WAVEFORM BOUNDARY DETECTION FOR PARTIALLY SPOOFED AUDIO

Zexin Cai[1,*], *Weiqing Wang*[1,*], *Ming Li*[1,2,†]

[1]Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708, USA
[2]Data Science Research Center, Duke Kunshan University, Kunshan 215316, PR China

ming.li369@duke.edu

ABSTRACT

The present paper proposes a waveform boundary detection system for audio spoofing attacks containing partially manipulated segments. Partially spoofed/fake audio, where part of the utterance is replaced, either with synthetic or natural audio clips, has recently been reported as one scenario of audio deepfakes. As deepfakes can be a threat to social security, the detection of such spoofing audio is essential. Accordingly, we propose to address the problem with a deep learning-based frame-level detection system that can detect partially spoofed audio and locate the manipulated pieces. Our proposed method is trained and evaluated on data provided by the ADD2022 Challenge. We evaluate our detection model concerning various acoustic features and network configurations. As a result, our detection system achieves an equal error rate (EER) of 6.58% on the ADD2022 challenge test set, which is the best performance in partially spoofed audio detection systems that can locate manipulated clips.

***Index Terms*—** Anti-spoofing, Wav2Vec, Partially spoofed audio detection, ADD challenge

1. INTRODUCTION

Deepfake audios refer to those altered or generated utterances that aim to fool human observers and machines. Deep learning frameworks have significantly improved the speech quality of text-to-speech (TTS) and voice conversion (VC), making the synthesized utterances too real to be distinguished from natural ones [1, 2, 3]. In addition, the robustness accomplished by zero-shot synthesis and any-to-any VC approaches makes the voice cloning performance much more powerful [4, 5, 6, 7]. Under this circumstance, high-fidelity synthesis/conversion systems inevitably allow criminals to commit fraud by impersonating others. Therefore, detecting spoofing utterances is critical to reduce the threat yielded by disinformation embedded in speech utterances. In addition, investigating spoofing detection techniques helps address the vulnerabilities of automatic speaker verification (ASV) systems against spoofing attacks [8, 9].

Regarding threats revealed by audio spoofing attacks, the ASVspoof challenge has been held biennially for countermeasures research against various spoofing attacks, including synthetic speech, voice conversion, replay, and impersonation [10]. Motivated by the challenge, researchers have proposed different advanced deep-learning structures for audio spoofing attacks. For example, Light-CNN (LCNN) [11] and RawNet [12] have been adopted as the backbone network in the anti-spoofing task and achieved satisfactory performances [13]. Even though many spoofing scenarios are included in the ASVspoof challenge, the spoofing scenario caused by partially spoofed/fake clips is not included. Spoofing utterances under this scenario are generated by altering the original bona fide utterances with natural or synthesized audio [14]. One example is that attackers can change a few words with synthesized clips and ultimately reverse the message carried by the utterance. Such attacks are similar to the audio splicing forgery attacks [15, 16, 17]. One distinction between partially spoofed attacks and audio splicing forgery attacks is that in the former, the partially spoofed audio sample may contain synthesized segments with the voice of the original speaker. Concerning partially spoofed attacks, two datasets, Half-Truth and PartialSpoof, along with their benchmark systems, are developed to advance the research in partially fake spoofing detection [18, 19, 20].

The recent Audio Deep Synthesis Detection (ADD) challenge has included the task of detecting partially spoofed utterances as one of the challenge tracks [14]. Lv et al. participated in the challenge and achieved the best performance with fake audio detection systems finetuned from unsupervised pretraining models [21]. However, defining the task as a binary classification problem under the utterance level, their proposed system has limitations in finding fake clips in a given utterance. Alternatively, Wu et al. incorporated a question-answering strategy in framework design, which allows the detection system to locate fake regions [22]. Typically, the partially fake audios include some artifacts, like the discontinuity between concatenated segments, that allow us to locate the fake utterance clips. Nonetheless, the best performance from [22] on ADD evaluation datasets needs further improvement compared to the detection system from [21].

In this paper, we propose a frame-level boundary detection system to detect partially fake audio. While previous approaches for this scenario mostly focus on utterance-level detection, our proposed system utilizes the discontinuity between segments and is designed to find the concatenation boundary from the acoustic information under the frame level. Specifically, our proposed system employs the Wav2Vec [23] as the feature extractor and the ResNet-1D as the main framework, followed by a Transformer encoder-based frame-level backend classifier for boundary detection. We use data simulated from datasets provided by the ADD challenge for training. Our proposed method is evaluated and analyzed with multiple test sets, including out-of-domain data. Results show that the proposed system outperforms other partially fake audio detection systems with boundary detection. It achieves an EER of 3.14% on the ADD adaptation set and an EER of 6.58% on the ADD test set. Overall, the main contribution of this paper is in proposing a novel framework for partially fake audio detection and achieving state-of-the-art performance among detection systems that can locate fake regions.

[*]Authors contributed equally
[†]Corresponding author: Ming Li

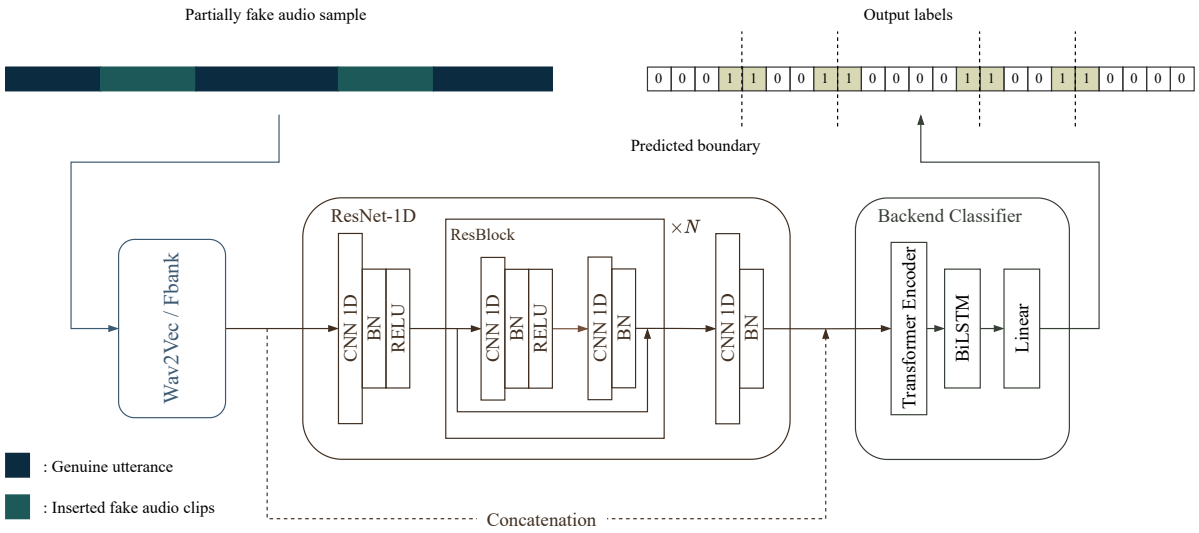

Fig. 1: The architecture of our proposed model.

2. METHOD

2.1. Task Definition

Considering an audio signal that contains fake audio segments, our purpose is to detect the frames that contain the information of discontinuity. Given the acoustic feature input $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T) \in \mathbb{R}^{D \times T}$ where D is the feature dimension and T is the number of frames, this task can be considered as a frame-level binary classification task with labeling information $\mathbf{y} = (y_1, y_2, ..., y_T) \in \{0, 1\}^T$.

Unlike other methods that only predict utterance-level binary decision [21], our proposed system can detect if a frame contains such discontinuity. If a frame is a boundary between a fake audio clip and a bona fide audio clip, the label is set to 1; otherwise, it is 0. In addition, the labels of frames near the boundary are also set to 1 to increase the robustness. The proposed method can not only find if an audio signal contains fake audio segments but can also identify where the segment is.

2.2. Model Architecture

Figure 1 shows the architecture of our proposed model. We employ a pre-trained Wav2Vec [23] to extract the features from raw audio. Next, we employ a ResNet-1D to further extract the frame-level embeddings. Then we concatenate the acoustic feature and the frame-level embeddings together as the input of a Transformer-based frame-level classifier, which outputs the probability of being a boundary for each frame.

2.2.1. Feature extraction

Wav2Vec is a self-supervised model which learns representations from raw audio data [24, 23]. It has shown impressive performance on many downstream tasks like automatic speech recognition (ASR) [25]. Therefore, we also employ this unsupervised model as a feature extractor for raw audio samples. More specifically, we employ the Wav2Vec 2.0 (wav2vec2-base-960h[1]). The model wav2vec2-base-960h is trained on the Librispeech dataset [26], which contains 960 hours of speech utterances. The output shape and the number of parameters of this model are shown in Table 1. The frame rate of the output feature is 20ms. On the other hand, we also use the Mel filterbank (Fbank) feature for comparison. We extract the 80-dimensional Fbank as well as 1st- and 2nd-order deltas with frame length of 25ms and frame shift of 10ms. Thus the total dimension of the final Fbank feature is 240 and the frame rate is 10ms when using Fbank feature.

Table 1: Acoustic feature size

Feature	Dimension	Frame rate	#Parameters
Wav2Vec	768	20ms	95m
Fbank	240	10ms	-

2.2.2. Frame-level embedding extraction

We employ a ResNet-1D network after the feature extractor to obtain frame-level embeddings. The architecture is shown in Figure 1. The ResNet-1D contains two CNN-1D layers sandwiching N residual blocks, each of which contains two CNN-1D layers and a residual connection from input to output.

Give the input feature $\mathbf{X} \in \mathbb{R}^{D \times T}$, the ResNet-1D extracts a frame-level embedding sequence $\mathbf{S} \in \mathbb{R}^{D' \times T}$, where D is the feature size, D' is the frame-level embedding size, and T is the number of frames.

2.2.3. Frame-level classifier

To capture the long-range global context within frames, we employ several Transformer encoders. Later, we use a BiLSTM to further model the sequential embedding from Transformer encoders, and finally, a fully connected layer predicts the boundary probability for each frame.

[1]https://huggingface.co/facebook/wav2vec2-base-960h

3. EXPERIMENT

3.1. Data Preparation

Our training data is constructed from several datasets provided by the ADD challenge [14]. Table 2 presents the statistics of datasets used in our experiments. As shown in Table 2, the ADD-train dataset contains 3012 bona fide utterances and 24072 fake utterances, while the ADD-dev dataset contains 2307 bona fide utterances and 26017 fake utterances. All fake utterances from ADD-train and ADD-dev are synthesized by mainstream speech synthesis and voice conversion systems. The ADD-adaptation dataset has 1052 partially fake utterances, and the ADD-test consists of 100625 utterances without labeling information. The Partially-fake dataset is generated from the ADD-train dataset. We first use a speech recognition model trained with AISHELL-2 [27] from Kaldi [28] to obtain the transcript and word boundary information of the ADD-train dataset. As we have the word boundary information of each utterance, we insert audio clips into the bona fide utterances according to the following strategies:

Table 2: The statistics of datasets (#Utterances)

Name	Bona fide	Fake	All
ADD-train	3012	24072	27084
ADD-dev	2307	21295	23602
ADD-test	-	-	100625
ADD-adaptation	0	1052*	1052
Partially-fake	0	35808*	35808

* denotes partially fake utterances

1. For each bona fide utterance, randomly replace $n \in \{1, 2, 3\}$ word segments with word segments from other bona fide utterances.
2. For each bona fide utterance, randomly replace $n \in \{1, 2, 3\}$ word segments with word segments from fake utterances.
3. For bona fide utterance i, randomly repeat $n \in [1, \lfloor N_i^w/3 \rfloor]$ word segments, where N_i^w is the number of word segments of the utterance.

Every strategy is applied to every utterance several times to construct partially fake utterances. Correspondingly, we generate 35808 partially fake utterances for training. As for model training, we combine the bona fide utterances from the ADD-train dataset and the Partially-fake dataset as the final training data. Besides the ADD-test set, we randomly select 1052 utterances from the ADD-dev set and combine them with the ADD-adaptation set as the adaptation dataset for evaluation.

3.2. Model Configuration

Table 3 shows the architecture of the proposed model. ResNet-1D contains 12 Residual blocks, each CNN-1D in the residual block has no bias with kernel size=1, and the input and output sizes are 512. The first CNN-1D has no bias with kernel size=5. The input and output sizes are 768 and 512, respectively. The kernel size of the final CNN-1D is 1. The input size is 512, the output is the frame-level embedding, and the embedding size is set to 128.

For the frame-level classifier, the Transformer encoder contains four heads and two encoder layers. The size of the feed-forward network (FFN) is 1024. The BiLSTM contains 128 hidden neurons followed by a ReLU activation. Finally, a 256-d fully-connected layer predicts the probability for each frame.

Table 3: The network architecture, where **C**(kernal size, padding, stride) denotes the convolutional layer, $[\cdot]$ denotes the residual block, **E**(layers, heads, FFN size) denotes the Transformer Encoder, BiLSTM(layers, hidden units) denotes BiLSTM layer, Linear(input size, output size) denotes the fully-connected layer; L relates to the duration of the speech and T is the number of label frames.

Layer	Output Size	Structure
Input audio	$L \times 1$	-
Wav2Vec	$T \times 768$	-
or Fbank	$T \times 240$	-
CNN 1D	$T \times 512$	$\mathbf{C}(5, 2, 1)\text{w/o bias}$
ResBlock(s)	$T \times 512$	$\begin{bmatrix}\mathbf{C}(1, 0, 1)\text{ w/o bias} \\ \mathbf{C}(1, 0, 1)\text{ w/o bias}\end{bmatrix} \times 12$
CNN 1D	$T \times 128$	$\mathbf{C}(1, 0, 1)$
Transformer Encoder	$T \times 128$	$\mathbf{E}(2, 4, 1024)$
BiLSTM	$T \times 128$	BiLSTM(1, 128)
Linear	$T \times 1$	Linear(256, 1)

3.3. Training Process

During training, we randomly pick an audio sample from the Partially-fake dataset and the bona fide data from ADD-train. The probability of selecting a positive sample is set to 0.5 to ensure data balance. Each audio is cut to a fixed length l, e.g., 0.64s, 1.28s, 2.56s. MUSAN [29] and RIRs [30] corpus are employed for online data augmentation. For positive samples, we set the labels to zero vectors. For negative samples, the labels of the boundaries between bonafide audio clips and fake audio clips are set to 1 and others are set to zeros. In addition, we also set the labels near boundaries to 1. In our experiments, we set labels of 4 closest frames to 1 for each corresponding boundary.

The model is trained with binary cross entropy loss and Adam optimizer for 100 epochs. The mini-batch size is set to 64. The learning rate is set to 10^{-4}, and Noam scheduler [31] with 1600 warm-up steps is employed. During training, we evaluate the equal error rate (EER) on the adaptation set for validation, and the five models with the lowest EER will be averaged for inference and evaluation.

3.4. Inference Process and Evaluation

We first split each audio signal into overlapping audio pieces during the inference stage. The length of each piece is the same as that of the training samples, while the step size is half of the duration of the length of each piece. For example, if the length we use in training is 1.28s, then the step size of 0.64s is accordingly set to obtain overlapped audio clips. The network can predict probabilities for each frame of each piece. Next, for each audio signal, we merge the results of all pieces by averaging the overlapped regions. The frames with probabilities greater than a threshold are considered the boundaries, and the mean of the largest n probabilities is the utterance level confidence score. In our experiments, n is set to 4 for best performance.

4. RESULTS AND DISCUSSION

Our first experiment is conducted to investigate our model's performance concerning different waveform lengths during training.

Our experiment evaluates and compares two systems: one trained with Wav2Vec feature extractor but without the concatenation operation shown in Figure 1, and the other trained with logarithmic Mel-spectrogram (Fbank) feature. As the Fbank feature is rather a raw acoustic feature from the frequency domain, we exclude the concatenation operation. Similarly, for a fair comparison, we also exclude the concatenation operation of the Wav2Vec-based model in our experiment. The results are shown in Table 4. We can see that the model based on Wav2Vec feature achieves the best performance when we set the fixed length l to 1.28s during training. With l set to 1.28s, the EER on the adaptation set is 3.71%, and the performance on the ADD-test set is 6.64%. A shorter waveform length, like 0.64s, causes significant performance degradation on evaluation sets, while increasing l to 2.56s results in minor performance degradation. The Fbank-based model achieves the best performance when $l = 0.64$s. Note that utterances from the adaptation set are constructed from the same dataset as our training set, while the ADD-test set contains utterances constructed from other data. Therefore, the adaptation set is an in-domain validation set, and the ADD-test contains out-of-domain data. Regarding the in-domain adaptation dataset, the performance of Fbank is close to the system using Wav2Vec. However, for the ADD-test set that contains out-of-domain data, the performance of Fbank is significantly worse than that of Wav2Vec, whereas the Wav2Vec model achieves an EER of 6.64% and the Fbank-based model achieves 12.71%. This result indicates that the model trained with features extracted by Wav2Vec generalizes better than Fbank in this task.

Table 4: System performances regarding various segment length l , reported in EER.

Feature	Test Sets	Wav Length l (s)		
		0.64	1.28	2.56
Wav2Vec (w/o concat)	Adaptation	4.85%	**3.71%**	4.09%
	ADD-test	9.39%	**6.64%**	6.69%
Fbank	Adaptation	**3.71%**	3.8%	4.66%
	ADD-test	**12.71%**	15.76%	20.56%

Therefore, we set the fixed waveform length l to 1.28s for the Wav2Vec-based model, while the l is set to 0.64s for the Fbank-based model. Then we conduct an ablation study to evaluate the models' performance regarding specific network components. As shown in Table 5, we remove part of the component from our proposed network and report the corresponding performance. The model 'w/o concatenation' denotes the model without the concatenation operation of the acoustic feature and the embedding output vector from the ResNet-1D. The model 'w/o ResNet-1D' has the frame-level embedding extractor ResNet-1D removed. As shown in Table 5, the EER on the adaptation set greatly increases as we remove the concatenation operation, while the performance on the ADD-test set degrades slightly. However, after removing the ResNet-1D module, the EER on adaptation is 3.23%, and the EER on the ADD-test set is 6.98%. Note that utterances from the adaptation set are constructed from the same dataset as our training set, while the ADD-test set contains utterances constructed from other data. Therefore, the adaptation set is an in-domain validation set, and the ADD-test contains out-of-domain data. The result indicates that the acoustic feature extracted by Wav2Vec generalizes well for in-domain data. Without the frame-level embeddings from ResNet-1D, the model still achieves an EER of 3.23% on the adaptation set. In contrast, the feature extracted by the module ResNet-1D works better on the out-of-domain data. We can see that the EER on ADD-test only increase 0.06%, while the EER on the adaptation set has increased from 3.14% to 3.71%. In general, adding the concatenation operation and the ResNet-1D module helps improve the model performance. For the Fbank-based model, when we remove the ResNet-1D module from the Fbank-based model, the performance degrades significantly on both evaluation sets. This shows that the Fbank is a lower-level acoustic feature compared to those extracted from the Wav2Vec. Thus the ResNet-1D module is much more crucial here to extract the frame-level feature for better performance.

Table 5: Performances (EER) regarding various feature extractor and architecture designs, w/o means without.

Model	Adaptation	ADD-test
Wav2Vec	**3.14%**	**6.58%**
w/o concat	3.71%	6.64%
w/o ResNet-1D	3.23%	6.98%
Fbank	3.71%	12.71%
w/o ResNet-1D	6.37%	20.66%
Utterance-level detection [21]	3.33%*	**4.8%**
Fake Span Discovery [22]	-	11.1%

* The number of bona fide utterances used for evaluation might be different

In addition, we compare our results to models in the literature. The utterance-level detection system from [21] also uses the Wav2Vec feature as the acoustic feature. The utterance-level system adopts a pretrained Wav2Vec model with one billion parameters. It achieves an EER of 3.33% on the adaptation set and 4.8% on the ADD-test set. Note that the fake utterances in the adaptation set are the same, but the number of bona fide utterances for evaluation is not reported in [21]. It is still the best performance on ADD-test set so far. We failed to fit the corresponding Wav2Vec model into our proposed framework as the parameter size is too large for our machine. Nevertheless, our model is designed as a frame-level detection system, which is capable of locating fake clips from partially fake utterances. The fake span discovery model from [22] also can perform frame-level analysis and uncover fake clips. The best single model based on fake span discovery achieves an EER of 11.1% on the ADD-test set, while our best performance is 6.58%, which makes our proposed method the best waveform boundary detection system on the ADD-test set.

5. CONCLUSION

We propose a frame-level partially fake audio detection method, which can not only provide an utterance-level binary decision for partially spoofed audio but also predict where the fake clips are inserted or replaced. We take an ablation study on our model components and evaluate different acoustic features including Wav2Vec and Fbank. Experimental results show that the proposed method has similar performance to the best utterance-level system on ADD challenge on adaptation set, and it has better performance than other boundary detection systems for partially spoofed audio.

6. REFERENCES

[1] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and

K. Kavukcuoglu, "WaveNet: A Generative Model for Raw Audio," in *9th ISCA Speech Synthesis Workshop*, 2016, pp. 125–125.

[2] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, et al., "Natural TTS Synthesis by Conditioning Wavenet on Mel Spectrogram Predictions," in *IEEE ICASSP*, 2018, pp. 4779–4783.

[3] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis," in *Advances in Neural Information Processing Systems*, 2020, vol. 33, pp. 17022–17033.

[4] J. chieh Chou and H.-Y. Lee, "One-Shot Voice Conversion by Separating Speaker and Content Representations with Instance Normalization," in *Proc. Interspeech 2019*, pp. 664–668.

[5] Z. Cai, C. Zhang, and M. Li, "From Speaker Verification to Multispeaker Speech Synthesis, Deep Transfer with Feedback Constraint," in *Proc. Interspeech 2020*, pp. 3974–3978.

[6] Y. Y. Lin, C.-M. Chien, J.-H. Lin, H.-y. Lee, and L.-s. Lee, "Fragmentvc: Any-to-any Voice Conversion by End-to-end Extracting and Fusing Fine-grained Voice Fragments with Attention," in *IEEE ICASSP*, 2021, pp. 5939–5943.

[7] H. Zhang, Z. Cai, X. Qin, and M. Li, "SIG-VC: A Speaker Information Guided Zero-Shot Voice Conversion System for Both Human Beings and Machines," in *IEEE ICASSP*, 2022, pp. 6567–65571.

[8] Z. Wu, N. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, and H. Li, "Spoofing and Countermeasures for Speaker Verification: A Survey," *Speech Communication*, vol. 66, pp. 130–153, 2015.

[9] M. K. Nandwana, M. Lomnitz, C. Richey, M. McLaren, D. Castan, L. Ferrer, and A. Lawson, "The VOiCES from a Distance Challenge 2019: Analysis of Speaker Verification Results and Remaining Challenges," in *Odyssey*, 2020, pp. 165–170.

[10] M. R. Kamble, H. B. Sailor, H. A. Patil, and H. Li, "Advances in Anti-spoofing: from the Perspective of ASVspoof Challenges," *APSIPA*, vol. 9, 2020.

[11] X. Wu, R. He, Z. Sun, and T. Tan, "A Light CNN for Deep Face Representation with Noisy Labels," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2884–2896, 2018.

[12] H. Tak, J. Patino, M. Todisco, A. Nautsch, N. Evans, and A. Larcher, "End-to-end Anti-spoofing with RawNet2," in *IEEE ICASSP*, 2021, pp. 6369–6373.

[13] X. Wang, X. Qin, T. Zhu, C. Wang, S. Zhang, and M. Li, "The DKU-CMRI System for the ASVspoof 2021 Challenge: Vocoder based Replay Channel Response Estimation," in *Proc. 2021 Edition of the Automatic Speaker Verification and Spoofing Countermeasures Challenge*, pp. 16–21.

[14] J. Yi, R. Fu, J. Tao, S. Nie, H. Ma, C. Wang, T. Wang, Z. Tian, Y. Bai, C. Fan, S. Liang, S. Wang, S. Zhang, X. Yan, L. Xu, Z. Wen, and H. Li, "ADD 2022: the first Audio Deep Synthesis Detection Challenge," in *IEEE ICASSP*, 2022, pp. 9216–9220.

[15] H. Zhao, Y. Chen, R. Wang, and H. Malik, "Audio splicing detection and localization using environmental signature," *Multimedia Tools and Applications*, vol. 76, pp. 13897–13927, 2017.

[16] H. Zhao, Y. Chen, R. Wang, and H. Malik, "Anti-forensics of environmental-signature-based audio splicing detection and its countermeasure via rich-features classification," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 7, pp. 1603–1617, 2016.

[17] X. Pan, X. Zhang, and S. Lyu, "Detecting splicing in digital audios using local noise level estimation," in *IEEE ICASSP*, 2012, pp. 1841–1844.

[18] J. Yi, Y. Bai, J. Tao, H. Ma, Z. Tian, C. Wang, T. Wang, and R. Fu, "Half-Truth: A Partially Fake Audio Detection Dataset," in *Proc. Interspeech 2021*, pp. 1654–1658.

[19] L. Zhang, X. Wang, E. Cooper, J. Yamagishi, J. Patino, and N. Evans, "An Initial Investigation for Detecting Partially Spoofed Audio," in *Proc. Interspeech 2021*, pp. 4264–4268.

[20] L. Zhang, X. Wang, E. Cooper, and J. Yamagishi, "Multi-task Learning in Utterance-level and Segmental-level Spoof Detection," in *Proc. 2021 Edition of the Automatic Speaker Verification and Spoofing Countermeasures Challenge*, pp. 9–15.

[21] Z. Lv, S. Zhang, K. Tang, and P. Hu, "Fake Audio Detection Based On Unsupervised Pretraining Models," in *IEEE ICASSP*, 2022, pp. 9231–9235.

[22] H. Wu, H.-C. Kuo, N. Zheng, K.-H. Hung, H.-Y. Lee, Y. Tsao, H.-M. Wang, and H. Meng, "Partially Fake Audio Detection by Self-Attention-Based Fake Span Discovery," in *IEEE ICASSP*, 2022, pp. 9236–9240.

[23] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "Wav2vec 2.0: A Framework for Self-supervised Learning of Speech Representations," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12449–12460, 2020.

[24] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "Wav2vec: Unsupervised Pre-Training for Speech Recognition," in *Proc. Interspeech 2019*, 2019, pp. 3465–3469.

[25] A. Baevski, W.-N. Hsu, A. Conneau, and M. Auli, "Unsupervised Speech Recognition," *Advances in Neural Information Processing Systems*, vol. 34, pp. 27826–27839, 2021.

[26] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR Corpus Based on Public Domain Audio Books," in *IEEE ICASSP*, 2015, pp. 5206–5210.

[27] J. Du, X. Na, X. Liu, and H. Bu, "Aishell-2: Transforming Mandarin ASR Research into Industrial Scale," *arXiv preprint arXiv:1808.10583*, 2018.

[28] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, et al., "The Kaldi Speech Recognition Toolkit," in *2011 Workshop on Automatic Speech Recognition and Understanding*, pp. 1–4.

[29] D. Snyder, G. Chen, and D. Povey, "MUSAN: A Music, Speech, and Noise Corpus," *arXiv:1510.08484*, 2015.

[30] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A Study on Data Augmentation of Reverberant Speech for Robust Speech Recognition," in *IEEE ICASSP*, 2017, pp. 5220–5224.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.