Poster: SirenAttack: Generating Adversarial Audio for End-to-End Acoustic Systems

Tianyu Du*, Shouling Ji*†, Jinfeng Li*, Qinchen Gu‡, Ting Wang§ and Raheem Beyah‡
* Institute of Cyberspace Research and College of Computer Science and Technology, Zhejiang University
Email: {zjradty, sji, lijinfeng0713}@zju.edu.cn
† Alibaba-Zhejiang University Joint Research Institute of Frontier Technologies
‡ Georgia Institute of Technology, Email: guqinchen@gatech.edu, rbeyah@ece.gatech.edu
§ Lehigh University, Email: inbox.ting@gmail.com

***Abstract*—In this poster, we present SIRENATTACK, a new class of attacks to generate adversarial audios. Compared with existing attacks, SIRENATTACK highlights with a set of significant features, i.e., versatile, targeted, and evasive. Experimental results on a set of state-of-the-art deep learning-based acoustic systems demonstrate the versatility, effectiveness, and stealthiness of SIRENATTACK.**

I. INTRODUCTION, PRELIMINARY AND ATTACK DESIGN

Nowadays deep learning-based acoustic systems are ubiquitous in our everyday lives, ranging from smart locks on mobiles to speech assistants on smart home devices and to machine translation services on clouds. However, deep neural networks (DNNs) are inherently vulnerable to adversarial inputs, which are maliciously crafted samples to trigger target models to misbehave [2]. Despite the plethora of work on the image domain, the research of adversarial attacks on the audio domain is still limited, due to a number of non-trivial challenges. First, the acoustic systems need to deal with information changes in the time dimension, which is more complex than image classification systems. Second, the audio sampling rate is usually very high, but images only have hundreds/thousands of pixels in total. Therefore, it is harder to craft adversarial audios than images since adding slight noise to audios are less likely to impact the local features.

In this poster, we present SIRENATTACK, a new class of adversarial attacks against deep neural network-based acoustic systems. Compared with prior work, SIRENATTACK departs in significant ways: *versatile* – SIRENATTACK is applicable to a range of end-to-end acoustic systems under both white-box and black-box settings; *targeted* – SIRENATTACK generates adversarial audio that trigger target systems to misbehave in a highly predictable manner (e.g., misclassifying the adversarial audio into a specific class); and *evasive* – SIRENATTACK is able to generate adversarial audios indistinguishable from their benign counterparts to human perception.

SIRENATTACK is based on the Particle Swarm Optimization (PSO) algorithm [1]. PSO is a heuristic and stochastic algorithm to find solutions for optimization problems by imitating the behavior of a swarm of birds. It can search a very large space of candidate solutions while does not require the gradient information. At a high level, it solves an optimization problem by iteratively making a population of candidate solutions (which we referred to as $particles$) move around in the search-space according to their fitness values. The fitness value of a particle is the evaluation result of the objective function on that particle's position in the solution space. In each iteration, each particle's movement is influenced by its local best position P_{best}, and meanwhile is guided toward the global best position G_{best} in the search-space. This iteration process is expected to move the swarm toward the best solution. Once a termination criterion is met, G_{best} should hold the solution for a local minimum.

The detailed black-box attack is shown in Algorithm 1. To fool a machine learning model, we feed it with a legitimate audio $\boldsymbol{x}$ and the target output t. First, we initialize the $epoch$ to zero and generate $n\_particle$ randomized sequences (collectively referred to as $seeds$) from a uniform distribution (line 1). Then we run the PSO subroutine (line 3) with the target output t and $seeds$. If any particle $\boldsymbol{p_i}$ produces the target output t when being added to the original audio $\boldsymbol{x}$, then the attack succeeds (line 4-5), and the particle $\boldsymbol{p_i}$ is the expected noise $\boldsymbol{\delta}$. Otherwise, we will preserve the best particle that has the minimum fitness value in the current PSO run as one of the $seeds$ in the next PSO run (line 7-8). The above steps iterate (line 2-11) till the attack succeeds or it reaches $epoch_{max}$. If succeed, we would obtain an adversarial audio $\boldsymbol{x_{adv}}$ that can be predicted as t by the victim model.

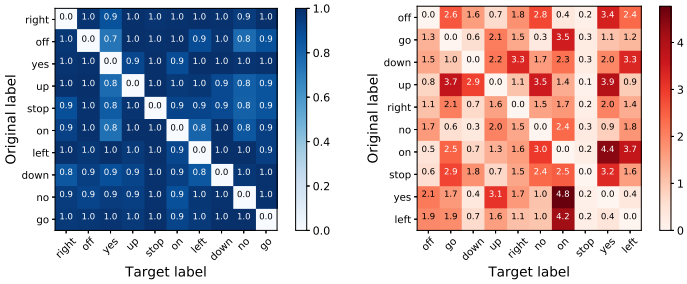
Algorithm 1 SIRENATTACK under black-box settings

Input: Original audio $\boldsymbol{x}$, target output t, $n\_particles$ and $epoch_{max}$
Output: A targeted adversarial audio $\boldsymbol{x_{adv}}$
1: Initialize $epoch = 0$ and $seeds$ and set Eq. (1) as the objective function;
2: **while** epoch reaches $epoch_{max}$ **do**
3: Run PSO subroutine with t and $seeds$;
4: **if** any particle produce target output t during PSO **then**
5: Solution is found. Exit.
6: **else**
7: Clear $seeds$;
8: $seeds \supseteq best\ particle$ that produce the minimum value of Eq. (1) from the current PSO run;
9: **end if**
10: $epoch$ = $epoch$ + 1;
11: **end while**
12: Get adversarial audio $\boldsymbol{x_{adv}}$ with target label t.

We would further emphasize two key aspects of our algorithm: (1) We modify the PSO algorithm to globally keep track

TABLE I. PERFORMANCE OF THE BLACK-BOX ATTACK.

Model	Accuracy	Success Rate	SNR(dB)	Time(s)
CNN	96.10%	95.25%	22.36	100.69
VGG19	91.39%	88.10%	18.22	332.26
DenseNet	94.93%	86.90%	15.34	458.13
ResNet18	92.06%	87.35%	15.87	340.31
ResNeXt	94.28%	90.05%	17.03	317.92
WideResNet18	90.80%	89.25%	17.57	368.29
DPN92	95.20%	83.60%	14.04	462.58



(a) Success Rate (b) Time (min)

Fig. 1. Performance of SIRENATTACK for every $\{source, target\}$ pair on the Speech Commands Dataset against the CNN model.

of the current saved best particle throughout all PSO iterations instead of using the standard PSO algorithm. (2) During each iteration, PSO aims to minimize an objective function defined as $g(\boldsymbol{x}+\boldsymbol{p_i})$. We experimented with several definitions of $g(\cdot)$ and found the following to be the most effective:

$$g(\boldsymbol{x}+\boldsymbol{p_i}) = \max(\max_{j\neq t}(\mathcal{O}(\boldsymbol{x}+\boldsymbol{p_i})_j) - \mathcal{O}(\boldsymbol{x}+\boldsymbol{p_i})_t, \kappa) \quad (1)$$

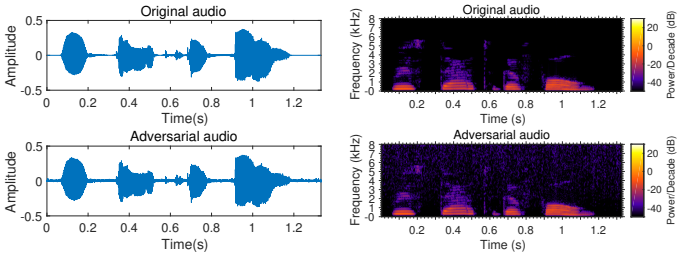
where $\mathcal{O}(\boldsymbol{x}+\boldsymbol{p_i})_j$ is the confidence value of label j for input $\boldsymbol{x}+\boldsymbol{p_i}$. The function can move the particles to the position that maximizes the probability of the target label t. In addition, we can control the confidence of misprediction with the parameter κ, and a smaller κ means that the found adversarial audio will be predicted as t with higher confidence. We set $\kappa = 0$ for SIRENATTACK but we note here that a side benefit of this formulation is that it allows one to control the desired confidence. In addition, this function can be used to conduct untarget attacks with trivial modifications.

II. EXPERIMENTS AND CONCLUSION

We conducted black-box attacks under four different scenes, including speech command recognition, speaker recognition, audio scene classification and music genre classification. Due to the limitation of pages, we only show part of the experimental results. For speech command recognition task, we evaluated SIRENATTACK on Speech Commands Dataset [4] against the *CNN* described in [3] and other six state-of-the-art speech command recognition models, i.e., *VGG19*, *DenseNet*, *ResNet18*, *ResNeXt*, *WideResNet18* and *DPN-92*. In addition, we use SNR (Signal Noise Ratio) to evaluate the audio quality, which is calculated as follows:

$$SNR(dB) = 10\ log_{10}(\frac{P_{\boldsymbol{x}}}{P_{\boldsymbol{\delta}}}) \quad (2)$$

where $\boldsymbol{x}$ is the original audio waveform, $\boldsymbol{\delta}$ is the added noise, and $P_{\boldsymbol{x}}$ and $P_{\boldsymbol{\delta}}$ are the power of the original signal and the noise signal, respectively.



(a) Waveform (b) Spectrogram

Fig. 2. Comparison of the waveform and spectrogram between an original audio (upper graphs) and the adversarial counterpart (lower graphs) with $\delta = 100$. The original transcription is "*restart the phone*" while the adversarial transcription is "*open the front door*".

TABLE II. TRANSFERABILITY EVALUATION RESULTS.

	Sphinx	Google	Bing	Houndify	Wit.ai	IBM
Success Rate	39.60%	10.00%	14.00%	12.80%	21.20%	20.40%

TABLE III. TRANSFERABILITY EVALUATION: EXAMPLE RESULTS.

Number	Original	Advesarial	ASR Platforms	Results
1	stop	no	Sphinx	no
2	off	on	IBM	on
3	down	no	Sphinx, Wit.ai	no
4	down	no	Wit.ai, Bing	no
5	go	no	Wit.ai	no
6	go	yes	Sphinx	yes
7	left	yes	Wit.ai, IBM	yeah
8	right	on	Google, Bing	play

Table I shows the main experimental results with $\delta = 800$ and $epoch_{max} = 300$. Fig. 1 shows the pair-to-pair success rate and the average time to generate an adversarial audio of SIRENATTACK. Fig. 2 shows the waveform and spectrogram of an example original audio and the adversarial couterpart. Furthermore, we also evaluated the transferability of the generated adversarial audios (against the VGG19 model) and show the results in Table II. Some successful transferred examples are shown in Table III. From the above experimental results we can see that (i) SIRENATTACK is effective when against all the targeted models, even when the models have high performance on the legitimate datasets; (ii) the average time of generating an adversarial audio is very short; (iii) the noise in the generated adversarial audios is almost ignorable; and (iv) the generated adversarial audio has transferability to some extent.

REFERENCES

[1] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the Sixth International Symposium on Micro Machine and Human Science (MHS'95)*. IEEE, 1995, pp. 39–43.

[2] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015, pp. 1–11.

[3] T. N. Sainath and C. Parada, "Convolutional neural networks for small-footprint keyword spotting," in *Proceedings of the 16th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2015, pp. 1478–1482.

[4] P. Warden, "Speech commands: A public dataset for single-word speech recognition." *Dataset available from http://download.tensorflow.org/data/speech_commands_v0.01.tar.gz*, 2017.

SirenAttack: Generating Adversarial Audio for End-to-End Acoustic Systems

Tianyu Du[1] Shouling Ji[1] Jinfeng Li[1] Qinchen Gu[2] Ting Wang[3] Raheem Beyah[2]

1.Zhejiang University 2. Georgia Institute of Technology 3. Lehigh University

Introduction

- Nowadays, deep learning-based acoustic systems are ubiquitous in our everyday lives, ranging from smart locks on mobiles to speech assistants on smart home devices. However, deep neural networks (DNNs) are inherently vulnerable to adversarial inputs, which are maliciously crafted samples to trigger target models to misbehave [1].
- We present **SirenAttack**, a new class of adversarial attacks against deep neural network-based acoustic systems. Compared with prior work, SirenAttack departs in significant ways: ***versatile*** – SirenAttack is applicable to a range of end-to-end acoustic systems under both white-box and black-box settings; ***targeted*** – SirenAttack generates adversarial audio that trigger target systems to misbehave in a highly predictable manner (e.g., misclassifying the adversarial audio into a specific class); and ***evasive*** – SirenAttack is able to generate adversarial audios indistinguishable from their benign counterparts to human perception.

SirenAttack

- Particle Swarm Optimization (PSO) [2] solves an optimization problem by iteratively making a population of candidate solutions move around in the search-space according to their fitness values.
- Update the *i-th* particle's velocity:
  $v_i^k = wv_i^{k-1} + c_1r_1(pbest_i - x_i^{k-1}) + c_2r_2(gbest_d - x_i^{k-1})$
- Update the *i-th* particle's position:
  $x_i^k = x_i^{k-1} + v_i^{k-1}$

Algorithm 1 SIRENATTACK under black-box settings

Input: Original audio $\boldsymbol{x}$, target output t, $n\_particles$ and $epoch_{max}$
Output: A targeted adversarial audio $\boldsymbol{x_{adv}}$

```
Initialize epoch = 0 and seeds and set Eq. (1) as the objective function;
while epoch reaches epoch_max do
    Run PSO subroutine with t and seeds;
    if any particle produce target output t during PSO then
        Solution is found. Exit.
    else
        Clear seeds;
        seeds ⊇ best particle that produce the minimum value of Eq. (1) from the current PSO run;
    end if
    epoch = epoch + 1;
end while
Get adversarial audio x_adv with target label t.
```

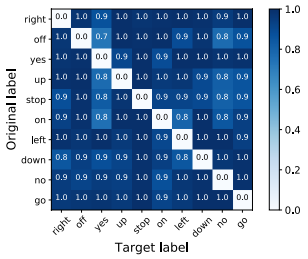
- We modified the PSO to globally keep track of the current saved best particle throughout all iterations.
- Objective function:
  $g(\boldsymbol{x}+\boldsymbol{p_i}) = \max(\max_{j\neq t}(\mathcal{O}(\boldsymbol{x}+\boldsymbol{p_i})_j) - \mathcal{O}(\boldsymbol{x}+\boldsymbol{p_i})_t, \kappa)$

Attack Evaluation

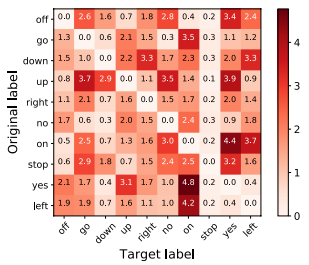
Dataset: Speech Commands
Targeted Model: CNN [3], VGG19, DenseNet, ResNet18, ResNeXt, WideResNet18, DPN92
Evaluation Metric: $SNR(dB) = 10\log_{10}(\frac{P_x}{P_\delta})$

Model	Accuracy	Success Rate	SNR(dB)	Time(s)
CNN	96.10%	95.25%	22.36	100.69
VGG19	91.39%	88.10%	18.22	332.26
DenseNet	94.93%	86.90%	15.34	458.13
ResNet18	92.06%	87.35%	15.87	340.31
ResNeXt	94.28%	90.05%	17.03	317.92
WideResNet18	90.80%	89.25%	17.57	368.29
DPN92	95.20%	83.60%	14.04	462.58

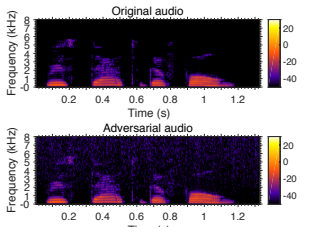
Pair-to-pair Success Rate



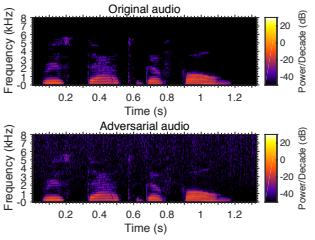
Average Time (min)



Waveform



Spectrogram



Transferability

- Offline VGG19 → Online ASR Platforms

	Sphinx	Google	Bing	Houndify	Wit.ai	IBM
Success Rate	39.60%	10.00%	14.00%	12.80%	21.20%	20.40%

Successful Examples

Number	Original	Adversarial	ASR Platforms	Results
1	stop	no	Sphinx	no
2	off	on	IBM	on
3	down	no	Sphinx, Wit.ai	no
4	down	no	Wit.ai, Bing	no
5	go	no	Wit.ai	no
6	go	no	Sphinx	no
7	go	yes	Sphinx	yes
8	left	yes	Wit.ai, IBM	yeah
9	on	right	Wit.ai	alright
10	right	on	Google, Bing	play
11	right	down	Google, Bing	play
12	right	go	Google	play
13	off	right	Bing	call
14	down	no	Wit.ai	okay
15	right	up	Bing	skype
16	stop	off	Wit.ai	the
17	down	up	Bing	phone
18	on	right	Sphinx	phone
19	stop	go	Wit.ai	tell
20	on	stop	Bing	home

Reference

[1] I. J. Goodfellow, J. Shlens, and C. Szegedy, Explaining and harnessing adversarial examples, *ICLR*, 2015.
[2] R. Eberhart and J. Kennedy, A new optimizer using particle swarm theory, *MHS*, 1995.
[3] T. N. Sainath and C. Parada, "Convolutional neural networks for small- footprint keyword spotting," *INTERSPEECH*, 2015, pp. 1478–1482.