PROCEEDINGS of the
23rd International Congress on Acoustics
9 to 13 September 2019 in Aachen, Germany

Blind estimation of reverberation time by neural networks

Rodrigo L. PRATES*, Mariane R. PETRAGLIA†, Julio C. B. TORRES‡, Antonio PETRAGLIA§

Program of Electrical Engineering, COPPE
Federal University of Rio de Janeiro, Brazil

ABSTRACT

In this paper we investigate a procedure for blind estimation of room reverberation time, that is, without a priori knowledge of the room impulse response. A neural network based method is proposed, which is robust to noise and to abrupt variations over time of the acquired signal. The input features of the neural network are calculated in the frequency domain, where training samples are generated in the Mel scale. A competent network architecture (i.e., with appropriate number and type of hidden layers and number of neurons comprising each layer), as well as the best training algorithm and regularization technique, are investigated. The use of cross-entropy error during training gives better results than those obtained by mean squared error. Comparative results are presented, considering two previously proposed blind methods: a deep neural network technique and a spectral decay distribution method.

Keywords: Reverberation Time, Neural Network, Power Spectral Density

1 INTRODUCTION

The reverberation time (RT_{60}) in a room is defined as the time the sound takes to decay 60 dB after the interruption of the sound emission. Its estimation is an important task in the characterization of sound quality of closed spaces. It contributes to the assessment of the quality and intelligibility of audio signals obtained from reverberant environments, and to the improvement of audio processing algorithms and speech recognition methods.

The reverberation time was initially calculated by Sabine [1], through the information of geometry of the environment and coefficients of absorption of the surfaces that compose the environment. If such information is not available, a controlled test signal can be generated to estimate the RT_{60} based on the noise-canceling method [2], or by the Schoroeder integration curve method [3], which uses the measured Room Impulse Response (RIR). Since the RIR may not be available, and the test signal may be difficult to implement, other ways of computing the RT_{60} have to be explored. Thus, in many applications it becomes necessary to determine the RT_{60} only from the recorded speech signal.

Several approaches have been proposed for the estimation of the RT_{60}, among which there are some methods that use maximum likelihood (ML) to model the decay of the reverberant signal [4], [5] and others that apply the reverberant signal decay distribution (Spectral Decay Distribution - SDD), calculated through the energy envelope of the RIR [6]. There are also techniques that use the frequency domain by modeling the power spectral density (PSD) through an IIR filter, whose pole is related to the reverberation time [7]. Finally, some papers use Machine Learning (ML) techniques, such as Deep Neural Networks (DNN) [8]. These techniques have already demonstrated significant improvement in problem solving such as voice recognition, pattern recognition in images and estimation of arrival direction [2].

In this work, the estimation of the reverberation time in a room is treated as a problem of mapping the reverberant signal to an estimate of the RT_{60} using DNN, based on the work reported in [2]. A classification network is used to select the most likely RT_{60}, given an input vector formed by the concatenation of the Mel

*rodrigolp01@gmail.com
†mariane@pads.ufrj.br
‡julio@poli.ufrj.br
§antonio@pads.ufrj.br

coefficients [9] of the reverberant signal. The reverberant signals were generated from simulated RIRs by the image method [10], and the addition of Gaussian white noise with different signal-to-noise ratios (SNRs). The results obtained with the implementation of the method based on DNN are compared to those achieved by the algorithms considered state-of-the-art techniques. In this work the same RT_{60} is assumed for all frequency bands, which is a simplification adopted for all implemented methods.

2 PROBLEM FORMULATION

A signal $y(n)$ recorded in a room consists of a direct sound $s(n)$, followed by reverberation $x(n)$ and possibly involved in noise $v(n)$, which can be expressed as

$$y(n) = s(n) + x(n) + v(n), \tag{1}$$

where n is the discrete-time index. The reverberant signal $x(n)$ can be expressed as

$$x(n) = h(n) * s(n), \tag{2}$$

where $*$ is the convolution operator and $h(n)$ represents the room impulse response (RIR). According to the *Polack* model [2], [7], the RIR can be described as one realization of the stochastic process

$$h(n) = b(n)e^{-\eta n}, \ n \geq 0, \tag{3}$$

where $b(n)$ is a zero mean Gaussian noise with variance σ_b^2, which defines the RIR structure modulated by an exponential function with decay rate η. This rate is defined as [7]

$$\eta = \frac{3ln(10)}{RT_{60}f_s}, \tag{4}$$

where f_s is the sampling frequency.

The decay rate η can be estimated by applying a linear fit to the natural logarithm of the time-frequency energy envelope. In [6] the decay rate of the energy envelope is blindly estimated from the observed reverberation speech signal in the short-time Fourier transform domain, using a property of the spectral decay distribution.

The method that applies neural networks also separates the signal into frames, but instead of using this information directly, the frames are transformed through coefficients of the Mel frequency scale [2]. These coefficients are used as input data for the neural network training, which will learn to estimate the reverberation time associated with the frame. In this way, several frames are concatenated until they form a signal segment large enough for the estimation to be performed.

3 REVERBERATION TIME ESTIMATION BY NEURAL NETWORKS

In this section the blind and supervised method for estimating RT_{60} based on deep neural networks (DNN) is presented. The method maps portions of an input signal into a value corresponding to the RT_{60} estimate. A sorting network with multiple outputs is used, where each output represents a set of RT_{60} values. Thus, given the input samples, the network selects one of the outputs as the most likely to be the class associated with the range of reverberation time values of the environment, and from the index of that output provides the estimated value of RT_{60}. The main motivation for using deep neural networks lies in their ability to generalize the learning process, thereby efficiently modeling complex problems. The DNN architecture employed in this work comprises an input layer, three hidden layers, and an output layer. In this section, the input, output and training data of the deep neural network will be presented.

3.1 Input Data

As input parameters, we used the coefficients obtained by passing the input signal through a filter bank on the Mel scale (Log Mel Filterbank Energies- LMFBE) [2, 8, 6]. This choice is due to the property that the Mel scale correctly represents the response of the human auditory system. The steps for generating the features, or input data, are described below [2, 9]. It is assumed that the signals are all sampled at the rate of 16 kHz.

1. Initially a pre-emphasis filter is applied to the reverberant speech signal, with coefficient $\alpha = 0.97$, that is,
$$y(n) = x(n) - \alpha x(n-1);$$
2. The signal is then divided into frames of 25ms, with 40% overlap;
3. The Hamming window
$$w(n) = 0.54 - 0.46cos(\frac{2\pi n}{N-1}),$$
is applied to each frame, where N is the window size;
4. The Fast Fourier Transform (FFT) of length 512 is applied to each frame in order to compute the power spectrum by
$$P = \frac{|FFT(x_i)|^2}{N},$$
where x_i is the i-th frame of the signal x. Finally, a bank of 23 triangular filters T_{256x23} of 256 coefficients is applied to the power spectrum of each frame. Turning each frame F_{1x23} into an array of 23 coefficients, that is,
$$F_{1x23} = f_{1x256} \times T_{256x23}.$$
Still with each frame, the base 10 logarithm is applied and the mean is removed for spectrum balancing and improvement on the signal-to-noise ratio [9].

In order to obtain the mapping between the LMFBE coefficients and the RT_{60}, it is necessary to concatenate a sufficient number of frames. Thus, a sequence of 51 continuous frames, proportional to the longer time to be estimated, is created as the input vector for DNN.

3.2 Output Data

It is usual to treat the estimation as a regression problem. DNN can be trained to minimize the mean squared error (MSE) between the predicted and the true RT_{60} of the training set. However, according to [2], such regression solution tends to generate high estimation errors within the range where RT_{60} is estimated. Thus, the predicted value is skewed around the center of the training reverberation time scale. This issue is a direct result of the regression choice as a way of treating the problem, and of the use of the MSE as a cost function for DNN training. To avoid this difficulty, we can treat the estimation as a classification problem, where we divide the values of RT_{60}, on the scale from 0.1 s to 1.0 s, into 19 bins, one per class for each bin [2]. We then have, for example, the output vector $\mathbf{y} = [0\ 1\ 0\ \cdots\ 0]^T$ in case the value 175 ms is the true RT_{60}.

3.3 Neural Network Training

With defined network input and output parameters, the DNN is trained to learn the best mapping from the input values to the output values. For each hidden layer of the network, the data undergoes an affine transformation through the weight matrix and the bias vector, and then by the sigmoid activation function. The last layer generates the output vector of the DNN using the softmax function. In this way, the output vector is given by

$\mathbf{p} = [p_1(\mathbf{o_t}), \ldots, p_N(\mathbf{o_t})]^T$, where $p_i(\mathbf{o_t})$ is the a posteriori probability of the i-th bin given the input feature $\mathbf{o_t}$, N is the number of RT_{60} classes/bins and t is the frame index. The value of RT_{60} is then estimated by [2]

$$\hat{RT}_{60} = \frac{\sum_{i=j-1}^{j+1} p_i(\mathbf{o_t}) c_i}{\sum_{i=j-1}^{j+1} p_i(\mathbf{o_t})}, \tag{5}$$

where c_i is the center of the i-th bin.

There is a compromise between the width of the bin and the number of classes. The larger the bin width, the more accurate the classification, but the worse the RT_{60} resolution. A total of 19 bins were used to cover the interval from 0.2 s to 1.0 s, with a bin of 0.042 s width, according to [2].

4 METHODOLOGY AND SIMULATIONS

In this section, the procedures for generating the network training and validation data, the methodology used for network calibration, and the simulated experiments for the evaluation of the algorithms will be presented.

4.1 Database Generation

A set of data was generated for training and validation of the estimating algorithms of $RT60$ through the VCTK-Corpus database. For the generation of impulse responses (RIRs) we applied the image method, for which 7 parameters were used: sound velocity in m/s, sampling frequency in $samples/s$, receiver position, source position and room dimension in [x, y, z] m, reverberation time in s, and number of samples.

A methodology was chosen so as to vary the parameters of the imaging method, and thus to generate different RIRs. In this procedure, 3 different room sizes were defined: small room $d = [4,3,4]$ m, average room $d = [6,4,4]$ m and large room $d = [10,7,4]$ m. Two possible distances between source and receiver, that is, close (1.5m) and distant (3.0m), were adopted. For sound velocity and sampling frequency, the values $c = 340$ m/s and $f_s = 16$ kHz, respectively, were set.

The positioning of both the source s and the receiver r were chosen to start at $[2,2,2]$ m. Then, an orientation to be altered is chosen randomly, which may either be x or y. The distance between source and receiver were randomly chosen as well. Finally, the displacement was applied to the position of the receiver, thereby verifying the room size, so that the receiver would not remain at an invalid position.

For the generation of training and test data, 3 second voice frames were used, generating a total of 9,586 voice files of 152 different people. From this amount of files, approximately 5% (479 files) were used for generating the final results, which contained voice signals from people who participated in the training and from people who did not participate. Data from the remaining 9,107 files were used for network calibration, separated into sets containing 70% for training, 15% for validation, and 15% for testing. The following steps were applied in order to generate the network input data:

1. Resample the speech signal at the rate of 16 kHz, and obtain a 3 s segment, with the removal of the initial silence part;
2. Convolve the signal segment with the RIR generated by the image method to obtain the reverberant signal. The RIR is generated by random selection of the following parameters: RT_{60} from 0.2 to 1.0 s, room size and distance between source and receiver;
3. Scale the signal to the range of $[0,1]$;
4. Add white Gaussian noise to the reverberant signal, whose level in dB is also chosen randomly among the following values: $[0,5,10,30]$ dB for the training signals, and $[-10,0,10,20]$ dB for the test signals;
5. Generate the Mel coefficients as explained above, as well as the output vectors. This step is performed 5 times on the same signal, in order to generate additional data for training and validation.

This generation was made in such a way as to guarantee a data balance between classes. The methodology applied to evaluate the results are presented below.

4.2 Training Procedures and Comparison among Models

In this work the following network architecture was adopted by means of exploring and proposing improvements to the method presented in [2]: (1) number of neurons of the input layer equal to the size of the input vector; (2) update of the weights and bias by the backpropagation algorithm with momentum and adaptive learning rate; (3) MSE and cross-entropy cost functions; (4) learning rate, number of neurons (in the range of $[25,400]$) and regularization values to be defined in the network calibration. Accordingly, the network calibration must provide the optimal values of number of hidden layer neurons, cost function type, learning rate value and regularization value. Comparison between MSE and cross-entropy networks was also performed, where the other parameters were tested until the best configuration was obtained for each network model. The best result considered in the calibration process corresponds to the one that generated the smallest mean square error, whereas comparisons with the other models proposed in the literature correspond to the average absolute error (which are the metrics available in the other works).

The following are the steps of the calibration process of the two network models:

1. Initially we sought the best learning rate lr from $[10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}]$ for 5 different networks, varying the number of neurons h among $[25, 50, 100, 200, 400]$.
2. After obtaining lr, the regularization λ is included, with values among $[10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}]$, using the best 3 values of h.
3. At this point, for each type of cost function, there are 3 network settings. The best of each is selected, obtaining a network configuration for each type.
4. The final comparison is performed by evaluating the absolute mean error for different room sizes, distance between the source and the receiver, and noise levels.

The selected model was compared to SDD and DNN reference methods, for different room sizes and noise levels. The RIRs were generated according to Eq. (3), where $b(n)$ is a stochastic process with variance $\sigma_b^2 = 0.1$. The following 6 values of RT_{60} were used to generate the synthetic RIRs: $[0.2, 0.33, 0.42, 0.6, 0.82, 1.0]$ s. Two voice signals (one female and one male) sampled at 16 kHz and two SNR values (0 dB and 20 dB) were employed.

The next section presents training results, generated from the methodology described above, and comparative conclusions with the other methods.

5 RESULTS

5.1 Network Selection

The networks were trained with either 5000 or 10000 runs. If convergence had not yet been observed with 5000 runs, training was continued up to 10000 runs. Figure 1 shows the convergence behaviors for the MSE and Cross-Entropy networks containing 200 neurons.

Tables 1 and 2 display the MSE values obtained after convergence for the MSE and Cross-Entropy networks, respectively, for different values of lr. The highlighted values correspond to the 3 smallest MSEs obtained. The term "tuned" refers to the automatic updating of the lr parameter.

Tables 3 and 4 show the MSE values obtained after convergence for the MSE and Cross-Entropy networks, respectively, for different values of λ and employing the tuned lr values. The highlighted values correspond to the smallest MSEs obtained.

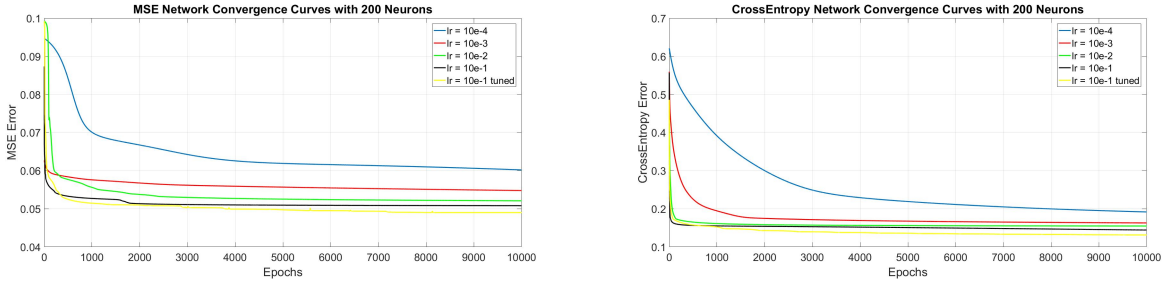

Figure 1. Convergence of MSE and Cross-Entropy networks for lr calibration.

Table 1. MSEs after convergence of the MSE network for different lr values.

# neurons \ lr	tuned	10^{-1}	10^{-2}	10^{-3}	10^{-4}
25	0.0558	0.0821	0.1157	0.0884	0.0888
50	**0.0517**	0.0751	0.0998	0.0973	0.1359
100	**0.0554**	0.0754	0.0853	0.0816	0.0736
200	**0.0530**	0.0726	0.0809	0.0859	0.0887
400	0.0673	0.0802	0.0762	0.0893	0.0642

Table 2. MSEs after convergence of the Cross-Entropy network for different lr values.

# neurons \ lr	tuned	10^{-1}	10^{-2}	10^{-3}	10^{-4}
25	**0.0427**	0.0691	0.0802	0.0820	0.2139
50	**0.0413**	0.0729	0.0787	0.0896	0.0957
100	**0.0425**	0.0733	0.0757	0.0801	0.0796
200	0.0511	0.0638	0.0749	0.0861	0.0805
400	0.0544	0.0643	0.0769	0.0799	0.0822

Table 3. MSEs after convergence of the MSE network for different λ values.

# neurons \ λ	10^{-1}	10^{-2}	10^{-3}	10^{-4}
50	0.0522	**0.0488**	0.0509	0.0494
100	**0.0530**	0.0549	0.0563	0.0613
200	**0.0581**	0.0605	0.0604	0.0534

Table 4. MSEs after convergence of the Cross-Entropy network for different λ values.

# neurons \ λ	10^{-1}	10^{-2}	10^{-3}	10^{-4}
25	0.0431	0.0428	**0.0394**	0.0427
50	**0.0408**	0.0428	0.0441	0.0476
100	0.0492	**0.0433**	0.0435	0.0438

Tables 5 and 6 contain the MAE values and the corresponding standard-deviations of the best MSE and Cross-Entropy networks, respectively, for different room sizes, source-receiver distances, and noise levels. It

Table 5. MAEs and their standard-deviation after convergence for the proposed MSE network.

ROOM	DIST	SNR					Mean Error
		-10 dB	0 dB	10 dB	20 dB	30 dB	
Small	Close	0.303 ± 0.177	0.151 ± 0.101	0.140 ± 0.093	0.122 ± 0.072	0.135 ± 0.086	0.175 ± 0.119
	Distant	0.294 ± 0.173	0.176 ± 0.107	0.129 ± 0.083	0.125 ± 0.078	0.105 ± 0.066	0.166 ± 0.114
Average	Close	0.294 ± 0.175	0.179 ± 0.113	0.135 ± 0.085	0.136 ± 0.088	0.112 ± 0.078	0.164 ± 0.116
	Distant	0.293 ± 0.189	0.202 ± 0.122	0.126 ± 0.083	0.126 ± 0.083	0.124 ± 0.0811	0.174 ± 0.122
Large	Close	0.305 ± 0.176	0.156 ± 0.099	0.146 ± 0.098	0.133 ± 0.086	0.133 ± 0.090	0.159 ± 0.110
	Distant	0.282 ± 0.177	0.160 ± 0.101	0.129 ± 0.082	0.111 ± 0.084	0.114 ± 0.081	0.151 ± 0.110
Mean Noise		0.295 ± 0.177	0.168 ± 0.106	0.132 ± 0.087	0.125 ± 0.082	0.117 ± 0.079	0.164 ± 0.115

Table 6. MAEs and their standard-deviations after convergence for the Cross-Entropy network.

ROOM	DIST	SNR					MEan Error
		-10 dB	0 dB	10 dB	20 dB	30 dB	
Small	Close	0.238 ± 0.134	0.156 ± 0.092	0.132 ± 0.088	0.126 ± 0.075	0.131 ± 0.082	0.162 ± 0.106
	Distant	0.228 ± 0.132	0.168 ± 0.099	0.123 ± 0.087	0.124 ± 0.080	0.103 ± 0.068	0.152 ± 0.103
Average	Close	0.243 ± 0.124	0.170 ± 0.101	0.132 ± 0.083	0.138 ± 0.089	0.107 ± 0.072	0.153 ± 0.101
	Distant	0.234 ± 0.134	0.185 ± 0.103	0.125 ± 0.082	0.117 ± 0.088	0.136 ± 0.086	0.162 ± 0.110
Large	Close	0.244 ± 0.146	0.144 ± 0.092	0.143 ± 0.093	0.129 ± 0.082	0.133 ± 0.086	0.150 ± 0.100
	Distant	0.221 ± 0.125	0.140 ± 0.087	0.121 ± 0.088	0.119 ± 0.079	0.112 ± 0.071	0.140 ± 0.094
Mean Noise		0.235 ± 0.134	0.159 ± 0.097	0.128 ± 0.084	0.122 ± 0.079	0.116 ± 0.074	0.153 ± 0.108

can be observed that the lowest MAE values were obtained for the "Large Room and Distant Source-Receiver" scenario for both networks. The Cross-Entropy network resulted in the smallest errors in all scenarios. It can also be noticed that the errors associated with the variation of noise level do not change abruptly, which is due to the robustness of the methods that use spectral information. In view of the results, the Cross-Entropy network model was chosen as the proposed model to be compared with the other blind and non-blind reference procedures.

5.2 Comparative Results

Figure 2 presents the MAE values obtained with the proposed Cross-Entropy network, DNN reference network [2], and with the SDD technique. The Cross-Entropy network presented lowest average error for all config-

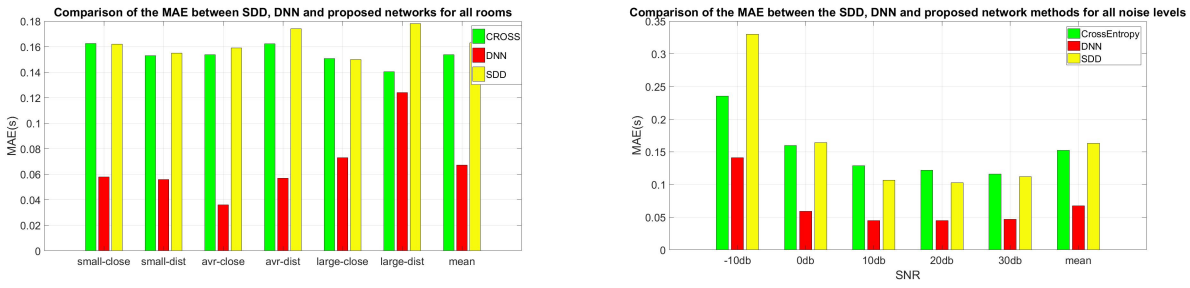

Figure 2. MAEs of SDD, proposed and reference DNN networks for different room sizes, source-receiver distances and noise levels.

urations when compared to the SDD approach, but worse than those obtained by the reference network. No information was provided in [2], such as number of neurons, use of regularization, algorithms for updating

weights and polarizations and the parameters used in the generation of RIRs by the image method. Therefore, the comparison with this method may not be fair.

6 CONCLUSION

The present work aimed to investigate the possibility of using neural networks for the estimation of reverberation time, by comparing their results with those of the SDD method and another DNN network, whose information was used to start the calibration of the proposed model. A methodology was created for the network calibration, where it was chosen to explore the use of cross-entropy cost function, commonly applied in classification networks. In this context, several parameter values were obtained experimentally, such as the number of neurons, the weight and bias updating algorithm, and the use of regularization and preprocessing of input data. The choice was made considering the smallest mean squared error. Regarding the model selection, this choice was made after evaluating the architecture that had the lowest mean absolute error, so as to be compatible with the results of the reference methods, in several configurations of room sizes, source-receiver distances, noise levels and ranges of RT_{60}. It was observed that, in almost all configurations, the cross-entropy network was the one that generated the lowest values of MAE and was therefore chosen for comparison with the other approaches. The network obtained in this study presented similar results as those of the SDD method, being higher in the average. In conclusion, the performed experiments confirmed the capacity of the neural networks for the estimation of the reverberation time, in a range of 0.2 s to 1 s, for different room conditions and signal-to-noise ratios. As future proposals we intend to explore new network topologies, such as recurrent and convolutional, and estimate other parameters, such as Early Decay Time (EDT), Speech Clarity Index (SCI) and Direct Reverberant Energy Ratio (DRR).

ACKNOWLEDGEMENTS

The authors thank the financial support of CAPES/DAAD PROBRAL Program for developing this work through project number 88881.198848/2018-01, and FAPERJ, Grant 202.844/2018.

REFERENCES

[1] Beranek, LWC. Sabine's personal papers. Proceedings of Meetings on Acoustics. 2009.

[2] Xiong X, Shengkui Z, Xionghu Z, Douglas L, Chng E, Haizhou L. Learning to estimate reverberation time in noisy and reverberant rooms. Proc INTERSPEECH; September 2015; pp 3431-3435.

[3] Schroeder, RM. New method of measuring reverberation time. The Journal of the Acoustical Society of America. 1965; 17(3):409-412.

[4] Ratnam R, Jones DL, Wheeler BC, O'Brien WD, Lansing Jr. CR, Feng AS. Blind estimation of reverberation time. The Journal of the Acoustical Society of America. 2003; 114(1):2877-2892.

[5] Löllmann HW, Yilmaz E, Jeub M, Vary P. Algorithm for blind reverberation time estimation. Proc IWAENC; August 2010; pp 1-4.

[6] Wen JYC, Habets EAP, Naylor PA. Blind estimation of reverberation time based on the distribution of signal decay rates. Proc ICASSP; March 2008; pp 329-332.

[7] Faraji N, Ahadi SM, Sheikhzadeh H. Reverberation time estimation based on a model for the power spectral density of reverberant speech. Proc EUSIPCO; August 2016; pp 1453-1457.

[8] Lee M, Chang J. Blind estimation of reverberation time using deep neural network. Proc IC-NIDC; September 2016; pp 308-311.

[9] https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html, Accessed: 2018-09-23.

[10] Allen JB, Berkley DA. Image method for efficiently simulating small-room acoustics. The Journal of the Acoustical Society of America. 1979; 65(4)943-950.