INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Dynamic Encoder RNN for Online Voice Activity Detection in Adverse Noise Conditions

Prithvi R.R. Gudepu[1], *Jayesh M. Koroth*[2], *Kamini Sabu*[3], *M A Basha Shaik*[4]

Samsung R&D Institute Bangalore, India

{p.gudepu, jayesh.mk, kamini.sabu, m.shaik}@samsung.com

Abstract

The majority of online Voice Activity Detection (VAD) models employ a Recurrent Neural Network (RNN) component to capture long context which helps to improve noise-robustness. These RNN components are static models which do not make efficient use of the model's predictions from previous frames. In this work, we introduce a new Dynamic Encoder RNN (DE-RNN) that encodes the target speech dynamically to facilitate distinguishing of target speech from noise. Experiments on different established baseline architectures by modifying their RNN component by the addition of DE-RNN, show improvement in both background noise and secondary competing speaker noise scenarios. We used publicly available datasets for experiments.

Index Terms: VAD, RNN, Encoder, Dynamic

1. Introduction

Voice assistants are an important interface between an AI system and humans for giving commands. It is deemed to eventually become the primary, if not the only, method. Hence, it is imperative for the components of voice assistants, including VAD, to work in all kinds of difficult scenarios possible. VAD is a pre-processing step to separate speech and non-speech portions of the audio by classifying each time frame into two classes. This helps downstream tasks like Automatic Speech Recognition (ASR) and speech enhancement to deal with only relevant audio signal portions.

The traditional VADs [1] [2] are known to work efficiently for audios in clean environments. They are based on thresholding short-time energy and other acoustic features. They fail in many real-life cases, especially where the noise is non-stationary and highly varying, due to the fixed thresholding nature of the methods [3]. Statistical models for VAD offered a substantial improvement over traditional methods [4] [5] [6] [7]. Recently neural network-based systems for VAD [8] [9] [10] [11] [12] [13] has been studied extensively and shown to benefit from non-linear operations to work well in difficult noisy scenarios. A challenging scenario for VAD is the presence of babble or speech-based noise due to the similar acoustic characteristics with the target speech. Such cases have been considered in [14] and [15]. The most challenging case is when the speech-based noise is not babble noise far in the background, but instead a secondary competing speaker close to the main speaker and is clearly intelligible making it a daunting task. Existing VAD architectures may not be able to handle such real-life scenarios [16].

Most of the VAD applications require both real-time lightweight online processing and robustness to challenging non-stationary noises. Convolutional Neural Networks (CNN) are good at extracting complex high-level features in addition to the commonly used STFT-based features [10] [12]. Attention mechanisms are good at relating different sections of the audio signal to leverage the context [9]. Online attention-based models are preferred in speech tasks for real-time requirements, but they face a trade-off between the low computation time and leveraging the full temporal context. RNN layers are also used in the VAD models to effectively use the context in systems with real-time requirements [8] [17]. The most popular RNNs used for long context are Long Short Term Memory (LSTM) and Gated Recurrent Units (GRU). However, they give more importance to recent frames processed compared to segments that occurred much earlier in time. This affects the most when the model needs to differentiate the target speech from the clearly intelligible secondary speech. Hence, it is essential to encode a representation of the target speech, which helps in distinguishing it from the rest of the signal.

RNNs have been used as encoders in different tasks like ASR [19], and machine translation. The input sequence is encoded into a representation vector used for further processing. Encoders became popular for sequential tasks when they were used in machine translation to represent the semantics of the sentence into a vector [20]. In speech-related tasks, encoders are widely used as a first step to transform the audio signal. Generally, encoders process the entire sequence which does not effectively capture the complex differentiating characteristics between speech and noise portions required for VAD.

A dynamic system can be used to encode only the speech frames predicted by the model. Dynamic Neural Networks [21] is an emerging research topic where the models, as opposed to static systems, can adapt their structure or parameters based on the input and model state during inference. Therefore, they have better representation power, efficiency, and interpretability than static models. Skip-RNN [22] updates a controlling signal in every step either to update or to copy the hidden state from the previous step. Structural-Jump-LSTM [23] has an agent to make the skipping decision conditioned on the previous state and the current input.

We employ a system to selectively encode only the target speech portion and skip the non-speech portion, for the model to learn a representation of target speech. We make use of the model's output for a particular frame as a trigger to make the skip decision dynamically. If the frame is predicted as speech, we process the frame to update the encoded representation; but if predicted as non-speech, we skip the frame. We experiment on publicly available datasets Librispeech [24], WHAM noise [25], and Audioset noise [26] from DNS challenge [27]. We simulate the noise conditions with and without a nearby secondary speaker who is slightly farther from the microphone than the main speaker. We modify different baselines with our pro-

10.21437/Interspeech.2023-2466

posed system and show improvement.

The novel contributions of this paper are listed below.

- The proposed method is the first of its kind to use *speech-only* encoder for VAD task.
- We dynamically use the model's output class prediction to isolate and learn a representation of target speech.
- We show that replacing RNN with DE-RNN in different baseline architectures improves their performance.
- Further, the challenging secondary speaker noise scenario is tested to find the improvement in the baseline system performance.

2. Architecture

In this section, we elaborate our proposed DE-RNN. We use a dynamic neural network model to learn the representation of target speech. As the model gets trained, it eventually learns to categorize every frame into speech (1) or non-speech (0) class.

An Encoder RNN layer in the proposed system runs alongside the prediction RNN layer of the baseline model as shown in Figure 1. The different steps involved are summarized below.

1. At time frame t, the prediction layer accepts the input (the concatenation of the features extracted from the speech frame at t and the hidden state of the Encoder RNN) and outputs softmax probabilities for speech and non-speech class.
2. Based on the predicted class, a decision is taken whether to unroll the encoder RNN cell for time frame t.
3. If the frame is predicted as non-speech class, the encoder RNN cell unroll is disabled. If the frame is predicted as speech class, it takes the feature frame at time t as input and unrolls the RNN cell. This decision-making is illustrated in the control algorithm 1.

As shown in Figure 1a, there is a control system after the prediction layer which decides the behavior of the encoder RNN. It can be noted that since the RNN cell does not process each of the time frames, it computationally costs less than a regular RNN layer, depending on the proportion of non-speech frames. Figure 1b illustrates the skipped updates to encoder RNN on non-speech frames.

2.1. Why Dynamic System?

Static RNN encoders process the entire temporal sequence. Since there is no feedback from the model predictions, they have to independently learn the distinguishing features of speech from noise. Using the model predictions helps reuse the already computed information (classified speech portions) to encode a representation of target speech. Since this encoded information embedded in the hidden state of the encoder RNN is passed as an additional feature to the prediction layer, the prediction layer can internally learn to compare the input feature with this encoded target speech characteristics for similarities to do a better classification of frames.

3. EXPERIMENTS

3.1. Data

We use three datasets for simulating the required training and testing datasets. Utterances from Librispeech corpus are used as clean speech utterances. The noise audios are taken from WHAM and Audioset. WHAM corpus consists of noises recorded in different environments like restaurants, cafes, bars, and parks which cover the regularly occurring noises in public areas. Audioset is a collection of about 2 million human-labeled sound clips of length 10 seconds drawn from YouTube videos and belongs to 600 different audio events. This covers diverse range of possible noises encountered.

Room Impulse Responses (RIRs) are generated by varying different acoustic conditions such as microphone position, speaker position, reverberation factor, and room dimensions as shown in Table 1. We simulate the datasets using RIRgenerator [28] to cover various scenarios encountered in real-life. We retain the default mic and speaker settings provided by the pyroomacoustics for generating RIR.

At time frame t:
Prediction_class$[t]$ = Prediction_RNN(embedding, Feature$[t]$)
if *Prediction_class*$[t]$ == *"1"* **then**
  Encoder_RNN unroll(Feature$[t]$),
  Update(Encoder_RNN_hiddenstate)
else
  Encoder_RNN_hiddenstate[t] = Encoder_RNN_hiddenstate[t-1]
end
embedding= Encoder_RNN_hiddenstate$[t]$

Algorithm 1: Control algorithm for the proposed system

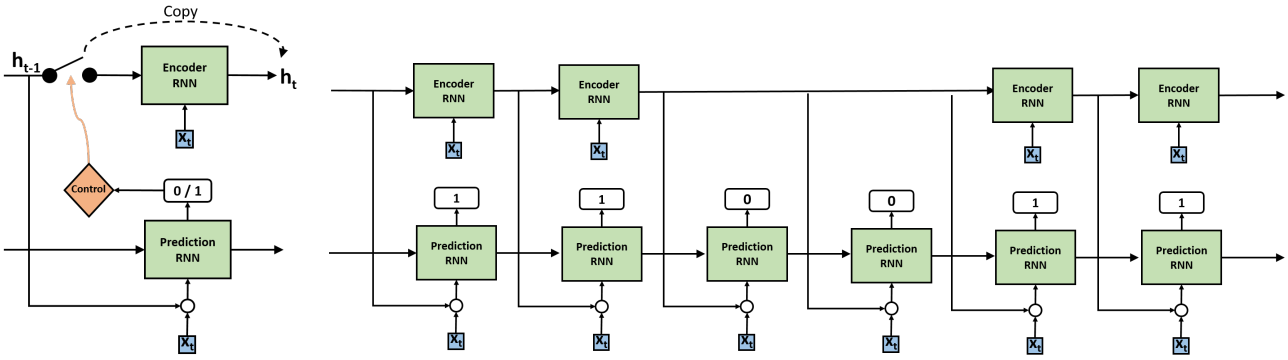
Table 1: *Configurations for simulating the noisy reverberated data.*

Room	x	U[6,10]
	y	U[6,10]
Absorption factor		U[2.5,5]
Mic. Pos [m]	x	$\frac{x_{room}}{2}$ + U[-0.5,0.5]
	y	U[0,0.5]
Source Pos. [m]	x	$U[0, x_{room}]$
	y	U[0, $3\frac{x_{room}}{4}$]
Secondary Source Pos. [m]	x	$U[0, x_{room}]$
	y	U[0, x_{room}]
dist(mic, sec. source) > dist(mic, source)		
SNR [dB] (with sec. source)		U[1,10]
SNR [dB] (mix with noise)		U[-3,20]

3.1.1. Datasets

We simulate three different datasets spanning three different noise scenarios.

1. Dataset-1 (Background Noise): This is the single-speaker scenario with no competing speaker. Librispeech dataset speech files are taken as the ground truth. The speech files are convolved with RIR and then mixed with noise files from WHAM corpus. The mixing was done with SNR chosen from the $[-3, 20]$ dB range. The methodology followed is similar to data generation in [29].
2. Dataset-2 (Secondary Speaker and Background Noise): This is the most challenging case for VAD in a practical scenario. Here, a secondary speaker is close to the main speaker though a little farther from the mic. This case is frequently seen by voice assistants but they tend to fail here since both the signals are speech signals. It can be noted that the target speech is originated from a source closer to the mic, while the secondary speech (which is the noise component here) is originated from a source further away. This characteristic needs to be leveraged by the model to differentiate between them.



(a) *Control system skips processing of non-speech frames by the encoder RNN and selectively processes speech frames*

(b) *The dynamic unrolling of the RNN cells in the DE-RNN system*

Figure 1: *The architecture of the proposed algorithm*

One-half of the audio files considered in this case are created with the same setup used for the Dataset-1. For the other half, an additional speaker, taken from the Librispeech dataset itself, is added in the RIR simulation. For significant amount of noise in the signal, the secondary speech signal must be of similar length to the target speech. Librispeech dataset is sorted according to audio lengths. For each of the target speech, the adjacent speech signal in the sorted list is chosen as secondary speech. It should be noted that the speaker ids of the target and secondary signal should be different. It is also ensured that a target signal in one simulation configuration will likely be chosen as a secondary signal in a different configuration.

3. Dataset-3 (Secondary Speaker and Diverse Noise): A similar setup to Dataset-2 is used here. But, instead of WHAM, the Audioset corpus is used to cover a wide range of noise types. The noise corpus is representative of real-world scenarios consisting of both synthetic and real recordings.

Librispeech dataset used for our data generation is not balanced with respect to the silence vs speech ratio. Therefore, the silence regions greater than 300 ms are elongated by adding 0s such that the speech and silence lengths are balanced. For this, the silence regions are detected using WebRTC-VAD [30]. An end silence of 3 seconds is added at the end of each audio before convolving with RIR. The speakers and the noises were divided prior to the simulation, to have different speakers and noise in the train and test. In each dataset, 30 hrs of training, 5 hrs of validation, and 5 hrs of testing audios were generated.

3.1.2. RIR simulation

The mic is positioned at the middle of one side of the room. The two types of RIR configurations used here are shown in Figure 2.

Secondary speaker case: For each target source audio, four secondary speakers at the two different position configurations are used, totaling 8 RIR simulations per target source. Source and noise are chosen such that their lengths are similar. Noise audios are slightly pushed in time by adding silences (of length chosen from U[1, 3] seconds) at the beginning to have noise portions after the speech ends. Then zeros are padded to equate both lengths as required. The position of the main speaker is randomly chosen within 3/4th room length from the mic. After the main position is chosen, random positions are picked until we get the distance of the Noise source from the mic greater than that with the main speaker. This characteristic is important to differentiate the main speaker at uniform distribution at a possibly larger distance than the source radii from the mic.

Single speaker case: For each source audio, 8 different position configurations are used, totaling 8 RIRs simulations per target source with speaker position chosen randomly anywhere in the room.

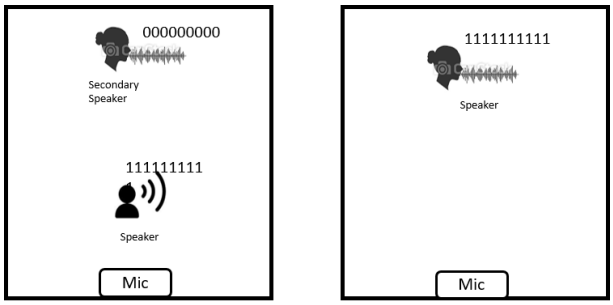


Figure 2: *The two types of room setups shown for secondary speaker and single speaker case*

3.2. Baseline Architectures

To assess the efficacy of the proposed DE-RNN, established baselines are implemented with the RNN component modified to DE-RNN system. The parameters and layer sizes of each individual model are set such that the computation times are small and comparable. For each of the baselines, we trained three models: with 1 layer RNN, with 2 layer RNN, and with DE-RNN.

- Gated Recurrent Unit VAD [31]: GRU is a simpler alternative to LSTM which does not have a separate cell state but only a hidden state. We use 100 dimensions GRU. 40-dimensional log-mel spectrograms are used as features with a frame step of 20 ms.
- Long Short-Term Memory network based VAD [32]: LSTMs are widely used recurrent neural networks that help in learn-

ing long-term dependencies. We use 100 dimensions hidden state LSTM. 40-dimensional log-mel spectrograms are used as features with a frame step of 20 ms.

- Hybrid CNN-LSTM VAD based on [29]: This consists of two 2D-convolutional layers each followed by MaxPool, a Dense layer followed by LSTM layer. The layer sized used as mentioned [29]. 32 x 32 spectrogram images are used as features. They are formed by stacking 32 of 32-dimensional log mel-filterbank energies with a frame step of 20ms.
- ResectNet [33]: State-of-the-art system for lightweight online VAD that does feature extraction by Sinc Convolutions and Resectnet blocks followed by LSTM layer. The layer size used as mentioned [33]. It operates in time-domain with 640 sample frame length (40 ms) with a 320-sample (20 ms) frame-shift is used.

3.3. Training Setup

In all the models, after the final RNN layer, a Dense layer with softmax activation is added to give probabilities for speech and non-speech classes. We use a Stochastic Gradient Descent optimizer with learning rate of 0.0005. Categorical cross-entropy loss is used. For 1-layer RNN and 2-layer RNN models, we use a batch size of 64. For DE-RNN, due to the dynamic nature of the system, a batch size of 1 is used. The base model can be pre-trained with a larger batch size before modifying to DE-RNN for faster convergence. Tensorflow v2.11 [34] is used for training the models.

4. Results

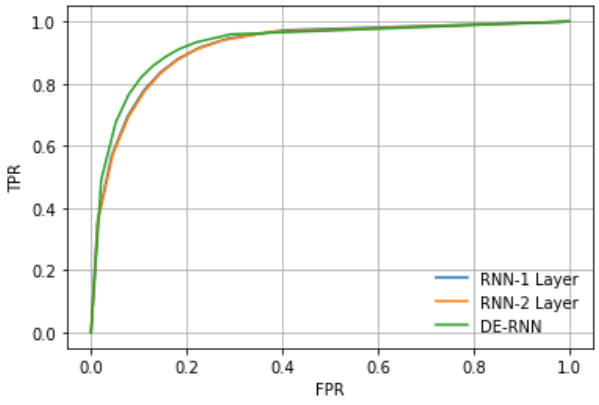


Figure 3: *ROC curve for ResectNet systems on Dataset-3*

We experimented on four baseline systems with 1-layer RNN, 2-layer RNN and DE-RNN and plotted the ROC (Receiver Operating Characteristics) curves for demonstrating the performance of the proposed method. In Figure 3, ROC is plotted for the baseline ResectNet model and the proposed DE modification. The x-axis corresponds to the False Positive Rate (FPR) and the y-axis corresponds to the True Positive Rate (TPR).

As shown in Table 2, in the baseline systems, the additional layer of RNN did not give any substantial improvement. On the other hand, modifying the baseline by the addition of a dynamic encoder layer increases the AUC values indicating the improvement in the classification ability. Also, the computations are lesser than an additional RNN layer since the encoder layer in DE-RNN skips the non-speech frames.

Table 2: *VAD accuracy in Area Under Curve percentage for the baseline systems with 1 RNN layer (L1), 2 RNN layers (L2) and their modified version with the addition of DE-RNN layer are tabulated. The performance of the models are tested for Background noise case, Secondary speaker case and Diverse noise case.*

Model	Dataset-1	Dataset-2	Dataset-3
GRU L1	89.10	90.59	88.39
GRU L2	88.91	90.20	88.56
GRU DE-RNN	**90.38**	**92.70**	**90.20**
LSTM L1	89.24	90.57	88.20
LSTM L2	88.86	90.88	87.37
LSTM DE-RNN	**90.48**	**92.69**	**90.03**
Hybrid cnn-LSTM L1	87.40	90.60	88.10
Hybrid cnn-LSTM L2	87.10	91.47	85.70
Hybrid cnn-LSTM DE-RNN	**91.65**	**93.90**	**90.60**
ResectNet L1	87.20	90.60	88.10
ResectNet L2	88.80	90.30	88.60
ResectNet DE-RNN	**91.47**	**94.40**	**89.72**

5. Conclusion

In this paper, we proposed a Dynamic Encoder RNN to dynamically learn a representation of the target speech using the model outputs from previous frames. We modified the RNN component of different VAD models, including state-of-the-art architectures, by incorporating our proposed Dynamic Encoder RNN system and showed that the proposed modification improves AUC of the ROC curve in different noise conditions. We tested and showed improvements for the general background noisy cases as well as the secondary speaker cases which is the most challenging real-life scenario for VAD task

6. References

[1] J. Sohn and W. Sung, "A voice activity detector employing soft decision based noise spectrum adaptation," in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181)*, vol. 1. IEEE, 1998, pp. 365–368.

[2] J. Sohn, N. S. Kim, and W. Sung, "A statistical model-based voice activity detection," *IEEE signal processing letters*, vol. 6, no. 1, pp. 1–3, 1999.

[3] S. S. Meduri and R. Ananth, "A survey and evaluation of voice activity detection algorithms," 2012.

[4] J.-H. Chang, N. S. Kim, and S. K. Mitra, "Voice activity detection based on multiple statistical models," *IEEE Transactions on Signal Processing*, vol. 54, no. 6, pp. 1965–1976, 2006.

[5] D. Ying, Y. Yan, J. Dang, and F. K. Soong, "Voice activity detection based on an unsupervised learning framework," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 8, pp. 2624–2633, 2011.

[6] Z. Ali and M. Talha, "Innovative method for unsupervised voice activity detection and classification of audio segments," *Ieee Access*, vol. 6, pp. 15 494–15 504, 2018.

[7] S. Alimi and O. Awodele, "Voice activity detection: Fusion of time and frequency domain features with a svm classifier," *Computer Engineering and Intelligent Systems*, vol. 13, no. 3, pp. 20–29, 2022.

[8] T. Hughes and K. Mierle, "Recurrent neural networks for voice activity detection," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 7378–7382.

[9] J. Kim and M. Hahn, "Voice activity detection using an adaptive context attention model," *IEEE Signal Processing Letters*, vol. 25, no. 8, pp. 1181–1185, 2018.

[10] A. Sehgal and N. Kehtarnavaz, "A convolutional neural network smartphone app for real-time voice activity detection," *IEEE Access*, vol. 6, pp. 9017–9026, 2018.

[11] A. Sofer and S. E. Chazan, "Cnn self-attention voice activity detector," *arXiv preprint arXiv:2203.02944*, 2022.

[12] N. Wilkinson and T. Niesler, "A hybrid cnn-bilstm voice activity detector," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6803–6807.

[13] G. Gelly and J.-L. Gauvain, "Minimum word error training of rnn-based voice activity detection," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[14] Y. Ma and A. Nishihara, "Efficient voice activity detection algorithm using long-term spectral flatness measure," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2013, no. 1, pp. 1–18, 2013.

[15] K. Srinivasan and A. Gersho, "Voice activity detection for cellular networks," in *Proceedings., IEEE Workshop on Speech Coding for Telecommunications,*. IEEE, 1993, pp. 85–86.

[16] J. H. Hansen, A. Joglekar, M. C. Shekhar, V. Kothapally, C. Yu, L. Kaushik, and A. Sangwan, "The 2019 inaugural fearless steps challenge: A giant leap for naturalistic audio," *ISCA INTERSPEECH-2019*, 2019.

[17] G.-B. Wang and W.-Q. Zhang, "An rnn and crnn based approach to robust voice activity detection," in *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2019, pp. 1347–1350.

[18] M. Schak and A. Gepperth, "A study on catastrophic forgetting in deep lstm networks," in *Artificial Neural Networks and Machine Learning–ICANN 2019: Deep Learning: 28th International Conference on Artificial Neural Networks, Munich, Germany, September 17–19, 2019, Proceedings, Part II 28*. Springer, 2019, pp. 714–728.

[19] A. Zeyer, P. Bahar, K. Irie, R. Schlüter, and H. Ney, "A comparison of transformer and lstm encoder decoder models for asr," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 8–15.

[20] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[21] Y. Han, G. Huang, S. Song, L. Yang, H. Wang, and Y. Wang, "Dynamic neural networks: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 7436–7456, 2021.

[22] V. Campos, B. Jou, X. Giró-i Nieto, J. Torres, and S.-F. Chang, "Skip rnn: Learning to skip state updates in recurrent neural networks," *arXiv preprint arXiv:1708.06834*, 2017.

[23] C. Hansen, C. Hansen, S. Alstrup, J. G. Simonsen, and C. Lioma, "Neural speed reading with structural-jump-lstm," *arXiv preprint arXiv:1904.00761*, 2019.

[24] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[25] G. Wichern, J. Antognini, M. Flynn, L. R. Zhu, E. McQuinn, D. Crow, E. Manilow, and J. L. Roux, "Wham!: Extending speech separation to noisy environments," *arXiv preprint arXiv:1907.01160*, 2019.

[26] R. at Google. [Online]. Available: https://research.google.com/audioset/index.html

[27] C. K. Reddy, V. Gopal, R. Cutler, E. Beyrami, R. Cheng, H. Dubey, S. Matusevych, R. Aichner, A. Aazami, S. Braun *et al.*, "The interspeech 2020 deep noise suppression challenge: Datasets, subjective testing framework, and challenge results," *arXiv preprint arXiv:2005.13981*, 2020.

[28] R. Scheibler, E. Bezzam, and I. Dokmanić, "Pyroomacoustics: A python package for audio room simulation and array processing algorithms," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 351–355.

[29] A. Sehgal and N. Kehtarnavaz, "A hybrid cnn-bilstm voice activity detector," *IEEE Access*, vol. 6, pp. 9017–9026, 2018.

[30] S. Salishev, A. Barabanov, D. Kocharov, P. Skrelin, and M. Moiseev, "Voice activity detector (vad) based on long-term mel frequency band features," in *Text, Speech, and Dialogue: 19th International Conference, TSD 2016, Brno, Czech Republic, September 12-16, 2016, Proceedings 19*. Springer, 2016, pp. 352–358.

[31] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[32] P. Sertsi, S. Boonkla, V. Chunwijitra, N. Kurpukdee, and C. Wutiwiwatchai, "Robust voice activity detection based on lstm recurrent neural networks and modulation spectrum," in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2017, pp. 342–346.

[33] O. Köpüklü and M. Taseska, "ResectNet: An Efficient Architecture for Voice Activity Detection on Mobile Devices," in *Proc. Interspeech 2022*, 2022, pp. 5363–5367.

[34] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/