INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

DoubleDeceiver: Deceiving the Speaker Verification System Protected by Spoofing Countermeasures

Mengao Zhang[1*†], *Ke Xu*[2*‡], *Hao Li*[3], *Lei Wang*[2], *Chengfang Fang*[2], *Jie Shi*[2]

[1]Nanyang Technological University, Singapore
[2]Huawei International, Singapore
[3]Huawei Technology, China

zh0024ao@e.ntu.edu.sg, {xuke64, lihao350, wang.lei12, fang.chengfang, shi.jie1}@huawei.com

Abstract

Automatic Speaker Verification (ASV) systems are vulnerable to various attacks, especially spoofing attacks, and therefore are typically protected by spoofing countermeasures. However, both spoofing countermeasures and ASV models are vulnerable to adversarial attacks. We propose DoubleDeceiver - a novel black-box attack method that incorporates text-to-speech synthesis and adversarial attack to deceive ASV systems even with the protection of spoofing countermeasures. Although the surrogate models and victim models differ in architectures, DoubleDeceiver achieved a successful attack rate (SAR) as high as 98.3%. DoubleDeceiver identified the vulnerabilities of ASV systems and issued a warning that solely relying on the spoofing countermeasures is not reliable to protect ASV systems' security. This work encourages the development of more secure anti-spoofing and ASV systems by highlighting the need to consider composite attacks in future designs.

Index Terms: speaker verification, spoofing countermeasure, adversarial attack, black-box attack, text-to-speech

1. Introduction

ASV systems aim to verify whether an input utterance is uttered by the registered speaker. ASV can be applied in many scenarios that require a high level of security and privacy [1,2], such as access control, and banking services. The advancement in Deep Neural Networks (DNN) promoted the development of ASV systems which resulted in state-of-the-art performance [2].

Despite the good performances, ASV systems are vulnerable to various attacks. The mainstream methods of attacks are called spoofing attacks [3] which include voice conversion, text-to-speech (TTS), and replay attacks. To mitigate these threats, spoofing countermeasures (anti-spoofing) [4] have been developed, which are widely used to protect real-world ASV systems [5]. However, the DNN-based systems are also susceptible to adversarial attacks [6, 7].

Previous works [5, 8–17] investigated the vulnerability of various ASV or anti-spoofing models to adversarial attacks in various settings. Few works achieved a high SAR in the black-box scenario. Notably, most of the research [8–17] are limited to conducting adversarial attacks against standalone models, while few studies investigate the ASV systems protected by spoofing countermeasures, which are more practical in a real-world setup. Although anti-spoofing models are designed to protect ASV systems from spoofing attacks, it has been proved in our experiments that they achieve outstanding performance in preventing adversarial attacks tailored for ASV models. For example, as one of the SOTA adversarial attacks against standalone ASV models, FAKEBOB's [13] performance heavily degraded by 70% when attacking ASV systems protected by spoofing countermeasures because only 38% of the adversarial examples can bypass an open-source anti-spoofing model.

To the best of our knowledge, the most related work [5] that targets the joint system of anti-spoofing and ASV models only considered a white-box scenario and was evaluated on only one ASV model. Furthermore, the area of anti-spoofing and ASV evolve quickly in recent years, and many of the previously studied models become obsolete. Thus, a real-world setup (i.e., an advanced black-box ASV system protected by spoofing countermeasures) renders most previous attacks impractical and the vulnerability of such a joint system still remains uncertain.

We propose DoubleDeceiver - a novel targeted black-box adversarial attack method against advanced ASV systems protected by spoofing countermeasures. DoubleDeceiver focuses on generating adversarial examples with transferability which are challenging [13]. Another challenge is to generate adversarial examples that can deceive two models with different functionalities simultaneously. DoubleDeceiver incorporates speech synthesis and adversarial attacks. Instead of relying on source speakers, DoubleDeceiver first synthesizes voices that mimic the target speaker using a zero-shot multi-speaker TTS (ZS-TTS) synthesizer [18]. Then it adds adversarial perturbations to the raw waveforms of the synthesized voices using a gradient-based method to bypass spoofing countermeasures and deceive the ASV model. DoubleDeceiver has the following advantages:

• **Effective and efficient.** The synthesized voices act as strong bases for generating adversarial examples, located closely to the decision boundaries of ASV models. DoubleDeceiver achieves an overall SAR as high as 98.3% even when the surrogate models and victim models are of different architectures. Besides, DoubleDeceiver is still effective with only small perturbations, achieving a SAR of 69% on average and 91.1%, the highest.

• **Practical.** DoubleDeceiver attacks advanced ASV systems protected by spoofing countermeasures, which are widely deployed in real-world applications. By utilizing ZS-TTS, adversarial examples with any length and content could be generated without the need for a source speaker, which gives the attackers great flexibility. By only applying publicly available surrogate models, DoubleDeceiver launches targeted attacks that better fit the goal of attackers in real world, i.e., impersonating the target speaker, and inducing severe threats to real-life users.

• **Silent.** DoubleDeceiver targets black-box scenarios where attackers have no access to the victim models even its output. Only a short utterance from the target speaker, e.g., voice recordings, is needed for ZS-TTS, which are both easily accessible in real-world, and stealthy attacks are enabled.

*Both authors contributed equally to this work.
†Work done during internship at Huawei International.
‡Corresponding author.

10.21437/Interspeech.2023-371

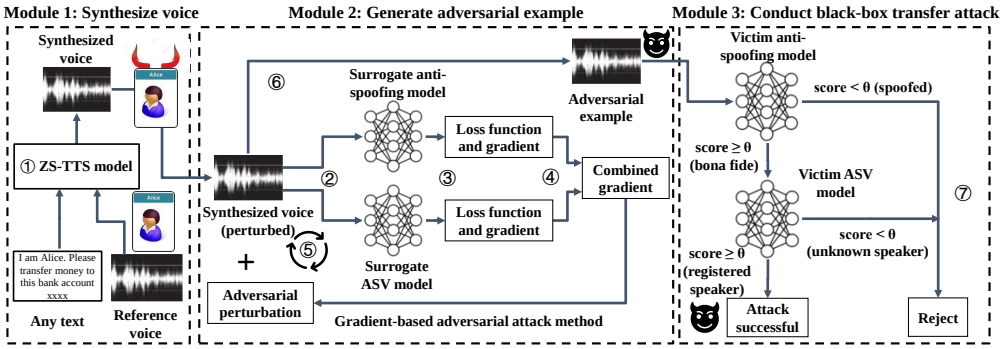

Figure 1: *DoubleDeceiver - attack flow. The arrows represent the order of steps.*

2. Adversarial attacks

• **Adversarial attack methods.** Adversarial attacks [6, 7] became a significant threat to machine learning models. Different adversarial attack methods aim to generate a small perturbation δ such that $f(x+\delta) = y^{target}$ where x is the clean input, y^{target} is the attacker's desired output and $y^{target} \neq f(x)$. Gradient-based methods, which is one of the most widely used adversarial attacks, utilize gradients to search for adversarial perturbation ϵ. Representative gradient-based methods include FGSM [7], BIM [19], PGD [20] and MI-FGSM [21].

• **Adversarial attack type.** In general, there are two categories of adversarial attacks: white-box and black-box attacks. In white-box attacks, the attacker has full access to the DNN system's information, e.g., architecture and parameters. The attackers can access at most the output of the system in black-box attacks, which are more challenging and practical. The adversarial attacks can also be categorized into targeted or non-targeted attacks according to the attacker's goal. In the context of ASV, non-targeted attacks aim to let the ASV system misclassify legal users, while targeted attacks aim to impersonate the registered speaker. DoubleDeceiver assumes a targeted black-box attack scenario as it is more practical and threatening.

• **Adversarial attacks on ASV and anti-spoofing models.** Previous works [8,9] proved that the most commonly used standalone ASV models, i-vector and x-vector, are vulnerable to adversarial attacks. The adversarial perturbation can be either added to acoustic features [10] or raw waveform of the audio [11]. Some further explored the effectiveness of adversarial examples in physical access scenarios [12, 13], while others tried to make adversarial perturbations more imperceptible [14, 15]. Standalone anti-spoofing models are also proved to be vulnerable to adversarial attacks in both white-box and black-box scenarios [16]. Zhang et al. [17] further investigated black-box transfer-based attacks against anti-spoofing models using model ensemble to enhance the transferability of adversarial examples.

However, adversarial attacks against a joint system of anti-spoofing and ASV models are not well studied and such systems deployed in real-world applications still face the potential threat of adversarial attacks. By carrying out adversarial attacks on such systems, this paper aims to highlight the need for a more secure and robust ASV application.

3. Methodology

DoubleDeceiver consists of three modules as shown in Figure 1: (1) synthesize voice, (2) generate adversarial example, and (3) conduct black-box transfer attack. DoubleDeceiver follows a modular setting, i.e., the ZS-TTS model, gradient combination function, and gradient-based adversarial attack method are highly flexible, as shown in Algorithm 1. This work employs the gradient combination function $g_t = g_{AS} + g_{ASV}$, and the gradient-based method BIM. Other ways of combining gradients and adversarial attacks will be explored in the future.

Algorithm 1 DoubleDeceiver. Parameter ω is specific to the attack method, e.g., step size α for BIM

Input: ZS-TTS model L_{TTS}, surrogate anti-spoofing model L_{AS}, surrogate ASV model L_{ASV}, a short reference voice γ, a short string S, perturbation budget ϵ, number of iteration T, adversarial attack method f_{adv} parameterized by ϵ and ω, gradient combination function f_g, loss function J that takes target class and model's output to give the loss.
Output: Adversarial example x^* with speech content S

$x_0^* \leftarrow$ synthesized voice from L_{TTS} subject to γ and S
for iteration time $t \leftarrow 1$ to T **do**
  $g_{AS} \leftarrow \nabla_x J_{AS}(x_{t-1}^*, y_{AS}^{target})$ ▷ L_{AS}'s gradient
  $g_{ASV} \leftarrow \nabla_x J_{ASV}(x_{t-1}^*, y_{ASV}^{target})$ ▷ L_{ASV}'s gradient
  $g_t \leftarrow f_g(g_{AS}, g_{ASV})$
  $x_t^* \leftarrow f_{adv}(g_t, x_{t-1}^*, x_0^*; \epsilon, \omega)$
end for
return x_T^*

Generating adversarial examples that deceive two models is challenging because the perturbations required to fool models with different functionalities may conflict with each other in the direction, which might diminish the effectiveness of adversarial examples. As no clue about the victim model can be accessed in the black-box setting, the resultant cross-architecture setting degrades the transferability of adversarial examples. DoubleDeceiver combines speech synthesis with adversarial attacks to achieve better results.

3.1. Threat model

We consider a practical black-box threat model where the utterance is accepted only if it passes both the anti-spoofing model and the ASV model in logical access scenarios. The attacker aims to impersonate the registered speaker and is able to obtain a short utterance from the target speaker but does not have any information about the victim system. Then the attacker uses publicly available models to generate synthesized voice and adversarial examples. The victim models each have a preset threshold θ and give an acceptance decision only if the input voice's score is not less than θ. There are other ways of combin-

Table 1: *Overall SARs(%). Surrogate anti-spoofing model: LCNN, victim anti-spoofing model: ResNet-OC.*

Sur.		T-P			Res.			Raw.			E-T		
SNR(dB)	Vic. / ϵ	Res.	Raw.	E-T	T-P	Raw.	E-T	T-P	Res.	E-T	T-P	Res.	Raw.
-	0.000	0.0	0.0	0.0	2.5e-4	0.0	0.0	2.5e-4	0.0	0.0	2.5e-4	0.0	0.0
27.1	0.004	81.3	81.0	80.0	75.9	68.9	64.2	80.3	73.4	64.6	91.1	87.4	88.7
21.6	0.008	93.5	93.7	93.1	89.9	83.8	79.4	90.0	83.0	76.0	95.8	94.6	95.4
18.2	0.012	96.5	96.3	96.4	**90.4**	86.0	83.7	**90.2**	85.0	79.5	**96.1**	96.8	**96.6**
15.8	0.016	97.6	97.5	97.6	89.3	**87.6**	85.5	89.7	**85.5**	**80.3**	96.0	97.8	96.5
13.9	0.020	**98.3**	**98.2**	**98.3**	87.3	87.3	**85.9**	88.6	85.5	80.0	94.6	**98.0**	96.2

* Sur. and Vic. represent surrogate and victim ASV models, respectively. T-P, Res., Raw., and E-T represents TDNN-PLDA, ResNetSE34V2, RawNet3, and ECAPA-TDNN respectively. The SARs are tested on the same 4,000 base voices with different ϵ and models. The same abbreviations and base voices are applied in the rest of the tables.

Table 2: *Overall SARs(%). Surrogate anti-spoofing model: AASIST, victim anti-spoofing model: ResNet-OC.*

Sur.		T-P			Res.			Raw.			E-T		
SNR(dB)	Vic. / ϵ	Res.	Raw.	E-T	T-P	Raw.	E-T	T-P	Res.	E-T	T-P	Res.	Raw.
-	0.000	0.0	0.0	0.0	2.5e-4	0.0	0.0	2.5e-4	0.0	0.0	2.5e-4	0.0	0.0
26.1	0.004	72.5	72.1	71.5	58.0	54.7	53.7	76.7	71.0	64.6	42.9	42.2	42.4
21.0	0.008	88.3	88.7	88.3	88.7	84.2	83.5	92.1	85.5	81.8	78.5	77.3	77.4
17.8	0.012	93.8	93.8	93.9	96.6	90.8	92.2	95.8	90.0	87.0	90.7	90.0	89.5
15.6	0.016	96.2	96.3	96.3	97.8	92.7	94.7	**96.8**	92.1	89.3	96.0	95.6	94.4
13.8	0.020	**97.2**	**97.3**	**97.3**	**98.0**	**92.7**	**95.2**	96.4	**92.8**	**89.8**	**97.0**	**97.2**	**95.5**

Table 3: *SARs(%) on standalone victim anti-spoofing model (ResNet-OC). Surrogate anti-spoofing model: AASIST*

Sur. / ϵ	T-P	Res.	Raw.	E-T
0.000	0.10			
0.004	74.53	59.40	80.05	43.68
0.008	90.05	90.65	94.85	79.35
0.012	94.83	98.25	98.78	91.90
0.016	97.00	99.65	99.65	97.05
0.020	**97.98**	**99.93**	**99.85**	**98.35**

ing anti-spoofing and ASV models other than this decision-level fusion method, and we will explore them in future work.

3.2. Attack flow

• **Synthesize voice.** Previous works use source speakers' voices as bases to generate adversarial examples. The source speaker and target speaker might differ significantly in terms of timbre, which reduces transferability of adversarial examples. DoubleDeceiver uses the ZS-TTS system to first generate synthesized voices based on a short utterance from the target speaker (①), as illustrated in Figure 1. Experiments reveal that synthesized voices act as strong bases for adversarial attacks, often lying close to the decision boundary or already being identified as the target speaker, reducing the risk of local optima. Another advantage of synthesizing base voices is that it can generate voices with any length and content, which enables the attacker to deceive the victim systems that require input voices to include specific content such as commands or random numbers.

• **Generate adversarial examples.** By feeding the synthesized voice into the surrogate anti-spoofing model and ASV model, each model generates a loss given the target label (② & ③). The gradients of both losses with respect to the input are obtained by performing back-propagation. To minimize both losses, the combined gradient is calculated by adding these two gradients (④). Then the gradient-based adversarial attack method (BIM) is performed to get the adversarial perturbation. BIM [19] is defined as follows:

$$x_0^* = x \tag{1}$$

$$x_{i+1}^* = Clip_x^{\epsilon}(x_i^* - \alpha \cdot sign(\nabla_x J(x_i^*,\ y^{target}))) \tag{2}$$

x is the clean input. The subscript i represents variables in the i^{th} iteration. i satisfies $0 <= i <= T$ where $i = 0$ represents the initial value before iterations and T is the maximum number of iterations. x_i^* is the perturbed input. $J(x_t^*,\ y^{target})$ is the loss function given input x_t^* and target class y^{target} and is defined as $J = \theta - Score$ in this paper. $\nabla_{x_i^*} J$ is the gradient of the loss function with respect to input x_t^*. In equation (2), α is the step size satisfying $0 < \alpha < \epsilon$ where ϵ is the budget of adversarial perturbation. $Clip_x^{\epsilon}$ clips x_i^* to make it stay within L_∞ ϵ-neighborhood of clean input x. During the iterations, the adversarial example moves toward the direction that minimizes J so that it could be recognized as the target label. In each iteration, the adversarial perturbation is added to the synthesized voice (first iteration) or the perturbed voice from the previous iteration (⑤) and ② continues. After the maximum T iterations, the adversarial example is obtained (⑥).

• **Conduct black-box transfer attack.** The generated adversarial example is fed into the victim system (⑦). The attack is successful only if the adversarial example passes both models, i.e., it is classified as bona fide and registered speaker. Meanwhile, if the adversarial example is classified as spoofed by the victim anti-spoofing model or unknown speaker by the victim ASV model, it is rejected by the system.

4. Experiments

4.1. Experiment setting

All codes are implemented in Python utilizing the PyTorch library. The hyper-parameters are as follows: maximum iteration time $T = 30$, step size $\alpha = \epsilon/10$.

• **Datasets.** Experiments have been carried out based on VoxCeleb1 [22]. The 40 speakers (20 males, 20 females) in the

Table 4: *SARs(%) on standalone ASV models. Surrogate anti-spoofing model: AASIST*

Sur.	T-P			Res.			Raw.			E-T		
Vic. / ϵ	Res.	Raw.	E-T	T-P	Raw.	E-T	T-P	Res.	E-T	T-P	Res.	Raw.
0.000	64.9	50.8	45.1	75.1	50.8	45.1	75.1	64.9	45.1	75.1	64.9	50.8
0.004	96.7	96.4	95.7	97.8	90.1	88.4	95.9	87.6	80.3	98.7	94.3	96.6
0.008	98.2	98.6	98.3	98.0	92.6	91.9	**97.1**	90.5	86.7	**98.9**	97.0	**97.8**
0.012	98.9	99.0	99.1	**98.3**	92.5	93.9	97.0	91.2	88.2	98.7	97.9	97.6
0.016	**99.2**	**99.3**	**99.3**	98.1	**93.0**	95.0	97.1	92.5	89.6	98.9	98.5	97.3
0.020	99.2	99.3	99.3	98.1	92.8	**95.3**	96.5	**92.9**	**89.9**	98.6	**98.8**	97.1

test set of verification split are used as target speakers. A randomly chosen genuine voice from each speaker is used for synthesizing base voices and the enrollment of the surrogate and victim ASV models. We synthesize 100 utterances per speaker with randomly chosen text from LibriSpeech [23], due to the availability of transcription, to simulate the speech content and length in real-life. For each speaker, there are 100 synthesized voices, resulting in 4,000 adversarial examples for each ϵ.

• **Models.** To determine the threshold θ, we evaluate pre-trained anti-spoofing and ASV models on the evaluation set of the ASVspoof 2019 [24] logical access subset and the test set of VoxCeleb1's verification split, respectively. θ is tuned to let the models achieve equal error rate (EER) on the evaluation set. The pre-trained models used in the experiment and their respective EER(%) are as follows:

Surrogate anti-spoofing models: (1) LCNN[1]: 6.34 (2) AASIST [25]: 2.65

Victim anti-spoofing model: ResNet-OC[2] [26]: 1.56

Surrogate/Victim ASV models: (1) TDNN-PLDA[3] [27]: 2.97 (2) ResNetSE34V2[4] [28]: 1.26 (3) RawNet3 [29]: 1.00 (4) ECAPA-TDNN [30, 31]: 1.15

• **Evaluation metric.** The SAR is used to assess the performance of the attack. It is defined as $SAR = \frac{N_s}{N_t}$, where the N_s is the number of successful adversarial examples and N_t is the total number of adversarial examples. The strength of adversarial perturbation is evaluated by perturbation budget ϵ and signal-to-noise ratio (SNR). SNR is defined as $10 \cdot log_{10}(\frac{P_x}{P_\delta})$ where P_x represents the power of the clean input, i.e., synthesized voice, and P_δ is the power of perturbation. Larger SNR means smaller perturbation. The experiment is conducted on a series of ϵ ranging from 0.004 to 0.02.

4.2. Experiment results and analysis

• **SARs on the victim systems.** Tables 1–2 present the overall SARs on victim systems, where adversarial examples are generated using the same synthesized voices as bases. With no adversarial perturbations ($\epsilon = 0$), synthesized voices have very low SARs due to the presence of spoofing countermeasures. However, after adding small adversarial perturbations ($\epsilon = 0.004$), the SARs increase significantly in all model settings, averaging at 69% and reaching as high as 91.1%, despite the differences in architectures between the surrogate and victim systems.

In the majority of model settings, the SARs increase in positive correlation with ϵ, peaking at above 90% when $\epsilon = 0.020$. The highest SAR achieved among all settings is 98.3%, strongly demonstrating the effectiveness of DoubleDeceiver. In some cases, the SARs first increase to the peak at an ϵ around 0.012 and then decrease. The reason could be that BIM has a fixed step size α and it is determined by ϵ. When the ϵ and step size α get larger, BIM may miss the optimal point and overshoot.

• **SARs on standalone models.** In order to better understand the reasons behind DoubleDeceiver's performance, the SARs on standalone victim anti-spoofing and ASV models are presented in Table 3–4, respectively. Specifically, the surrogate anti-spoofing model is AASIST while the corresponding performances when the surrogate anti-spoofing model is LCNN exhibit the same pattern but are not shown due to the page limit.

Without adversarial perturbations, ResNet-OC achieves outstanding performance in detecting the synthesized voices ($\epsilon = 0$) with a SAR of only 0.1%. However, after adding a small perturbation ($\epsilon = 0.004$), there is a qualitative leap in the SARs from 0.1% to 64% on average. Notably, the SARs quickly rises to 99.85% with the increasing ϵ, demonstrating the severe threats posed by adversarial attacks to the anti-spoofing model.

As for ASV models, the synthesized voices ($\epsilon = 0$) achieve an average SAR of 58.98% which indicates the vulnerability of ASV models to speech synthesis attacks. The threats become even more significant after incorporating adversarial attacks, as evidenced by the high SARs of approximately 90% in most settings when the perturbation is small ($\epsilon = 0.004, average\ SNR \approx 26.5dB$), which is on average 59.33% relative improvement compared to the SARs without adversarial attack. This suggests that synthesized voices act as strong base voices for the adversarial attack on ASV models. Adversarial examples generated by surrogate systems that use RawNet show less transferability than the ones that use other ASV models because of the substantial difference in the input feature, where RawNet takes raw waveform as input and other models take human-designed acoustic features, but DoubleDeceiver still achieves a SAR above 80% when $\epsilon = 0.004$.

When ϵ is small, the SARs on the anti-spoofing model impedes the overall SARs, as previously shown. This is reasonable because anti-spoofing models are designed to detect synthesized voices. The synthesized voices have a very close timbre to the one from the target speaker resulting in a better performance for the adversarial examples to deceive the ASV model than the anti-spoofing model when the perturbation is small.

5. Conclusions

We propose DoubleDeceiver - a black-box attack method against advanced ASV systems protected by spoofing countermeasures, by incorporating TTS and adversarial attacks. Extensive experiments show that DoubleDeceiver achieves an overall SAR as high as 98.3% and demonstrate that advanced ASV models remain vulnerable to adversarial attacks, even when protected by spoofing countermeasures. To protect ASV systems, defense mechanisms against adversarial attacks are needed.

[1] ASVspoof 2021: https://github.com/asvspoof-challenge/2021/tree/main/LA/Baseline-LFCC-LCNN

[2] one-class softmax version

[3] baseline model

[4] H/ASP AP+Softmax

6. References

[1] D. A. Reynolds, "An overview of automatic speaker recognition technology," in *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, 2002, pp. 4072–4075.

[2] B. Saritha, M. A. Laskar, and R. H. Laskar, *A Comprehensive Review on Speaker Recognition*. Cham: Springer International Publishing, 2023, pp. 3–23.

[3] 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.

[4] J. Yamagishi, X. Wang, M. Todisco, M. Sahidullah, J. Patino, A. Nautsch, X. Liu, K. A. Lee, T. Kinnunen, N. Evans, and H. Delgado, "ASVspoof 2021: accelerating progress in spoofed and deepfake speech detection," in *Proc. 2021 Edition of the Automatic Speaker Verification and Spoofing Countermeasures Challenge*, 2021, pp. 47–54.

[5] W. Zhang, S. Zhao, L. Liu, J. Li, X. Cheng, T. F. Zheng, and X. Hu, "Attack on practical speaker verification system using universal adversarial perturbations," in *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 2575–2579.

[6] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2014.

[7] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015.

[8] X. Li, J. Zhong, X. Wu, J. Yu, X. Liu, and H. Meng, "Adversarial attacks on gmm i-vector based speaker verification systems," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[9] J. Villalba, Y. Zhang, and N. Dehak, "X-Vectors Meet Adversarial Attacks: Benchmarking Adversarial Robustness in Speaker Verification," in *Proc. Interspeech 2020*, 2020, pp. 4233–4237.

[10] F. Kreuk, Y. Adi, M. Cisse, and J. Keshet, "Fooling end-to-end speaker verification with adversarial examples," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 1962–1966.

[11] Y. Gong and C. Poellabauer, "Crafting adversarial examples for speech paralinguistics applications," in *DYnamic and Novel Advances in Machine Learning and Intelligent Cyber Security (DYNAMICS) Workshop in conjunction with ACSAC'18, San Juan, Puerto Rico*, Dec 2018.

[12] Z. Li, C. Shi, Y. Xie, J. Liu, B. Yuan, and Y. Chen, "Practical adversarial attacks against speaker recognition systems," in *Proceedings of the 21st International Workshop on Mobile Computing Systems and Applications*, ser. HotMobile '20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 9–14.

[13] G. Chen, S. Chenb, L. Fan, X. Du, Z. Zhao, F. Song, and Y. Liu, "Who is real bob? adversarial attacks on speaker recognition systems," in *2021 IEEE Symposium on Security and Privacy (SP)*, 2021, pp. 694–711.

[14] Q. Wang, P. Guo, and L. Xie, "Inaudible Adversarial Perturbations for Targeted Attack in Speaker Recognition," in *Proc. Interspeech 2020*, 2020, pp. 4228–4232.

[15] A. S. Shamsabadi, F. S. Teixeira, A. Abad, B. Raj, A. Cavallaro, and I. Trancoso, "Foolhd: Fooling speaker identification by highly imperceptible adversarial disturbances," in *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 6159–6163.

[16] S. Liu, H. Wu, H.-Y. Lee, and H. Meng, "Adversarial attacks on spoofing countermeasures of automatic speaker verification," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 312–319.

[17] Y. Zhang, Z. Jiang, J. Villalba, and N. Dehak, "Black-Box Attacks on Spoofing Countermeasures Using Transferability of Adversarial Examples," in *Proc. Interspeech 2020*, 2020, pp. 4238–4242.

[18] Y. Jia, Y. Zhang, R. J. Weiss, Q. Wang, J. Shen, F. Ren, Z. Chen, P. Nguyen, R. Pang, I. L. Moreno, and Y. Wu, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, ser. NIPS'18. Red Hook, NY, USA: Curran Associates Inc., 2018, p. 4485–4495.

[19] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*, 2017.

[20] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations*, 2018.

[21] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, June 2018, pp. 9185–9193.

[22] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: A Large-Scale Speaker Identification Dataset," in *Proc. Interspeech 2017*, 2017, pp. 2616–2620.

[23] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[24] X. Wang, J. Yamagishi, M. Todisco, H. Delgado, A. Nautsch, N. Evans, M. Sahidullah, V. Vestman, T. Kinnunen, K. A. Lee, L. Juvela, P. Alku, Y.-H. Peng, H.-T. Hwang, Y. Tsao, H.-M. Wang, S. L. Maguer, M. Becker, F. Henderson, R. Clark, Y. Zhang, Q. Wang, Y. Jia, K. Onuma, K. Mushika, T. Kaneda, Y. Jiang, L.-J. Liu, Y.-C. Wu, W.-C. Huang, T. Toda, K. Tanaka, H. Kameoka, I. Steiner, D. Matrouf, J.-F. Bonastre, A. Govender, S. Ronanki, J.-X. Zhang, and Z.-H. Ling, "Asvspoof 2019: A large-scale public database of synthesized, converted and replayed speech," 2020.

[25] 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," in *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2022, pp. 6367–6371.

[26] Y. Zhang, F. Jiang, and Z. Duan, "One-class learning towards synthetic voice spoofing detection," *IEEE Signal Processing Letters*, vol. 28, pp. 937–941, 2021.

[27] M. Kumar, T. J. Park, S. Bishop, and S. Narayanan, "Designing neural speaker embeddings with meta learning," *CoRR*, vol. abs/2007.16196, 2020. [Online]. Available: https://arxiv.org/abs/2007.16196

[28] H. S. Heo, B.-J. Lee, J. Huh, and J. S. Chung, "Clova baseline system for the voxceleb speaker recognition challenge 2020," 2020. [Online]. Available: https://arxiv.org/abs/2009.14153

[29] J. weon Jung, Y. Kim, H.-S. Heo, B.-J. Lee, Y. Kwon, and J. S. Chung, "Pushing the limits of raw waveform speaker recognition," in *Proc. Interspeech 2022*, 2022, pp. 2228–2232.

[30] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized Channel Attention, propagation and aggregation in TDNN based speaker verification," in *Interspeech 2020*, 2020, pp. 3830–3834.

[31] R. K. Das, R. Tao, and H. Li, "HLT-NUS SUBMISSION FOR 2020 NIST Conversational Telephone Speech SRE," 2021.