

# Attack on Speech/Speaker Recognition

---

## Attack on Speech/Speaker Recognition

[Todo List](#)

[Overview](#)

\* [Cocaine Noodles: Exploiting the Gap between Human and Machine Speech Recognition](#)

[Contribution](#)

[Notes](#)

[Shortcoming](#)

[Links](#)

[Hidden Voice Commands](#)

[Contribution](#)

[Notes](#)

[Black-box Attacks](#)

[White-box Attacks](#)

[Defense](#)

[Shortcoming](#)

[Links](#)

[DolphinAttack: Inaudible voice commands](#)

[Contribution](#)

[Notes](#)

[Shortcoming:](#)

[Links](#)

\* [Did you hear that? Adversarial Examples Against Automatic Speech Recognition](#)

[Contribution](#)

[Notes](#)

[Links](#)

[Audio Adversarial Examples: Targeted Attacks on Speech-to-Text](#)

[Contribution](#)

[Notes](#)

[Links](#)

[Adversarial Attacks Against Automatic Speech Recognition Systems via Psychoacoustic Hiding](#)

[Contribution](#)

[Notes](#)

[Links](#)

\* [Targeted adversarial examples for black box audio systems](#)

[Contribution](#)

[Notes](#)

[Links](#)

[Robust Audio Adversarial Example for a Physical Attack](#)

[Contribution](#)

[Notes](#)

[Links](#)

[Hear "No Evil", See "Kenansville": Efficient and Transferable Black-Box Attacks on Speech Recognition and Voice Identification Systems](#)

[Contribution](#)

[Notes](#)

[Links](#)

[Imperceptible, Robust, and Targeted Adversarial Examples for Automatic Speech Recognition](#)

[Contribution](#)

[Notes](#)

[Links](#)

- \* Practical Hidden Voice Attacks against Speech and Speaker Recognition Systems
  - [Contribution](#)
  - [Notes](#)
  - [Links](#)
- \* Adversarial Black-Box Attacks on Automatic Speech Recognition Systems using Multi-Objective Evolutionary Optimization
  - [Contribution](#)
  - [Notes](#)
  - [Links](#)
- [Who is Real Bob? Adversarial Attacks on Speaker Recognition Systems](#)
  - [Contribution](#)
  - [Notes](#)
  - [Links](#)
- [SoK: The Faults in our ASRs: An Overview of Attacks against Automatic Speech Recognition and Speaker Identification Systems](#)
  - [Notes](#)
  - [Links](#)

## Todo List

---

1. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.
2. Biggio, B., Corona, I., Maiorca, D., Nelson, B., "Srndi'c, N., Laskov, P., Giacinto, G., and Roli, F. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pp. 387–402. Springer, 2013.
3. A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Conference on Computer Vision and Pattern Recognition*. IEEE, Jun. 2015, pp. 427–436.
4. N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Symposium on Security and Privacy*. IEEE, May 2017, pp. 39–57.
5. I. Evtimov, K. Eykholt, E. Fernandes, T. Kohno, B. Li, A. Prakash, A. Rahmati, and D. Song, "Robust physical-world attacks on machine learning models," *CoRR*, vol. abs/1707.08945, pp. 1–11, Jul. 2017.
6. Moustapha Cissé, Yossi Adi, Natalia Neverova, and Joseph Keshet. Houdini: Fooling deep structured visual and speech recognition models with adversarial examples. In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems*, pages 6980–6990, 2017.
7. Dibya Mukhopadhyay, Maliheh Shirvanian, and Nitesh Saxena. 2015. All your voices are belong to us: Stealing voices to fool humans and machines. In *Proceedings of the European Symposium on Research in Computer Security*. Springer, 599–621.
8. Mel-Frequency Spectral Coefficients (MFSC)
9. Linear Predictive Coding
10. Perceptual Linear Prediction (PLP)
11. Wang, Qing, Pengcheng Guo, and Lei Xie. "Inaudible Adversarial Perturbations for Targeted Attack in Speaker Recognition." *arXiv preprint arXiv:2005.10637* (2020).
12. Dangerous Skills: Understanding and Mitigating Security Risks of Voice-Controlled Third-Party Functions on Virtual Personal Assistant Systems
13. DeepSec: A Uniform Platform for Security Analysis of Deep Learning Models
14. Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks

15. AdvPulse: Universal, Synchronization-free, and Targeted Audio Adversarial Attacks via Subsecond Perturbations
16. Dangerous Skills Got Certified: Measuring the Trustworthiness of Skill Certification in Voice Personal Assistant Platforms
17. When the Differences in Frequency Domain are Compensated: Understanding and Defeating Modulated Replay Attacks on Automatic Speech Recognition
18. PatternListener: Cracking Android Pattern Lock Using Acoustic Signals
19. Y. Gong and C. Poellabauer, "Crafting adversarial examples for speech paralinguistics applications," in DYNAMICS, 2018.
20. F. Kreuk, Y. Adi, M. Cisse, and J. Keshet, "Fooling end-to-end speaker verification with adversarial examples," in ICASSP, 2018.

## Overview

---

1. 语音识别的流程  
todo there
2. 针对不同截断的攻击  
todo there
3. 语音识别领域对抗攻击的演变  
todo there

## \* Cocaine Noodles: Exploiting the Gap between Human and Machine Speech Recognition

---

### Contribution

1. 这篇文章可以说是语音识别对抗攻击的开山之祖；
2. 攻击特征提取模块；

### Notes

1. 攻击 **特征提取(MFCC)** 模块的对抗攻击；
2. ↗ 攻击方法的流程图如下：

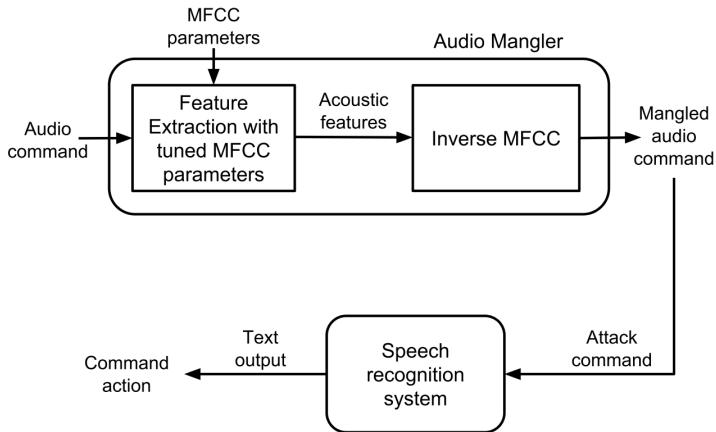


Figure 2: Attack outline.

作者的攻击思路非常简单，从语音中提取出 MFCC 特征（这个过程会 **丢失一些语音的信息**，如 MFCC 特征往往只取前 13 个参数，而高维的 MFCC 特征代表着能量变化的高频信息），然后把 **MFCC 特征逆转为语音信号**，只要该样本同时满足“**人耳无法理解**”，“**机器可以理解**”两个条件，那么这就是一个成功的对抗样本。

## Shortcoming

1. 需要大量的时间去生成一个对抗样本，因为要调参数使得满足“**人耳无法理解**”，“**机器可以理解**”两个条件；
2. ~~乍一看（纯属吐槽）文章的编写我觉得挺烂的，你想知道的你都没有知道，你不想知道的他介绍了好一堆。如果你熟悉语音识别、MFCC的话，会发现整篇文章就只有两块（介绍了语音识别和做了个问卷调查），对于实际的攻击算法的实现、如何去调参生成一个对抗样本（作者的描述是：我生成了 500 个样本）并没有提及，甚至没有介绍相关的一些参数，代码也是没有开源的（计算 MFCC 的链接在下面）；~~
3. ~~（猜测一下）整个算法的流程大概是：正常计算得到MFCC特征，然后用**逆 DCT 变换**（对应 Mel Filter Bank Energy 到 MFCC 过程）和**逆 DFT 变换**（对应时域信号到频谱过程）。虽然算法特别简单，但是因为涉及到帧之间的重叠（分帧的时候一般 `step_length < frame_length`），整个调试过程应该是比较麻烦的事情；~~

## Links

- 论文链接：[Vaidya, Tavish, et al. "Cocaine noodles: exploiting the gap between human and machine speech recognition." 9th {USENIX} Workshop on Offensive Technologies \({WOOT} 15\), 2015.](#)
- MFCC 实现：[PLP and RASTA \(and MFCC, and inversion\) in Matlab using melfcc.m and invmelfcc.m \(columbia.edu\)](#).

## Hidden Voice Commands

### Contribution

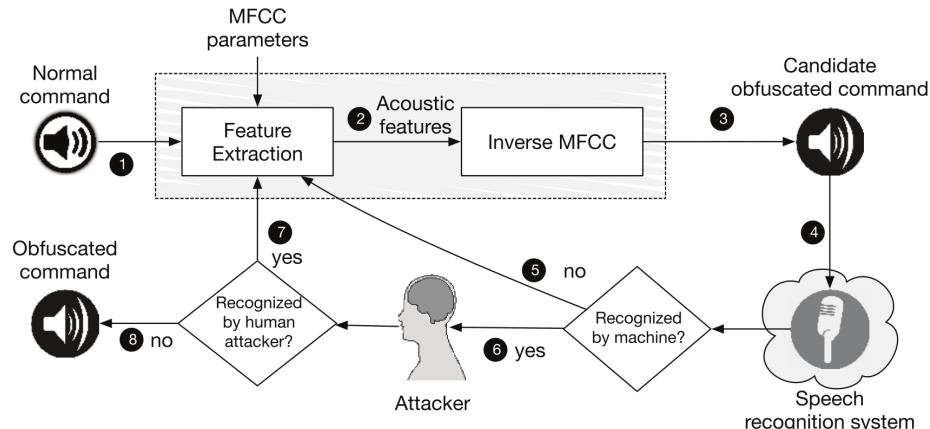
1. “实验设定”值得借鉴，让阅读者很清晰地了解实验情景，能够对攻击性有一个简单的预估；
2. 攻击特征提取模块；
3. 使用梯度下降算法攻击 GMM-HMM 模型；
4. 虽然这篇文章有挺多的不足，但语音识别的对抗攻击从这篇文章开始有了比较清晰的思路；
5. 可以看到，Carlini 等人在对抗攻击上面是非常有经验的，其实他们已经看到了全部的攻击向量：**特征提取模块能不能攻击，模型能不能攻击，环境如何影响攻击的成功率，人对对抗样本**

的感受是怎样的，如何防御对抗样本 等等。

## Notes

### Black-box Attacks

1. 整体思路：将正常的 TTS 语音模糊化；
2. 这部分是对文章 "[Cocaine Noodles: Exploiting the Gap between Human and Machine Speech Recognition](#)" 的完善。算法流程如下：



通过计算 MFCC 然后 MFCC 逆转为时域信号这个过程，算法就能够从理论上**保留语音识别算法关注的语音特征，而抹除不相关的语音特征**（上一篇论文中已经解释这个原理），而抹除的这部分特征又很可能对人的听觉影响是很大的，导致人无法听清对抗样本；

3. 实验设定：介绍了使用的扬声器的型号，实验房间的大小、背景噪声，测试时设备的距离为 **3米**，使用三星和苹果手机中的 Google；背景噪声使用 JBL 播放，噪声的大小约为 53dB；
4. Evaluation：
  - (1) Attack Range：当攻击距离大于 **3.5米** 或者 **SNR<5dB** 时，攻击就无法成功；、
  - (2) Obfuscating Parameter：

Parameter	Description
wintime	time for which the signal is considered constant
hoptime	time step between adjacent windows
numcep	number of cepstral coefficients
nbands	no. of warped spectral bands for aggregating energy levels

(2) Results：简单比较人和机器在指令模糊后的识别率的变化

	Ok Google		Turn on airplane mode		Call 911	
	Machine	Human	Machine	Human	Machine	Human
Normal	90% (36/40)	89% (356/400)	75% (30/40)	69% (315/456)	90% (36/40)	87% (283/324)
Obfuscated	95% (38/40)	22% (86/376)	45% (18/40)	24% (109/444)	40% (16/40)	94% (246/260)

### White-box Attacks

1. 整体思路：梯度下降算法寻找一个样本点；
2. Simple Approach：使用梯度下降算法，生成对抗样本；但是作者发现这样找到的样本**并没有比 Black-box Attacks 找到的样本要更好**；梯度下降算法的目标函数：

$$f(x) = (MFCC(x) - y)^2 \cdot z$$

即希望计算得到的 MFCC 特征和目标的 MFCC 相近，z 是一个权重因子，作者直接取单位向量；

3. Improved Approach：

(1) 扩展 MFCC 维度：计算 MFCC 后输出的是 13 维的 MFCC 特征，通常在语音识别中会用它的导数将其扩展到 39 维；这边有一个不太好理解的地方，前面输出 13 维的时候是将 MFCC 特征给截断了，为什么又要把它扩展到 39 维呢？因为前面截断特征舍弃的是一帧中 MFCC 的高维分量，这个分量指的是该帧能量谱变化较快的信息（这部分对我们语音识别是没有用的，但可能对说话人识别是有用的），保留的低维分量是能量谱变化较慢的信息（能量谱的包围）；而后面扩展维度作导数是将前后帧 MFCC 作差求导，得到的是前后帧 MFCC 的变化信息（语音的前后帧具有很强的相关性）。求导的方式就是前后做差，如下：

$$y_i = (x_i, x_{i+2} - x_{i-2}, (x_{i+3} - x_{i-1}) - (x_{i+1} - x_{i-3}))$$

(2) 模糊 MFCC 序列：Black-box Attacks 中 MFCC 的序列是固定的，然后去模糊输入序列。在 White-box Attacks 中 MFCC 也是“模糊”的，其原理在于不同的人发同一个音的方式是不同的，那么他发这个音的方式只是这个音所有发音方式的其中一种。有一个目标字符串，从中抽取去目标音素序列，一个音素对应一个 HMM 状态，一个 HMM 状态对应一个 GMM 模型（由多个高斯分布函数组成，代表着一个音素的不同发音方式），我们从中随机挑选一个高斯分布作为我们的目标分布，即可将 MFCC 给“模糊”化了；

(3) 缩短音素的发音时长：GMM-HMM 模型识别一个音素最短只需要 4 帧即可，但这么短的时间（大约为 0.5s）对人来说识别一个音素是比较困难的，因此作者通过让音素发音尽可能地短来增强指令的隐藏性；

(4) 优化目标：（作者这边的公式有问题）

假设我们确定了一个目标 GMM 序列，我们希望通过梯度下降算法让 MFCC 分类为该序列的概率最大（Maximize the likelihood）：

$$\prod_i \exp\left\{\sum_{j=1}^{39} \frac{\alpha_i^j - (y_i^j - \mu_i^j)^2}{\delta_i^j}\right\}$$

其中， $y$  为 39 维 MFCC， $\alpha$  为混合高斯中的权重系数， $\mu$  为均值向量， $\delta$  为方差向量， $i$  为帧的序号， $j$  为维度的序号。Maximize the log likelihood：

$$\log \left[ \prod_i \exp\left\{\sum_{j=1}^{39} \frac{\alpha_i^j - (y_i^j - \mu_i^j)^2}{\delta_i^j}\right\} \right] = \sum_i \sum_j \frac{\alpha_i^j - (y_i^j - \mu_i^j)^2}{\delta_i^j}$$

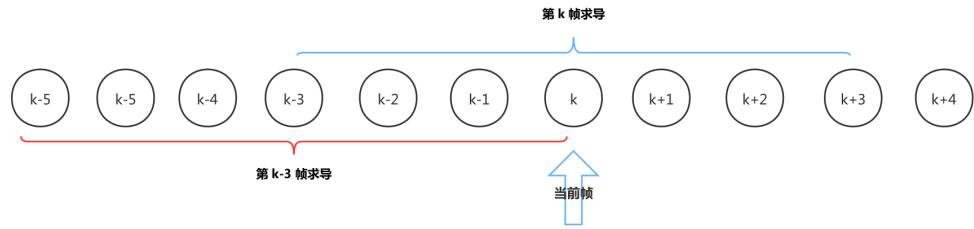
等同于最小化负的 log 似然：

$$\sum_i \sum_j \frac{-\alpha_i^j + (y_i^j - \mu_i^j)^2}{\delta_i^j}$$

这是一个非线性的最小二乘问题（作者在文章中称  $y = MFCC(x)$  是线性关系，但我认为  $x$  和  $y$  之间是非线性的，因为在求能量谱时存在平方）。

(5) 算法简述：（作者的描述实在是无法理解，在没有代码参考的情况下，以下算法简述含有大量的个人猜测成分，请阅读文章后再看）

- 贪婪算法：算法每一次都往前走一帧，生成最优的 160 个采样点（帧移大小），使得当前的最小二乘的累和最小；（对应伪代码第 10 行开始的代码块）
- 每一个 HMM 状态对应着多个高斯分布，作者期望的是逼近其中某一个高斯分布使得最小二乘最小化即可；（对应伪代码第 14 行开始的代码块）
- LSTDERIV：根据前  $k - 1$  帧的  $MFCC_{13}$ ，期望拟合的前  $k - 1$  帧的  $MFCC_{39}$ ，和当前期望拟合的第  $k$  帧的  $MFCC_{39}$ ，求解最小二乘优化问题，得到当前帧局部最优的  $MFCC_{13}$  和  $MFCC_{39}$ 。为什么是求解一个当前局部最优的解呢？1. 采用的是贪婪算法，每看一帧便确定一个局部最优的  $MFCC$ ；2. 每一步实际并没有求得一个时序序列使得其  $MFCC$  满足当前局部最优，看函数的参数，我们的  $f$  是从时序序列里面求出来的  $MFCC$  特征；3. 每一步的最小 likelihood 值受到前后 3 帧  $MFCC$  的影响（查看前面求导公式），同样当前帧的  $MFCC$  会影响前 3 帧的 likelihood，见下图；（我不是很理解文章里面的  $k + 6$  怎么来的）（对应伪代码第 15 行）



- GRADDESC: 使用 Conjugate Gradient 算法, 生成 160 个采样点; (对应伪代码第21行)

```

1 变量定义:
2 s := 语音序列
3 f := 语音序列的 13维MFCC 特征
4 g := 期望的 39维MFCC 特征
5 h := 目标 HMM 状态序列
6
7 def main():
8     init variables as empty list: s,f,g
9     select HMM state sequence according to Target Command: h
10    for i in range(len(h)):
11        m_l_score = MAX_FLOAT # min LST score
12        t_mfcc = None # target MFCC, 13维
13        t_g = None # target expanded MFCC, 39维
14        for j in range(len(h[i].g_mean_list)) # h[i].g_mean_list is
15            the mean value of each Gaussian for h[i]
16            (l_score, tmp_mfcc, tmp_g) = LSTDERIV(f, g,
17            h[i].g_mean_list[j]) # tmp_mfcc is the optimal MFCC(13维) to
18            minimize l_score, and tmp_g is the expanded MFCC(39维) extracted
19            according to the optimal MFCC(13维).
20            if l_score < m_l_score:
21                m_l_score = l_score
22                t_g = tmp_g
23                t_mfcc = tmp_mfcc
24            g.append(t_g)
25        s = GRADDESC(s, t_mfcc) # 使用梯度下降算法生成新的语音序列 (新增
26        160个点)
27        f = MFCC(s)
28    return s

```

#### 4. ↗ Playing over the air:

- (1) 让音频在扬声器上更易播放: 在梯度下降过程中增加音频二阶导的惩罚项;
- (2) 预测音频在物理信道上的失真;
- (3) 通过实际的物理播放来调整目标 MFCC;

#### 5. Evaluation:

	Human Understanding	Machine Understanding
Normal	74% (230/310)	-
Obfuscated	0% (1/377)	82% (82/100)

## Defense

1. 使用**提醒用户的方式**进行防御；
2. 使用**要求用户确认的方式**进行防御；
3. 使用**低通滤波器的方式**进行防御；
4. 训练一个**对抗样本的分类器**进行防御；

## Shortcoming

1. 对于实验目标的指令选择上，不够充分，只选择了三个较短的指令进行尝试；
2. 算法的解释上面不清楚；
3. 攻击十分的耗时，文章中提到，在白盒攻击下生成一个好的对抗样本可能需要 30 个小时左右；
4. 白盒攻击中 “Playing over the air” 依赖于设备和周围环境，实际中我们并不能很好地模拟攻击场景下的噪声环境和设备；

## Links

- 论文链接：[Carlini, Nicholas, et al. "Hidden voice commands." 25th USENIX Security Symposium \(USENIX Security 16\). 2016.](#)
- 论文主页：[Hidden Voice Commands](#)
- ↗ 背景噪声数据集：[Crowd Sounds — Free Sounds at SoundBible.](#)

# DolphinAttack: Inaudible voice commands

## Contribution

1. 利用麦克风非线性的特征，使得高频信号采样后出现额外的低频分量；
2. 攻击的原理：**让正常语音隐藏到高频波段中，从而让其无法被听到**；

## Notes

1. 攻击**语音前端采集模块（麦克风）**的对抗攻击（或者说，更像是一种漏洞）。
2. **麦克风工作原理**: 语音信号是一种波，被麦克风捕获，麦克风把波的声压转换成电信号，再通过对电信号进行采样便可获得离散时间的波形文件（引自[李理的博客](#)）。这个过程中，LPC 模块会过滤掉超过 20kHz 的信号，ADC 模块负责采样信号：

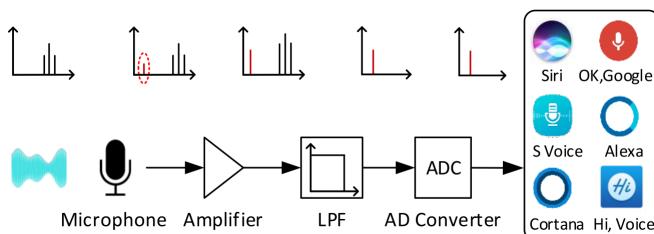
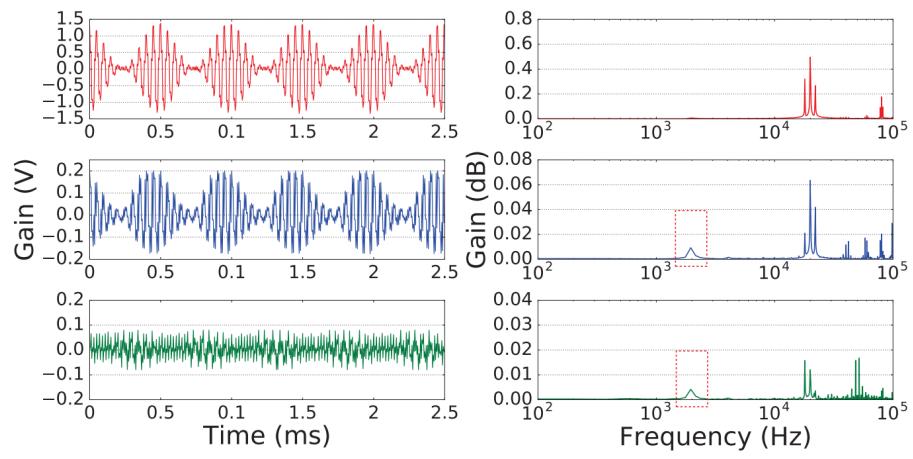


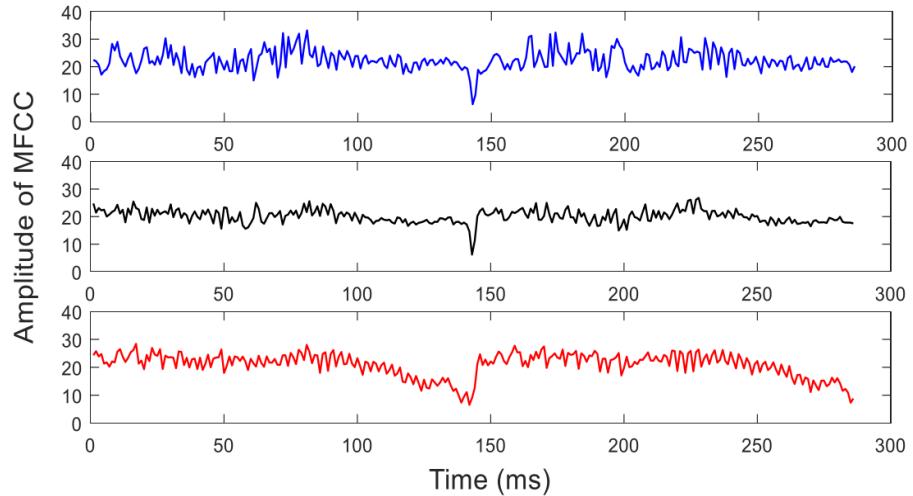
Figure 3: An illustration on the modulated tone traversing the signal pathway of a voice capture device in terms of FFT.

3. ↗ 麦克风的非线性特征，能够使得高频的语音信号被 **downconversion** (或可以理解为解调) 出低频的能量分量。  
(1) 非线性特征对**单频率语音**转录的影响，如下图：



其中第一行是原始语音的频谱，第二行是 MEMS 麦克风接收后的频谱，第三行是 ECM 麦克风接收后的频谱。这里采用的载波信号为 20kHz，语音信号为 2kHz。

(2) 非线性特征对正常说话的影响，如下图：



其中第一行是原始的 TTS 语音（发音为 Hey）的 MFCC 特征，第二行是正常播放-录音后的 MFCC 特征，第三行是经过“调制-解调”后的 MFCC 特征。计算得到他们的 MCD 距离分别为 3.1 和 7.6。（**我不是理解 Amplitude of MFCC 是什么意思，时频的 MFCC 特征应该像热力图才对？**）

4. Voice Command Generation，生成指令用于后续调制过程：

(1) Activation Commands Generation：使用 TTS (Text to Speech) 的方法或者是语音拼接的方法生成一个唤醒词。（可以看到在智能语音助手领域，说话人的识别并不是很有效的，可以被说话声音相似的人激活）；

(2) General Control Commands Generation：直接使用 TTS 生成；

5. Evaluation：这部分和信号的调制非常相关，不太易懂，直接简略地贴上结果

Manuf.	Model	OS/Ver.	SR System	Attacks		Modulation Parameters			Max Dist. (cm)	
				Recog.	Activ.	$f_c$ (kHz) & [Prime $f_c$ ] ‡	Depth	Recog.	Activ.	
Apple	iPhone 4s	iOS 9.3.5	Siri	✓	✓	20–42 [27.9]	≥ 9%	175	110	
Apple	iPhone 5s	iOS 10.0.2	Siri	✓	✓	24.1 26.2 27 29.3 [24.1]	100%	7.5	10	
Apple	iPhone SE	iOS 10.3.1	Siri	✓	✓	22–28 33 [22.6]	≥ 47%	30	25	
			Chrome	✓	N/A	22–26 28 [22.6]	≥ 37%	16	N/A	
Apple	iPhone SE †	iOS 10.3.2	Siri	✓	✓	21–29 31 33 [22.4]	≥ 43%	21	24	
Apple	iPhone 6s *	iOS 10.2.1	Siri	✓	✓	26 [26]	100%	4	12	
Apple	iPhone 6 Plus *	iOS 10.3.1	Siri	✗	✓	— [24]	—	—	2	
Apple	iPhone 7 Plus *	iOS 10.3.1	Siri	✓	✓	21 24–29 [25.3]	≥ 50%	18	12	
Apple	watch	watchOS 3.1	Siri	✓	✓	20–37 [22.3]	≥ 5%	111	164	
Apple	iPad mini 4	iOS 10.2.1	Siri	✓	✓	22–40 [28.8]	≥ 25%	91.6	50.5	
Apple	MacBook	macOS Sierra	Siri	✓	N/A	20–22 24–25 27–37 39 [22.8]	≥ 76%	31	N/A	
LG	Nexus 5X	Android 7.1.1	Google Now	✓	✓	30.7 [30.7]	100%	6	11	
Asus	Nexus 7	Android 6.0.1	Google Now	✓	✓	24–39 [24.1]	≥ 5%	88	87	
Samsung	Galaxy S6 edge	Android 6.0.1	S Voice	✓	✓	20–38 [28.4]	≥ 17%	36.1	56.2	
Huawei	Honor 7	Android 6.0	HiVoice	✓	✓	29–37 [29.5]	≥ 17%	13	14	
Lenovo	ThinkPad T440p	Windows 10	Cortana	✓	✓	23.4–29 [23.6]	≥ 35%	58	8	
Amazon	Echo *	5589	Alexa	✓	✓	20–21 23–31 33–34 [24]	≥ 20%	165	165	
Audi	Q3	N/A	N/A	✓	N/A	21–23 [22]	100%	10	N/A	

‡ Prime  $f_c$  is the carrier wave frequency that exhibits highest baseband amplitude after demodulation.

— No result

† Another iPhone SE with identical technical spec.

\* Experimented with the front/top microphones on devices.

## Shortcoming:

这篇文章的攻击非常有效，因为他利用的是麦克风的“漏洞”，所以几乎能够攻击全部平台设备。但它的缺点是需要一台超声波发生设备。

## Links

- 论文链接：[Roy, Nirupam, et al. "Inaudible voice commands: The long-range attack and defense." 15th {USENIX} Symposium on Networked Systems Design and Implementation \({NSDI} 18\). 2018.](#)
- Github 主页：[USSLab/DolphinAttack: Inaudible Voice Commands \(github.com\)](#)

## \* Did you hear that? Adversarial Examples Against Automatic Speech Recognition

## Contribution

- 针对关键词识别模型进行黑盒攻击；

## Notes

- 黑盒、有目标的针对语音关键词识别的对抗攻击算法。** 攻击的模型是 **Speech Commands Classification Model**，其中涉及的关键词有 `yes`、`no`、`up`、`down`、`left`、`right`、`on`、`off`、`stop` 和 `go`；

- 算法流程：

---

**Algorithm 1:** Generation of Targeted Adversarial Audio Files using Genetic Algorithm

---

**Inputs :**Original benign example  $x$   
target classification label  $t$

**Output :**Targeted attack example  $x_{adv}$

```
/* Initialize the population of candidate solutions */  
population ← InitializePopulation(x)  
iter_num = 0  
while iter_num < max_iter do  
    scores ← ComputeFitness(population)  
    xadv ← population [argmax(scores)]  
    if argmax f(xadv) = t then  
        | break // Attack succeeded, Stop early.  
    end  
    /* Compute selection probabilities. */  
    select_probs ← Softmax( $\frac{\text{scores}}{\text{Temp}}$ )  
    Next population ← {}  
    for i ← 1 to size do  
        | Select parent1 from population according to probabilities select_probs  
        | Select parent2 from population according to probabilities select_probs  
        | child = Crossover(parent1, parent2)  
        | Next population = Next population ∪ {child}  
    end  
    foreach child of Next population do Mutate(child)  
    population ← Next population  
    iter_num = iter_num + 1  
end  
return xadv
```

---

使用遗传算法生成对抗样本；

## Links

- 论文链接：[Alzantot, Moustafa, Bharathan Balaji, and Mani Srivastava. "Did you hear that? adversarial examples against automatic speech recognition." NIPS Machine Deception Workshop \(2017\).](#)
- 论文主页：[Adversarial Speech Commands | adversarial audio \(nesl.github.io\)](#)
- 论文代码：[nesl / adversarial audio \(github.com\)](#)

---

# Audio Adversarial Examples: Targeted Attacks on Speech-to-Text

---

## Contribution

- 白盒、有目标的、攻击端到端 DeepSpeech 模型 (CTC) 的对抗攻击算法；

## Notes

- 白盒、有目标的**对抗攻击算法。攻击的模型为 **DeepSpeech** 模型，攻击的指令为**任意长度**；
- 基础的**攻击方法，loss 函数（后半部分为 CTC-Loss）如下：

$$\begin{aligned} & \text{minimize } |\delta|_2^2 + c \cdot \ell(x + \delta, t) \\ & \text{such that } dB_x(\delta) \leq \tau \end{aligned}$$

作者提到使用 2 范数而不用无穷范数的原因是，无穷范数可能会导致不收敛的问题，难以训练。在参数的选择上，作者使用 Adam 算法，学习率为 5，迭代论述为 5000。在实验过程中，作者发现**目标指令越长**，需要添加越多的扰动来生成对抗样本；而如果**原始指令越长**，似乎更加容易生成对抗样本（这一点我的想法是，**如果原始指令越长，原始存在更多的音素和能**

量可以被梯度下降过程利用)。

3. ↗ 改进的攻击方法 (作者称: 这种改进的攻击方法只能在 DeepSpeech 使用 Greedy-Search 的情况下有效), loss 函数如下:

$$\begin{aligned} & \text{minimize } |\delta|_2^2 + \sum_i c_i \cdot L_i(x + \delta, \pi_i) \\ & \text{such that } dB_x(\delta) < \tau \end{aligned}$$

其中  $L_i(x, \pi_i) = \ell(f(x)^i, \pi_i)$  表示对于当前 alignment, 第 i 帧的 loss 值。作者这样修改 loss 函数的原因大致有两个:

- (1) 如果使用 CTC-Loss, 会添加不必要的修改。如果已经解码出 "ABCX", 目标指令为 "ABCD", 在使用 CTC-Loss 时, 梯度下降算法仍然会在 "A" 上添加扰动使得其变得更像 "A";
- (2) 不同的字符生成的难易程度是不同的, 所以把权重系数  $c$  移到了累加的里面。(这一点作者称是在 [Hidden Voice Command](#) 中发现的规律, 但其实只是在附录中给出了不同的单词可能需要的最短音素帧的数量是不同的, 并没有给出字符难易程度的结论; 并且这篇文章开源的代码中也没有给出这个改进的 loss 函数, 所以可以直接把这个  $c$  移出去作为单个参数进行调参);

训练的 **trick**: 首先用 CTC-Loss 生成一个对抗样本, 以这个对抗样本为参照固定 alignment (在 CTC 中, 可能有许多种 alignment, 作者通过这种方法来确定选择其中一种), 然后用改进的 loss 函数来生成; (这边, 我的想法是, 改进的攻击方法会使得对抗样本丧失其迁移性, 因为它只是恰好将特征拟合到模型的边界而已, 而没有去进一步地逼近泛化的特征上)

#### 4. Evaluation:

- (1) 作者在原始指令的基础上通过非常小 (约为 -30dB) 的扰动生成对抗样本, 并且在白盒的情况下最多可以在 1s 的语音中插入 50 个字符;
- (2) 对于 Non-Speech, 作者发现更难生成对抗样本;
- (3) 作者还对比了 FGSM 和 Iterative Optimization 两种生成对抗样本的算法, 发现在语音识别领域 FGSM 只适合生成 un-targeted 样本, 而不适合生成 targeted 样本 (或者说生成的效率很差, 几乎没有办法生成) ;
- (4) 作者发现这种方法生成的对抗样本是对噪声不鲁棒的;

## Links

- 论文链接: [Carlini, Nicholas, and David Wagner. "Audio adversarial examples: Targeted attacks on speech-to-text." 2018 IEEE Security and Privacy Workshops \(SPW\). IEEE, 2018.](#)
- 论文主页: [Audio Adversarial Examples \(carlini.com\)](#)
- 论文代码: [carlini/audio\\_adversarial\\_examples: Targeted Adversarial Examples on Speech-to-Text systems \(github.com\)](#)

## Adversarial Attacks Against Automatic Speech Recognition Systems via Psychoacoustic Hiding

# Contribution

1. 白盒、有目标的、针对API的、针对Kaldi DNN-HMM模型的对抗攻击算法；
2. 首次提出使用声学掩蔽效应；
3. 实验部分做的很全面，值得借鉴；

# Notes

1. **白盒、有目标的、只针对API的对抗攻击算法。** 攻击的模型为 Kaldi 的 **WSJ** 模型（或称为 recipe）；
2. 攻击方法整体架构如下：

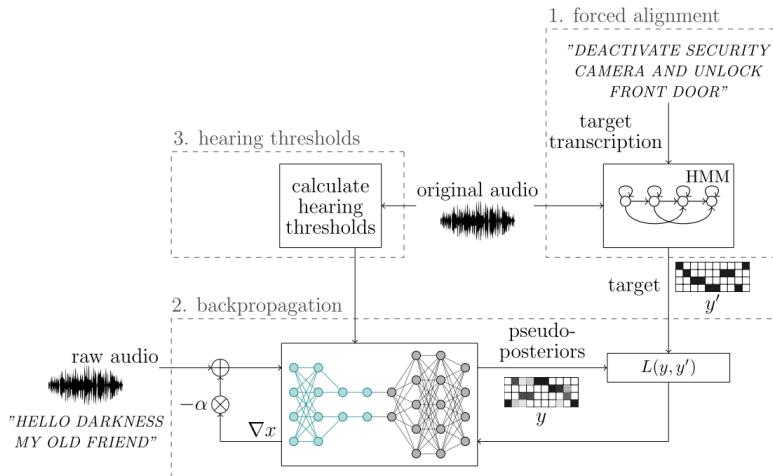


Fig. 3: The creation of adversarial examples can be divided into three components: (1) *forced alignment* to find an optimal target for the (2) backpropagation and the integration of (3) the hearing thresholds.

(1) **Forced Alignment**: 时序信号经过分帧后，每一帧都有对应音素的概率分布（Kaldi 中使用 [tri-phone](#)，直接说音素比较好理解）。作者根据目标指令和原始语音找到一个最好的对齐方式，目的是为了使得修改量最小；

(2) **Back-propagation in Feature Extraction**: 语音识别过程给网络的一般是 MFCC、Mel-log Filter Bank 等语音特征，把它简单地理解成是一张**二维热力图**，算法需要把梯度从这个特征回传到时域信号。（Kaldi 不像 tensorflow 那样直接就帮你把梯度计算好了，所以作者去推导了相关的梯度计算公式。不过，[这里作者只推导了对数能量谱的梯度，但是 WSJ 里面用的应该是 MFCC 才对。另外不清楚作者用的是优化器，还需要看一下 Kaldi 代码。](#)）。

(3) **Hearing Thresholds**: **心理声学掩蔽效应**，可以计算出音频各个时间、各个频率点的能量掩蔽值，只要修改量不超过这个值，那么人就不会察觉。下图展示了在 1kHz 处 60dB 信号的能量掩蔽曲线（黑色），绿色的为人耳能够感受到声音的最小能量（如 20kHz 的声音，至少要达到 70dB 我们才听得到）：

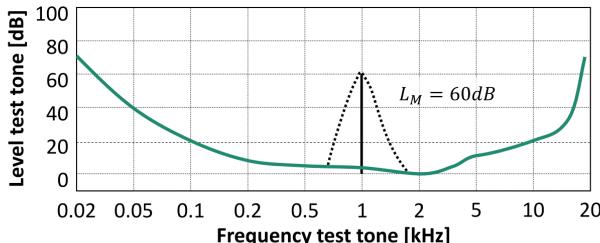


Fig. 2: Hearing threshold of test tone (dashed line) masked by a  $L_{CB} = 60 \text{ dB}$  tone at 1 kHz [64]. In green, the hearing threshold in quiet is shown.

作者计算样本的能量变化  $D$ ，并期望  $D$  在任何时间、频率点均小于掩蔽阈值  $H$ ，公式如下（[论文中的公式有个小错误，f 应该是 k](#)）：

$$D(t, f) \leq H(t, k), \quad \forall t, k,$$

with  $D(t, k) = 20 \cdot \log_{10} \frac{|S(t, k) - M(t, k)|}{\max_{t, k}(|S|)}$

变量  $\Phi$  度量能量变化 D 和 掩蔽阈值 H 之间的差值。如果 D 在任何点都不能超过 H，这样的限制条件过于苛刻，可能会导致无法生成对抗样本。故作者添加一个系数来放宽这个限制条件，公式如下：

$$\begin{aligned}\Phi &= \mathbf{H} - \mathbf{D} \\ \Phi^* &= \Phi + \lambda\end{aligned}$$

将  $\Phi$  小于 0 的值置为 0 并归一化到 0~1，公式如下：

$$\begin{aligned}\Phi^*(t, k) &= 0, \text{ if } \Phi^*(t, k) < 0 \\ \hat{\Phi}(t, k) &= \frac{\Phi^*(t, k) - \min_{t, k}(\Phi^*)}{\max_{t, k}(\Phi^*) - \min_{t, k}(\Phi^*)}, \quad \forall t, k\end{aligned}$$

只添加  $\Phi$  到梯度回传中时，作者发现差点意思。将 H 归一化到 0~1，公式如下：

$$\hat{H}(t, k) = \frac{H(t, k) - \min_{t, k}(H)}{\max_{t, k}(H) - \min_{t, k}(H)}$$

最后作者将这两个系数结合到 DFT 的梯度回传上（声学特征的计算这里不做解释了，推荐 [Mel Frequency Cepstral Coefficient \(MFCC\) tutorial](#)），公式如下：

$$\nabla X^*(t, k) = \nabla X(t, k) \cdot \hat{\Phi}(t, k) \cdot \hat{H}(t, k), \quad \forall t, k$$

我对这一块的理解：整体来看，作者想要使用“心理声学掩蔽效应”来生成更具隐藏性（或者说修改量小）的对抗样本，他认为“当前掩蔽值大”、并且“修改量远小于掩蔽值”的点可以添加更多的扰动，回传的梯度可以更大；而“当前掩蔽值小”、或者是“修改量已经接近掩蔽值”的点不应该再做更多的修改，回传的梯度趋近于 0。相对而言，我更喜欢“Imperceptible, Robust, and Targeted Adversarial Examples for Automatic Speech Recognition”的工作，他们直接将“心理掩蔽效应”添加到了 loss 函数中去，让模型自己来选择梯度的变化。

最后思考一个问题：这样用掩蔽阈值和 perturbation 的差值来度量真的是一种好的方法吗？可能不是，我们其实更希望的是去度量频率掩蔽曲线的变化有多大。举例来说，计算掩蔽阈值的时候首先得到的是 masker（可以理解为一个频率点，其能量是个极值点），我们在 masker 处增加能量来抬高 masker（完全可以做到增加的能量低于 masker 处的掩蔽值，因为这个点的掩蔽值等于 masker 的能量，这个值是很大的），这样人耳的听觉感受已经发生了改变。但是如果要这么来做，就要用可求解梯度的方法来实现“计算掩蔽值”的过程，过程实在是很复杂，这也可能是大家不这么做的原因（代价太大，做出来还不知道能不能收敛，效果好不好）。

### 3. ↗ Evaluation:

(1) 目标指令：

```

01: DO NOT BLAME YOU
02: THE COMMAND IS PLANTED
03: THE CAKE IS A LIE
04: THE COMMAND IS IN MY BRAIN
05: I'M AN INVADER COMING FOR YOU
06: WINTER IS COMING ZOMBIE COMING
07: IN MY RIGHT HAND
08: PRINCESS IN THE CASTLE
09: THEY DON'T BLAME YOU FIND A BOY
10: WELCOME TO THE JUNGLE ZOMBIE COMING WINTER IS COMING
11: THE CAKE IS A LIE DON'T BLAME YOU
12: I BELIEVE MOST PEOPLE ARE GOOD
13: THE HEAD THEY ARE STILL FIGHTING
14: I BELIEVE ALL PEOPLE ARE GOOD
15: THE SOUND OF SILENCE
16: IN THE MONEY CASTLE
17: WINTER IS COMING
18: DEACTIVATE SECURITY CAMERA AND UNLOCK FRONT DOOR
19: HE IS A MAN HE'S A GHOST
20: INTO YOUR FACE
21: TODAY I AM GOING NOWHERE

```

(2) 原始音频: Speech (从 WSJ 数据集中获取) + Music

(3) 评估指标: **WER** 和 **平均修改能量**, 前者越小越好, 后者越大越好;

(4) 分析 Hearing Threshold 和 Forced Alignment 的效果, 学习率 **0.05** (这个和其他的工作相差挺大, 猜测可能是因为有像 librosa 那样的 normalization), 迭代 500 轮:

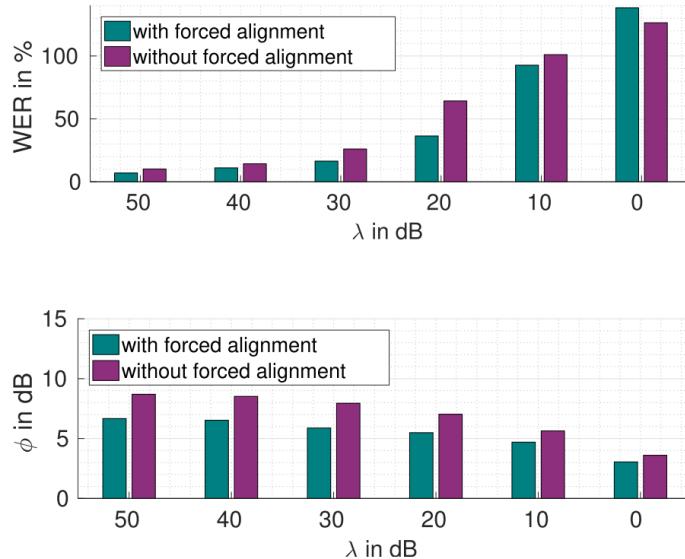


Fig. 6: Comparison of the algorithm with and without forced alignment, evaluated for different values of  $\lambda$ .

(5) 分析单位时间嵌入词数量的影响:

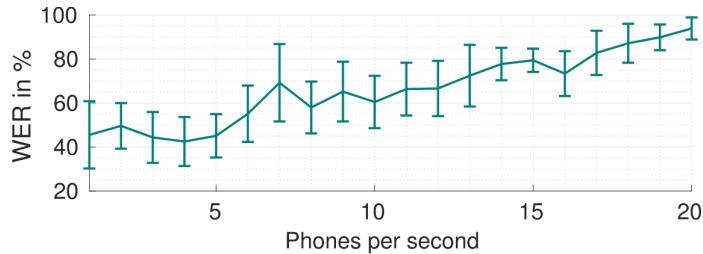
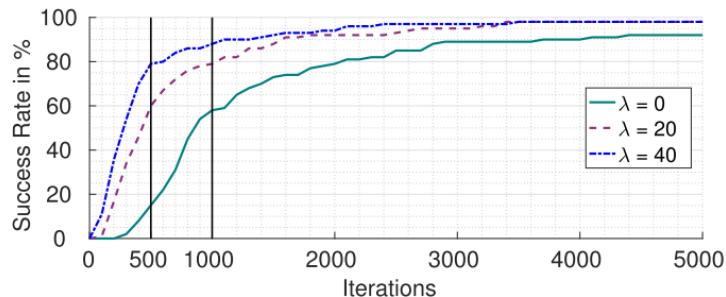
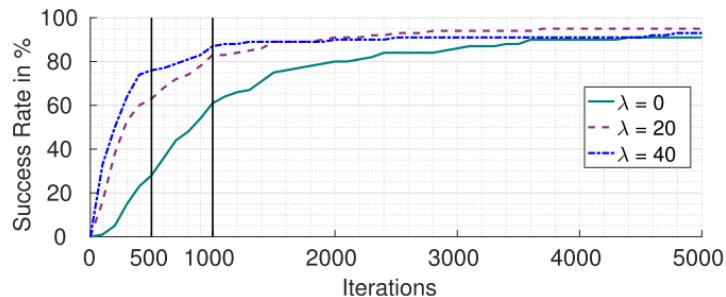


Fig. 7: Accuracy for different phone rates. To create the examples, 500 iterations of backpropagation and  $\lambda = 20$  are used. The vertical lines represent the variances.

(6) 分析迭代轮数的影响:



(a) Speech



(b) Music

Fig. 8: Success rate as a function of the number of iterations. The upper plot shows the result for speech audio samples and the bottom plot the results for music audio samples. Both sets were tested for different settings of  $\lambda$ .

(7) 和 CommandSong 进行对比，对比的指标为 SNR：

TABLE III: Comparison of SNR with *CommanderSong* [61], best result shown in bold print.

	None	40 dB	20 dB	0 dB	<i>CommanderSong</i> [61]
SNR	15.88	17.93	<b>21.76</b>	19.38	15.32

$$\text{SNR(dB)} = 10 \cdot \log_{10} \frac{P_x}{P_\sigma},$$

## Links

- 论文链接: [Schönherr, Lea, et al. "Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding." arXiv preprint arXiv:1808.05665 \(2018\).](#)
- 论文主页: [Adversarial Attacks \(adversarial-attacks.net\)](#)
- 论文代码: [rub-ksv/adversarialattacks: Adversarial Attacks \(github.com\)](#)

## \* Targeted adversarial examples for black box audio systems

# Contribution

1. 黑盒、有目标的攻击 DeepSpeech 的对抗攻击算法；
2. 结合遗传算法和梯度下降算法 (在语音上没人这么来做过，但是其实在图像识别上面这种黑盒攻击不是新鲜事了，所以在算法上面没有创新型，而且直接应用到语义领域，出现了 query 次数巨大的问题)；
3. 思路是值得借鉴的，黑盒攻击一定有比白盒攻击更加 interesting 的问题，但是不能照搬图像领域的办法；

## Notes

1. **黑盒、有目标的对抗攻击算法。** 攻击的模型为 **DeepSpeech** 模型，选择的方式是 **遗传算法和梯度下降算法的结合**；
2. 原始音频与目标指令：从 CommonVoice 测试集中挑选出前 100 个样本作为原始音频，目标指令都是 2 个词的指令，比较短；
3. 攻击场景：作者假设 DeepSpeech 模型是不可知探知的，但是知道模型最后的概率分布的输出，并且针对 Greedy Decoding 进行攻击（我的想法：这样的攻击场景其实是不常见的，所以这个工作可能指导意义不大，但是我们应该思考一下，如果 ASR 模型经过了 LM 模型的修饰，还能不能用黑盒探测的方法来生成对抗样本？如果能，代价又有多大？）；
4. 算法流程：

---

**Algorithm 1** Black box algorithm for generating adversarial audio sample

---

**Require:** Original benign input  $x$  Target phrase  $t$   
**Ensure:** Adversarial Audio Sample  $x'$

```
population ← [x] * populationSize
while iter < maxIters and Decode(best)! = t do
    scores ← -CTCLoss(population, t)
    best ← population[Argmax(scores)]

    if EditDistance(t, Decode(best)) > 2 then
        // phase 1 - do genetic algorithm
        while populationSize children have not been made do
            Select parent1 from topk(population) according to softmax(their score)
            Select parent2 from topk(population) according to softmax(their score)
            child ← Mutate(Crossover(parent1, parent2), p)
        end while
        newScores ← -CTCLoss(newPopulation, t)
        p ← MomentumUpdate(p, newScores, scores)

    else
        // phase 2 - do gradient estimation
        top-element ← top(population)
        grad-pop ← n copies of top-element, each mutated slightly at one index
        grad ← (-CTCLoss(grad-pop) - scores) / mutation-delta
        pop ← top-element + grad
    end if
end while
return best
```

---

(1) 当样本解码的字符串距离目标指令较大时，使用遗传算法生成对抗样本，遗传算法的评分函数使用 CTC-Loss，其变异概率  $p$  由函数 `MomentumUpdate` 进行更新；

$$p_{new} = \alpha \times p_{old} + \frac{\beta}{|currScore - prevScore|}$$

(2) 当样本解码的字符串距离目标指令较小时, 使用黑盒-梯度下降算法生成对抗样本, 对每个序列样本点 (花费巨大, 对于一个 16kHz 的 5s 语音, 每轮都要调用目标模型进行解码 80k 次) 都分别添加小的扰动, 根据 CTC-Loss 值的变化, 确定扰动的影响是正面的还是负面的、是重要的还是不重要的;

$$FD_x(x, \delta) = \begin{bmatrix} (g(x + \delta_1) - g(x))/\delta \\ \vdots \\ (g(x + \delta_n) - g(x))/\delta \end{bmatrix}$$

5. Evaluation:

- (1) 使用 100 条原始语音, 每个语音的目标指令是随机从 1000 个最常用的英语单词中抽取的 2 个单词, 每个对抗样本, 设置生成 3000 轮;
- (2) 使用 Success Rate 来评估成功率, 对抗样本的成功率为 35%; 使用 Cross Correlation Coefficient 来评估相似性, 对抗样本与原始语音的相似性为 94.6% (这里只看成功的对抗样本);

## Links

- 论文链接: [Taori, Rohan, et al. "Targeted adversarial examples for black box audio systems." 2019 IEEE Security and Privacy Workshops \(SPW\). IEEE, 2019.](#)
- 论文代码: [rtaori/Black-Box-Audio: Targeted Adversarial Examples for Black Box Audio Systems \(github.com\)](#)

# Robust Audio Adversarial Example for a Physical Attack

---

## Contribution

1. 引入脉冲响应;
2. 实现了较高的物理攻击成功率, 并且用了两组播放和接收设备;
3. 指令过短, 实验应该增加更多的物理环境;

## Notes

1. **白盒、有目标的针对物理环境的对抗攻击算法。** 攻击的模型为 DeepSpeech 模型, 选取的指令都比较短;
2. 在图像领域的对抗攻击算法中, [Athalye et al., 2018] 等人提出了用一个抽象函数  $t$  来模拟物理环境下打印和拍照在样本上带来的扰动。将这个抽象函数  $t$  结合到对抗样本的生成过程中去, 可以大大增强生成的对抗样本的鲁棒性;
3. 作者提出的方法, 关键点有三个:
  - (1) **带通滤波器。** 因为人的听觉频率范围是有限的, 听筒-扬声器在工作的时候很多会直接丢弃其他频率的能量, 所以作者设置了一个 1000~4000 范围的带通滤波器来减少修改量 (我的看法: 我觉得 4000 这个上界是比较靠谱的, 而 1000 这个下界可能并不合理, 因为语音中低频的能量是比较高的, 这部分的能量应该也是比较重要的; 而这种带通滤波器的方法是否真的能够减少修改量也是存在问题的, 因为依靠梯度下降算法, 可能你限制了它修改的频带范围, 需要的修改量可能是更多的)。形式化公式如下:

$$\underset{\mathbf{v}}{\operatorname{argmin}} \underset{f}{Loss}(MFCC(\tilde{\mathbf{x}}), \mathbf{l}) + \epsilon \|\mathbf{v}\|$$

where  $\tilde{\mathbf{x}} = \mathbf{x} + \underset{1000 \sim 4000 \text{Hz}}{BPF}(\mathbf{v})$

(2) **脉冲响应**。作者在生成对抗样本的过程中，添加脉冲响应的卷积来增强对抗样本对不同房间环境的鲁棒性。形式化公式如下：

$$\begin{aligned} & \underset{\mathbf{v}}{\operatorname{argmin}} \mathbb{E}_{h \sim \mathcal{H}} \left[ \underset{f}{\operatorname{Loss}}(MFCC(\tilde{\mathbf{x}}), \mathbf{l}) + \epsilon \|\mathbf{v}\| \right] \\ & \text{where } \bar{\mathbf{x}} = \underset{h}{\operatorname{Conv}} \left( \mathbf{x} + \underset{1000 \sim 4000 \text{Hz}}{\operatorname{BPF}}(\mathbf{v}) \right) \end{aligned}$$

(3) **高斯白噪声**。作者在生成对抗样本的过程中，添加高斯白噪声来增强对抗样本对背景白噪声的鲁棒性。形式化公式如下：

$$\begin{aligned} & \underset{\mathbf{v}}{\operatorname{argmin}} \mathbb{E}_{h \sim \mathcal{H}, \mathbf{w} \sim \mathcal{N}(0, \sigma^2)} \left[ \underset{f}{\operatorname{Loss}}(MFCC(\tilde{\mathbf{x}}), \mathbf{l}) + \epsilon \|\mathbf{v}\| \right] \\ & \text{where } \bar{\mathbf{x}} = \underset{h}{\operatorname{Conv}} \left( \mathbf{x} + \underset{1000 \sim 4000 \text{Hz}}{\operatorname{BPF}}(\mathbf{v}) \right) + \mathbf{w} \quad (7) \end{aligned}$$

#### 4. Evaluation:

(1) 作者其他的实现的细节与文章 ["Audio Adversarial Examples: Targeted Attacks on Speech-to-Text"](#) 是一样的，Adam 迭代器和 CTC-Loss 函数；

(2) **提到了一个比较有意思的攻击场景：FM radio**；

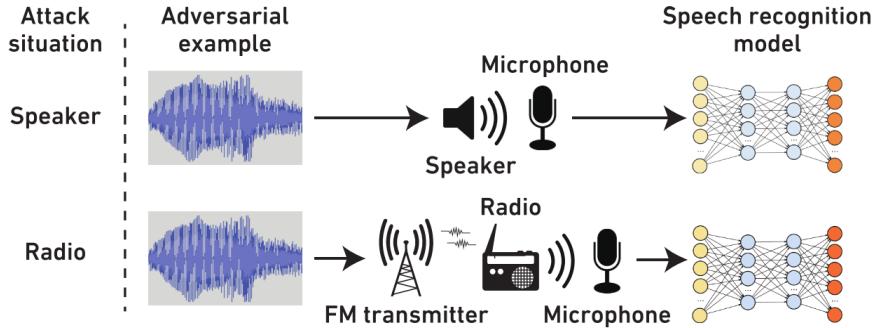


Figure 2: Two attack situations of the evaluation: speaker and radio. In the first situation, the adversarial examples were played and recorded by a speaker and a microphone. In the second situation, the adversarial examples were broadcasted using an FM radio.

思考一下：这种攻击场景多吗？未来的车辆控制可能会受到 FM 的攻击！

(3) 分析针对 API 的攻击：

	Input sample	Target phrase	SNR
(A)	Bach	hello world	9.3dB
(B)	Bach	open the door	5.3dB
(C)	Bach	ok google	0.2dB
(D)	Owl City	hello world	11.8dB
(E)	Owl City	open the door	13.4dB
(F)	Owl City	ok google	2.6dB

Table 1: Details of the generated audio adversarial examples, which showed 100% success by both the speaker and the radio and having the maximum value of SNR<sup>8</sup>.

(4) 分析针对物理的攻击：

		Input sample	Target phrase	SNR	Attack situation	Success rate	Edit dist.
(G)	Bach	hello world	11.9dB	Speaker Radio	60%	1.1	
					50%	1.3	
(H)	Bach	open the door	6.6dB	Speaker Radio	60%	1.8	
					60%	1.8	
(I)	Bach	ok google	4.2dB	Speaker Radio	80%	0.6	
					70%	0.9	
(J)	Owl City	hello world	12.2dB	Speaker Radio	70%	0.9	
					50%	1.5	
(K)	Owl City	open the door	14.6dB	Speaker Radio	90%	0.2	
					100%	0.0	
(L)	Owl City	ok google	8.7dB	Speaker Radio	90%	0.6	
					70%	0.9	

Table 2: Details of the generated audio adversarial examples, which showed at least 50% success by both the speaker and the radio and having the maximum value of SNR<sup>8</sup>.

每个样本尝试 10 次（测试次数太少了），统计成功率，并且发现只保证成功率高于 50% 的情况下，可以适当减少修改量。

## Links

- 论文链接: [Yakura, Hiromu, and Jun Sakuma. "Robust audio adversarial example for a physical attack." arXiv preprint arXiv:1810.11793 \(2018\).](#)
- 论文主页: [Robust Audio Adversarial Example for a Physical Attack \(yumetaro.info\)](#)
- 论文代码: [hiromu/robust\\_audio\\_ae: Robust Audio Adversarial Example for a Physical Attack \(github.com\)](#)
- 冲击响应:
 

[The reverb challenge: A common evaluation framework for dereverberation and recognition of reverberant speech \(ieee.org\)](#)

[A binaural room impulse response database for the evaluation of dereverberation algorithms \(ieee.org\)](#)

[Acoustical Sound Database in Real Environments for Sound Scene Understanding and Hands-Free Speech Recognition \(naist.jp\)](#)

[Evaluation of speech dereverberation algorithms using the MARDY database \(2006\) \(ist.psu.edu\)](#)

[Acoustic measurement data from the varechoic chamber \(nist.gov\)](#)

# Hear "No Evil", See "Kenansville": Efficient and Transferable Black-Box Attacks on Speech Recognition and Voice Identification Systems

## Contribution

- 一种黑盒的无目标对抗攻击

## Notes

## Links

- 论文链接: [Abdullah, Hadi, et al. "Hear" No Evil", See" Kenansville": Efficient and Transferable Black-Box Attacks on Speech Recognition and Voice Identification Systems." S&P \(2021\).](#)

# Imperceptible, Robust, and Targeted Adversarial Examples for Automatic Speech Recognition

## Contribution

1. 白盒、有目标的、针对端到端 LAS 模型的对抗攻击算法;
2. 心理掩蔽效应;
3. 模拟房间声学响应;

## Notes

1. **白盒、有目标的**对抗攻击算法。攻击的模型为 Lingvo 框架的 **LAS** 模型，攻击的指令选取了 1000 条**中等长度**的字符串;
2. 论文方法的关键点在于两方面，一是使用了**心理掩蔽效应**来提高对抗样本的隐蔽性，另一是**模拟房间声学响应**来提高对抗样本的鲁棒性;
3. **心理掩蔽效应**，(简单来讲)就是能量大的声音可以频闭能量小的声音，主要分为**时间掩蔽**和**频率掩蔽**。与 "[Adversarial Attacks Against Automatic Speech Recognition Systems via Psychoacoustic Hiding](#)" 相同，作者也用频率掩蔽效应。添加心理掩蔽效应后的 loss 函数：

$$\ell(x, \delta, y) = \ell_{net}(f(x + \delta), y) + \alpha \cdot \ell_\theta(x, \delta)$$

$$\ell_\theta(x, \delta) = \frac{1}{\lfloor \frac{N}{2} \rfloor + 1} \sum_{k=0}^{\lfloor \frac{N}{2} \rfloor} \max \{ \bar{p}_\delta(k) - \theta_x(k), 0 \}$$

前面部分保证**样本的成功率**，后面部分保证**样本的隐蔽性**，**alpha** 控制两者的权重。作者生成对抗样本的时候有一个 **trick** (因为作者把两个放在一起时发现很难生成对抗样本)：(**Stage-1**)先根据前面的 loss 函数生成一轮对抗样本，(**Stage-2**)然后根据后面的 loss 函数生成一轮对抗样本，如果 stage-2 迭代 **20** 轮后，成功生成了对抗样本，那就把 alpha 增大一些 (**说明可以增加一些隐蔽性**)；如果 stage-2 迭代 **50** 轮，都没能生成对抗样本，那就把 alpha 减小一些 (**说明需要牺牲一些隐蔽性**)。具体的迭代生成算法如下：

---

**Algorithm 1** Optimization with Masking Threshold

---

**Input:** audio waveform  $x$ , target phrase  $y$ , ASR system  $f(\cdot)$ , perturbation  $\delta$ , loss function  $\ell(x, \delta, y)$ , hyperparameters  $\epsilon$  and  $\alpha$ , learning rate in the first stage  $lr_1$  and second stage  $lr_2$ , number of iterations in the first stage  $T_1$  and second stage  $T_2$ .

# Stage 1: minimize  $\|\delta\|$

Initialize  $\delta = 0$ ,  $\epsilon = 2000$  and  $\alpha = 0$ .

**for**  $i = 0$  to  $T_1 - 1$  **do**

$\delta \leftarrow \delta - lr_1 \cdot \text{sign}(\nabla_\delta \ell(x, \delta, y))$

Clip  $\|\delta\| \leq \epsilon$

**if**  $i \% 10 = 0$  and  $f(x + \delta) = y$  **then**

**if**  $\epsilon > \max(\|\delta\|)$  **then**

$\epsilon \leftarrow \max(\|\delta\|)$

**end if**

$\epsilon \leftarrow 0.8 \cdot \epsilon$

**end if**

**end for**

# Stage 2: minimize the perceptibility

Reassign  $\alpha = 0.05$

**for**  $i = 0$  to  $T_2 - 1$  **do**

$\delta \leftarrow \delta - lr_2 \cdot \nabla_\delta \ell(x, \delta, y)$

**if**  $i \% 20 = 0$  and  $f(x + \delta) = y$  **then**

$\alpha \leftarrow 1.2 \cdot \alpha$

**end if**

**if**  $i \% 50 = 0$  and  $f(x + \delta) \neq y$  **then**

$\alpha \leftarrow 0.8 \cdot \alpha$

**end if**

**end for**

**Output:** adversarial example  $x' = x + \delta$

---

4. 模拟房间声学响应，简单来说，当固定了房间的参数和你设备的参数，你可以将整个物理信道用一个函数  $t(x)$  来建模。添加房间声学响应后的 loss 函数：

$$\text{minimize } \ell(x, \delta, y) = \mathbb{E}_{t \sim \mathcal{T}} [\ell_{\text{net}}(f(t(x + \delta)), y)]$$

such that  $\|\delta\| < \epsilon$ .

训练的 **trick**：( **Stage-1** ) 使用 `lr_1=50` 迭代 2000 轮保证在其中 **1 个房间声学响应下**能够生成对抗样本，( **Stage-2** ) 然后使用 `lr_2=5` 迭代 5000 轮来保证在另外随机采样的 **10 个房间声学响应下**都能够生成对抗样本 (这个期间不再减小 **perturbation** 的上限)。

5. 结合心理掩蔽效应和模型房间声学响应。结合后的 loss 函数：

$$\ell(x, \delta, y) = \mathbb{E}_{t \sim \mathcal{T}} [\ell_{\text{net}}(f(t(x + \delta)), y) + \alpha \cdot \ell_\theta(x, \delta)]$$

训练的 **trick**：( 在 4 的对抗样本基础上 ) 结合整个 loss 函数来生成具有隐藏性的对抗样本，和 3 中的分两步生成不同。

6. Evaluation:

(1) 直接攻击 API: 成功率 100%，添加的扰动几乎无法察觉；( 可以直接上主页听一下效果，直接攻击的效果还是非常不错的，不过意义不是很大，打比赛的时候可能效果会很好 )

(2) 转录后攻击 API: 大概的成功率在 60% 左右

Input	Clean	Robust ( $\Delta = 300$ )	Robust ( $\Delta = 400$ )	Imperceptible & Robust
<b>Accuracy (%)</b>	31.37	62.96	64.64	49.65
<b>WER (%)</b>	15.42	14.45	13.83	22.98

(3) 隐藏性，隐藏性没有采用常用的 SNR 来度量，而是直接采用问卷调查的形式，作者的问卷调查的问题分别为：

- 音频是否清晰；
- 分辨两个音频哪一个是原始音频；
- 判断两个音频是否相同；

## Links

- 论文链接：[Qin, Yao, et al. "Imperceptible, robust, and targeted adversarial examples for automatic speech recognition." International Conference on Machine Learning. PMLR, 2019.](#)
- 论文主页：[Imperceptible, Robust and Targeted Adversarial Examples for Automatic Speech Recognition \(ucsd.edu\)](http://imperceptible.csail.mit.edu/)

- 论文代码: [cleverhans/examples/adversarial\\_asr at master · tensorflow/cleverhans \(github.com\)](https://github.com/cleverhans/examples/tree/master/adversarial_asr)

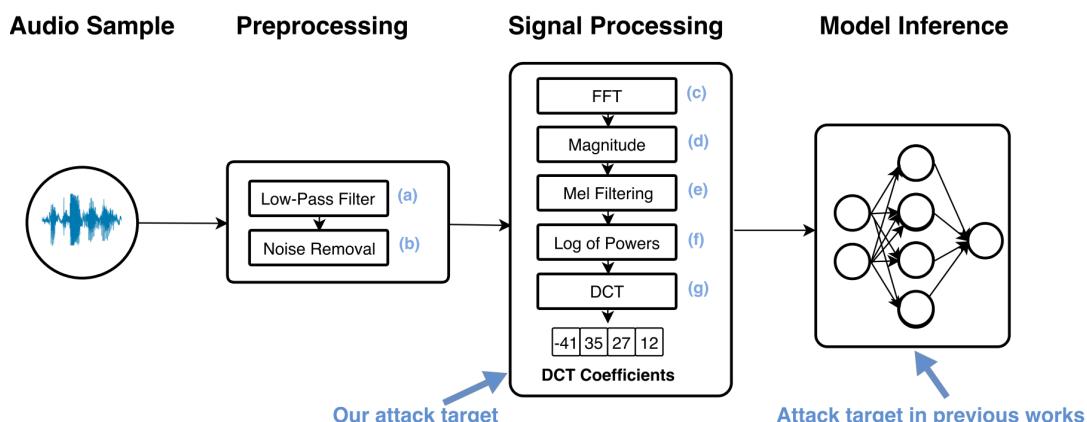
## \* Practical Hidden Voice Attacks against Speech and Speaker Recognition Systems

### Contribution

1. Hidden Voice Commands 的延续, 将正常语音模糊化, 攻击特征提取模块;
2. 从这篇文章可以看到语音对抗攻击在物理, 黑盒环境下的重重困难;

### Notes

1. 黑盒, (自称) 对不同硬件设备鲁棒的, 攻击特征提取模块的语音对抗攻击.
2. 作者指出前面工作的一些问题:
  - 需要借助白盒知识进行攻击;
  - 对不同的硬件 (麦克风和扬声器) 有不同的效果;
3. 这篇文章是对 Hidden Voice Commands Black-box 攻击的扩展, 首先看一下语音识别的各个模块:



大多数的白盒攻击关注的是深度神经网络模块, 对这个模块运用梯度下降算法生成对抗样本, 而这里作者则是考虑**攻击 Signal Processing 模块**, 即攻击特征提取模块;

思考一下: 文章指出, 使用低维有限的语音特征训练神经网络为 (添加扰动)生成对抗样本 提供了可能, 那么高采样率训练的模型一定不容易受到对抗攻击吗?

4. 攻击距离约为 34 cm;
5. ↗ 作者的攻击方法很简单, 利用特征提取过程中的“多对一”问题, 结合下面这幅图来看:

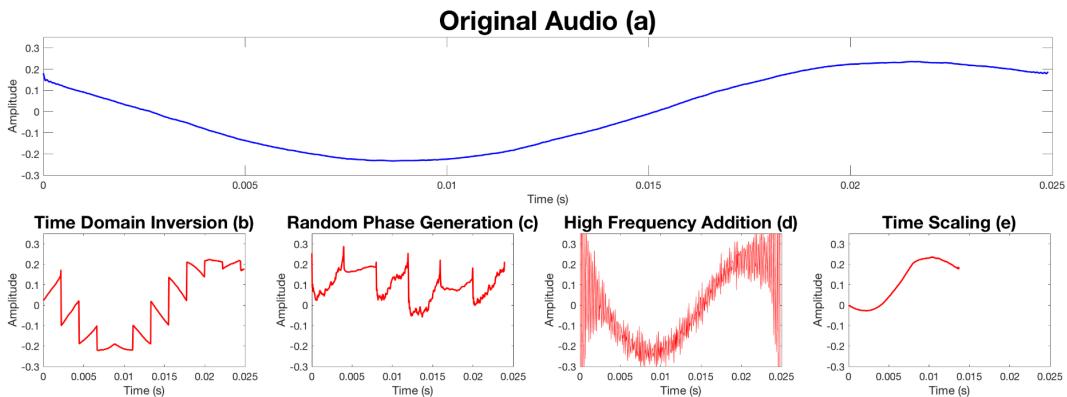


Fig. 2: The above figure shows the different perturbation techniques applied to the original signal (a). Signals (b) to (e) show the result of applying the perturbation schemes to (a).

- Time Domain Inversion (TDI): 在时域上, 在每个小窗内前后翻转信号. 攻击的是特征提取过程中的分帧, 如果窗口大小刚好重合的话, 那么翻转后的信号计算FFT是不变的 (这里作者没有考虑到帧移的问题, 作者在实验部分称, TDI 的窗口大小并不需要和特征提取算法的窗口大小一致). 而翻转后的语音因为不连续的问题, 人耳听起来更像噪声;
- Random Phase Generation (RPG): 随机挑选一些实数域和虚数域的值来代替原频谱的值. 攻击的是频谱求能量谱的环节( 平方求能量 ), 如下公式:

$$magnitude_{original} = Y = \sqrt{a_0^2 + b_0^2 i} = \sqrt{a_n^2 + b_n^2 i}$$

这里可以看到, 一个能量谱可以对应多个不同的频谱, 所以就可以去替换  $a_0$  和  $b_0$  的值达到模糊的效果; (进一步思考后, 我认为这种变换是不靠谱的, 因为语音信号是实信号, 如果用复数域来表示的话, 在存储时应该会丢弃虚数部分, 那么这样变换后的频谱就发生了改变)

- High Frequency Addition (HFA): 高频掩蔽低频 ( 心理声学掩蔽效应 ), 而高频的声音会在特征提取之前就被过滤;
- Time Scaling (TS): 加速语音, 语音加速之后人耳不容易听清 ( 作者这里没有考虑应用 TS 会导致语音实际计算出的 MFCC 值会发生改变. 这种改变, 不只是时间上的不对应; 想象一下, 语音加快了, 那么声音的频率势必会上升啊 ). 作者实际在做的时候, 直接就探测黑盒能够承受的最快加快速度;
- Improved Attack Method: 不同的词可以修改的程度是不一样的, 所以可以让不同词经过不同的参数生成对抗样本;
- 从后面的实验来看, 作者在物理攻击下其实只用了 TDI 和 TS 两种修改方法, 故我挺好奇该攻击方法的实用性的;

## 6. Evaluation

(1) 不同的攻击场景: Over-the-Line 和 Over-the-Air, 直接将样本交给 API 或者是经过物理信道以后再交给 API. 这里作者提到了一个点: 在 Over-the-Air 下成功的样本, 并不一定能够在 Over-the-Line 情况下成功, 作者在实际攻击的过程中, 还是拿的 Over-the-Line 的样本进行 Over-the-Air 的测试;

(2) 攻击的模型: 测试的模型还是比较全面的;

Voice Processing System	Model Type	Task	Feature Extraction	Phrase ID	Online/Local
Azure Verification API [4]	Unknown	Identification	Unknown	D	Online
Azure Attestation API [3]	Unknown	Identification	Unknown	A,B,C	Online
Bing Speech API [6]	Unknown	Transcription	Unknown	E,F,G,H	Online
Google Client Speech API [7]	Unknown	Transcription	Unknown	E,F,G,H	Online
Houndify Speech API [8]	Unknown	Transcription	Unknown	E,F,G,H	Online
IBM Speech API [9]	Unknown	Transcription	Unknown	E,F,G,H	Online
Mozilla DeepSpeech [12]	RNN	Transcription	End-to-End	E,F,G,H	Local
Intel Neon DeepSpeech [10]	RNN	Transcription	MFSC	I,J,K,L	Local
Kaldi [58]	HMM-GMM	Transcription	MFCC	E,F,G,H	Local
Kaldi-DNN [11]	DNN	Transcription	MFCC	E,F,G,H	Local
Sphinx [41]	HMM-GMM	Transcription	MFCC	E,F	Local
Wit.ai Speech API [14]	Unknown	Transcription	Unknown	E,F,G,H	Online

(3) 指令: 在 ASR 的测试上, 作者使用了 4 条指令 ( 其中 I, J, K, L 只用在 Intel Neon DeepSpeech 上 ), 长度为 2~4 个单词;

ID	Model	Phrase	Success Rate (%)
A	Identification	When suitably lighted	100
B		Don't ask me to carry an oily rag like that	100
C		What would it look like	100
D		My name is unknown to you	100
E	ASR	Pay money	100
F		Run browser	100
G		Open the door	100
H		Turn on the computer	100
I		Spinning indeed	100
J		Very Well	100
K		The university	100
L		Now to bed boy	100

☆ 测试的指令少, 并且短, 是黑盒, 物理的语言对抗攻击存在的通病. 这里作者提到了一个点: 他没有使用如 Ok Google 这样的唤醒词作为目标指令, 因为唤醒词通常使用的模型, 训练的方法都会和正常的语音识别有所不同, 并且不同平台之间唤醒词的鲁棒性也存在差异, 因此对这些词进行攻击是存在偏差的;

思考一下:

我们如何增强黑盒, 物理语音对抗攻击的攻击能力( 鲁棒性, 迁移性, 隐藏性 )呢?

对于攻击的偏差, 不仅仅存在于唤醒词部分. 每个目标模型的训练过程都不可能相同, 可能存在 "不同模型被攻击的成功率不同", "相同模型对不同词攻击的成功率不同" 等偏差, 这些偏差如何来衡量?

黑盒, 物理语音对抗攻击缺乏一个衡量他们攻击能力的指标.

(4) 实验设备: Audioengine A5 speaker + Behringer microphone, iMac speaker + Motorola Nexus 6; ( 作者只用了两套设备进行测试, 但是直接下了一个设备鲁棒的结论, 这个我是不太赞同的 )

(5) 实验结果:

- Over-the-Line: 如"指令"图中, 作者测试为 100% 成功率, 但是未给出相关的参数;
- Over-the-Air: 如下图

Models	Over -the- Air	Attack Type	Min TDI Size (ms)	Max TS Factor (%)
Bing Speech API	4/4	TDI	3.36	-
Google Speech API	4/4	TDI	1.47	-
Houndify	3/4	TDI	1.00	-
IBM Speech API	3/4	TDI	2.42	-
Mozilla DeepSpeech	15/15	TDI	2.00	-
Kaldi-DNN	4/4	TDI+TS	1.00	300
Wit.ai Speech API	2/4	TDI	1.94	-

- 对参数的解释: -- 攻击中使用的窗口大小( 影响 TDI 和 RPG )越小, 音频的修改量越大; -- 最小的窗口大小为 1ms; -- 作者在实验中使用 150% 的加速, 1ms 左右的窗口大小以及 8000Hz 左右的高频扰动;
- 选择音质最差的音频: 作者称, 窗口越小, 或高频扰动约多, 或加速越快, 使得音频修改量越大; 在这个前提下, 作者从 2W 个样本中挑选出 10 个样本, 然后通过人耳来听( 隐蔽性的度量问题, 作者并不考虑修改量的问题 );
- 样本: 在论文主页可以看到 4 个他们展示的样本( 实在是太少了 ), 第四个样本的指令是 "call to mom", 我认为听觉还算比较明显, 其他几条指令听起来更像是噪音;

## Links

- 论文链接: [Abdullah, Hadi, et al. "Practical hidden voice attacks against speech and speaker recognition systems." NDSS \(2019\).](#)
- 论文主页: [pratical hidden voice](#)

## \* Adversarial Black-Box Attacks on Automatic Speech Recognition Systems using Multi-Objective Evolutionary Optimization

### Contribution

- 设想在黑盒攻击的过程中, 引入两个评价指标, 以解决 声学相似性和文本相似性 这两个 trade off 的问题;

### Notes

1. 黑盒, 有目标 / 无目标的对抗攻击;
2. 方法:

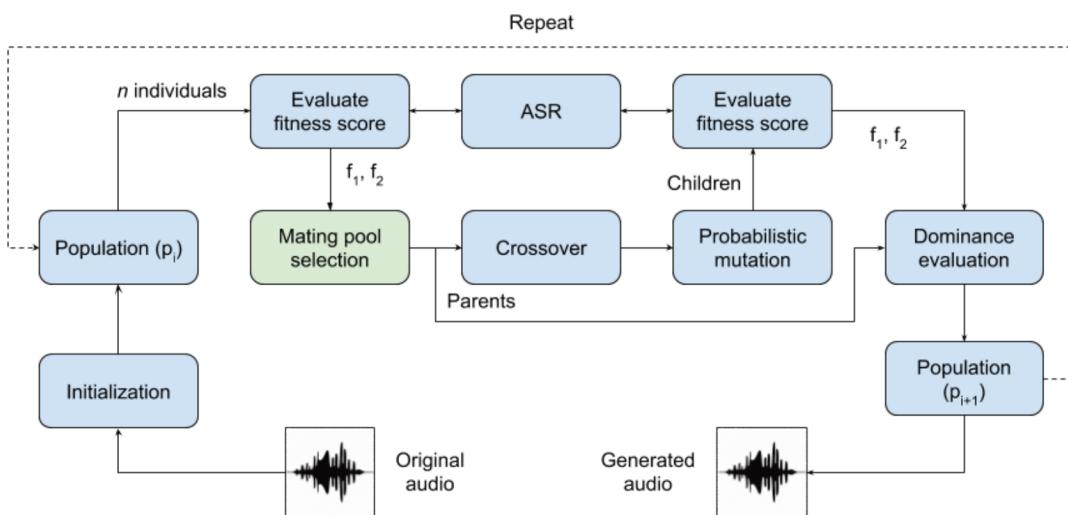


Figure 2: System Architecture

整体的方法是遗传算法, 和前面用的遗传算法的不同之处在于作者使用了两个目标作为 fitness 的度量: (1) 声学的相似性 - MFCC距离; (2) 文本的相似性 - 文字编辑距离;

3. 需要大量 query 是当前黑盒攻击的痛点;

## Links

- 论文链接: [Khare, Shreya, Rahul Aralikatte, and Senthil Mani. "Adversarial black-box attacks on automatic speech recognition systems using multi-objective evolutionary optimization." arXiv preprint arXiv:1811.01312 \(2018\).](#)
- 论文主页: <https://shreyakhare.github.io/audio-adversarial/>

# Who is Real Bob? Adversarial Attacks on Speaker Recognition Systems

## Contribution

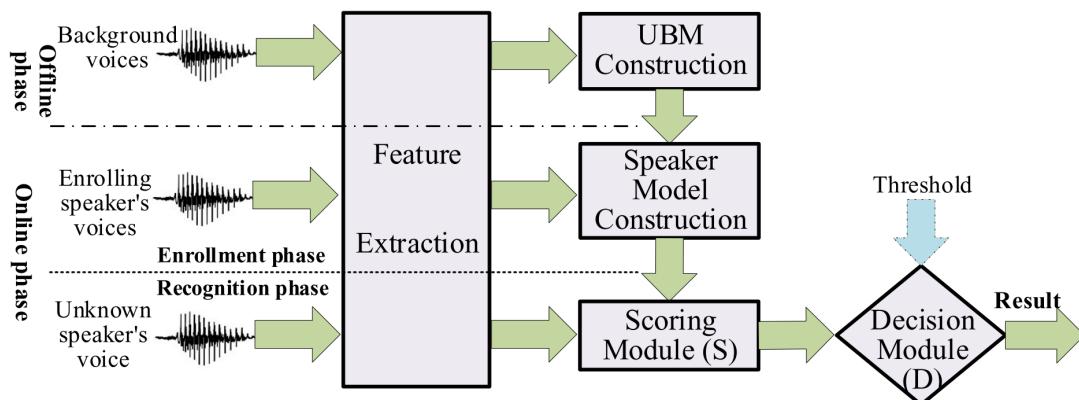
1. 实现了对说话人识别的对抗攻击, 将说话人识别中的判别阈值很好地加入到对抗样本的生成过程中;
2. 针对黑盒, 实现了有目标/无目标地攻击攻击;
3. 添加的扰动非常的少, 实现的效果可观;
4. 进行了大量的实验;
5. 这个攻击的一个缺点是需要依赖 API 输出相应的标签概率;

## Notes

1. 黑盒的, 物理/API的, 有/无目标的说话人识别对抗攻击;

2. 说话人识别模型:

- (1) 经典的 UBM-GMM 模型



- (2) 说话人识别处理的任务:

- Open-set Identification (OSI): 识别为哪一个说话人或返回空;
- Close-set Identification (CSI): 识别为其中一个说话人 (不会返回空);
- Speaker Verification (SV): 验证是否是目标说话人;

- (3) 是否依赖文本: 从后面的实验来看, 依赖文本的语音识别系统可能具有更好的安全性;

- 依赖文本;
- 不依赖文本;

- (4) 模型结构:

- ivector-PLDA;
- GMM-UBM;
- xvector-PLDA;

3. 威胁场景:

- 攻击黑盒模型;
- 黑盒模型需要输出识别的结果和得分, 如果没有得分的话, 就使用迁移攻击 (如在 Microsoft Azure 上);
- 作者总共考虑 16 中可能的攻击组合:

$$\left\{ \begin{array}{l} \left( \begin{array}{c} \text{targeted} \\ \text{untargeted} \end{array} \right) \times \left( \begin{array}{c} \text{intra-gender} \\ \text{inter-gender} \end{array} \right) \times \text{API} \times \left( \begin{array}{c} \text{OSI} \\ \text{CSI} \\ \text{SV} \end{array} \right) \times \text{D.&S.} \\ + \\ \text{targeted} \times \left( \begin{array}{c} \text{OSI} \\ \text{CSI} \\ \text{SV} \end{array} \right) \times \text{API} \times \text{decision-only} \\ + \\ \text{targeted} \times \left( \begin{array}{c} \text{OSI} \\ \text{CSI} \\ \text{SV} \end{array} \right) \times \text{over-the-air} \times \text{D.&S.} \\ + \\ \text{targeted} \times \text{OSI} \times \text{over-the-air} \times \text{decision-only} \end{array} \right\}$$

#### 4. ↗ 算法:

(1) 迭代算法的选择: NES 算法是梯度估计算法 (**梯度估计算法的特点是需要知道目标标签的概率**) 中最佳的, PSO 算法是遗传算法中最佳的, 这里作者选用的是 **NES** 算法;

(2) 形式化问题:

$$\begin{aligned} & \operatorname{argmin}_{\delta} f(x + \delta) \\ & \text{such that } \|x + \delta, x\|_{\infty} < \epsilon \text{ and } x + \delta \in [-1, 1]^n \end{aligned}$$

在一定扰动范围内, 是的目标 loss 函数最小化;

(3) Attack on OSI:

- Targeted Attack:

$$f(x) = \max \left\{ (\max\{\theta, \max_{i \in G \setminus \{t\}} [S(x)]_i\} - [S(x)]_t), -\kappa \right\}$$

最大化目标概率, 是的目标概率超过阈值  $\theta$ , 添加一个系数  $k$  增强样本的鲁棒性,  $k$  越大越鲁棒.

- Untargeted Attack: (文章的公式可能有点小错误)

$$f(x) = \max \{(\theta - \max_{i \in G \setminus \{t\}} [S(x)]_i), -k\}$$

这一块作者并没有考虑 **reject** 也是无目标攻击的一种, 故会有上面这个式子. 如果转换为 **平常我们遇到的无目标攻击 (考虑 reject)**, 公式形式如下:

$$f(x) = \max \{[S(x)]_t - \theta, -k\}$$

即我们让标签小于  $\theta$  就完成了无目标攻击, 但如果这样的话, 便无法和下面的  **$\theta$  估计算法** 相结合, 因为我们这里需要对  $\theta$  向下估值, 而非向上估值.

- $\theta$  估计算法:

---

#### Algorithm 1 Threshold Estimation Algorithm

---

**Input:** The target OSI system with scoring  $S$  and decision  $D$  modules  
**Output:** An arbitrary voice  $x$  such that  $D(x) = \text{reject}$

**Output:** Estimated threshold  $\hat{\theta}$

```

1:  $\hat{\theta} \leftarrow \max_{i \in G} [S(x)]_i$ ; ▷ initial threshold
2:  $\Delta \leftarrow |\frac{\hat{\theta}}{10}|$ ; ▷ the search step
3:  $\acute{x} \leftarrow x$ ;
4: while True do
5:    $\hat{\theta} \leftarrow \hat{\theta} + \Delta$ ;
6:    $f' \leftarrow \lambda x. \max\{\hat{\theta} - \max_{i \in G} [S(x)]_i, -\kappa\}$ ; ▷ loss function
7:   while True do
8:      $\acute{x} \leftarrow \text{clip}_{x, \epsilon}\{\acute{x} - \eta \cdot \text{sign}(\nabla_x f'(\acute{x}))\}$ ; ▷ craft sample using  $f'$ 
9:     if  $D(\acute{x}) \neq \text{reject}$  then; ▷  $\max_{i \in G} [S(\acute{x})]_i \geq \theta$ 
10:    return  $\max_{i \in G} [S(\acute{x})]_i$ ;
11:    if  $\max_{i \in G} [S(\acute{x})]_i \geq \hat{\theta}$  then break;

```

---

大致的思想是, 先初始化一个较小的估计值  $\hat{\theta}$ , 如果迭代生成对抗样本超过了这个估计值, 但却未输出目标说明人标签时, 增大估计值继续生成对抗样本; (伪代码第 6 行的  $\lambda x$  挺奇怪的, 没太理解)

- 梯度估计 - NES 算法:

$$\frac{1}{m \times \sigma} \sum_{j=1}^m f(\hat{x}_{i-1}^j) \times u_j$$

其中,  $u_j = -u_{m+1-j}$ ,  $\sigma$  是高斯分布的方差;

- 梯度更新 - BIM 算法:

$$\hat{x}_i = \text{clip}_{x,\epsilon}\{\hat{x}_{i-1} - \eta \cdot \text{sign}(\nabla_x f(\hat{x}_{i-1}))\}$$

- 参数选择:  $m = 50$ ,  $\delta = 1e-3$ ,  $\eta \in [1e-3, 1e-6]$ , max iteration = 1000 ;

(4) Attack on CSI: 和 OSI 不同指出是, CSI 一定会输出一个标签, 因此不需要考虑  $\theta$  的问题

- Targeted Attack:

$$f(x) = \max \left\{ (\max_{i \in G \setminus \{t\}} [S(x)]_i - [S(x)]_t), -\kappa \right\}$$

- Untargeted Attack:

$$f(x) = \max \{ ([S(x)]_m - \max_{i \in G \setminus \{m\}} [S(x)]_i), -\kappa \}$$

(5) Attack on SV: SV 是一个单分类的识别系统, 如果为目标说话人则返回 True, 否则返回 False, 因此这种攻击下没有 Targeted / Untargeted 之分.

$$f(x) = \max \{ \theta - S(x), -\kappa \}$$

这里将非目标说话人的语音转化为目标说话人的标签;

## 5. ↗ Evaluation on Effectiveness and Efficiency:

(1) 数据集:

Datasets	#Speaker	Details
Train-1 Set	7,273	Part of <b>VoxCeleb1</b> [69] and whole <b>VoxCeleb2</b> [70] used for training ivector and GMM
Train-2 Set	2,411	Part of <b>LibriSpeech</b> [71] used for training system C in transferability
Test Speaker Set	5	5 speakers from <b>LibriSpeech</b> 3 female and 2 male, 5 voices per speaker, voices range from 3 to 4 seconds
Imposter Speaker Set	4	Another 4 speakers from <b>LibriSpeech</b> 2 female and 2 male, 5 voices per speaker, voices range from 2 to 14 seconds

(2) 评价指标:

Metric	Description
Attack success rate (ASR)	Proportion of adversarial voices that are recognized as the target speaker
Untargeted success rate (UTR) for CSI	Proportion of adversarial samples that are not recognized as the source speaker
Untargeted success rate (UTR) for OSI	Proportion of adversarial samples that are not rejected by the target system

(3) 本地训练的黑盒模型: 设置阈值参数  $\theta_{ivector} = 1.45$ ,  $\theta_{GMM} = 0.091$  以保证 FAR 在 10% 左右 ;

Task	Metrics	ivector	GMM
<b>CSI</b>	Accuracy	99.6%	99.3%
<b>SV</b>	FRR	1.0%	5.0%
	FAR	11.0%	10.4%
<b>OSI</b>	FRR	1.0%	4.2%
	FAR	7.9%	11.2%
	OSIER	0.2%	2.8%

- *FRR* : False Rejection Rate;
- *FAR* : False Acceptance Rate;
- *OSIER* : Open Set Identification Error Rate is the rate of voices that can not be correctly classified;

(3) 修改量的大小: 实验中选择  $\epsilon = 0.002$ ;

$\epsilon$	ivector				GMM			
	#Iter	Time (s)	SNR (dB)	ASR (%)	#Iter	Time (s)	SNR (dB)	ASR (%)
0.05	18	422	12.0	100	18	91	16.7	100
0.01	23	549	16.2	100	16	81	19.1	100
0.005	44	1099	21.8	100	19	102	22.3	100
0.004	56	1423	23.8	100	21	104	24.0	100
0.003	76	2059	26.3	100	27	124	26.1	100
<b>0.002</b>	124	2845	30.2	99	40	218	29.3	99
0.001	276	6738	36.4	41	106	551	35.7	87

(4) 攻击结果: 相比之下, ivector的对抗样本更难生成, 最少的一个样本需要迭代 25 轮(即 query 1250 次);

Task	System				System (Intra-gender attack)				System (Inter-gender attack)								
	ivector		GMM		ivector		GMM		ivector		GMM						
	#Iter	Time (s)	SNR (dB)	ASR (%)	#Iter	Time (s)	SNR (dB)	ASR (%)	#Iter	Time (s)	SNR (dB)	ASR (%)	#Iter	Time (s)	SNR (dB)	ASR (%)	
CSI	124	2845	30.2	99.0	40	218	29.3	99.0	92	2115	29.3	100.0	25	126	28.8	100.0	
SV	84	2014	31.6	99.0	39	241	31.4	99.0	31	751	31.7	98.0	30	185	31.7	100.0	
OSI	86	2277	31.5	99.0	38	226	31.4	99.0	32	833	31.3	98.0	31	178	31.5	100.0	
														146	3340	30.8	98.0
														50	278	29.62	98.0
														48	298	31.2	98.0
														45	274	31.2	98.0

(5) 过程中得到的阈值估计:

ivector			GMM		
$\theta$	$\dot{\theta}$	Time (s)	$\theta$	$\dot{\theta}$	Time (s)
<b>1.45</b>	<b>1.47</b>	<b>628</b>	<b>0.091</b>	<b>0.0936</b>	<b>157</b>
1.57	1.60	671	0.094	0.0957	260
1.62	1.64	686	0.106	0.1072	269
1.73	1.75	750	0.113	0.1141	289
1.84	1.87	804	0.119	0.1193	314

(6) 攻击 Talentedsoft 平台: 成功攻击;

## 6. Evaluation on Transferability:

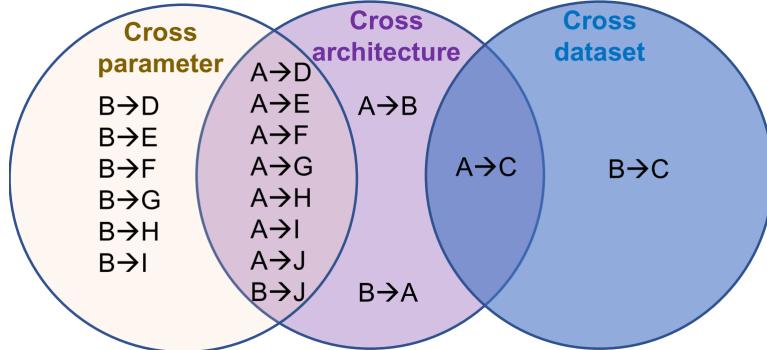
(1) 目标模型结构: A, B, J 为前面实验用到的模型, 这边针对 ivector 和 GMM 增加了 C~I 模型;

System ID	A	B	C	D	E	F	G	H	I	J
Architecture	GMM	ivector	xvector							
Training set	Train-1 Set	Train-1 Set	Train-2 Set	Train-1 Set						
Feature	MFCC	MFCC	MFCC	PLP	MFCC	MFCC	MFCC	MFCC	PLP	MFCC
DF	24×3	24×3	24×3	24×3	13×3	24×3	24×3	24×3	13×3	30
FL/FS (ms)	25/10	25/10	25/10	25/10	25/10	50/10	25/10	25/10	50/10	25/10
GC	2048	2048	2048	2048	2048	2048	2048	2048	1024	-
DV	-	400	400	400	400	400	400	400	600	512

(2) 目标模型训练结果:

System		C	D	E	F	G	H	I	J
Task	Accuracy	99.8%	99.4%	99.2%	99.8%	99.6%	99.8%	99.2%	99.2%
SV	<b>FAR</b>	10.0%	9.8%	9.4%	10.0%	11.2%	9.8%	10.4%	10.2%
	<b>FRR</b>	1.2%	0.6%	1.6%	1.2%	0.8%	1.0%	2.2%	0.8%
OSI	<b>FAR</b>	9.1%	8.8%	10.9%	9.2%	8.5%	8.1%	11.0%	7.7%
	<b>FRR</b>	1.4%	0.6%	1.6%	1.4%	1.2%	0.8%	2.2%	0.8%
	<b>OSIER</b>	0.0%	0.2%	0.2%	0.0%	0.2%	0.0%	0.4%	0.2%

(3) Transferability 的种类: 包括 跨平台, 跨模型种类 和 跨数据集;



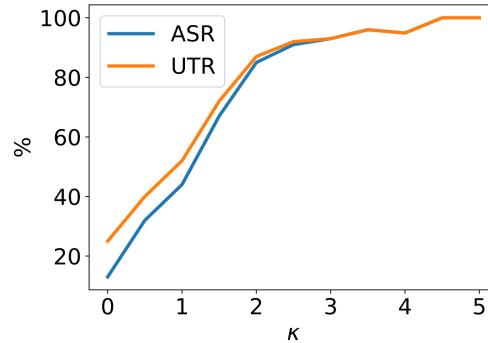
(4) 为了提高 Transfer 能力, 作者对参数的设置如下:

- 修改量:  $\epsilon = 0.05$ , 可以看到在迁移攻击中需要更大的修改量;
- CSI Task:  $k_{GMM} = 0.2$ ,  $k_{ivector} = 10$ ;
- SV Task:  $k_{GMM} = 3$ ,  $k_{ivector} = 4$ ;
- OSI Task:  $k_{GMM} = 3$ ,  $k_{ivector} = 5$ ;

(5) 实验结果:

S \ T	A		B		C		D		E		F		G		H		I		J	
S	ASR	UTR	ASR	UTR	ASR	UTR	ASR	UTR	ASR	UTR	ASR	UTR	ASR	UTR	ASR	UTR	ASR	UTR	ASR	UTR
A	—	—	62.0	64.0	48.0	48.0	55.2	56.9	68.0	68.0	64.0	64.0	52.0	54.0	68.0	68.0	38.0	40.0	34.0	42.0
B	5.0	5.0	—	—	67.5	67.5	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	72.5	75.0	40.0	41.7

(6) 讨论  $k$  的影响:  $k$  越大, transferability 的能力越好;



(7) 攻击 Microsoft Azure 平台: 由于 Azure 上不输出相应的概率, 因此使用 transfer 攻击.

- Text-Independent OSI-Azure:
- ☆ Text-Dependent SV-Azure: 只实现了 10% 的成功率, 其他的都因为添加的噪声过多而出现 "Error, too noisy";

文本相关的, 并且判断语音中的噪声, 这样的语音认证对于攻击更加鲁棒, 能否对这个系统进行攻击?

## 7. Evaluation on Over-the-Air

(1) 实验环境:

	System	Loudspeaker	Microphone	Distance	Acoustic Environment
Different Systems	GMM OSI/CSI/SV ivektor OSI/CSI/SV Azure OSI	JBL clip3 portable speaker	iPhone 6 Plus (iOS)	1 meter (65 dB)	relatively quiet
Different Devices	ivektor OSI	DELL laptop JBL clip3 portable speaker Shinco brocast equipment	iPhone 6 Plus (iOS) OPPO (Android)	1 meter (65 dB)	relatively quiet
Different Distances	ivektor OSI	JBL clip3 portable speaker	iPhone 6 Plus (iOS)	0.25 meter (70 dB) 0.5 meter (68 dB) 1 meter (65 dB) 2 meters (62 dB) 4 meters (60 dB) 8 meters (55 dB)	relatively quiet
Different Acoustic Environments	ivektor OSI	JBL clip3 portable speaker	iPhone 6 Plus (iOS)	1 meter (65 dB)	white noise (45/50/60/65/75 dB) bus noise (60 dB) restaurant noise (60 dB) music noise (60 dB) absolute music noise (60 dB)

(2) Result of Different Systems:

System		SNR (dB)	Result (%)					
			Normal voices			Adversarial voices		
ivektor	CSI	6.6	Accuracy: 100			ASR: 80, UTR: 80		
	SV	9.8	FAR: 0, FRR: 0			ASR: 76		
	OSI	7.8	FAR: 4, FRR: 0, OSIER: 0			ASR: 100, UTR: 100		
GMM	CSI	6.1	Accuracy: 85			ASR: 90, UTR: 100		
	SV	7.9	FAR: 0, FRR: 62			ASR: 100		
	OSI	8.2	FAR: 0, FRR: 65, OSIER: 0			ASR: 100, UTR: 100		
Azure	OSI	6.8	FAR: 5, FRR: 2, OSIER: 0			ASR: 70, UTR: 70		

(3) Result of Different Devices:

M L	iPhone 6 Plus (iOS)					OPPO (Android)				
	Normal voices			Adv. voices		Normal voices			Adv. voices	
	FAR	FRR	OSIER	ASR	UTR	FAR	FRR	OSIER	ASR	UTR
DELL	10	0	0	100	100	13	6	0	78	80
JBL clip3	4	0	0	100	100	6	0	0	80	80
Shinco	8	5	0	89	91	14	0	0	75	75

(4) Result of Different Distance:

Distance (meter)			0.25	0.5	1	2	4	8
Normal Voices	FAR		4	3	4	6	0	0
	FRR		0	0	0	5	10	32
	OSIER		0	0	0	0	0	0
Adversarial Voices	ASR		100	100	100	70	40	10
	UTR		100	100	100	70	50	10

(5) Result of Different Acoustic Environment:

Environment		Quiet	White (45 dB)	White (50 dB)	White (60 dB)	White (65 dB)	White (75 dB)	Bus (60 dB)	Rest. (60 dB)	Music (60 dB)	Abs. Music (60 dB)
Normal voices	FAR	4	0	6	0	0	10	0	0	0	4
	FRR	0	5	12	30	40	97	25	20	10	10
	OSIER	0	0	0	0	0	0	0	0	10	0
Adv. voices	ASR	100	75	70	57	20	2	50	50	66	48
	UTR	100	75	70	60	20	2	50	50	67	48

## Links

- 论文链接: [Chen, Guangke, et al. "Who is real bob? adversarial attacks on speaker recognition systems." S&P \(2021\).](#)
- 论文主页: <https://sites.google.com/view/fakebob>
- 论文代码: <https://github.com/FAKEBOB-adversarial-attack/FAKEBOB>

# **SoK: The Faults in our ASRs: An Overview of Attacks against Automatic Speech Recognition and Speaker Identification Systems**

---

**Notes**

**Links**

-