in-

Table 2. Mean results of MUSHRA listening test

Hidden Size	Mean MUSHRA Score		
	H_{None}	H_{HP}	H_{AW}
32	74	76	79
64	77	82	86

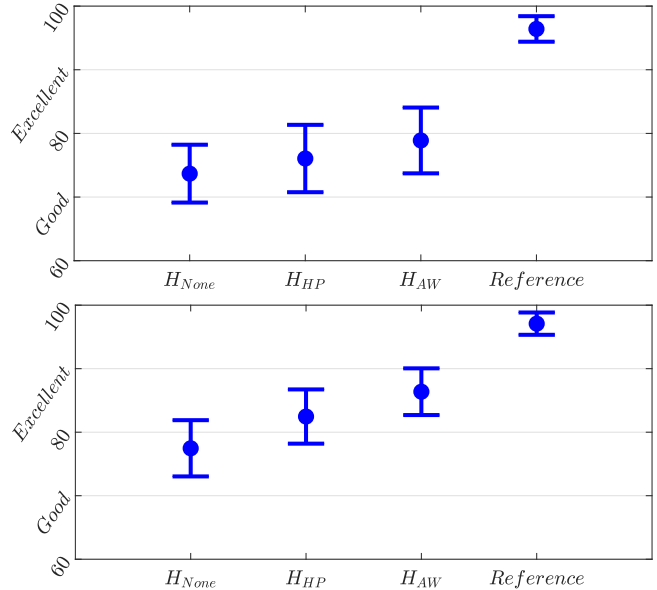


Fig. 5. Mean results of MUSHRA test with 95% confidence interval, for LSTM hidden size 32 (top) and 64 (bottom).

terval being shown. The anchor achieved a mean score of 3 out of 100, but has not been included in the plot so that the test results can be observed more clearly. For both the LSTM hidden sizes tested, a trend can be observed, with the no pre-emphasis filtering case performing worst, and the H_{HP} and H_{AW} pre-emphasised cases performing second best and best respectively. For the models with an LSTM hidden size of 64, the use of the H_{AW} pre-emphasised loss over the non pre-emphasised loss results in a statistically significant improvement in perceived emulation quality. However, in the other cases the differences are not statistically significant.

6. CONCLUSION

This work has evaluated the use of different pre-emphasis filters during the training of a previously proposed RNN model, used for nonlinear audio circuit modelling. The pre-emphasised loss achieved by each training condition was compared. Listening test results indicate that the use of a pre-emphasis filter can improve the model’s perceptual similarity to the target device, with the A-weighting filter performing best. As the pre-emphasis filter is only used during training, the improvement in perceptual similarity comes at no extra computational cost when running the resulting model.

7. REFERENCES

- [1] V. Välimäki, S. Bilbao, J. O. Smith, J. S. Abel, J. Pakarinen, and D. Berners, “Virtual analog effects,” in *DAFX: Digital Audio Effects*, U. Zölzer, Ed., pp. 473–522. Wiley, Chichester, UK, second edition, 2011.
- [2] J. Pakarinen and D. T. Yeh, “A review of digital techniques for modeling vacuum-tube guitar amplifiers,” *Computer Music J.*, vol. 33, no. 2, pp. 85–100, 2009.
- [3] M. Karjalainen and J. Pakarinen, “Wave digital simulation of a vacuum-tube amplifier,” in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP-06)*, Toulouse, France, May 2006, vol. 5, pp. 153–156.
- [4] F. Santagata, A. Sarti, and S. Tubaro, “Non-linear digital implementation of a parametric analog tube ground cathode amplifier,” in *Proc. Int. Conf. Digital Audio Effects (DAFx-07)*, Bordeaux, France, Sept. 2007, pp. 169–172.
- [5] R. C. D. Paiva, S. D’Angelo, J. Pakarinen, and V. Välimäki, “Emulation of operational amplifiers and diodes in audio distortion circuits,” *IEEE Trans. Circ. Syst. II*, vol. 59, no. 10, pp. 688–692, Oct. 2012.
- [6] K. J. Werner, V. Nangia, A. Bernardini, J. O. Smith III, and A. Sarti, “An improved and generalized diode clipper model for wave digital filters,” in *Proc. Audio Eng. Soc. 139th Conv.*, New York, NY, Oct. 2015.
- [7] A. Novak, L. Simon, P. Lotton, and J. Gilbert, “Chebyshev model and synchronized swept sine method in non-linear audio effect modeling,” in *Proc. Int. Conf. Digital Audio Effects (DAFx-10)*, Graz, Austria, Sept. 2010, pp. 423–426.
- [8] S. Orcioni, A. Terenzi, S. Cecchi, F. Piazza, and A. Carini, “Identification of Volterra models of tube audio devices using multiple-variance method,” *J. Audio Eng. Soc.*, vol. 66, no. 10, pp. 823–838, Oct. 2018.
- [9] R. Kiiski, F. Esqueda, and V. Välimäki, “Time-variant gray-box modeling of a phaser pedal,” in *Proc. Int. Conf. Digital Audio Effects (DAFx-16)*, Brno, Czech Republic, Sept. 2016, pp. 31–38.
- [10] F. Eichas and U. Zölzer, “Gray-box modeling of guitar amplifiers,” *J. Audio Eng. Soc.*, vol. 66, no. 12, pp. 1006–1015, Dec. 2018.
- [11] E.-P. Damskögg, L. Juvela, E. Thuillier, and V. Välimäki, “Deep learning for tube amplifier emulation,” in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP-19)*, Brighton, UK, May 2019, pp. 471–475.
- [12] E.-P. Damskögg, L. Juvela, and V. Välimäki, “Real-time modeling of audio distortion circuits with deep learning,” in *Proc. Int. Sound and Music Computing Conf. (SMC-19)*, Malaga, Spain, May 2019, pp. 332–339.
- [13] Z. Zhang, E. Olbrych, J. Bruchalski, T. J. McCormick, and D. L. Livingston, “A vacuum-tube guitar amplifier model using long/short-term memory networks,” in *Proc. IEEE SoutheastCon*, Saint Petersburg, FL, Apr. 2018.
- [14] A. Wright, E.-P. Damskögg, and V. Välimäki, “Real-time black-box modelling with recurrent neural networks,” in *Proc. Int. Conf. Digital Audio Effects (DAFx-19)*, Birmingham, UK, Sept. 2019.
- [15] T. Schmitz and J.-J. Embrechts, “Nonlinear real-time emulation of a tube amplifier with a long short time memory neural-network,” in *Proc. Audio Eng. Soc. 144th Conv.*, Milan, Italy, May 2018.
- [16] Blackstar Amplification, “HT-1 guitar amplifier – Product page,” Available online at: <https://www.blackstaramps.com/uk/ranges/ht-1>, Accessed: 2019-04-05.
- [17] J. D. Markel and A. H. Gray, Jr., *Linear Prediction of Speech*, Springer-Verlag, Berlin, Heidelberg, 1980.
- [18] IEC, “Electroacoustics – Sound level meters,” International Standard IEC 61682:2013, International Electrotechnical Commission, Geneva, CH, 2013.
- [19] A. Rimell, N. Mansfield, and G. Paddan, “Design of digital filters for frequency weightings (A and C) required for risk assessments of workers exposed to noise,” *Industrial Health*, vol. 53, no. 1, pp. 21–27, Jan. 2015.
- [20] IET, “BS.1534 : Method for the subjective assessment of intermediate quality levels of coding systems,” Recommendation ITU-R BS.1534-1, International Telecommunication Union, Geneva, CH, 2015.
- [21] M. Schoeffler, S. Bartoschek, F.-R. Stöter, M. Roess, S. Westphal, B. Edler, and J. Herre, “WebMUSHRA – A comprehensive framework for web-based listening tests,” *J. Open Research Software*, vol. 6, no. 1, Feb. 2018.