Adversarial Music: Real world Audio Adversary against Wake-word Detection Systems

10-708A S19 Final Report

Billy Li (junchenl) [1] **Bingqing Chen (bingqinc)** [1] **Zhuoran Zhang (zhuoran1)** [1]

Abstract

Voice Assistants (VAs) such as Amazon Alexa, Google Assistant rely on *wake word detection* to respond to people's commands, which could potentially be vulnerable to audio adversarial examples.

In this work, we target our attack on the wake-word detection system, and our goal is to jam the model with some inconspicuous background music, so as to deactivate the VAs while our audio adversary is present. We reverse-engineered the wake-word detection system used in Amazon Alexa based on recent publications. We trained emulated models with different assumptions and tested against the real Alexa in terms of wake-word detection accuracy to measure the fidelity of our models. Then we computed our audio adversaries with consideration of Expectation of Transformation and we implemented our audio adversary with a differentiable synthesizer. Next, we verified our audio adversaries digitally on hundreds of samples of utterances collected from the real world, we can effectively reduce the recognition accuracy of our emulated model from 86% to 12%. Finally, we test our audio adversary over the air, and verified it works reasonably well against Alexa.

1. Introduction

Adversarial attacks on machine learning systems are a topic of growing importance. As machine learning becomes ever more present in all aspects of modern life, concerns about safety tend to also gain prominence. As such, recent demonstrations of the easiness with which machine learning systems can be "fooled" have caused a strong impact in the field and in the general media. Systems that use voice and audio such as Amazon Alexa, Google Assistant, and Microsoft Cortana are growing in popularity. The hidden risk of those advancements is that those systems are potentially vulnerable to adversarial attacks from an ill-intended third-party. Despite the recent growth in consumer presence of audio-based artificial intelligence products, compared to the image and language domains, attacks on audio and speech systems have received much less attention so far.

Despite a number of works recently attempting to create adversarial examples against ASR systems Carlini & Wagner (2018); Schonherr et al. (2018); Qin et al. (2019), robust playable-over-the-air real-time audio adversary against ASR system still does not exist. Meanwhile, there exists no adversary that can be played from a different speaker rather than the source. Moreover, Voice Assistants (VAs) such as Amazon Alexa, Google Assistant are well-maintained by the infrastructure teams, which enable them to retrain and redeploy a new model weekly on their cloud back-end. A Robust audio adversary that can consistently work against these ASR systems are almost impossible to craft not only due to lack of knowledge of the backend models' gradients, but also due to the challenging nature of the task.

However, all the existing VAs rely on wake word (WW) detection to respond to people's commands, which could potentially be vulnerable to audio adversarial examples. In this work, rather than directly attacking the general ASR models, we target our attack on the WW detection system. WW detection models always have to be stored and executed on-board within a smart-home hardware which is usually very limited in terms of computing power. Besides, updates to the model is infrequent and way more difficult. Thus, our proposed attack could be particularly more damaging. Our goal is to jam the model so as to deactivate the VAs while our audio adversary is present. Specifically, we create a parametric attack that resembles a piece of background music, making the attack inconspicuous to humans.

We reverse-engineered the wake-word detection system used in Amazon Alexa based on latest publications on the architecture (Wu et al., 2018). We collected 100 samples of "Alexa" utterances from 10 people and augmented the data set to 20x by varying the tempo and speed. We created a synthetic data set using publicly available data sets as background noise and negative speech examples. We created a

synthetic dataset by adding "Alexa" and other utterances onto background noises. This collected database is used to train and validate our emulate model. We trained emulated models with different configurations and evaluated over the test set.

We implemented two types of attack. One approach is the vanilla projected gradient descent (PGD), which allows the attack model to modify the raw audio sequence in arbitrary way within the allowable frequency band. The other attack is parameterized by our threat model, PySynth Doege (2013), a music synthesizer. Such threat model disguises our attack in a sequence of inconspicuous background music notes.

Here are our main contributions:

1. *We create a threat model in audio domain that allows us to disguise our adversarial attack as a piece of music playable over the air in the physical space.*
2. *In order to make our adversarial example work in the physical world, we took the expectation of transform from digital audio to physical sound into account. We considered psychoacoustic effects in human hearing perception, we also considered room impulse response.*
3. *Our adversarial attack is jointly optimizing the attack nature while fitting the threat model to the perturbation achievable by the microphone hearing range of Amazon Alexa, this is challenging since our attack budget is very limited compared with previous works.*
4. *Our adversarial attack works reasonably well in the real world separate source setting, which is the first real-time attack against Alexa to our knowledge.*

2. Related Works

Most current adversarial attacks work by trying to find a way to modify a given input (hopefully by a very small amount) in such a way that the machine learning system's proper functioning is disrupted. A classic example is to take an image classifier and modify an input with a very small perturbation (difficult for human to tell apart from original image) that still changes the output classification to a completely distinct (and incorrect) one.

To achieve such a goal, the general idea behind many of the attack algorithms is to optimize an objective that involves maximizing the likelihood of the intended (incorrect) behavior, while being constrained to a small perturbation. For differentiable systems such as deep networks, which are the current state of the art for many classification tasks, utilizing gradient-based methods is a common approach. We describe such methods and their relation to our work in more depth in Section 3.2. In this work, our target of attack would be WW systems.

Adversarial attacks were initially introduced for images Szegedy et al. (2013) and have been studied the most in the domain of computer vision (Nguyen et al., 2015; Kurakin et al., 2016; Moosavi-Dezfooli et al., 2016; Elsayed et al., 2018). Following successful demonstrations in the vision domain, adversarial attacks were also successfully applied to natural language processing (Papernot et al., 2016; Ebrahimi et al., 2018; Reddy & Knight, 2016; Iyyer et al., 2018; Naik et al., 2018). This trend gives rise to defensive systems such as (Cisse et al., 2017; Wong & Kolter, 2018), and thus provides a guideline to the community about how to build robust machine learning models.

However, attacks on audio and speech systems have received much less attention so far. Only as recently as last year, Zhang et al. (2017) did a pioneering proof-of-concept work that proved the feasibility of real-world attacks on speech recognition models. This work, however, had a larger focus on the hardware part of the Automatic Speech Recognition (ASR) system, instead of its machine learning component. Not until very recently, there was not much work done on exploring adversarial perturbation on speech recognition model. Carlini et al. (2016) was the first to demonstrate that attack against HMM models are possible. They claimed to effectively attack based on the inversion of feature extractions. Nevertheless, this work was preliminary since it only showcased a limited number of discrete voice commands, and the majority of perturbations are not able to be played over air. As a follow-up work, Carlini & Wagner (2018); Qin et al. (2019) showcased that curated white-box attack based on adversarial perturbation can easily fool the Mozilla speech recognition system[1]. Again, their attacks would only work in with their special setups and are very brittle in real world. More recently, Schonherr et al. (2018) attempted to psycho-acoustic hiding to improve the chance of success of playable attacks. They claimed to verified their attacks against the Kaldi ASR system, whereas the real-world success rate is still not satisfying, and the adversary itself cannot be played from a different source. Rather than failing to exploit the robust ASR systems, our proposed attack targets at the more manageable Wake Word detection system, and really demonstrates that it can be playable over the air.

Currently, the techniques used in attacking audio/speech systems are very similar to that are used in attacking image/vision system, which is dominantly gradient based attacks. Fast Gradient Sign Method (FGSM) is simple and effective method (Goodfellow et al., 2014). Projected Gradient Descent (PGD) is a more robust and generalizable form

[1]Examples can be found at https://nicholas.carlini.com/code/audio_adversarial_examples

of attack that was first introduced in (Madry et al., 2017). In order to improve the robustness of the attacks, more work is seen going into exploring the universal perturbation (Moosavi-Dezfooli et al., 2017). Meanwhile, there is also a growing effort to explore black-box attack on audio systems (Taori et al., 2018). Our theoretical foundation in this work does not differ much from these previous works, which mostly involves first-order gradient based methods. However, we made a lot of improvements to enable it works in real-time and real-world.

3. Methods

3.1. Baseline Emulate Model

WW detection is the first important step before any interactions with distant speech recognition. However, due to the compacted space of embedded platform and need for quick reflection time, models of WW detection are usually compact and vulnerable to be attacked. Thus, we target our attack on the wake-word detection function.

The architecture of Amazon Alexa was published in (Panchapagesan et al., 2016; Kumatani et al., 2017; Guo et al., 2018), allowing us to emulate the model for white-box attack. We implemented the time-delayed bottleneck highway networks with Discrete Fourier Transform (DFT) features following the details in (Guo et al., 2018), which is the most up-to-date information on the model architecture.

The architecture of the emulate model is shown in Figure 1. The model contains a 4-layer highway block as feature extractor, a linear layer acting as the bottleneck, a temporal context window that concatenates features from adjacent frames, and a 6-layer highway block for classification. Finally, we use a cross-entropy loss for classification.

Highway networks were proposed in (Srivastava et al., 2015) as an effective way to deal with the vanishing gradient problem common in deep neural networks. The output of layer l in the highway block can be expressed by two gating functions, as shown by Eq. 1.

$$\mathbf{h}_l = f(\mathbf{h}_{l-1})T(\mathbf{h}_{l-1}) + \mathbf{h}_{l-1}C(\mathbf{h}_{l-1}) \tag{1}$$

The carry (C) and transform (T) gate functions are defined by a nonlinear layer with Sigmoid function, as shown by Eq. 2.

$$\begin{aligned} T(\mathbf{h}_{l-1}) &= \sigma(\mathbf{W}_T\mathbf{h}_{l-1} + \mathbf{b}_T) \\ C(\mathbf{h}_{l-1}) &= \sigma(\mathbf{W}_C\mathbf{h}_{l-1} + \mathbf{b}_C) \\ f(\mathbf{h}_{l-1}) &= \sigma(\mathbf{W}_l\mathbf{h}_{l-1} + \mathbf{b}_l) \end{aligned} \tag{2}$$

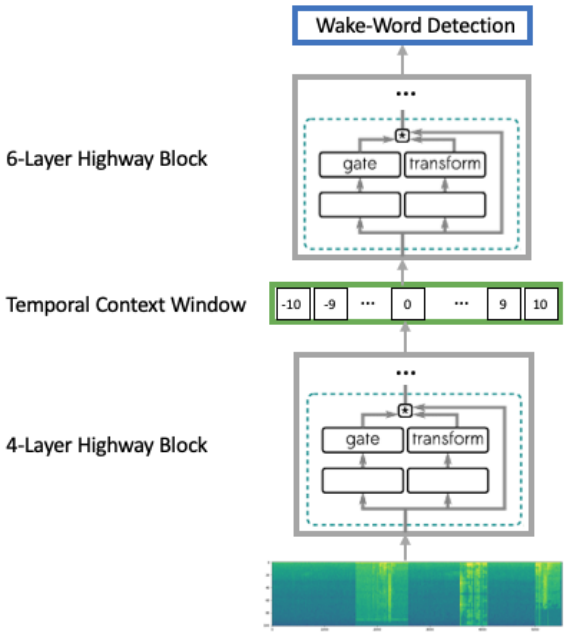

Figure 1. Wake-word Detection Network Architecture (Guo et al., 2018)

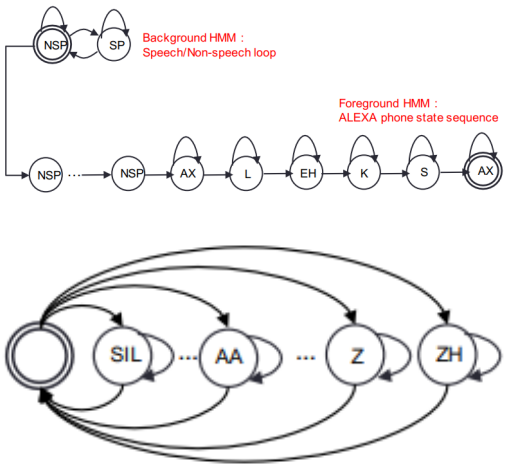

Figure 2. Hidden Markov Model for Speech/Non-speech detection

3.2. Projected Gradient Descent for Adversarial Attacks

Normally, classification problems are formulated as a minimization of $E_{x,y\sim D}[L(f(x), y)]$ where L is the loss function, f is the classifier mapping from input x to label y, and D is the data distribution. We evaluate the quality of our classifier based on the loss, and a smaller loss usually indicates a better classifier. However, this standard formulation could be vulnerable against a perturbed input x', and thus we need a more stringent formulation of classification.

We form $E_{x,y\sim D}[\max_{x'\in C(x)} L(f(x'), y)]$, where $C(x)$ is our predefined perturbation set which we injected perturbation but did not change the true label. In order to learn such a robust classifier, we still try to minimize the empirical loss, and the only difference is there is perturbation applied: $\min_\theta \frac{1}{n}\sum_{i=1}^{n}[\max_{x'\in P(x_i)} L(f_\theta(x'), y_i)]$. This formulation brings about a mini-max problem, but since we are focusing on attack in this work, we only focus on the inner maximization.

We are thus looking to find an example x' that maximizes the loss of the classifier, that is $\max'_x L(f(x'), y)$. In a completely differentiable system, an immediately obvious initial approach to this would be to use gradient ascent in order to search for an x' that maximizes this loss.

However, for this maximization to be interesting both practically and theoretically, we need x' to be close to the original datapoint x, according to some measure. It is thus common to define a perturbation set $C(x)$ that constrains x', such that the maximization problem becomes $\max_{x'\in C(x)} L(f(x'), y)$. The set $C(x)$ is usually defined as a ball of small length (of either ℓ_∞, ℓ_2 or ℓ_1) around x.

Since we have to solve such a constrained optimization problem, we cannot simply apply the gradient descent method to maximize the loss, as this could take us out of the constrained region. One of the most common methods utilized to circumvent this issue is called Projected Gradient Descent (PGD). To conform to the usual literature on gradient descent methods, we will invert the sign of the aforementioned problem to write it as a minimization, *i.e.*, $\min_{x'\in C(x)} -L(f(x'), y)$.

The constrained maximization problem described above can be rewritten as the unconstrained problem $\min'_x -L(f(x'), y) + I_{C(x)}(x')$, where $I_{C(x)}$ is the indicator function on $C(x)$, with value ∞ outside the set $C(x)$ and 0 inside it. Since we have a differentiable function L and a function $I_{C(x)}$, we can frame PGD as a case of proximal gradient descent. We thus have, for step size t,

$$\text{prox}_t(x') = \underset{z}{\text{argmin}} \|x' - z\|^2 + I_{C(x)}(z) \qquad (3)$$

$$\text{prox}_t(x') = \underset{z\in C(x)}{\text{argmin}} \|x' - z\|^2 = P_{C(x)}(x') \qquad (4)$$

where $P_{C(x)}$ is the projection operator onto $C(x)$. With this proximal operator, our proximal gradient descent update step is defined by $x'_+ = P_{C(x)}(x' + t\nabla L)$, that is, we first take a gradient step, then project onto the set $C(x)$. In sum, such an optimization procedure allows us to search for inputs x', constrained to be in the set $C(x)$ near x, that cause the machine learning system to produce an output y with high loss.

3.3. Psychoacoustic Model

Our ultimate task is to deceive the voice assistant with voice that similar to human hearing. So the definition of the similarity between our modified sound and the original sound should be consistent to how humans perceive various sounds, which is the psychoacoustic definition. The principles of the psychoacoustic model are similar to what used in the compression process of audio files, e.g. compress the loose-less file format "wav" to the loosely file format "mp3". In this process, the information carried by the audio file is changed while human's ears is hard to tell the differences between these two sounds.

Specifically, a louder signal (the "masker") can make other signals at nearby frequencies (the "maskees") imperceptible (Lingaiah, 2004). When we add an perturbation δ, the normalized PSD estimate of the perturbation $\bar{p}_\delta(k)$ is under the frequency masking threshold of the original audio $\eta_x(k)$,

$$\bar{p}_\delta(k) = 96 - \max_k p_x(k) + p_\delta(k) \qquad (5)$$

where $p_\delta(k) = 10\log_{10}|\frac{1}{N}s_\delta(k)|^2$, $p_x(k) = 10\log_{10}|\frac{1}{N}s_x(k)|^2$ are power spectral density estimation of the perturbation and the original audio input. $s_x(k))$ is the k_{th} bin of the spectrum of frame x. This results in the loss function term:

$$\ell_\eta(x, \delta) = \frac{1}{|\frac{N}{2}|+1}\sum_{k=0}^{|\frac{N}{2}|}\max\{\bar{p}_\delta(k) - \eta_x(k), 0\} \qquad (6)$$

Exploiting the psychoacoustic model of human hearing allows the adversary to be injected into the signal at so-called *critical bands* which, when present with other frequency components, are inaudbile to the listener. Since speech is dynamically changing throughout the temporal domain, the psychoacoustic analysis is typically carried out on short segements, or frames, of audio. The raw waveform x is segmented into N frames of length L given as

$$x^n(kT) = x(kT + nL)w_L(t - nL)\ k \in [0, N-1] \qquad (7)$$

where n is the frame index and w_L is a window function.

The psycho-acoustic model used is based on the MPEG-ISO (ISO/IEC-11172-3:1993) and was included in attacks presented in (Schonherr et al., 2018; Qin et al., 2019). We will not explain in detail the weighting generated by the psycho acoustic model and were refer the reader to (Painter & Spanias, 2000; Defraene et al., 2012; Carlini & Wagner, 2018) but it consists of 5 steps:

1. The power spectral density (PSD) of the signal is normalized to standard sound pressure level (SPL). This is important as the signals have various amplitudes based on room dynamics, microphone responses and so forth.

2. Once the PSD is estimated and the signal has been normalized the so-called tonal and non-tonal maskers are identified. These maskers represent parts of the human auditory system that will mask other frequencies when presented simultaneously.

3. The number of maskers is then reduced, or decimated, by comparing the tonal and non-tonal maskers with a sliding window scheme.

4. The individually masking thresholds and then used to generate a masking pattern that encompasses the adjacent frequencies affected by the tonal and non-tonal maskers.

5. The global masking threshold is the then found by combining the masking thresholds from the previous stem. This global threshold then represents a perceptual weighting that is based on the power and frequency components of the signal and the psychoacoustic properties of the human auditory system.

The resulting global masking threshold t can then be found for each frame N and frequency f, $t_n(f) \in [0, \frac{f_s}{2}]$, where f_s is the sampling frequency.

Figure (3) shows the absolute threshold of hearing compared to that of the global masking threshold calculated for a single frame. In a normal environment, the human auditory has a peak response between roughly 3kHz and 4kHz. This means in a quiet environment that if an adversary were to be added around these frequencies with a level greater than -5 dB SPL (less than the sound of light leaf rustling) it would be perceptible. However, if other tones are present are presented simultaneously the create a masking effect which allows for a higher amount of noise to be added that would be imperceptible. In the case of Figure (3), by exploiting the the sounds present in analysis frame, noise can be added to higher than 40 dB SPL in some bands which is equivalent to a normal conversational level of sound.

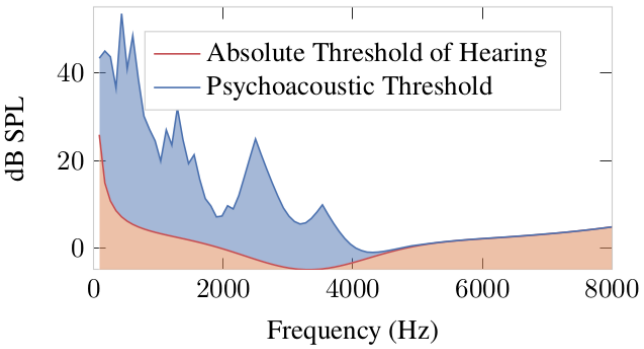

Figure 3. The absolute threshold of hearing compared to the threshold of speech frame where a psychoacoustic model is used to extract a global masking threshold.

3.4. Expectation of Transform

When using the voice assistant in a room, the real sound caught by the microphone includes both the original sound speaked by human and the reflected sound. The "room impulse response" function explained the transform of original audio and the audio caught by the microphone. Therefore, to make our adversarial attack effective in the physical domain, i.e. attack the voice assistant over the air, it is necessary to consider the room impulse response in our work.

we use the classic Image Source Method introduced in (Allen & Berkley, 1979; Scheibler et al., 2018) to create the room impulse response r based on the room configurations (dimension, source location, reverberation time).

$$t(x) = x \times r \tag{8}$$

Here, x denotes clean audio and $*$ denotes convolution operation. The transformation function t follows a chosen distribution $\mathcal{T}$ over different room configurations.

3.5. Adversarial Audio Synthesizer(Threat Model)

To perform the adversarial attack on the audio domain, we introduce a parametric model to define a realistic construction of our adversary δ_θ parameterized by θ.

We use a music synthesizer, Pysynth(Doege, 2013) as our parametric attack model. We used karplus-strong algorithm to synthesis guitar-timbre sound. The mechanism of the algorithm could be referred to Sullivan (1990). The gist is to generate a sequence of guitar notes with given BeatsPerMinute(BPM), Sampling Frequency, and volume, and the algorithm could be regarded as an addition of sine waves. We implemented this algorithm using the auto-differentiation toolkit PyTorch. All of these parameters are referred to as θ, they are used to generate this piece of mu-

sic, and they are optimizable to generate our adversarial music.

3.6. Loss Formulation

To attack the voice assistant in the digital domain, one can use the method illustrated in the Section 3.2 to solve the constrained maximization optimization problem. The loss function of the vanilla PGD in the is :

$$\max_{z \in C(x)} \ell(x, y) = \ell_{net}(f(x')) - \alpha \|x' - z\|^2 \quad (9)$$

In the audio domain, we aim at attack the voice assistant via parametric model and considering psychoacoustic effect, as illustrated in Section 3.3 and Section 3.5. The loss function of attack in audio domain is

$$\max \ell(x, \delta_\theta, y) = \ell_{net} f(x + \delta_\theta) - \alpha \cdot \ell_\eta(x, \delta) \quad (10)$$

Our final loss formulation is attack the voice assistants over the air. This is the attack under the physical condition, we want to maximize it to craft our adversary:

$$\max \ell(x, \delta_\theta, y) = \mathbb{E}_{t \in \mathcal{T}}[\ell_{net} f(t(x + \delta_\theta)) - \alpha \cdot \ell_\eta(x, \delta)] \quad (11)$$

Here, y is the groud truth label of the audio, and ℓ_{net} is the original loss of the emulated model for wake word detection.

4. Experiments

4.1. Datasets

We collected 100 positive speech samples (speaking "Alexa") from 10 peoples (4 males and 6 females; 4 native speakers of English, 6 non-native speakers of English). Each person provided 10 utterances, under the requirement of varying their tone and pitch as much as possible. We further augmented the data to 20x by varying the speed and tempo of the utterance, resulting in 2000 samples.

We used publicly available data [2] for background noise and negative speech examples (speak anything but "Alexa"). We created a synthetic data set by randomly adding positive and negative speech examples onto a 10s background noise and created binary labels accordingly. While "hearing" positive speech examples, we set label values as 1. One sample of the spectrum figure and the corresponding label is shown at Figure 4.

To train the speecn/non-speech detection model, we used large corpus including LibriSpeech ASR corpus (1000 hours) [3], Mozilla common voice corpus (582 hours) [4], Switch Board Corpus (260 hours) [5].

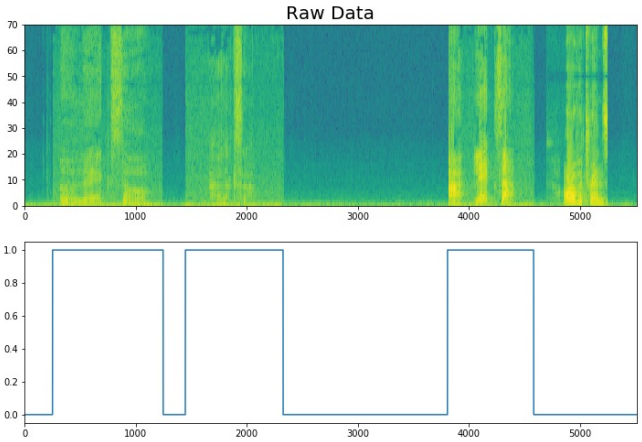

Figure 4. Collected raw data sample

4.2. Baseline Model

Compare to the original implementation in (Guo et al., 2018), we simplified our loss function. We trained the model as a binary classification problem over time sequence, distinguishing between wake-word and non wake-word. The performance is evaluated over a reserved test set. Care has been taken to ensure that augmented copies of the same raw audio sample will not occur in the train set and test set simultaneously. Common performance metrics is listed in Table 1. Figure 5 shows the Detection Error Tradeoff (DET), zoomed in to the same scale as reported in (Guo et al., 2018). The curve is visually comparable to the results in (Guo et al., 2018). Unfortunately, metrics in (Guo et al., 2018) were reported in relative terms compared to other models. This makes it difficult for us to directly benchmark our model against theirs.

4.3. Vanilla Projected Gradient Descent

Our first approach is vanilla projected gradient descent, which allows the attack model to modify the raw audio sequence in arbitrary way, using the approach described in Section 3.2. We use the emulate model developed in Section 4.2 to estimate the gradient and maximizes the classification loss following Eq. 3.2.

The performance metrics of the emulate model on adversary examples are also shown in Table 1. An example of modified adversarial attack example is shown in Figure 6.

While the vanilla PGD approach manages to greatly de-

[2]The LJ Speech Dataset https://keithito.com/LJ-Speech-Dataset/

[3]http://www.openslr.org/12/

[4]https://voice.mozilla.org/en

[5]https://catalog.ldc.upenn.edu/LDC97S62

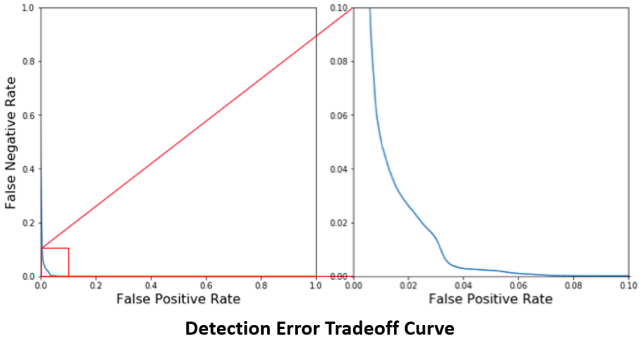

Figure 5. Detection Error Tradeoff Curve

Models	Precision	Recall	F1 Score	AUC
Emulate Model	0.98	0.98	0.98	0.997
Vanilla Attack	0.38	0.28	0.32	0.223
Parametric Attack	0.12	0.10	0.11	0.15

Table 1. Performance of models

crease the detection performance, this attack model is not practical for attack on audio domain. To make the point, we highlight the difference between the raw audio data and the adversarial example, i.e. the change made to the spectrogram by the attack model in Figure 7. The attack model smears the entire spectrogram with emphasis on certain frequency bands. Such approach is effective when we evaluate it over the digital domain, but it is not playable over the air. Furthermore, a noise at a fixed frequency is suspicious, which enable people to identify the source of the attack easily.

4.4. Parametric Adversarial Attack

Using the loss function defined by Equation 10 and parametric method illustrated in Section 3.5, we performed the parametric attack. An example of the parametric attack is provided in Figure 8. It highlights the difference of parametric attack with and without The comparison of parametric adversarial attack in audio domain and the vanilla adversarial attack in digital domain is shown in Table 1.

5. Conclusion and Discussion

As a first step, we created an emulate model for the WW detection on Amazon Alexa following the implementation details published in (Guo et al., 2018). The model is the basis to design our white-box attacks. We collected augmented and synthesize a data set for training and testing. Our model achieved qualitatively comparable performance to that in (Guo et al., 2018) over our test set.

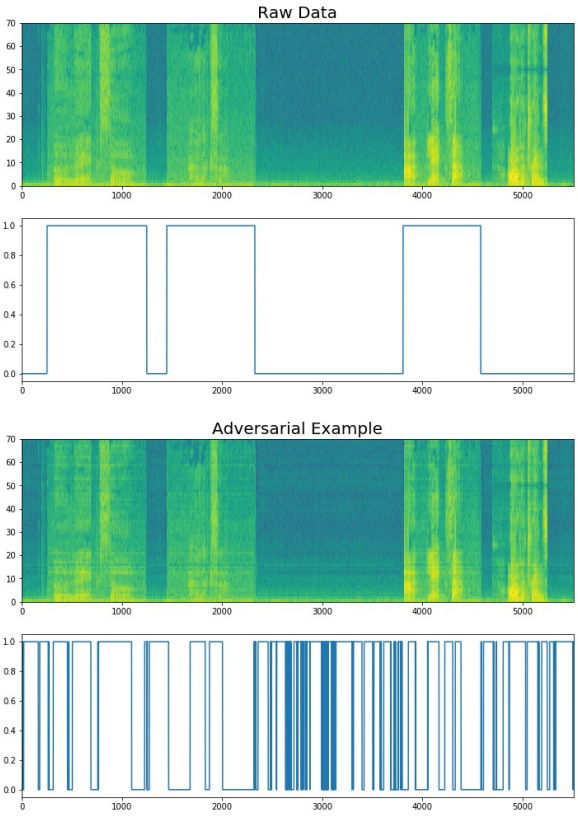

Figure 6. Vanilla Adversarial Attack Example

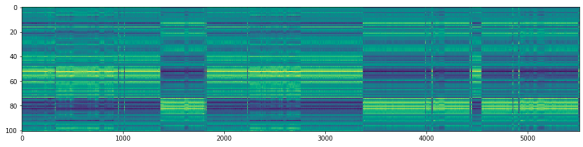

Figure 7. Changes Made by the Attack Model

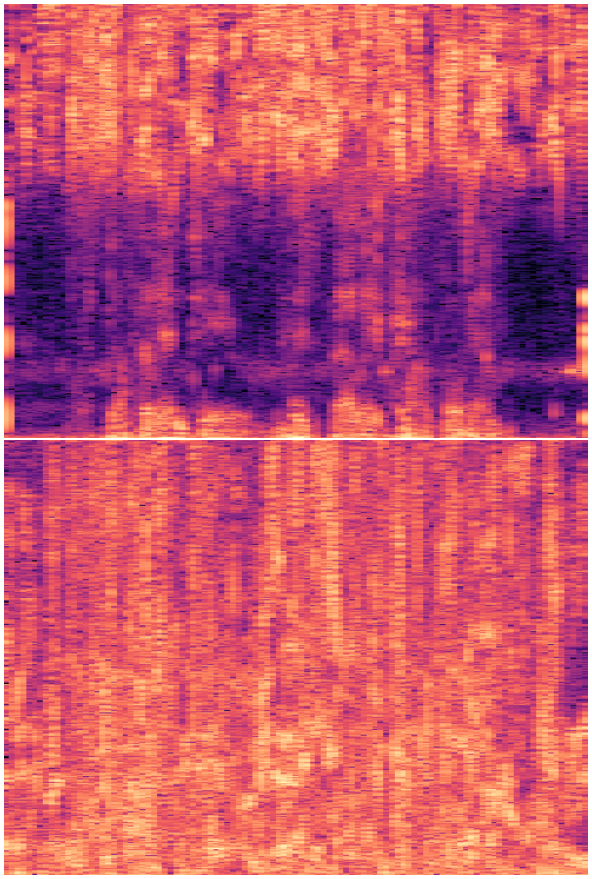

Figure 8. Parametric Adversarial Attack Example, the sample in the upper figure is the one with psychoacoustic effect consideration, and the lower one did not consider it. As we can observe, if consider psychoacoustic masking, the adversary would be much more concentrated in certain frequency bands without smearing the entire spectrum.

We implemented two types of attack. One approach is the vanilla PGD, which allows the attack model to modify the raw audio sequence in arbitrary way within the allowable frequency band. The other attack is parameterized by our threat model, PySynth Doege (2013), a music synthesizer. Such threat model disguises our attack in a sequence of inconspicuous background music notes.

Our work showed the potential of adversarial attack in the audio domain, specifically, for the wake-word detection widely applied in voice assistants. This paper illustrated steps to achieve the goal of attacking in the physical space: reverse-engineed the emulate model, attack from the digital domain, attack from the audio domain, and finally attack in the physical space.

6. Future Work

6.1. Over-the-Air Attack

The ultimate goal is to create a threat model that would allow us to play our audio adversarial example over the air in the physical space. To the best of the authors' knowledge, there is no adversarial attack on ASR that could be played on the air. Our second approach disguises the attack in a sequence of inconspicuous music notes. However, a parametric model alone is not sufficient to achieve the objective of over-the-air attack.

We also need to take into account the transformation from digital audio to physical sound. The adversarial examples will be implemented by jointly optimizing the attack nature while fitting the threat model to the perturbation achievable by the microphone hearing range through verbal command, using the approach described in Section 3.3 and 3.4.

6.2. Black-box Attack

In order to explore attacks on wake-word detection model, we developed an implementation of PGD suitable for applications in the audio domain. However, we were very limited by the number of the samples we can collect for the WW compared to that of Amazon Alexa, and we were also constrained by our heuristic assumptions while emulating the Alexa model. This naturally give rise to a black-box settings where it calls for zero-order methods such as bandit algorithm to tune our perturbation more directly.

References

Allen, J. B. and Berkley, D. A. Image method for efficiently simulating small-room acoustics. *The Journal of the Acoustical Society of America*, 65(4):943–950, 1979.

Carlini, N. and Wagner, D. Audio adversarial examples: Targeted attacks on speech-to-text. *arXiv preprint*

arXiv:1801.01944, 2018.

Carlini, N., Mishra, P., Vaidya, T., Zhang, Y., Sherr, M., Shields, C., Wagner, D., and Zhou, W. Hidden voice commands. pp. 513–530, 2016.

Cisse, M., Bojanowski, P., Grave, E., Dauphin, Y., and Usunier, N. Parseval networks: Improving robustness to adversarial examples. *arXiv preprint arXiv:1704.08847*, 2017.

Defraene, B., van Waterschoot, T., Ferreau, H. J., Diehl, M., and Moonen, M. Real-time perception-based clipping of audio signals using convex optimization. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(10):2657–2671, Dec 2012.

Doege, M. C. Pysynth. https://github.com/mdoege/PySynth, 2013.

Ebrahimi, J., Rao, A., Lowd, D., and Dou, D. Hotflip: White-box adversarial examples for text classification. 2:31–36, 2018.

Elsayed, G. F., Shankar, S., Cheung, B., Papernot, N., Kurakin, A., Goodfellow, I., and Sohl-Dickstein, J. Adversarial examples that fool both human and computer vision. *NIPS*, 2018.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and Harnessing Adversarial Examples. *ArXiv e-prints*, December 2014.

Guo, J., Kumatani, K., Sun, M., Wu, M., Raju, A., Ström, N., and Mandal, A. Time-delayed bottleneck highway networks using a dft feature for keyword spotting. pp. 5489–5493, 2018.

ISO/IEC-11172-3:1993. Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 3: Audio. Standard, International Organization for Standardization, Geneva, CH, 1993.

Iyyer, M., Wieting, J., Gimpel, K., and Zettlemoyer, L. Adversarial example generation with syntactically controlled paraphrase networks. *arXiv preprint arXiv:1804.06059*, 2018.

Kumatani, K., Panchapagesan, S., Wu, M., Kim, M., Strom, N., Tiwari, G., and Mandai, A. Direct modeling of raw audio with dnns for wake word detection. pp. 252–257, 2017.

Kurakin, A., Goodfellow, I., and Bengio, S. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

Lingaiah, D. Introduction to digital audio coding and standards [book review]. *Circuits and Devices Magazine, IEEE*, 20:52– 53, 12 2004. doi: 10.1109/MCD.2004.1364776.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Moosavi-Dezfooli, S.-M., Fawzi, A., and Frossard, P. Deepfool: a simple and accurate method to fool deep neural networks. pp. 2574–2582, 2016.

Moosavi-Dezfooli, S.-M., Fawzi, A., Fawzi, O., and Frossard, P. Universal adversarial perturbations. pp. 1765–1773, 2017.

Naik, A., Ravichander, A., Sadeh, N., Rose, C., and Neubig, G. Stress test evaluation for natural language inference. *arXiv preprint arXiv:1806.00692*, 2018.

Nguyen, A., Yosinski, J., and Clune, J. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. June 2015.

Painter, T. and Spanias, A. Perceptual coding of digital audio. *Proceedings of the IEEE*, 88(4):451–515, April 2000.

Panchapagesan, S., Sun, M., Khare, A., Matsoukas, S., Mandal, A., Hoffmeister, B., and Vitaladevuni, S. Multi-task learning and weighted cross-entropy for dnn-based keyword spotting. pp. 760–764, 2016.

Papernot, N., McDaniel, P., Swami, A., and Harang, R. Crafting adversarial input sequences for recurrent neural networks. pp. 49–54, 2016.

Qin, Y., Carlini, N., Goodfellow, I., Cottrell, G., and Raffel, C. Imperceptible, Robust, and Targeted Adversarial Examples for Automatic Speech Recognition. *arXiv e-prints*, art. arXiv:1903.10346, Mar 2019.

Reddy, S. and Knight, K. Obfuscating gender in social media writing. pp. 17–26, 2016.

Scheibler, R., Bezzam, E., and Dokmanić, I. Pyroomacoustics: A python package for audio room simulation and array processing algorithms. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 351–355. IEEE, 2018.

Schonherr, L., Kohls, K., Zeiler, S., Holz, T., and Kolossa, D. Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding. *arXiv preprint arXiv:1808.05665*, 2018.

Srivastava, R. K., Greff, K., and Schmidhuber, J. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

Sullivan, C. R. Extending the karplus-strong algorithm to synthesize electric guitar timbres with distortion and feedback. *Computer Music Journal*, 14(3):26–37, 1990.

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.

Taori, R., Kamsetty, A., Chu, B., and Vemuri, N. Targeted adversarial examples for black box audio systems. *arXiv preprint arXiv:1805.07820*, 2018.

Wong, E. and Kolter, Z. Provable defenses against adversarial examples via the convex outer adversarial polytope. pp. 5283–5292, 2018.

Wu, M., Panchapagesan, S., Sun, M., Gu, J., Thomas, R., Vitaladevuni, S. N. P., Hoffmeister, B., and Mandal, A. Monophone-based background modeling for two-stage on-device wake word detection. pp. 5494–5498, 2018.

Zhang, G., Yan, C., Ji, X., Zhang, T., Zhang, T., and Xu, W. Dolphinattack: Inaudible voice commands. pp. 103–117, 2017.