From Speaker Verification to Deepfake Algorithm Recognition: Our Learned Lessons from ADD2023 Track3

Xiaoyi Qin[1,†], Xingming Wang[1,†], Yanli Chen[2], Qinglin Meng[2] and Ming Li[1,*]

[1]Data Science Research Center, Duke Kunshan University, Kunshan, China
[2]MaShang Consumer Finance Co.Ltd, Chongqing, China

Abstract

This paper presents our learned lessons from the ADD2023 track3, Deepfake Algorithm Recognition (AR). In recent years, speech synthesis has made remarkable progress, where it has become increasingly difficult for human listeners to differentiate between synthesized speech and genuine human speech. Previous research has demonstrated that improving the recognition of deepfake algorithms can significantly enhance spoofing detection. Therefore, the primary focus of this paper is to investigate deepfake algorithm recognition, with experiments conducted based on the ADD2023 challenge. Inspired by speaker verification, we approach deepfake algorithm recognition as an open-set task. We propose a center-based similarity maximum method for determining the category of deepfake algorithms. Finally, by combining the scores from multiple models at the score level, we achieve an impressive F1 score of 0.8312 on the evaluation set.

Keywords

Deepfake algorithm recognition, Speaker verification,

1. Introduction

Text-to-speech (TTS) and voice conversion (VC) systems have made significant progress in recent years, thanks to the progress made in deep learning techniques and the availability of large-scale corpora [1]. These advancements have enabled the generation of audio that is almost virtually indistinguishable from human speech, posing serious threats to human users and Automatic Speaker Verification (ASV) systems in terms of potential attacks and security vulnerabilities. To protect the integrity of ASV systems, audio anti-spoofing countermeasure (CM) systems are commonly employed to detect spoofing audios [2].

In recent years, significant efforts have been devoted to the audio anti-spoofing task, with notable challenges such as ASVSpoof [3, 4] gaining prominence. However, most of these endeavors have primarily focused on binary classification tasks, distinguishing between bona fide and spoof audios, while neglecting the recognition of deepfake algorithms. Recognizing deepfake algorithms presents a more challenging multi-classification problem, with the presence of unseen categories, compared to fake audio detection.

In this paper, we evaluate the performance of deepfake algorithm recognition, focusing on the ADD2023 track3 [5], which aims to recognize the specific deepfake algorithm that a fake utterance is generated from. The evaluation set consists of both known and unknown deepfake algorithms, making the task an open-set recognition problem. To address this challenge, we propose a center-based maximum similarity evaluation method inspired by speaker verification. In our approach, we adopt an 1D and 2D convolution-based backbone to extract highly discriminative embedding. And then, each test embedding is computed the similarity with each known class centroid embedding. Finally, by fusing scores from multiple models at the score level, we achieve an impressive F1 score of 0.8312.

2. Related Work

2.1. Anti-spoofing

Synthesized audio spoofing is generally considered a logical access (LA) attack [6]. In recent years, mainstream CM systems for synthetic speech detection usually comprise two modules: feature extraction and binary classification. The feature extraction module extracts applicable features suitable for synthetic speech detection tasks. The classifier module determines whether the test audio is bonafide or spoofed based on the extracted features. Therefore, the performance of the classification module heavily relies on feature extractors that can provide highly discriminative features. Features based on audio self-supervised learning models have been shown to be effective [7]. On the other hand, various models and network architectures have been used for back-end classifiers, some of them takes frequency acoustic features as input, including ResNet [8], SE-Net [9], LCNN [10], etc., while some take original waveform as input, such as RawNet2 [11], AASIST [12], etc. Additionally, there are also some angle-based loss functions used for audio anti-spoofing, such as

IJCAI 2023 Workshop on Deepfake Audio Detection and Analysis (DADA 2023), August 19, 2023, Macao, S.A.R
*Corresponding author.
†These authors contributed equally.
xiaoyi.qin@dukekunshan.edu.cn (X. Qin); xingming.wang@dukekunshan.edu.cn (X. Wang); yanli.chen@msxf.com (Y. Chen); qinglin.meng@msxf.com (Q. Meng); ming.li369@duke.edu (M. Li)

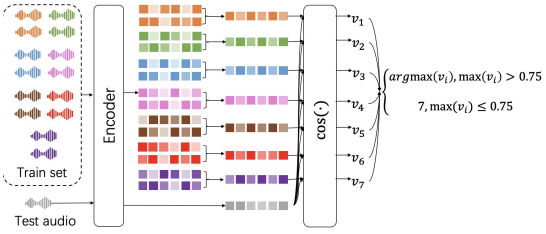


Figure 1: The implement pipeline of our proposed Center-based Maximum Similarity Method

AM-softmax [13] and Arcface [14], which are designed to increase the inter-class distance while reducing the intra-class distance. The OC-softmax is also adopted to customizes metric parameters for the different class distributions of bonafide and spoof audios [15].

The second Audio Deepfake Detection challenge (ADD 2023)[5] first sets up a track for deepfake algorithm recognition. Prior to that, only few researches focused on this task. M. Muller et al. presents several methods for creating attacker signatures using low-level acoustic descriptors and machine-learning embeddings for attacker attribution [16]. Zhu et al. uses separate module classifying attributes for the spoofing attack, which can also be helpful for binary spoof detection [17]. Li et al. used a multi-task learning approach to algorithm recognition as an auxiliary task for binary classification, which is proved effective [18].

2.2. Speaker verification

Automatic Speaker Verification (ASV) is a voice identity technology that determines whether two speech segments belong to the same person. To evaluate the performance of an ASV system, a trial list is provided, where each trial consists of an enrollment segment and a test segment. In the deep learning stage, we usually extract the speaker embedding, a fixed d-dimensional vector, from variable-length waveform. Then, we calculate the similarity between the enrollment embedding and the test embedding, which represents the probability that they belong to the same person. ASV is an open-set task, meaning that all test speakers are unseen during the training stage. On the other hand, deepfake algorithm recognition is a semi-open task, where both seen and unseen classes exist during the evaluation stage.In this case, we can adopt the method of speaker verification for deepfake algorithm recognition.

In this paper, we regard the deepfake algorithms recognition as a speaker verification task. Accordingly, we propose a center-based maximum similarity method to determine the category of test audio.

3. Center-based Maximum Similarity Method

The pipeline of our proposed method is illustrated in Figure 1. We employ a supervised learning approach to train a closed-set classification model on the training dataset. Subsequently, we feed all variable-length audio samples from the training set into the encoder of trained model, obtaining fixed-length d-dimensional embeddings $\mathbf{z} \in \mathbb{R}^d$. These embeddings correspond to the final linear outputs before the classifier. The classifier output can be seen as the result of inner product between the test embedding and the linear weight $\mathbf{w}_i \in \mathbb{R}^d$, where $\mathbf{w}_i$ can be consider the centroid i. When $\mathbf{z}$ and $\mathbf{w}_i$ are normalized, the classifier output is the cosine similarity between $\mathbf{z}$ and $\mathbf{w}_i$. However, the training process is not always stable, which may make center shifting. To overcome this, we calculate the average embedding for each class, obtaining stable intra-class centroids:

$$\mathbf{c}_k = \frac{1}{N}\sum_{i=0}^{N}\mathbf{z}_{k,i} \tag{1}$$

where $\mathbf{c}_k$ represents the embedding centroid of class k, and $\mathbf{z}_{k,i}$ corresponds to the embedding of the i-th sample from class k. Next, we evaluate the test embedding $\mathbf{z}^{test}$ by scoring it against all intra-class centroids. The scores are denoted as $v_i \in V$ and $i \in \{0,1,2,3,4,5,6\}$.

$$v_i = \cos(\mathbf{z}^{test}, \mathbf{c}_i) \tag{2}$$

where $\cos(\cdot)$ indicates the cosine similarity between test embedding and various intra-class centroids. During the training process, we employ a loss function with angular margin, specifically ArcFace [14], to optimize the embeddings. ArcFace not only incorporates cosine similarity during optimization, but also increases the inter-class margin while reducing intra-class variance.

Finally, if any score v_i surpasses the predefined centroid threshold, the test audio is assigned to the class with the highest score. Conversely, if all scores v_i fall below the threshold, the test audio is considered as belonging to an unknown class 7.

$$\begin{cases} \text{argmax}_{v_i \in V}(v_i), & \max(v_i) > 0.75, \\ 7, & \max(v_i) \leq 0.75. \end{cases} \tag{3}$$

The value 0.75 is a hyperparameter that has been obtained through tuning on the development set.

4. Implement Details

4.1. Network

In our deepfake algorithm recognition system, we employ a two-stage training approach: pre-training and large margin fine-tuning (LMFT). During the first stage, we set the scalar and margin of the angle softmax to 32 and 0.2, respectively. In the LMFT stage, we increase the margin to 0.4. To capture the distinctive features of deepfake audio,

we leverage various network architectures. Below are the specific details of each network utilized in our system:

ResNet34SimAM-ASP: We adopt the ResNet34SimAM-ASP [19] as our baseline system. This network utilizes the residual modules of ResNet and is equipped with the SimAM (Simple attention module). We use a thin ResNet structure with a backbone width of $\{32, 64, 128, 256\}$ to prevent overfitting on small-scale training data. The output feature maps are then processed using attentive statistics pooling (ASP) [20]. The output embedding has a dimension of 256.

ResNet34-GSP: ResNet34-GSP is based on the standard ResNet backbone, which is a well-known framework for 2D Convolutional Neural Networks (2D-CNNs) in image recognition tasks. We use the same backbone width as ResNet34SimAM-ASP. The encoding layer of ResNet34-GSP is based on global statistic pooling (GSP), and the embedding has a dimension of 256.

ResNet34SE-ASP: ResNet34SE-ASP is similar to ResNet34SimAM-ASP in terms of architecture but replaces the SimAM attention module with a squeeze-and-excitation (SE) module. The SE module aggregates global channel information as attention weights for all feature maps. The embedding dimension is also 256.

ECAPATDNN-ASP: ECAPA-TDNN [21] is a 1D-CNN model that has achieved great success in speaker verification. The SE-Res2Block is used to capture speaker features, and Multi-layer Feature Aggregation (MFA) is employed to process concatenated information before feeding it to the ASP. The resulting embedding has a dimension of 256.

LCNN: LCNN is a network commonly used for audio anti-spoofing, particularly in the ASVSpoof 2021 challenge [4]. The Max-Feature-Map (MFM) operation, based on the Max-Out activation function, is a key component in LCNN. The Bi-LSTM layer is used for pooling to aggregate utterance-level embeddings in LCNN. The resulting embedding has a dimension of 256.

AASIST-SAP: AASIST [12] is a waveform-level deepfake detection model. It includes a RawNet2-based [11] encoder and an attention network-based graph module. AASIST operates on raw waveforms to extract meaningful high-dimensional spectro-temporal feature maps. It then extracts graph nodes from the feature maps in both the temporal and frequency domains [12]. The final embedding is obtained by concatenating the mean and maximum values of various nodes. AASIST-SAP is an improved architecture where the max pooling layer of the encoded feature maps is replaced by a 2D self-attentive pooling (SAP) [22]. The resulting embedding has a dimension of 256.

wav2vec-ECAPA and wavlm-ECAPA: We adopt the Wav2vec 2.0 [23] and WavLM [24] models for front-end acoustic feature extraction in our system, respectively. Both models are large-scale self-supervised learning (SSL) pre-trained models widely used in the speech field. The effectiveness of Wav2vec in extracting acoustic features has been demonstrated in anti-spoofing scenarios [25]. By utilizing Wav2vec 2.0 and WavLM, we extract acoustic features from the input audio. These features are then fed into the ECAPA-TDNN model for further processing and embedding extraction. The resulting embedding has a dimension of 256.

4.2. Data processing and augmentation

For our input features, we adopt the log Mel-filterbank, log-spectrum, and waveform. The log Mel-filterbank energies have a dimension of 80, and the log-spectrum has a dimension of 257. The frame length is set to 25ms, and the hop size is set to 10ms.

To diversify the training samples, we apply on-the-fly data augmentation techniques [26]. We employ two types of augmentation methods: 1) Adding noise: We use the MUSAN dataset [27], specifically the non-speech parts, to add noise to the audio samples during training; 2) Adding convolutional reverberation: We utilize the RIR Noise datasets [28] to simulate convolutional reverberation effects and apply them to the training data.

4.3. Dataset analysis

4.3.1. Statistic Analysis

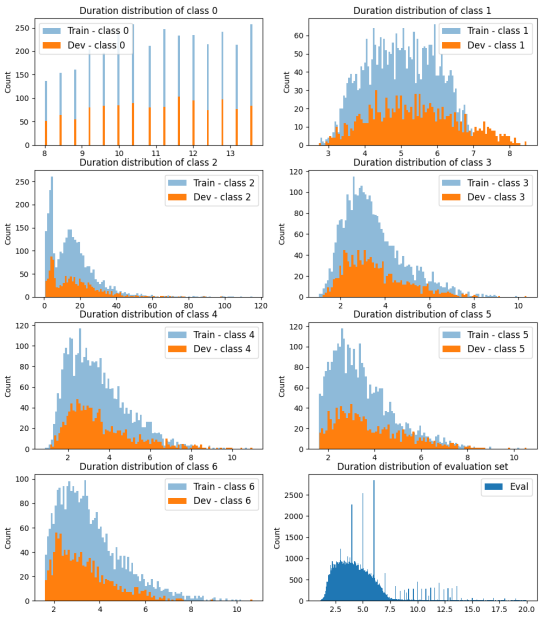


Figure 2: Distribution of audio durations in the train, development and evaluation set. The evaluation set is focusing on audio samples that are 20 seconds or less for ease of comparison.

The experiments are performed on the ADD2023 Track3 dataset. We start by performing a statistical analysis of the train set and development set. The train set consists of 22,400 audio samples, while the development set contains

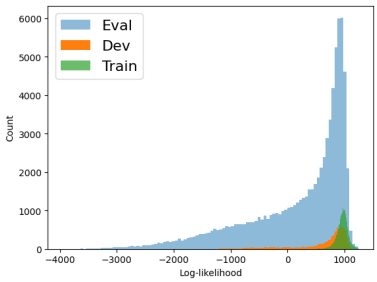


Figure 3: Distribution of log-likelihoods for all embeddings of the three sets. The higher the score, the higher the likelihood that the sample is from the same distribution as the training set.

8,400 samples. The average duration of the audio samples in the train set is 6.58 seconds, and in the development set, it is 6.84 seconds. Therefore, we randomly divide the samples into 6-second segments during training to ensure consistency in the input size. The duration distribution of the training and development sets can be observed in Figure 2.

Additionally, we observe that while the sampling rate of the training set is uniformly set to 16kHz, the development set contains audio samples with both 24kHz and 16kHz sampling rates. The sample rate of the 3rd and 4th classes in the development set is 24kHz. Some audio samples in the development set were originally at an 8kHz sampling rate but were upsampled to 16kHz. We notice that these upsampled samples may lack high-frequency information. To maintain consistency, we downsample or upsample all audio samples to a 16kHz sampling rate during training and testing.

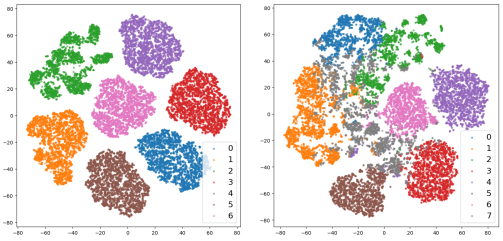
However, the evaluation set exhibits some statistical differences compared to the training set. The evaluation set includes audio samples with sampling rates of 16kHz, 24kHz, and 44.1kHz. On the other hand, the duration distribution of the evaluation set can be seen in Figure 2. Although there are certain statistical inferences present in the evaluation set, we choose not to exploit these statistical inference in decision to ensure the generalizability of our method.

4.3.2. Visualization Analysis

In order to assess the similarity between the training and test data distributions, we utilized a trained embedding extractor to extract embeddings from the train, development, and evaluation sets. Subsequently, we employed a Gaussian Mixture Model (GMM) with 512 components to perform one-class modeling on the train set. By calculating the log-likelihood for all embeddings on the one-class GMM, we obtained score distributions of three sets as shown in Figure 3.

Upon analyzing the results, we observe that the score distributions of the evaluation set and development set are relatively similar. However, there is a significant disparity between the score distribution of the evaluation set and the training sets. This observation highlights the considerable challenge presented by the evaluation set in terms of testing the generalization performance of our proposed method.

In addition, we also present the t-SNE visualization of the training set and the evaluation set in Figure 4. The embeddings t-SNE used are extracted by ResNet34SimAM-ASP model. The label of evaluation set is marked by center-based maximum similarity method. We have observed that, except for the samples at the margins, most of the known classes can be recognized effectively. However, the unknown class does not exhibit a clear intra-class center and is not easily detected. Therefore, we speculate that the unknown class may not consist of a single category but rather a collection of multiple unknown classes.



(a) t-SNE of embeddings in the training set

(b) t-SNE of embeddings in the evaluation set

Figure 4: Visualization analysis of t-SNE

5. Results

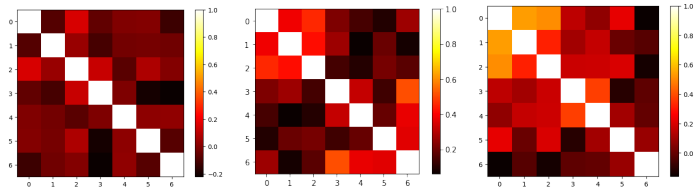
Table 1 presents the final performance of various systems using center-based maximum similarity method. Firstly, by comparing different models with the addition of the development set (e.g., ID 1&2, ID 4&5, and ID 8&9), no clear improvement or decrease in performance was observed. This could potentially be attributed to the limited amount of data, which makes the models prone to overfitting and results in unstable performance. Based on the score distribution depicted in Figure 3, the score trends of evaluation set and development set are similar. Therefore, in order to select the best model of each system, many models are not trained using the development set.

Secondly, we analyze the impact of acoustic features on the recognition performance. Since different models exhibited similar performance trends with different features, the focus is mainly on comparing model ID 4 and model ID 6 in terms of different features. It was found that the spectrum feature tended to overfit on the development set, indicating that the performance on the development set alone might not fully reflect the generalization performance of the models. In addition, performance of the SSL-based models are also not satisfactorily.

Table 1
The performances of various systems with center-based maximum similarity method in the development and evaluation sets. ID 0 indicates that we only adopt 7 known centroids without unknown category.

ID	Model	Train data	Feature	Dev		Eval
				Acc	F1	
0	ResNet34SimAM-ASP (7 categories)	Train set	Mel	96.536	0.965	0.7150
1	ResNet34SimAM-ASP	Train set	Mel	96.536	0.965	0.7838
2	ResNet34SimAM-ASP	Train + Dev	Mel	-	-	0.7880
3	ResNet34GSP	Train	Mel	94.238	0.942	0.7650
4	ResNet34SE-ASP	Train	Mel	93.107	0.930	0.7693
5	ResNet34SE-ASP	Train + Dev	Mel	-	-	0.7398
6	ResNet34SE-ASP	Train	Spectrum	98.798	0.988	0.6006
7	ECAPA-ASP	Train	Mel	92.107	0.920	0.6402
8	LCNN	Train	Mel	96.27	0.9626	0.7628
9	LCNN	Train+Dev	Mel	-	-	0.7825
10	AASIST-SAP	Train	Waveform	95	0.9476	0.7422
11	Wav2vec(fixed)-ECAPA	Train	Waveform	97.75	0.9773	0.7075
12	Wav2vec(finetune)-ECAPA	Train	Waveform	91.17	0.9122	0.4493
13	WavLM(fixed)-ECAPA	Train	Waveform	93.04	0.93	0.6401
fus(0.5×ID 1+ 0.5×ID 4)						0.7943
fus(0.3×ID 1+ 0.25×ID 4+ 0.3×ID 9+ 0.25×ID 10)						**0.8312**

Figure 5: Inter-class similarity matrix for different system



(a) Inter-class similarity matrix of ResNet34SimAM-ASP with Mel feature (b) Inter-class similarity matrix of ResNet34SE-ASP with Mel feature (c) Inter-class similarity matrix of ResNet34SE-ASP with Spectrum

Considering the utilization of the center-based maximum similarity method, the inter-class similarity matrix is computed for different models, as shown in Figure 5. By combining Table 1 and Figure 5, it is observed that better results are achieved when the inter-class similarity distances are larger. Therefore, the best single model of each system is selected by considering both the inter-class similarity matrix and the results on the development set.

Model fusion is performed at the score level to achieve complementary performance between models. Several optimal models with different modeling approaches are selected. Finally, an F1 score of 0.8312 is achieved on the evaluation set.

6. Conclusion

In this paper, we regard the deepfake algorithm recognition as speaker verification and propose the center-based maximum similarity method to determine the test audio category. We select the best single model according to the intra-class similarity matrix and result of development set. Finally, by model fusion under the score level, we achieved the 0.8312 F1-score in the evaluation set of ADD2023 Track3.

7. Acknowledgments

This research is funded in part by the National Natural Science Foundation of China (62171207) and MaShang Consumer Finance Co.Ltd. Many thanks for the computational resource provided by the Advanced Computing East China Sub-Center.

References

[1] X. Tan, T. Qin, F. Soong, T.-Y. Liu, A survey on neural speech synthesis, arXiv preprint arXiv:2106.15561 (2021).

[2] Z. Wu, N. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, H. Li, Spoofing and countermeasures for speaker

verification: A survey, speech communication 66 (2015) 130–153.

[3] X. Wang, J. Yamagishi, M. Todisco, H. Delgado, A. Nautsch, N. Evans, M. Sahidullah, V. Vestman, T. Kinnunen, K. A. Lee, et al., Asvspoof 2019: A large-scale public database of synthesized, converted and replayed speech, Computer Speech & Language 64 (2020) 101114.

[4] J. Yamagishi, X. Wang, M. Todisco, M. Sahidullah, J. Patino, A. Nautsch, X. Liu, K. A. Lee, T. Kinnunen, N. Evans, H. Delgado, ASVspoof 2021: accelerating progress in spoofed and deepfake speech detection, in: Proc. ASVspoof2021 workshop, 2021, pp. 47–54.

[5] J. Yi, J. Tao, R. Fu, X. Yan, C. Wang, T. Wang, C. Y. Zhang, X. Zhang, Y. Zhao, Y. Ren, L. Xu, J. Zhou, H. Gu, Z. Wen, S. Liang, Z. Lian, H. Li, Add 2023: the second audio deepfake detection challenge, accepted by IJCAI 2023 Workshop on Deepfake Audio Detection and Analysis (DADA 2023) (2023).

[6] Z. Wu, T. Kinnunen, N. Evans, J. Yamagishi, M. Sizov, Asvspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge, in: Proc. Interspeech, 2015.

[7] H. Tak, M. Todisco, X. Wang, J. weon Jung, J. Yamagishi, N. Evans, Automatic Speaker Verification Spoofing and Deepfake Detection Using Wav2vec 2.0 and Data Augmentation, in: Proc. Odyssey, 2022, pp. 112–119. doi:10.21437/Odyssey.2022-16.

[8] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, in: Proc. CVPR, 2016, pp. 770–778.

[9] J. Hu, L. Shen, G. Sun, Squeeze-and-Excitation Networks, in: Proc. CVPR, 2018.

[10] X. Wu, R. He, Z. Sun, T. Tan, A light cnn for deep face representation with noisy labels, IEEE Transactions on Information Forensics and Security 13 (2018) 2884–2896.

[11] H. Tak, J. Patino, M. Todisco, A. Nautsch, N. Evans, A. Larcher, End-to-end anti-spoofing with rawnet2, in: Porc. ICASSP, 2021, pp. 6369–6373.

[12] J.-w. Jung, H.-S. Heo, H. Tak, H.-j. Shim, J. S. Chung, B.-J. Lee, H.-J. Yu, N. Evans, AASIST: Audio anti-spoofing using integrated spectro-temporal graph attention networks, in: Proc. ICASSP, 2022.

[13] F. Wang, J. Cheng, W. Liu, H. Liu, Additive margin softmax for face verification, IEEE Signal Processing Letters 25 (2018) 926–930.

[14] J. Deng, J. Guo, N. Xue, S. Zafeiriou, Arcface: Additive angular margin loss for deep face recognition, in: Proc. CVPR, 2019, pp. 4690–4699.

[15] Y. Zhang, F. Jiang, Z. Duan, One-class learning towards synthetic voice spoofing detection, IEEE Signal Processing Letters 28 (2021) 937–941.

[16] N. Müller, F. Diekmann, J. Williams, Attacker Attribution of Audio Deepfakes, in: Proc. Interspeech, 2022, pp. 2788–2792.

[17] T. Zhu, X. Wang, X. Qin, M. Li, Source tracing: Detecting voice spoofing, in: Proc. APSIPA ASC, 2022, pp. 216–220.

[18] R. Li, M. Zhao, Z. Li, L. Li, Q. Hong, Anti-Spoofing Speaker Verification System with Multi-Feature Integration and Multi-Task Learning, in: Proc. Interspeech, 2019, pp. 1048–1052.

[19] X. Qin, N. Li, C. Weng, D. Su, M. Li, Simple attention module based speaker verification with iterative noisy label detection, in: Proc. ICASSP, 2022, pp. 6722–6726.

[20] K. Okabe, T. Koshinaka, K. Shinoda, Attentive Statistics Pooling for Deep Speaker Embedding, Proc. Interspeech (2018).

[21] D. Desplanques, J. Thienpondt, K. Demuynck, ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification, in: Proc. Interspeech, 2020, pp. 3830–3834.

[22] Y. Zhu, T. Ko, D. Snyder, B. Mak, D. Povey, Self-attentive speaker embeddings for text-independent speaker verification., in: Proc. Interspeech, volume 2018, 2018, pp. 3573–3577.

[23] A. Baevski, Y. Zhou, A. Mohamed, M. Auli, wav2vec 2.0: A framework for self-supervised learning of speech representations, Advances in neural information processing systems 33 (2020) 12449–12460.

[24] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, J. Wu, L. Zhou, S. Ren, Y. Qian, Y. Qian, J. Wu, M. Zeng, F. Wei, Wavlm: Large-scale self-supervised pre-training for full stack speech processing (2021). arXiv:2110.13900.

[25] H. Tak, M. Todisco, X. Wang, J.-w. Jung, J. Yamagishi, N. Evans, Automatic speaker verification spoofing and deepfake detection using wav2vec 2.0 and data augmentation, arXiv preprint arXiv:2202.12233 (2022).

[26] W. Cai, J. Chen, J. Zhang, M. Li, On-the-Fly Data Loader and Utterance-Level Aggregation for Speaker and Language Recognition, IEEE/ACM Transactions on Audio, Speech, and Language Processing (2020) 1038–1051.

[27] D. Snyder, G. Chen, D. Povey, MUSAN: A Music, Speech, and Noise Corpus, arXiv:1510.08484 (2013).

[28] T. Ko, V. Peddinti, D. Povey, M. Seltzer, S. Khudanpur, A study on data augmentation of reverberant speech for robust speech recognition, in: Proc. ICASSP, 2017, pp. 5220–5224.