Self-weighted score fusion for the SASV Challenge 2022

Thomas Thebaud[1,2], *Anthony Larcher*[2], *Gaël LeLan*[1]

[1]Orange Innovation
[2]Laboratoire d'Informatique de l'Université du Mans

thomas.thebaud@orange.com, anthony.larcher@univ-lemans.fr, gael.lelan@orange.com

Abstract

This paper describes the attentive system proposed for the SASV 2022 challenge [1]. This challenge is following the ASVspoof 2019 challenge [2], which goal was to promote counter measures against spoofing, for automatic speaker verification systems, both for physical and logical access. Considering the low equal error rates assured by speaker verification and counter measure subsystems separately, the SASV challenge seeks to find the best global EER by using them jointly. Using the baseline subsystems, we propose a self-weighted fusion system to combine their outputs embeddings and achieve a 1.478% EER for our Spoofing Aware Speaker Verification system.

Index Terms: automatic speaker verification, anti spoofing counter measures, spoofing aware speaker verification

1. Introduction

Over the past years, we have seen steady progress in speech-based authentication technologies. Automatic speaker verification (ASV) systems have reached lower and lower [3, 4, 5, 6, 7, 8] equal error rates (EER [9]), down to less than 0.5% [7, 8]. This progression, as well as the incorporation of microphone devices in the population, made speech-based authentication usages more and more common through the population, jointly rising the privacy stakes linked to speech and speech-based systems.

However, high performances for genuine speakers does not imply an effective resistance to attackers. The active attempts to falsify or replay voice characteristics in order to gain unauthorized access are referred to respectively as spoofing attacks or presentation attacks (ISO/IEC 30107-1), and they are currently one of the biggest threats for voice biometric systems. To increase the protection against those attacks, Automatic Speaker Verification Spoofing and Counter measure initiative gathered the research community around the ASVspoof challenges, which objectives were to detect spoofing and replay attacks under various conditions. ASVspoof challenges were first organized in 2015 [10], then pushed further on replay attacks in 2017 [11], logical and physical accesses in 2019 [2] and deep fakes detection in 2021 [12]. The counter measures (CM) proposed for those challenges can deliver EER of less than 2% [13] for the detection of spoofing attacks.

The purpose of the SASV challenge 2022 [1] is to further improve robustness to both naive impostors (by the ASV system) and to the spoofing attacks (by the CM system) by making both systems work together. The related works section presents the ASV and CM state of the art systems. The data to be used is set by the challenge, and is going to be presented section 3. The sections 4 and 5 respectively describe the systems used and the results obtained. The last section will conclude on our results and possible future works on the subject.

2. Related works

2.1. Automatic Speaker Verification

Automatic Speaker Verification seeks to differentiate speakers based on their voice. The voice utterances are often preprocessed into spectral/temporal representations using determinist methods such as the MFCC computation [14] (one of the main methods used for ASV preprocessing), or more recently, neural methods such as WavLM [8] (currently gives state of the art performances).

Then from the bi-modal representations are computed high dimensional vectors representative of the speaker's discriminant characteristics. Such vectors are commonly referred to as *x-vectors*, from the first system that presented this architecture [5] in 2018. The architecture evolved using the ResNet [15] architecture, temporal self-attention [16](TDNN system) and a squeeze-excitation module [7](ECAPA-TDNN system). The ECAPA-TDNN system trained with 80-dimension MFCC on VoxCeleb2 [17] has a 0.87% EER [7] on the VoxCeleb1 test set [18] (presented section 3).

When the ECAPA-TDNN is trained with WavLM [8] preprocessing instead of MFCC, the test EER on VoxCeleb1 drops to 0.383% [8]. However, because of the data constraints of the SASV challenge, we can not use the pretrained WavLM system for pre-processing, so we will use MFCC for pre-processing.

2.2. Counter Measures

The audio anti-spoofing task is one of the base requirements to bring a voice-based authentication system to reality. The goal of the task is to sort genuine (*bona-fide*) authentication attempts from spoofing attacks, to improve the global robustness of the system to attacks. The ASVspoof community was brought together around a series of challenges [10, 11, 2, 12] about audio anti-spoofing.

As for ASV systems (see subsection 2.1), Counter Measure systems proved to behave best when considering both temporal and spectral dimensions of speech utterances [19]. CM systems first used similar pre-processing (Filter Banks [20]) before switching to end-to-end systems, using directly the raw audio [21]. The spectro-temporal representations are then computed by the first layers of a RawNet2 [22] adapted for anti-spoofing. A well used solution for audio anti-spoofing is the use of separate Graph Attention Networks (GAN [20, 21, 13]) for the spectral and the temporal dimensions of the extracted representations.

The AASIST model [13] provide state of the art performances using those GAN and merging temporal and spectral graphs with a *max graph* operation. It achieves 1.13% EER on detecting the 13 attacks proposed in the ASVspoof LA challenge 2019 [2].

3. Data

The data used is a fixed parameter of the challenge, we are meant to use the datasets presented in the table 1.

Table 1: *Table of the datasets used for the challenge, with number of speakers and utterances included in each dataset.*

Datasets	Speakers	Utterances
VoxCeleb 1 test [18]	40	4 874
VoxCeleb 2 [17]	5994	1 045 732
ASVspoof 2019 [23]		
LA train partition	20	25 380
LA development partition	20	24 844
LA evaluation partition	49	102 579

The VoxCeleb2 dataset is composed only of bonafide utterances. The ASVspoof datasets are composed of around 10% bonafide utterances (from the VCTK dataset [24]) and 90% spoofing utterances, generated using Voice Conversion [25] and Text To Speech [26] Systems. There are 19 different spoofing attacks performed using various VC and TTS systems, readers are referred to [23] for full details.

4. The systems

The system is composed of three subsystems :

1. The ASV subsystem, which produce an *x-vector* from a raw speech utterance, containing discriminant data about the speaker.
2. The CM subsystem, which produce an embedding containing data about the type of spoofing attack happening.
3. The fusion subsystem, which produce a confidence score (the user is authorized if the score is high enough) from three vectors :
   (a) An enrollment *x-vector* from an enrollment utterance of the speaker trying to authenticate.
   (b) A test *x-vector* from the test utterance provided.
   (c) A CM embedding extracted from the same test utterance.

Those subsystems are described in the following sections, but the experiments were only conducted on the last subsystem : the fusion of the scores.

4.1. Automatic Speaker Verification subsystem

For our ASV subsystem, we used MFCC and the ECAPA-TDNN [7] system proposed in the second baseline of the challenge [1], as presented section 2. It was pretrained using the VoxCeleb2 [17] dataset. The results of this system alone are presented in the table 2. It presents a SV-EER of 1.63% ASVspoof 2019 LA evaluation dataset [23] (presented section 3). This system outputs *x-vectors* of dimension 192 from raw audio.

4.2. Counter Measure Subsystem

For our CM subsystem, we also used the baseline proposed system : the AASIST model [13] presented section 2.2, pre-trained on the LA train partition of the ASV spoof 2019 dataset [23]. This system outputs *embeddings* of dimension 160 from raw audio.

To provide an estimation of the CM system performances alone, we used the second baseline only with the CM outputs, the results are presented line 2 of the table 2

4.3. Score Fusion subsystem

The Score fusion subsystem takes as input the output of both previous subsystems, and outputs a 2 dimensional vector, where the higher of both score determine if the user is accepted or not. It is trained on the ASVspoof 2019 LA train partition [23] presented section 3, and evaluated on the development and evaluation partitions. We tried different variations from the baseline2 provided by the challenge [1] : Those are detailed in the following subsections, as well as the baseline2.

4.3.1. The Baseline

The baseline system is a multi-layer perceptron (MLP [27]), taking the concatenate three embeddings as an input ($192 \times 2 + 160 = 544$ dimensions). It is composed of 4 Layers of dimensions [544, 256, 128, 64, 2] separated by LeakyReLU activation functions. It is trained for 10 epochs with the ADAM optimizer [28] and a learning rate of 10^{-4}. The baseline parameters are presented line 3 of the table 2.

4.3.2. Facilitating the Cosine Similarity

The performances of ASV systems are usually measured using EER and Cosine similarity. Cosine Similarity, for two embeddings of unit norm, is simply the dot product of the two. To help the system comparing embeddings, we added the element-wise product of the the x-vectors to the inputs, meaning the input is now of dimension 736 ($= 192 * 3 + 160$). This system is now composed of 4 Layers of dimensions [736, 256, 128, 64, 2] separated by LeakyReLU activation functions. The Cosine Similarity Facilitation model is presented line 4 of the table 2.

4.3.3. Weighted Fusion

The global authentication system is supposed to let a user pass following two conditions :

1. If the utterance is not a spoofing attack.
2. If the user is really who he pretends to be.

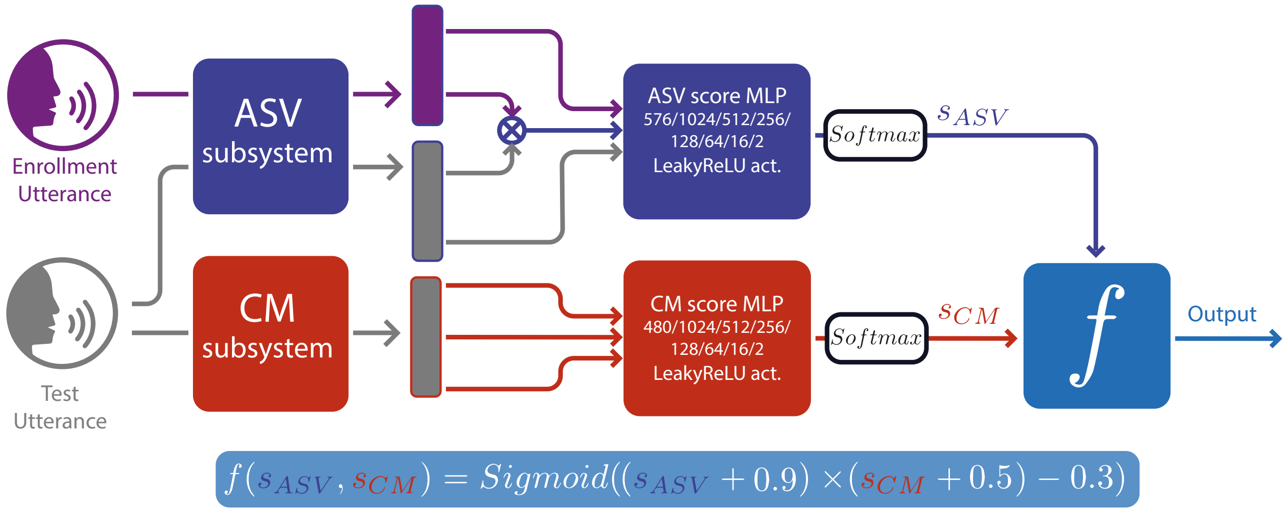
The first one is determined by the CM subsystem, the second by the ASV subsystem. We propose to use the embeddings of both subsystems separately, then merge them. This system is detailed in the figure 1

First, we compute a score from the ASV embeddings using a 6-layers MLP (dimensions [192×3, 1024, 512, 256, 128, 64, 2], separated by LeakyReLU activation functions), with a Softmax function at the end, to get a score s_{ASV} between 0 and 1. This MLP takes as inputs the two ASV vectors and their dot product, concatenated.

Then, we compute a score from the CM embedding using a similar MLP (layers of dimensions [160×3, 1024, 512, 256, 128, 64, 2]) and get a second score (s_{CM}) between 0 and 1. This MLP takes as inputs the CM vector repeated three times and concatenated.

Finally, we merge those score using a function presented in the equation 1, with manually chosen coefficients. This function acts like a parametric **AND** logical gate : for the result to be

Figure 1: *Schematic of the weighted fusion scoring subsystem used in the 4.3.3 section.*



high, both scores needs to be high.

$$\begin{aligned} f(s_{ASV}, s_{CM}) = & Sigmoid(\\ & (s_{ASV} + 0.9) \\ & \times (s_{CM} + 0.5) \\ & - 0.3) \end{aligned} \tag{1}$$

The results of this subsystem are presented line 5 of the table 2.

4.3.4. Trainable Weighted Fusion

To further improve our system, we tried a function with variable coefficients, being trainable parameters of the system. With $a, b, c, d, e \in \mathbb{R}^5$ the trainable parameters, we can re define the merge function as in the equation 2 We added ReLU functions so that if some parameters were negatives, we would still be using positives scores in the product.

$$\begin{aligned} f(s_{ASV}, s_{CM}) = & Sigmoid(\\ & ReLU(a \times s_{ASV} + b) \\ & \times ReLU(c \times s_{CM} + d) \\ & - e) \end{aligned} \tag{2}$$

The results of using that merge function are presented line 6 if the table 2.

4.3.5. Self-Weighted fusion (Our Best)

To further improve the adaptability of the system, we decided to compute the coefficients for each utterance, using another MLP, as presented in the figure 2. The scalar coefficients, $(\alpha, \beta, \gamma, \delta) \in [0, 1]^4$, are computed from all vectors using a third MLP similar to the two previous ones (with layers of dimensions $[192 \times 2 + 160, 1024, 512, 256, 128, 64, 2]$). We only chose 4 parameters instead of 5 previously, because a variable fifth one would make the SASV-EER rise. The function used to merge the score is presented in the equation 3.

$$\begin{aligned} f(\alpha, \beta, \gamma, \delta, s_{ASV}, s_{CM}) = & Sigmoid(\\ & ReLU(\alpha \times s_{ASV} + \beta) \\ & \times ReLU(\gamma \times s_{CM} + \delta) \\ & - 0.5) \end{aligned} \tag{3}$$

The results of this subsystem are presented line 7 of the table 2.

5. The results

In this section, the results for different fusion subsystems are presented. All fusion subsystems were trained for 10 epochs using the ASVspoof 2019 LA train partition, presented section 3. The results are presented in the table 2.

Comparing the lines 1 and 3 in the table, we see a huge drop in the SV-EER performances, because ASV systems EER are usually computed using Cosine Similarity or PLDA as scoring systems, not 3-layers MLP. Thus, facilitating the computation of a Cosine Similarity by adding the dot product of the ASV embeddings (line 4 of the table 2) dropped the SV-EER from 11.48% to 2.53%.

We can see that the best results (considering lines 1-4 of the table 2) for SV-EER and SPF-EER were obtained using respectively only the ASV and the CM embeddings. This is gave us the idea to process separately the embeddings, before merging the results. The merging was first made using handcrafted coefficients, leading to a small improvement, as seen line 5 of the table 2. Then, we used trainable coefficients, that lead to further improvement (line 6).

Finally, we used utterance-specific coefficients, thus leading to a self-weighted fusion. The separated processing, combined to the cosine similarity facilitation and self-weighted fusion of the scores gave us a **1.48% SASV-EER** for on the evaluation partition, as seen line 7 of the table 2.

6. Conclusions

This article presents a submission system to the SASV challenge 2022 [1]. The goal of the challenge was to make a spoofing aware speaker verification, getting at the same time good Automatic Speaker Verification and Counter Measure performances. We based our proposition on the second baseline of the challenge :

1. An ECAPA-TDNN [7] subsystem producing embeddings for the ASV sub-task.
2. An AASIST [13] subsystem producing an embedding for the CM sub-task.

Figure 2: *Schematic of the self-weighted fusion scoring subsystem used in the 4.3.5 section.*

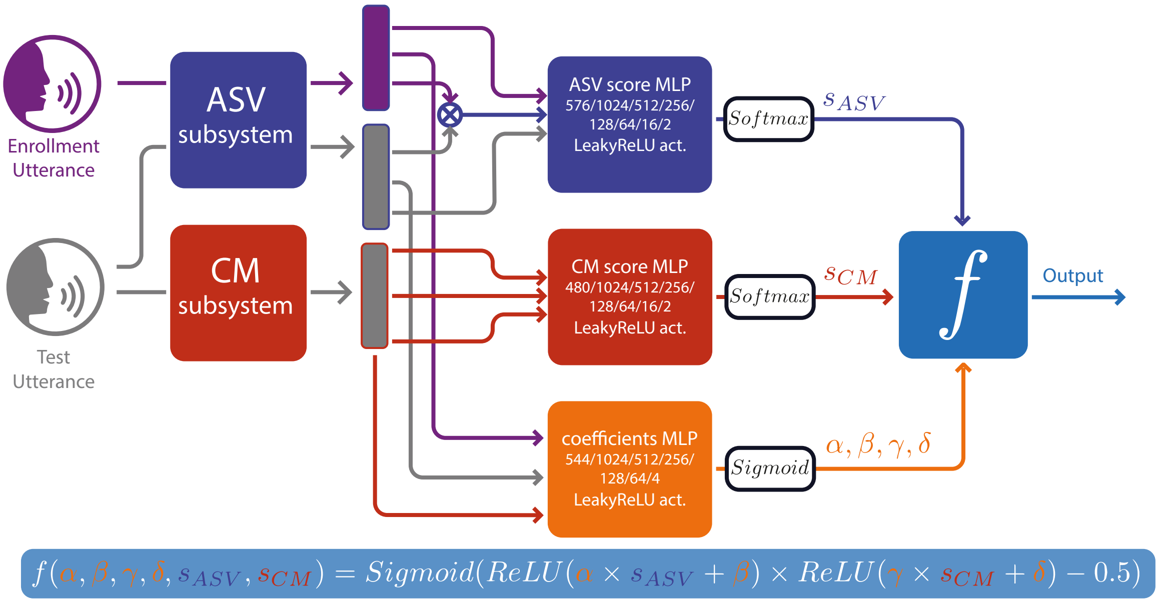

Enrollment Utterance
Test Utterance
ASV subsystem
CM subsystem
ASV score MLP 576/1024/512/256/128/64/16/2 LeakyReLU act.
CM score MLP 480/1024/512/256/128/64/16/2 LeakyReLU act.
coefficients MLP 544/1024/512/256/128/64/4 LeakyReLU act.
$Softmax$
$Softmax$
$Sigmoid$
s_{ASV}
s_{CM}
$\alpha, \beta, \gamma, \delta$
f
Output
$f(\alpha, \beta, \gamma, \delta, s_{ASV}, s_{CM}) = Sigmoid(ReLU(\alpha \times s_{ASV} + \beta) \times ReLU(\gamma \times s_{CM} + \delta) - 0.5)$

Table 2: *The three different EERs (%) for the SASV 2022 development and evaluation partitions. The results are shown for the base ASV system (ECAPA-TDNN), the second baseline solution and the proposed system. The last column show the number of parameters for each of the fusion subsystems considered. Best results are* ***bold****, second bests are italic.*

		SV-EER		**SPF-EER**		**SASV-EER**		N parameters
		Dev	Eval	Dev	Eval	Dev	Eval	of the fusion subsystem
1	ECAPA-TDNN	**1.88**	**1.63**	20.30	30.75	17.38	23.83	
2	AASIST + DNN	46.90	47.48	**0.067**	0.71	15.83	24.30	180K
3	Baseline2(back-end ensemble model)	12.87	11.48	0.13	0.78	4.85	6.37	180K
4	Cosine Similarity Facilitation (Ours)	2.49	2.53	0.31	2.36	*1.38*	2.43	229K
5	w Weighted Fusion (Ours)	5.32	3.89	0.076	*0.60*	2.37	2.35	2.5M
6	w Trainable Weighted Fusion (Ours)	3.09	2.89	0.25	0.97	1.63	*1.81*	2.5M
7	w Self-Weighted Fusion (Ours)	*1.89*	*2.44*	*0.071*	**0.56**	**0.94**	**1.48**	3.7M

3. A DNN fusion subsystem.

We proposed a self-weighted fusion subsystem to merge the embeddings produced by the ASV and CM subsystems, using dot product of the ASV embeddings to facilitate the cosine similarity computation by the system. Our system achieve a **1.478% SASV-EER** on the evaluation partition.

Our best system did not present the best Evaluation SV-EER or the best Development SPF-EER, from those presented table 2. In future works, we target the improvement of the separated performances on the SV and SPF sub-tasks, that should lead to a direct improvement in the global SASV task. We also wanted to measure the performances for an ASV system trained using WavLM pre-processing [8], but this is out of the challenge limits, given that WavLM is a system pre-trained with another set of data than the ones authorized.

7. References

[1] J.-w. Jung, H. Tak, H.-j. Shim, H.-S. Heo, B.-J. Lee, S.-W. Chung, H.-G. Kang, H.-J. Yu, N. Evans, and T. Kinnunen, "Sasv challenge 2022: A spoofing aware speaker verification challenge evaluation plan," *arXiv preprint arXiv:2201.10283*, 2022.

[2] M. Todisco, X. Wang, V. Vestman, M. Sahidullah, H. Delgado, A. Nautsch, J. Yamagishi, N. Evans, T. Kinnunen, and K. A. Lee, "Asvspoof 2019: Future horizons in spoofed and fake audio detection," *arXiv preprint arXiv:1904.05441*, 2019.

[3] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using gmm supervectors for speaker verification," *IEEE signal processing letters*, vol. 13, no. 5, pp. 308–311, 2006.

[4] N. Dehak, P. J. 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, 2010.

[5] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.

[6] 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)*. IEEE, 2016, pp. 5115–5119.

[7] B. Desplanques, J. Thienpondt, and K. Demuynck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," *arXiv preprint arXiv:2005.07143*, 2020.

[8] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *et al.*, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *arXiv preprint arXiv:2110.13900*, 2021.

[9] J.-M. Cheng and H.-C. Wang, "A method of estimating the equal error rate for automatic speaker verification," in *2004 International Symposium on Chinese Spoken Language Processing*. IEEE, 2004, pp. 285–288.

[10] Z. Wu, T. Kinnunen, N. Evans, J. Yamagishi, C. Hanilçi, M. Sahidullah, and A. Sizov, "Asvspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge," in *Sixteenth annual conference of the international speech communication association*, 2015.

[11] T. Kinnunen, M. Sahidullah, H. Delgado, M. Todisco, N. Evans, J. Yamagishi, and K. A. Lee, "The asvspoof 2017 challenge: Assessing the limits of replay spoofing attack detection," *ISCA*, 2017.

[12] J. Yamagishi, X. Wang, M. Todisco, M. Sahidullah, J. Patino, A. Nautsch, X. Liu, K. A. Lee, T. Kinnunen, N. Evans *et al.*, "Asvspoof 2021: accelerating progress in spoofed and deepfake speech detection," *arXiv preprint arXiv:2109.00537*, 2021.

[13] J.-w. Jung, H.-S. Heo, H. Tak, H.-j. Shim, J. S. Chung, B.-J. Lee, H.-J. Yu, and N. Evans, "Aasist: Audio anti-spoofing using integrated spectro-temporal graph attention networks," *arXiv preprint arXiv:2110.01200*, 2021.

[14] V. Tiwari, "Mfcc and its applications in speaker recognition," *International journal on emerging technologies*, vol. 1, no. 1, pp. 19–22, 2010.

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

[16] Y. Zhu, T. Ko, D. Snyder, B. Mak, and D. Povey, "Self-attentive speaker embeddings for text-independent speaker verification." in *Interspeech*, vol. 2018, 2018, pp. 3573–3577.

[17] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," *arXiv preprint arXiv:1806.05622*, 2018.

[18] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.

[19] J. Yang, R. K. Das, and H. Li, "Significance of subband features for synthetic speech detection," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2160–2170, 2019.

[20] H. Tak, J.-w. Jung, J. Patino, M. Todisco, and N. Evans, "Graph attention networks for anti-spoofing," *arXiv preprint arXiv:2104.03654*, 2021.

[21] H. Tak, J.-w. Jung, J. Patino, M. Kamble, M. Todisco, and N. Evans, "End-to-end spectro-temporal graph attention networks for speaker verification anti-spoofing and speech deepfake detection," *arXiv preprint arXiv:2107.12710*, 2021.

[22] H. Tak, J. Patino, M. Todisco, A. Nautsch, N. Evans, and A. Larcher, "End-to-end anti-spoofing with rawnet2," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6369–6373.

[23] 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*, vol. 64, p. 101114, 2020.

[24] C. Veaux, J. Yamagishi, K. MacDonald *et al.*, "Superseded-cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit," *University of Edinburgh. The Centre for Speech Technology Research (CSTR)*, 2016.

[25] S. H. Mohammadi and A. Kain, "An overview of voice conversion systems," *Speech Communication*, vol. 88, pp. 65–82, 2017.

[26] Y. Ning, S. He, Z. Wu, C. Xing, and L.-J. Zhang, "A review of deep learning based speech synthesis," *Applied Sciences*, vol. 9, no. 19, p. 4050, 2019.

[27] H. Ramchoun, Y. Ghanou, M. Ettaouil, and M. A. Janati Idrissi, "Multilayer perceptron: Architecture optimization and training," 2016.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.