Adversarial Audio Attacks that Evade Temporal Dependency

Heng Liu and Gregory Ditzler
Department of Electrical & Computer Engineering
The University of Arizona
Tucson, AZ 85721
{hengl, ditzler}@email.arizona.edu

***Abstract*—As the real-world applications (image segmentation, speech recognition, machine translation, etc.) are increasingly adopting Deep Neural Networks (DNNs), DNN's vulnerabilities in a malicious environment have become an increasingly important research topic in adversarial machine learning. Adversarial machine learning (AML) focuses on exploring vulnerabilities and defensive techniques for machine learning models. Recent work has shown that most adversarial audio generation methods fail to consider audios' temporal dependency (TD) (i.e., adversarial audios exhibit weaker TD than benign audios). As a result, the adversarial audios are easily detectable by examining their TD. Therefore, one area of interest in the audio AML community is to develop a novel attack that evades a TD-based detection model. In this contribution, we revisit the LSTM model for audio transcription and propose a new audio attack algorithm that evades the TD-based detection by explicitly controlling the TD in generated adversarial audios. The experimental results show that the detectability of our adversarial audio is significantly reduced compared to the state-of-the-art audio attack algorithms. Furthermore, experiments also show that our adversarial audios remain nearly indistinguishable from benign audios with only negligible perturbation magnitude.**

I. Introduction

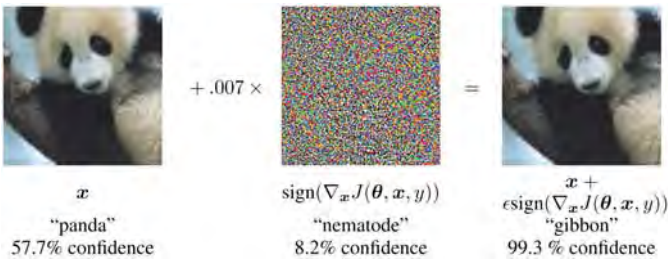
Deep Neural Networks (DNNs) have achieved remarkable success in numerous real-world applications (e.g., image/video analysis [1], audio analysis [2], natural language processing [3], etc.). However, recent contributions have shown that DNNs can be easily fooled by adversarial inputs that appear to be legitimate for the oracle's perspective [4]. For example, Figure 1 shows Goodfellow et al.'s classic example of an image of a panda that has been maliciously perturbed with a signal that is not observable by the oracle (i.e., human) eye. In the past decade, many studies have explored the impact of an adversary on various applications (i.e., image analysis [5]–[7], text classification [8], and malware detection [9]–[11]); however, there are fewer works that discuss DNNs' behaviors against adversaries that are built for audio analysis despite the large number of real-world applications that rely on accurate audio transcription technologies (e.g., Google Home, Amazon's Alexa).

An attack algorithm used against a benign audio X finds an adversarial audio X^a that leads to a wrong output Y^a by maximizing the loss. This procedure of manipulating the audo is similar to attack algorithms against images with the difference being the medium of the attack. Generally, the audio attack objective is given by:

$$X^a = \arg \max_{(X,Y)\in\mathcal{D}\cup\{(X^a,Y^a)\}} \mathbb{L}(f_\theta(X), Y) \qquad (1)$$

where $\mathcal{D}$ is the benign audio dataset, $\mathbb{L}$ is a cost function, X is the benign audio, Y is the ground truth for the audio X, and f_θ is a neural network with parameters θ.

Attack algorithms against audios are generally categorized into two groups that correspond to different audio tasks: speech-to-text and speech-to-label attacks. A speech-to-text task takes an input audio X and generates a sequence of texts Y. For example, let X be an audio signal that is represented as a sequence of length m, where $X = \{X[1], \ldots, X[t] \ldots, X[m]\}$ and Y be the correct transcription of X, which could be Y = "*What time is my doctor's appointment*?". The adversary seeks to generate audio X^a such that $\|X - X^a\|_2^2$ is arbitrarily small and Y^a = is different than the ground truth transcription of Y. Recently, a speech-to-text attack algorithm against the DeepSpeech model [12] uses a gradient-based method to arbitrarily modify audios' machine transcriptions Y to Y^a (i.e., $Y^a \neq Y$). Moreover, the attack algorithm can inject imperceptible perturbations (i.e., $\min ||X^a - X||_2^2$) directly to the raw audio waveform X. Figure 2 shows an example of the audio attack in [12]. The perturbation σ is found by maximizing the audio attack objective (1) using backpropagation. On the other hand, the speech-to-label application takes an input audio X and yields



(a) Adversarial example for image classification

Fig. 1. Goodfellow et al.'s demonstration of fast adversarial example generation applied to GoogLeNet on ImageNet challenge [4]. By adding an imperceptibly small perturbation to the image, the classification result of GoogLeNet for "panda" has changed to "gibbon" with high confidence.

978-1-7281-2547-3/20/$31.00 ©2020 IEEE

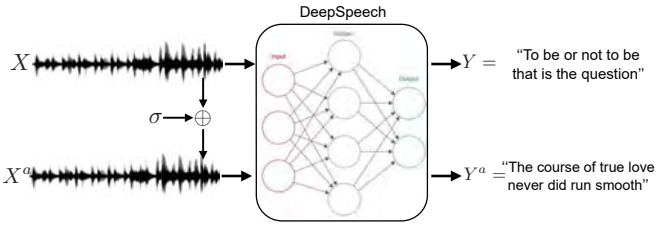


Fig. 2. Overview of Carlini's attack algorithm [12]. Benign audio X produces the correct transcription Y via DeepSpeech. The adversary uses an attack to obtain a perturbed adversarial audio X^a which produces $Y^a \neq Y$.

a class label C. In speech-to-label attack, an adversarial audio X^a subverts the authentic class label C while remains close to X. For example, let X be an audio signal that is represented as a sequence of length m and Y be the correct category of the audio. For the audio signal $X =$ "*You have a doctor's appointment in 30 minutes*," the correct assignment could be $Y =$ "*Calendar Notification*," where class labels could be in the set $\mathcal{Y} := \{$*New Text Message*, *Calendar Notification*$\}$. Alzantot et al. proposed a generic audio attack algorithm against a speech command classification model. Their attack added an imperceptible random noise to the original audio signal [13] (i.e., simply changing the least significant bits in the audio signal). Alzantot et al.'s attack achieved an 87% attack success rate by adding small background noise without having to know the underlying model parameter and architecture. In both speech-to-text and speech-to-label attacks, the resulting adversarial audio $X^a = X + \sigma$ is almost identical to X in both time and frequency domains.

Defensive countermeasures for audio analysis applications are also under-explored. The research community first proposed to use the defensive feature transformation techniques that are proven to be quite effective against image attacks to defend against audio attacks. These feature transformation techniques include waveform quantization, local smoothing, downsampling, auto-encoder transformation, etc [14]. Unfortunately, these techniques provide limited security to audio analysis applications in a malicious environment [15]. This limited security is due to the images and audios having fundamentally different structures. That is the images are spatially structured while audios are sequentially structured. Recently, Yang et al. proposed a novel adversarial audio detection algorithm based on the empirical result that adversarial audios behave differently from benign audios in terms of temporal dependency [15]. Their experiments showed that the their detection algorithm can easily identify a variety of state-of-the-art audio attacks. Temporal Dependency (TD) is a new concept emerged in the AML. As a result, the TD's impact on audio attacks and audio defenses remains unknown. An audio's temporal dependency is an innate and intrinsic characteristic that depicts the relations among different temporal steps in an audio sequence. Generally, DNN-based audio analysis applications model the temporal dependency explicitly through various intermediate results during neural network training, e.g., hidden states in LSTM, attention in transformer models.

In this contribution, we focus on generating adversarial audios that are against speech-to-text transcription tasks. We first revisit the LSTM model that is commonly used for performing the transcription then we shed light on the TD's role in generating an adversarial audios' against a speech recognition model. Then we propose a new audio attack algorithm that evades the TD-based adversarial audio detection and benchmark our algorithm, as well as the state-of-the-art, on the Mozilla dataset. Our results show that our adversarial speech model can evade the TD detection methods.

The rest of this paper is organized as follows: In section II, we review the TD-based detection method and analyze the TD's impact on audio attacks in section III. We present the experimental evaluations in section IV and draw the conclusions in section V.

II. Related Works

In this section, we review the latest contributions in adversarial machine learning for audio analysis applications then we review the technical details of TD-based detection methods.

A. Adversarial Audio Examples

The adversarial audio example generation against DeepSpeech (i.e., a model developed for speech recognition and is the state-of-the-art) proposed by Carlini and Wagner is the first targeted speech-to-text audio attack algorithm (i.e., explicitly specify the attack target Y^a) [12]. This audio attack method is particularly effective on various benchmarks, given that the slight noise is imperceptible to a human ear. In [16], Yukura and Sakuma take into account the impacts when audios are played over-the-air (e.g., background white noise, frequency filtering during analog to digital converting, etc.) and designed a robust audio attack method against DeepSpeech. Although the adversarial audios in [12] achieved an almost 100% success rate, Carlini and Wagner assumed a white-box setting which requires detailed information of the victim's model (e.g., DNN structure, trained weights, etc). In [17], Taori et al. proposed a black-box audio attack method by combining the genetic algorithms and gradient estimation.

As for audio defenses, feature transformation techniques (e.g., image quantization, filtering, image reprocessing, autoencoder reformation) are widely adopted as countermeasures against an adversary in real-world tasks [18]–[21]. These feature transformation techniques are widely used in many applications due to their low cost and the fact that they can be used with various DNN architectures. Feature transformation defenses aim to filter the adversarial perturbations of the raw input. While feature transformations are effective on images, they provide limited security against adversarial audio attacks [15]. In [22], Rajaratnam and Kalita proposed to flood particular frequency bands with random noise to detect adversarial audios. Unfortunately, an adversary can specify the frequency bands that carry the adversarial perturbations to evade the noise flooding detection method [16]. Recently, Yang et al. proposed an empirical test to discriminate against

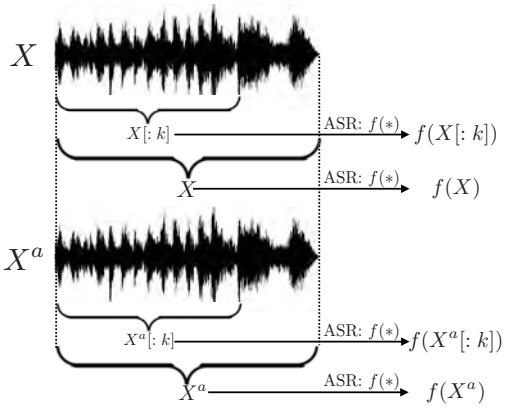


Fig. 3. This figure shows the TD-based detection method. X, X^a are benign and adversarial audios respectively. $X[:k], X^a[:k]$ denote the first k percent partial audios of X, X^a respectively. We use $f(*)$ to represent the Automatic Speech Recognition (ASR) system.

various adversarial audios by measuring the audios' temporal dependency [15]; however, we will show in this paper that this detection technique can be easily fooled by suppressing the temporal dependency in adversarial audios.

B. Adversarial Audio Detection with Temporal Dependency

In this section, we review the details of the detecting adversarial audios with TD methods. Figure 3 shows the general procedures for a TD-based detection method. We first explain the notations in Figure 3. We refer to the benign and adversarial audios as X and X^a, respectively. The Automatic Speech Recognition (ASR) system is denoted as $f(*)$. We use $X[:k]$ and $X^a[:k]$ to denote the first k percent partial audio of X and X^a, respectively. For the complete audios, the machine transcriptions from ASR for X and X^a are $f(X)$ and $f(X^a)$, respectively. For the partial audios, the machine transcriptions for $X[:k]$ and $X^a[:k]$ are $f(X[:k])$ and $f(X^a[:k])$, respectively. Specifically, the TD-based detection performs the following steps:

- Slice the first k percent partial audios $X[:k]$ and $X^a[:k]$ then obtain the transcriptions $f(X[:k])$ and $f(X^a[:k])$ from ASR (e.g., DeepSpeech);
- Apply ASR to the complete audios X and X^a to obtain the complete transcriptions $f(X)$ and $f(X^a)$;
- Slice the complete transcriptions $f(X)$ and $f(X^a)$ to have the same length as $f(X[:k])$ and $f(X^a[:k])$, respectively. The complete transcriptions after slicing are denoted as $f(X)[:k]$ and $f(X^a)[:k]$;
- Calculate the transcription consistencies between $f(X)[:k]$ and $f(X[:k])$ as well as between $f(X^a)[:k]$ and $f(X^a[:k])$.

In the last step, the transcription consistencies can be measured by Word Error Rate (WER) or Character Error Rate (CER). WER/CER is defined as the word/character errors (i.e., substitution, insertion, and deletion errors) divided by the total number of word/character in the reference text.

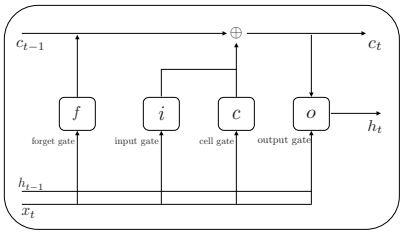


Fig. 4. This figure ties together the different gates in LSTM. Specifically, the input gate, forget gate, cell gate, and output gate correspond to $z = \{i, f, c, o\}$, respectively.

In [15], empirical experiments demonstrated that the transcription consistency between $(f(X)[:k], f(X[:k]))$ is significantly higher than consistency between $(f(X^a)[:k], f(X^a[:k]))$. In terms of WER and CER, the error rates between $(f(X)[:k], f(X[:k]))$ is significantly lower than error rates between $(f(X^a)[:k], f(X^a[:k]))$. The TD-based detection method exploits the above observation to discriminate against adversarial audios. The experiments in [15] concluded that the TD-based adversarial audio detection is adequate to detect a variety of state-of-the-art audio attacks. In this contribution, we show that the TD-based detection method can be fooled by suppressing the temporal dependency when the adversarial audio is being generated.

III. A New Audio Attack Algorithm

The LSTM is widely adopted to model the temporal dependency in audios [23] (see Figure 4 for a representation of an LSTM neuron.). The output of LSTM at each time t takes into account both the corresponding input x_t and the hidden states h_{t-1}. Here we provide the equations of LSTM as follows. For different gates, $z = \{f, i, o, c\}$, the W_z matrices are used to form representations of inputs, and the U_z matrices form representations of hidden states. More formally, these expressions are given by:

$$
\begin{aligned}
f_t &= \zeta(W_f x_t + U_f h_{t-1}) \\
i_t &= \zeta(W_i x_t + U_i h_{t-1}) \\
o_t &= \zeta(W_o x_t + U_o h_{t-1}) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \zeta(W_c x_t + U_c h_{t-1}) \\
h_t &= o_t \circ \zeta(c_t)
\end{aligned}
$$

In [15], the TD-based detection method achieved high discriminating power against the audio attacks that target DeepSpeech. The DeepSpeech is an open-sourced bidirectional recurrent neural network model that has two LSTMs for forward and backward dependencies [24]. Figure 5 shows the structure of DeepSpeech. Both of the LSTMs in DeepSpeech evaluate the hidden states sequentially. Thus removing any part of the audio can result in a loss of essential transitions of hidden states. As a result, in the TD-based detection method, removing part of the audio near the end impacts the backward hidden states' transitions. Similarly, removing part of the audio at the beginning impacts the forward hidden

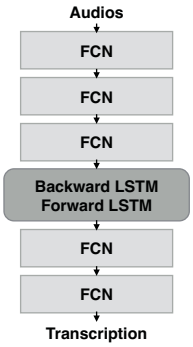


Fig. 5. Overview of the DeepSpeech neural network that starts with three full connected layers followed by backward/forward LSTMs then followed by two more fully connected layers.

states' transitions (note this was not tested in [15]). Moreover, the impact on hidden states causes incoherence in the final transcription. Therefore, when audios are tested by the TD-based detection method, the benign audio exhibits a larger transcription consistency because of its intrinsic TD remains intact or slightly impacted. On the other hand, the adversarial audio is perturbed to provide the desired output while not modifying the TD (i.e., hidden states) accordingly. Thus, this is the reason the adversarial audio exhibits low transcription consistency.

We now motivate our adversarial audio generation method by discussing two scenarios that the TD-based detection method fails. The first failure scenario is when the audio attack algorithm explicitly designs the audio waveform as well as the hidden states. The second failure scenario is when the adversary completely removes the TD from the adversarial audio such that the transcription is independent of hidden states in DeepSpeech. Our audio attack algorithm exploits the second failure scenario of the TD-based detection method. Specifically, we propose our approach based on Carlini and Wagner's audio attack algorithm [12].

In [12], the audio attack algorithm minimizes objective in (2). The first part of the objective is the CTC-Loss [25] which measures X^a's output transcription's distance compared to Y^a. The second part of the objective minimizes X^a's perturbation magnitude comparing with X.

$$\arg\min_{X^a} \mathbb{L}_{\text{CTC}}(X^a, Y^a) + \|X - X^a\|_2^2 \tag{2}$$

As we can observe, (2) only takes into account the attack effect and the perturbation magnitude. We propose to add a third term that modifies the adversarial audios' TD. The new objective is shown in (3). The proposed third term consists of a penalization term $||U_z h_t^a||_2^2$ and a rewarding term $||W_z X_t^a||_2^2$. X_t^a is the component of X^a at time step t and h_t^a is the backward LSTM's hidden state at time t. Recall that W_z/U_z matrices are used to form representations of inputs and hidden states in LSTM, respectively. The reasoning of the third term is that $U_z h_t^a$ and $W_z X_t^a$ correspond to the contributions from previous time's hidden states h_t^a and current time's input X_t^a to the final output, respectively (note we only discuss the backward LSTM but this can be easily extended to the forward LSTM). Therefore, penalizing $U_z h_t^a$ and rewarding $W_z X_t^a$ will force the adversarial audio X^a to depend more on the input instead of the hidden states from different time steps. The new objective function to optimizes becomes:

$$\begin{aligned}&\arg\min_{X^a} \mathbb{L}_{\text{CTC}}(X^a, Y^a) + ||X - X^a||_2^2 \\ &+ \sum_{z\in\{f,i,o,c\}} \sum_t \{||U_z h_t^a||_2^2 - ||W_z X_t^a||_2^2\}\end{aligned} \tag{3}$$

We further add a scaling factor to (3). In the final objective, see (4), Φ_1 corresponds to minimizing the CTC Loss and perturbation magnitude whereas Φ_2 corresponds to suppressing the temporal dependency across different time steps. The scaling factor α is used to control the trade-off between Φ_1 and Φ_2. Because Φ_1 and Φ_2 are both depending on X^a (directly via X_t^a or indirectly via h_t^a). Thus, we use gradient descend as used in [12] to minimize Eq. (4) to solve for X^a

$$\begin{aligned}\Phi_1 &= \mathbb{L}_{\text{CTC}}(X^a, Y^a) + ||X - X^a||_2^2 \\ \Phi_2 &= \sum_{z\in\{f,i,o,c\}} \sum_t \{||U_z h_t^a||_2^2 - ||W_z X_t^a||_2^2\} \\ &\arg\min_{X^a} \alpha \frac{\Phi_1}{\Phi_1 + \Phi_2} + (1-\alpha)\frac{\Phi_2}{\Phi_1 + \Phi_2}\end{aligned} \tag{4}$$

where $\alpha \in [0, 1]$ is a convex combination parameter that provides a trade-off between the Carlini objective and the penalization for TD. Note that this optimization problem in (3) can easily be solved using the automatic gradient estimators in Tensorflow [26].

TABLE I
THE TARGETS FOR AUDIOS WITH DIFFERENT DURATION.

Duration (seconds)	*Adversarial Target*
$[0, 2.5)$	hello google
$[2.5, 4.5)$	this is an adversarial example
$[4.5, 7.8)$	hello google please cancel my medical appointment

IV. EXPERIMENTS

In this section, we demonstrate the efficacy of the proposed audio attack when facing the TD-based detection method by comparing it with Carlini's audio attack [12]. The victim model is the open-sourced DeepSpeech [1]. We use the Mozilla Common Voice dataset to perform the benchmark, and we chose the 100 16KHz-sampled audios released in [12]. The audio durations are between 1.73s to 7.8s, with an average of 4s. Table I gives the speech-to-text attack target sentences (i.e., the desired transcripts), which are consistent with the related works [12], [15].

[1]https://github.com/mozilla/DeepSpeech

In the experiments, we first perform the TD-based detection test (see Section II-B for details) for each benign audio and corresponding adversarial audios generated by Carlini's and our audio attack algorithms. The obtained word(character) error rates are denoted as WER_{benign}, WER_{Carlini} and WER_{Ours} (CER_{benign}, CER_{Carlini} and CER_{Ours}) and are calculated using open-sourced python implementations [2]. Then, the WER's and CER's are averaged over 100 audio samples. We tested with multiple choices $k \in [0.3,\ 0.95]$ in TD-based detection test (note that $k \geq 0.3$ such that the partial audio contains transcription). We tested with $\alpha \in [0.1,\ 0.9]$ in the proposed audio attack.

Next, we demonstrate the proposed audio attack's efficacy from four perspectives:

A) We first show that the proposed audio attack has decreased WER and CER compared with Carlini's audio attack.
B) Secondly, we show the TD-based detection results comparison measured by AUC score.
C) We then show the comparisons of attack successes and perturbation magnitudes.
D) Lastly, we show the influence of hyper-parameter α in the proposed audio attack.

A. Word (Character) Error Rate Comparison

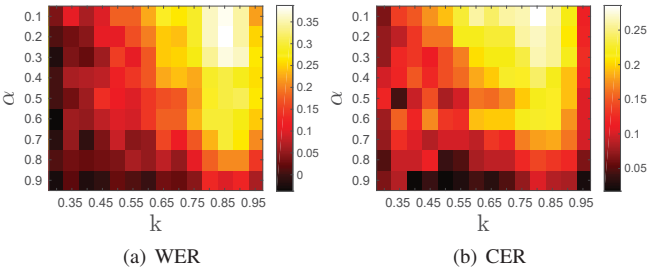
The TD-based detection method's high discriminating power relies on the empirical result that benign audios have lower WER and CER than various adversarial audios [15].

In this section, based on the previously calculated WER_{benign}, WER_{Carlini} and WER_{Ours} (CER_{benign}, CER_{Carlini} and CER_{Ours}), we show the proposed audio attack method exhibits decreased WER and CER by visualizing two set of statistics: (a) $(WER_{\text{Carlini}} - WER_{\text{Ours}})$ and $(CER_{\text{Carlini}} - CER_{\text{Ours}})$; and (b) $(WER_{\text{Ours}} - WER_{\text{benign}})$ and $(CER_{\text{Ours}} - CER_{\text{benign}})$.

The (a) statistics measure how much our adversarial audios' error rates have decreased compared with Carlini's adversarial audios' error rates. This statistics indicate the proposed audio attack method's advantage over Carlini's method in terms of detectability because higher error rates are more likely to trigger the TD-based detector. The (b) statistics measure how much our adversarial audios' error rates exceed the benign audios' error rates. Figure 6 and 7 show the above two set of statistics from our experiments for different choices of α and k.

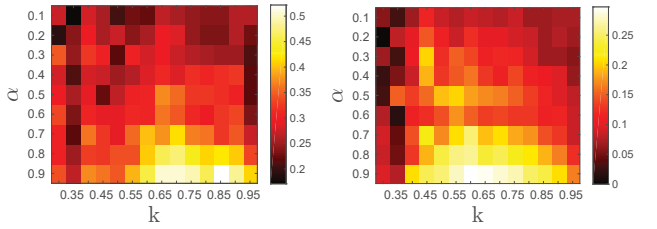
The results for this experiment can be found in Figure 6. The first observation is that the proposed audio attack algorithm has an overall advantage (i.e., more likely to evade the TD-based detector) over Carlini's audio attack in terms of both WER and CER for most choices of α and k. Specifically, our method has a more substantial advantage when there is a small α. The reasoning is that smaller α allows for more concentration on modifying the TD in adversarial audios. The second observation is that for the largest k, Carlini's and our audio attack algorithms perform similarly in terms of CER regardless of α. The reasoning is that the partial audio and the complete audio almost have the same duration. Hence, the transcriptions are both simply the complete adversarial transcription (i.e., desired attack target).

[2]http://pythonhosted.org/asr/index.html
http://pypi.org/project/asrtoolkit/



(a) WER (b) CER

Fig. 6. The above two heatmaps give the (a) statistic: WER(CER) differences of Carlini's adversarial audio minus our adversarial audio. The x-axis is choices of k and y-axis is values of hyper-parameter α. As showed in the colorbar, brighter color indicates our algorithm's advantage and vice versa.



(a) WER (b) CER

Fig. 7. The above two heatmaps give the (b) statistic: WER(CER) differences of our adversarial audio minus the benign audio. The x-axis is different choices of k and y-axis is different values of hyper-parameter α. A darker color means the adversarial audio is more indistinguishable compared with the benign audios and vice versa.

In Figure 7, we observe that our adversarial audio behaves nearly indistinguishably with the benign audio for small α. This observation is more substantial in Figure 7(b) than in Figure 7(a). Moreover, although we noticed that in Figure 7(a) the benign audios' error rates are at least 0.2 lower than our adversarial audios', the reasoning is that the WER is a much harsher metric, especially for short texts compared with CER.

B. TD-based Detection Results Comparison

The TD-based detection test obtained the WER's and CER's for benign and two adversarial audios, i.e., WER_{benign}, WER_{Carlini} and WER_{Ours} (CER_{benign}, CER_{Carlini} and CER_{Ours}). In this section, we show the detection result comparisons based on WER and CER, respectively. We use the AUC score to measure the detection result.

In the comparison, we choose $\alpha = 0.1$ in our proposed audio attack and choose $k = 0.5$ to be consistent with [15]. The comparison is given in Table II. Note the detection score for Carlini's audio attack is provided in [15].

As we can see in Table II, the TD-based detection method discriminates Carlini's adversarial audios accurately. However, the detection against our adversarial audios is not much better

TABLE II
COMPARISON OF TWO AUDIO ATTACKS IN TERMS OF DETECTION RESULTS MEASURED BY THE AUC SCORE. $\alpha = 0.1$ AND $k = 0.5$.

	Proposed Attack	Carlini's Method
Character Error Rate	62.15%	91.6%
Word Error Rate	66.18%	93.6%

TABLE III
WE GIVE THE ATTACK EFFICACY COMPARISON HERE. WE CALCULATE THE WER AND CER BETWEEN EACH ADVERSARIAL AUDIOS' TRANSCRIPTION AND THE ASSIGNED ADVERSARIAL TARGET (SEE TABLE I).

	Word Error Rate					*Character Error Rate*				
α	0.1	0.3	0.5	0.7	0.9	0.1	0.3	0.5	0.7	0.9
Carlini [12]	0.13					0.04				
Ours	0.21	0.12	0.09	0.14	0.11	0.09	0.04	0.03	0.03	0.03

than random guessing. Thus, our proposed audio attack has much lower detectability compared with Carlini's audio attack when facing the TD-based detection method. Moreover, we can further decrease the value of α to encourage even lower detectability for the proposed method.

C. Attack Efficacy and Perturbation Magnitude Comparison

A successful audio attack method needs to achieve the attack target (i.e., see Table I) with minimal perturbation magnitude. In this section, we show the comparisons in terms of attack efficacy and perturbation magnitude.

We first assess whether the proposed audio attack method can successfully achieve the assigned attack target or not. In Table III, we give the attack efficacy comparison compared with Carlini's audio attack. The attack efficacies of two audio attacks are measured by the consistency between adversarial audios' transcriptions and corresponding assigned attack targets. The consistency is measured by WER/CER and is averaged over 100 audio samples. From Table III, we observe that our audio attack's efficacy is worse (i.e., WER/CER is higher) than Carlini's audio attack only at $\alpha = 0.1$. However, this results only means that our method made 2.47 errors out of 27.47 characters on average and Carlini's method made 1.09 errors (1.05 errors oppose to 0.65 errors out of 5 words on average, note that WER is a much harsher metric than CER especially for short text).

We next compare the perturbation magnitudes of two audio attacks. We measure the perturbation in Decibels (dB) to be consistent with [12]. Decibels (dB) is a logarithmic scale that measures the relative loudness of an audio sample:

$$\mathrm{dB}(x) = \max_i 20 \log_{10} x_i$$

The perturbation magnitude of $\sigma = |X - X^a|$ to the original audio X is given by:

$$\mathrm{dB}_X(\sigma) = \mathrm{dB}(\sigma) - \mathrm{dB}(X)$$

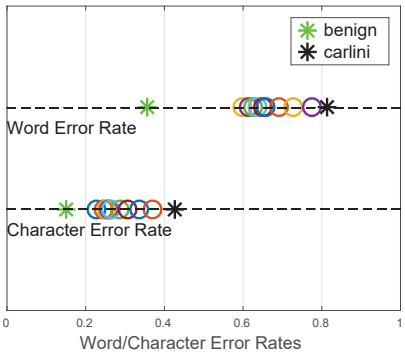
The perturbation magnitude is a relative quantity and is a negative number where smaller values indicate quieter perturbations because the perturbation σ introduced is quieter than the original signal X. The average relative perturbation magnitude of proposed audio attack is -30dB for $\alpha = 0.1$ and -45dB for Carlini's audio attack [12]. As we can see, we sacrificed some perturbation magnitude in exchange for lower detectability against the TD-based detection.

We also highlight our main results in Table IV which shows the two adversarial audios' transcriptions for Carlini's and our audio attack when different k's are applied. The complete adversarial audio transcription texts are shown in $k = 1$ rows. In Table IV, we observe that our method provides better adversarial audio transcriptions as k decreases whereas Carlini's adversarial audio transcriptions quickly become incoherent (i.e., easier to detect when the audio becomes incoherent). As a result, the proposed adversarial audios exhibit lower WER/CER than Carlini's adversarial audios and have lower detectability when applying a TD-based detection test.

D. Influence of Hyper-Parameter α

In this section, we show the hyper-parameter α's influence on the proposed audio attack. Specifically, we give the proposed adversarial audios' averaged WER's and CER's for various α values. We also provide the averaged CER and WER for benign audios and Carlini's adversarial audios for reference.

In Figure 8, the circles with different colors correspond to the proposed audio attack with different α values. The leftmost circle corresponds to $\alpha = 0.1$ and rightmost circle corresponds to $\alpha = 0.9$. As we can see, our audio attack's WER/CER decreases as α decreases. Furthermore, the CER statistics become very close to the benign audio's CER for $\alpha = 0.1$.



(a) Error rates

Fig. 8. The WER and CER averaged over different k's for benign and two adversarial audios. The circles with different colors represent our methods with different α's. The circles moves leftwards as α decreases.

V. CONCLUSION

Deep neural networks (DNNs) have excelled automated speech recognition tasks and have become the state-of-the-art in their field. Several works have shown that DNNs, such

TABLE IV
EXAMPLES OF PROPOSED METHOD COMPARED WITH THE METHODOLOGY IN [12] USING "ADVERSARIAL TARGETS" IN TABLE I FOR $\alpha = 0.1$, THE SPLITTING RATIO k TAKES DIFFERENT CHOICES. $k = 1$ SIMPLY MEANS NO SLICING.

k	*Proposed attack: partial transcription*	*Attack in [12]: partial transcription*
$k = 0.3$		sf
$k = 0.35$		
$k = 0.4$	h	l
$k = 0.45$	hel	ol
$k = 0.5$	helg	ilo
$k = 0.55$	hellg	elkot
$k = 0.6$	hello	elkutg
$k = 0.65$	hellogo	elgutgo
$k = 0.7$	hello go	elotgoop
$k = 0.75$	hello gol	elotgole
$k = 0.8$	hello goog	elotgole
$k = 0.85$	hello gogl	ellot google
$k = 0.9$	hello gogl	hello google
$k = 0.95$	hello gogl	hello google
$k = 1$	hello google	hello google

k	*Proposed attack: partial transcription*	*Attack in [12]: partial transcription*
$k = 0.3$	ti is an dver	the ma n dver
$k = 0.35$	this is an advers	the man averk
$k = 0.4$	thi is an advers	the ma an averk
$k = 0.45$	thi is an adversa	the me an adverot
$k = 0.5$	this is an adversai	the man everycont oude
$k = 0.55$	this is an adversaria	the mandedvery conti youdius
$k = 0.6$	this is an adversarial	the me an avercontds
$k = 0.65$	this is an adversarial	the me an avertse
$k = 0.7$	this is an adversariale	the ma an aversar
$k = 0.75$	this is an adversarialea	this i an adversariral
$k = 0.8$	this is an adversarialea	this i an adversaryfral
$k = 0.85$	this is an adversarial eam	thi mi an adver otsarifalxam
$k = 0.9$	thi is an adversarial exampl	the maan edvery contisarial examply
$k = 0.95$	thi is an adversarial exampl	the i an edvery conti oudisarial exampley
$k = 1$	this is an adversarial exampl	this is an adversarial example

k	*Proposed attack: partial transcription*	*Attack in [12]: partial transcription*
$k = 0.3$	helgole plea	hte
$k = 0.35$	helo gogle pleasea	trag
$k = 0.4$	helo google please can	stragong
$k = 0.45$	helo google please cance	straage ginl
$k = 0.5$	helo google please cancel m	strage inlee
$k = 0.55$	helo google please cancel m med	strag ginlee
$k = 0.6$	hello google please cancel m medical	strig goleple a caov
$k = 0.65$	hello google please cancel m medical	straage finglile o ca as
$k = 0.7$	helo google please cancel my medicalpon	srig gonlile o ca ask ym
$k = 0.75$	hello google please cancel my medical appoin	srig gonlile o ca aske rymma
$k = 0.8$	hello google please cancel my medical appointm	ri e gioglepleas i cavase hymemal
$k = 0.85$	hello google please cancel my medical appointme	dil e gioglepleas i cavase hymadmal
$k = 0.9$	hello google please cancel my medical appointme	il e gioglepleas i caase hymemal
$k = 0.95$	hello google please cancel my medical appointme	elli google pleas i caace hy memalunt
$k = 1$	hello google please cancel my medical appointment	hello google please cancel my medical appointment

as DeepSpeech, can be easily fooled by adversarial inputs and more recent work has shown that adversarial audio can be detected with TD. In this work, we investigated generating adversarial audio that can avoid the TD-based techniques for detecting such audios. We argue that our audio signals are more difficult to detect because we exploit the intrinsic property of LSTMs in the DeepSpeech models. The experiments showed the drastically reduced detectability in the face of the TD-based detection method comparing with its counterpart in [12] on the benchmark dataset. We also observed that the proposed attacking algorithm's generated adversarial audio has negligible differences when comparing with corresponding benign audio on character level and low difference on word level.

ACKNOWLEDGEMENTS

This work was supported by grants from the Department of Energy under #DE-NA0003946 and National Science Foundation CAREER #1943552.

REFERENCES

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[2] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Y. Ng, "Deep speech: Scaling up end-to-end speech recognition." https://arxiv.org/abs/1412.5567.

[3] R. Socher, C. C.-Y. Lin, A. Ng, and C. Manning, "Parsing natural scenes and natural language with recursive neural networks," in *ICML*, 2011.

[4] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations*, 2014.

[5] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *International Conference on Learning Representations*, 2013.

[6] J. Kos, I. Fischer, and D. Song, "Adversarial examples for generative models," in *IEEE Symposium on Security and Privacy Workshops*, 2018.

[7] A. Arnab, O. Miksik, and P. H. Torr, "On the robustness of semantic segmentation models to adversarial attacks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[8] R. Jia and P. Liang, "Adversarial examples for evaluating reading comprehension systems," in *Conference on Empirical Methods in Natural Language Processing*, 2017.

[9] Q. Jiang, X. Zhao, and K. Huang, "A feature selection method for malware detection," in *IEEE International Conference on Information and Automation*, pp. 890–895, 2011.

[10] K. Rieck, P. Trinius, C. Willems, and T. Holz, "Automatic analysis of malware behavior using machine learning," *Journal of Computer Security*, vol. 19, no. 4, pp. 639–668, 2011.

[11] U. Pehlivan, N. Baltacı, C. Acartürk, and N. Baykal, "The analysis of feature selection methods and classification algorithms in permission based android malware detection," in *IEEE Symposium Series on Computational Intelligence in Cyber Security*, pp. 1–8, 2014.

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

[13] M. Alzantot, B. Balaji, and M. Srivastava, "Did you hear that? adversarial examples against automatic speech recognition," in *31st Conference on Neural Information Processing Systems*, 2017.

[14] H. Xu, Y. Ma, H. Liu, D. Deb, H. Liu, J. Tang, and A. K. Jain, "Adversarial attacks and defenses in images, graphs and text: A review," *https://arxiv.org/pdf/1909.08072.pdf*, 2019.

[15] Z. Yang, B. Li, P.-Y. Chen, and D. Song, "Characterizing audio adversarial examples using temporal dependency," in *International Conference on Learning Representations*, 2019.

[16] H. Yakura and J. Sakuma, "Robust audio adversarial example for a physical attack," in *International Joint Conferences on Artificial Intelligence*, 2019.

[17] R. Taori, A. Kamsetty, B. Chu, and N. Vemuri, "Targeted adversarial examples for black box audio systems," in *IEEE Security and Privacy Workshops*, 2019.

[18] Y. Luo, X. Boix, G. Roig, T. Poggio, and Q. Zhao, "Foveation-based mechanisms alleviate adversarial examples," *arXiv preprint arXiv:1511.06292*, 2015.

[19] G. K. Dziugaite, Z. Ghahramani, and D. M. Roy, "A study of the effect of jpg compression on adversarial images," *arXiv preprint arXiv:1608.00853*, 2016.

[20] C. Guo, M. Rana, M. Cisse, and L. van der Maaten, "Countering adversarial images using input transformations," in *International Conference on Learning Representations*, 2018.

[21] J. Buckman, A. Roy, C. Raffel, and I. Goodfellow, "Thermometer encoding: One hot way to resist adversarial examples," in *International Conference on Learning Representations*, 2018.

[22] K. Rajaratnam and J. Kalita, "Noise flooding for detecting audio adversarial examples against automatic speech recognition," in *IEEE International Symposium on Signal Processing and Information Technology*, pp. 197–201, 2018.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, 1997.

[24] M. Schuster and K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, 1997.

[25] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *International Conference on Machine Learning*, 2006.

[26] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.