Defending and Detecting Audio Adversarial Example using Frame Offsets

Yongkang Gong[1], Diqun Yan*, Terui Mao, Donghua Wang, and Rangding Wang
College of Information Science and Engineering, Ningbo University
Zhejiang Ningbo, 315211 China
[e-mail: gongyibei@gmail.com, yandiqun@nbu.edu.cn, 1911082213@nbu.edu.cn,
923237475@qq.com, wangrangding@nbu.edu.cn]
*Corresponding author: Diqun Yan

Received July 20, 2020; revised February 25, 2021; accepted March 8, 2021; published April 30, 2021

Abstract

Machine learning models are vulnerable to adversarial examples generated by adding a deliberately designed perturbation to a benign sample. Particularly, for automatic speech recognition (ASR) system, a benign audio which sounds normal could be decoded as a harmful command due to potential adversarial attacks. In this paper, we focus on the countermeasures against audio adversarial examples. By analyzing the characteristics of ASR systems, we find that frame offsets with silence clip appended at the beginning of an audio can degenerate adversarial perturbations to normal noise. For various scenarios, we exploit frame offsets by different strategies such as defending, detecting and hybrid strategy. Compared with the previous methods, our proposed method can defense audio adversarial example in a simpler, more generic and efficient way. Evaluated on three state-of-the-arts adversarial attacks against different ASR systems respectively, the experimental results demonstrate that the proposed method can effectively improve the robustness of ASR systems.

Keywords: Speech Recognition Safety, Adversarial Defense, Adversarial Detection, Audio Adversarial Example, ASR

http://doi.org/10.3837/tiis.2021.04.019 **ISSN : 1976-7277**

1. Introduction

Recently, machine learning models are widely used in various fields, but many security issues also exist, especially, the adversarial attack. Studies show that several machine learning models are vulnerable to adversarial examples which generated by adding an elaborately designed perturbations to a benign sample [1, 2].

For automatic speech recognition (ASR) system, its accuracy can be improved by deep learning network. However, potential security risk is also aroused. A benign audio could be decoded as a harmful command by adversarial attack. Existing defense methods against audio adversarial examples have following deficiencies: necessity to retrain ASR models [3] or use additional machine learning models [4-6], only applicable to specific models [7, 8]. Most of those methods may be inspired by image adversarial example domain and not consider the characteristics of ASR systems.

The main contributions of our work: 1) Utilizing the characteristics of ASR systems, we offer a novel insight in explore countermeasures against audio adversarial example. 2) We propose a simple, generic and efficient method against on audio adversarial attack. Our method does not need to retrain ASR systems, and can be applied for different ASR systems (such as Classification, Kaldi and DeepSpeech [9-11]) and achieves better performance than existing methods. 3) For different scenarios, we give a variety of application strategies of our method.

The rest of the paper is organized as follows. In Section II, related works of the adversarial attacks and countermeasures for ASR system are described. Details of the proposed method, which includes the frame offsets, defense strategies and theoretical analysis, are described in Section III. The evaluation results of our method are present in Section IV. We conclude our work in Section V.

2. Related Work

According to the output types of different ASR systems, the attack task can be divided into two categories: speech-to-label and speech-to-text. For speech-to-label task, an audio is discriminated to a label by ASR classification model. The adversary is targeted at making the audio be discriminated to a label which is different from the real one. Alzantot et al. [12] proposed a genetic algorithm-based method to generate audio adversarial examples against speech-to-label model. For speech-to-text task, an audio is directly decoded as a text by ASR system. The adversary is targeted at making the audio be decoded as a pre-specified text. Carlini et al. [13] are the first to make the audio adversarial examples work on speech-to-text models. Yuan et al. [14] choose the music as the carrier and achieved practical over-the-air audio adversarial attacks. Several of the later works are based on [13] to improve the generation efficiency and robust of adversarial example [15-17].

On the other hand, there are several countermeasures to audio adversarial example. Sun et al. [3] add adversarial examples to the training dataset to make the ASR system more robust. Samizade et al. [6] design a CNN based classification model to detect audio adversarial example. Some other methods denoise the audio adversarial example with self-attention U-Net [5] or GAN [4] to fail the adversarial attacks. Yang et al. [18] propose a novel detection method using temporal dependency. Several transformation-based methods [18, 19] (i.e. down-sampling, quantization, local-smoothing, compressions et al.) are utilized to defense the audio adversarial examples.

3. Proposed Method

The frame offsets [20] are defined as the shifting samples of the frame grid between the first and second encoding. In our works, we make an audio get frame offsets by appending silence clip (ASC) at the beginning of it. In the section, we first introduce how frame offsets resists the audio adversarial examples, then give three application strategies for different scenarios and finally give a theoretical analysis to which length silence clip is appropriate for appending.

3.1 Frame offsets with ASC

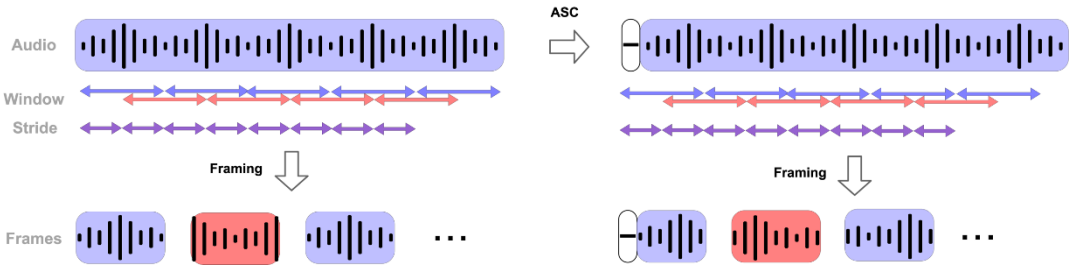


Fig. 1. Comparison of framing process for an audio (left) and its processed sample (right) by frame offsets with ASC

Most of modern ASR systems often need to extract features (e.g., MFCCs) from audio and the value of the features will largely determine the output of the system. It should be noted that the extracted features will be affected by the window size and frame stride. As shown in **Fig. 1**, if the first frame gets offsets by appending a silence clip at the beginning of an audio, the remaining frames will also get the same offsets and the values of extracted features will be changed. For a benign audio, the recognition results will not be affected by frame offsets. Generally, at the beginning of recording, there will be a silence clip with a random length. The length of the silence clip will not affect the recognition output due to the temporal dependency [18] which has been widely used in audio systems. For adversarial audio, the adversarial perturbation is elaborately designed for the original audio temporal. When the temporal is changed by frame offsets, the perturbations will degenerate into ordinary noise. **Fig. 2** shows the change in recognition results of benign audio and adversarial audio after frame offsets with ASC.

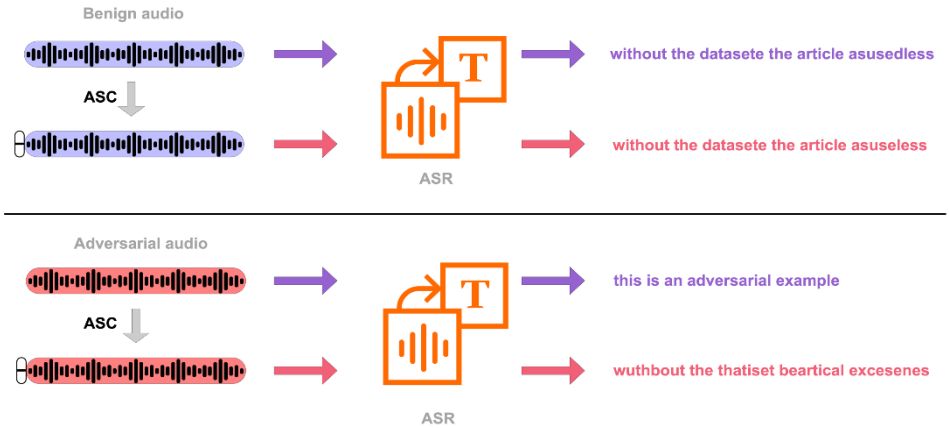


Fig. 2. Recognition results of benign audio and adversarial audio after frame offsets with ASC

3.2 Exploit frame offsets by different strategies

As mentioned in Section 3.1, the adversarial example is more vulnerable to frame offsets. The phenomenon can be exploited by different strategies such as defending, detecting and hybrid strategy as the countermeasure against audio adversarial example.

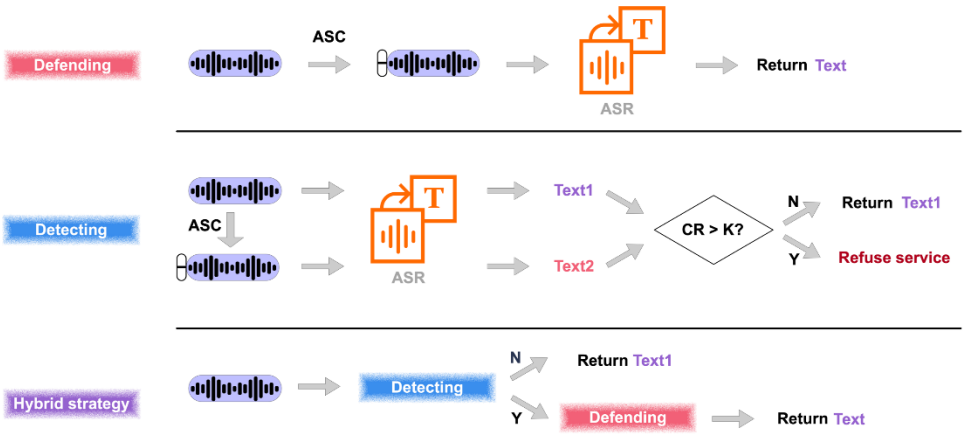


Fig. 3. Pipeline of proposed method with different strategies

3.2.1 Defending

The purpose of defending strategy is to make the adversarial audio harmless. The first row of **Fig. 3** shows that the pipeline of defending strategies. Given an audio instance x (benign sample or adversarial sample), we append an appropriate length silence clip at the beginning of the audio and fed it into the ASR system. For benign audio, the recognition results will not be changed. For adversarial audio, the recognition results will different from before and harmless. We will explore which length silence clip is appropriate for appending by theoretical analysis and experiments.

3.2.2 Detecting

The purpose of detecting strategy is to identify whether unknow types of audio are adversarial samples. The recognition result of adversarial example is more likely to be affected by frame offsets. Hence, we can use the amount of change in recognition results before and after using frame offsets to detect whether an audio sample is adversarial sample. Given an audio instance x (benign sample or adversarial sample), ASR function $g(\cdot)$ and manipulation function $ASC_\epsilon(\cdot)$. $ASC_\epsilon(x)$ means to append a silence clip at the beginning of x. The details of $ASC_\epsilon(\cdot)$ will be discussed in Section 3.2.3. We use the word/character change rate (CR) to measure the amount of change in recognition results of x, CR is defined as follow:

$$CR = \frac{\min\left(D\left(g\left(ASC_\epsilon(x)\right), g(x)\right), L\right)}{L} \tag{1}$$

where L denotes the number of words/characters of $g(x)$, $D(\cdot,\cdot)$ denote the distance of two texts ($\left(ASC_\epsilon(x)\right)$ and $g(x)$ in our case). The word error rate (WER) and character error rate (CER) [21] is used as the distance function $D(\cdot,\cdot)$. Corresponding, we can get the word change rate (WCR) and character change rate (CCR). The magnitude of CR can characterize the possibility whether x is an adversarial example. The closer CR is to 1, the more likely x is

regarded as adversarial example.

3.2.3 Hybrid strategy

Different strategies may be suitable for different scenarios. The defending strategy is straightforward, but the recognition result of benign audio also be modified. The detecting strategy does not modify the recognition result, but the users may have a bad experience when the adversarial attack occurs during transmission and the users are not aware of it. Hence, we can combine the defending strategy and detecting strategy. The last row of **Fig. 3** shows the pipeline of hybrid strategy. Firstly, it needs to detect whether an audio is adversarial example. If the audio is an adversarial example, we can use frame offsets or other defending methods to defend it.

3.3 Theoretical Analysis

Given an audio signal x, with f Hz sampling rate, t seconds and l samples ($l = f * t$). When extracting features, different audio system could use different window size and stride. We suppose that the window size is t_w seconds and the stride is t_s second. One window has $l_w = f * t_w$ samples and one stride has $l_s = f * t_s$ samples. Splitting x into a series of frames:

$$\boldsymbol{x} = \left[\boldsymbol{x}^{(1)}, \cdots, \boldsymbol{x}^{(N)}\right], \quad N = \lfloor \frac{l-(l_w-l_s)}{l_s} \rfloor \tag{2}$$

$$\boldsymbol{x}^{(k)} = \left[\boldsymbol{x}_{i+1}, \cdots, \boldsymbol{x}_{i+l_w}\right], \quad i = (k-1) * l_s \tag{3}$$

where $\boldsymbol{x}$ is vector-notation of x, $\boldsymbol{x}^{(k)}$ denotes the k-th frame of x and $\boldsymbol{x}_i$ denotes the i-th sample of x. Finally, we can get N frames.

In order to offset the frames of x, we append a silence clip at the beginning of x. Let us formula the operation as follows:

$$ASC_\epsilon(x) = [\underbrace{0, \cdots, 0}_{l_a}, \boldsymbol{x}], \quad l_a = \lfloor \epsilon * l_s \rfloor \tag{4}$$

where $ASC_\epsilon(\cdot)$ is the frame offset function and ϵ is the coefficient which is used to control the length of silence clip (l_a).

We can get $\widehat{\boldsymbol{x}}$ by feeding $\boldsymbol{x}$ into $ASC_\epsilon(\cdot)$:

$$\widehat{\boldsymbol{x}} = ASC_\epsilon(\boldsymbol{x}) \tag{5}$$

$$\widehat{\boldsymbol{x}} = \left[\widehat{\boldsymbol{x}}^{(1)}, \cdots, \widehat{\boldsymbol{x}}^{(\widehat{N})}\right], \quad \widehat{N} = \lfloor \frac{l+l_a-(l_w-l_s)}{l_s} \rfloor \tag{6}$$

$$\widehat{\boldsymbol{x}}^{(k)} = \left[\widehat{\boldsymbol{x}}_{i+1}, \cdots, \widehat{\boldsymbol{x}}_{i+l_w}\right], \quad i = (k-1) * l_s \tag{7}$$

Via $\widehat{\boldsymbol{x}}_i = \boldsymbol{x}_{i-l_a}$, we can get:

$$\hat{\boldsymbol{x}}^{(k)} = \begin{cases} [\underbrace{0,\cdots,0}_{l_w}], & k \le \lfloor \frac{l_a}{l_s} \rfloor \\ [\underbrace{0,\cdots,0}_{l_a \bmod l_s}, \boldsymbol{x}_1, \cdots, \boldsymbol{x}_{l_w - l_a}], & k \le \left\lfloor \frac{l_a}{l_s} \right\rfloor + 1 \\ [\underbrace{\boldsymbol{x}_{i-l_a+1}, \cdots, \boldsymbol{x}_i}_{l_a \bmod l_s}, \boldsymbol{x}_{i+1}, \cdots, \boldsymbol{x}_{i+l_w-l_a}], & k \le \left\lfloor \frac{l_a}{l_s} \right\rfloor + 1 \end{cases} \tag{8}$$

For any $\boldsymbol{x}^{(k)}$, we can find $\hat{\boldsymbol{x}}^{\left(k+\left\lfloor \frac{l_a}{l_s} \right\rfloor\right)}$ or $\hat{\boldsymbol{x}}^{\left(k+\left\lfloor \frac{l_a}{l_s} \right\rfloor+1\right)}$that is closest to $\boldsymbol{x}^{(k)}$ in $\hat{\boldsymbol{x}}$. We denote $LDP(\cdot,\cdot)$ as the length of different part for two frames. Then we can get:

$$LDP_1 = LDP\left(\boldsymbol{x}^{(k)}, \hat{\boldsymbol{x}}^{\left(k+\left\lfloor \frac{l_a}{l_s} \right\rfloor\right)}\right) = l_a \bmod l_s \tag{9}$$

$$LDP_2 = LDP\left(\boldsymbol{x}^{(k)}, \hat{\boldsymbol{x}}^{\left(k+\left\lfloor \frac{l_a}{l_s} \right\rfloor+1\right)}\right) = l_s - (l_a \bmod l_s) \tag{10}$$

$$LDP_{\min} = \min\left(LDP_1, LDP_2\right) \tag{11}$$

where $LDP_{\min}$ is the minimal length of different part for $\boldsymbol{x}^{(k)}$ and is the frame which is closest to $\boldsymbol{x}^{(k)}$ in $\hat{\boldsymbol{x}}$.

LDP_{min} can measure the degree to which the value of features changes after frame offsets with different ϵ. **Fig. 4** shows that the value of LDP_{min} via different frame offsets. It presents a certain periodicity, which means appending more silence frame may make no sense. In this work, the range of ϵ is set to [0,1]. When $\epsilon = \frac{1}{2}$, our method could obtain a best performance.

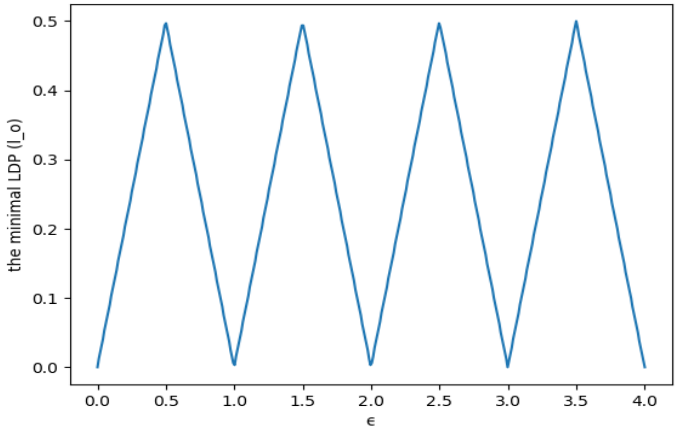


Fig. 4. $LDP_{\min}$ via different frame offsets ($l_a = \lfloor \epsilon * l_s \rfloor$)

4. Experimental Results

4.1 Experimental Setup

4.1.1 Attack Method

For speech-to-label task, we evaluate on the genetic algorithm-based attack (**GA**). The Commander Song attack (**CommanderSong**) and the optimization-based attack (**OPT**) are evaluated for speech-to-text task. In our experiment, the audio adversarial samples which are generated by these attack methods are sent directly to the ASR system.

Different attack methods use different ASR models as the threat model respectively. We used For **GA**, a convolutional speech commands classification model is used as same in [12]. For **CommanderSong** attack, we evaluate the performance on Kaldi speech recognition platform. For **OPT** attack, we use DeepSpeech which is a biRNN based speech-to-text transcription network.

GA: GA is a state-of-the-art speech-to-label attack proposed in [12]. Here an audio classification model is attacked and the output consists of 10 different labels. They aimed to attack such a network to misclassify an adversarial audio based on either targeted or untargeted attack.

CommanderSong: CommanderSong [14] is a speech-to-text targeted attack which use songs as the original audio. The adversarial audio can even be played over the air with its adversarial characteristics. Since the source codes of CommanderSong are not available, we evaluate on the generated adversarial audios provided by the authors.

OPT: We consider the targeted speech-to-text attack proposed by [13], which uses CTC-loss in a speech recognition system as an objective function and solves the task of adversarial attack as an optimization problem.

4.1.2 Dataset

Speech Commands: Speech Commands dataset contains 65000 audio files. Each audio is a single command and has one second duration. In this work, we choose 10 types commands. the commands are “yes”, “no”, “up”, “ own”, “left”, “right”, “on”, “off”, “stop”, and “go”.

LibriSpeech: LibriSpeech is a corpus of approximately 1000 hours of 16Khz English speech. The data is derived from read audiobooks from the LibriVox project. In this work, we use its test-clean dataset in their website [22].

Common Voice: Common Voice is a free audio dataset provided by Mozilla for ASR system. This dataset is public and contains samples from human speaking audio files. In this work, we used its subset which is 16Khz-sampled and has 3.998s average duration. The dataset can be found in [23].

Timit: Timit dataset contains 6300 audio files and consists of only 10 sentences. Each sentence is 30 seconds long and is spoken by 630 different speakers. In this work, wo use its first 100 sample to generate adversarial audio.

4.1.3 Compared Method

For defense method, **Down Sampling**, **Local Smoothing** and **Quantization** are considered here. For detection method, a novel method using temporal dependency method (**TD Method**) is considered.

Down Sampling: Based on sampling theory, it is possible to down-sample a band-limited audio file without sacrificing the quality of the recovered signal while mitigating the

adversarial perturbations in the reconstruction phase. In this work, we first down-sample the original 16kHz audio to 8kHz, then up-sample the audio to 16kHz again.

Local Smoothing: We use a sliding window with a fixed length for local smoothing to reduce the adversarial perturbation. Given an audio sample x, we consider the $2K+1$samples window which is denoted by $[\boldsymbol{x}_{i-K+1}, \cdots, \boldsymbol{x}_i, \cdots, \boldsymbol{x}_{i+K-1}]$, and replace x_i by the smoothed value (median in our case) of the window.

Quantization: Since the amplitude of adversarial perturbation is usually small in the input space, it could be disrupted by rounding the amplitude of audio sampled data into the nearest integer multiple of q. In this work, we choose q = 256, 512 which obtains the best performance in [18].

TD Method: TD Method [18] is a detection method which can exploits the temporal dependency property of audio data to detect audio adversarial examples.

4.1.4 Evaluation Metrics

$\boldsymbol{ASR_{avg}}$**:** ASR_{avg} is an average value of attack success rate for every type **GA** attack. After applying a defense method to input samples, ASR_{avg} will decrease. A low ASR_{avg} means that the defense method has good performance.

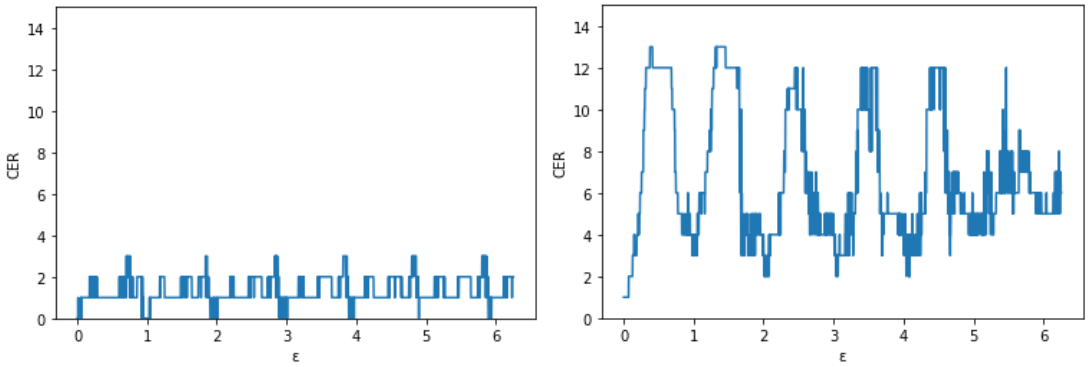
$\boldsymbol{R_{benign}}$**/** $\boldsymbol{R_{adv}}$**:** To evaluate the effectiveness of transformation methods and our method against speech-to-text attack, we report the ratio of translation distance between instance and corresponding ground truth before and after transformation**.** R_{benign} is the effectiveness ratio for benign instances. R_{adv} is the similar effectiveness ratio for adversarial audio.

$$R_{benign} = \frac{D(g(T(x_{benign})),y)}{D(g(x_{benign}),y)}, \quad R_{adv} = \frac{D(g(T(x_{adv})),y)}{D(g(x_{adv}),y)} \tag{12}$$

where x_{benign} denotes a benign audio, x_{adv} denotes an adversarial audio, y is a text of the ground truth, $D(\cdot,\cdot)$ denotes the distance function (WER and CER in our case) and $T(\cdot)$ denotes the input transformation function (i.e. down-sampling, quantization, local-smoothing, compressions et al.). Particularly, in our method, $ASC_{\epsilon}(\cdot)$ is used as $T(\cdot)$.

AUC score: For detection task, AUC score is a commonly used metric. It stands for the area under the ROC Curve. In this work, we use CR which is mentioned in Section 3.2.2 as the output probability to calculate AUC score.

4.2 Length of Silence Clip



(a) (b)

Fig. 5. CER via different frame offsets ($l_a = \lfloor \epsilon * l_s \rfloor$): (a) benign sample (b) adversarial sample

In this section, we explore which length silence clip is appropriate to appending to achieve a better defending performance. We first evaluate on DeepSpeech by taking a benign sample and using OPT attack to generate an adversarial sample, then keep appending silence clip at the beginning of the both audios and record the value of CER. **Fig. 5** shows the results of this process. Both of these curves show a certain periodicity and have the same trend with **Fig. 4.** The CER fluctuation range of the benign sample is smaller than the adversarial sample, which indicates that the adversarial examples is more vulnerable to frame offsets.

4.3 Defense Result

In this section, we will measure the performance of our method of defense strategy on three different attacks: **GA**, **CommanderSong** and **OPT**. For comparison, we also measure the performance of input transformation methods which are proposed in [18, 19].

GA: We first evaluate our method on the GA attack [12] which is a speech-to-label type attack. We choose 10 types samples from SpeechCommand dataset. For each type we use the remaining other types as the target to generate adversarial example, every type attack has 50 samples. The average of attack success rate (ASR_{avg}) is 83% without defense. From the source code of the Classification model, we can know the frame stride is 10ms, so we append 5ms ($\epsilon = \frac{1}{2}$) silence clip at the beginning of samples. Finally, by using our method, the ASR_{avg} fall to 5.6% and is lower than input transformation method, the result of all method measured here is listed in **Table 1**, and the detailed results for every type attack of our method shown in **Fig. 6**.

Table 1. Evaluation results for defense with different on Classification

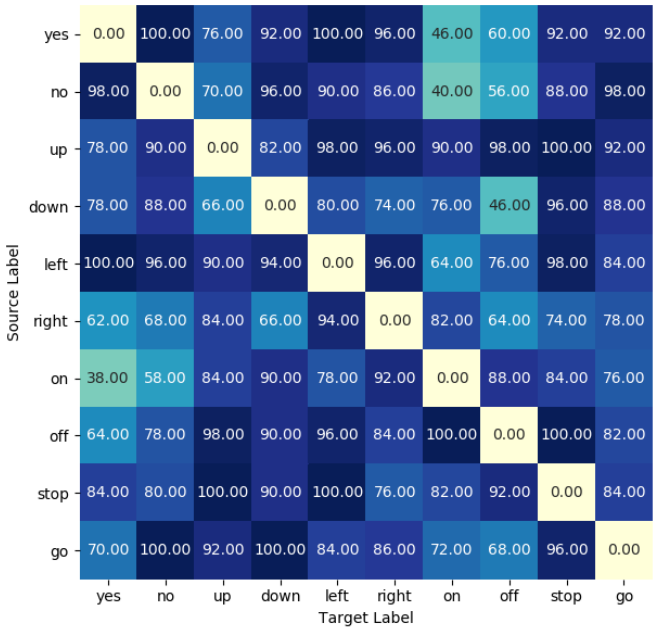
Metric	Without defense	Down Sampling	Smoothing	Quan-256	Quan-512	ASC ($\epsilon = \frac{1}{2}$)
ASR_{avg}	83%	10.2%	20.8%	18.7%	8.2%	**5.6%**

CommanderSong: We also evaluate our method on CommanderSong attack [12]. The dataset we used here can be found in [24]. It contains only 10 samples, 5 samples of which are generated by WTA attack and the other samples are generated by WAA attacks. Due to too few samples, ASR_{avg} cannot be calculated accurately. In **Table 2**, We list the recognition results of some adversarial examples in this dataset before and after using our method. As shown in **Table 2,** our method can effectively defend the **ConmmanderSong** attack.

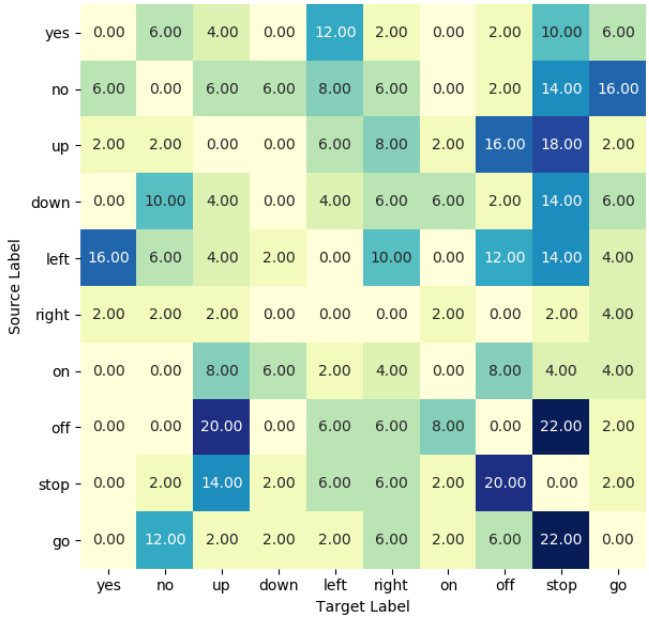
Table 2. Evaluation results for defense on CommanderSong

Song Name	Method	Sentence
Castle in the sky	Without defense	echo open the front door
	ASC （$\epsilon = \frac{1}{2}$）	uh huh
Gold	Without defense	okay google clear notification wild inter cell in marin it's
	ASC （$\epsilon = \frac{1}{2}$）	right they're still in there and it's
Remember the name	Without defense	okay google call one one zero one one nine one two zero manner
	ASC （$\epsilon = \frac{1}{2}$）	okay
Love story	Without defense	okay google turn on g. p. s. h. a. diane reid

	ASC $(\epsilon = \frac{1}{2})$	and uh
A loaded smile	Without defense	okay google good night will oh we're neighbors around i spend
	ASC $(\epsilon = \frac{1}{2})$	but



(a)



(b)

Fig. 6. Attack success rate (a) without defense for every type attack, (b) with our defense method.

OPT: Eventually, we evaluate our defense method on OPT attack (OPT) [13] which is a text-to-speech type attack, and DeepSpeech is used as the victim model. We choose the effectiveness ration for benign instances (R_{benign}) and for adversarial sample (R_{adv}) as the metrics, and experiments on three different datasets (LibriSpeech, CommonVoice and Timit). The results of our method and input transformation method are listed in **Table 3**. The R_{benign} of our method is smaller than other methods and closer to 1.0. This means that the manipulation of our method to audio has little effect on the recognition accuracy of benign instances. **Table 4** gives some other results when ϵ takes other values. When $\epsilon = \frac{1}{2}$, our method achieved the best result, and this coincides with our conjecture in Chapter 3. With ϵ gets bigger, the experimental results start to deteriorate. Particularly, when ϵ up to 1, we get the worst result. Because this is only equivalent to adding 1 frame in front of samples and does not offset the remaining frames. R_{adv} is close to 1 and the recognition result of adversarial example dose not change much. Considering such a countermeasure against our method: if ϵ is fixed, the attacker can add $1-\epsilon$ silence clip to the generated adversarial example and the true ϵ will become 1. Hence, we randomly selected ϵ in the range from $\frac{1}{4}$ to $\frac{3}{4}$, the results show that our method is still effective on this condition.

Table 3. Evaluation results for defense with different on DeepSpeech

Dataset	$\boldsymbol{D(\cdot,\cdot)}$	$\boldsymbol{R_{benign}/R_{adv}}$				
		Down Sampling	**Smoothing**	**Quan-256**	**Quan-512**	**ASC** ($\epsilon = \frac{1}{2}$)
LibriSpeech	WER	1.25 / 0.59	1.70 / 0.61	1.03 / 0.58	1.11 / 0.47	**0.95 / 0.53**
	CER	1.44 / 0.33	2.25 / 0.33	1.09 / 0.31	1.32 / 0.23	**0.95 / 0.26**
Common Voice	WER	1.77 / 0.61	2.14 / 0.59	1.72 / 0.61	2.14 / 0.60	**1.05 / 0.51**
	CER	2.17 / 0.41	2.95 / 0.39	2.10 / 0.38	3.00 / 0.38	**1.11 / 0.30**
Timit	WER	1.29 / 0.94	1.49 / 0.93	1.48 / 0.92	1.84 / 1.01	**0.99 / 0.86**
	CER	1.59 / 0.62	2.03 / 0.60	1.95 / 0.63	2.98 / 0.78	**1.00 / 0.52**

Table 4. Evaluation results for defense with different ϵ on DeepSpeech

Dataset	$\boldsymbol{D(\cdot,\cdot)}$	$\boldsymbol{R_{benign}/R_{adv}}$				
		$\boldsymbol{\epsilon = \frac{1}{3}}$	$\boldsymbol{\epsilon = \frac{1}{2}}$	$\boldsymbol{\epsilon = \frac{3}{4}}$	$\boldsymbol{\epsilon = 1}$	$\boldsymbol{\epsilon = rand(\frac{1}{4}, \frac{3}{4})}$
LibriSpeech	WER	0.93 / 0.62	**0.95 / 0.53**	0.96 / 0.75	1.00 / 1.00	0.93 / 0.62
	CER	0.96 / 0.32	**0.95 / 0.26**	0.96 / 0.44	1.00 / 0.97	0.93 / 0.31
Common Voice	WER	1.10 / 0.60	**1.05 / 0.51**	0.96 / 0.74	1.00 / 1.00	1.08 / 0.59
	CER	1.09 / 0.38	**1.11 / 0.30**	0.92 / 0.50	0.94 / 0.98	1.05 / 0.37
Timit	WER	1.00 / 0.92	**0.99 / 0.86**	0.98 / 0.96	0.99 / 1.00	0.99 / 0.90
	CER	1.00 / 0.61	**1.00 / 0.52**	0.96 / 0.71	0.99 / 0.99	0.98 / 0.96

4.4 Detection Result

In this section, we will measure the performance of our method for detection on **OPT**. Because the output of the Classification model is only a label that can't be calculated the distance by $D(\cdot,\cdot)$, so **GA** is not considered here. For comparison, we also measure the TD method that is proposed by [18] in the same experimental setting. Different from defense, the metrics used for detection are AUC score. **Table 5** summarizes the evaluation results of detection strategy.

Table 5. Evaluation results for detection with different method on DeepSpeech

Dataset	$\boldsymbol{D(\cdot,\cdot)}$	**AUC score**	
		TD Method	**ASC** $(\epsilon = \frac{1}{2})$
LibriSpeech	WER	0.910	1.00
	CER	0.930	1.00
Common Voice	WER	0.915	1.00
	CER	0.936	1.00
Timit	WER	0.893	1.00
	CER	0.901	1.00

OPT: Here we use AUC score (with WER and CER) as the metric, and evaluate on three different datasets. All the target texts we used is "This is an adversarial example". **Table 5** shows that the performance of our method is better than TD Method and the AUC scores in our method are close to 1 no matter on what datasets. **Table 6** gives some other results when ϵ takes other values. When $\epsilon = \frac{1}{3}, \frac{1}{2}, \frac{3}{4}$, the AUC score of our method is close to 1. Particularly, when ϵ up to 1, we get the worst result.

Table 6. Evaluation results for detection with different ϵ on DeepSpeech

Dataset	$\boldsymbol{D(\cdot,\cdot)}$	**AUC score**				
		$\boldsymbol{\epsilon = \frac{1}{3}}$	$\boldsymbol{\epsilon = \frac{1}{2}}$	$\boldsymbol{\epsilon = \frac{3}{4}}$	$\boldsymbol{\epsilon = 1}$	$\boldsymbol{\epsilon = rand(\frac{1}{4}, \frac{3}{4})}$
LibriSpeech	WER	1.00	1.00	1.00	0.91	1.00
	CER	1.00	1.00	1.00	0.91	1.00
Common Voice	WER	1.00	1.00	1.00	0.82	1.00
	CER	1.00	1.00	1.00	0.85	1.00
Timit	WER	1.00	1.00	1.00	0.71	1.00
	CER	1.00	1.00	1.00	0.75	1.00

5. Conclusion

In this work, we proposed a simple, generic and efficient method against audio adversarial attack. For different scenarios, we give a variety of usage strategies (defend, detect and hybrid strategy) of our method. Evaluating on three state-of-the-arts adversarial attacks against on different ASR systems respectively, the results demonstrate that the proposed method can effective improving the robust of audio systems.

Although our method can detect and defend the audio adversarial example, the perturbations still exist as noise which will affect the recognition accuracy of ASR system. The future work should focus on recovering the ground truth of original audio from adversarial audio. One possible way is to combine our method with some denoising methods. First, by using our method, the perturbations will degenerate into ordinary noise. Then, denoising methods can be used to reduce the noise.

For attack methods in this work, the audio samples are sent directly to the ASR system without playing in physical world. With the deepening of research in this field, future attack methods will be able to generate audio adversarial examples that are effective in physical world. Hence, the physical scene should also be considered in corresponding countermeasures.

References

[1] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. of the 3rd International Confernce at Learn. Represent(ICLR)*, 2015. Article (CrossRef Link)

[2] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. of IEEE Symposium on Security and Privacy(SP)*, pp. 39-57, 2017. Article (CrossRef Link)

[3] S. Sun, C. F. Yeh, M. Ostendorf, M. Y. Hwang, and L. Xie, "Training augmentation with adversarial examples for robust speech recognition," in *Proc. of Annuual Conference International Speech Communncation Association(INTERSPEECH)*, pp. 2404-2408, 2018. Article (CrossRef Link)

[4] S. Latif, R. Rana, and J. Qadir, "Adversarial machine learning and speech emotion recognition: utilizing generative adversarial networks for robustness," *arXiv Prepr. arXiv1811.11402*, 2018. Article (CrossRef Link)

[5] C. Yang, J. Qi, P. Chen, X. Ma, and C. Lee, "Characterizing speech adversarial examples using self-attention U-Net enhancement," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP)*, pp. 3107-3111, 2020. Article (CrossRef Link)

[6] S. Samizade, Z. Tan, C. Shen and X. Guan, "Adversarial example detection by classification for deep speech recognition," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP)*, pp. 3102-3106, 2020. Article (CrossRef Link)

[7] K. Rajaratnam, K. Shah, and J. Kalita, "Isolated and ensemble audio preprocessing methods for detecting adversarial examples against automatic speech recognition," in *Proc. of the 30th Conference on Computational Linguistics and Speech Processing(ROCLING)*, pp. 16-30, 2018. Article (CrossRef Link)

[8] P. Ma, S. Petridis, and M. Pantic, "Detecting adversarial attacks on audio-visual speech recognition," *arXiv Prepr. arXiv1912.08639*, 2019. Article (CrossRef Link)

[9] T. N. Sainath and C. Parada, "Convolutional neural networks for small-footprint keyword spotting," in *Proc. of INTERSPEECH 2015*, pp. 1478-1482, 2015. Article (CrossRef Link)

[10] Kaldi. [Online]. Available: https://kaldi-asr.org

[11] DeepSpeech. [Online]. Available: https://github.com/mozilla/DeepSpeech

[12] M. Alzantot, B. Balaji, and M. Srivastava, "Did you hear that? Adversarial examples against automatic speech recognition," *arXiv Prepr. arXiv1801.00554*, 2018. Article (CrossRef Link)

[13] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *Proc. of 2018 IEEE Symposium on Security and Privacy Workshops*, pp. 1-7, 2018. Article (CrossRef Link)
[14] X. Yuan, Y. Chen, Y. Long, X. Liu, K. Chen, H. Huang, and X. Wang, "CommanderSong: A systematic approach for practical adversarial voice recognition," in *Proc. of the 27th USENIX Security Symposium*, pp. 49-64, 2018. Article (CrossRef Link)
[15] X. Liu, K. Wan, Y. Ding, X. Zhang, and Q. Zhu, "Weighted-sampling audio adversarial example attack," *AAAI Technical Track: Machine Learning*, vol. 34, no. 04, pp. 4908-4915, Apr. 2020. Article (CrossRef Link)
[16] Y. Qin, N. Carlini, I. Goodfellow, G. Cottrell, and C. Raffel, "Imperceptible, Robust, and targeted adversarial examples for automatic speech recognition," in *Proc. of the 36th International Conference on Machine Learning(ICML)*, pp. 9141-9150, 2019. Article (CrossRef Link)
[17] L. Schonherr, K. Kohls, S. Zeiler, T. Holz, and D. Kolossa, "Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding," in *Proc. of Network and Distributed Systems Security(NDSS) Symposium*, 2019. Article (CrossRef Link)
[18] Z. Yang, P. Y. Chen, B. Li, and D. Song, "Characterizing audio adversarial examples using temporal dependency," in *Proc. of the 7th International Conference on Learning Represent(ICLR)* 2019. Article (CrossRef Link)
[19] H. Kwon, H. Yoon, and K. W. Park, "Poster: Detecting audio adversarial example through audio modification," in *Proc. of the ACM Conference on Computer and Communications Security*, pp. 2521-2523, 2019. Article (CrossRef Link)
[20] R. Yang, Z. Qu, and J. Huang, "Detecting digital audio forgeries by checking frame offsets," in *Proc. of the 10th ACM Workshop on Multimedia and Security*, 2008. Article (CrossRef Link)
[21] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics Doklady*, vol. 10, no. 8, pp. 707-710, 1966. Article (CrossRef Link)
[22] Open SLR. [Online]. Available: http://www.openslr.org/12
[23] Audio Adversarial Examples. [Online]. Available: https://nicholas.carlini.com/code/audio adversarial_examples
[24] Kaichen. [Online]. Available: http://kaichen.org

Yongkang Gong received the B.S. degree from Ningbo University, in 2018. He is currently pursuing the master's degree with Ningbo University. His research interests include adversarial machine learning and information security.

Diqun Yan is currently an Associate Professor at the Faculty of Electrical Engineering and Computer Science in Ningbo University, China. He is the head of Computer Science Department. He received the B.S., M.S., and Ph.D. degrees from Ningbo University in 2002, 2008, 2012, respectively. He was a Visiting Scholar with New Jersey Institute of Technology, Newark, NJ, USA, from 2014 to 2015. His current research interests include speech processing and multimedia forensics.

Terui Mao received the B.S. degree from Zhejiang Wanli University, in 2019. He is currently pursuing the master's degree with Ningbo University,. His research interests include deeping machine learning and information forensics.

Donghua Wang received the B.S. degree from Jiangxi Normal University, in 2018. He is currently pursuing the master's degree with the Faculty of Electrical Engineering and Computer Science, Ningbo University. His research interests include adversarial machine learning and information security.

Rangding Wang is currently a full professor at Ningbo University, China. He received the Ph. D. from Tongji University in 2004. His research interests mainly include multimedia security, digital watermarking for digital rights management, data hiding, and steganography.