INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Voting for the right answer: Adversarial defense for speaker verification

Haibin Wu[1,2*], *Yang Zhang*[1*], *Zhiyong Wu*[1†], *Dong Wang*[3†], *Hung-yi Lee*[2†]

[1]Tsinghua-CUHK Joint Research Center for Media Sciences, Technologies and Systems, Shenzhen International Graduate School, Tsinghua University, China
[2]Graduate Institute of Communication Engineering, National Taiwan University
[3]Center for Speech and Language Technologies, Tsinghua University, China

Abstract

Automatic speaker verification (ASV) is a well developed technology for biometric identification, and has been ubiquitous implemented in security-critic applications, such as banking and access control. However, previous works have shown that ASV is under the radar of adversarial attacks, which are very similar to their original counterparts from human's perception, yet will manipulate the ASV render wrong prediction. Due to the very late emergence of adversarial attacks for ASV, effective countermeasures against them are limited. Given that the security of ASV is of high priority, in this work, we propose the idea of "voting for the right answer" to prevent risky decisions of ASV in blind spot areas, by employing random sampling and voting. Experimental results show that our proposed method improves the robustness against both the limited-knowledge attackers by pulling the adversarial samples out of the blind spots, and the sufficient-knowledge attackers by introducing randomness and increasing the attackers' budgets.

Index Terms: adversarial attack, speaker verification

1. Introduction

Automatic speaker verification (ASV), is to determine whether a claimed identity is speaking during a speech segment [1]. Thanks to previous efforts [2–5], ASV is now a well-developed technology for biometric identification and widely adopted in various security-critic applications. But, ASV models with high performance are vulnerable to spoofing audios [6] generated by audio replay, text-to-speech and voice conversion, back-door attacks [7], and related emerged adversarial attacks [8, 9]. In this paper, we mainly focus on tackling the adversarial attacks.

Adversarial samples were first raised by [10] and they regarded them as blind spots of models. Harnessing adversarial samples to attack well-trained models is *adversarial attack*. Adversarial samples are generated by slightly modifying the genuine samples with deliberately crafted perturbations, and [10] shows that such elaborated samples can manipulate image classification models with high performance predict wrong answers. Also, audio processing models are vulnerable to adversarial attacks. Carlini and Wagner [11] shows that adversarial attacks can hallucinate state-of-the-art automatic speech recognition (ASR) model. Given a piece of audio, either speech, silence, or music, [11] can generate another perceptually indistinguishable adversarial sample, which can deceive the ASR output text predefined by the attackers. The vulnerability of ASR to adversarial attacks has been also studied by [12–15]. Other audio processing tasks, such as anti-spoofing for ASV [16–18], voice conversion [19] and sound event classification [20], are also subject to adversarial attacks.

The performance of ASV will drop catastrophically under adversarial attacks [8, 9, 21–27]. [8] firstly shows ASV is subject to adversarial attacks. [25] attacks ASV trained by only a few speakers. [27] and [21] respectively show the vulnerability to adversarial attacks of i-vector and x-vector models. Recently, researchers also investigated more risky adversarial attacks which are inaudible [26] and universal [22, 23], and can spread over the air [22, 24]. Only limited works [28–31] are conducted to protect ASV from adversarial attacks. So it is still an open question to effectively tackle this issue. [30] trains a deep neural network based filter to eliminate adversarial noise. A binary classification model [29] is trained to spot the adversarial samples. [28] adopts adversarial training to improve the robustness of ASV. Self-supervised learning based filters are used to alleviate the adversarial noise for ASV [31, 32].

In this paper, we propose to improve the adversarial robustness by sampling the neighbors of a given utterance and letting the neighbors vote for whether the utterance should be accepted or not by the ASV model. The adversarial samples are blind spots of ASV and we believe the radius of such blind spots is small, as shown in Table 2. So if the spread of the sample space is large, then there is a large probability that most of the neighbours are outside of the blind spot. In this case, the voting will result in the right decision. Contrary to [28–30] which require the system designers to model the in-the-wild attackers and know exactly the adversarial attack methods adopted during attack in advance, our method is attack-agnostic. In contrast to [31, 32] which counter adversarial noise in the frequency domain, our method is in the time domain. Our method harnesses the idea of "voting for the right answer" to let the ASV make the decision based on an utterance's neighbors rather than the utterance itself, thus equips the ASV model with the capacity of countering adversarial attacks. The experimental results show the effectiveness of our defense method.

2. Background

2.1. Automatic Speaker verification

ASV aims to determine whether a piece of given speech belongs to an alleged speaker. Given an enrollment and test utterance, x_e and x_t, the ASV model will certify whether they are pronounced by the same speaker. Specifically, the procedure of ASV can be divided into three steps: firstly feature engineering which maps the audio waveform to acoustic features, secondly speaker embedding extraction which extracts the fixed-dimensional speaker embeddings from the variable-length acoustic feature sequences, and finally a back-end model that measures the similarity between speaker embeddings. For

[*] equal contribution
[†] corresponding authors

http://dx.doi.org/10.21437/Interspeech.2021-1452

brevity, the three steps can be denoted by a scoring function f:

$$s = f(x_t, x_e), \tag{1}$$

where, x_t and x_e are the test and enroll utterance, respectively; s measures the similarity between x_t and x_e. The smaller s is, the less likely they belong to the same person, and vice versa.

2.2. Adversarial attack

Adversarial samples are generated by slightly modifying the genuine samples with deliberately crafted perturbations. The deliberately crafted perturbations are usually indistinguishable from human's perception, yet significantly change the output of the model. Different strategies to generate the adversarial noise result in different attack algorithms, and in this paper, we adopt Basic Iterative Method (BIM) [33], an efficient iterative adversarial attack method. In BIM, attackers initialize $x_t^0 = x_t$, where x_t means the testing utterance in Eq. 1,then iteratively update it as this equation:

$$x_t^{n+1} = Clip_{\epsilon}^{x_t}\left(x_t^n + \alpha \cdot \lambda \cdot sign\left(\nabla_{x_t^n} f(x_t^n, x_e)\right)\right), \quad for\ n = 0, 1, \ldots, N-1, \tag{2}$$

where $Clip(.)$ is a clipping operation that ensures $||x_t^{n+1} - x_t||_\infty \leq \epsilon$, $\epsilon \geq 0$ and $\epsilon \in \mathbb{R}$, α is the step size, $\lambda = 1$ for the non-target trial and $\lambda = -1$ for the target trial, and N is the number of iterations. ϵ determines the attack intensity which is a predefined value by attackers. In the target trial, the testing and enroll speech are uttered by the same speaker, while in the non-target trial, they are generated by different speakers. For example, if the attacker want to attack ASV for the non-target trial, they will adopt BIM to make the similarity score between the testing and enrollment speech as high as possible in order to let the ASV model falsely accept the imposter.

2.3. System evaluation metrics

The testing trials can be represented as: $\mathbb{T} = \mathbb{T}_{tgt} \cup \mathbb{T}_{ntgt}$, where $\mathbb{T}_{tgt}$ and $\mathbb{T}_{ntgt}$ denote the target trials and non-target trials respectively. We divide the testing trials into two sets, $\mathbb{T} = \mathbb{D} \cup \mathbb{E}$, where $\mathbb{D} = \mathbb{D}_{tgt} \cup \mathbb{D}_{ntgt}$ is the development set, $\mathbb{E} = \mathbb{E}_{tgt} \cup \mathbb{E}_{ntgt}$ is the evaluation set. We determine the decision threshold τ_{ASV} by the development set as below:

$$DevFAR(\tau) = \frac{|\{s_i \geq \tau : i \in \mathbb{D}_{ntgt}\}|}{|\mathbb{D}_{ntgt}|} \tag{3}$$

$$DevFRR(\tau) = \frac{|\{s_i < \tau : i \in \mathbb{D}_{tgt}\}|}{|\mathbb{D}_{tgt}|} \tag{4}$$

$$\tau_{ASV} = \{\tau \in \mathbb{R} : DevFAR(\tau) = DevFRR(\tau)\} \tag{5}$$

where $DevFAR(\tau)$ and $DevFRR(\tau)$ mean the development false acceptance rate and false rejection rate respectively under the decision threshold τ, $\mathbb{D}_{tgt}$ and $\mathbb{D}_{ntgt}$ denote the development trial sets containing target and non-target trials, respectively, s_i is the ASV score for the trial i and $|\mathbb{A}|$ denotes the number of elements in set $\mathbb{A}$.

Then we evaluate the ASV performance on evaluation set, by FAR and FRR:

$$FAR = \frac{|\{s_i \geq \tau_{ASV} : i \in \mathbb{E}_{ntgt}\}|}{|\mathbb{E}_{ntgt}|} \tag{6}$$

$$FRR = \frac{|\{s_i < \tau_{ASV} : i \in \mathbb{E}_{tgt}\}|}{|\mathbb{E}_{tgt}|} \tag{7}$$

where τ_{ASV} is the decision threshold determined by Eq. 5.

3. Voting for the right answer

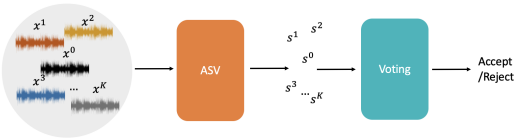

Figure 1: *Proposed framework.*

3.1. Proposed method

In this subsection, we only introduce the procedure of our proposed method, and detailed explanation and intuition of why it works will be illustrated in next subsection. For the sake of simplicity, we omit the enroll utterance x_e, and denote the testing utterance x_t as x^0. As shown in Fig. 1, a Gaussian ball with x^0 as the center, and variance ,σ, as the radius, is constructed. Then we sample K neighbors around x^0, and denote $\mathbb{X} = \{x^0, x^1, x^2 ..., x^K\}$, where $\mathbb{X}$ means the set consisting of x^0 and its K neighbors. Based on $\mathbb{X}$, $\{s^0, s^1, s^2 ..., s^K\}$, are derived as illustrated in Fig. 1. Then, the derived score for voting, s^{vote} is calculated as below:

$$s^{vote} = \frac{1}{K+1}\sum_{k=0}^{K} s^k \tag{8}$$

Based on s^{vote}, the utterance will be accepted if $s^{vote} > \tau_{ASV}$, and vice versa. Then the modified FAR and FRR for voting, FAR_{vote} and FRR_{vote}, are derived:

$$FAR_{vote} = \frac{|\{s_i^{vote} \geq \tau_{ASV} : i \in \mathbb{E}_{ntgt}\}|}{|\mathbb{E}_{ntgt}|} \tag{9}$$

$$FRR_{vote} = \frac{|\{s_i^{vote} < \tau_{ASV} : i \in \mathbb{E}_{tgt}\}|}{|\mathbb{E}_{tgt}|} \tag{10}$$

where s_i^{vote} is the derived score for voting of the i_{th} utterance.

3.2. Why "voting for the right answer" works

It is hard to guarantee the well-trained ASV model is perfect. So we can regard the ASV we derived is a slightly skewed mapping function from the data space to the label space, which makes some areas of the input space cannot be well covered by the ASV model. The uncovered input space can be regarded as blind spots of ASV. The malicious attackers deliberately try every attempt to find the blind spots and hallucinate the ASV model, and we think that adversarial samples are also blind spots of the ASV model. Data augmentation is an intuitive way to enrich the model's distribution manifolds to cover the blind spots. However, exhausting all the data augmentation strategies and making the model fill in all the manifolds of data distribution during training is extremely resource-consuming, not to mention that we can't model all the data augmentation strategies. So we provide another alternative way to handle the blind spots: voting for the right answer. In case that the testing sample falls into the blind spot, we propose to randomly sample K samples within a Gaussian ball. And then we let the neighbors of the testing sample vote for the right answer. These areas of the blind spots are small in volume as shown in Table 2, and when random sampling is conducted, the samples tend to jump out of the blind spots if the sampling variance, σ, is sufficiently large. The sampled neighbours are with more probabilities to

be in a health area rather than in a blind spot area. Then after voting, the improved probability of being in a health area leads to improved probability of getting a 'normal' decision.

3.3. Threat model and our countermeasures

We divide the threat models according to the knowledge obtained by the attackers and followed by our countermeasures.

Limited-knowledge attackers: Limited knowledge attackers have the access to the internals of the target ASV model, including parameters and gradients, but are unaware of the defense method. The attackers elaborate the adversarial samples which are blind spots of ASV. In this scenario, our defense method can choose a suitable σ and sample a large proportion of neighbours outside the blind spots, so that we have a large probability to get the right answer by voting.

Sufficient-knowledge attackers: Compared with limited-knowledge attackers, sufficient knowledge attackers even know the defense method. They do gradient descent to modify the original sample and craft an adversarial counterpart to change the s_{vote} in Eq. 8 to fool the ASV. However, we conduct voting during inference and we don't know exactly the neighbors sampled for voting in advance, neither do the attackers. And it is impractical the attackers can exhaust all the possible neighbors for attack, so the exactly gradient for s_{vote} during inference is hard to obtain. The randomness introduced by the voting procedure will hinder and obfuscate the gradient of s_{vote} and make the gradient-based attacks ineffective. Our method attains more potential because we can even determine σ and K randomly.

4. Experimental setup

4.1. Data

Voxceleb 1&2 datasets [34, 35] are used in our experiments. We train models on the development set of VoxCeleb 1&2. To test our ASV system, we use the trials provide in VoxCeleb1 test set, which contains 37,720 enrollment-testing pairs. To conduct the experiment more rigorously and without loss of generality, we randomly select 27,720 trails as development set to evaluate the performance of our ASV system on genuine samples and determine the decision threshold in Eq. 5, and the rest 10,000 trials are left as the evaluation set for generating adversarial samples and evaluating defense performance.

4.2. ASV system setup

We build our ASV systems following [36], using **Fast ResNet-34** model structures and angular margin (AM) loss. Self-Attentive Pooling (**SAP**) [37] and Attentive Statistics Pooling (**ASP**) [38] are used to aggregate frame-level features into utterance-level representations. **SAP** and **ASP** denote these two systems in the following description. In our experiments, we set the hyperparameters of AM loss $scale$ as 30 and $margin$ as 0.1. The models are trained for 50 epochs by the Adam optimizer with an initial learning rate of 0.01 decreasing by 10% every 2 epochs. During training, we use 2-second fixed length audio segments, extracted randomly from each utterance, and the batch size is 200. Spectrograms are extracted with a hamming window of width 25ms and shift 10ms, and 64-dimensional FBanks are used as the models' inputs. Our focus is countering the adversarial noise, rather than obtaining state-of-the-art ASV system. Therefore, no data augmentation and voice activity detection are conducted during training, and we use cosine distance for scoring. Our model achieved 2.52% equal error rate with ASP and 2.58% with SAP in development set. We determine and fix the models' decision thresholds τ_{ASV} by Eq. 5.

4.3. BIM attack setup

We generate adversarial samples by BIM. In our experiments, we fix the number of iterations N as 5, set the step size α as ϵ divided by N, and use different settings of ϵ: 1,5,10. The ASV systems' performance are reported in Table 1. After attack, FAR and FRR increase drastically, which shows the effectiveness of BIM. The larger ϵ is, the worse of the ASV performance, indicating more intense the attack is.

Table 1: *ASV performance for genuine and adversarial samples*

		No attack	$\epsilon = 1$	$\epsilon = 5$	$\epsilon = 10$
ASP	FAR	2.24	18.38	71.83	89.38
	FRR	2.56	20.17	74.92	91.94
SAP	FAR	2.35	18.43	72.11	89.55
	FRR	2.23	20.21	74.33	91.7

5. Experimental results

5.1. Defense for limited-knowledge attackers

In this scenario, the attackers can access the ASV model, but do not know the defense method. There is few baseline for reference and in order to make a clear comparison, we follow our previous work [31] which applies hand-crafted filters including Gaussian filter, mean filter and median filter for adversarial defense, and set them as our baseline. We use different settings of the voting variance σ, randomly sample 50 neighbors around the testing audio, and let the neighbors vote for whether the utterance should be accepted or not. The performance with **SAP** exhibits a similar trend as with **ASP**, and due to limited space, we only report the results of **ASP**. In Table 2, we evaluate the FAR and FRR of different defense methods, and note that the FAR and FRR for voting is calculated as Eq. 9 and Eq. 10. We evaluate both the negative effects of the defense methods for genuine samples (rows labeled by No attack), and the positive effects on adversarial samples in different settings of ϵ. The observations and analysis are concluded as follows: (1) Gaussian, mean and the voting method with different σ slightly affect the FAR and FRR of genuine samples, while median filter enlarges the FRR from 2.56% to 19.55%, indicating that median filter highly distorts the speech signal. (2) The three filter-based models help alleviate the adversarial attacks, but the defense performance of them is greatly outperformed by the voting-based defense. What's more, as ϵ become large, the performances of the three filter-based methods are significantly degraded, while the voting based method still decrease FAR and FRR to a great deal. (3) Note that when σ is small, the voting strategy can't alleviate the adversarial noise, as the neighbors are still with large probability in the blind spot as claimed in Section 3.2. And when σ increases, the FAR and FRR drop drastically, as a large proportion of neighbors are outside the blind spot and they will vote for the right answer.

Fig. 2 shows FAR and FRR on genuine samples (a and c) and adversarial samples (b and d) with different number of votes, K, given $\epsilon = 5$. We observe that: (1) When the number of votes increases, the FAR and FRR for adversarial samples drops drastically. And when K is around 10, the FAR and FRR for adversarial samples become saturated, which means we can use about 10 votes to achieve effective defense performance.

Table 2: *Defense performance against Limited-knowledge attackers*

		No defense	Gaussian	Mean	Median	Voting					
						$\sigma = 1$	$\sigma = 15$	$\sigma = 30$	$\sigma = 60$	$\sigma = 90$	$\sigma = 120$
FAR	No attack	2.24	1.88	2.69	0.98	2.25	2.32	2.16	2.14	1.84	1.61
	$\epsilon = 1$	18.38	11.47	12.47	1.85	16.36	27.33	2.89	2.42	2.13	**1.83**
	$\epsilon = 5$	71.83	56.86	59.66	6.3	70.83	27.55	10.62	4.63	3.05	**2.54**
	$\epsilon = 10$	89.38	80.7	82.89	12.35	89.13	69.2	29.66	9.43	5.32	**3.6**
FRR	No attack	2.56	3.04	2.05	19.55	2.27	2.82	3.32	4.95	6.06	8.12
	$\epsilon = 1$	20.17	17.38	14.77	24.24	16.59	4.9	**4.53**	5.33	6.98	8.76
	$\epsilon = 5$	74.92	62.22	65.47	40.8	72.79	33.56	16.78	**10.74**	10.81	12.18
	$\epsilon = 10$	91.94	87.58	87.13	55.42	91.1	73.74	42.52	21.77	17.05	**16.67**

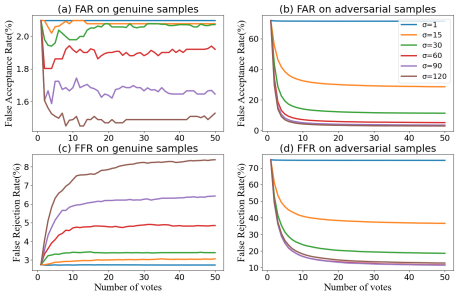

Figure 2: *Defense performance in different number of votes*

(2) Fig. 2.c shows as K increases, the FRR on genuine samples increases, which means the voting strategy results in negative effect on genuine samples. However, this is just a mild negative effect outweighed by the performance improvement on adversarial samples. We can choose a reasonable number K to balance the trade-off between the negative effect on genuine samples and positive effect on adversarial samples, which take the system security and user experience into consideration according to the system requirements.

5.2. Defense for sufficient-knowledge attackers

In this subsection, we give a case study to evaluate the robustness of our method under a more extreme scenario, where attackers have sufficient knowledge in particular the defense method. Sufficient knowledge BIM aims to attack not only the original sample, but also its K neighbors which are randomly sampled for voting. Therefore, the new BIM against voting defense requires around $K + 1$ times computing resource compared to the original BIM counterpart, however it's hard to accomplish by our computing resource when K is too large [1]. So we just set the number of votes as 5 to illustrate a case study. We fix ϵ as 5, attack iteration N as 5, and σ as 60. For filter-based defense methods, the sufficient-knowledge attackers view the ASV system equipped with hand-crafted filters as an entire system, and directly get the gradient of such system and attack it. Due to limited space, we only show the results of Gaussian filter, and the other filters are with the same trends. Table 3 reports the results. Note that the performance of limited knowledge attackers in Table 3 is worse than Table 2 because the numbers of votes are different. We observe that: (1) The voting based method achieves better defense performance than Gaussian filter. (2) Though the attackers know the voting defense and use more computing resource, they still can't do attack effectively, as the FAR and FRR only increase slightly.

[1]This indicates when attackers have sufficient knowledge, the cost of attacking the system would increase.

Table 3: *Defense results for sufficient-knowledge attackers*

	No defense	Gaussian		Voting	
		limited	sufficient	limited	sufficient
FAR(%)	71.83	56.86	60.13	10.66	13.29
FRR(%)	74.92	62.22	64.53	24.68	27.75

Furthermore, Fig. 3 shows the attack results when the attack iteration N increases, which means the attackers get more attack budgets by enlarging N. However, as N becomes larger than 10, the values of FAR and FRR become saturated. This is another evidence to show the effectiveness of the voting based defense method, as it introduces randomness and obfuscates the gradient based attack algorithm.

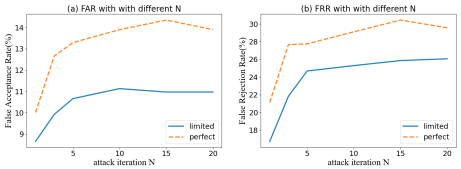

Figure 3: *Attack performance with different N*

6. Conclusion

In this paper, we propose the idea of "voting for the right answer" to prevent risky decisions of ASV in blind spot areas, by employing random sampling and voting. Experimental results show our proposed method can both counter limited-knowledge attackers, and sufficient-knowledge attackers with more attack budgets. We opened source the code [2] to make our method a comparable baseline. Future work will be dedicated to investigate other sample strategies e.g., sampling within the phone subspace, rather than the blind Gaussian to improve the defense performance of the voting strategy. We are not familiar with adaptive attacks [39] when writing this paper. We will refer to [40] and dedicate efforts to invest more powerful adaptive attacks in the future work.

7. Acknowledge

This work was done when H. Wu was a visiting student at Shenzhen International Graduate School, Tsinghua University. H. Wu is supported by Frontier Speech Technology Scholarship of National Taiwan University. Y. Zhang is supported by National Natural Science Foundation of China (NSFC) (62076144) and joint research fund of NSFC-RGC (Research Grant Council of Hong Kong) (61531166002, N_CUHK404/15).

[2]https://github.com/thuhcsi/adsv_voting

8. References

[1] T. F. Zheng and L. Li, *Robustness-related issues in speaker recognition*. Springer, 2017.

[2] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1435–1447, 2007.

[3] 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.

[4] L. Li, Y. Chen, Y. Shi, Z. Tang, and D. Wang, "Deep speaker feature learning for text-independent speaker verification," *arXiv preprint arXiv:1705.03670*, 2017.

[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] 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.

[7] T. Zhai, Y. Li, Z. Zhang, B. Wu, Y. Jiang, and S.-T. Xia, "Backdoor attack against speaker verification," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 2560–2564.

[8] 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)*. IEEE, 2018, pp. 1962–1966.

[9] R. K. Das, X. Tian, T. Kinnunen, and H. Li, "The attacker's perspective on automatic speaker verification: An overview," *arXiv preprint arXiv:2004.08849*, 2020.

[10] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[11] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2018, pp. 1–7.

[12] X. Yuan, Y. Chen, Y. Zhao, Y. Long, X. Liu, K. Chen, S. Zhang, H. Huang, X. Wang, and C. A. Gunter, "Commandersong: A systematic approach for practical adversarial voice recognition," in *27th {USENIX} Security Symposium ({USENIX} Security 18)*, 2018, pp. 49–64.

[13] L. Schönherr, K. Kohls, S. Zeiler, T. Holz, and D. Kolossa, "Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding," *arXiv preprint arXiv:1808.05665*, 2018.

[14] H. Yakura and J. Sakuma, "Robust audio adversarial example for a physical attack," *arXiv preprint arXiv:1810.11793*, 2018.

[15] Y. Qin, N. Carlini, G. Cottrell, I. Goodfellow, and C. Raffel, "Imperceptible, robust, and targeted adversarial examples for automatic speech recognition," in *International Conference on Machine Learning*. PMLR, 2019, pp. 5231–5240.

[16] S. Liu, H. Wu, H.-y. Lee, and H. Meng, "Adversarial attacks on spoofing countermeasures of automatic speaker verification," *arXiv preprint arXiv:1910.08716*, 2019.

[17] H. Wu, A. T. Liu, and H.-y. Lee, "Defense for black-box attacks on anti-spoofing models by self-supervised learning," *arXiv preprint arXiv:2006.03214*, 2020.

[18] H. Wu, S. Liu, H. Meng, and H.-y. Lee, "Defense against adversarial attacks on spoofing countermeasures of asv," *arXiv preprint arXiv:2003.03065*, 2020.

[19] C.-y. Huang, Y. Y. Lin, H.-y. Lee, and L.-s. Lee, "Defending your voice: Adversarial attack on voice conversion," in *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2021, pp. 552–559.

[20] V. Subramanian, E. Benetos, and M. B. Sandler, "Robustness of adversarial attacks in sound event classification," 2019.

[21] X. Li, J. Zhong, X. Wu, J. Yu, X. Liu, and H. Meng, "Adversarial attacks on gmm i-vector based speaker verification systems," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6579–6583.

[22] Y. Xie, C. Shi, Z. Li, J. Liu, Y. Chen, and B. Yuan, "Real-time, universal, and robust adversarial attacks against speaker recognition systems," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 1738–1742.

[23] M. Marras, P. Korus, N. D. Memon, and G. Fenu, "Adversarial optimization for dictionary attacks on speaker verification." in *Interspeech*, 2019, pp. 2913–2917.

[24] 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*, 2020, pp. 9–14.

[25] G. Chen, S. Chen, L. Fan, X. Du, Z. Zhao, F. Song, and Y. Liu, "Who is real bob? adversarial attacks on speaker recognition systems," *arXiv preprint arXiv:1911.01840*, 2019.

[26] Q. Wang, P. Guo, and L. Xie, "Inaudible adversarial perturbations for targeted attack in speaker recognition," *arXiv preprint arXiv:2005.10637*, 2020.

[27] J. Villalba, Y. Zhang, and N. Dehak, "x-vectors meet adversarial attacks: Benchmarking adversarial robustness in speaker verification," *Proc. Interspeech 2020*, pp. 4233–4237, 2020.

[28] Q. Wang, P. Guo, S. Sun, L. Xie, and J. H. Hansen, "Adversarial regularization for end-to-end robust speaker verification." in *Interspeech*, 2019, pp. 4010–4014.

[29] X. Li, N. Li, J. Zhong, X. Wu, X. Liu, D. Su, D. Yu, and H. Meng, "Investigating robustness of adversarial samples detection for automatic speaker verification," *arXiv preprint arXiv:2006.06186*, 2020.

[30] H. Zhang, L. Wang, Y. Zhang, M. Liu, K. A. Lee, and J. Wei, "Adversarial separation network for speaker recognition," *Proc. Interspeech 2020*, pp. 951–955, 2020.

[31] H. Wu, X. Li, A. T. Liu, Z. Wu, H. Meng, and H.-y. Lee, "Adversarial defense for automatic speaker verification by cascaded self-supervised learning models," *arXiv preprint arXiv:2102.07047*, 2021.

[32] H. Wu, X. Li, Z. Wu, H. Meng, and H.-y. Lee, "Improving the adversarial robustness for speaker verification by self-supervised learning," *arXiv preprint arXiv:2106.00273*, 2021.

[33] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," *arXiv preprint arXiv:1611.01236*, 2016.

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

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

[36] J. S. Chung, J. Huh, S. Mun, M. Lee, H. S. Heo, S. Choe, C. Ham, S. Jung, B.-J. Lee, and I. Han, "In defence of metric learning for speaker recognition," in *Interspeech*, 2020.

[37] 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.

[38] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," *arXiv preprint arXiv:1803.10963*, 2018.

[39] F. Tramer, N. Carlini, W. Brendel, and A. Madry, "On adaptive attacks to adversarial example defenses," *arXiv preprint arXiv:2002.08347*, 2020.

[40] C.-H. Yang, J. Qi, P.-Y. Chen, X. Ma, and C.-H. Lee, "Characterizing speech adversarial examples using self-attention u-net enhancement," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 3107–3111.