# ADAGIO: Interactive Experimentation with Adversarial Attack and Defense for Audio

Nilaksh Das[1(✉)], Madhuri Shanbhogue[1], Shang-Tse Chen[1], Li Chen[2], Michael E. Kounavis[2], and Duen Horng Chau[1]

[1] Georgia Institute of Technology, Atlanta, GA, USA
{nilakshdas,madhuri.shanbhogue,schen351,polo}@gatech.edu
[2] Intel Corporation, Hillsboro, OR, USA
{li.chen,michael.e.kounavis}@intel.com

**Abstract.** Adversarial machine learning research has recently demonstrated the feasibility to confuse automatic speech recognition (ASR) models by introducing acoustically imperceptible perturbations to audio samples. To help researchers and practitioners gain better understanding of the impact of such attacks, and to provide them with tools to help them more easily evaluate and craft strong defenses for their models, we present Adagio, the first tool designed to allow interactive experimentation with adversarial attacks and defenses on an ASR model in real time, both visually and aurally. Adagio incorporates AMR and MP3 audio compression techniques as defenses, which users can interactively apply to attacked audio samples. We show that these techniques, which are based on psychoacoustic principles, effectively eliminate targeted attacks, reducing the attack success rate from 92.5% to 0%. We will demonstrate Adagio and invite the audience to try it on the Mozilla Common Voice dataset. Code related to this paper is available at: https://github.com/nilakshdas/ADAGIO.

**Keywords:** Adversarial ML · Security · Speech recognition

## 1 Introduction

Deep neural networks (DNNs) are highly vulnerable to adversarial instances in the image domain [3]. Such instances are crafted by adding small imperceptible perturbations to benign instances to confuse the model into making wrong predictions. Recent work has shown that this vulnerability extends to the audio domain [1], undermining the robustness of state-of-the-art models that leverage DNNs for the task of automatic speech recognition (ASR). The attack manipulates an audio sample by carefully introducing faint "noise" in the background that humans easily dismiss. Such perturbation causes the ASR model to transcribe the manipulated audio sample as a target phrase of the attacker's choosing. Through this research demonstration, we make two major contributions:


U. Brefeld et al. (Eds.): ECML PKDD 2018, LNAI 11053, pp. 677–681, 2019.
https://doi.org/10.1007/978-3-030-10997-4_50

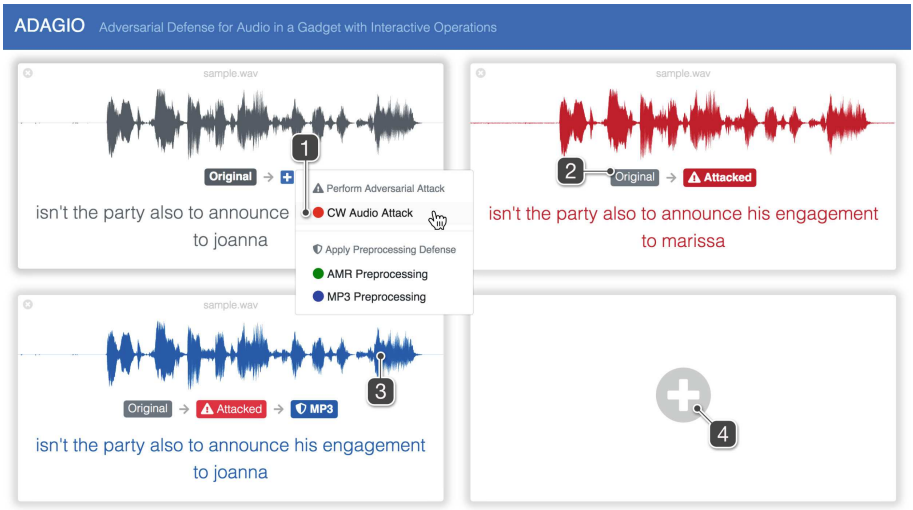


**Fig. 1.** Adagio usage scenario. (1) Jane uploads an audio file that is transcribed by DeepSpeech [5]; then she performs an adversarial attack on the audio in real time by entering a target transcription after selecting the attack option from the dropdown menu, e.g., the state-of-the-art Carlini-Wagner Audio Attack [1]. (2) Jane decides to perturb the audio to change the last word of the sentence from "joanna" to "marissa"; she can listen to the original audio and see the transcription by clicking on the "Original" badge. (3) Jane applies MP3 compression to recover the original, correct transcription from the manipulated audio; clicking on a waveform plays back the audio from the selected position. (4) Jane can experiment with multiple audio samples by adding more cards. For presentation, operations 1, 2 and 3 are shown as separate cards.

**1. Interactive exploration of audio attack and defense.** We present Adagio, the first tool designed to enable researchers and practitioners to interactively experiment with adversarial attack and defenses on an ASR model in real time (see demo: https://youtu.be/0W2BKMwSfVQ). Adagio incorporates AMR and MP3 audio compression techniques as defenses for mitigating perturbations introduced by the attack. Figure 1 presents a brief usage scenario showing how users can experiment with their own audio samples. Adagio stands for **A**dversarial **D**efense for **A**udio in a **G**adget with **I**nteractive **O**perations.

**2. Compression as an effective defense.** We demonstrate that non-adaptive adversarial perturbations are extremely fragile, and can be eliminated to a large extent by using audio processing techniques like Adaptive Multi-Rate (AMR) encoding and MP3 compression. We assume a non-adaptive threat model since an adaptive version of the attack is prohibitively slow and often does not converge.

## 2 ADAGIO: Experimenting with Audio Attack and Defense

We first provide a system overview of ADAGIO, then we describe its primary building blocks and functionality. ADAGIO consists of four major components: (1) an interactive UI (Fig. 1); (2) a speech recognition module; (3) a targeted attack generator module; and (4) an audio preprocessing (defense) module. The three latter components reside on a back-end server that performs the computation. The UI communicates the user intent with the back-end modules through a websocket messaging service, and uses HTTP to upload/download audio files for processing. When the messaging service receives an action to be performed from the front-end, it leverages a custom redis-based job queue to activate the correct back-end module. When the back-end module finishes its job, the server pings back the UI through the websocket messaging service to update the UI with the latest results. Below, we describe the other three components in ADAGIO.

### 2.1 Speech Recognition

In speech recognition, state-of-the-art systems leverage Recurrent Neural Networks (RNNs) to model audio input. The audio sample is broken up into frames $\{x^{(1)}, \ldots, x^{(T)}\}$ and fed sequentially to the RNN function $f(\cdot)$ which outputs another sequence $\{y^{(1)}, \ldots, y^{(T')}\}$, where each $y^{(t)}$ is a probability distribution over a set of characters. The RNN maintains a hidden state $h^{(t)}$ which is used to characterize the sequence up until the current input $x^{(t)}$, such that, $(y^{(t)}, h^{(t)}) = f(x^{(t-1)}, h^{(t-1)})$. The most likely sequence based on the output probability distributions then becomes the transcription for the audio input. The performance of speech-to-text models is commonly measured in Word Error Rate (WER), which corresponds to the minimum number of word edits required to change the transcription to the ground truth phrase.

ADAGIO uses Mozilla's implementation [5] of DeepSpeech [4], a state-of-the-art speech-to-text DNN model, to transcribe the audio in real time.

### 2.2 Targeted Audio Adversarial Attacks

Given a model function $m(\cdot)$ that transcribes an audio input $x$ as a sequence of characters $y$, i.e., $m(x) = y$, the objective of the targeted adversarial attack is to introduce a perturbation $\delta$ such that the transcription is now a specific sequence of characters $y'$ of the attacker's choosing, i.e., $m(x + \delta) = y'$. The attack is only considered successful if there is no error in the transcription.

ADAGIO allows users to compute adversarial samples using a state-of-the-art iterative attack [1]. After uploading an audio sample to ADAGIO, the user can click the attack button and enter the target transcription for the audio (see Fig. 1(1)). The system then runs 100 iterations of the attack and updates the transcription displayed on the screen at each step to show progress of the attack.

### 2.3 Compression as Defense

In the image domain, compression techniques based on psychovisual theory have been shown to mitigate adversarial perturbations of small magnitude [2]. We extend that hypothesis to the audio domain and let users experiment with AMR encoding and MP3 compression on adversarially manipulated audio samples. Since these techniques are based on psychoacoustic principles (AMR was specially developed to encode speech), we posit that these techniques could effectively remove the adversarial components from the audio which are imperceptible to humans, but would confuse the model.

To determine the efficacy of these compression techniques in defending the ASR model, we created targeted adversarial instances from the first 100 test samples of the Mozilla Common Voice dataset using the attack as described in [1]. We constructed five adversarial audio instances for every sample, each transcribing to a phrase randomly picked from the dataset, yielding a total of 500 adversarial samples. We then preprocessed these samples before feeding it to the DeepSpeech model. Table 1 shows the results from this experiment. We see that the preprocessing defenses are able to completely eliminate the targeted success rate of the attack.

**Table 1.** Word Error Rate (WER) and the targeted attack success rate on the DeepSpeech model (lower is better for both). AMR and MP3 eliminate all targeted attacks, and significantly improves WER.

| Defense | WER (no attack) | WER (with attack) | Targeted attack success rate |
|---|---|---|---|
| None | 0.369 | 1.287 | 92.45% |
| AMR | 0.488 | 0.666 | **0.00%** |
| MP3 | 0.400 | 0.780 | **0.00%** |

## 3 Conclusion

We present Adagio, an interactive tool that empowers users to experiment with adversarial audio attacks and defenses. We will demonstrate and highlight Adagio's features using a few usage scenarios on the Mozilla Common Voice dataset, and invite our audience to try out Adagio and freely experiment with their own queries.

## References

1. Carlini, N., Wagner, D.: Audio adversarial examples: targeted attacks on speech-to-text. arXiv:1801.01944 (2018)
2. Das, N., et al.: Keeping the bad guys out: protecting and vaccinating deep learning with jpeg compression. arXiv:1705.02900 (2017)

3. Goodfellow, I., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: ICLR (2015)
4. Hannun, A., et al.: Deep speech: scaling up end-to-end speech recognition. arXiv:1412.5567 (2014)
5. Mozilla: Deepspeech. https://github.com/mozilla/DeepSpeech