

LEARNING TO FOOL THE SPEAKER RECOGNITION

Jiguo Li^{1,2,4}, Xinfeng Zhang⁴, Jizheng Xu³, Li Zhang³, Yue Wang³, Siwei Ma^{2†}, Wen Gao²

¹Institute of Computer Technology, Chinese Academic of Sciences,
²Peking University,
³Bytedance.Inc,
⁴University of Chinese Academic of Sciences

ABSTRACT

Due to the widespread deployment of fingerprint/face/speaker recognition systems, attacking deep learning based biometric systems has drawn more and more attention. Previous research mainly studied the attack to the vision-based system, such as fingerprint and face recognition. While the attack for speaker recognition has not been investigated yet, although it has been widely used in our daily life. In this paper, we attempt to fool the state-of-the-art speaker recognition model and present *speaker recognition attacker*, a lightweight model to fool the deep speaker recognition model by adding imperceptible perturbations onto the raw speech waveform. We find that the speaker recognition system is also vulnerable to the attack, and we achieve a high success rate on the non-targeted attack. Besides, we also present an effective method to optimize the speaker recognition attacker to obtain a trade-off between the attack success rate with the perceptual quality. Experiments on the TIMIT dataset show that we can achieve a sentence error rate of 99.2% with an average SNR 57.2dB and PESQ 4.2 with speed rather faster than real-time.

Index Terms— deep neural network attack, speaker recognition, convolution neural networks, adversarial examples generation

1. INTRODUCTION

Deep network based biometric systems, such as fingerprint/face/speaker recognition, have been widely deployed in our daily life. Meanwhile, finding the weakness and attacking these recognition systems also draw more and more attention. Although many works have been done on vision-based systems [1, 2, 3], the attack to speaker recognition has not been well-studied. There are two main applications of attacking the speaker recognition systems and finding the adversarial examples: (1) disturbing the speaker recognition systems when they are not wanted; (2) helping improve the performance and robustness of the speaker recognition

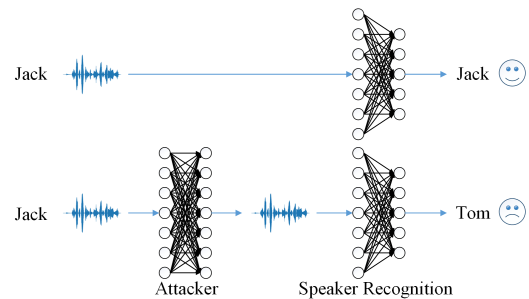


Fig. 1. Illustration of the Speaker Recognition Attacker

systems. In this work, we focus on attacking the speaker recognition and present a model as well as its optimization method to attack the well-trained state-of-the-art deep speaker recognition model by adding the perturbations on the input speech, as illustrated in Fig. 1.

Attacking the deep neural networks (DNNs) has become an emerging topic along with the development of DNNs since the weaknesses of DNNs have been found by Szegedy *et al.* [1]. On the vision tasks, some optimization methods, such as L-BFGS [1], Adam [4], or genetic algorithm [5], are used to modify the input pixels to obtain the adversarial examples. But these methods need the gradient or iterations during the testing phase, which are not practical in realistic scenarios. Baluja *et al.* [3] proposed adversarial transformation networks (ATNs), which create a separated attacker network, to transform all inputs into adversarial ones.

Base on the works on vision tasks, some methods are proposed to attack the automatic speech recognition (ASR) model. Alzantot *et al.* [6] proposed to attack the ASR model via a gradient-free genetic algorithm to generate adversarial examples iteratively. However, different from visual images, the psychoacoustic model shows that no difference will be perceived by humans if the distortion is under certain hearing thresholds. Therefore, Schonherr *et al.* [7] and Szurley *et al.* [8] proposed to optimize the attack with the psychoacoustic model, and add perturbations under the hearing threshold. Our work is different from these audio attack works in two

The work was done when Jiguo Li interned in Bytedance.Inc.

†Siwei Ma (swma@pku.edu.com) is the corresponding author.

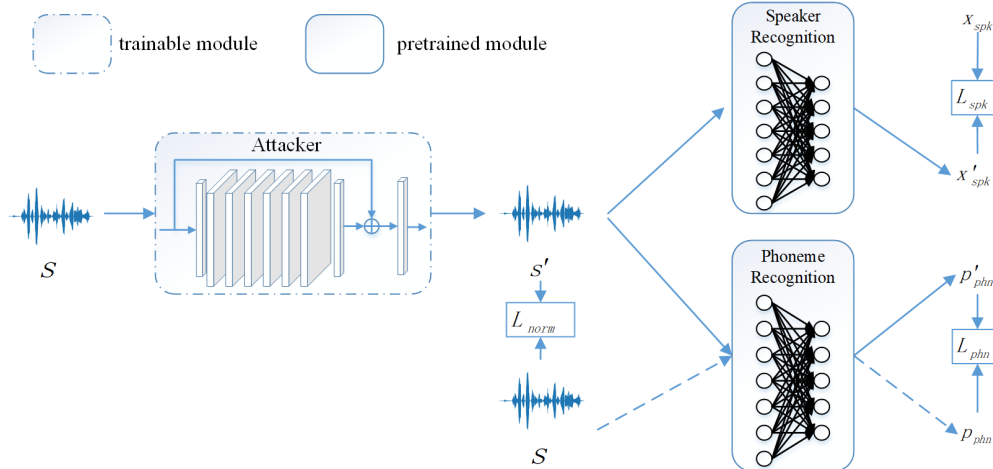


Fig. 2. Illustration of our framework. Our speaker recognition attacker is applied to the raw speech input and generates a speech with perturbations, which can fool the following speaker recognition model although the perturbations are imperceptible. Besides, a pretrained phoneme recognition model is used to help train the attacker network. No finetuning for the pretrained modules when training the attacker model.

aspects. First, our work mainly focuses on the different task to attack the speaker recognition model. Second, our model is based on ATNs, which need no gradient during the testing phase and is fast for inference. Besides, although the replaying attack has been explored [9], learning based attack has not been well-studied. Our contributions can be summarized as follows:

- We attempt to attack the state-of-the-art speaker recognition model, and find that it is vulnerable to the attack.
- We propose a model to attack the speaker recognition model. In the non-targeted experiments conducted on TIMIT dataset [10], we achieve a sentence error rate (SER) of 99.2% with the SNR up to 57.2dB and PESQ up to 4.2 with speed rather faster than real-time.
- We present an optimization method to train our model, and experimental results show that our method can achieve a trade-off between the attack success rate and the perceptual quality.

2. SPEAKER RECOGNITION ATTACKER

As illustrated in Fig. 2, our proposed speaker recognition attacker, a trainable network, is used to add generative perturbations onto the input speech to attack the following pretrained speaker recognition model. A pretrained phoneme recognition model is used to help train the attacker. Given a speech s and its speaker label y_{spk} , the non-targeted attack for the speaker recognition model can be formulated as:

$$\arg \min_{s'} L(s, s') \quad \text{s.t.} \quad f(s') = y'_{spk} \quad (1)$$

where $s' = T_{f,\theta}(s)$ and $y'_{spk} \neq y_{spk}$,

where T is the attack model to transform the input speech signal into an adversarial example, f is a well-trained speaker recognition model, L is a metric function to measure the distance between two samples (e.g., the L_2 norm). The constraint $y'_{spk} \neq y_{spk}$ is changed to $y'_{spk} = y_{target}$ for the targeted attack.

2.1. Attacker Network

The proposed attacker network is a fully convolution residual network, including 5 convolution blocks totally in the residual branch, as illustrated in Fig. 2. 1-D convolution, batchnorm, and ReLU are applied in every convolution block, following the setting in [3]. The kernel size for all convolution layers is set as 3 and the channel is set to be 32. To increase the receptive field, we use different dilations in the different convolution layers. The dilations for 5 convolution layers are 1, 2, 5, 2, 1, respectively. Besides, we init the weight and bias of the last convolution layer as zero so that our model adds no perturbation at the start of the training, which is important for the optimization to keep the perturbations on a small scale.

2.2. Optimization

The intuitive method to train the attacker network is gradient ascent, however, in practice, it fails because a well-trained speaker recognition model propagates back almost zero gradient due to the softmax layer. Motivated by the Wasserstein GAN [11] to optimize the Wasserstein distance between two distributions, we just solve this gradient missing problem to optimize directly on the immediate activation before the softmax layer. On the other hand, we also need to ensure the perturbations are imperceptible. L_2 norm is used to constraint the scale of the perturbations. Besides, we also take the phoneme

information into the account via a pretrained phoneme recognition network to optimize the perceptual quality. In summary, we optimize our attacker network from three aspects:

$$L_{total} = L_{spk} + \lambda_{phn}L_{phn} + \lambda_{norm}L_{norm} \quad (2)$$

$$L_{spk} = \begin{cases} x'_{spk}[I_{1st}] - x'_{spk}[I_{2nd}], & I_{1st} = y_{spk} \\ 0, & \text{else} \end{cases} \quad (3)$$

$$L_{phn} = \text{KL}(p_{phn} \| p'_{phn}) \quad (4)$$

$$L_{norm} = [\max(s - s' - m, 0)]^2, \quad (5)$$

where x' is the immediate activation before the softmax layer of the speaker recognition model with the input s' , p/p' is the output distribution of the softmax layer of the phoneme recognition model with the input s/s' , I_{1st}/I_{2nd} is the index of 1st/2nd largest value in x'_{spk} . In L_{phn} , we use Kullback–Leibler divergence (KLD) to measure the distance between two distributions. In L_{norm} , m is a hyper-parameter to give a margin in which the perturbations are thought imperceptible. λ_{phn} and λ_{norm} are used to fuse the three loss items. For the targeted attack, the loss is the same except that L_{spk} is changed to

$$L_{spk,target} = \begin{cases} x'_{spk}[I_{1st}] - x'_{spk}[y_{target}], & I_{1st} \neq y_{target} \\ 0, & \text{else,} \end{cases} \quad (6)$$

where y_{target} denotes the target speaker class.

2.3. Inference

The input speeches with variable length are split into frames with fixed length due to the fully connection layer in the speaker recognition model in the training stage. However, in the testing stage, the input speeches can be of arbitrary length because our attacker network is a fully convolution residual network. The inference is fast because our attacker network is lightweight with only 5 convolution blocks and small kernel size filters.

3. EXPERIMENTAL RESULT

3.1. Experimental Setup

Pretrained Speaker/Phoneme Recognition Model. We use the state-of-the-art speaker recognition model, SincNet [12], as the target model to attack. SincNet replaces the first layer of a CNN with a group of learnable bandpass filters. In this way, the network is more interpretable and show better performance [13]. Besides, SincNet also works on phoneme recognition¹. In our experiment, we use the official released pretrained SincNet model for speaker recognition. The phoneme recognition model is referred from

¹<https://github.com/mravanelli/pytorch-kaldi>

λ_{phn}	λ_{norm}	SER(%) \uparrow	SNR(dB) \uparrow	PESQ \uparrow
-	-	1.52*	-	-
0	0	99.7	18.56	1.09
0	1000	96.5	56.39	3.72
0	2000	86.7	57.79	3.61
1	1000	99.2	57.20	4.20
5	1000	93.9	58.00	4.25
10	1000	90.5	59.01	4.28

Table 1. Non-targeted attack results. The first row is the baseline performance of speaker recognition without attack. The rest of the table shows the attack results of our model with different trade-offs between SER with SNR and PESQ by tuning λ_{phn} and λ_{norm} . *This result is not the same as that in [12], but this model is released by the author.

Pytorch-Kaldi [14] and achieve a 26.4% frame error rate on the TIMIT[10] dataset².

Dataset and Metric. Following the setting in [12], we conduct experiments on TIMIT (462 speakers, train chunk) [10] to demonstrate the performance of our proposed model. The signal-to-noise ratio (SNR) and Perceptual Evaluation of Speech Quality (PESQ) score [15] are used to evaluate the objective and perceptual quality, respectively. SNR is calculated as follows:

$$\text{SNR} = 10 \log_{10}[(\sigma_s^2/\sigma_e^2)]^2, \quad (7)$$

where σ_s^2/σ_e^2 is the mean square of input signal/error. PESQ [15] is an integrated perceptual model to map the distortion to a prediction of subjective mean opinion score (MOS) with range [0.5, 4.5], which is an ITU-T recommendation technology [16]. Following the works for attacking image classification [1, 2], we use the classification error rate (CER) to evaluate the performance of our attacker. For the non-targeted attack, sentence error rate (SER), which is defined as the CER of the speech sentences, is used to measure our attacker’s performance. For the targeted attack, our attack is successful as long as the prediction is the target, so we use prediction target rate (PTR), which is the percentage of the target in the predictions over the testing set, to measure our attacker’s performance.

Training Details. The speech sentences, with sampling rate 16k, are split into 200ms frames with 10ms stride, following [12]. The data will be normalized before being feed into the attacker model and de-normalized when they output the attacker model. After finetuning the hyper-parameters, we set $\lambda_{phn} = 1$, $\lambda_{norm} = 1000$ and $m = 0.01$. We use Adam [17] optimizer with a learning rate of 3×10^{-4} to train the attack model for 10 epochs. Data, code, and pretrained models have been released on our project home page³.

²we use the same train/test split for phoneme and speaker recognition, which is different with the typical split manner for phoneme recognition.

³<https://smallflyingpig.github.io/speaker-recognition-attacker/main>

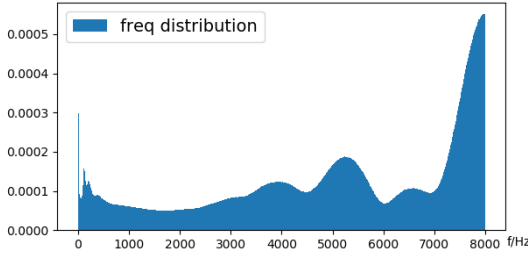


Fig. 3. The spectrogram energy distribution of the perturbations over the testing set.

3.2. Non-targeted Attack

To demonstrate the effectiveness of our proposed model, we conduct non-targeted attack experiments on the TIMIT dataset. The results are illustrated in Table. 1, some conclusions can be drawn:

- Our proposed model successfully attacks the trained state-of-the-art speaker recognition model with an SER of 99.2% on the testing set. Meanwhile, the perturbations are small enough to be imperceptible for humans because the SNR is up to 57.2dB and PESQ is no less than 4.2, indicating the efficiency of our model.
- Tuning λ_{norm} fails to get a trade-off between SER and PESQ (row 1st, 2nd, 3rd in Table. 1), while tuning λ_{phn} works for that (2nd, 4th~6th in Table. 1), demonstrating our optimization is effective to achieve a trade-off between the attack success rate and the perceptual quality.
- The results in the 2nd row and 4th row show that L_{phn} can improve the performance of the attacker model (measured by SER), as well as the quality of the adversarial examples (measured by SNR and PESQ).

Besides, we also give the distribution of the perturbations on the frequency domain to study if the perturbations have frequency selectivity. The frequency distributions of the perturbations from the attacker model with $\lambda_{phn} = 1$ are shown in Fig. 3. The spectrogram energy distribution shows that: (1) the perturbations are full-band, and all the frequencies are useful for the attack; (2) the energy in the 7k~8k band is significantly stronger than that in other bands, indicating the high-frequency band has much affect on the speaker recognition’s performance. The frequency characteristics of perturbations have a great influence on the model performance and signal quality, but looking deeper into this question is beyond the scope of this paper.

3.3. Targeted Attack

Besides the non-targeted attack, we also evaluate our model on the targeted attack. We fix the hyper-parameters as

Target ID	PTR(%) \uparrow	SNR(dB) \uparrow	PESQ \uparrow
0	91.4	57.55	3.36
100	89.3	56.83	3.16
200	63.3	58.42	3.69
300	58.7	56.92	3.52
400	57.6	58.36	3.68
avg	72.1	57.64	3.48

Table 2. Targeted attack results. The first five rows are the results of the targeted attack for five randomly selected speakers from the TIMIT dataset. The last row is the average result of the above five rows. PTR denotes the prediction target rate.

$\lambda_{norm} = 1000, \lambda_{phn} = 1, m = 0.01$ in the targeted attack. Five speakers are randomly selected from the 462 speakers of the TIMIT dataset as targets. The attack results are illustrated in Table. 2. The results show that:

- Our model can attack the speaker recognition with a PTR of 72.1% on average over the five targets. Meanwhile, the perturbations are small enough because the SNR is up to 57.64dB.
- The PESQ (3.48 on average) is well enough although it is not as good as that in the non-targeted attack, given the fact that targeted attack is more challenging than the non-targeted attack.

3.4. Real-time Attack

Our attack model is very lightweight with only 5 convolution blocks, so it is very fast. To verify it is fast enough to process the speech data in real-time, we calculate the real-time factor(RTF) over the testing set. RTF is defined as the ratio of the processing time to the input duration and the system is real-time if $RTF \leq 1$. We test our model in CPU mode on a machine with an Inter(R) Core(TM) i7-6700K @ 3.4GHz CPU and get an average RTF 0.042, indicating that our attacker is more than 20 times faster than the real-time requirement, although it runs in the CPU mode.

4. CONCLUSION

In this paper, we proposed a model to attack the speaker recognition model by training a lightweight attacker network to add perturbations on the input speech. Experiments show that our model was effective and efficient on both non-targeted and targeted attacks. We have built a pioneer work on the learning based speaker recognition attack and established the corresponding benchmark for such study. In the future, the black-box attack and transferable attack will be explored because the gradient of the target is usually inaccessible in the real scenarios.

5. REFERENCES

- [1] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus, “Intriguing properties of neural networks,” *arXiv preprint arXiv:1312.6199*, 2013.
- [2] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy, “Explaining and harnessing adversarial examples,” *arXiv preprint arXiv:1412.6572*, 2014.
- [3] Shumeet Baluja and Ian Fischer, “Learning to attack: Adversarial transformation networks,” in *AAAI*, 2018, pp. 2687–2695.
- [4] Nicholas Carlini and David Wagner, “Towards evaluating the robustness of neural networks,” in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 39–57.
- [5] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai, “One pixel attack for fooling deep neural networks,” *IEEE Transactions on Evolutionary Computation*, 2019.
- [6] Moustafa Alzantot, Bharathan Balaji, and Mani Srivastava, “Did you hear that? adversarial examples against automatic speech recognition,” *arXiv preprint arXiv:1801.00554*, 2018.
- [7] Lea Schönherr, Katharina Kohls, Steffen Zeiler, Thorsten Holz, and Dorothea Kolossa, “Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding,” *arXiv preprint arXiv:1808.05665*, 2018.
- [8] Joseph Szurley and J Zico Kolter, “Perceptual based adversarial audio attacks,” *arXiv preprint arXiv:1906.06355*, 2019.
- [9] Pavel Korshunov, Sébastien Marcel, Hannah Muckenhirn, André R Gonçalves, AG Souza Mello, RP Velloso Violato, Flávio O Simoes, M Uliani Neto, Marcus de Assis Angeloni, José Augusto Stuchi, et al., “Overview of btas 2016 speaker anti-spoofing competition,” in *2016 IEEE 8th international conference on biometrics theory, applications and systems (BTAS)*. IEEE, 2016, pp. 1–6.
- [10] John S Garofolo, Lori F Lamel, William M Fisher, Jonathan G Fiscus, and David S Pallett, “Darpa timit acoustic-phonetic continuous speech corpus cd-rom. nist speech disc 1-1.1,” *NASA STI/Recon technical report n*, vol. 93, 1993.
- [11] Martin Arjovsky, Soumith Chintala, and Léon Bottou, “Wasserstein gan,” *arXiv preprint arXiv:1701.07875*, 2017.
- [12] Mirco Ravanelli and Yoshua Bengio, “Speaker recognition from raw waveform with sincnet,” in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 1021–1028.
- [13] Mirco Ravanelli and Yoshua Bengio, “Interpretable convolutional filters with sincnet,” *arXiv preprint arXiv:1811.09725*, 2018.
- [14] M. Ravanelli, T. Parcollet, and Y. Bengio, “The pytorch-kaldi speech recognition toolkit,” in *In Proc. of ICASSP*, 2019.
- [15] Antony W Rix, John G Beerends, Michael P Hollier, and Andries P Hekstra, “Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs,” in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*. IEEE, 2001, vol. 2, pp. 749–752.
- [16] “Itu-t recommendation p.862: Perceptual evaluation of speech quality (pesq): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs,” 2001.
- [17] Diederik P Kingma and Jimmy Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.