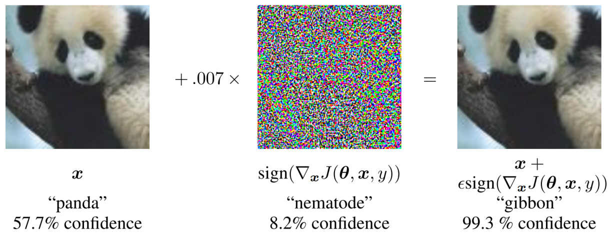
Detecting Adversarial Audio via Activation Quantization Error

Heng Liu and Gregory Ditzler

Abstract— **The robustness and vulnerability of Deep Neural Networks (DNN) are quickly becoming a critical area of interest since these models are in widespread use across real-world applications (i.e., image and audio analysis, recommendation system, natural language analysis, etc.). A DNN's vulnerability is exploited by an adversary to generate data to attack the model; however, the majority of adversarial data generators have focused on image domains with far fewer work on audio domains. More recently, audio analysis models were shown to be vulnerable to adversarial audio examples (e.g., speech command classification, automatic speech recognition, etc.). Thus, one urgent open problem is to detect adversarial audio reliably. In this contribution, we incorporate a separate and yet related DNN technique to detect adversarial audio, namely model quantization. Then we propose an algorithm to detect adversarial audio by using a DNN's quantization error. Specifically, we demonstrate that adversarial audio typically exhibits a larger activation quantization error than benign audio. The quantization error is measured using character error rates. We use the difference in errors to discriminate adversarial audio. Experiments with three the-state-of-the-art audio attack algorithms against the DeepSpeech model show our detection algorithm achieved high accuracy on the Mozilla dataset.**

I. Introduction

There is an ever-growing need to deploy Deep Neural Networks (DNN) in complex tasks of prediction and forecasting in diverse settings (e.g., image classification [1], semantic segmentation [2], speech recognition [3], autonomous driving [4], etc.), given their superior benchmark performances in real-world applications. Despite DNN's success in real-world applications, recent work has shown that human-imperceptible adversarial perturbations can easily fool DNNs. For example, Figure 1 shows Goodfellow et al.'s classic example of a panda image that has been perturbed with a signal that is not observed by the human eye [5]; however, the DNN makes drastically different classifications of the images. In addition to image classification tasks, attacks against other DNN-based applications are also explored extensively, such as semantic segmentation, image captioning, text classification, and medical prediction [6]–[9]. The focus of this work is on audio analysis, and DNN-based architectures have outperformed traditional HMM-based techniques in benchmarks for speech recognition and machine translation tasks. Unfortunately, the DNNs designed for these speech tasks are also susceptible to adversarial attacks. Unfortunately, despite the importance of audio applications, the vast majority of adversarial machine learning research has focused on image analysis attacks [10].


$+ .007 \times$
$=$
$\boldsymbol{x}$
"panda"
57.7% confidence
$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"nematode"
8.2% confidence
$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"gibbon"
99.3 % confidence

(a) Adversarial example for image classification

Fig. 1: Goodfellow et al.'s demonstration of fast adversarial example generation applied to GoogLeNet (Szegedy et al., 2014a) on ImageNet. By adding an imperceptibly small perturbation to the image, the classification result of GoogLeNet for "panda" has changed to "gibbon".

Targeted adversarial examples against convolutional neural networks and recurrent neural networks can be quite successful against speech recognition/machine translation tasks. Generically, the attack algorithm generates an adversarial audio X^a through a gradient-based optimization, which will lead to an incorrect output Y^a by minimizing the loss. For example, let X be an audio signal that is represented as a sequence of length m, where $X = \{X[1], \dots, X[t] \dots, X[m]\}$ and Y be the correct transcription of X, which could be $Y =$ "*Can you pick up the car at 5PM*?". The adversary seeks to generate audio X^a such that $\|X - X^a\|_2^2$ is arbitrarily small and $Y^a =$ "*Can you please cancel my medical appointment tomorrow?*" Thus, the adversary wants to add a small perturbation to the X; however, the transcription is drastically different from the original transcription. The adversary's goal is to form an optimization problem over a variable σ where $X^a = X + \sigma$ and σ is an adversarial perturbation vector. Mathematically, the optimization is given by:

$$X^* = \arg\min_{\sigma \in \mathbb{R}^m} \{\text{Loss}(f_\theta(X^a = X + \sigma), Y^a)\}$$

where f_θ is a pre-trained neural network with parameters θ and X is a benign audio with $f_\theta(X) \neq Y^a$. Note that σ is a sequence of length m where its entries are not all the same value (i.e., σ is a time-varying waveform).

There are typically two types of tasks in audio analysis: (i) speech-to-label, and (ii) speech-to-text tasks. Speech-to-label tasks receive an input audio X that corresponds to a class label C (e.g., "yes" or "no"). An adversarial audio X^a acts against a task by subverting the authentic class label C while remaining close to X. Alzantot et al. proposed an attack algorithm against a speech command classification model. Their attack added an imperceptible noise to the original audio signal [11] (i.e., simply changing the least significant

H. Liu and D. Ditzler are with the Department of Electrical Engineering, The University of Arizona, Tucson, AZ 85721, USA.

Email: {hengl, ditzler}@email.arizona.edu

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

bits in the audio signal). Alzantot et al.'s attack led to an 87% success-rate simply by adding small background noise without having to know the underlying model parameter and architecture. On the other hand, the speech-to-text task takes input audio X and generates a sequence of text Y. A speech-to-text attack against DeepSpeech was proposed by Carlini et al. [10]. Carlini's gradient-based attack arbitrarily manipulates an audio's machine transcription Y (i.e., $Y^a \neq Y$) by injecting imperceptible perturbations to the original signal (i.e., $\min ||X^a - X||_2^2$). The perturbation σ can be easily found using backpropagation such that the new signal leads to arbitrary transcriptions with software such as DeepSpeech. It is important to understand that these adversarial audio signals are almost identical to X in both time and frequency domains.

Although there is an urgent need for adequate defenses against adversarial audio examples since recognition/machine translation tasks rely more and more heavily on DNN based models, such countermeasures remain severely under-explored. Early contributions have adopted image defense strategies (e.g., feature selection/transformation) to filter the adversarial perturbations in audio. Unfortunately, feature selection is vulnerable in a malicious environment [12]. Empirical works showed that simple input transformation delivers very limited security [13]. Although there are numerous defense strategies proposed in the image domain, detecting adversarial examples in the audio domain is a different and challenging task. This is because the audio are sequentially-structured and are inter-correlated in the time axis.

DNN quantization is a popular technique for compressing model size and reducing computational complexity. We noted that DNN quantization can be beneficial for a DNN's robustness. In this contribution, we propose an approach to detect malicious audio signals by using a quantized neural network. Specifically, we empirically show that benign and adversarial audio exhibit significantly different activation quantization error levels in a neural network. These differences in word/character error rates inspired a straightforward detection model that can accurately differentiate between benign and adversarial audios. Finally, the proposed adversarial audio detection method is benchmarked against three the-state-of-the-art audio attack algorithms.

II. Related Works

In this section, we briefly review the technical details of the neural network model quantization, adversarial audio generation and detection of malicious audio signals.

A. Adversarial Audio Examples

The adversarial audio examples against DeepSpeech that were generated by Carlini and Wagner were the first targeted speech-to-text audio attacking algorithm (i.e., explicitly specify the attack target) [10]. These adversarial audios are particularly effective, given that the slight noise is utterly inaudible to a human ear. Unfortunately, the adversarial perturbations fail to attack when played over-the-air. In [14], Yukura and Sakuma take into account the impact (e.g., white noise, band filtering, etc.) when audios are played over-the-air then devised a robust audio attack against DeepSpeech. Moreover, although adversarial audio in [10] achieved an almost 100% success rate, Carlini and Wagner assumed a white-box setting, which requires detailed information of the victim's model. In [15], Taori et al. proposed a black-box audio attack by combining the approaches of both genetic algorithms and gradient estimation. In this contribution, we test the proposed detection algorithm against all the above three attacking methods.

Feature transformations are widely adopted as countermeasures against an adversary in real-world tasks (e.g., image quantization, filtering, image reprocessing, autoencoder reformation) [16], [17]. These feature transformations are widespread due to their low cost and the fact they can be used with various DNN architectures. Feature transformation aims to filter the adversarial perturbation of the raw image. While feature transformations are effective on images, they provide limited security against adversarial audio [13]. Yang et al. proposed an empirical test to discriminate adversarial audio by measuring the intrinsic temporal dependency [13]; however, this detection technique can be easily fooled by suppressing the temporal dependency in adversarial audio. Rajaratnam and Kalita proposed to flood particular frequency bands with random noise to detect adversarial audio [18]. Unfortunately, an adversary can specify the frequency bands that carry the adversarial perturbations to evade noise flooding [14].

B. Neural Network Model Quantization

DNN models have achieved remarkable accuracy in a variety of real-world applications, albeit at the expense of high computational cost. For example, the ILSVRC 2015 competition winner ResNet has 152 layers and GBs of parameters [19]. These neural network sizes pose a tremendous challenge to real-time implementations on a resource-constrained platform (e.g., FPGA chips, mobile devices, etc). In the past decade, a plethora of contributions have presented techniques to compress DNN models (i.e., reduce memory and computation requirements without incurring significant learning loss) [20]. In particular, weight and activation quantization techniques showed significant model size reduction with limited accuracy degradation [21].

Huang and Sung demonstrated the possibility of quantizing DNN weights to 1 bit (binary) or 2 bits (ternary), which allows a DNN to fit efficiently on resource-constrained platforms [22]. Zhu further improved the effectiveness of weight quantization by ternarizing weights using a statistical distribution of weight values [23]. On the other hand, activation quantization can also improve the model quantization by reducing the activations during the DNN model's inference. Hubara et al. proposed an activation-binarized DNN model [24].

Several different methods can be considered when a neural network needs to be quantized. Weight and activation quantization is equivalent to discretizing the DNN hypothesis

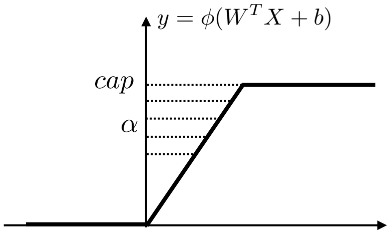


Fig. 2: In the DNN activation quantization, the activation values are quantized after clipping. cap defines the maximal activation value, α is the calculated grid size for a bit width k.

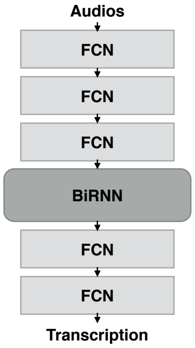


Fig. 3: The structure of DeepSpeech. FCN is the fully connected layer and BiRNN is the bidirectional RNN layer.

space of the loss function. Therefore, such quantization schemes cause quantization errors when compared to full precision models. When training quantized DNN models, such quantization error can be compensated by quantization-aware retraining [20]. In this contribution, we begin with a pre-trained full precision DNN model for speech recognition then we incorporate the DNN activation quantization to discriminate against the adversarial audio.

III. Adversarial Audio Detection via Activation Quantization Error

In this section, we examine the neural network activation quantization's impact on audio by comparing the activation quantization errors on benign and adversarial audio. After observing the differences between the two quantization errors, we propose a new method to discriminate against the adversarial audio via the activation quantization error.

A. Activation Quantization Errors on Audio

The first step to quantize a neural network's activation is to clip the activations (i.e., bound the range of the output). While some activation functions already have a boundedness property, many activations have the property of unboundedness (e.g., ReLu). Specifically, the activation $y = \phi(W^T X + b)$ is clipped to avoid extreme activation values then for a given bit width k the activation range is discretized to 2^k grids. This quantization procedure is the same as quantization that is performed in classical digital signal processing [25]. Lastly, the activation values are mapped to lower precision values. Figure 2 shows the quantization of ReLu activation. The activation is first clipped with the predefined "cap", then the activation range is discretized with grid size equals α. The activations are mapped to lower precision values before feeding to the next layer.

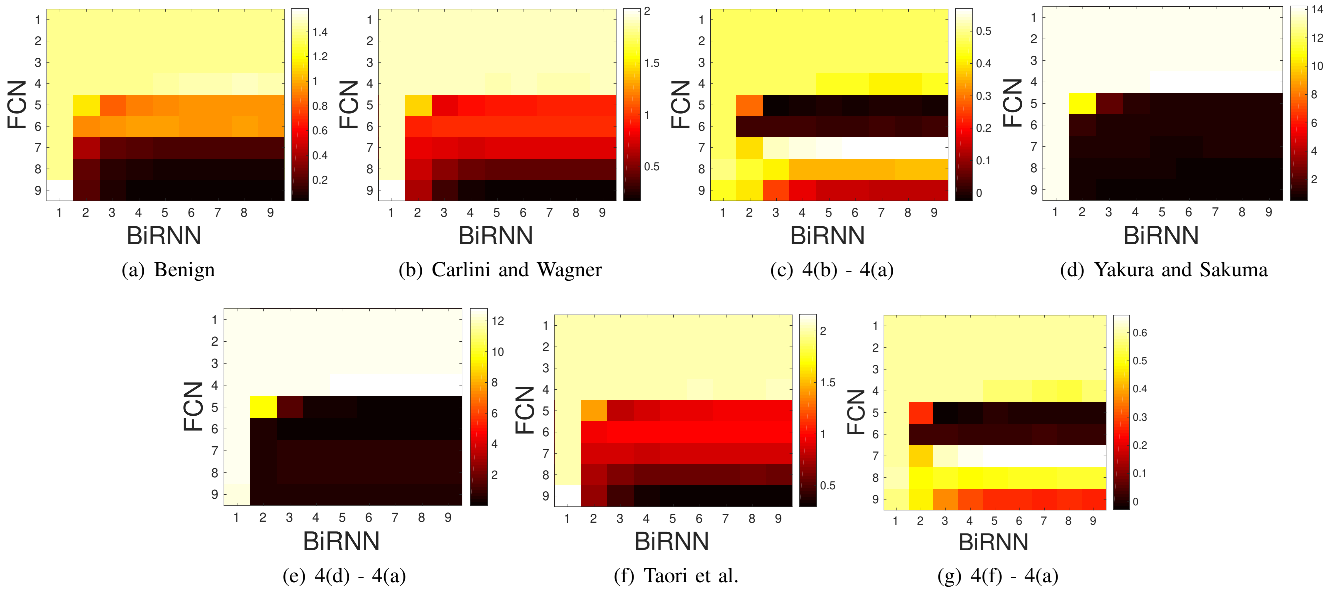
We use the open-source DeepSpeech model as our victim speech recognition model and the Mozilla Common Voice dataset as our benchmark. DeepSpeech[1] is a lite bidirectional recurrent neural network that achieves the state-of-the-art performances on speech recognition tasks [26]. The DeepSpeech neural network is a mixture of five Fully Connected Layers (FCN) and one Bidirectional RNN (BiRNN) layer. Figure 3 shows the architecture of the DeepSpeech neural network.

Our motivation to exploit neural network quantization is a result of the significant difference in the benign and adversarial audio's quantization errors. The quantization error is referred to as the performance degradation on a quantized model comparing with a full-precision model. Specifically, in the context of speech recognition, which usually outputs transcripts, the quantization error can be quantified by the transcripts inconsistency (transcripts from quantized model and full-precision model). In our analysis, we measure the transcripts inconsistency using the Character Error Rate (CER). The CER is defined by $\frac{S+D+I}{N}$, where S, D, and I are the hypothesis transcript's substitutions, deletions, insertion errors, N is the number of characters in the reference transcript. Hypothesis and reference transcripts refer to transcripts from quantized and full-precision models, respectively. Note the CER can be larger than one when there are more errors than the character number in the reference.

We begin our analysis by using three attack algorithms [10], [14], [15] to generate the adversarial audio for 100 benign audio clips. Once the adversarial audio is generated, then the audio is passed through the quantized and full-precision networks, and we measure the activation quantization errors for both benign and adversarial audio. We quantize the FCN and BiRNN layers separately with various bit widths: $k = 1, 2, \ldots, 9$ when the activation is quantized in DeepSpeech.

The CER results across various quantization bit widths for benign audio and three types of adversarial audio are shown in Figure 4. Specifically, Figure 4(a) is the benign audio averaged CER. Figure 4(b), 4(d), and 4(f) are the averaged CER's for three attack algorithms, respectively. Figure 4(c), 4(e), and 4(g) are the CER differences (adversarial audio average CER minus benign audio average CER). One observation and conclusion to draw from this experiment is that the benign audio achieves an overall lower CER than three types of adversarial audio; however, the differences vary across different quantization bit widths, which is expected. This conclusion is more significant for the attack algorithm in [14]. In other words, the adversarial audio incurs larger activation quantization errors, which can detect adversarial audio.

[1] The pretrained DeepSpeech model can be found at `https://github.com/mozilla/DeepSpeech`



(a) Benign

(b) Carlini and Wagner

(c) 4(b) - 4(a)

(d) Yakura and Sakuma

(e) 4(d) - 4(a)

(f) Taori et al.

(g) 4(f) - 4(a)

Fig. 4: Here are the average quantization errors for benign (plot 4(a)) and three adversarial audio (plots 4(b), 4(d), 4(f)) quantitively measured by CER. The x-axis and y-axis are the quantization bit widths for FCN layers and BiRNN layer, respectively. In plots 4(c), 4(e), 4(g) we showed the three CER differences between the adversarial and benign audio.

B. *Adversarial Audio Detection*

We here propose our adversarial audio detection algorithm based on the previous empirical analysis. We deem audio with activation quantization error higher (lower) than a threshold as adversarial (benign) audio. However, the optimal threshold δ for a given quantization bit widths (k_1 and k_2) need to be determined *a priori*. In our algorithm, we first empirically estimate the threshold and associated quantization bit widths that have the highest detection accuracy using a training/validation step. Then we use the validated threshold σ^* and associated bit widths k_1^*, k_2^* to detect adversarial audio. We propose our adversarial audio detection technique in Algorithm 1.

Algorithm 1 Adversarial Audio Detection

```
1:  Input: Benign audio for training: $X \in \mathcal{D}_1$, benign audio for validation: $X \in \mathcal{D}_2$, audio $\widehat{X}$ for detection, bit widths $W = \{1, 2, 3, \ldots, 9\}$.
2:  Initialization: Adversarial audio for training: $\widehat{\mathcal{D}}_1 = \{adver(X) : \forall X \in \mathcal{D}_1\}$, adversarial audio for validation: $\widehat{\mathcal{D}}_2 = \{adver(X) : \forall X \in \mathcal{D}_2\}$.
3:  for $k_1 \in W$ do
4:      for $k_2 \in W$ do
5:          $err_{\mathcal{D}_1,k_1,k_2} = \{\text{CER}(X, k_1, k_2) : \forall X \in \mathcal{D}_1\}$
6:          $err_{\widehat{\mathcal{D}}_1,k_1,k_2} = \{\text{CER}(X, k_1, k_2) : \forall X \in \widehat{\mathcal{D}}_1\}$
7:          $\delta_{k_1,k_2} = \frac{\max(err_{\mathcal{D}_1,k_1,k_2}) + \min(err_{\widehat{\mathcal{D}}_1,k_1,k_2})}{2}$
8:          $err_{\mathcal{D}_2,k_1,k_2} = \{\text{CER}(X, k_1, k_2) : \forall X \in \mathcal{D}_2\}$
9:          $err_{\widehat{\mathcal{D}}_2,k_1,k_2} = \{\text{CER}(X, k_1, k_2) : \forall X \in \widehat{\mathcal{D}}_2\}$
10:         $Accu_{k_1,k_2} = \text{Valid}(err_{\mathcal{D}_2,k_1,k_2} \bigcup err_{\widehat{\mathcal{D}}_2,k_1,k_2}, \delta_{k_1,k_2})$
11:     end for
12: end for
13: $k_1^*, k_2^*, \delta^* = \arg\max_{k_1,k_2,\delta_{k_1,k_2}} Accu_{k_1,k_2}$
14: # Testing if $\widehat{X}$ is adversarial audio
15: if $\text{CER}(\widehat{X}, k_1^*, k_2^*) \geq \delta^*$ then
16:     Return "Adversarial Audio"
17: else
18:     Return "Benign Audio"
19: end if
```

We here explain the Algorithm 1 in detail. In step 1, the inputs to the algorithm are datasets $\mathcal{D}_1$ and $\mathcal{D}_2$, which are the benign training and validation audio, respectively. Furthermore, we choose a range for the number of bit widths, W, involved in the activation quantization of the DeepSpeech model. We train the detection algorithm generating adversarial audio using an existing attack algorithm (e.g., see [10], [14], [15]). In step 2, the notation $adver(X)$ is a call to the adversarial audio generation algorithm for an audio X. These adversarial audio are given by the datasets $\widehat{\mathcal{D}}_1$ and $\widehat{\mathcal{D}}_2$.

Our detection algorithm implements different quantization bit widths on the fully connected (k_1) and BiRNN layers (k_2). We use two "for" loops that iterate two different levels of quantization on the network to determine the best validated threshold σ^* and associated k_1^*, k_2^*. Specifically, for a quantized network with k_1 bits on the FCN layers and k_2 bits on the BiRNN layer, we first calculate the CER's for the training benign audio $X \in \mathcal{D}_1$ and adversarial audio $X \in \widehat{\mathcal{D}}_1$ in steps 5 & 6. Then the detector calculates a threshold δ_{k_1,k_2} that is the average of the minimum and maximum CER from $\mathcal{D}_1$ and $\widehat{\mathcal{D}}_1$, respectively (see step 7). In steps 8 & 9, the CER's are calculated for the validating benign audio $X \in \mathcal{D}_2$ and adversarial audio $X \in \widehat{\mathcal{D}}_2$, which is used to validate δ_{k_1,k_2} by obtaining the detection accuracy $Accu_{k_1,k_2}$ in step 10. In step 13, the output of the detector is the threshold δ^*, and bit widths k_1^* and k_2^* associated with the maximal validation accuracy $Accu_{k_1,k_2}$. Finally, whether a test audio $\widehat{X}$ is benign or not is determined if the CER of the test audio is lower than δ^* or not.

IV. Experimental Results

In this section, we perform a comprehensive evaluation and demonstrate the efficacy of our detection algorithm. Specifically, we benchmark the Algorithm 1 against three state-of-the-art audio attack algorithms ([10], [14], [15]) on Mozilla Common Voice dataset.

A. Dataset and Evaluation Methods

We chose a subset from the Mozilla dataset released in [10][2]. There are 100 audio clips in the dataset that are sampled at 16KHz, which are the same data used in the works of Carlini et al. and Yang et al. [10], [13]. The duration of the audio clips are between 1.73s to 7.8s, with an average of 4s per clip. Table I shows the attack target sentences (i.e., the desired transcripts of the adversarial audio for a normal audio X). Note that these attack targets are the same ones that were used in [10], [13].

TABLE I: The targets for audio 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

In the experiments, because the Algorithm 1 requires a training/validation stage to determine the best threshold σ^* and associated quantization bit widths k_1^*, k_2^*, we test with various choices of the ratio of training/validation. We also show the training/validation data's scarcity's impact on the detector's performance. The training/validation ratio range is $0.1 \sim 0.8$.

The figures of performance for the detection algorithm are the testing accuracy, precision, recall, and area under the ROC (AUC) for each training/validation ratio. The accuracy is the ratio of correctly identified benign and adversarial audio in the testing data. The precision and recall are given by $\frac{tp}{tp+fp}$ and $\frac{tp}{tp+fn}$, respectively. However, we need to transform the CER's to a probabilistic score to calculate the Receiver Operating Characteristic (ROC). To perform the calculation of the ROC, we negate the CER's from benign and adversarial testing audio such that higher scores indicates benign audio. Then we transform the negated CER's on $[0, 1]$. The transformation is given in (1) where $\mathcal{T}$ denotes the testing audio (benign and adversarial).

$$\begin{aligned} \delta_{\text{max}} &= \max\left\{\text{CER}(X) : X \in \mathcal{T}\right\} \\ \delta_{\text{min}} &= \min\left\{\text{CER}(X) : X \in \mathcal{T}\right\} \\ \text{score}(X) &= \frac{CER(X) - \delta_{\text{min}}}{\delta_{\text{max}} - \delta_{\text{min}}} \end{aligned} \tag{1}$$

We should note that this step can be included into the end of Algorithm 1 to provide a confidence score.

The temporal dependency-based detection algorithm proposed by Yang et al. achieved a remarkable performance that outperformed many other detection techniques [13]. Their detection algorithm achieved AUC scores as high as 93.6% on the Mozilla dataset. In our experiment, we benchmark against Yang's detection algorithm, and we also use the open-source implementation of CER calculation.[3]

B. Experimental Results

Table II shows the detection performances for our algorithm against three different attack algorithms being used to generate adversarial audio. The results are averaged over 50 runs where the training/validation data are sampled from a bootstrap. We experiment with different training/validation ratios: $0.1 \sim 0.8$ with a step size equals 0.05 since our detection algorithm requires a training/validation stage. One of the first observations to draw from Table II is that the proposed detection algorithm achieved high detection accuracy, precision, recall, and AUC against each attack algorithm. The second observation is that as the ratio of the training/validation audio increases, so does the detection efficacy. Comparing with Yang's detection method, our detection algorithm achieved a higher AUC than [13] regardless of the training/validation ratio. The last observation is that the detection algorithm is not influenced by the training/validation data's scarcity. For example, the detection algorithm achieved at least 95% AUC when the training and validation set both have only five audio clips.

Figure 6 shows the CER's of the testing audio for each attack algorithm when the training/validation ratio is 10%. The x-axis is the CER's of testing audio. The black circles indicate benign audio CERs, and red crosses indicate adversarial audio CERs. The vertical dashed lines are the threshold from the training/validation stage. We provide the associated accuracy, precision, and recall in each plot. The primary observation to make from Figure 6 is that the validated threshold well separates the benign and adversarial audio CERs. Figure 5 shows the Receiver Operating Characteristic (ROC) curves for the three attack algorithms. Note that the ROC was generated with the training/validation ratio set to 0.5. The detector is quite efficient at detecting the audio samples that are adversarial.

V. Conclusions

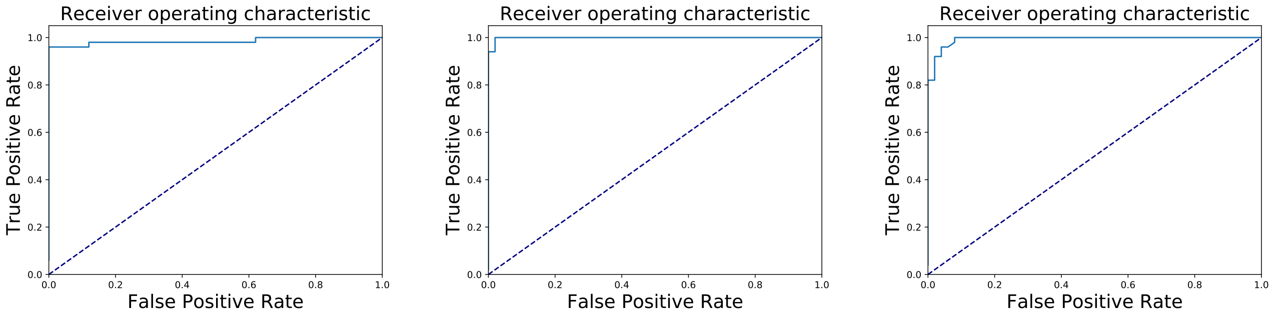
Deep neural networks (DNNs) have excelled at computer vision and machine translation tasks, and these algorithms have become a critical component in many data analysis pipelines. Unfortunately, DNNs are quite vulnerable to adversarial attacks, which questions the robustness of DNNs to imperceptible adversarial perturbations. There is a significant amount of work that evaluates the robustness of DNNs for image domains; however, there are far fewer works that examine the robustness of audio. While our work does not

[2] https://nicholas.carlini.com/code/audio_adversarial_examples

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

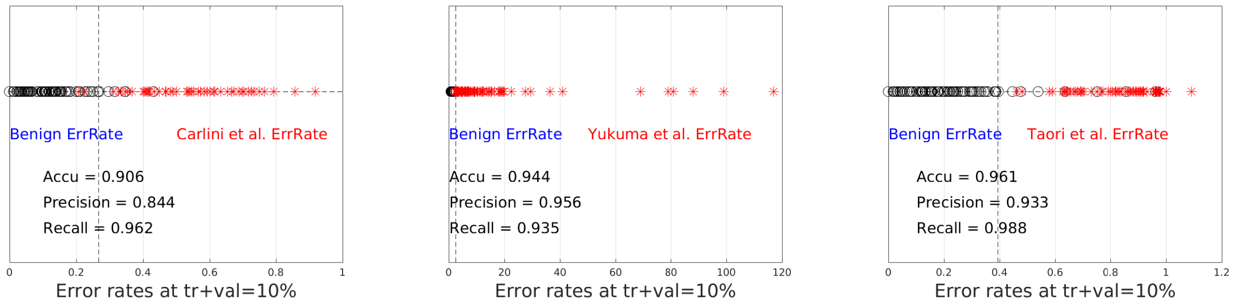
TABLE II: Figure of merit for the detection of adversarial audio that are generated by three the-state-of-the-art attack algorithms. For each training/validation ratio, we report the accuracy, precision, recall, and AUC scores. All experiments are averaged over 50 runs.

	Carlini and Wagner				**Yakura and Sakuma**				**Taori et al.**			
Tr+Val Ratio	Accu	Precisn	Recall	AUC	Accu	Precisn	Recall	AUC	Accu	Precisn	Recall	AUC
0.1	0.9106	0.9072	0.9186	0.9531	0.9382	0.9597	0.9244	0.9765	0.914	0.911	0.9186	0.9628
0.15	0.9315	0.9185	0.9457	0.9711	0.9381	0.947	0.9344	0.9713	0.9496	0.9463	0.9557	0.9863
0.2	0.9361	0.937	0.9375	0.9721	0.9407	0.9502	0.9347	0.9759	0.9485	0.9482	0.9527	0.9853
0.25	0.9388	0.9245	0.954	0.9716	0.9521	0.9512	0.9544	0.9883	0.9536	0.944	0.9648	0.9861
0.3	0.944	0.9448	0.9466	0.9798	0.9552	0.9608	0.952	0.99	0.9502	0.9606	0.9441	0.9874
0.35	0.9372	0.9322	0.9445	0.9763	0.9512	0.9529	0.9514	0.9832	0.9599	0.9513	0.97	0.9881
0.4	0.9447	0.94	0.9501	0.9776	0.954	0.954	0.9557	0.9881	0.9558	0.957	0.9572	0.9932
0.45	0.9426	0.9403	0.9469	0.9769	0.9479	0.9568	0.942	0.989	0.9554	0.9568	0.9567	0.991
0.5	0.9434	0.9424	0.9455	0.9781	0.9556	0.9624	0.9515	0.993	0.9604	0.9556	0.966	0.9934
0.55	0.9415	0.9391	0.9459	0.9757	0.9613	0.9695	0.9551	0.9944	0.9599	0.9552	0.9659	0.9907
0.6	0.9453	0.9365	0.9548	0.982	0.9585	0.9615	0.9576	0.9929	0.9562	0.9595	0.9554	0.9894
0.65	0.9469	0.9353	0.9604	0.9876	0.9668	0.9724	0.9625	0.9979	0.9554	0.9519	0.9608	0.9919
0.7	0.949	0.9393	0.9595	0.9786	0.964	0.9728	0.9577	0.9953	0.955	0.962	0.9521	0.9912
0.75	0.956	0.9432	0.9694	0.9858	0.9612	0.9592	0.9644	0.9967	0.9536	0.9496	0.9597	0.9899
0.8	0.9515	0.934	0.9696	0.9753	0.966	0.967	0.9668	0.9977	0.9575	0.949	0.9674	0.9944



(a) Carlini and Wagner (b) Yakura and Sakuma (c) Taori et al.

Fig. 5: The ROC curve of the detection algorithm against three audio attacking algorithms when training/validation ratio is 0.5. The dashed line represents the random chance line.



(a) Carlini and Wagner (b) Yakura and Sakuma (c) Taori et al.

Fig. 6: The CER's of benign audio against each type of adversarial audio when the train/validation ratio is 10%. X axis is the error rate level. The vertical broken line is the threshold calculated from training/validation. The evaluations metrics on testing audio are also reported in each figure.

focus on how to increase the robustness of the DNN applied to audio (i.e., the DeepSpeech model was used in this work), we presented an approach that can be used to accurately and reliably detect adversarial samples. In this work, we proposed an adversarial audio detection algorithm that exploits DNN's activation quantization error to detect when an adversary has generated a perturbation. The experiments on the Mozilla benchmark dataset demonstrated that our detection algorithm can achieve high accuracy, precision, and recall against three state-of-the-art audio attack algorithms. This paper is the first to examine the DNN model quantization's utility on detecting adversarial audio.

There are several paths to pursue in future work: (1) develop a theoretical basis or bound on the probability of error for the detector, and (2) incorporate the theoretical understandings from (1) into a DNN that can be used to increase the robustness of the model.

VI. Acknowledgements

G. Ditzler was supported by grants from the Department under #DE-NA0003946 and National Science Foundation (NSF) CAREER #1943552.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Neural Information Processing Systems*, 2012.

[2] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer Assisted Intervention*, 2015.

[3] G. Hinton, L. Deng, D. Yu, G. Dahl, A. rahman Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, and B. K. T. Sainath, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, 2012.

[4] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "Deepdriving: Learning affordance for direct perception in autonomous driving," in *IEEE International Conference on Computer Vision*, 2015.

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

[6] 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.

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

[8] 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.

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

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

[11] 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.

[12] H. Liu and G. Ditzler, "Data poisoning attacks against mrmr," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

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

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

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

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

[17] 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.

[18] 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.

[19] 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.

[20] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, "A survey of model compression and acceleration for deep neural networks," *IEEE SIGNAL PROCESSING MAGAZINE*, 2017.

[21] Z. Cai, X. He, J. Sun, and N. Vasconcelos, "Deep learning with low precision by half-wave gaussian quantization," in *The IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[22] K. Hwang and W. Sung, "Fixed-point feedforward deep neural network design using weights +1, 0, and 1," in *IEEE Workshop on Signal Processing Systems (SiPS)*, 2014.

[23] C. Zhu, S. Han, H. Mao, and W. Dally, "Trained ternary quantization," in *The International Conference on Learning Representations*, 2017.

[24] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Quantized neural networks: training neural networks with low precision weights and activations," *The Journal of Machine Learning Research*, vol. 18, 2017.

[25] A. V. Oppenheim, *Digital Signal Processing*. Prentice-Hall, 1975.

[26] 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.