INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Detecting Audio Attacks on ASR Systems with Dropout Uncertainty

Tejas Jayashankar[1,2]*, Jonathan Le Roux*[1]*, Pierre Moulin*[1,2]

[1]Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA
[2]University of Illinois at Urbana-Champaign, Champaign, IL, USA

Abstract

Various adversarial audio attacks have recently been developed to fool automatic speech recognition (ASR) systems. We here propose a defense against such attacks based on the uncertainty introduced by dropout in neural networks. We show that our defense is able to detect attacks created through optimized perturbations and frequency masking on a state-of-the-art end-to-end ASR system. Furthermore, the defense can be made robust against attacks that are immune to noise reduction. We test our defense on Mozilla's CommonVoice dataset, the UrbanSound dataset, and an excerpt of the LibriSpeech dataset, showing that it achieves high detection accuracy in a wide range of scenarios.
Index Terms: Automatic speech recognition, adversarial machine learning, audio attack, dropout, uncertainty distribution, noise reduction

1. Introduction

An adversarial example is an input to a neural network designed by an adversary to produce an incorrect or malicious output [1]. Early work on adversarial machine learning has shown that a small and imperceptible optimized perturbation to an image can cause misclassification by neural networks [2]. The field has further expanded to tasks such as image segmentation [3], reinforcement learning [4], and reading comprehension [5].

Recently, there has been growing interest in creating adversarial audio examples for automatic speech recognition (ASR) systems. Carlini and Wagner [6] showed that an audio sample can be perturbed slightly to cause mistranscription by an ASR engine. Building on this model, Qin et al. [7] and Schönherr et al. [8, 9] created nearly imperceptible audio attacks by leveraging the principle of auditory masking [10]. There have also been attempts to create attacks embedded in ultrasound frequencies [11] as well as phonetically constrained attacks [12].

Adversarial machine learning exploits a vulnerability of neural network models but also provides an avenue for making models more robust and formulating defenses against such attacks. There has been a significant effort in understanding the underlying mechanism of adversarial attacks to formulate effective defenses against attacks [13, 14], however such work has largely focused on domains other than audio.

In this paper, we propose a defense against adversarial audio attacks based on dropout. Dropout [15] is heavily used as an effective regularizer for neural network training, particularly in ASR systems. There have been successful attempts at using dropout as a defense in the image domain [16]. We here investigate whether similar principles can be applied to ASR, where varying sequence lengths pose an additional challenge. While the analysis of discrepancies in dropout outputs has been investigated to model uncertainty in ASR hypotheses [17], to our knowledge this is its first use as a defense against audio attacks.

This work was performed while T. Jayashankar was an intern and P. Moulin on sabbatical at MERL.

2. End-to-end automatic speech recognition

Many recent ASR systems obtaining state-of-the-art results are based on end-to-end architectures [18]. In contrast with conventional hybrid ASR systems, which consist in multiple complex modules such as acoustic, lexicon, and language models, end-to-end systems typically use a single deep network trained to directly map an input audio sample to a sequence of words or characters, alleviating the need for expert knowledge to build competitive systems.

The most popular end-to-end ASR approaches are connectionist temporal classification (CTC) [19, 20], attention [21], CTC/attention [22], RNN-T [23], and the Transformer [24, 25]. Since these models are differentiable, they can be trained with backpropagation, which is appealing due to the ease with which the model parameters can be updated by employing the chain rule of differentiation. However, this can also be a weakness, because an adversary may craft an adversarial input to fool a model into producing a wrong or a malicious output by backpropagating through it in order to minimize an error loss between the output of the model and the desired output.

In this paper, we focus on a CTC based architecture, as implemented in Mozilla's DeepSpeech system [20]. That system has been used in past work on adversarial audio attacks [6] and is publicly available, making it a convenient subject for our study. However, the methods we detail can be applied to models based on other end-to-end architectures as well.

3. Dropout

Dropout [15] is a regularization technique used to make neural networks robust to different inputs. Dropout deactivates a certain number of neurons in a layer, i.e., the weights corresponding to the neurons are set to zero. In each training iteration, a layer with dropout rate p drops neurons uniformly at random with probability p. During inference, dropout is typically turned off, and the learned weight matrices are scaled by p so that the expected value of an activation is the same as during training. Intuitively, dropout enables the neural network to learn various internal representations for the same input and output pair.

Adversaries typically exploit loopholes within a network by crafting an input perturbation such that small finely-tuned differences accumulate within the network to eventually result in a malicious output. Since these adversarial attacks are often created based on knowledge of the underlying architecture of the model, we hope to disarm such attacks by perturbing that architecture via a random process like dropout.

4. Adversarial attacks on ASR systems

In this section, we present the various adversarial audio attacks that we use in our experiments. We first introduce the Carlini & Wagner attack (CW attack) as it forms the foundation for the other attacks that we consider.

http://dx.doi.org/10.21437/Interspeech.2020-1846

4.1. Carlini & Wagner Attack

Given an original waveform x, Carlini and Wagner [6] propose to construct a waveform $x' = x + \delta$ such that x and x' sound nearly the same but are transcribed differently by an ASR engine. The perturbation δ is optimized such that the perturbed waveform $x + \delta$ is transcribed as a specific alternate (typically malicious) target sentence t with the least distortion, by minimizing a recognition loss function $\ell(x+\delta, t)$ (here, a CTC loss) under the constraint that the peak energy of the perturbation be at least τ dB smaller than that of the original waveform:

$$\min_{\delta} \ell(x + \delta, t) \text{ s.t. } \mathrm{dB}(\delta) \leq \mathrm{dB}(x) - \tau, \tag{1}$$

where $\mathrm{dB}(x) = 20 \max_i \log(|x_i|)$. Because ℓ is differentiable, backpropagation is easily performed. During optimization, the values of δ are limited to avoid clipping, and the threshold τ is progressively decreased to strengthen the constraint.

4.2. Dropout Robust Attack

Adversarial examples generated by the CW attack are optimized to be transcribed as certain target sentences at test time, with the model in inference mode, and they are thus typically optimized through the model with dropout turned off. A key insight is that if inference is performed with dropout turned on, adversarial examples tend to be transcribed as incorrect or garbled sentences, and dropout may thus be used to detect them. But we may consider a dropout robust (DR) attack including dropout in the construction of the adversarial example, so that the optimization procedure may have the chance to account for it. Since the adversarial example should be transcribed as target sentence t both with and without dropout turned on at inference time, the loss in (1) is replaced by a multi-task loss formulated as

$$\min_{\delta} \ell(x+\delta, t) + \beta \ell_{p_{\mathrm{DR}}}(x+\delta, t) \text{ s.t. } \mathrm{dB}(\delta) \leq \mathrm{dB}(x) - \tau, \tag{2}$$

where $\ell_{p_{\mathrm{DR}}}(x + \delta, t)$ is the same loss as ℓ except that dropout is turned on with a rate p_{DR}, and β is a weight (we used $\beta = 1$). Dropout is applied to all layers except the LSTM layers.

4.3. Noise Reduction Robust (NRR) Attack

Many audio systems perform pre-processing steps involving denoising to clean the input audio signal. We observed that denoising could partially and often completely eliminate the perturbation in the CW adversarial samples. Thus, denoising could act in itself as an effective defense for the vanilla CW attack.

To make the attack more robust, we trained the adversary to transcribe as the target sentence t with and without a preprocessing denoising stage by backpropagating through spectral subtraction [26], which was chosen for its simplicity:

$$\min_{\delta} \ell(x+\delta, t) + \beta \ell_{\mathrm{ss}}(x+\delta, t) \text{ s.t. } \mathrm{dB}(\delta) \leq \mathrm{dB}(x) - \tau, \tag{3}$$

where $\ell_{\mathrm{ss}}(x + \delta, t)$ is the same loss as ℓ except that the network processes the perturbed input after spectral subtraction denoising. We also tried to make this attack robust to dropout as above, but the optimization failed to converge in a reasonable time, illustrating the difficulty to find a solution under such constraints.

We did not experiment with neural network denoising algorithms as these could themselves be susceptible to adversarial attacks through similar optimization procedures.

4.4. Imperceptible Audio Attack

Recently, Qin et al. [7] devised a new attack on ASR systems based on frequency masking, the phenomenon whereby a softer sound (the maskee) is rendered inaudible by a louder sound (the masker) [10]. The vanilla CW attack is modified to enforce that the power spectral density p_δ of the perturbation in the short-time Fourier transform (STFT) domain must fall below the masking threshold θ_x of the original audio sample. The complete optimization problem is formulated as

$$\min_{\delta} \ell(x + \delta, t) + \alpha \sum_{k=0}^{\lfloor \frac{N}{2} \rfloor} \max\{p_\delta(k) - \theta_x(k), 0\}, \tag{4}$$

where α controls the relative importance of the term making the perturbation imperceptible, and N is the STFT window size.

After first optimizing with $\alpha = 0$ to find a perturbed sample transcribing as t, α is slowly increased to gradually satisfy the imperceptibility constraint by fine-tuning the perturbation.

4.5. Urban Sound Attack

We also apply the vanilla CW attack to audio recordings of every day noises such as construction sounds, cars honking, and leaves rustling. The aim of this experiment is two-fold: 1) Can the vanilla CW attack be applied to general sounds? 2) Can our defense detect attacks concealed in such audio recordings?

4.6. Universal Perturbation Attack

Finally, we study adversarial examples generated by a model based on universal adversarial perturbations [27]. A universal perturbation is a single perturbation which when added to different input audio samples causes a mistranscription by the ASR engine. Unlike the perturbations considered so far, universal perturbations are not targeted attacks, i.e., the transcription produced is not fixed. Moreover, in most cases, the transcription is not a meaningful sentence.

5. Proposed Defense

5.1. Dropout defense in the image domain

Feinman et al. [16] showed that dropout can be used to build an uncertainty estimator in neural networks for image classification. Specifically, dropout in neural networks mimics a deep Gaussian process and hence Bayesian estimates can be inferred. In their experiments, they subject an input image to I realizations of dropout during inference. The intuition is that the realizations obtained from an adversarial example will show more variation than those obtained from an original example. Let us denote by $y(x, \mathbf{W}) \in \mathbb{R}^C$ the output probability vector of an image classification network with parameters $\mathbf{W}$ for an input image x, where C denotes the number of classes. Each realization of dropout results in a new set of network parameters $\mathbf{W}^{(i)}$, $i = 1, \ldots, I$. The output for realization i is denoted as

$$y_i = y(x, \mathbf{W}^{(i)}). \tag{5}$$

The uncertainty $U(x)$ of the network with respect to input x is defined as the trace of the covariance matrix of the realizations, or equivalently as the average Euclidean distance between the realizations and their mean $\hat{y} = \frac{1}{I} \sum_{i=1}^{I} y_i$:

$$U(x) = \frac{1}{I} \sum_{t=1}^{I} \|y_i - \hat{y}\|^2. \tag{6}$$

A simple threshold-based classifier using the scalar $U(x)$ as input can now be designed to classify original and adversarial samples, as the uncertainty of adversarial samples is expected to be higher than that of original samples on average.

5.2. Extending the notion of uncertainty

Before we move to the audio domain, let us first introduce a generalization to the notion of uncertainty used in Feinman et al. [16], which will be useful later on. Instead of a single number, we would like to extract richer features for classification. We assume we have a set $\{y_i\}_{i=1}^I$ of I points in some space X, obtained as realizations of a neural network output with dropout. We also assume that we have some function d measuring a notion of distance between two points in X, as well as a mechanism to obtain a point $\hat{y}$ from the set $\{y_i\}_{i=1}^I$ encompassing some notion of average with respect to these points. In the image classification case above, the space X is the Euclidean space $\mathbb{R}^C$, the function d is the squared Euclidean distance, and $\hat{y}$ is obtained as the mean of the points in $\{y_i\}_{i=1}^I$. Based on these components, we can define the uncertainty distribution

$$\mathbb{P}(z) = \sum_i \mathbb{1}_{\{d(\hat{y},y_i)=z\}},\ z \in \mathbb{R}^+, \tag{7}$$

from which we can extract features to be used by a classifier. For instance, in the image case, the uncertainty $U(x)$ of Eq. (6) is none other than the second moment of $\mathbb{P}$.

5.3. Designing a notion of uncertainty for ASR

The defense devised by Feinman cannot be directly applied to the ASR case. Indeed, contrary to image classification where the network output is a vector of class probability predictions with fixed length, the corresponding output of an ASR system, in the case of CTC, is a sequence of such posterior probabilities, whose length depends on the input length. The problem is even more complex for decoder-based models, where the length of that sequence also depends on the internal processing of the network, as the decoder determines itself the output length.

A direct extension of Feinman's defense to the ASR case could be to consider the sequence of CTC posterior probabilities for an input x as a large vector used as the realization y_i, and to compute the uncertainty as in Eq. (6), normalizing by the input length. We consider this our baseline defense. We can further generalize this defense by considering the uncertainty distribution $\mathbb{P}_x^{\text{prob}}$ obtained using Eq. (7) in this context, and deriving features from it for a classifier.

For greater generalizability to various architectures and to reduce the dependence on the audio input length, we consider designing a defense based not on the sequence of CTC posterior probabilities but on the final output character sequence, which stems from all components of the network, including a potential language model. The final character sequence length for a given input may however vary depending on the internal processing of the network. Transcriptions for different dropout realizations may thus be of different lengths. Furthermore, while probability vectors can be considered within a Euclidean space, this is not possible for character sequences.

To define an uncertainty distribution following Section 5.2, we thus need to use a (non-Euclidean) distance metric d that can be calculated between character sequences with potentially different lengths. We use the Levenshtein distance, also known as the edit distance, as d. Because there is no notion of average in the non-Euclidean space X of character sequences with the edit distance, we use the medoid of the I different output transcriptions y_i as our "mean" $\hat{y}$. The medoid $\hat{y}$ of a set $\{y_i\}_{i=1}^I$ is defined as an element of the set whose average distance to all other elements for a distance d is the smallest:

$$\hat{y} = \underset{y \in \{y_1,\dots,y_I\}}{\operatorname{argmin}} \sum_i d(y, y_i). \tag{8}$$

Now that all required notions have been defined, we can define the uncertainty distribution $\mathbb{P}_x^{\text{char}}$ of an audio input x following Eq. (7), where $\{y_1, \dots, y_I\}$ are I character sequences output by the ASR engine for different dropout realizations. Note that, as the distances are integers, this distribution is a histogram.

5.4. Adversarial ASR Sample Classification

We can now classify an input audio sample x as adversarial or not by using a binary classifier taking as input some features derived from the distribution $\mathbb{P}_x^{\text{prob}}$ or $\mathbb{P}_x^{\text{char}}$. In our experiments, we consider the following classifiers: a decision stump trained on the second moment of the distribution (DS), simply comparing that moment to a threshold; a support vector machine (SVM) trained on the first four moments of the distribution (SVM-4); for $\mathbb{P}_x^{\text{char}}$, we also consider an SVM trained on the complete distribution (SVM-F), as we can obtain a fixed-length input vector by considering $(\mathbb{P}_x^{\text{char}}(0), \dots, \mathbb{P}_x^{\text{char}}(C))$, with C set to 19 (no distances larger than 18 were observed on our data); as SVM-F cannot be used for $\mathbb{P}_x^{\text{prob}}$, we replace it with a decision tree trained on the first four moments of the distribution (DecTree). We use a linear SVM in our experiments; other SVM variants did not provide better results. Note that the DS classifier for $\mathbb{P}_x^{\text{prob}}$ corresponds to the most direct extension of Feinman et al.'s defense to ASR, as mentioned in Section 5.3.

6. Experiments and Results

We implement our attacks and test our defenses on the Mozilla DeepSpeech [20] ASR engine. DeepSpeech is based on a bidirectional RNN network trained with CTC loss. The model's default dropout rate is $p_{\text{tr}} = 0.05$ during training. The adversarial samples are targeted to transcribe as "okay google unlock phone and delete files". In our defense, we use $I = 50$ realizations of dropout to compute the uncertainty of an audio sample. Results are reported in terms of defense accuracy in Table 1 and area under the ROC curve (AUC) in Table 2.

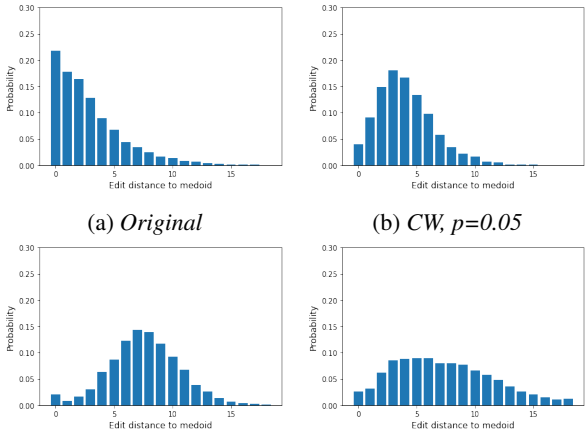
6.1. Choosing a defense dropout rate p

We first experimented with varying dropout rates to detect the CW attack samples. While there is little difference between the histograms of original and adversarial samples for $p \leq 0.04$, we can notice significant differences for $p \geq 0.05$. Figure 1a shows the mean uncertainty distribution $\mathbb{E}_x[\mathbb{P}_x^{\text{char}}]$ on all original training samples, and Fig. 1b that on all CW adversarial samples while using a defense dropout rate of 0.05 (CW $p = 0.05$).

As discussed in Section 4.2, an adversary may know the defense dropout rate and try training through dropout to break the defense. We observed that as the dropout rate p_{DR} used during adversarial training increases above the default dropout rate $p_{\text{tr}} = 0.05$, there is a sharp decrease in the *forgery success rate* of the attack, i.e., the adversarial sample did not get transcribed as the desired target sentence. An adversary attempting to fool our defense by training through different dropout rates will have a very low forgery success rate for p_{DR} above 0.05 and an attack will be nearly impossible for $p_{\text{DR}} = 0.1$. Hence, we use a dropout rate $p = 0.1$ in our defense for all future experiments, and $p_{\text{DR}} = 0.05$ in creating our dropout robust attacks.

6.2. Results on various attacks

We trained all classifiers for defense against the DR attack, as it is likely to be the most difficult to detect, and test those models on all attacks (except UrbanSound as it is a separate task). We use a 70-30 train-test split on 500 random samples from



(a) *Original* (b) *CW, p=0.05*

(c) *CW, p=0.1* (d) *DR, p=0.1*

Figure 1: *Mean uncertainty distribution* $\mathbb{E}_x[\mathbb{P}_x^{char}]$ *of distances to medoid for original audio samples and adversarial audio samples from different attacks, computed as the empirical mean of the distributions over all samples in the training set. The defense dropout rate is denoted by p.*

the CommonVoice dataset, where each of the original samples is used to generate corresponding adversarial samples for each attack. The average duration of each audio sample is about 5 s.

CW and DR attacks: The mean uncertainty distributions $\mathbb{E}_x[\mathbb{P}_x^{\text{char}}]$ for CW and DR with $p = 0.1$ are shown in Figs. 1c and 1d. We see in Tables 1 and 2 that using a defense dropout rate of 0.1 in conjunction with an SVM trained on the first four moments of the character-sequence-based uncertainty distribution ($\mathbb{P}_x^{\text{char}}$ - SVM-4) led to the best results on these attacks.

Noise reduction robust (NRR) attack: We observed that the NRR attacks were indeed robust to denoising techniques, and despite having been optimized through a spectral subtraction algorithm, our adversarial examples were also fairly robust to other noise reduction algorithms such as the `logmmse` algorithm [28]. From Table 1, we see that the accuracy results for $\mathbb{P}_x^{\text{char}}$ on NRR and DR are similar, which may be explained by the fact that they are both designed to be robust to perturbations.

Imperceptible audio attack (IA): The IA attack was originally implemented to work with attention-based models. We re-implemented it for the Mozilla DeepSpeech engine with a few modifications: the learning rate for the initial $\alpha = 0$ stage was decreased from 100 to 10, and the learning rate for the $\alpha > 0$ stages was decreased from 1 to 0.1; furthermore, the loss function used is CTC instead of cross-entropy loss. The IA attacks are audibly sharper and cleaner compared to the vanilla CW attack. Their behavior against our defense is however similar to CW in terms of accuracy. We also implemented the IA attack on samples from the LibriSpeech [29] dataset to investigate performance on longer utterances than the CommonVoice dataset. The extra length resulting in longer computation times to create adversarial examples, we only evaluated 20 samples. Our defense was able to detect all samples without error.

Urban sound attack: We were able to successfully apply the vanilla CW attack to the UrbanSound (US) [30] dataset. Unlike the previous results, the mean distribution of the original samples did not resemble Fig. 1a as the input was no longer speech, but the mean distribution of the adversarial samples did resemble Fig. 1d despite not being trained through dropout. The classifier was trained on a similar dataset as that used to train the DR defense, but based on data from UrbanSound instead of

Table 1: *Detection accuracy [%] on various attacks for the different classifiers. p denotes the defense dropout rate.*

		$p = 0.05$	$p = 0.1$				
		CW	CW	DR	NRR	IA	US
$\mathbb{P}_x^{\text{prob}}$	DS	71.7	83.3	82.5	75.5	91.0	90.4
	SVM-4	66.7	80.8	68.0	53.3	68.0	64.4
	DecTree	65.0	80.8	72.0	70.0	73.3	91.8
$\mathbb{P}_x^{\text{char}}$	DS	72.3	**96.5**	81.0	81.0	**92.0**	79.0
	SVM-4	76.7	**96.5**	**88.5**	**88.5**	**92.0**	**93.9**
	SVM-F	74.0	85.8	86.5	87.5	88.3	83.0

Table 2: *AUC score on various attacks for the different classifiers. p denotes the defense dropout rate.*

		$p = 0.05$	$p = 0.1$				
		CW	CW	DR	NRR	IA	US
$\mathbb{P}_x^{\text{prob}}$	DS	0.72	0.85	0.83	0.84	0.82	0.91
	SVM-4	0.84	0.91	0.88	0.89	0.90	**0.98**
	DecTree	0.72	0.85	0.83	0.84	0.82	0.91
$\mathbb{P}_x^{\text{char}}$	DS	0.72	0.82	0.81	0.82	0.73	0.86
	SVM-4	**0.88**	**0.92**	**0.95**	**0.93**	**0.95**	0.94
	SVM-F	0.75	0.91	0.92	0.93	0.94	0.74

CommonVoice. The results are shown in Tables 1 and 2.

Universal Perturbation: The universally perturbed audio attacks proposed in [27] do not fall under our definition of an attack as the adversarial example often does not transcribe as a meaningful sentence and hence has no malicious nature. Nevertheless, our $\mathbb{P}_x^{\text{char}}$ - SVM-4 defense trained on the CW attack data was able to detect the adversarial examples on the authors' website with 100% accuracy.

Entropy as a measure of uncertainty: After this article was submitted, concurrent work exploring various options for detecting audio attacks was released on arXiv [31]. That work included a method based on dropout and the Feinman-like variance similar to our $\mathbb{P}_x^{\text{prob}}$ - DS method, and reported obtaining better results with entropy. In preliminary experiments, we found that our $\mathbb{P}_x^{\text{char}}$ - SVM-4 defense performed similarly to or better than an entropy-based method in terms of accuracy (e.g., 96.5 % vs 90.5 % on CW, 88.5 % vs 88.0 % on DR), and significantly better in terms of AUC (e.g., 0.92 vs 0.81 on CW, 0.95 vs 0.88 on DR). The entropy feature may also potentially be combined with the features derived using our method. A more thorough comparison will be the object of future work.

7. Conclusion

In this paper, we showed that it is possible to extend the vanilla CW attack to create adversarial examples robust to dropout and denoising, and that such attacks can also be embedded within everyday urban sounds. We developed a defense that can detect a wide range of attacks on ASR engines by leveraging the uncertainty introduced by dropout. Using simple classifiers, we can detect adversarial examples with high confidence. Particularly, training an SVM on the first four moments of the distributions of distances between character sequences realized with dropout and their medoid achieves the best results. Finally, our defense is able to detect adversarial examples obtained with frequency masking or with a model based on universal perturbations.

8. References

[1] B. Biggio, I. Corona, D. Maiorca, B. Nelson, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," in *Proc. European Conference on Machine Learning (ECML) and European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, Jan. 2013.

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

[3] A. Arnab, O. Miksik, and P. H. S. Torr, "On the robustness of semantic segmentation models to adversarial attacks," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017.

[4] S. H. Huang, N. Papernot, I. J. Goodfellow, Y. Duan, and P. Abbeel, "Adversarial attacks on neural network policies," *arXiv preprint arXiv:1702.02284*, 2017.

[5] R. Jia and P. Liang, "Adversarial examples for evaluating reading comprehension systems," in *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Sep. 2017.

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

[7] Y. Qin, N. Carlini, G. Cottrell, I. Goodfellow, and C. Raffel, "Imperceptible, robust, and targeted adversarial examples for automatic speech recognition," in *Proc. International Conference on Machine Learning (ICML)*, Jun. 2019, pp. 5231–5240.

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

[9] L. Schönherr, S. Zeiler, T. Holz, and D. Kolossa, "Robust over-the-air adversarial examples against automatic speech recognition systems," *arXiv preprint arXiv:1908.01551*, 2019.

[10] Y. Lin and W. Abdulla, "Principles of psychoacoustics," in *Audio Watermark*, 2015, pp. 15–49.

[11] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu, "DolphinAttack: Inaudible voice commands," *arXiv preprint arXiv:1708.09537*, 2017.

[12] M. Cissé, Y. Adi, N. Neverova, and J. Keshet, "Houdini: Fooling deep structured prediction models," *arXiv preprint arXiv:1707.05373*, 2017.

[13] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *IEEE Symposium on Security and Privacy*, May 2017.

[14] F. Tramèr, A. Kurakin, N. Papernot, I. J. Goodfellow, D. Boneh, and P. D. McDaniel, "Ensemble adversarial training: Attacks and defenses," *arXiv preprint arXiv:1705.07204*, 2017.

[15] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[16] R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner, "Detecting adversarial samples from artifacts," *arXiv preprint arXiv:1703.00410*, 2017.

[17] A. Vyas, P. Dighe, S. Tong, and H. Bourlard, "Analyzing uncertainties in speech recognition using dropout," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 6730–6734.

[18] R. Prabhavalkar, K. Rao, T. N. Sainath, B. Li, L. Johnson, and N. Jaitly, "A comparison of sequence-to-sequence models for speech recognition," in *Proc. ISCA Interspeech*, Aug. 2017.

[19] A. Graves, S. Fernández, F. J. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proc. International Conference on Machine Learning (ICML)*, Jun. 2006.

[20] A. Y. 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," *arXiv preprint arXiv:1412.5567*, 2014.

[21] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2016.

[22] S. Kim, T. Hori, and S. Watanabe, "Joint CTC-attention based end-to-end speech recognition using multi-task learning," *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2016.

[23] A. Graves, A. rahman Mohamed, and G. E. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2013.

[24] S. Zhou, L. Dong, S. Xu, and B. Xu, "Syllable-based sequence-to-sequence speech recognition with the transformer in mandarin chinese," in *Proc. ISCA Interspeech*, Sep. 2018.

[25] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang, S. Watanabe, T. Yoshimura, and W. Zhang, "A comparative study on transformer vs rnn in speech applications," in *Proc. IEEE ASRU*, 2019.

[26] S. F. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Trans. Acoustics, Speech, Signal Process.*, no. 2, Apr. 1979.

[27] P. Neekhara, S. Hussain, P. Pandey, S. Dubnov, J. J. McAuley, and F. Koushanfar, "Universal adversarial perturbations for speech recognition systems," *arXiv preprint arXiv:1905.03828*, 2019.

[28] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator," *IEEE Trans. Acoustics, Speech, Signal Process.*, vol. 33, no. 2, pp. 443–445, Apr. 1985.

[29] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "LibriSpeech: An ASR corpus based on public domain audio books," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2015.

[30] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proc. ACM International Conference on Multimedia (ACM-MM)*, Nov. 2014.

[31] S. Däubener, L. Schönherr, A. Fischer, and D. Kolossa, "Detecting adversarial examples for speech recognition via uncertainty quantification," *arXiv preprint arXiv:2005.14611*, 2020.