Analysis of RoyalFlush Submission in INTERSPEECH 2020 Far-Field Speaker Verification Challenge

Chenyi Yu[1], *Xinhui Hu*[1], *Xinkang Xu*[1], *Chien-Lin Huang*[2]

[1]Hithink RoyalFlush AI Research Institute, Zhejiang, China
[2]PAII Inc., Palo Alto, CA, USA

{yuchenyi, huxinhui, xuxinkang}@myhexin.com, chiccocl@gmail.com

Abstract

This study shows the post-evaluation analysis of our efforts in INTERSPEECH 2020 Far-Field Speaker Verification Challenge (FFSVC 2020). There are one task of far-field text-dependent speaker verification from single microphone array, and one task of far-field text-independent speaker verification from single microphone array, and another task of far-field text-dependent speaker verification from distributed microphone arrays in this challenge. Our systems were based on x-vectors with different front-end feature extraction methods and neural network topologies. The score fusion was used to combine different system results. On the FFSVC 2020 evaluation set, we achieved the minimum detection cost function (minDCF) of 0.70, 0.86, and 0.68 which are Equal Error Rate (EER) of 7.77%, 8.97%, and 7.53% for task1, task2, and task3, respectively. We also achieved the log-likelihood ratio based cost metric (*Cllr*) of 0.29, 0.33, and 0.29 for these three tasks, respectively, ranking second on the leaderboard among all participants for task1 and task3, ranking third on the leaderboard for task2 in mid-term submission.

Index Terms: speaker verification, far-field, cross channel matching, distributed microphone array, speaker embedding

1. Introduction

Speaker verification (SV) is one of biometric authentication methods like iris scanning, facial recognition and fingerprinting sensing. SV is a process of verifying speaker's identification, that is, based on speaker's existing utterances, to judge whether an utterance belongs to the target speaker [1]. Since speech based human machine interaction has become popular in many fields such as smart home, mobile devices and automobiles, speaker verification demonstrates its important role in these applications. SV can be classified into text-dependent and text-independent tasks according to the applications [2]. In terms of technical composition, a complete SV system can be divided into acoustic feature phase, model phase and scoring phase. With the continuous increase in data and computations, SV technique has been making great progresses. The i-vector model [3] and neural network based speaker embedding methods [4] have demonstrated high performances when they are compared with the traditional approaches. However, in the contexts of noisy environment, short-term utterance verification, and far-field conditions [5], there are still great challenges for both text-dependent and text-independent SV tasks, in particularly in the far-field and complex environment [6]. Recent years, many challenges related to speaker recognition have been held to promote the speaker recognition technology. With these challenges, the evaluation methods are also developed in diversity and maturity. For example, the evaluation metric on speaker recognition system changed from EER to minDCF. From a practical point of view, the minDCF reflects system performance more in line with common usage habits because the impact of false acceptance and false rejection which are embodied by the EER on system usage is inconsistent.

The INTERSPEECH 2020 Far-Field Speaker Verification Challenge was designed to boost the SV research focusing on far-field distributed microphone arrays under noisy conditions in real scenarios [6]. In this paper we will present our analyses on multiple SV systems, which were based on neural networks with various speaker embeddings including the state-of-the-art speaker embedding x-vector systems [7]. We will also analyze the impact of different front-end feature analysis, training data, data augmentation, and back-end scoring for far-field data represented in the FFSVC 2020 benchmarks. The main objective of this study is to provide a description and analysis of our submission to the FFSVC 2020 challenge. This paper is organized as follows. Section 2 introduces our system setup including dataset, feature analysis and speaker embedding neural network topologies. Section 3 describes the experimental results and analyses. Finally, Section 4 concludes this work.

2. System Setup

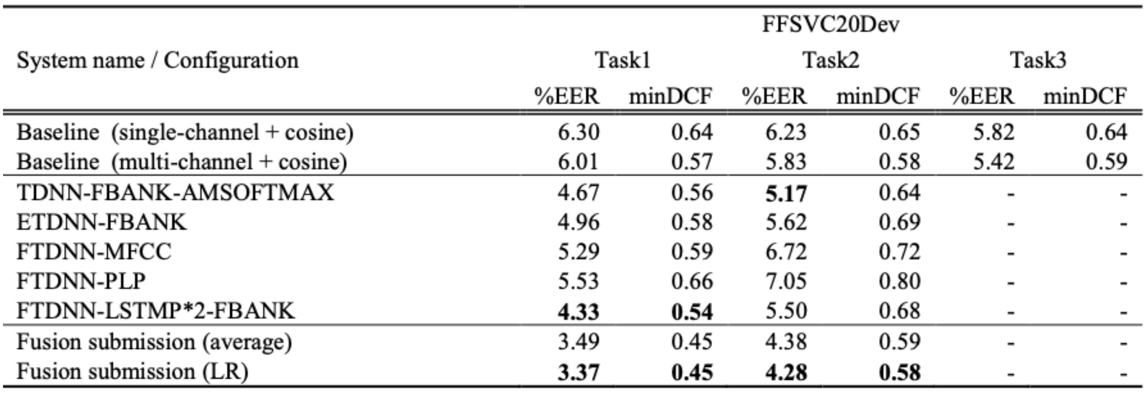
In this section we provide a description of all the components used in our systems. We have set up five SV systems based on x-vector architectures. In the acoustic features phase of the system, F-bank, MFCC and PLP were all used as the acoustic features. In the systems of TDNN and ETDNN, the F-bank was used as the input features. In the phase of model, we also explored to improve system performance by adding LSTM layer as in [8]–[10] and used additive margin softmax (AM-softmax) loss function in the neural network architecture [11]. In the scoring phase of system, we adopted the score discrimination method of multi-channel score average and multi-model fusion to calibrate and optimize the system at the score level.

2.1. Training data and augmentations

In our systems, we only used the training set of the FFSVC20 (FFSVC20Train) as training data. The FFSVC20Train consists of about 1,400,000 utterances, including 120 speakers, and its total duration is over 1,100 hours. The close-talking recorded speech in 48k Hz and 16 bits among the training data was down-sampled to 16k Hz and 16 bits.

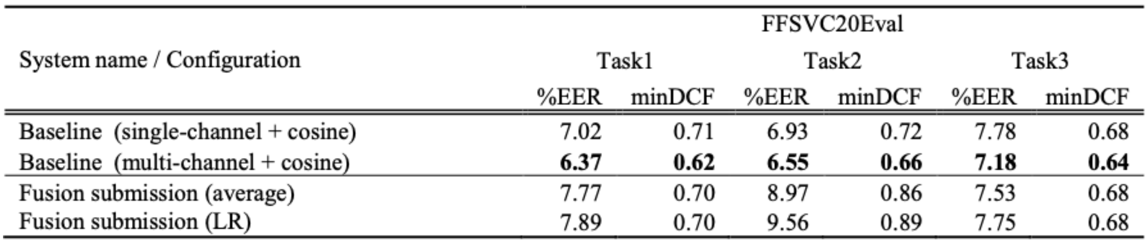
The conventional data augmentation in Kaldi for ASR can be realized by using simulated room impulse responses (RIRs) [35], producing multiple versions of the original signal with

Table 1: Analysis of the systems on the development set (FFSVC20Dev) of the FFSVC 2020 challenge.



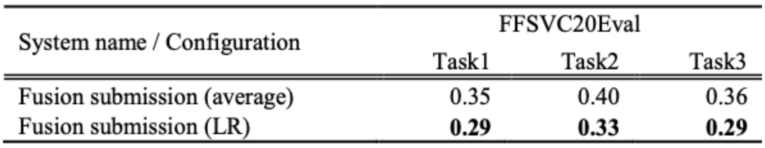
System name / Configuration	FFSVC20Dev					
	Task1		Task2		Task3	
	%EER	minDCF	%EER	minDCF	%EER	minDCF
Baseline (single-channel + cosine)	6.30	0.64	6.23	0.65	5.82	0.64
Baseline (multi-channel + cosine)	6.01	0.57	5.83	0.58	5.42	0.59
TDNN-FBANK-AMSOFTMAX	4.67	0.56	**5.17**	0.64	-	-
ETDNN-FBANK	4.96	0.58	5.62	0.69	-	-
FTDNN-MFCC	5.29	0.59	6.72	0.72	-	-
FTDNN-PLP	5.53	0.66	7.05	0.80	-	-
FTDNN-LSTMP*2-FBANK	**4.33**	**0.54**	5.50	0.68	-	-
Fusion submission (average)	3.49	0.45	4.38	0.59	-	-
Fusion submission (LR)	**3.37**	**0.45**	**4.28**	**0.58**	-	-

Table 2: Analysis of the systems on the evaluation set (FFSVC20Eval) of the FFSVC 2020 challenge.



System name / Configuration	FFSVC20Eval					
	Task1		Task2		Task3	
	%EER	minDCF	%EER	minDCF	%EER	minDCF
Baseline (single-channel + cosine)	7.02	0.71	6.93	0.72	7.78	0.68
Baseline (multi-channel + cosine)	**6.37**	**0.62**	**6.55**	**0.66**	**7.18**	**0.64**
Fusion submission (average)	7.77	0.70	8.97	0.86	7.53	0.68
Fusion submission (LR)	7.89	0.70	9.56	0.89	7.75	0.68

Table 3: *Cllr* performance on the evaluation set (FFSVC20Eval) of the FFSVC 2020 challenge.



System name / Configuration	FFSVC20Eval		
	Task1	Task2	Task3
Fusion submission (average)	0.35	0.40	0.36
Fusion submission (LR)	**0.29**	**0.33**	**0.29**

different speed factors [33], or adding noises to clean speech data [34]. The SpecAugment is a simple data augmentation method which is realized by frequency masking and time masking, it has been verified effective for improving speech recognition by Google [12]. The augmentation is directly applied to the feature inputs of a neural network, it can be easily utilized as an on-the-fly augmentation method. This method is also regarded useful in getting good results and for speaker recognition systems with high efficiency [13]. We compared the above two data augmentation methods, and we found that the computation of Kaldi's data augmentation is much higher than the SpecAugment. So, we adopted the SpecAugment for our data augmentation.

2.2. Development and evaluation data

The development set of FFSVC20 (FFSVC20Dev) consists of about 370,000 utterances, including 35 speakers, and its total duration is about 300 hours. We used it as a reference for evaluation during our developments. For each task of the challenge, an unlabeled evaluation set (FFSVC20Eval) and its corresponding trials files (containing 121,200 pairs of trial, referring to 80 speakers) were provided for the attendees. In experiments, we used the FFSVC20Dev for adaptation to reduce domain mismatch between the FFSVC20Train and the FFSVC20Eval, because FFSVC20Dev and FFSVC20Eval were from the same database [1].

2.3. Front-end feature analysis

All the close-talking (48kHz, 16 bits) records were resampled to 16kHz and pre-emphasized before feature extraction. At the same time, weighted prediction error (WPE) [14, 15] methods were used for de-reverberation for the circular microphone arrays records in both FFSVC20Dev and FFSVC20Eval. We employed log Mel-filterbank (F-bank) as the main acoustic features which is popular in speaker recognition. The MFCC features and PLP features were used since they were complementary in identifying information. All the acoustic features were 40-dimensions with a frame length of 25ms and hop size of 10ms. All the extracted features were mean-normalized before feeding into the deep speaker network.

2.4. Voice activity detection

During training x-vector extractors, an energy-based voice activity detection (VAD) was used since allowing a certain amount of noise during training helps improve the robustness of neural networks [16]. The NN-based VAD has been proven useful during the evaluation stage [17], but it did not work in our systems.

2.5. X-vector extractors

It has been shown that SV systems based on TDNN structures have achieved good performance. Therefore we focused on

TDNN and its variants to extract x-vectors. Following neural network architectures and features were used in our systems. All the networks were trained with mini-batch size of 64, initial learning rate of 0.001 and final learning rate of 0.0001 for 6 epochs.

2.5.1. TDNN-FBANK-AMSOFTMAX

A standard x-vector system as described in [7, 18] was constructed and the AM-softmax was used as its discriminative classification loss. Since speaker verification in open settings is essentially a metric learning problem, AM-softmax has stronger requirements for correct classification than the common softmax, and it is able to generate an additive classification margin between embeddings of different classes. In this system, the TDNN architecture had 9 layers.

2.5.2. ETDNN-FBANK

Compared with the standard TDNN, the ETDNN architecture [19] has a wider temporal context with alternating dense and convolutional layers in the frame-level hidden layers. Meanwhile, each time-delay layer of ETDNN has different dilation factors or kernel size. The ETDNN architecture increased to 14 layers compared to the TDNN architecture.

2.5.3. FTDNN-MFCC

Compared with ETDNN, FTDNN has fewer parameters because it factorize one weight matrix into two low-rank matrices [20]. The first matrix is constrained to be semi-orthogonal in order to retain the main information. To further reduce risk of gradient vanishing of deeper networks, FTDNN introduces skip connections [21] between the low-rank interior layers, where previous layers are concatenated to form the input of current layer. The FTDNN architecture also has 14 layers, but it has a faster training speed than the ETDNN.

2.5.4. FTDNN-PLP

This system was consistent with the above system FTDNN-MFCC's configuration, the only difference is that the acoustic features were replaced with PLP. This system was expected to enrich the acoustic features of the final fusion system.

*2.5.5. FTDNN-LSTMP*2-FBANK*

We proposed to insert recurrent layers in the frame level layers in order to better capture the long-range dependencies in speech than using a feed-forward structure alone as in conventional x-vector systems. We combined FTDNN and two LSTM layers into a unified architecture as shown in [16] and took F-bank as input. Besides, the statistics pooling layer were replaced by a self-attention layer as shown in [22]. The self-attention mechanism enabled speaker embedding to be focused on important frames and to obtain long-term speaker representation with higher discriminative power. As a result, the final network had 16 layers.

2.6. Backend LDA-PLDA scoring

The probabilistic linear discriminant analysis (PLDA) [23] served as the back-end scoring method. The back-ends consisted of LDA with dimension reduction to 200, centering, whitening, length normalization, PLDA and score normalization [24].

2.7. Score normalization

After finishing the above procedures, the scores of all single systems were normalized and calibrated before fusion. We used AS-Norm [25] and PLDA to further weaken the out-of-domain problem at score stage. We needed to introduce another dataset as cohort dataset to realize AS-norm. The cohort dataset should be as similar as the evaluation dataset, but it cannot contain the same utterances or speakers in the evaluation dataset. Specifically, the utterances of the adaptive cohort dataset were selected from the development dataset of SLR85 (SVC2019Dev) [26] by using the top *N* scores, where *N* was set to 1000 for all systems.

2.8. Calibration and score fusion

System fusion is usually a good way to improve system performance for building an effective SV system. For our tasks, the fusion and calibration were performed by using linear logistic regression from the Bosaris toolkit [27], and the SVC2019Dev was used for calibration and tuning fusion parameters. We also explored another fusion method called average fusion strategy. We found the average fusion strategy was a stable and efficient method according to our experience. Both of the above fusion methods would be used, and the final submission systems were obtained by the fusion of all subsystems at the score level.

3. Results and Analysis

3.1. Evaluation metrics

In this challenge, several metrics were used to evaluate the system performance. Unlike most conventional evaluations for SV systems, the minimum Detection Cost Function (minDCF) was taken as the primary metric. In addition, Equal Error Rate (EER) and log-likelihood-ratio cost function (*Cllr*) were also provided to participants as auxiliary metrics. Here, the EER is the operating point in the detection error trade-off (DET) curve where both the miss and false alarm rates are equal. DCF is a weighed sum of the missed detection and false alarm error probabilities shown as follows:

$$DCF = C_{Miss} \times P_{Miss|Target} \times P_{Target}$$
$$+C_{FalseAlarm} \times P_{FalseAlarm|NonTarget} \times (1 - P_{Target}) \quad (1)$$

where $C_{Miss} = 10$, $P_{Target} = 1$ and $C_{FalseAlarm} = 0.01$ in SRE-2008, and $C_{FalseAlarm} = 0.001$ in SRE-2010 [28]. The log-likelihood-ratio (LLR) cost function, *Cllr*, is computed as follows:

$$Cllr = \frac{1}{2\times log(2)} \times (\frac{\sum log(1+1/s)}{N_{Target}} + \frac{\sum log(1+s)}{N_{NonTarget}})(2)$$

where *s* is the likelihood ratio for a trial. N_{Target} and $N_{NonTarget}$ are the number of target and nontarget trials in the evaluation set, respectively [29].

3.2. Official baseline systems

A baseline system for the FFSVC 2020 was provided in [6]. In this baseline system, a pyroomacoustics toolkit [30] was used to simulate the room acoustics and generate far-field training data. Instead of using an energy-based VAD, a gradient boosting algorithm-based voice activity detection (GVAD) [31] was utilized. The GVAD was trained on a simulated far-field speech dataset which was originated from the AISHELL-1 dataset (SLR33) [32]. The 64-dimensional log Mel-filterbank energies were extracted with mean-normalization. The ResNet-34 structure [33] was adopted as the speaker embedding extractor. Networks were pre-trained with large scale (10,554 speakers) datasets of close-talking and simulation data including SLR33, SLR38, SLR47, SLR49, SLR62, and SLR68 from openslr.org. The learning rate was divided by 10 every 20 epochs.

3.3. Experimental analyses and discoveries

We listed the performances of all single systems on FFSVC20Dev in Table 1. The best result was in the form of bold face. Although the cosine scoring is a common method in SV, our tasks' performances embodied by it were modest. Therefore, in this challenge, we mainly adopted relevant methods of PLDA. It was observed that keeping the same amount of close-talking data and far field data from FFSVC20Train and FFSVC20Dev to train PLDA model could reduce the domain mismatch. For processing multi-channel data, we found that beamforming processing [34] was not currently the best method. Therefore, we drew on the idea of top-n, since the utterances for evaluation came from the circular microphone arrays which four recording channels were used. In scoring process, we first calculated the multi-channel records separately, then averaged the scores of the same recorded utterance from the same circular microphone array, and took the averaged score as the final score in the trial file.

3.4. FFSVC 2020 Dev baseline and submission results

We demonstrated the performances of all single systems on FFSVC20Dev in Table 1. It shows that the best single system in task1 was the FTDNN-LSTMP*2-FBANK, which achieved 0.54 of minDCF and 4.33% of EER. The best single system in task2 was the TDNN-FBANK-AMSOFTMAX, which achieved 0.64 of minDCF and 5.17% of EER. Compared with the standard FTDNN with 14 layers, the FTDNN-LSTMP*2-FBANK had two more layers of LSTM and adopted the attention pooling, which enabled itself to learn higher-level representations, especially in text-dependent tasks. Although the standard TDNN had a shallow structure and a small amount of parameters, TDNN still showed a strong classification ability by relying on AM-softmax loss function when the amount of training data is not particularly large, especially in text-independent tasks.

As can be seen from Table 1, a fusion of all subsystems yields minDCF of 0.45 and EER of 3.37% by Fusion (LR) method. Compared with the baseline result of FFSVC20Dev, our best system improves the minDCF by 21.05% and the EER by 43.93% relatively. Meanwhile, for the best system of task2, a fusion of all subsystems yields minDCF of 0.58 and EER of 4.28% by Fusion (LR) method. Compared with baseline result on FFSVC20Dev, our best system improves the EER by 26.59% relatively and the minDCF was basically consistent with the baseline system. In addition, we use the evaluation results of task1 to evaluate the relative performance of task3, because task1 and task3 belong to far-field text-dependent speaker verification from microphone arrays.

3.5. FFSVC 2020 Eval baseline and submission results

The experimental results of the submitted fusion systems were accordingly shown in Table 2. The best result is in the form of bold face. By analyzing Table 1 and Table 2, we found that the performance of our system on the FFSVC20Dev was obviously better than that on the FFSVC20Eval, this was due to that we selected part of the FFSVC20Dev data to train the PLDA model, and resulted in the phenomenon of overfitting. However, the PLDA model overfitting on the FFSVC20Dev was also beneficial to the performance on the FFSVC20Eval, we can still predict the performance on the FFSVC20Eval through the system performance evaluation results on the FFSVC20Dev. On the other hand, since the FFSVC20Dev and the FFSVC20Eval were composed of different speakers of the same dataset, we conducted additional experiments to prove that selecting a part of the data from the FFSVC20Dev and adding it to the PLDA model training can further improve the performance of the whole system. Although the minDCF and EER of our systems do not exceed the baseline system, we believe that if we increase the amount of training data resources, our system would achieve better performance.

Table 3 showed the performance of the FFSVC20Eval in the respect of *Cllr* metric. By analyzing Table 2 and Table 3, it was demonstrated that although the performance of our systems were not as well as the baseline in the respect of metrics minDCF and EER, the *Cllr* of our systems achieved 0.29, 0.33 and 0.29 for task1, task2 and task3, respectively. We hope to get the SV systems with stable performance, so our systems were calibrated across all operating points well and could get more discriminative scores between target and nontarget trials, from the viewpoint of metric *Cllr*, our task1 and task3 both ranked second on the leaderboard among all participants, and task2 ranked third on the leaderboard among all participants.

4. Conclusions

In this study, we explored several neural network based speaker embeddings for the INTERSPEECH 2020 Far-Field Speaker Verification Challenge. We provided an overview of the RoyalFlush systems submitted to this challenge. We described details about our systems including datasets, various x-vector extractors, back-end models and fusion strategies. On the FFSVC20Eval, we achieved the minDCF of 0.70, 0.86, and 0.68 which were EER of 7.77%, 8.97%, and 7.53% for task1, task2, and task3 in mid-term submission, respectively. Our systems showed very good results in the metric of log-likelihood-ratio cost function, *Cllr*, which were 0.29, 0.33, and 0.29 for task1, task2, and task3, respectively. Good *Cllr* scores denote high discrimination between target and nontarget scores. What we learned from FFSVC 2020 include: a small amount of data can quickly build SV system but system performance was limited; adding LSTM layers and using AM-softmax loss function to the neural networks significantly improved the performance; adjusting the composition of the back-end training data can reduce the domain mismatch; and finally, fusion can effectively improve performance of the submitted SV system.

5. References

[1] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5115–5119, 2016.

[2] C. Zhang, and K. Kazuhito. "End-to-End text-independent speaker verification with triplet loss on short utterances." in *Proceedings of INTERSPEECH*, pp. 1487–1491, 2017.

[3] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[4] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *Proceedings of INTERSPEECH*, pp. 999–1003, 2017.

[5] X. Qin, D. Cai, and M. Li, "Far-field end-to-end text-dependent speaker verification based on mixed training data with transfer learning and enrollment data augmentation," in *Proceedings of INTERSPEECH*, pp. 999–1003, 2017.

[6] X. Qin, M. Li, H. Bu, R. K. Das, W. Rao, S. Narayanan, and H. Li, "The FFSVC 2020 evaluation plan," *arXiv preprint arXiv:2002.00387*, 2020.

[7] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5329–5333,2018.

[8] C.-L. Huang, "Exploring effective data augmentation with TDNN-LSTM neural network embedding for speaker recognition," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019.

[9] C.-P. Chen, S.-Y. Zhang, C.-T. Yeh, J.-C. Wang, T. Wang, and C.-L. Huang, "Speaker characterization Using TDNN-LSTM based speaker embedding," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

[10] C.-L. Huang, "Speaker characterization using TDNN, TDNN-LSTM, TDNN-LSTM-Attention based speaker embeddings for NIST SRE 2019," in *Odyssey 2020: The Speaker and Language Recognition Workshop*, 2020.

[11] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," IEEE Signal Processing Letters, vol. 25, no. 7, pp. 926–930, 2018.

[12] D. S. Park, W. Chan, Y. Zhang, C. C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:2904.08779*, 2019.

[13] S. Wang, J. Rohdin, O. Plchot, L. Burget, K. Yu, et al., "Investigation of specaugment for deep speaker embedding learning," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*,2020.

[14] T. Yoshioka and T. Nakatani, "Generalizition of multi-channel linear prediction methods for blind mimo impulse respense shortening," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 10, pp. 2707–2720, 2012.

[15] T. Nakatani, T. Yoshioka, K. Kinoshita, M.Miyoshi, and B. Juang, "Speech dereverberation based on variance-normalized delayed linear prediction," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1717–1731, 2010.

[16] R. Li, D. Chen, and W. Zhang, "Voiceai systems to NIST SRE19 evaluation: Robust speaker recognition on conversational telephone speech," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[17] B. J. Borgstrom, M. S. Brandstein, and R. B. Dunn, "Improving statistical model-based speech enhancement with deep neural networks," in *Proc. IWAENC*, 2018.

[18] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contextualize," in *Sixteeth Annual Conference of the International Speech Communication Association*, 2015.

[19] J. Villalba, N. Chen, D. Snyder, D. Garcia-Romero, A. McCree, et al., "The JHU-MIT system description for NIST SRE18," 2019.

[20] D. Snyder, J. Villalba, N. Chen, D. Povey, G. Sell, N. Dehak, and S. Khudanpur, "The JHU speaker recognition system for the VOiCES 2019 challenge," in *Proceedings of INTERSPEECH*, pp. 2468–2472, 2019.

[21] K. He, X. Zhang, S. Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[22] Y. Zhu, T. Ko, D. Snyder, B. Mak, and D. Povey, "Self-attentive speaker embedding for text-independent speaker verification," in *Proceedings of INTERSPEECH*, pp. 3573–3577, 2018.

[23] S. Ioffe, "Probabilistic linear discriminant analysis," in *European Conference on Computer Vision. Springer*, pp. 531–542, 2006.

[24] J. Villalba, N. Chen, D. Snyder, D. Garcia-Romero, A. McCree, et al., "State-of-the-art speaker recognition for telephone and video speech: The JHU-MIT submission for NIST SRE18," in *Proceedings of INTERSPEECH*, pp. 1488–1492, 2019.

[25] S. Cumani, P. D. Batzu, D. Colibro, C. Vair, P. Laface, and V. Vasilakakis, "Comparison of speaker recognition approaches for real applications," in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[26] X. Qin, H. Bu, and M. Li, "Hi-mia: A far-field text-dependent speaker verification database and the baselines," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7609–7613, 2020.

[27] N. Brummer and E. D. Villiers, "The BOSARIS toolkit: Theory, algorithms and code for surviving the new DCF," in *NIST SRE11 Speaker Recognition Workshop*, pp. 1-23, 2011.

[28] C.-L. Huang, C. Hori, H. Kashioka, and B. Ma, "Ensemble classifiers using unsupervised data selection for speaker recognition," in *Proceedings of INTERSPEECH*, 2012.

[29] M. K. Nandwana, J. van Hout,M.McLaren, C. Richey, A. Lawson, and M. A. Barrios, "The voices from a distance challenge 2019 evaluation plan," *arXiv:1902.10828*, 2019.

[30] R. Scheibler, E. Bezzam, and I. Dokmanic, "Pyroomacoustics: A python package for audio room simulation and array processing algorithms," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 351–355, 2018.

[31] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "Catboost: unbiased boosting with categorical features," in *Advances in Neural Information Processing Systems*, pp. 6638–6648, 2018.

[32] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, "AISHELL-1: an open-source mandarin speech corpus and a speech recognition baseline," *arXiv preprint arXiv:1709.05522*, 2017.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016.

[34] L. Mosner, P. Matejka, O. Novotny, and J. H Cernocky, "Dereverberation and beamforming in far-field speaker recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5254–5258, 2018.