AudiDoS: Real-Time Denial-of-Service Adversarial Attacks on Deep Audio Models

Taesik Gong*
KAIST, Republic of Korea
taesik.gong@kaist.ac.kr

Alberto Gil C. P. Ramos
Nokia Bell Labs, UK
alberto.gil_ramos@nokia-bell-labs.com

Sourav Bhattacharya
Nokia Bell Labs, UK
sourav.bhattacharya@nokia-bell-labs.com

Akhil Mathur
Nokia Bell Labs, UK
akhil.mathur@nokia-bell-labs.com

Fahim Kawsar
Nokia Bell Labs, UK and TU Delft
fahim.kawsar@nokia-bell-labs.com

***Abstract*—Deep learning has enabled personal and IoT devices to rethink microphones as a multi-purpose sensor for understanding conversation and the surrounding environment. This resulted in a proliferation of Voice Controllable Systems (VCS) around us. The increasing popularity of such systems is also prone to attracting miscreants, who often want to take advantage of the VCS without the knowledge of the user. Consequently, understanding the robustness of VCS, especially under adversarial attacks, has become an important research topic. Although there exists some previous work on audio adversarial attacks, their scopes are limited to embedding the attacks onto pre-recorded music clips, which when played through speakers cause VCS to misbehave. As an attack-audio needs to be played, the occurrence of this type of attacks can be suspected by a human listener. In this paper, we focus on audio-based Denial-of-Service (DoS) attack, which is unexplored in the literature. Contrary to previous work, we show that adversarial audio attacks in *real-time* and *over-the-air* are possible, while a user interacts with VCS. We show that the attacks are effective *regardless of the user's command and interaction timings*. In this paper, we present a first-of-its-kind imperceptible and always-on universal audio perturbation technique that enables such DoS attack to be successful. We thoroughly evaluate the performance of the attacking scheme across (i) two learning tasks, (ii) two model architectures and (iii) three datasets. We demonstrate that the attack can introduce as high as 78% error rate in audio recognition tasks.**

***Index Terms*—Speech recognition, Voice controllable system, Adversarial attack, Universal adversarial perturbation**

I. Introduction

Deep learning-based classifiers are becoming ubiquitous around us and an increasing number of applications are using some form of deep learning for accurate context inference, e.g., recognizing speech and understanding images. Based on their success, more and more smart devices are becoming equipped with microphones and voice controllable systems (VCS), such as Siri, Alexa, and Google Home, which allow users to control appliances solely using their voice. These audio input based intelligent systems can in principle leverage not only the users' voices but also their surrounding audio context such as ambient scene detection to provide informed contextual services. It is expected that these systems powered by the conjunction of audio input and deep learning technologies will continue to proliferate and augment our daily lives.

Since these systems are triggered by audio inputs, one major concern is that they may be especially vulnerable to unwanted sound inputs generated nearby the system microphone, which could be intended to force the system to activate and trigger an unwanted action. Examples of such unwanted actions include purchase an item online without user's consent, or command a vehicle to accelerate or decelerate unexpectedly with potential life-threatening consequences.

Adversarial attacks in the audio domain focus on the worst form of such vulnerabilities, namely on audio attacks which are almost imperceptible, and therefore may successfully compromise VCS without its user realizing it until it is too late. The principle of adversarial attacks is to fool the model by finding and injecting a small, imperceptible perturbation onto a correctly classified normal input so that the model incorrectly classifies the perturbed input even though humans cannot perceive any significant difference between the original and the corrupted input.

Although impressive, state-of-the-art adversarial attacks on deep audio models [1], [2], [6]–[9], [12], [23] have a restrictive form that first (i) selects a pre-recorded benign audio clip that will be used as an attack carrier in a future attack, and then (ii) computes and injects an adversarial perturbation to the original clip prior to the attack to create another audio file which sounds similar to the average human but which is designed to fool a VCS into activating and producing an unwanted action. Once this is done, an attack can then be carried out by downloading and playing the pre-computed attack near a VCS. Within this setting however, previous work is mostly limited to creating and studying such attacks without actually playing them over-the-air in so called offline analysis [2], [7]–[9], [12], [23]. Recent studies have shown that those attacks could be realized via uninterpretable audio [1], [6] or a song [26]. However, such attacks are limited in that they produce *audible* sounds (with only the attack components being almost imperceptible), namely unintelligible audio or a song, and can therefore be noticed by the user and guarded against by having the user turn off every audible sound source in the vicinity of the VCS.

In this paper we open a new audio adversarial attack scenario space, namely Denial-of-Service (DoS) attacks on

* The work was conducted when this author was in Nokia Bell Labs.

978-1-7281-4550-1/19/$31.00 ©2019 IEEE
DOI 10.1109/ICMLA.2019.00167

VCS, which complements the aforementioned work. Unlike existing attacks, our attack is aimed to instead compromise the users' voice directly without needing any additional noticeable audio having to be played in order to carry out the attack, where imperceptible audio adversarial perturbations get superimposed in real-time and over-the-air with the user's unconstrained interaction with the VCS. Furthermore, our attack method is aimed at working regardless of the user's voice content and the specific time the user decides to interact with the VCS.

To build such DoS attacks, there are two important challenges to resolve, which were not considered in previous work: (i) An attacker has no knowledge about *what* will be spoken from the target user (i.e., the victim). Given this, to be useful in practice, an attack should be applicable to most inputs from the user. (ii) An attacker has no knowledge about *when* the voice will be spoken by the user. Unlike previous work which embeds adversarial attacks onto pre-recorded audio clips, given that in this paper the adversarial perturbation and the legitimate audio are emitted by two separate entities, there is no guarantee that the adversarial perturbation played by an attacker's device and the victim's voice will arrive in a synchronized fashion at the victim's device (i.e., a VCS). We therefore study the timing and lack of synchronization of these two signals in real-time over-the-air and find that it directly affects the performance of the audio task. With these two challenges in mind our goal is to design an audio adversarial attack that is applicable to *any input* from the user, at *any time*.

We propose *AudiDoS*, a DoS attack for audio deep models that meets the aforementioned objectives. Our key insight is that there exist such a "universal" adversarial perturbation for audio inputs that compromises most of users' speech, so that it would make a deep audio model misclassify most of the users' input. More specifically, AudiDoS trains a universal adversarial perturbation in a way that it maximizes the misclassification rate when combined with the possible inputs, while limiting the magnitude of the perturbation to minimize the perceptibility of the attack. The attacker plays this small perturbation continuously near the target deep audio model, which causes the model to incorrectly classify inputs which it would normally classify correctly. In essence, AudiDoS is applicable to any deep audio model without knowing the content (what will be spoken) and the time (when it will be spoken) ahead of the time.

We summarize our main contributions as follows:

- This is the first study that identifies the content and delivery time independence problems that need to be overcome to achieve an effective DoS attack in the audio domain, and that designs a realistic DoS attack for audio models through universal perturbations that work irrespective of what is spoken or when it is spoken.
- Our evaluation across two learning tasks (keyword spotting and speech-to-text problems), two different models (e.g., SoundNet [5] and DeepSpeech2 [3]) and three different datasets (e.g., Speech Commands [24], LibriSpeech [16] and TED-LIUM [19]) in both offline analysis and in-the-wild experiment shows AudiDoS is more effective than the random noise baseline with the same magnitude of the attack.
- Our results indicate that it is possible to create intelligent attacks which greatly outperform baseline attacks based on random perturbations with the same magnitude, i.e., perceptibility level. For instance, when we target SoundNet trained with Speech Commands dataset, the error rate of the model with our attack is 78% error rate in the real-world experiment (the baseline is 48%).

We believe this work opens avenues for the development of more sophisticated audio adversarial attacks and this, in turn, furthers the development of more robust deep audio models to such threats.

II. Background

To describe the main challenges when developing audio adversarial attacks that are effective in real-world settings, in this section we explain adversarial examples and the increasing levels of complexity that arise in developing adversarial attacks: first for software audio adversarial attacks, and then for the setting of this paper namely real-world audio attacks. We also formalize the concepts of: white-box versus black-box attacks and input dependent versus input independent attacks.

A. Adversarial Examples

Consider a multi-class classification problem, where the goal is to learn a mapping between an input $\boldsymbol{x} \in \mathbb{R}^p$ (e.g., an audio clip) and a label $y \in \{0, \dots, m-1\}$ (e.g., whether an audio contains the word 'cat'). For the modeling purpose we use a neural network $f(\boldsymbol{x}; \boldsymbol{\theta}) \in \mathbb{R}^m$ where $\theta \in \mathbb{R}^n$ denotes a vector of parameters and

$$0 < [f(\boldsymbol{x}; \boldsymbol{\theta})]_i < 1, \qquad \sum_{i=0}^{m-1} [f(\boldsymbol{x}; \boldsymbol{\theta})]_i = 1,$$

with the interpretation that the neural network will correctly capture the probability of the various labels matching the given input, i.e., $[f(\boldsymbol{x}; \boldsymbol{\theta})]_i \approx \mathbb{P}(y = i | \boldsymbol{x}, \boldsymbol{\theta})$, and one takes as a neural network label the one corresponding to the maximum probability, i.e., $\hat{y}(\boldsymbol{x}; \boldsymbol{\theta}) := \arg\max_i \{[f(\boldsymbol{x}; \boldsymbol{\theta})]_i\}$. Having trained the neural network, i.e., settled into a particular choice for the parameters $\boldsymbol{\theta}^*$, given an input/label example $(\boldsymbol{x}, y)$ which is correctly classified by the trained neural network, i.e., $\hat{y}(\boldsymbol{x}; \boldsymbol{\theta}^*) = y$. One calls an *adversarial perturbation* [22] any $\boldsymbol{\epsilon} \in \mathbb{R}^p$, dependent or independent of the particular input/label pair, for which:

- most humans cannot perceive any significant difference between the original input $\boldsymbol{x}$ and the corrupted input $\boldsymbol{x}+\boldsymbol{\epsilon}$, but
- the neural network changes its output so that it incorrectly classifies the perceptually indistinguishable input, i.e., $\hat{y}(\boldsymbol{x} + \boldsymbol{\epsilon}; \boldsymbol{\theta}^*) \neq y$.

In this setting, $\boldsymbol{x} + \boldsymbol{\epsilon}$ is referred to as an *adversarial example*. When we adopt adversarial examples to attack a

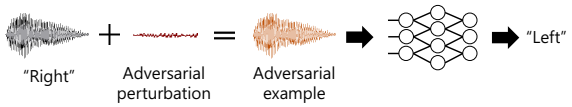

Fig. 1. An illustration of audio adversarial attack.

system based on neural networks, we call them *adversarial attacks*. An illustrative example of an adversarial attack for audio is depicted in Fig. 1, where an adversarial example is fed into the trained neural network to fool it. We magnify the adversarial perturbation for illustration purposes, but note that it would otherwise be imperceptible to humans, i.e., they would not 'hear' any dissimilarity between the original and corrupted audio.

Adversarial perturbations are desired to be as small as possible in some norm as a means to minimize perceptibility of the adversarial examples. In order to create an adversarial example, one may solve for the minimization problem

$$\boldsymbol{\epsilon}(\boldsymbol{x}) := \arg\min_{\boldsymbol{\epsilon}} \|\boldsymbol{\epsilon}\| \text{ s.t. } \hat{y}(\boldsymbol{x}+\boldsymbol{\epsilon};\boldsymbol{\theta}^*) \neq \hat{y}(\boldsymbol{x};\boldsymbol{\theta}^*). \quad (1)$$

As for the norm $\|\cdot\|$ in (1), one can choose any sub-differentiable norm, but most works use the $\|\cdot\|_\infty$ norm (ℓ-infinity norm) for constraining adversarial perturbations [7], [13], [14]. Approximate solutions to the aforementioned and other minimization problems for adversarial perturbations can be obtained in various ways. A well-known approach is via the fast gradient sign method [10] and more generally via projected gradient descent [4].

B. White-Box vs. Black-Box Attacks & Input Dependent vs. Input Independent Attacks

In the aforementioned attack and in similar minimization problems, if the attacker has the complete knowledge about the model architecture and parameters, i.e., knows $f(\boldsymbol{x};\boldsymbol{\theta}^*)$ when performing the minimization, then it is a white-box attack. Otherwise, it is a black-box attack [1], [17], [23].

As an important observation for the remainder of this paper, it is crucial to note that in (1) each attack is tailored to one specific input, so that different inputs would lead to different attacks, i.e., these attacks are input dependent. However, this is not necessary and input-independent attacks can be generated in a way that they are applicable to most possible inputs. We will mainly discuss attacks which are *input independent* throughout the remainder of the paper.

C. Limitations of Existing Audio Adversarial Attacks

Recent studies have began to show the possibilities of applying adversarial attacks to audio inputs [7], [8], [12], [23]. While these studies have revealed some of the threats that adversarial attacks pose to VCS, most of such work has conducted only offline evaluation, i.e., it simulated adversarial attacks in a single machine without playing the adversarial examples over-the-air. In real-world environments where many distorting factors exist, such as reflection, attenuation, absorption of audio signal, analog-to-digital and digital-to-analog conversions, and noise sources, the performance of the proposed attacks remains broadly untested.

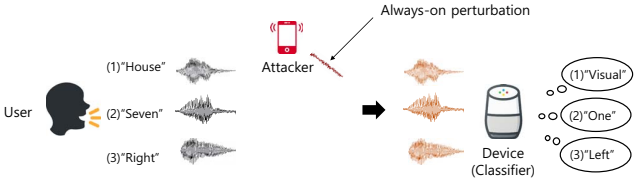

Fig. 2. Illustration of the threat model.

Several studies have realized those attacks by compromising a pre-downloaded benign audio and generating uninterpretable audio [1], [6] or a song [26]. However, when those attacks are played and successfully carried out over-the-air, they are limited in the sense that they produce *audible* sounds (with only the attack components being almost imperceptible), and can therefore be noticed by the user and guarded against by having the user turning off every audible sound source in the vicinity of the VCS.

III. Threat Model

We propose a new audio adversarial attack scenario, namely real-world audio DoS attack. Unlike the aforementioned existing audio adversarial attacks, we aim to compromise the users' voice directly without needing any additional noticeable audio having to be played in order to carry out the attack. Furthermore, given the small magnitude of the proposed attacks, the DoS audio adversarial attacks are almost completely inaudible which therefore makes it difficult to identify their source, compared to the previous work that generates audible sounds.

We propose white-box DoS attacks for VCS in practical settings aimed at unconstrained audio from users. We focus and overcome the following two important challenges that are crucial to realize the DoS attack on audio.

Input independence: An attacker has no knowledge in advance about *what* sound will be uttered by the legitimate user. Previous work generates an adversarial perturbation based on a specific input from a static dataset to generate an adversarial example [1], [2], [6]–[9], [12], [23], which is not possible in a practical DoS attack scenario. For an effective DoS to fool the classifier in a VCS, the attacker should generate an adversarial perturbation that is applicable to any possible inputs from the user.

Time independence: An attacker has no a priori knowledge about *when* the audio command will be spoken by the user. As hinted in the previous section, synchronization between the adversarial perturbation and the user input directly affects the fooling rate of the attack. Since most existing work trains adversarial examples for a specific input, they assume that the adversarial perturbation is perfectly synchronized with the particular input. For a reliable DoS, adversarial perturbations should work without needing to be perfectly synchronized with the specific input.

Fig. 2 demonstrates our threat model. There is a user (victim) who speaks with her voice, illustrated as "house", "seven", and "right". There is a device (classifier) which recognizes user's spoken language. The attacker, a device with

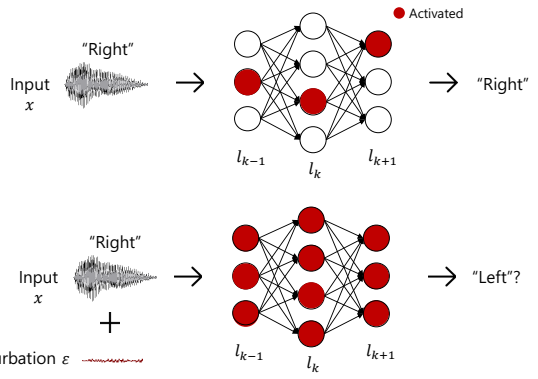

Fig. 3. Illustration of the fooling algorithm, where perturbation is crafted to alter the response of each layer from informative features to uninformative similarly activated features.

a speaker (e.g., smartphone), continuously generates input- and time-independent adversarial perturbations, which are detailed in the next section. The played adversarial perturbations from the attacker are then combined with the user's spoken words naturally over-the-air. Note that the attacker has no prior knowledge about what and when the user speaks and there is no guarantee that the adversarial perturbations and the user's spoken audio will arrive ideally synchronized at the device. The combined audio signals nevertheless become adversarial examples, which fool the device and incorrectly classify the inputs as "visual", "one", and "left".

IV. Methodology

In this section we describe methods for generating input- and time-independent perturbations for effective audio DoS attacks. Our key starting point is that there might exist a *universal* adversarial perturbation for audio inputs that compromises most of users' speech commands. We build on the concept of universal adversarial attacks designed for images [15]. However, unlike images, audios are sensitive to the signal propagation distortion and synchronization between the benign audio and the adversarial perturbation. We present training algorithms for adversarial perturbations to be tolerant to interaction timing. Our proposed solution overcomes the input- and time-independence challenges that are unique for practical audio DoS adversarial attacks. Together with the development of DoS attack, we also present how we can define and measure the imperceptibility of the attacks.

A. Attack Algorithms

Two main requirements of AudiDoS are: (i) universal audio adversarial perturbation generation, and (ii) overcoming time-synchronization. Our main objective is to find a perturbation that works for most inputs and their possible shifts, i.e., there exists an input independent perturbation $\boldsymbol{\epsilon}$ such that $\hat{y}(\boldsymbol{x}+\boldsymbol{\epsilon};\boldsymbol{\theta}^*) \neq \hat{y}(\boldsymbol{x};\boldsymbol{\theta}^*)$ for most inputs $\boldsymbol{x}$. To create such a universal perturbation, we leverage the Fast Feature Fool (FFF) algorithm [15] from the vision domain, which we modify and augment to solve the aforementioned problems unique to the audio domain via the proposed AudiDoS system. The FFF algorithm generates a universal adversarial perturbation by *falsely firing* activations in the neural network. The intuition of the algorithm lies in compromising one of the basic principles of neural networks namely that the activations of each neuron act as informative features for next layer. For example, Fig. 3 illustrates that when a benign audio input "right" is presented, a specific subset of neurons are activated, which provides useful information for the decision of the classifier. However, if we artificially activate all the other neurons, then the classifier performs poorly as it looses important feature extraction property. The FFF algorithm generates an adversarial perturbation $\boldsymbol{\epsilon}$ that is designed for this purpose in the vision domain; A universal adversarial perturbation $\boldsymbol{\epsilon}$ is obtained by minimizing the following loss function given $\boldsymbol{x}$ as a normal input:

$$\mathcal{L}(\boldsymbol{\epsilon}) = -\log\left(\prod_{k=1}^{K} \overline{l_k(\boldsymbol{x}+\boldsymbol{\epsilon})}\right) \text{ s.t. } \|\boldsymbol{\epsilon}\|_\infty \leq \mathcal{E}, \quad (2)$$

where K is the number of layers to be falsely activated, and $\overline{l_k(\boldsymbol{x}+\boldsymbol{\epsilon})}$ is the mean absolute value of the k-th layer output.

However, the original FFF-algorithm produces perturbations that requires perfect synchronization with the benign audio, which limits its practicality in real-world situations. To overcome this problem, we propose two novel training methodological extensions for the audio domain. First, we randomly rotate the perturbation while training, which is a modification aimed at preventing the learned perturbation from being overly synced with the training audio inputs. Second, we halve the amplitude of the perturbation periodically throughout training, which is an enhancement aimed to promote a thorough search of the best perturbation in the neighbourhood defined by $\|\epsilon\|_\infty \leq \mathcal{E}$, rather than allowing the algorithm to quickly navigate the shortest path from the center to the boundary of the $\|\epsilon\|_\infty$-ball without sufficient exploration.

B. Perceptibility Measure

As adversarial attacks are based on the premise that they should be imperceptible to human ears, it is important to select an appropriate value of the perturbation constraint $\mathcal{E}$. For example, a higher value of $\mathcal{E}$ would be effective to fool the classifier with a noticeability of the attack, while a lower value would end up being a useless attack. In order to quantify the trade-off between the effectiveness and the perceptibility, we use the decibel (dB) to quantify the relative loudness of adversarial perturbations to normal audio inputs [7]. The decibel of an audio sample a is represented as:

$$dB(a) = \max_i 20 \cdot \log_{10}(a_i). \quad (3)$$

To measure the relative loudness of the perturbation $\boldsymbol{\epsilon}$ to the normal $\boldsymbol{x}$, we subtract the dB of $\boldsymbol{x}$ from the dB of $\boldsymbol{\epsilon}$, i.e.,

$$dB_{\boldsymbol{x}}(\boldsymbol{\epsilon}) = dB(\boldsymbol{\epsilon}) - dB(\boldsymbol{x}). \quad (4)$$

Since human ears interpret sound in dB scale, a low $dB_{\boldsymbol{x}}(\boldsymbol{\epsilon})$ can be interpreted as a low perceptibility audio perturbation.

TABLE I
SYNCED ERROR RATES (ER) AND PERCEPTIBILITY (DB) OF AUDIDOS ACCORDING TO DIFFERENT CONSTRAINTS $\mathcal{E}$ IN THE SPEECH COMMANDS DATASET.

$\mathcal{E}$	**Synced ER**	**dB**
0.5	0.968	1.02
0.2	0.970	-6.94
0.1	0.968	-12.96
0.05	0.951	-18.98
0.02	0.893	-26.94
0.01	0.777	-32.96
0.005	0.514	-38.98
0.001	0.182	-52.96
Benign	0.034	—

V. EVALUATION PART I: OFFLINE ANALYSIS

We begin by an offline analysis, where we measure the effectiveness of AudiDoS when performing attacks on neural networks that are trained for the audio keyword detection task. Next, we also present performance result of AudiDoS on Automatic Speech Recognition (ASR) tasks.

A. Keyword Spotting Experiment

In the following we describe the details of our offline experiments. For training the neural networks and computing the perturbations, we used the PyTorch [18] framework.

1) Dataset: We use Speech Commands [24] dataset as the default dataset unless otherwise mentioned. The dataset contains 105,829 1-second long utterance of 35 common English words from a large number of users, recorded with 16 KHz sampling rate. The vocabulary in this dataset spans digits, and words useful in IoT applications, e.g., *on, off, start, stop*. The dataset is balanced in terms of the number of samples for each class and includes example of natural background noise.

2) Model: We adopted 5-layered convolutional neural networks (CNN) used in SoundNet [5] and modified for our task. The model is composed of five convolutional layers with three max-pooling layers. Each convolutional layer is followed by a batch normalization and a ReLU activation layers. After five convolutions layers, it has two fully-connected layers for classification. We trained this model with the aforementioned Speech Commands dataset. This model gets 1-second of audio input and outputs the classified word among 35 words described above. The trained model shows 0.034 ER on the test set, i.e., the percentage of incorrect prediction, without any attacks.

3) Perturbation: The adversarial perturbation generated by AudiDoS is a vector with 16K samples, i.e., the same size as the input for the neural networks. We used 10K randomly selected training examples from the Speech Commands dataset for generating the perturbation. While training, we falsely activated each activation and max pooling layer of the model.

B. Keyword Spotting Results

In this subsection we report the error rate (ER) of the model, when inputs are combined with the adversarial perturbations generated with varying $\mathcal{E}$. To quantify the effect of synchronization and lack of synchronization between the attacker's sound and the victim's utterance, we report both synced and unsynced ERs. Synced ER refers to the setting when the perturbations are (respectively, are not for unsynched) in perfect synchronization with victim's utterance.

TABLE II
SYNCED AND UNSYNCED ERROR RATES (ER) OF RANDOM AND AUDIDOS IN THE SPEECH COMMANDS DATASET.

Method	$\mathcal{E}$	**Synced ER**	**Unsynced ER**	**dB**
Random	0.02	0.307	—	-26.94
Random	0.01	0.242	—	-32.96
AudiDoS	0.02	**0.873**	**0.780**	-26.94
AudiDoS	0.01	**0.753**	**0.632**	-32.96
Benign	—	0.034	—	—

1) Impact of Constraint $\mathcal{E}$: When training an adversarial perturbation, the limit of the magnitude of the perturbation (i.e., the constraint $\mathcal{E}$) directly affects both the success and the perceptibility of the attack. To investigate the precise impact of $\mathcal{E}$ in the audio domain, we evaluate the ER of AudiDoS with different values of $\mathcal{E}$. Table I summarizes the ER for different constraints $\mathcal{E}$, as well as the reference benign case corresponding to the original performance of the model without any attack. We calculated the ERs with 12k test examples from the Speech Commands dataset. As shown in Table I, the ERs drop gradually as the constraint $\mathcal{E}$ decreases from 0.5 to 0.001. In particular, constraints higher or equal to 0.05 achieve more than 0.95 ERs, while the lowest constraint 0.001 yields lower than 0.2 ER. In addition, Table I also reports on the perceptibility of the attacks using Equation 4, with which we calculated the dB levels of the attacks relative to the entire training dataset from Speech Commands and reported the averaged dB. In this regard, except for $\mathcal{E} = 0.5$, all measured dBs are negative, which means the attacks are smaller than the normal unperturbed audio keywords. As a point of reference, -30 dB is similar to the dB difference between normal speech and ambient noise in a quite room [7]. Thus Table I informs us that to achieve a practical attack in terms of ER and perceptibility, we should focus on values of $\mathcal{E}$ of 0.01 or 0.02.

2) Unsynced Attack: Although the previous results yield important insights into the performance of AudiDoS, in a real-world scenario an attacker continuously plays an adversarial perturbation through the air and has no control as to when the user will interact with his/her device, which means that synchronization between the target input and the adversarial perturbation cannot be guaranteed. To quantify the effect of this phenomenon on the effectiveness of real-world attacks, Table II compares the ER of attacks with random (baseline) and AudiDoS, not only for synced but also for unsynced attacks. We tested each method with 12K test examples from the Speech Command dataset. To simulate the effect of an unsynced perturbation on an audio input, we right-rotated the perturbations to generate ten different perturbation versions. For example, $\boldsymbol{\epsilon}_0$ is generated by right-rotating $\boldsymbol{\epsilon}$ by 100 ms (i.e., one tenth of the input duration). Next, $\boldsymbol{\epsilon}_1$ is generated by rotating $\boldsymbol{\epsilon}_0$ by 100 ms and so on. Finally, we evaluated the

TABLE III
THE CER AND WER OF RANDOM AND AUDIDOS USING LIBRISPEECH DATASET AND DEEPSPEECH2 MODEL.

Method	Norm	CER	CER*	WER	WER*	dB
Random	0.02	0.583	—	0.932	—	-28.00
Random	0.01	0.490	—	0.859	—	-34.02
AudiDoS	0.02	**0.805**	**0.756**	**0.999**	**0.991**	-28.00
AudiDoS	0.01	**0.762**	**0.703**	**0.994**	**0.978**	-34.00
Benign	—	0.079	—	0.236	—	—

**unsynced result.*

TABLE IV
THE CER AND WER OF RANDOM AND AUDIDOS USING TED-LIUM DATASET AND DEEPSPEECH2 MODEL.

Method	Norm	CER	CER*	WER	WER*	dB
Random	0.02	0.577	—	0.929	—	-30.61
Random	0.01	0.607	—	0.931	—	-36.64
AudiDoS	0.02	**0.701**	**0.717**	**0.985**	**0.983**	-30.61
AudiDoS	0.01	**0.752**	**0.662**	**0.992**	**0.971**	-36.64
Benign	—	0.115	—	0.378	—	—

**unsynced result.*

ER of $\epsilon_0, \epsilon_1, \ldots, \epsilon_9$ and averaged the results into the unsynced ER column in Table II.

From Table II we observe that the results of AudiDoS vary between synced and unsynced attacks (random has no impact). In particular, the fact that unsynced ERs for AudiDoS are lower than synced ERs shows that the synchronization or lack thereof should be considered when designing attacks for audio applications. Although there is some degradation from synced to unsynced results, AudiDoS shows higher ERs than random ones. Furthermore, it can be observed from Table II that as the norm of the perturbation $\mathcal{E}$ increases, the ER increases in all the cases; in particular, while random gives the lowest ER of 0.2-0.3, AudiDoS show ERs as high as 0.8.

C. Automatic Speech Recognition (ASR) Experiment

Moving away from the keyword spotting task, in this section we consider a more general task of ASR to further validating effectiveness of AudiDoS. ASR is more challenging task and contrary to prevision scenario, inputs to the neural networks can be of variable length and the output can potentially be any arbitrary text supporting an open vocabulary.

1) Datasets: For ASR experiments, we consider LibriSpeech [16] and the TED-LIUM [19] datasets. We use these datasets containing unconstrained speech to understand the effectiveness of our attacks in more general settings. In detail, LibriSpeech is a public automatic speech recognition corpus, which contains 1,000 hours of speech data derived from 14,500 English audio books sampled at 16 KHz. TED-LIUM is an English-language TED talks corpus with transcriptions, which is extracted from 1,495 TED talks which are 207 hours in total under 16 KHz sampling rate. The corpus contains over 2.6 M words.

2) Models: The model we used in this experiment is the state-of-the-art speech-to-text model DeepSpeech2 [3]. DeepSpeech2 uses connectionist temporal classification (CTC) loss, which enables to process unsegmented sequence data directly [11]. We trained two DeepSpeech2 models with LibriSpeech and TED-LIUM dataset respectively. The structure of the models are the same. Specifically, the models start with two convolutional layers, followed by five gated recurrent unit (GRU) layers and ends up with one fully-connected layer. The model uses batch normalization for each layer.

3) Perturbations: DeepSpeech2 model gets variable length of audio and breaks it down into 20 ms-sized chunks. Accordingly, the trained adversarial perturbation for this model is 320 dimensional vector. While training, we falsely activated the two activation layers after the first two convolutional layers of the model.

Fig. 4. Real-world experiment setup.

4) Results: Table III shows the results for the DeepSpeech2 model trained with the LibriSpeech dataset and Table IV shows the results for the DeepSpeech2 model trained with the TED-LIUM dataset. The tables show that AudiDoS is generally better than random perturbations for both datasets. Specifically in Table III, AudiDoS achieves more than 0.2 CER degradation compared to random. We found that if the benign performance of model is not good (e.g., benign WER is 0.378 in Table IV), we can achieve high error rate with small perturbation or even with random noise. The performance gain using other methods becomes noticeable when the benign performance is good (e.g., benign CER is 0.079 in Table III). We conjecture that if the audio model performance itself is not enough, then it would be more susceptible to adversarial attack or even random noise, and we believe this is something to consider when building a speech-to-text model that works in the wild.

VI. EVALUATION PART II: REAL-WORLD EXPERIMENT

In this section we complement our study by evaluating our attacks in a real-world environment. In particular, we investigate the performance of AudiDoS in the wild and uncover the challenges associated with real-world audio adversarial attacks.

A. Experimental Setting

Fig. 4 shows the setup for the real-world experiment. For this experiment, we used SoundNet model and Speech Commands dataset. To account for different speech utterances and noise levels, we automated the user's interaction with the device running the classifier by playing speech commands through i) a Nexus 5X smartphone and ii) an Anker SoundCore2 portable speaker. The sound emitted by these devices thus represents the user's speech which is being broadcasted over the air to be picked up by the microphone of the device running the classifier. These two devices, i.e., the users are about 30 cm away from the device running the classifier. The classifier is running on a LG Gram laptop, connected to an external microphone. Finally, we placed a Samsung Galaxy A8 next to the classifier as the attack device. Throughout the

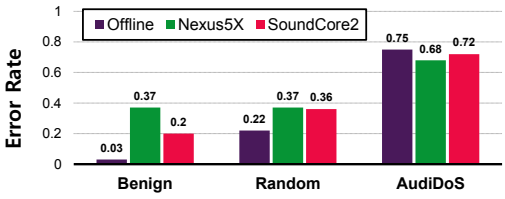

Fig. 5. Error rate of the classifier with attacks ($\mathcal{E} = 0.01$).

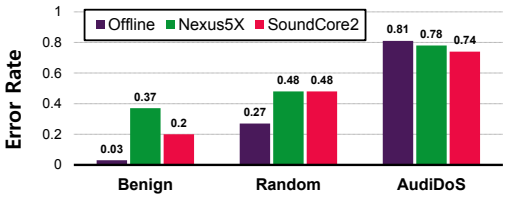

Fig. 6. Error rate of the classifier with attacks ($\mathcal{E} = 0.02$).

experiment, the devices were placed on top of a desk in an office room. We played 100 audios from Speech Command test set at the speakers representing the user's speech. Meanwhile, we played 1 second of the attack sound repeatedly through the attacker's device. Accordingly the test audios and the attack sound were not synchronized when arriving at the classifier device. We measured ERs of the classifier by comparing classified outputs and ground-truth labels of each sample across various attack methods.

B. Results

Fig. 5 shows the ERs when the constraint ($\mathcal{E}$) is equal to 0.01. On the x-axis *benign* refers to a non-attack scenario, *random* refers to using random noise as perturbations and AudiDoS refers to the attacks generated by our proposed method. We use offline as a baseline, where the evaluation is done on a single machine without playing the user's speech and the attack sound over the air and thus the audio inputs that 'reach' the classifier are artificially synchronized. We first observe that even without any attack, the ER of the classifier in-the-wild increases from 0.03 to 0.37 (Nexus 5X) and from 0.03 to 0.2 (SoundCore2), which is mainly caused by the poor speaker and microphone quality, distortion, multi-path and noise. With a higher constraint of 0.02, the ERs drop consistently as shown in Fig. 6. It is worthwhile to observe that the ERs with attacks (Random and AudiDoS) in Fig. 5–6 are in line with the result in Table II.

In all cases, AudiDoS is consistently shown to be a more effective attack compared to Random noise. However, AudiDoS underperforms compared to the Offline evaluation. We believe this is caused by distortions in the audio in-the-wild, the same reason which caused the classifier error to increase between Offline and in-the-wild settings in the Benign scenario. A recent study [26] overcomes this problem by injecting random noise while training the perturbation; a technique that is orthogonal to AudiDoS, and can therefore be incorporated alongside AudiDoS. In summary, AudiDoS's performance remains valid in the wild and is more effective than random attacks, with ERs as high as 78%.

VII. Related Work

In this section, we review prior work on attacking voice-controllable systems (VCS) from the domain of audio adversarial attacks, and universal adversarial perturbation.

A. Audio Adversarial Attacks

Recent studies show the existence of adversarial attack on audio models such as speech recognition [8], [9] and speaker verification model [12]. Carlini et al. [7] applied adversarial perturbation for targeted attack on speech-to-text model to trigger commands of attacker's interest. Other studies [2], [23] demonstrate the possibility of adversarial examples for black-box audio systems. However, their evaluation is limited to offline analysis where the adversarial examples did not play through the air and thus had no issues such as distortion, noise, and synchronization between the adversarial perturbations and the target audio.

Several studies demonstrate the possibility of the attacks in the wild. The hidden voice commands [1], [6] work generates obfuscated voice commands (i.e., noises that humans cannot interpret but VCS do) for launching commands of the attacker's interest. However, the obfuscated commands used in this work are directly played over the air and thus makes it noticeable to users. CommanderSong [26] embeds adversarial perturbations on common songs to trigger attacker-intended commands. Yakura et al. [25] design malicious voice commands generated by adversarial examples and evaluated them over the air. While these studies play malicious commands directly over the air, AudiDoS aims to compromise benign inputs from users (DoS attack), thus making it more stealthy in nature. That said, the fundamental techniques proposed in the previous works to account for audio distortion when the adversarial samples are played over the air could be combined with AudiDoS to make a more robust attacking system.

B. Inaudible Voice Commands

A series of works have shown that it is possible to give almost inaudible malicious commands to these systems by generating nefarious ultra sounds embedded with legitimate voice commands that humans cannot very easily recognize but microphones can [20], [21], [27]. These works leverage the non-linearity of microphones so that the generated high-frequency (over around 30kHz) ultra sounds could be captured as lower frequency (below around 18kHz) sounds at the microphone. The limitation of these works is that they require specialized speakers that are able to generate ultra sound. Modern devices such as iPhone 6 Plus [26] have already patched the non-linearity problem with their microphones, making the aforementioned attacks invalid. Our work is independent of the hardware peculiarities and instead focuses on generating adversarial perturbations in the software to target the inherent vulnerabilities of deep audio models, and is thus applicable to recent smartphones even if they mitigate the non-linearity effect such as iPhone 6S [26].

VIII. Discussion

In this section we discuss future directions for this line of research and limitations of this work.

A. Defense

Our paper is limited to designing attacks on audio deep models albeit in a universal way. Defending against these attacks remains an open problem that we leave this task for future work. However, we now discuss a few potential attack-prevention techniques that can be employed. Firstly, the classifier model can monitor the prediction probabilities for the current input and if there is a spurious high probability despite the small decibel of the input, then it could be flagged as a potential attack. Another approach could be to detect the presence of a continuous signal with a small amplitude, since the current version of out attack system plays an adversarial perturbation continuously.

B. Perceptibility

An attack is meaningful when it is stealthy enough to be imperceptible to users. Although we quantify the loudness of the perturbations on the dB scale, we acknowledge that it is not sufficient to measure the true perceptibility by humans. During our attack experiments, we observed that the adversarial perturbations result in an audible noise, which albeit very low in amplitude and completely unintelligible, could still be sensed by humans if they are very close to the attack device and are trained to know what to hear for. However, as the distance between a human and the attack device increases, the ability to sense the attack goes down. As a future research, we plan to study whether the adversarial perturbations could be generated in the inaudible frequency range, i.e., over 20 KHz.

IX. Conclusion

In this paper, we propose AudiDoS, a system for denial of service attack targeted to VCS employing deep neural networks. The proposed system works at real-time and while humans are interacting with VCS. To the best of our knowledge, this is the first study to adopt the universal adversarial perturbation concept for the DoS attack on audio deep models. Our evaluation shows that with a small distortion it is possible to increase the error rate of a classifier significantly high. Our real-world experiments demonstrate that such an attack is indeed feasible in the wild with error rates as high as 78%. That said, there remain a number of open challenges pertaining to real-world adversarial attacks as well as for designing defense mechanisms against them. Apart from widening the space on real-time DoS attacks on deep audio models over the air, we believe our findings further illustrate the possible threats of adversarial attacks on audio deep models and call for future research to thwart such attacks.

References

[1] H. Abdullah, W. Garcia, C. Peeters, P. Traynor, K. R. Butler, and J. Wilson, "Practical hidden voice attacks against speech and speaker recognition systems," *NDSS*, 2019.

[2] M. Alzantot, B. Balaji, and M. Srivastava, "Did you hear that? adversarial examples against automatic speech recognition," *arXiv preprint arXiv:1801.00554*, 2018.

[3] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," in *ICML*, 2016, pp. 173–182.

[4] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *ICML*, 2018.

[5] Y. Aytar, C. Vondrick, and A. Torralba, "Soundnet: Learning sound representations from unlabeled video," in *NIPS*, 2016, pp. 892–900.

[6] N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. Wagner, and W. Zhou, "Hidden voice commands." in *USENIX Security Symposium*, 2016, pp. 513–530.

[7] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2018, pp. 1–7.

[8] M. M. Cisse, Y. Adi, N. Neverova, and J. Keshet, "Houdini: Fooling deep structured visual and speech recognition models with adversarial examples," in *NIPS*, 2017, pp. 6977–6987.

[9] N. Das, M. Shanbhogue, S.-T. Chen, L. Chen, M. E. Kounavis, and D. H. Chau, "Adagio: Interactive experimentation with adversarial attack and defense for audio," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2018, pp. 677–681.

[10] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," *arXiv e-prints*, p. arXiv:1412.6572, Dec. 2014.

[11] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *ICML*. ACM, 2006, pp. 369–376.

[12] F. Kreuk, Y. Adi, M. Cisse, and J. Keshet, "Fooling end-to-end speaker verification with adversarial examples," in *ICASSP*. IEEE, 2018, pp. 1962–1966.

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

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

[15] K. R. Mopuri, U. Garg, and R. V. Babu, "Fast feature fool: A data independent approach to universal adversarial perturbations," *arXiv preprint arXiv:1707.05572*, 2017.

[16] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *ICASSP*. IEEE, 2015, pp. 5206–5210.

[17] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples," *arXiv preprint arXiv:1605.07277*, 2016.

[18] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS-W*, 2017.

[19] A. Rousseau, P. Deléglise, and Y. Esteve, "Enhancing the ted-lium corpus with selected data for language modeling and more ted talks." in *LREC*, 2014, pp. 3935–3939.

[20] N. Roy, H. Hassanieh, and R. Roy Choudhury, "Backdoor: Making microphones hear inaudible sounds," in *MobiSys*. ACM, 2017, pp. 2–14.

[21] N. Roy, S. Shen, H. Hassanieh, and R. R. Choudhury, "Inaudible voice commands: The long-range attack and defense," in *NSDI*. USENIX Association, 2018, pp. 547–560.

[22] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

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

[24] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *arXiv preprint arXiv:1804.03209*, 2018.

[25] H. Yakura and J. Sakuma, "Robust audio adversarial example for a physical attack," *arXiv preprint arXiv:1810.11793*, 2018.

[26] X. Yuan, Y. Chen, Y. Zhao, Y. Long, X. Liu, K. Chen, S. Zhang, H. Huang, X. Wang, and C. A. Gunter, "Commandersong: A systematic approach for practical adversarial voice recognition," in *USENIX Security*, 2018, pp. 49–64.

[27] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu, "Dolphinattack: Inaudible voice commands," in *CCS*. ACM, 2017, pp. 103–117.