Attacking a Speaker Recognition System

Weiji Li [1]

Abstract

In this paper, I conclude my research findings on audio in machine learning, especially on the speaker recognition system. At first, I briefly give an introduction to the audio domain in the machine learning world and the related attacks. I also present selected papers that I have read in different sub-domains of audio. Then I focus on one of the state-of-the-art speaker recognition systems. I did several experiments related to attacks in the speaker recognition system to test the robustness and vulnerability of an existing system. At last, I present my own attacking system for the speaker recognition system and give some future improvements related to audio in machine learning.

1. Overview of Audio in Machine Learning

Audio is an emerging domain in machine learning as it involves a lot of applications in our real life. Its task is mainly about getting information from audio or generating some audio from some given information. It is very similar to computer vision, but the target becomes audio instead of images.

As a quite new domain in machine learning, audio could take advantage of many existing algorithms from other domains such as computer vision and natural language processing. The only difference is about transferring between an audio and feature vectors. The most common technique is using frequency and spectrogram analysis such as MFCC (Mel-Frequency Cepstral Coefficients). They have been widely used in various state-of-the-art systems.

The most popular topics in audio with machine learning nowadays include speech recognition (speech-to-text), speaker verification and audio generation.

1.1. Three main topics

Speech recognition: Speech recognition is mainly about transferring speech into human-readable text and it is used most widely in our life.

Speaker recognition: Speaker recognition is about identifying the speaker from their voice. It could identify whether one piece of voice comes from a specific person, or from a person in a group of people. Details on this topic would be covered in section 3.

Audio generation: Audio generation is about generating audio samples from previous recordings and given text. It is usually used combined with video so that people are able to create a new video that looks like real capturing.

1.2. Application

There are a lot of applications of audio in the real world. Popular commercial virtual assistants such as Siri, Google Home and Alexa are all using audio systems widely to bring convenience to people's lives. They use speech recognition systems to understand people's commands and perform appropriate behaviors. They also use speaker verification systems to verify a person's voice. For example, "Hey Siri" would only wake your iPhone and only your voice could be recognized by your Siri. And when these virtual assistants are responding to your voice, they would generate audio from the generated transcript.

1.3. Related Attacks

For all kinds of machine learning systems, there would be attacks that could fool the system or make them vulnerable. For example, in computer vision, there is some physical attack that could fool the vision recognition system on the car and cause damage to the autonomous vehicle (Evtimov et al., 2017). Similarly, there are also different kinds of attack in audio.

In speech-to-text scenario, the attack could add very little noise to the audio samples but it would make the system produce incorrect transcript or even produce targeted transcript. The people are usually not able to distinguish the difference between original samples and adversarial samples. This might be very dangerous because the attacker

[1] University of Michigan, Ann Arbor. Correspondence to: Weiji Li <weijili@umich.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

could potentially give some bad commands to the voice assistants.

In the audio generation scenario, the attack is more like DeepFake in audio. DeepFake is to create new video clips of someone from old video clips. And the attack in audio generation is to create audio samples of someone saying a target text. This kind of attack is usually trained on the previous audio recording of the person. And the system could output audio samples from the input target text.

Similar to computer vision, the security aspect of machine learning systems is very important nowadays. Many commercial systems are using machine learning in audio and some of them involve controlling IoT, mobile devices and even cars. Therefore, I think it is important to conduct research in the attacks and defenses in the audio ML system.

2. Existing Papers

This section includes the most representative and interesting papers I have read in the domain of audio and attacks in machine learning.

2.1. Who is Real Bob? Adversarial Attacks on Speaker Recognition Systems (Chen et al., 2020)

This paper relates to my research focus directly. In this paper, the authors proposed an adversarial attack on speaker recognition under different scenarios. It used the same system I used for speaker recognition built with Kaldi.

The highlights in this paper are its multiple settings and its successful result. Firstly, the authors used 16 attack scenarios in total. Overall, they have attack type(untargeted, targeted), gender(intra-gender, inter-gender), attack channel(API, Over-the-air), target system(SV, CSI, OSI) and output(scores, decision) five different categories. And with each scenarios, they changed their attack setting and did experiments. I think it is really helpful to see the attack under different attack settings and would prove that their attack is successful across different scenarios.

Besides, they also achieved a great result in most cases. The attack success rate (ASR) is over 98% in most attacking scenarios, even for over-the-air attack. They also extended their attack to existing commercial systems such as Microsoft Azure. Therefore, I think this paper is a comprehensive and systematic paper for black-box attack on the speaker recognition system.

2.2. Audio Adversarial Examples: Targeted Attacks on Speech-to-Text (Carlini & Wagner, 2018)

This paper is also about targeted attack on audio, but it is about speech-to-text(aka speech recognition) and the system is white-box. In this paper, the authors proposed an attack to produce a very similar audio but would transcribe as any phase by Mozilla's DeepSpeech.

The highlight in this paper is its algorithm. The goal is to minimize the noise while keeping the recognition wrong. They used some novel ideas when solving the minimization problem. Because the DeepSpeech model they were attacking is end-to-end, they differentiated through the entire classifier to generate adversarial examples. They also did the minimization over the complete audio sample instead of frame by frame.

2.3. Hidden Voice Commands (Carlini et al., 2016)

This paper is also about targeted attack on audio. It is about issuing specific commands to devices through some random voices. The author talked about black-box attack, white-box attack and related defenses in the paper.

In the black-box scenario, they make use of crowdsourcing and machine understanding at the same time in order to produce audio samples that are recognizable by machine but not recognizable by humans. In the white-box scenario, because they assume they knew the internal models and parameters for the speech recognition system, they used gradient descent to generate samples frame by frame.

They also proposed several defense techniques. Some defenses would notify the user when a command was executed. Some defenses would use audio CAPTCHA to authenticate the commands. And some defenses would apply speaker recognition to the system in order to verify the speakers.

2.4. Adversarial Audio Synthesis (Donahue et al., 2019)

This paper is about generating audio samples. The authors proposed a method named WaveGAN which could produce intelligible audios of words as well as drums, piano and bird vocalizations.

The authors mainly applied the deep convolutional GAN(DCGAN) algorithm from computer vision to both the waveform and spectrogram in audio. There are some differences between audio and image. In general, natural audio signals are more likely to exhibit periodicity than natural images. And audios are usually much larger compared to image. The waveGAN is targeting the waveform and it could produce hours of audio samples in a few seconds. The specGAN is targeting the spectrogram and it could achieve good invertibility. Both of them are rated high (about 6/10) by humans. This paper is a good starting point for audio generation and it built a good connection between audio and images.

3. Overview of speaker recognition

3.1. Introduction

Speaker recognition system is a system that could recognize the speaker from voice. There are mainly two kinds of speaker recognition tasks – speaker verification (SV) and speaker identification (SI).

SV is defined as speaker verification. It is the simplest one as it only has one enrolled speaker. The system would simply output whether a given audio sample is from this enrolled speaker. SV is a simple binary classification problem.

SI is defined as speaker identification. It means a group of enrolled speakers. In a previous paper (Chen et al., 2020), SI is further divided into open-set identification (OSI) and close-set identification (CSI). In OSI, for a given audio sample, the system would accept it as one of the enrolled speakers or reject it. But for CSI, the system assumes the given audio must be spoken by one of the enrolled speakers. It would always classify the input audio as one of the enrolled speakers.

For both SV and SI (including OSI and CSI), the score for each audio sample is the key term. The score $S(x)$ represents how possible this audio sample x is spoken by this speaker. In SV, the output $D(x)$ on a given audio x is defined as

$$D(x) = \begin{cases} accept, & \text{if } S(x) > \delta \\ reject, & \text{otherwise} \end{cases} \tag{1}$$

In OSI, the output $D(x)$ is

$$D(x) = \begin{cases} \arg\max_{i \in G} [S(x)]_i, & \text{if } \max_{i \in G} [S(x)]_i > \delta \\ reject, & \text{otherwise} \end{cases} \tag{2}$$

In CSI, the output $D(x)$ is

$$D(x) = \arg\max_{i \in G} [S(x)]_i \tag{3}$$

3.2. GMM-UBM and ivector-PLDA model

One of the state-of-the-art systems in speaker recognition is the GMM-UBM and ivector-PLDA model. In the rest of the paper, I would focus on this kind of system.

3.2.1. Feature Extraction

The first step is feature extraction from raw audio. First we need to do some transformations and preprocessing to the raw audio so that we could represent all the features from the audio in a trainable format. We would use Mel-frequency cepstral coefficients (MFCC) and Voice activity detection (VAD) as two measurements for the input audio. MFCC is a commonly-used spectrum analysis in audio that could extract the sound feature from the audio. VAD is another technique that could extract the sound and pause time pattern from the audio. Both MFCC and VAD could be represented as a n-dimensional vector and could be directly used for training in the next step.

3.2.2. GMM-UBM

The next step is the Universal Background Model based on the Gaussian Mixture Model (GMM-UBM) (Reynolds et al., 2000). GMM-UBM is only trained once in the training stage. It is used for training an ivector extractor for the given audio. When we have all the MFCC and VAD vectors for all the training data, we could use the GMM method to train a UBM model (typical with 2048 Gaussians). This represents our initial classification for the input audio. Then we would use this UBM model to train an ivector extractor. This i-vector extractor could directly generate a small-dimensional feature vector(usually around 400-dimensional) from the MFCC and VAD vector.

3.2.3. ivector-PLDA

When we have the i-vector extractor, we could apply it to all the training audio. With all the training i-vectors, we could then train our Probabilistic Linear Discriminant Analysis (PLDA) (Prince & Elder, 2007) model. PLDA is an effective machine learning scoring algorithm that is previously used in computer vision for face recognition. It could classify the feature vectors and output a score for each audio sample i on each speaker j. Higher score means this audio sample i is more possible to be spoken by speaker j. The threshold could be set differently depending on the audio noise and different speaker recognition goal.

4. Experiments

4.1. Robustness of the speaker recognition system

The first experiment I did was to test the robustness of the speaker recognition system. I added two interference, room impulse and background noise to the speaker recognition system and tested the effectiveness of the system.

4.1.1. Room Impulse

For the room impulse simulation, I use a python acoustic library (pyroomacoustics – `https://pypi.org/project/pyroomacoustics/`). It is a python library that could simulate a real room and real sound impulse. I use the following setup. The room is 10 * 10. The source is placed at the center of the room ([5,5]) and the microphone is placed near the corner ([8, 8]).

Table 1. Classification scores before and after room impulse

Audio	Target?	Original Scores	New Scores
1	√	7.675023	7.440361
2	×	-26.6342	-26.23978
3	√	9.409912	9.179755
4	×	-11.48497	-9.844815
5	√	8.653882	4.682043
6	×	-11.57315	-9.225708
7	√	13.32605	10.60043
8	×	-11.27807	-11.77835
9	√	8.027048	7.133739
10	×	-10.36581	-4.991007

And after running the simulation on 100 different audios, the result is shown in Table 1. We could see that the scores didn't change too much. It showed that this speaker recognition system is robust to room impulse.

4.1.2. Background Noise

For the background noise, I use the physical method directly – adding background noise when playing the original audio over-the-air. The setup is as follows: an iPhone played the original audio near the microphone of a PC (macbook pro) and I, as the background speaker, said "Describe the city you live in " for several times about 50cm away. Because of the manual setup, I only did this experiment on 10 different audios (5 for the enrolled speaker and 5 for different unenrolled speakers).

Table 2. Classification scores before and after adding backgroudn noise

Audio	Target?	Original Scores	New Scores
1	√	7.675023	5.058002
2	×	-26.6342	-22.47963
3	√	9.409912	6.154756
4	×	-11.48497	-8.867513
5	√	8.653882	6.125574
6	×	-11.57315	-10.35437
7	√	13.32605	7.63099
8	×	-11.27807	-7.076748
9	√	8.027048	5.813542
10	×	-10.36581	-1.473995

The result is shown in Table 2. The background noise could be heard in the audio but the system was still able to classify correctly on all the 10 audios. However, the scores were different from the original audio. The difference between true and false samples is decreased on some examples. This experiment showed this speaker recognition system is robust to background noise. The reason might be that the UBM model would remove the background noise and the i-vector extracted would not contain the background noise feature.

4.2. FGSM attack

The next experiment I did was a simple attack to the speaker recognition system — Fast Gradient Sign Method (FGSM). FGSM is a very famous and relatively simple algorithm for adversarial attack in computer vision. I thought of adopting a similar approach to the audio domain and see whether it is effective to attack the speaker recognition system.

In the original FGSM, the attack is effective mainly because of the gradient. However, in this speaker recognition system, we need to do a black-box attack because we were not able to get the intermediate result or the loss function. Therefore, we need to do a gradient estimation in order to perform a FGSM attack to the audio. For this experiment, I didn't use a very complicated gradient estimation function. I just used the output score times a random noise array as the gradient.

Table 3. Classification scores before and after adding backgroudn noise

Audio	Target?	Original Scores	New Scores
1	√	7.675023	-10.16234
2	×	-26.6342	-9.514515
3	√	9.409912	-5.98375
4	×	-11.48497	-11.22756
5	√	8.653882	-3.369277
6	×	-11.57315	-12.87101
7	√	13.32605	-4.266976
8	×	-11.27807	-1.725142
9	√	8.027048	-9.701138
10	×	-10.36581	-6.60997

From the result shown in Table 3, we could see that the FGSM attack is effective. In most cases, it would make the target samples be not recognized and it would make the nontarget samples' scores higher. However, when listening to the audio itself, it contains loud and noticeable noise within the audio. Therefore, I conclude that the FGSM is effective but not applicable for attacking the black-box speaker recognition system.

4.3. FAKEBOB reproduction

I have also reproduced the Fakebob attack (Chen et al., 2020). Fakebob is also using the same speaker recognition system as the one I use. The repo for code is public on github at `https://github.com/FAKEBOB-adversarial-attack/FAKEBOB/`. It was written to run on a Linux multi-core server, so I made a few syntax changes to run it on my own PC (macOs).

Fakebob has different configurations such as model type(GMM or i-vector PLDA), speaker recognition mode (SV, OSI or CSI). I used the i-vector PLDA mode with SV for testing purposes. The result is successful as it could fool the speaker verification system with adversarial samples and I could not distinguish between the original audio and adversarial audio. It takes about 20s for each iteration and about 100 iterations for each sample. The total running time is about 30mins for each sample. This experiment showed Fakebob is reproducible. And I think the running speed could be improved on servers with higher computing resources.

5. The system

In the last part, I have built my own attacking system targeting the GMM-UBM and ivector-PLDA speaker recognition system. I mainly focus on the speaker verification (SV) scenario where there is only one enrolled speaker. Because there is not too much existing support for audio-related machine learning, I didn't use any existing library such as pytorch or tensorflow. At the same time, some of my code was inspired by Fakebob. The code is located at `https://github.com/weiji-li/audio_attack-ML`.

5.1. Kaldi Helper

The first part of my attacking system is the Kaldi Helper, a wrapper for necessary functions for the kaldi library. The speaker recognition system I am attacking is built in Kaldi, which is written in C and is used with shell script. Therefore, I need to build a kaldi helper at first to deal with the system conveniently.

The testing pipeline for the system consists of 7 steps. And I wrote these functions in my kaldi helper library.

data_prepare: organize the required data directory(utt2spk, spk2utt, wav.scp) from the input audio. These files are basically dictionaries between speakers and utterances. The system in kaldi requires those files as input to train and test.

make_mfcc: compute MFCC for all the input audio

compute_vad: compute VAD matrix for all the input audio

extract_ivectors: extract i-vectors for all the input audio using the trained extractor

get_score: compute the PLDA score for all the input audio using the trained i-vector scoring model

compute_eer: an optional function for computing Equal error rate(EER). EER could be used to evaluate the performance of the model.

score: compare the PLDA scores with the correct label.

run: run the whole pipeline

5.2. Main Attack

My attack is an iterative algorithm based on gradient descent. For each iteration, I would update the audio by adding estimated gradients to the previous audio. The algorithm is not complicated and the key step is about estimating the gradient from the black-box model.

My algorithm for gradient estimation is inspired by Fakebob. They used Natural Evolution Strategies (NES) (Wierstra et al., 2011) for gradient estimation. NES is an algorithm that only depends on the recognition result. Let x_i be the input voice for iteration i. In each iteration, we would create n Gaussian noises (g_j) and add them to the current x_i. The σ is search variance. Then we would query the system with these new audios. The gradient is computed as

$$x_i^j = x_i + \sigma \cdot g_j \tag{4}$$

$$\frac{1}{n \cdot \sigma} \sum_{i=1}^{n} f(x_i^j) \cdot g_j \tag{5}$$

This gradient estimation algorithm works effectively on the Fakebob attack, which is black-box. Therefore, I implemented a similar algorithm for gradient estimation in my system. For the whole attacking algorithm, I use an iteration-based gradient descent attack. The whole algorithm is shown as Algorithm 1. The attacking parameter κ could be used to control the strength of attacking audio.

Algorithm 1 Attacking algorithm

```
Input: Scoring function $S(\cdot)$, Loss function $f(\cdot)$, input voice $x$, learning rate $\eta$, system threshold $\delta$, attacking parameter $\kappa$.
Output: output voice $x_{output}$
while $S(x_i) < \delta$ do
    $f(x_i) = \max\{\delta - S(x_i),\ \kappa\}$
    $x_{i+1} = x_i + \eta \cdot \nabla_{x_i} f(x_i)$
    $i = i + 1$
end while
$x_{output} = x_i$
```

5.3. Result

So far, the result of my attacking system is not very successful. I used the score = 0 as the benchmark so a positive score means verified audio and a negative score means unverified audio. During the experiments I have done, I was able to increase the negative score of false audios, but not able to change its sign so they could not be classified as true audios. However, I also noticed a serious issue when listening to the output audio. The noise is too loud even after a few iterations. Therefore, I didn't list the scores here because I think this system is not successful yet.

In the future, I think the key step of the system is still in the gradient estimation part. We may need to find a more effective algorithm to estimate the gradient only based on the recognition result scores. In conclusion, I think this version of the attacking system is a working system that includes the whole pipeline of attacking but needs more research to be an effective and successful attack.

6. Conclusion

From my research, I think of several future improvements that could be done.

Attacks: Because the state-of-the-art audio system, especially the speaker recognition system, are usually end-to-end and black-box. Also, the frequency transformation is similar to image processing in computer vision. Therefore, in order to attack those systems, I think we could apply similar attacking techniques from computer vision.

Defenses:: The state-of-the-art speaker recognition systems nowadays are still very vulnerable to attacks. Therefore, unlike fingerprint or face recognition, there is still no application that uses speaker recognition to verify identity. In my opinion, how to make a very robust and secure speaker recognition system is a very interesting topic related to defense. For example, does there exist any special words that could make the speaker recognition robust?

Potential Topics:: The audio generation is also a very interesting field in audio with machine learning. It is usually related to video generation(Deepfake). This technology could also be widely used in daily life and could also lead to potential attacks. I think audio generation is a great sub-domain to explore and conduct some security-related research along with video generation in computer vision.

My research mainly focuses on exploring audio in machine learning and related attacks, especially in speaker recognition systems. It is a quite new field in machine learning compared to computer vision or natural language processing. But I think it is very interesting, promising and important. There are many voice virtual assistants like Siri and Alexa nowadays. With the help of machine learning in audio, there are much more tasks we could accomplish such as controlling IoT conveniently. Meanwhile, we also need to pay attention to the security aspect in audio because this kind of technology could bring us serious problems while bringing us convenience.

At last, I want to appreciate the genuine help from Prof. Atul Prakash and Ryan Feng. As a new undergrad to research, they taught me so many useful methods and gave me a lot of advice in machine learning and security from their experience. In 6 months, I became familiar with audio systems in machine learning and I got in touch with independent research for the first time. I believe this could be a very great starting point for me to start my future career in research.

References

Carlini, N. and Wagner, D. Audio adversarial examples: Targeted attacks on speech-to-text. *2018 IEEE Security and Privacy Workshops (SPW)*, 2018. doi: 10.1109/spw.2018.00009.

Carlini, N., Mishra, P., Vaidya, T., Zhang, Y., Sherr, M., Shields, C., Wagner, D., and Zhou, W. Hidden voice commands. In *25th USENIX Security Symposium (USENIX Security 16)*, pp. 513–530, Austin, TX, August 2016. USENIX Association. ISBN 978-1-931971-32-4.

Chen, G., Chen, S., Fan, L., Du, X., Zhao, Z., Song, F., and Liu, Y. Who is real bob? adversarial attacks on speaker recognition systems, 2020.

Donahue, C., McAuley, J., and Puckette, M. Adversarial audio synthesis, 2019.

Evtimov, I., Eykholt, K., Fernandes, E., Kohno, T., Li, B., Prakash, A., Rahmati, A., and Song, D. Robust physical-world attacks on machine learning models. *CoRR*, abs/1707.08945, 2017. URL http://arxiv.org/abs/1707.08945.

Prince, S. J. D. and Elder, J. H. Probabilistic linear discriminant analysis for inferences about identity. In *2007 IEEE 11th International Conference on Computer Vision*, pp. 1–8, 2007. doi: 10.1109/ICCV.2007.4409052.

Reynolds, D. A., Quatieri, T. F., and Dunn, R. B. Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*, 10(1-3):19–41, 2000. doi: 10.1006/dspr.1999.0361.

Wierstra, D., Schaul, T., Glasmachers, T., Sun, Y., and Schmidhuber, J. Natural evolution strategies, 2011.