Attacking Speaker Recognition with Deep Generative Models

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

In this paper we investigate the ability of generative adversarial networks (GANs) to synthesize spoofing attacks on modern speaker recognition systems. We first show that the modern architectures of SampleRNN and WaveNet are unable to fool CNN-based speaker recognition systems. We propose a modification of the Wasserstein GAN objective function to make use of data that is real but not from the class being learned. Our method is able to perform both targeted and untargeted attacks against state of the art systems, which calls attention to issues related with security.

1 Introduction

Speaker authentication systems are increasingly being deployed for security critical applications in industries like banking, forensics, and home automation. Like other domains, such industries have benefited from recent advancements in deep learning that lead to improved accuracy and trainability of the speech authentication systems. Despite the improvement in the efficiency of these systems, evidence shows that they can be susceptible to adversarial attacks[23], thus motivating a current focus on designing AI-based systems that are provably correct with respect to mathematically-specified requirements [18], including understanding adversarial attacks ([20], [6]) and finding countermeasures to detect and deflect them.

Parallel to advancements in speech authentication, neural speech *generation* (the process of using deep neural networks to generate speech) has also seen huge progress in recent years ([22], [1]). The combination of these advancements begs a natural question that has, to the best of our knowledge, not yet been answered:

Are state-of-the-art speech authentication systems robust
to adversarial attacks by speech generative models?

Generative Adversarial Networks (GANs) have recently been found to produce incredibly authentic samples in a variety of fields. The core idea of GANs, a minimax game played between a generator network and a discriminator network, extends naturally to the field of speaker authentication and spoofing. We show that a variant of GAN training motivates the model's use as an attacking architecture.

With regards to this question, we offer in this paper the following contributions:

- We evaluate SampleRNN and WaveNet in their ability to fool text-independent state-of-the-art speaker recognizers.
- We propose strategies for untargeted attacks using Generative Adversarial Networks.

Submitted to 31st Conference on Neural Information Processing Systems (NIPS 2017). Do not distribute.

- We propose strategies for targeted attacks using a new objective function based on the improved Wasserstein GAN.

2 Related work

Modern generative models are sophisticated enough to produce fake[1] speech samples that can be indistinguishable from real human speech. Here, we provide a summary of some existing neural speech synthesis models and their architectures. WaveNet [21] is a generative neural network that is trained end-to-end to model quantized audio waveforms. The model is fully probabilistic and autoregressive, using a stack of causal convolutional layers to condition the predictive distribution for each audio sample on all previous ones. It has produced impressive results for generation of speech audio conditioned on speaker and text and has become a standard baseline for neural speech generative models.

SampleRNN [13] is another autoregressive architecture that has been successfully used to generate both speech and music samples. SampleRNN uses a hierarchical structure of deep RNNs to model dependencies in the sample sequence. Each deep RNN operates at a different temporal resolution so as to model both long term and short term dependencies.

Recent work on deep learning architectures has also introduced the presence of *adversarial examples*: small perturbations to the original inputs, normally imperceptible to humans, which nevertheless cause the architecture to generate an incorrect or deliberately chosen output. In their brilliant papers, [20] and [6] analyze the origin of adversarial attacks and describe simple and very efficient techniques for creating such perturbations, such as the fast gradient sign method (FGSM).

In the vision domain, [19] describe a technique for attacking facial recognition systems. Their attacks are physically realizable and inconspicuous, allowing an attacker to impersonate another individual. In the speech domain, [3] describe attacks on speech-recognition systems which use sounds that are hard to recognize by humans but interpreted as specific commands by speech-recognition systems.

To the best of our knowledge, GANs have not been used for the purpose of speech synthesis[2]. [15] uses a conditional GAN for the purpose of speech *enhancement*, i.e. taking as input a raw speech signal and outputting a denoised waveform. The model in [4] tackles the reverse problem of using GANs to learn certain representations given a speech spectrogram.

3 Data

In this section we describe the datasets used and the data engineering pipeline, including pre-processing and feature extraction.

3.1 Datasets

In our experiments we use three speech datasets, as shown in Table1. The datasets used are public and provide audio clips of different lengths, quality, language and content. In addition to the samples listed in Table1, we used globally conditioned sampleRNN and WaveNet fake samples available on the web. The fake samples are from the Blizzard dataset and CSTR VCTK (P280) respectively.

	Speakers	Language	Duration	Context
2013 Blizzard	1	English	73 h	Book narration
CSTR VCTK	109	English	400 Sentences	Newspaper narration
2004 NIST	100	Multiple	5 min / speaker	Conversational phone speech.

Table 1: Description of the datasets used in our experiments.

[1]We use the term fake to refer to computer generated samples

[2]More specifically, Mel-Spectrogram synthesis

3.2 Pre-processing

Data pre-processing is dependent on the model being trained. For SampleRNN and WaveNet, the raw audio is reduced to 16kHz and quantized using the μ-law companding transformation as referenced in [13] and [21]. For the model based on the Wasserstein GAN, we pre-process the data by converting it to 16kHz and removing silences by using the WebRTC Voice Activity Detector (VAD) as referenced in [24]. For the CNN speaker recognition system, the data is pre-processed by resampling to 16kHz when necessary and removing silences by using the aforemetioned VAD.

3.3 Feature extraction

SampleRNN and WaveNet operate at the sample level, i.e. waveform, thus requiring no feature extraction. The features used for the neural speaker recognition system are based on Mel-Spectrograms with dynamic range compression. The Mel-Spectrogram is obtained by projecting a spectrogram onto a mel scale. We use the python library librosa [12] to project the spectrogram onto 64 mel bands, with window size equal to 1024 samples and hop size equal to 160 samples, i.e. 100ms long frames. Dynamic range compression is computed as described in [11], with $log(1 + C * M)$, where C is a compression constant scalar set to 1000 and M is a matrix representing the Mel-Spectrogram. Training the GAN is also done with Mel-Spectrograms of 64 bands and 64 frames image patch.

4 Attacking speaker recognition models

In this section, we define our neural speaker recognition system and define the targeted and untargeted adversarial attacks we investigate.

4.1 Neural speaker recognition system

The speaker recognition system used in our experiments is based on the state-of-the-art framework by [11] and is described in Figure 1. The first module at the bottom is a pre-processing step that extracts the Mel-Spectrogram from the waveform as described in section 3.2. The second module is a convolutional neural network (CNN) that performs multi-speaker classification using the Mel-Spectrogram. The CNN is a modified version of Alexnet [9]. We warn the readers that unlike 1, our classifier operates on 64 by 64 Mel -Spectrogram and has slightly different number of nodes on each layer.

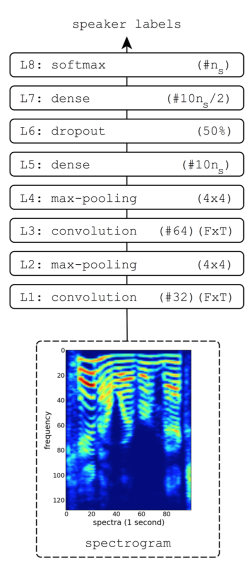


Figure 1: Architecture for CNN speaker verifier

We train the CNN on our training set using 64 by 64 Mel-Spectrograms [3] consisting of balanced samples from 102 speakers from the NIST 2004, Blizzard, and VCTK(P280) datasets. Our model achieves 85% test set accuracy.

4.2 Adversarial attacks

We define adversarial attacks on speaker recognition systems as *targeted* or *untargeted*. In targeted attacks, an adversary is interested in designing an input that makes the classification system predict a target class chosen by the adversary. In untargeted attacks, the adversary is interested in a confident prediction, regardless of the class being predicted. Untargeted attacks are essentially designed to fool the classifier into thinking a fake speech sample is real. Notice that a successful targeted attack is by definition a successful untargeted attack as well.

5 Adapting Wasserstein GAN for Attacks

In this section, we describe our Generative Adversarial Network (GAN), and its usage as a speech recognition attacker.

5.1 Model

The GAN framework proposed by [5] involves training a *generator* network, which is trained to learn a function from noise to samples that approximate the real data distribution. Simultaneously, a *discriminator* network is trained to identify whether a sample came from the real distribution or not - i.e., it is trained to try to output 1 if a sample is real, and 0 if a sample is fake. The generator and discriminator can be arbitrary networks.

The GAN framework has been shown to be able to produce very realistic samples with low training overhead. However, since the generator is trained to minimize the Kullback-Leibler (KL) divergence between its constructed distribution and the real one, it suffers from an exploding loss term when the real distribution's support is not contained in the constructed one. To counter this, the *Wasserstein GAN* [2] (WGAN) framework instead uses the Wasserstein (Earth-Mover) distance between distributions instead, which in many cases does not suffer from the same explosion of loss and gradient. In the WGAN framework, the loss functions of the generator and *critic* (which no longer emits a simple probability, but rather an approximation of the Wasserstein distance between the fake distribution and real) become:

$$L_G = -\underset{\widetilde{\boldsymbol{x}}\sim\mathbb{P}_g}{\mathbb{E}}\big[D(\widetilde{\boldsymbol{x}})\big] \tag{1}$$

$$L_C = \underset{\widetilde{\boldsymbol{x}}\sim\mathbb{P}_g}{\mathbb{E}}\big[D(\widetilde{\boldsymbol{x}})\big] - \underset{\boldsymbol{x}\sim\mathbb{P}_r}{\mathbb{E}}\big[D(\boldsymbol{x})\big] \tag{2}$$

where P_r is the real distribution and P_g the learnt distribution of the generator.
The original WGAN framework uses weight clipping to ensure that the critic satisfies a Lipschitz condition. As pointed by [7], however, this clipping can lead to problems with gradient stability. Instead, [7] suggest adding a gradient penalty to the critic's loss function, which indirectly tries to constrain the original critic's gradient to have a norm close to 1. Equation (2) thus becomes (taken from [7]):

$$L_C = \underbrace{\underset{\widetilde{\boldsymbol{x}}\sim\mathbb{P}_g}{\mathbb{E}}\big[D(\widetilde{\boldsymbol{x}})\big] - \underset{\boldsymbol{x}\sim\mathbb{P}_r}{\mathbb{E}}\big[D(\boldsymbol{x})\big]}_{\text{Original critic loss}} + \underbrace{\lambda \underset{\hat{\boldsymbol{x}}\sim\mathbb{P}_{\hat{x}}}{\mathbb{E}}\big[(\|\nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}})\|_2 - 1)^2\big]}_{\text{Gradient Penalty}} \tag{3}$$

In all of our experiments, we use the Wasserstein GAN with this gradient penalty (WGAN-GP), which we found makes the model converge better than regular WGAN or GAN. We will henceforth use WGAN, IWAGN, GAN, and WGAN-GP interchangeably to refer to WGAN-GP.

5.2 Attacks

Performing *untargeted* attacks with the WGAN-GP (i.e., training the network to output speech samples that mimic the distribution of speech) is relatively straightforward - we simply train the

[3] 64 mel bands and 64 frames, 100 ms each

WGAN-GP using all speakers in our dataset. However, the most natural attack is one that is *targeted*: where the GAN is trained to directly fool a speaker recognition system, i.e., to produce samples that the system classifies as matching a target speaker with reasonable confidence.

A naive approach for targeted attacks is to train the GAN on the data of the single target speaker. A drawback of this approach is that *discriminator*, and by consequence the *generator*, does not have access to universal properties of speech[4]. To circumvent this problem, we propose a modification to the critic's objective function that allows it to learn to differentiate between not only real samples and generated samples, but also between real speech samples from a target speaker and real speech samples from other speakers. We do this by adding a term to the critic's loss that encourages its discriminator to classify real speech samples from untargeted speakers as fake. From (3), the critic's loss L_C changes to:

$$\underbrace{\underset{\tilde{\boldsymbol{x}}\sim\mathbb{P}_g}{\mathbb{E}}\big[D(\tilde{\boldsymbol{x}})\big]}_{\text{Generated Samples}} \underbrace{+\alpha * \underset{\dot{\boldsymbol{x}}\sim\mathbb{P}_{\dot{x}}}{\mathbb{E}}\big[D(\dot{\boldsymbol{x}})\big]}_{\text{Different Speakers}} - \underbrace{\underset{\boldsymbol{x}\sim\mathbb{P}_r}{\mathbb{E}}\big[D(\boldsymbol{x})\big]}_{\text{Real Speaker}} + \underbrace{\lambda \underset{\hat{\boldsymbol{x}}\sim\mathbb{P}_{\hat{x}}}{\mathbb{E}}\big[(\|\nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}})\|_2 - 1)^2\big]}_{\text{Gradient Penalty}} \tag{4}$$

where $P_{\dot{x}}$ is the distribution of samples from other speakers and α is a tunable scaling factor. Note that (4) is no longer a direct approximation of the Wasserstein distance. Rather, it provides a balance of the distance between both the fake distribution and real one, and the distance between other speakers' distribution and the target speaker's one.

6 Experimental setup

6.1 WGAN setup

In our experiments, we trained a WGAN-GP to produce mel-spectrograms from 1 target speaker, against a set of over 101 "other" speakers. On each critic iteration, we fed it with a batch of samples from one target speaker, and a batch of data uniformly sampled from the other speakers.

We used two popular architectures for generator/critic pairs:

- *DCGAN* [17] models the generator as a series of deconvolutional layers with ReLU activations, and the discriminator as a series of convolutional ones with leaky ReLU activations. Both architectures use batch normalization after each layer.
- *ResNet* [10] models the generator and discriminator each as very deep convnets (30 layers in our experiments) with upsampling/downsampling respectively. Residual (skip) connections are added every few layers to make training easier.

Initially, we were able to converge the targeted loss model used the same parameters as [7], namely 5 critic iterations per generator iteration, a gradient penalty weight of 10, and batch size of 64. Both the generator and critic were trained using the Adam optimizer [8]. However, under these parameters we found that the highest α weight we could successfully use was 0.1 (we found that not including this scaling factor led to serious overfitting and poor convergence of the GAN).

In order to train a model with α set to 1, we made several modifications to the setup, including changing the standard deviation of the DCGAN discriminator's weight initialization to 0.05 and iterations to 20. To accommodate the critic's access to additional data in the mixed loss function (4), we increased the generator's learning rate to $1e^{-4}$, whereas the critic's learning rate was kept at $1e^{-5}$. Finally, we added of Gaussian noise to the target speaker data to prevent overfitting.

6.2 WaveNet

Due to constraints on computing power, we used samples from WaveNet models that had been pre-trained for 88 thousand iterations. Parameters of the models were kept the same as those in [21]. The ability of WaveNet to perform *untargeted* attacks amounts to using a model trained on an entire corpus. Targeted attacks are more difficult - we found that a single speaker's data was not enough to train WaveNet to converge successfully. To construct speaker-dependent samples, we relied on samples from pre-trained models that were *globally conditioned* on speaker ID. Auditorily, such samples do sound very similar to the real speech of the ID in question. We ran the feature-extraction in section 3 on these samples to produce data fed to the classifier.

[4]We draw a parallel with Universal Background Models in speech.

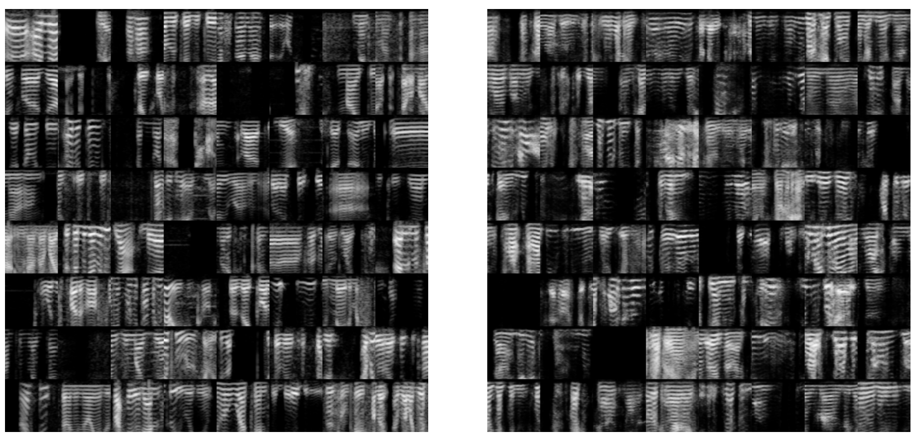
6.3 sampleRNN

Similarly to WaveNet, we found that the best (least noisy) sampleRNN samples came from models which were pretrained with a high number of iterations. Accordingly, we obtained samples from the three-tiered architecture, trained on the Blizzard 2013 dataset [16], which as mentioned in Section 3 is a 300 hour corpus of a single female speaker's narration. We also downloaded 10 second samples from the original paper's online repository at `https://soundcloud.com/samplernn/sets`, which we qualitatively found to have less noise than our generated ones.

7 Results

7.1 GAN Mel-Spectrogram

Using the improved Wasserstein GANs framework, we trained generators to construct 64x64 mel-spectrogram images from a noise vector. Visual results are demonstrated below in Figure 2. We saw recognizable Mel-Spectrogram-like features in the data after only 1000 generator iterations, and after 5000 iterations the generated samples were indistinguishable from real ones. Training took around 10 hours for 20000 iterations on a single 4 GB Nvidia GK104GL GPU.



(a) Real (actual) (b) Fake (generated)

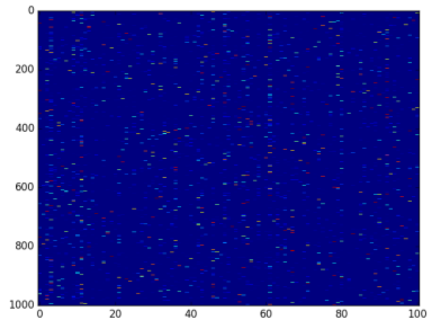
Figure 2: Comparison of real and generated ($\sim$ 5000 generator iterations) spectrogram samples from all speakers. Each grid contains 64 samples.

7.2 GAN Adversarial attacks

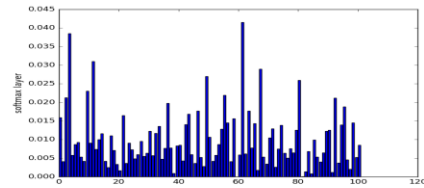
Within the GAN framework, we train models for untargeted attacks by using all data available from speakers that the speaker recognition systems was trained on, irrespective of class label. We show that an untargeted model able to generate data from the real distribution with enough variety can be used to perform adversarial attacks. We provide details in the untargeted attacks subsection 7.2.1. Figure 3b depicts that our GAN-trained generator successfully learns all speakers across the dataset, without mode collapsing.

As we described earlier, the models for targeted attacks can be trained in two manners: 1) conditioning the model on additional information, e.g. class labels, as described in [14]; 2) using only data from the label of interest. While the first approach might result in mode collapse, a drawback of the second approach is that the discriminator, and by consequence the generator, does not have access to universal[5] properties of speech. In the targeted attacks subsection 7.2.2 we show results using our a new objective function that allows using data from all speakers.

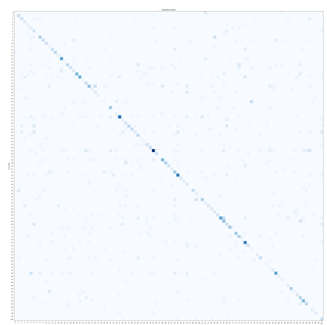
[5]We draw a parallel with Universal Background Models in speech.



(a) Our speaker classifier's softmax distribution of 1000 samples on approximately 100 speakers.



(b) Our speaker classifier's distribution of randomly sampled speech from the generative model.



(c) Confusion matrix of untargeted model. x-axis corresponds to predicted label, y-axis to ground truth.

Figure 3: Summary of untargeted attacks. Red represents high confidence.

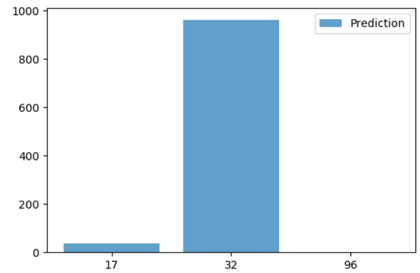
7.2.1 Untargeted attacks

For each speaker audio data in the test set, we compute a Mel-Spectrogram as descibred in section 3.2. The resulting Mel-Spectrogram is then fed into the CNN recognizer and we extract a 505-dimensional feature G from the penultimate fully-connected layer (L7) in the pre-trained CNN model (1) trained on the train partition of the real speech dataset with all speaker IDs. This deep feature/embedding G is then used to train a K-nearest-neighbor (KNN) classifier, with K equal to 5.

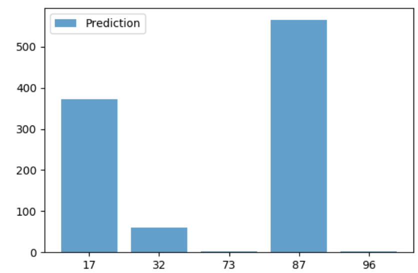
To control the generator trained by our WGAN, we feed the generated Mel-Spectrograms into the same CNN-L2 pipeline to extract their corresponding feature $\widehat{G}$. Utilizing the pre-trained KNN, each sample is assigned to the nearest speaker in the deep feature space. Therefore, we know which speaker our generated sample belongs to when we attack our CNN recognizer. We evaluate our controlled WGAN samples against the state-of-the-art CNN recognizer, and the confusion matrix can be found in Figure 3.

7.2.2 Targeted attacks

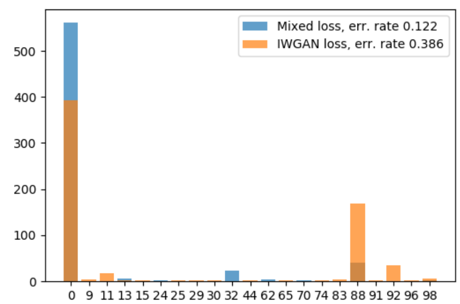
We ran all three models (WGAN-GP, SampleRNN, WaveNet) on a mixed corpus containing the entirety of the NIST 2004 corpus, a single speaker (280) from the VCTK Corpus, and the single speaker from the Blizzard dataset. The mixed corpus therefore contains 102 speakers. Samples were created from WaveNet globally conditioned on the single VCTK corpus speaker, and on SampleRNN trained only on data from the Blizzard dataset. Results are demonstrated in Figure 4. Neither WaveNet samples nor sampleRNN samples were able to attack the recognition model in the same way. In both models, **none** of the predictions made by the classifier match the target speaker.
We also trained the WGAN-GP with mixed loss/without mixed loss on speaker 0. The histogram of predictions in Figure 4c shows that using the mixed loss model, most of the energy is concentrated on the target speaker 0. The improved WGAN-GP loss achieves 0.38 error rate and our mixed



(a) Histogram of sampleRNN predictions. **Ground truth label: 100**.



(b) Histogram of WaveNet predictions. **Ground truth label: 101**.



(c) Histogram of predictions given improved WGAN and mixed loss models. **Ground truth label: 0.**

Figure 4: Summary histograms of targeted attacks

loss achieves 0.12 error rate, producing a 75% increase in accuracy. It is therefore clear that the WGAN-GP mixed loss framework is an improvement on the original loss function, which is expected given the network's access to additional speaker data.

8 Discussion and Conclusion

In this paper we have investigated the use of speech generative models to perform adversarial attacks on speaker recognition systems. We show that the autoregressive models we trained, i.e. SampleRNN and WaveNet, were not able to fool the CNN speaker recognizers we built. On the other hand, we show that adversarial examples generated with GAN networks are successful in performing targeted and untargeted adversarial attacks.

A natural question to ask is whether existing speech synthesis architectures like WaveNet and SampleRNN can be augmented with an adversarial-type loss in the same way as GANs. Both WaveNet and SampleRNN are trained to minimize the cross entropy loss between their generated samples and the real data. If one could attach a term to this loss function in the same way (e.g., maximizing the l2 distance between the generated samples and the data from other speakers, and tuning the weight of this distance to allow convergence), perhaps such a modification could be made. This modification would valuable as well when considering more sophisticated architectures like [22].

With this paper we hope to raise attention to issues that generative models bring to security and biometric systems. We foresee that samples produced with generative models have a signature that can be used to identify the source of the data and leave this investigation for future work.

References

[1] Sercan O Arik, Mike Chrzanowski, Adam Coates, Gregory Diamos, Andrew Gibiansky, Yongguo Kang, Xian Li, John Miller, Jonathan Raiman, Shubho Sengupta, et al. Deep voice: Real-time neural text-to-speech. *arXiv preprint arXiv:1702.07825*, 2017.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[3] Nicholas Carlini, Pratyush Mishra, Tavish Vaidya, Yuankai Zhang, Micah Sherr, Clay Shields, David Wagner, and Wenchao Zhou. Hidden voice commands. In *25th USENIX Security Symposium (USENIX Security 16), Austin, TX*, 2016.

[4] Jonathan Chang and Stefan Scherer. Learning representations of emotional speech with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1705.02394*, 2017.

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[6] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[7] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*, 2017.

[8] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[10] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint arXiv:1609.04802*, 2016.

[11] Yanick Lukic, Carlo Vogt, Oliver Dürr, and Thilo Stadelmann. Speaker identification and clustering using convolutional neural networks. In *Machine Learning for Signal Processing (MLSP), 2016 IEEE 26th International Workshop on*, pages 1–6. IEEE, 2016.

[12] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, 2015.

[13] Soroush Mehri, Kundan Kumar, Ishaan Gulrajani, Rithesh Kumar, Shubham Jain, Jose Sotelo, Aaron Courville, and Yoshua Bengio. Samplernn: An unconditional end-to-end neural audio generation model. *arXiv preprint arXiv:1612.07837*, 2016.

[14] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[15] Santiago Pascual, Antonio Bonafonte, and Joan Serrà. Segan: Speech enhancement generative adversarial network. *arXiv preprint arXiv:1703.09452*, 2017.

[16] Kishore Prahallad, Anandaswarup Vadapalli, Naresh Elluru, G Mantena, B Pulugundla, P Bhaskararao, HA Murthy, S King, V Karaiskos, and AW Black. The blizzard challenge 2013–indian language task. In *Blizzard Challenge Workshop*, volume 2013, 2013.

[17] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[18] Sanjit A Seshia, Dorsa Sadigh, and S Shankar Sastry. Towards verified artificial intelligence. *arXiv preprint arXiv:1606.08514*, 2016.

[19] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1528–1540. ACM, 2016.

[20] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[21] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *CoRR abs/1609.03499*, 2016.

[22] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al. Tacotron: A fully end-to-end text-to-speech synthesis model. *arXiv preprint arXiv:1703.10135*, 2017.

[23] Zhizheng Wu, Nicholas Evans, Tomi Kinnunen, Junichi Yamagishi, Federico Alegre, and Haizhou Li. Spoofing and countermeasures for speaker verification: a survey. *Speech Communication*, 66:130–153, 2015.

[24] Adham Zeidan, Armin Lehmann, and Ulrich Trick. Webrtc enabled multimedia conferencing and collaboration solution. In *WTC 2014; World Telecommunications Congress 2014; Proceedings of*, pages 1–6. VDE, 2014.