INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

x-Vectors Meet Adversarial Attacks: Benchmarking Adversarial Robustness in Speaker Verification

Jesús Villalba[1,2], *Yuekai Zhang*[1], *Najim Dehak*[1,2]

[1]Center for Language and Speech Processing, Johns Hopkins University, Baltimore, USA
[2]Human Language Technology Center of Excellence, Johns Hopkins University, Baltimore, USA

{jvilla17,yzhan400,ndehak3}@jhu.edu

Abstract

Automatic Speaker Verification (ASV) enables high-security applications like user authentication or criminal investigation. However, ASV can be subjected to malicious attacks, which could compromise that security. The ASV literature mainly studies spoofing (a.k.a impersonation) attacks such as voice replay, synthesis or conversion. Meanwhile, other kinds of attacks, known as adversarial attacks, have become a threat to all kind of machine learning systems. Adversarial attacks introduce an imperceptible perturbation in the input signal that radically changes the behavior of the system. These attacks have been intensively studied in the image domain but less in the speech domain.

In this work, we investigate the vulnerability of state-of-the-art ASV systems to adversarial attacks. We consider a threat model consisting in adding a perturbation noise to the test waveform to alter the ASV decision. We also discuss the methodology and metrics to benchmark adversarial attacks and defenses in ASV. We evaluated three x-vector architectures, which performed among the best in recent ASV evaluations, against fast gradient sign and Carlini-Wagner attacks. All networks were highly vulnerable in the white-box attack scenario, even for high SNR (30-60 dB). Furthermore, we successfully transferred attacks generated with smaller white-box networks to attack a larger black-box network.

Index Terms: speaker verification, x-vectors, adversarial

1. Introduction

The x-vector paradigm [1], is the current state-of-the-art (SOTA) for automatic speaker verification (ASV) used in recent evaluations [2, 3, 4, 5]. ASV is applied for person authentication, forensics, criminal surveillance, etc. These applications require very secure systems. Until recently, the main threats to ASV were spoofing (a.k.a impersonation) attacks [6]. ASVSpoof challenges [7] have fostered research to investigate spoofing countermeasures [8].

In recent times, a new type of attacks, named *adversarial*, has attracted the attention of the whole machine learning (ML) community. Adversarial attacks add a perturbation to the input signal, which is imperceptible to humans, but that changes the behavior of the ML system [9]. These attacks have been intensively studied in computer vision [10, 11, 12, 13]. For automatic speech recognition (ASR), most works attack end-to-end systems [14, 15, 16, 17]. Less works approach Hybrid ASR [18]. More recent works improve the attacks by psychoacoustics [19, 20], or by making the audio attacks effective in the physical world [21]

For speaker recognition, there are fewer studies. Early works attack classification tasks with few speakers [22, 23]. However, they do not use SOTA systems. A very recent work [24] attacks a public pre-trained Kaldi TDNN x-Vector model [1], also in a small classification task. Another recent work attacks an i-vector model [25]: also the generated examples are used to attack the black-box Kaldi TDNN model in VoxCeleb1 [26] ASV task. Related to ASV, there are a few works that attack spoofing detection systems [27, 28].

In this paper, we evaluated the robustness of SOTA x-vector architectures, i.e., ResNet and E-TDNN, to adversarial attacks. We considered a threat model where we add a perturbation noise to the test waveform to create impersonation or evasion attacks in the VoxCeleb1 ASV task. We adapted frequent attacks in the literature to the verification task (most previous works deal with classification). We considered a white-box and a black-box scenario, where adversarial examples were created using a weaker x-vector network and used to attack stronger x-vectors. We also widely discuss the proper metrics to evaluate these attacks in an ASV task. We concluded that we need to use calibration sensitive metrics, e.g., actual DCF, to avoid sub-estimating the attack damage. We propose representing the ASV metric against a perturbation budget related to auditory perception, e.g., actual DCF vs SNR.

2. x-Vector speaker verification

The x-vector approach uses a neural network to encode the identity information in each speech utterance into a single embedding vector [1]. The x-vector network consists of three parts. First, an encoder network extracts frame-level representations from acoustic features (MFCC, filter-banks). This is followed by a global temporal pooling layer that produces a single vector per utterance. Finally, a feed-forward network computes speaker class posteriors. The network is trained on a large set of speakers, different from those that appear in the evaluation, using some form of cross-entropy loss. We employed additive angular margin softmax (AAM-softmax) [29] in this work. In the evaluation phase, the x-vector embedding is obtained from the first affine transform after pooling, while the last layers of the network are discarded. Different x-vector systems are characterized by different encoder architectures and pooling methods. In this work, we used a ResNet34 encoder similar to the one in [5] with 64 to 512 channels in the residual blocks. We also used ThinResNet34 with 16 to 128 channels, and a residual version of Extended TDNN [30, 3], with 5 E-TDNN blocks with 512 dimension. We used mean plus standard deviation pooling for all networks. Given an enrollment and a test utterance, we just need to compare its corresponding x-vectors to decide if they belong to the same or different speakers. For this, we can use a cosine scoring or PLDA back-end [31].

This project was supported by DARPA Award HR001119S0026-GARD-FP-052

http://dx.doi.org/10.21437/Interspeech.2020-2458

3. Threat model

We assumed the following threat model for the speaker verification (ASV) task. The enrollment phase is not subjected to attacks, so we operate as usual. We acquire one or several utterances from each target speaker and compute the corresponding x-vectors, which are stored in a database. In the test phase, we craft an adversarial example by adding a small perturbation noise to the original test waveform and compute the corresponding x-vector. Finally, the back-end compares enrollment and test x-vectors to decide whether they correspond to the same (target trial) or different speakers (non-target trial). The adversarial perturbation is optimized to alter the system's decision while remaining imperceptible for human listeners.

We separately evaluated attacks to non-target trials–to be classified as targets–, and to target trials–to be classified as non-targets. We will refer to these attacks as *adversarial impersonation* and *adversarial evasion*, respectively. In this manner, we intend to find out which type of trials are the most vulnerable.

First, we considered a white-box scenario, where we assume that the attacker has full knowledge of the system, including architecture and parameters. In this case, the attacker can back-propagate the gradient of the loss function–e.g., binary cross-entropy between the system output and the adversarial label–through the calibrator, back-end, x-vector network and feature extractor up to the input waveform. Thus, the perturbation can be optimized by gradient descent methods. We considered two frequent attacks in the literature (fast gradient sign [10] and Carlini-Wagner [11]) and adapted them to ASV.

Furthermore, we considered transfer-based black-box attacks. In this scenario, we assume that the attacker does not have access to the x-vector network under attack, but he can build his own x-vector system with a different architecture and use it to generate adversarial examples. Then, those examples are utilized as input to the black-box model.

4. Adversarial attacks

4.1. Fast gradient sign methods

4.1.1. FGSM

The basic fast gradient sign method (FGSM) [10] computes an adversarial example $\mathbf{x}'$ given a bona-fide audio $\mathbf{x}$ as

$$\mathbf{x}' = \mathbf{x} - \varepsilon\,\mathrm{sign}(\nabla_{\mathbf{x}} L(g(\mathbf{x}), t)) \ , \tag{1}$$

where, for ASV, L is binary cross-entropy loss, t is the adversarial label of the trial, and $g(\mathbf{x})$ is the target posterior

$$g(\mathbf{x}) = \mathrm{sigmoid}(h(\mathbf{x}) + \mathrm{logit} P_{\mathcal{T}}) \ , \tag{2}$$

where $h(\mathbf{x})$ is the log-likelihood ratio (LLR) from the ASV system and $P_{\mathcal{T}}$ is the target prior. Note that $h(\mathbf{x})$ also depends on the enrollment x-vector, we omit the dependency to keep the notation uncluttered. From Eq. (2), also note that $h(\mathbf{x})$ needs to produce a well-calibrated LLR to properly compute the posterior, and that the adversarial example will depend on the operating point defined by $P_{\mathcal{T}}$. For this reason, we think that score calibration is important in the context of adversarial attacks.

The coefficient ε is equal to the L_∞ norm of the perturbation that we want to generate. ε is chosen small enough to be undetectable. It is important to point out that this method was designed to be fast but not to produce optimal/minimal adversarial perturbations.

4.1.2. Randomized FGSM

Randomized FGSM [32] applies a small random perturbation to the signal before applying FGSM,

$$\tilde{\mathbf{x}} = \mathbf{x} + \alpha\,\mathrm{sign}(\mathcal{N}(\mathbf{0}, \mathbf{I})) \tag{3}$$

$$\mathbf{x}' = \tilde{\mathbf{x}} - (\varepsilon - \alpha)\,\mathrm{sign}(\nabla_{\tilde{\mathbf{x}}} L(g(\tilde{\mathbf{x}}), t)) \ , \tag{4}$$

with $\alpha < \varepsilon$ ($\alpha = \varepsilon/5$ in our experiments). This simple method increases the attack robustness against models that have been adversarially trained.

4.1.3. Iterative FGSM

Iterative FGSM [12] instead of taking a single step ε in the direction of the gradient, it takes iterative smaller steps α ($\alpha = \varepsilon/5$ in our experiments),

$$\mathbf{x}'_{i+1} = \mathbf{x} + \mathrm{clip}_\varepsilon(\mathbf{x}'_i - \alpha\,\mathrm{sign}(\nabla_{\mathbf{x}'_i} L(g(\mathbf{x}'_i), t)) - \mathbf{x}) \ , \tag{5}$$

where $\mathbf{x}'_0 = \mathbf{x}$, and the clip function makes sure that the L_∞ norm of perturbation w.r.t. the original signal is smaller than ε after each optimization step i. This attack has better performance than FGSM, at the cost of higher computation.

4.2. Carlini-Wagner

The Carlini-Wagner (CW) attacks [11] write the adversarial signal as $\mathbf{x}' = \mathbf{x} + \boldsymbol{\delta}$, and searches for the minimal perturbation $\boldsymbol{\delta}$ that makes the classifier to fail. $\boldsymbol{\delta}$ is obtained by minimizing,

$$C(\boldsymbol{\delta}) = D(\mathbf{x}, \mathbf{x} + \boldsymbol{\delta}) + c\,f(\mathbf{x} + \boldsymbol{\delta}) \ . \tag{6}$$

There are three elements in the above equation. D is a distance metric. By minimizing D, we minimize the amplitude of the perturbation. For images, D is usually the perturbation L_0, L_2 or L_∞ norm. For audio, L_2 norm may not be a good choice since it depends on the duration of the signal. Longer signals may need larger L_2 perturbations. Thus the CW optimizer configuration may depend on the signal duration. We propose two alternative metrics. First, the L_2 norm normalized by the square root of the number of samples n, $D(\mathbf{x}, \mathbf{x}+\boldsymbol{\delta}) = \|\boldsymbol{\delta}\|_2 / n^{1/2}$–usually known as root mean square value (RMS). Second, the negative signal-to-noise ratio (SNR) between the original signal and the perturbation noise (in dB), $D(\mathbf{x}, \mathbf{x} + \boldsymbol{\delta}) = -\mathrm{SNR}(\mathbf{x}, \boldsymbol{\delta})$. We compared three distance metrics L_2, RMS and SNR.

The function f is defined in such a way that the system fails if and only if $f(\mathbf{x}+\boldsymbol{\delta}) \leq 0$. In [11], f is defined for a closed-set classification problem. Here, we define f for verification as,

$$f(\mathbf{x}') = \begin{cases} \max(0, h(\mathbf{x}') - (\theta - \kappa)) & \text{if } t_{\mathrm{true}} = \text{target} \\ \max(0, -h(\mathbf{x}') + (\theta + \kappa)) & \text{if } t_{\mathrm{true}} = \text{nontarget} \end{cases} , \tag{7}$$

where $h(\mathbf{x}')$ is the log-likelihood ratio score, θ is the decision threshold and κ is a confidence value. Eq. (7), means that, for adversarial evasion, we need to make the score smaller than $(\theta - \kappa)$; and for adversarial impersonation, we need to make the score larger than $(\theta + \kappa)$. Again, we note the importance of having a well calibrated system to set the threshold θ. If we are not sure about the operating point of the system under attack, we can set $\kappa > 0$ to increase the confidence that the attack will be successful.

The weight c balances D and f objectives. For each trial, we used a binary search procedure to find the optimal c. For each value of c, we optimize $C(\boldsymbol{\delta})$. If we find a solution where $f(\mathbf{x}+\boldsymbol{\delta}) > 0$ (failed attack), we enlarge c to increase the weight of the f objective over D, and repeat the optimization. If the attack is successful, we reduce c.

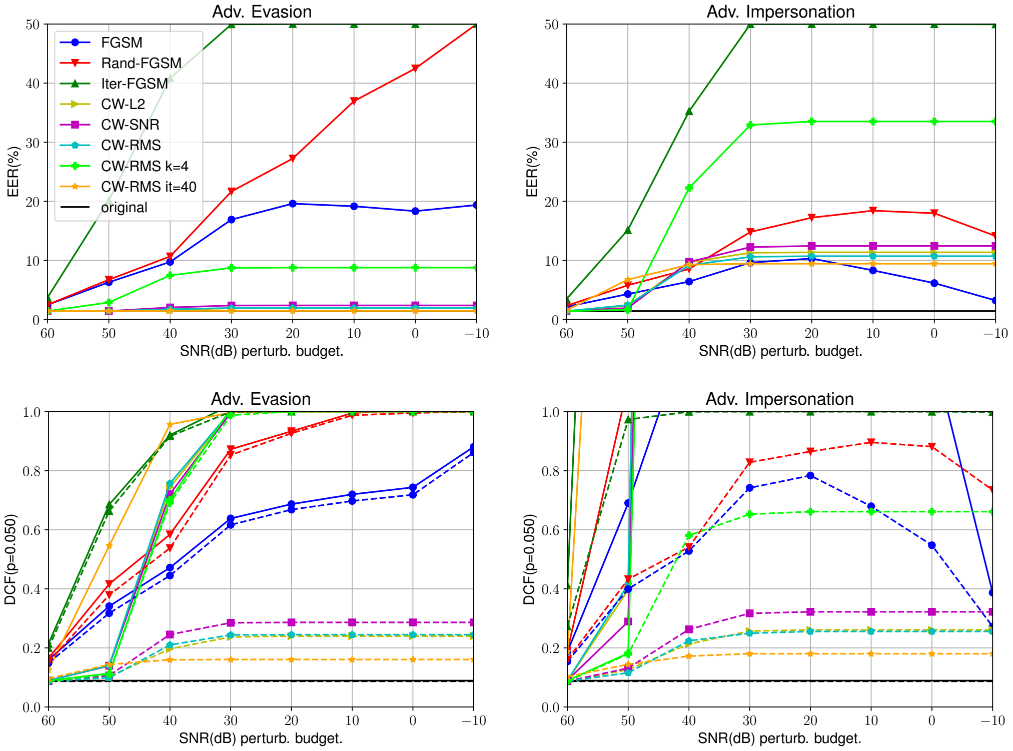

Figure 1: *White-box FGSM and Carlini-Wagner attacks on ResNet34 x-vector. Upper figures show EER(%) vs SNR perturbation budget. Lower figures show minimum DCF (dashed lines) and actual DCF (solid lines). For CW attacks, the confidence $\kappa = 0$ and num. iters in the inner loop is 10, unless indicated otherwise in the legend. Black-line indicates results without attack.*

5. Benchmarking adversarial robustness

Here, we discuss how to evaluate ASV robustness to adversarial attacks. Existing works compare accuracy [24] or EER [25] w.r.t. the FGSM ϵ parameter. However, ϵ may not be the best metric to evaluate the perturbation magnitude. Metrics such as SNR or PESQ, more common in audio applications, maybe better. Furthermore, there are attacks such as Carlini-Wagner (CW) [11], which do not have a parameter to control the magnitude of the perturbation. CW tries to find the smallest perturbation that fools the system, but if the system is very robust to attacks CW may produce a large perturbation.

The authors in [13], propose to represent accuracy against a given perturbation budget for classification tasks. Following that idea, we propose to represent typical ASV metrics (EER, MinDCF, ActDCF), against the perturbation budget. A perturbation budget b indicates that we do not accept perturbations larger than b–because we want to keep the perturbation undetectable. We can consider several metrics to measure the perturbation. In this work, we use the SNR between the original signal and the perturbation noise. However, this methodology generalizes for other metrics such us PESQ, L_∞, etc. For the examples that we generated, the perturbation starts being slightly audible for SNR$<$ 40 dB. Most humans would not notice it, or they would attribute to standard channel noise. For SNR$<$ 20 dB, the noise was clearly audible, but it did not contain any distinctive characteristic, which a human could identify as an attack.

The method to calculate ASV performance against the SNR budget is as follows. We assume an evaluator function $E(\mathbf{s},\mathbf{t})$ that computes ASV metrics given the scores $\mathbf{s}$ and labels $\mathbf{t}$ vectors of an ASV task with N trials. Let us assume that $\mathbf{s}$ are bonafide scores and $\mathbf{s}'$ are the corresponding adversarial scores. Let us assume a vector $\mathbf{p}$ containing the SNRs of the adversarial trials. Then, for each value of perturbation budget b that we want to evaluate, we build a score vector $\tilde{\mathbf{s}}(b)$ with elements

$$\tilde{s}_i(b) = \begin{cases} s'_i & \text{if } p_i \leq b \wedge i \in K \\ s_i & \text{otherwise} \end{cases} \quad i = 1,\dots,N; \ . \tag{8}$$

When evaluating *impersonation*, K is the set of non-target trials; and for *evasion* K is the set of target trials. Finally, ASV metrics for budget b are obtained by evaluating $E(\tilde{\mathbf{s}}(b),\mathbf{t})$.

For FGSM style attacks, we may want to pool together the scores from several ε values in a single curve. In such a case, we modify (8) to choose the adversarial score from the ε corresponding to the lowest SNR$>$ b.

6. Experiments

6.1. Experimental Setup

We experimented using VoxCeleb 1 and 2 datasets [26]. The acoustic features employed were 80 dimension log-Mel filterbanks with short-time mean normalization. We experimented with three SOTA x-vector architectures: ResNet34, ThinResNet34 and Residual E-TDNN. The networks were trained on VoxCeleb2 dev+eval augmented 6× with noise from the MUSAN corpus and impulse responses from the RIR dataset. We used cosine scoring as back-end since, for this task, it performed better than PLDA. We evaluated on VoxCeleb1 *Original-Clean* trial list (37k trials, 40 speakers). Note that, these experiments have high computing cost (20 GTX 1080 GPUs were used), so it was not feasible to evaluate on the larger Entire and Hard lists. Scores were calibrated by linear logistic regression on the bonafide trials. The full pipeline was implemented in PyTorch [33]

`http://www.openslr.org/resources/17`
`http://www.openslr.org/resources/28`

Table 1: *SV. systems comparison without attack.*

System	EER(%)	Min/Act DCF(0.05)	Min/Act DCF(0.01)
ThinResNet34 [26]	2.87	-	0.31/-
ft-CBAM [34]	2.03	-	-
BLSTM-ResNet [35]	1.87	-	-
ResNet34 [5]	**1.22**	-	**0.157/-**
ResETDNN	3.06	0.202/0.205	0.301/0.332
ThinResNet34	2.24	0.152/0.156	0.224/0.236
ResNet34	**1.42**	**0.087/0.089**	**0.130/0.170**

so we can back-propagate gradients from the final score to the waveform.

Table 1 shows the performance of our systems for bona-fide trials compared with the best reported in the literature [5, 35, 34]. The table shows that our systems are comparable to the current state-of-the-art. For DCF, we used the SRE19 AV [2] operating point $P_{\mathcal{T}} = 0.05$, since the number of errors in lower operating points were too small to compute reliable metrics.

6.2. White-box attack results

Figure 1 shows results for white-box attacks on ResNet34 x-vectors in terms of EER and DCF at $P_{\mathcal{T}} = 0.05$. Minimum DCF is represented by dashed lines and actual DCF by solid lines. Regarding FGSM attacks, Rand-FGSM (red) outperformed FGSM (blue) in all metrics, with the same computing cost. Iterative-FGSM (green) performed the best, but having 5× higher cost than simpler FGSM variants. Iter-FGSM achieved actual DCF$>$ 1 for perturbation budgets as high as 30 and 50 dB in evasion and impersonation attacks, respectively. Impersonation damaged DCF more than evasion for high SNR budgets. This is explained because impersonation increments false alarms and, for our operating point, the false alarm rate has a higher weight than the miss rate in the DCF formula. Thus, small increments in the number of false alarms can lead to large increments in DCF.

Regarding Carlini-Wagner (CW) attacks, first, we compare different perturbation metrics (L_2, SNR and RMS) setting the confidence $\kappa = 0$ in (7). As mentioned in Section 4.2, CW attack consists of two nested loops, the outer loop optimizes the weight c in (6), while the inner loop optimizes the perturbation $\boldsymbol{\delta}$. We set the iterations for outer and inner loops to 9 and 10, respectively. Thus, CW queried the model 18× more than Iter-FGSM. The L_2, SNR and RMS versions performed similarly, being CW-RMS slightly better in Act. DCF. CW attacks performed well in terms of Act. DCF but not in terms of EER and min. DCF, compared to FGSM attacks. This is because CW is very dependent on the operating point (decision threshold) of the system. It searches for the minimum perturbation that makes the score to cross the threshold, but it does not care about other operating points. As a consequence, the CW had a smaller impact on EER than FGSM (almost no impact in evasion attack). Similarly, CW min. DCFs were smaller than FGSM variants, which exhibit min. DCF close to act. DCFs. Note that to calculate min. DCF, we find a new decision threshold that minimizes the cost. Using min. DCF as metric is like assuming that we know that the system is under attack, and we adapt the decision threshold (change op. point) to the attack. This situation is not realistic, from our point of view. If we increase the CW inner loop iterations to 40 (orange), we increase act. DCF–getting close to the Iter-FGSM result– but EER and min. DCF reduce even further. Thus, the more optimal the perturbation for a given threshold, the worse may be for other operating points. To make CW generalize to other operating points, a solution is increasing κ in (7). By setting $\kappa = 4$ (greenlime), we significantly increased EER and min. DCF.

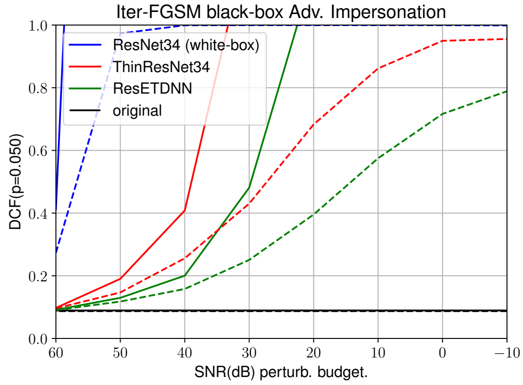

Figure 2: *Transfer-based impersonation attack on black-box ResNet34 using white-box ThinResNet34 and ResTDNN (MinDCF: dashed; ActDCF: solid; No-attack: black).*

From the above discussion, we conclude that metrics for adversarial attacks to ASV need to be calibration (threshold) sensitive. Metrics like EER or min. DCF may sub-estimate the impact of the attack on our system. Actual DCF is our preferred metric since the ASV community widely uses it. However, other metrics such as miss rate and false alarm rate at a given threshold could also be used.

ThinResNet34 and Res-ETDNN x-vector were as vulnerable as ResNet34 to white-box attacks. We do not include the results due to the limited space available.

6.3. Transfer-based black-box attack results

We also considered attacking black-box x-vectors, for which we do not know its architecture or parameters. However, we assumed that the attacker could obtain a white-box system and use it to generate adversarial examples. We attacked black-box ResNet34 generating examples with white-box ThinResNet34 and ResETDNN x-vectors using Iter-FGSM–we observed that Iter-FGSM examples had more transferability than CW. Figure 2 shows the results for adversarial impersonation. Though the black-box attacks (red, green) performance was not as good as the white-box (blue), they were highly successful for high SNR budgets. ThinResNet34 achieved DCF=0.5 with a budget of 40 dB, and ResETDNN with 30 dB. Note that for these attacks, we still assumed white-box feature extractor and enrollment utterance. We will analyze fully black-box setups in future work.

7. Conclusions

In this work, we studied the problem of adversarial attacks to the current state-of-the-art speaker verification (SV) systems. We proved that x-vector systems are highly vulnerable to white-box and transfer-based black-box attacks, even using simple attack methods such as fast gradient sign. We proposed a methodology for benchmarking ASV robustness based on plotting ASV metrics against a perturbation budget. We argue that the ASV metric should be calibration (threshold) sensitive, e.g., actual DCF. We justified this claim based on the fact that some attacks, e.g., Carlini-Wagner, may focus on fooling the system at a given operating point, not affecting others. Thus, EER and min. DCF are over-optimistic metrics. The perturbation budget metric should be related to human perception. We used SNR, but in future work we will explore metrics more related to psychoacoustics.

8. References

[1] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-Vectors : Robust DNN Embeddings for Speaker Recognition," in *ICASSP 2018*, Alberta, Canada, apr 2018, pp. 5329–5333.

[2] S. O. Sadjadi, C. Greenberg, E. Singer, D. A. Reynolds, L. Mason, and J. Hernandez-cordero, "The 2019 NIST Speaker Recognition Evaluation CTS Challenge," in *Odyssey 2020*, Tokyo, Japan, 2020.

[3] J. Villalba, N. Chen, D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, J. Borgstrom, L. P. García-Perera, F. Richardson, R. Dehak, P. A. Torres-Carrasquillo, and N. Dehak, "State-of-the-art speaker recognition with neural network embeddings in NIST SRE18 and Speakers in the Wild evaluations," *Computer Speech & Language*, vol. 60, p. 101026, mar 2020.

[4] J. Villalba, D. Garcia-Romero, N. Chen, G. Sell, J. Borgstrom, A. McCree, L. P. Garcia-Perera, S. Kataria, P. S. Nidadavolu, P. A. Torres-Carrasquillo, and N. Dehak, "Advances in Speaker Recognition for Telephone and Audio-Visual Data : the JHU-MIT Submission for NIST SRE19," in *Odyssey 2020*, Tokyo, Japan, 2020.

[5] H. Zeinali, S. Wang, A. Silnova, P. Matějka, and O. Plchot, "BUT System Description to VoxCeleb Speaker Recognition Challenge 2019," in *The VoxSRC Workhsop 2019*, oct 2019.

[6] 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.

[7] 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," in *INTERSPEECH 2019*, Graz, Austria, sep 2019, pp. 1008–1012.

[8] C.-I. Lai, N. Chen, J. Villalba, and N. Dehak, "ASSERT: Anti-Spoofing with Squeeze-Excitation and Residual neTworks," in *INTERSPEECH 2019*, Graz, Austria, sep 2019.

[9] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *ICLR 2014*, 2014.

[10] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," in *ICLR 2015*, dec 2015.

[11] N. Carlini and D. Wagner, "Towards Evaluating the Robustness of Neural Networks," in *IEEE Symposium on Security and Privacy, 2017*, aug 2016.

[12] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *CoRR 2017*, jul 2017.

[13] Y. Dong, Q.-A. Fu, X. Yang, T. Pang, H. Su, Z. Xiao, and J. Zhu, "Benchmarking Adversarial Robustness," dec 2019.

[14] M. Cisse, Y. Adi, N. Neverova, and J. Keshet, "Houdini: Fooling Deep Structured Prediction Models," in *NIPS 2017*, jul 2017, pp. 6977—-6987.

[15] D. Iter, J. Huang, and M. Jermann, "Generating adversarial examples for speech recognition," Tech. Rep., 2017.

[16] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *SPW 2018*, 2018.

[17] P. Neekhara, S. Hussain, P. Pandey, S. Dubnov, J. McAuley, and F. Koushanfar, "Universal Adversarial Perturbations for Speech Recognition Systems," in *INTERSPEECH 2019*, Graz, Austria, sep 2019, pp. 481–485.

[18] 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 *USENIX Security 2018*, jan 2018.

[19] L. Schonherr, K. Kohls, S. Zeiler, T. Holz, and D. Kolossa, "Adversarial Attacks Against Automatic Speech Recognition Systems via Psychoacoustic Hiding," in *NDSS 2019*, Reston, VA, 2019.

[20] Y. Qin, N. Carlini, I. Goodfellow, G. Cottrell, and C. Raffel, "Imperceptible, Robust, and targeted adversarial examples for automatic speech recognition," in *ICML 2019*, 2019, pp. 9141–9150.

[21] H. Yakura and J. Sakuma, "Robust Audio Adversarial Example for a Physical Attack," in *IJCAI 2019*. California: IJCAI 2019, aug 2019, pp. 5334–5341.

[22] F. Kreuk, Y. Adi, M. Cisse, and J. Keshet, "Fooling End-To-End Speaker Verification With Adversarial Examples," in *ICASSP 2018*, apr 2018, pp. 1962–1966.

[23] 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*, San Juan, Puerto Rico, dec 2018.

[24] 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*, may 2020, pp. 1738–1742.

[25] 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*, Barcelona, Spain, may 2020, pp. 6579–6583.

[26] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Computer Speech and Language*, vol. 60, 2020.

[27] 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 - Proceedings*, Singapore, dec 2019, pp. 312–319.

[28] H. Wu, S. Liu, H. Meng, and H.-y. Lee, "Defense Against Adversarial Attacks on Spoofing Countermeasures of ASV," in *ICASSP 2020*, no. 14208718, Barcelona, Spain, may 2020, pp. 6564–6568.

[29] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," in *CVPR 2019*, 2019.

[30] J. Villalba, N. Chen, D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, J. Borgstrom, F. Richardson, S. Shon, F. Grondin, R. Dehak, L. P. Garcia-Perera, D. Povey, P. A. Torres-Carrasquillo, S. Khudanpur, and N. Dehak, "State-of-the-art Speaker Recognition for Telephone and Video Speech: the JHU-MIT Submission for NIST SRE18," in *INTERSPEECH 2019*, Graz, Austria, sep 2019.

[31] P. Kenny, "Bayesian Speaker Verification with Heavy-Tailed Priors," in *Odyssey 2010*, Brno, Czech Republic, jul 2010.

[32] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble Adversarial Training: Attacks and Defenses," in *ICLR 2018*, may 2017.

[33] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *NeurIPS 2019*. Curran Associates, Inc., 2019, pp. 8024–8035.

[34] S. Yadav and A. Rai, "Frequency and Temporal Convolutional Attention for Text-Independent Speaker Recognition," in *ICASSP 2020*, may 2020, pp. 6794–6798.

[35] Y. Zhao, T. Zhou, Z. Chen, and J. Wu, "Improving Deep CNN Networks with Long Temporal Context for Text-Independent Speaker Verification," in *ICASSP 2020*, may 2020, pp. 6834–6838.