Investigating Self-supervised Learning-Based Front-End for Multi-Channel Replay Attack Detection

Takuo Yamaguchi*, Sayaka Shiota*, and Naohiro Tawara†
* Graduate School of Systems Design, Tokyo Metropolitan University, Tokyo, Japan
E-mail: yamaguchi-takuo@ed.tmu.ac.jp, sayaka@tmu.ac.jp
† NTT, Inc., Kyoto, Japan
E-mail: naohiro.tawara@ieee.org

***Abstract*—As the use of voice-operated devices for handling personal information becomes more widespread, automatic speaker verification is implemented to enhance security. However, automatic speaker verification systems are vulnerable to spoofing attacks, particularly replay attacks. These attacks can be challenging to detect and pose a significant security threat. Many voice-operated devices are typically equipped with multiple microphones, enabling detection that leverages spatial features. Nevertheless, previous spoofing speech detection researches primarily focused on single-channel audio, and the effectiveness of multi-channel information has received limited exploration. This paper, therefore, investigates spoofing detection methods to leverage multi-channel information. Specifically, we evaluated two types of front-end processing combined with two different backend classification models. As part of our investigation, we focused particularly on the effectiveness of using self-supervised learning models for the front-end feature extraction. The results of experiments using a ReMASC corpus confirmed that the method, which utilizes the self-supervised learning-based front-end, improves detection accuracy, reducing the equal error rate by approximately 14.2% compared to the conventional methods.**

I. Introduction

In recent years, the use of voice-operated devices that take instructions via voice, such as smart speakers and voice assistants, has become widespread. While these devices enable users to perform tasks easily through voice commands, they also raise security concerns, since they can execute tasks linked to personal information. Therefore, automatic speaker verification (ASV) is expected to be widely implemented in the voice-operated devices [1]. However, ASV is known to be vulnerable to spoofing attacks [2], posing a significant security risk. In particular, replay attacks using pre-recorded speech are a significant threat to the voice-operated devices, making countermeasures to detect the spoofing attacks an urgent issue.

Spoofing attack detection is actively researched [3]–[6]. The major benchmark, the ASVspoof challenge [3], has designed tasks such as logical access (LA), which assumes attacks using synthesized or voice-converted speech, and physical access (PA), which assumes attacks using replayed pre-recorded speech [4]. Since ASVspoof has released some single-channel corpora for speech spoofing detection, the majority of research on spoofing detection has focused on single-channel speech. Consequently, research on spoofing detection using multi-channel speech is limited [7], [8].

Most of the voice-operated devices are equipped with microphone arrays [9], [10], which allows for the use of multi-channel information to detect spoofing attacks. Therefore, few methods have been proposed that utilize multi-channel speech from the voice-operated devices to detect spoofing attacks [11], [12]. In [11], the multi-channel information is integrated into one feature representation using filter-sum beamformer [13]. The backend uses a long short-term memory (LSTM) network and fully connected layers as a binary classifier to detect spoofing speech. In [12], three-dimensional features obtained from a short-time Fourier transform (STFT), consisting of channels, time frames, and frequency bins, are directly input into a VGG-16-based classifier [14]. Both methods are based on signal processing-based feature extraction and have reported that using multi-channel information improves the detection accuracy of the replay attacks. However, [11], [12] still face challenges regarding their robustness to differences in recording conditions. Several studies have demonstrated the superior performance of self-supervised learning (SSL)-based features over signal processing-based features for single-channel spoofing attack detection [6], [15]. These findings suggest that signal processing-based approaches may have limitations in capturing the discriminative representations needed to distinguish between bona fide and spoofed speech. From another perspective, the extraction of discriminative features from multi-channel audio for spoofing attack detection using SSL has been insufficiently explored. Furthermore, investigations into the impact of integrating front-end feature extraction with back-end classification on detection performance remain limited.

In this paper, we investigate the impact of front-end feature extraction methods and backend classification models on the detection accuracy for replay attacks, utilizing multi-channel information. For the front-end feature extraction, we investigated a method based on SSL, in addition to the conventional signal processing-based methods. For the backend, two types of classification models are used: one based on a

979-8-3315-7206-8/25/$31.00

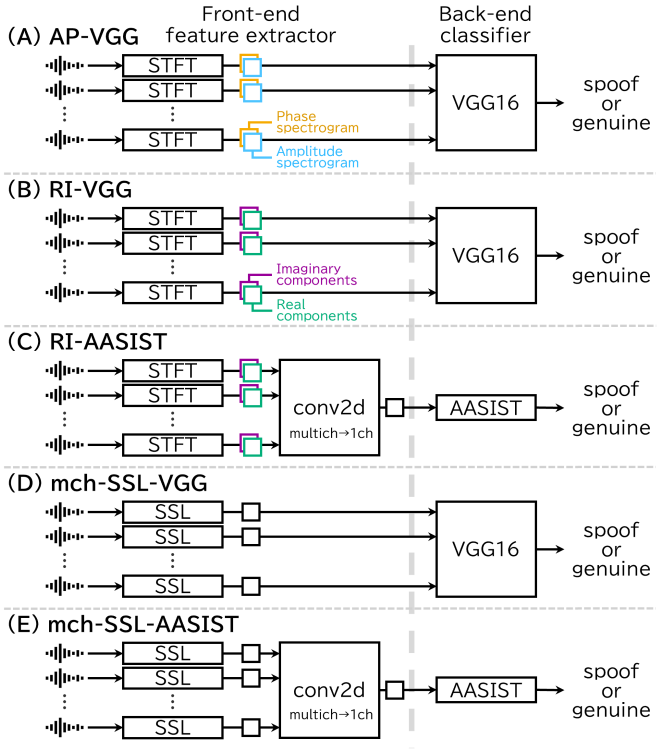

Fig. 1. Spoofing speech detection systems using multi-channel information.

VGG network and the other on an integrated spectro-temporal graph attention network (AASIST) [16]. The AASIST model serves as a baseline in the latest ASVspoof challenge [5]. The effects on detection accuracies were analyzed by evaluating multiple combinations of the front-end and backend methods. We use ReMASC corpus [10] for training and evaluation in our experiments. It is a spoofing speech corpus recorded in a multi-channel format for the detection of replay attacks. The experimental results confirmed that the SSL-based features are more robust to the background noise and the differences in the recording devices than the conventional signal processing-based features, achieving high detection accuracies.

II. Spoofing Speech Detection Using Multi-Channel Speech

The methods investigated in this paper are classified into the signal processing-based methods and the SSL-based methods, with a focus on the front-end.

A. Signal Processing-based Methods

This section describes the spoofing speech detection methods that use multi-channel information, specifically those employing signal processing-based features. One of the previous studies employs a filter-sum beamformer for the front-end processing and LSTM with fully connected layers for the backend (filter-sum-beamformer-LSTM) [11]. The filter-sum beamformer is a technique that captures spatial information by utilizing the inter-channel time difference of arrival (TDoA). In filter-sum-beamformer-LSTM, the beamformer is achieved by convolving a multi-channel signal with a P-order filter bank following the method proposed in [17]. This process allows the TDoA for each channel to be captured implicitly through learnable filters. Consequently, filter-sum-beamformer-LSTM explicitly makes use of spatial information.

Figure 1 shows the diagram of the methods described below. AP-VGG (Fig. 1 (A)) uses both amplitude and phase (AP) spectrograms as its input features, and feeding them into a VGG-16-based classifier as the backend [12]. In the front-end, STFT is applied to each audio channel to create amplitude and phase spectrograms. This produces a feature map with $2 \times C$ channels from C-channel audio. The backend VGG-16 network is modified to take $2C$ channels as input and has the output dimension of two for binary classification. AP-VGG uses the phase spectrogram as a feature; however, the handling of wrapped phase in the range $(-\pi, \pi]$ is known to be difficult, with several possible approaches [18]–[21]. To address this issue, some approaches using the real and imaginary (RI) parts of the complex spectrogram as features have been proposed [18]–[20], which are thought to yield more stable results compared to using phase and amplitude. In this paper, we use RI-VGG (Fig. 1 (B)), which replaces the front-end features of AP-VGG with the RI spectrograms. In the frontend of RI-VGG, STFT is performed on each channel to obtain a complex spectrogram:

$$\boldsymbol{X}_{\mathrm{STFT}} = \left[\{\mathrm{STFT}(\boldsymbol{x}_c)\}_{c=1}^{C}\right] \in \mathbb{C}^{T\times F\times C}. \quad (1)$$

Here, $\boldsymbol{x}_c \in \mathbb{R}^t$, T is the number of time frames, and F is the number of frequency bins. The operator $[\cdot]$ represents stacking matrices along the channel axis. By separating this complex representation into its RI parts, the RI feature $\boldsymbol{X}_{\mathrm{RI}}$ is obtained:

$$\boldsymbol{X}_{\mathrm{RI}} = [\mathrm{Re}(\boldsymbol{X}_{\mathrm{STFT}}), \mathrm{Im}(\boldsymbol{X}_{\mathrm{STFT}})] \in \mathbb{R}^{T\times F\times 2C}. \quad (2)$$

For the backend of AP-VGG, VGG-16 is also used. Furthermore, to investigate the influence of the backend classification model, AASIST is also used as the backend. AASIST has shown high performance in speech spoofing detection [16] and was adopted as the baseline for the ASVspoof5 challenge [5]. RI-AASIST (Fig. 1 (C)) is a combination of the frontend, which extracts the RI features, and AASIST for the backend. Since AASIST is designed to process single-channel audio as input, the RI features are first integrated into a single-channel feature map using a 2D convolution and then fed into AASIST.

B. Self-Supervised Learning-based Methods

This section describes spoofing detection methods that utilize multi-channel information, specifically those employing features extracted from an SSL model.

First, mch-SSL-VGG (Fig. 1(D)) replaces the front-end of the signal processing-based RI-VGG with an SSL model. Most SSL-based pre-trained models are designed for single-channel audio and therefore cannot be applied directly to the multi-channel scenarios. The mch-SSL-VGG model performs feature extraction for each channel individually, using shared parameters, where the same SSL model is applied to each

channel. For each channel $\boldsymbol{x}_c$, the SSL model extracts a feature sequence $\mathrm{SSL}(\boldsymbol{x}_c) \in \mathbb{R}^{T \times D}$, where D is the SSL feature dimension. These sequences are then concatenated as follows:

$$\boldsymbol{X}_{\mathrm{SSL}} = \left[\{\mathrm{SSL}(\boldsymbol{x}_c)\}_{c=1}^{C} \right]. \tag{3}$$

The resulting tensor, $\boldsymbol{X}_{\mathrm{SSL}} \in \mathbb{R}^{T \times D \times C}$, is subsequently fed into a VGG-16 network as a C-channel feature map for classification.

Next, mch-SSL-AASIST (Fig. 1(E)) is an extension of the SSL-AASIST [15] model to handle the multi-channel audio. It extracts SSL features $\boldsymbol{X}_{\mathrm{SSL}}$ in the same way as mch-SSL-VGG, and then integrates these features into one single-channel feature map in the same way as RI-AASIST. Finally, the backend AASIST performs the classification.

III. Experiments

In the experiments, the appropriate handling of phase spectrograms in the STFT-based feature extraction is initially evaluated. This is followed by a comparative investigation of the spoofing speech detection accuracy for each method detailed in Section II.

A. Dataset

For the experiments, we used ReMASC [10], a speech corpus recorded in multi-channel format and designed for spoofing attacks, with a focus on replay attacks. The ReMASC corpus aimed to improve spoofing speech detection in the voice-operated devices. It included multi-channel audio recordings captured with various playback and recording devices in various environments that simulate real-world conditions. Four types of microphone arrays were utilized to simulate voice-operated recording devices. The arrays include D1: Google AIY Voice Kit (2-channel), D2: Respeaker 4-mic Linear Array, D3: Respeaker Core V2 (6-channel circular array), and D4: Amlogic A113X1 (7-channel circular array, comprising a 6-channel setup plus a center microphone). The audio was recorded in four types of environments simulating actual usage situations: Outdoor, Indoor1 (a quiet room), Indoor2 (a room with music or TV playing), and In-Car (inside a moving vehicle). Furthermore, for the three environments other than Indoor2, the relative positions of the sound source were varied to simulate actual use cases. The sampling frequency also differs depending on the recording device: D4 is 16 kHz, while the other three devices are 44.1 kHz [10].

B. Experimental Conditions

This section details our experimental conditions.

Following the previous studies [9], [12], we used the Core set of the ReMASC dataset for training and the QuickEvaluation set for evaluation. The first second of each audio file was used as input [11], [12]. For feature extraction, we used the STFT with a 10 ms window length, a 5 ms hop length, and a 512-point FFT, which resulted in the complex feature tensor $\boldsymbol{X}_{\mathrm{STFT}} \in \mathbb{C}^{201 \times 257 \times C}$. The models were trained using a cross-entropy loss function, weighted by the inverse ratio of the label counts to address class imbalance [9]. We used the AdamW optimizer [22] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a weight decay of 0.0001. The learning rate was warmed up from 10^{-5} to 10^{-4} over the first 20 epochs and subsequently halved every 20 epochs. Each model was trained for 100 epochs, with the model from the final epoch used for evaluation. Since architectures depend on the number of channels, models were trained individually for each recording device. Equal error rate (EER) was our evaluation metric.

TABLE I
Comparison of different phase unwrapping methods for spectrogram-based approaches.

Method	EER (%) per Recording Device				
	D1	D2	D3	D4	Average
AP-VGG [12]	6.6	11.0	9.2	15.7	10.6
(1) Frequency axis (F)	8.1	13.6	25.8	24.7	18.0
(2) Time axis (T)	22.3	13.4	14.1	26.0	18.9
(3) F then T	7.0	12.2	24.9	26.1	17.6
(4) T then F	7.8	13.9	23.6	27.4	18.2
(5) sin, cos	10.4	27.7	25.6	23.3	21.7
(6) RI	8.3	7.8	24.7	13.1	13.5
(7) Amplitude only	10.7	22.4	27.8	26.9	21.9

The evaluation of the phase handling methods aimed to identify the most effective approach for spoofing detection, as implementation details were lacking in [12]. We explored seven approaches for handling phase information: (1) unwrapping along the frequency axis only, (2) unwrapping along the time axis only, (3) frequency-then-time unwrapping, (4) time-then-frequency unwrapping, (5) decomposing the phase into sine and cosine components [21], (6) using RI parts as features [18]–[20], and (7) using only the magnitude spectrogram. For all approaches, the backend was based on that of AP-VGG. However, for the approaches (5) and (7), the number of input channels was adjusted to $3C$ and C, respectively. To ensure a direct comparison with [12], all data was resampled to 44.1 kHz.

In the comparative investigation of SSL-based systems, we evaluated six models: the conventional filter-sum beamformer [11] and the five methods illustrated in Fig. 1. For mch-SSL-VGG and mch-SSL-AASIST, we employed the wav2vec2.0-XLSR (0.3B) [23] pre-trained model, which resulted in the feature tensor $\boldsymbol{X}_{\mathrm{SSL}} \in \mathbb{R}^{137 \times 1024 \times C}$. The SSL front end is fine-tuned jointly with the back end on ReMASC, using the same training parameters. Since this SSL model was pre-trained on 16 kHz data, all methods in this comparison (except for AP-VGG) used 16-kHz resampled data. For AP-VGG, we referred to the results from the original experiments, which used 44.1-kHz sampled data. RI-AASIST and mch-SSL-AASIST used a 2D convolution with a 3×3 kernel to integrate the multi-channel features into one single-channel representation.

To investigate the effectiveness of using multi-channel audio, we also evaluated the systems with single-channel audio. For this single-channel evaluation, a total of four models were adapted. For RI-VGG and SSL-VGG (the single-channel variant of mch-SSL-VGG), the number of input channels was

TABLE II
EERs (%) FOR EACH RECORDING DEVICE AND AVERAGE EER ACROSS DEVICES.

Spoofing Speech Detection System			EER (%) per Recording Device				
System	Front-end Feature Extraction	Back-end Classifier	D1	D2	D3	D4	Average
filter-sum-beamformer-LSTM [11]	filter-sum beamformer	LSTM	16.2	21.1	19.7	20.2	19.3
(A) AP-VGG [12]†	spectrogram	VGG16	**6.6**	11.0	9.2	15.7	10.6
(B) RI-VGG	RI	VGG16	9.1	18.0	19.5	17.3	16.0
(C) RI-AASIST	RI	AASIST	28.0	35.0	30.3	28.3	30.4
(D) mch-SSL-VGG	wav2vec2.0	VGG16	12.1	9.1	**8.8**	9.6	9.9
(E) mch-SSL-AASIST	wav2vec2.0	AASIST	10.1	**7.6**	10.6	**8.3**	**9.1**

† The result for AP-VGG [12] is directly cited from the original paper and was obtained using 44.1kHz audio, whereas results for other systems used 16kHz audio.

TABLE III
EERs (%) FOR EACH SYSTEM ACROSS RECORDING ENVIRONMENTS.

Recording Device	System	Recording Environment			
		Outdoor	Indoor1††	Indoor2	In-Car
D1	B	0.0	-	12.3	9.3
	E	3.1	-	8.8	11.9
D2	B	32.3	7.5	13.0	13.0
	E	2.5	2.5	9.6	9.6
D3	B	29.6	6.4	47.6	13.5
	E	10.4	3.5	23.1	8.3
D4	B	20.3	9.5	0.1	17.8
	E	1.2	4.1	8.8	10.5

†† EER for D1's Indoor1 is not available as no bona fide speech was recorded due to device crash [10].

set to two for the RI features and one for the SSL features, respectively. For RI-AASIST, the 2-channel RI features were integrated into one single feature map using 2D convolution. For SSL-AASIST (the single-channel variant of mch-SSL-AASIST), the single-channel SSL feature map was directly fed into AASIST without the convolutional integration used for the multi-channel cases.

C. Experimental Results

First, to investigate the impact of the different phase handling approaches, we evaluated the signal processing-based methods (1)-(7) detailed in III-B. Table I shows the EERs for these signal processing-based methods. The methods involving phase unwrapping (Table I (1)-(4)) showed significant performance degradation compared to AP-VGG, especially for the recording devices D1, D3, and D4. Similarly, decomposing the phase into sine and cosine components (Table I (5)) resulted in higher EERs for D2, D3, and D4. In contrast, while the method using the RI features (Table I (6)) exhibited high EER for D3, its performance was close to AP-VGG for D1 and D4. These results indicate that the methods requiring explicit phase unwrapping or decomposition lack stability and are sensitive to the variations in recording devices. Among the approaches, using the RI features demonstrated the most balanced and stable performance. Therefore, we selected RI as the feature representation for the STFT-based frontends in the subsequent comparative evaluation.

Next, Table II shows the results of the comparative investigation of the six methods as detailed in Section III-B. The average EER across devices shows that systems with an SSL-based front-end, mch-SSL-VGG and mch-SSL-AASIST, achieve lower EERs compared to other signal processing-based methods. In particular, mch-SSL-AASIST reduces the average EER by approximately 14.2% compared to AP-VGG, highlighting the effectiveness of the SSL-based features for the replay spoofing detection. Note that the AP-VGG scores are cited from [12] and were obtained using a different sampling rate (44.1 kHz); therefore, these results serve only as a reference for comparison. Furthermore, the SSL-based methods exhibit the smaller EER variations between the devices, indicating high robustness to the device differences. Regarding the combination of the frontend and the backend, the SSL-based front-ends consistently yield the lower EERs, with mch-SSL-AASIST achieving the best overall performance. For the methods using the RI features, however, the performance varies notably with the backend; RI-AASIST obtains a remarkably higher EER than RI-VGG. This suggests the importance of co-designing the frontend and backend.

To further analyze the system robustness, Table III shows the EERs in each recording environment for RI-VGG and mch-SSL-AASIST. The EERs of mch-SSL-AASIST show less fluctuation across environments than those of RI-VGG. The result suggests that the SSL-based features provide high robustness not only to the variations in the recording device characteristics but also to the changes in the recording environment. The pre-trained SSL model, wav2vec2.0-XLSR (0.3B), was trained on a diverse dataset, and its ability to extract robust features is considered a key factor in the high stability of mch-SSL-AASIST. Nevertheless, under certain conditions, such as recordings with D1 outdoors and in-car, or with D4 in Indoor2, RI-VGG achieves a lower EER. This finding suggests that while RI-VGG performs well under the specific conditions, its sensitivity to changes in the recording devices and the background noise results in a higher overall average EER.

To better understand the advantages of the multi-channel approaches, we analyzed the performance of the models for each channel. Table IV outlines the mean and standard deviation of the EERs for each system, evaluated across these single channels. For each system, the average EER of each device is consistently higher than that of the corresponding multi-channel model. This trend is consistent with previous findings, where the EER for the filter-sum-beamformer-LSTM [11] improved from 22.9% to 16.7%, and for AP-VGG [12] on D2 from 17.1% to 11.0%. These results provide evidence

TABLE IV
COMPARISON OF EERS FOR SINGLE- AND MULTI-CHANNEL SYSTEMS. FOR EACH DEVICE, EERS WERE COMPUTED PER CHANNEL, AND THE MEAN AND STANDARD DEVIATION (IN PARENTHESES) ACROSS CHANNELS ARE PRESENTED.

System	EER (%) per Recording Device (Averaged over channels)					Average EER (%) of multichannel system
	D1	D2	D3	D4	Average	
RI-VGG	10.9 (0.7)	19.9 (2.2)	20.4 (0.9)	19.6 (3.7)	17.7	16.0
RI-AASIST	26.1 (2.9)	34.5 (3.1)	34.5 (2.7)	38.0 (2.5)	33.3	30.4
SSL-VGG	11.2 (0.1)	10.4 (1.5)	13.0 (2.2)	11.1 (1.2)	11.4	9.9
SSL-AASIST	10.7 (1.6)	10.4 (1.6)	13.5 (1.0)	10.2 (1.6)	11.2	9.1

that the multi-channel methods enhance the spoofing detection accuracy. To further investigate the improvement, we analyzed the standard deviation of the EERs within each device. The results show the EER variations of several percentage points across channels, indicating differences in audio quality and acoustic characteristics for each channel. Consequently, the multi-channel approach performs better, likely due to inter-channel diversity. One possible explanation is that the integrating information from these varied channels, rather than relying on one single source, produces an ensemble-like effect that improves the overall performance.

IV. DISCUSSION

The utilization of multi-channel information can be examined from three main aspects: spatial information, speech enhancement effects, and ensemble effects. Although the 2D convolution used in this paper offers a straightforward integration method, it may not fully capture the spatial information across channels, such as inter-channel phase and timing differences. To better leverage this spatial information, future work could incorporate techniques that explicitly model inter-channel correlations. Promising directions include 3D convolutional kernels, channel-wise attention mechanisms, and strategies that capture spatio-temporal correlations [24], as well as approaches that address the temporal dynamics of mouth movements during speech [25]. Regarding the speech enhancement effect, its effectiveness can be evaluated by applying or omitting the enhancement processing. This remains an important topic for future research. Furthermore, the contribution of the ensemble effect can be evaluated. The effectiveness of this approach could be investigated by integrating the results of the individual single-channel models in a voting-based manner, which also presents a direction for future investigation.

V. CONCLUSIONS

This paper investigated the effectiveness of the SSL model for detecting replay attacks, utilizing multi-channel information. The experimental results showed that the SSL-based method achieved the lowest EER of approximately 14.2% compared to the method that relied on the conventional signal processing features. Additionally, the SSL-based frontend models demonstrated high robustness against the variations in the recording devices and the environments. However, the specific mechanisms by which multi-channel information enhances the performance are not yet fully understood. As future work, we will clarify the respective contributions of the spatial information utilization, speech enhancement effects, and ensemble effects.

REFERENCES

[1] D. G. Shin and M. S. Jun, "Home IoT device certification through speaker recognition," *Proc. ICACT*, pp. 600–603, 2015.

[2] 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.

[3] https://www.asvspoof.org/.

[4] J. Yamagishi, X. Wang, M. Todisco, *et al.*, "ASVspoof 2021: Accelerating progress in spoofed and deepfake speech detection," *Proc. ASVspoof Challenge Workshop*, pp. 47–54, 2021.

[5] X. Wang, H. Delgado, H. Tak, *et al.*, "ASVspoof 5: Crowdsourced speech data, deepfakes, and adversarial attacks at scale," *Proc. ASVspoof Challenge Workshop*, pp. 1–8, 2024.

[6] M. Li, Y. Ahmadiadli, and X.-P. Zhang, "Audio anti-spoofing detection: A survey," *arXiv:2404.13914*, 2024.

[7] S. Shiota, F. Villavicencio, J. Yamagishi, N. Ono, I. Echizen, and T. Matsui, "Voice liveness detection algorithms based on pop noise caused by human breath for automatic speaker verification," *Proc. Interspeech*, pp. 239–243, 2015.

[8] R. Yaguchi, S. Shiota, N. Ono, and H. Kiya, "Replay attack detection using generalized cross-correlation of stereo signal," *Proc. EUSIPCO*, pp. 1–5, 2019.

[9] Y. Gong and C. Poellabauer, "Protecting voice controlled systems using sound source identification based on acoustic cues," *Proc. ICCCN*, pp. 1–9, 2018.

[10] Y. Gong, J. Yang, J. Huber, M. MacKnight, and C. Poellabauer, "ReMASC: Realistic replay attack corpus for voice controlled systems," *Proc. Interspeech*, pp. 2355–2359, 2019.

[11] Y. Gong, J. Yang, and C. Poellabauer, "Detecting replay attacks using multi-channel audio: A neural network-based method," *IEEE Signal Processing Letters*, vol. 27, pp. 920–924, 2020.

[12] Z. Li, C. Shi, T. Zhang, *et al.*, "Robust detection of machine-induced audio attacks in intelligent audio systems with microphone array," *Proc. ACM SIGSAC*, pp. 1884–1899, 2021.

[13] J. Benesty, J. Chen, and Y. Huang, *Microphone array signal processing*. Springer Science & Business Media, 2008, vol. 1.

[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv:1409.1556*, 2015.

[15] H. Tak, M. Todisco, X. Wang, J. W. Jung, J. Yamagishi, and N. Evans, "Automatic speaker verification spoofing and deepfake detection using wav2vec 2.0 and data augmentation," *Proc. Odyssey*, pp. 112–119, 2022.

[16] J. W. Jung, H. S. Heo, H. Tak, *et al.*, "AASIST: Audio anti-spoofing using integrated spectro-temporal graph attention networks," *Proc. ICASSP*, pp. 6367–6371, 2022.

[17] T. N. Sainath, R. J. Weiss, K. W. Wilson, *et al.*, "Multichannel signal processing with deep neural networks for automatic speech recognition," *Trans. on TASLPRO*, vol. 25, no. 5, pp. 965–979, 2017.

[18] D. S. Williamson, Y. Wang, and D. Wang, "Complex ratio masking for monaural speech separation," *Trans. on TASLPRO*, vol. 24, no. 3, pp. 483–492, 2015.

[19] D. S. Williamson, Y. Wang, and D. Wang, "Complex ratio masking for joint enhancement of magnitude and phase," *Proc. ICASSP*, pp. 5220–5224, 2016.

[20] S. W. Fu, T. Y. Hu, Y. Tsao, and X. Lu, "Complex spectrogram enhancement by convolutional neural network with multi-metrics learning," *Proc. MLSP*, pp. 1–6, 2017.

[21] M. Tammen and S. Doclo, "Deep multi-frame mvdr filtering for binaural noise reduction," pp. 1–5, 2022.

[22] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *Proc. ICLR*, 2019.

[23] A. Babu, C. Wang, A. Tjandra, *et al.*, "XLS-R: Self-supervised cross-lingual speech representation learning at scale," *Proc. Interspeech*, pp. 2278–2282, 2022.

[24] S. Horiguchi, Y. Takashima, P. García, S. Watanabe, and Y. Kawaguchi, "Multi-channel end-to-end neural diarization with distributed microphones," *Proc. ICASSP*, pp. 7332–7336, 2022.

[25] Q. Yang, K. Cui, and Y. Zheng, "Room-scale voice liveness detection for smart devices," *Trans. on TDSC*, vol. 21, no. 5, pp. 4982–4996, 2024.