Attacking Speaker Recognition with Deep Generative Models

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

In this paper we investigate targeted and untargeted attacks on speaker recognition systems. We first investigate the efficiency of SampleRNN and Wavenet in fooling CNN speaker recognizers. We design untargeted and targetted attacks based on the GAN framework. We propose a modification of the WGAN objective function to make use of data that is real but not from the class being learned. Our method if efficient in performing targetted and untargeted attacks, thus raising attention to issues related with security.

1 Introduction

Speaker recognition and authentication systems are being deployed for security critical applications such as banking, forensics, home automation, etc. Like other domains, these systems benefit from recent advancements in deep learning that lead to improved accuracy and trainability of such systems. Despite the improvement in the efficiency of these systems, evidence shows that they can be suspectible to adversarial attacks[20], thus motivating a current trend focused understanding adversarial attacks ([17], [6]) and finding countermeasures to detect of deflect them.

Parallel to these advancements, neural speech *generation* (the process of using deep neural networks to generate human-sounding speech) has also seen huge progress ([19], [2]). Generative Adversarial Networks (GANs) have recently been found to produce incredibly authentic samples in a variety of fields. The core idea of GANs - namely the minimax game played between a generator network and a discriminator network - extends very naturally to the field of speaker authentication and spoofing. The combination of these advancements begs a natural question that has, to the best of our knowledge, not been answered:

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

More specifically, we contemplate this question and 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.
- We propose strategies for targeted attacks using a new objective function based on the improved Wasserstein GAN.

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

2 Related work

Generative models for speech can produce fake[1] samples that look very similar to real samples, leading humans to believe that the fake samples are real. WaveNet [18] is a generative neural network trained end-to-end to model quantized audio waveforms that has produced impressive results for generation of speech audio conditioned on speaker and text. 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;

SampleRNN [12], 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. Another impressive model is Adobe's VoCo [1]. Although VoCo's research is unpublished, Adobe its hability to *edit a recorded audio clip via text and make a audio clip of someone saying sentence that he never said!*. Training the model requires as litle as 20 minutes of speech.

Another interesting framework for generative models is the Generative Adversarial Networks (GAN) framework proposed by [5], in which a *generator* network 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 isn't contained in the constructed one. To counter this, the *Wasserstein GAN* [3] (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. Based on this, 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:

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

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

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 norm close to 1. Equation (2) thus becomes (taken from [7]):

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

In the context of deep learning architectures as the ones descriebd in this section, adversarial examples can make small perturbations to the original inputs, normally imperceptible to humans, to obtain an incorrect, target or untargeted, output from the neural network. In their brilliant papers, [17] and [6] analyze the origin of adversarial attacks and describe simple and very efficient techniques for adversarial attacks, such as the fast gradient sign method.

In the vision domain, [16] 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, [4] describe attacks on speech-recognition systems that use sounds that are hard to recognize by humans but interpreted as specific commands by speech-recognition systems.

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

3 Method

In this section we will describe the datasets used, the data engineering pipeline, including pre-processing and feature extraction, the speaker recognition models evaluated and the adversarial attacks investigated in this paper.

3.1 Datasets

In our experiments we use three datasets, each assigned to a model as described in Table1. The datasets used are public and provide audio clips of different lengths, quality, language and content.

	Speakers	Language	Duration	Context	Model
2013 Blizzard	1	English	73 h	Book narr.	SampleRNN
CSTR VCTK	109	English	400 Sentences	Newspaper ++	WaveNet
2004 NIST	100	Multiple	5 min / speaker	Conversational tel.	WGAN

Table 1: Description of datasets used in our experiments. Book narr. refers to book narrations. Newspaper ++ refers to newspapers and other documents. Conversational tel. refers to conversational telephone speech.

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 $\mu - law$ companding transformation as referenced in SampleRNN [12] and WaveNet [18]. 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 [21]. For the speaker recognition system, the data is pre-processed by converting it 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 is 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 [11] to project the spectrogram onto 64 mel bands, with window size equal to 1024 samples and hop size equal to 160 samples, i.e. frames of 100ms long. Dynamic range compression is computed as described in [10], with $log(1 + C * M)$, where C is the a compression constant scalar set to 1000 and M is the matrix representing the Mel-Spectrogram. The speaker recognition system operates on the Mel-Spectrogram with dynamic range compression as previously described.

3.4 Neural speaker recognition system

The speaker recognition system used in our experiments is based on the state-of-the-art framework by [10] and is described in Figure 1. The first module at the bottom is a pre-processing step that extracts 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 [8].

We train the CNN on our training set using 64*64 Mel-Spectrograms [2] consisting of balanced samples from 101 speakers from the NIST 2004 and Blizzard datasets. Our model achieves 85% test set accuracy.

[2] 64 mel bands and 64 frames, 100 ms each

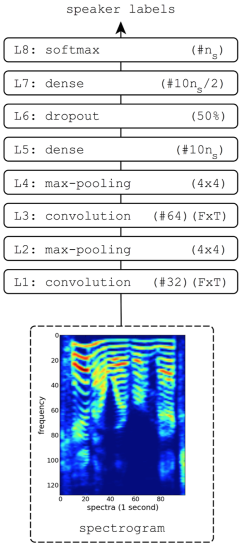


Figure 1: Architecture for CNN speaker verifier

4 Adversarial attacks

We define adversarial attacks on speaker recognition systems as targeted or untargeted. In targeted attacks, an adversary is interested in designing an adversarial input that makes the classification system predict a target class chosen by the adversary. In untargeted attacks, the adversary is interest in a confident prediction, regardless of the class being predicted. In the following section we describe our results using targeted and untargeted attacks using existing generative methods (WaveNet and sampleRNN) and using the GAN framework.

5 Results on existing methods

This section investigates the output of speaker recognition systems when features computed from fake data generated with WaveNet and sampleRNN is used as input. As described in 3.4, we train a speaker recognition systems with several speakers, including the speakers used to train WaveNet and sampleRNN. None of the WaveNet or sampleRNN samples produced with our model or downloaded from the authors website were successful in targeted or untargetted attacks to our speaker recognition system.

5.1 WaveNet

We attempted to replicate the model described in [18] but, unfortunately, we do not have access to Google's computing power nor to the North American Speech dataset Google used to train the WaveNet model that produced the samples referenced in [18]. Nonetheless, we used the data from CSTR VCTK to train speaker dependent WaveNet speech synthesis models that converged to producing speech that resembled the speakers voice but sounded like babbling.

5.2 sampleRNN

We also tested samples from sampleRNN [12], 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. We generated samples from the three-tiered variant, trained on the Blizzard 2013 dataset [14], 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.

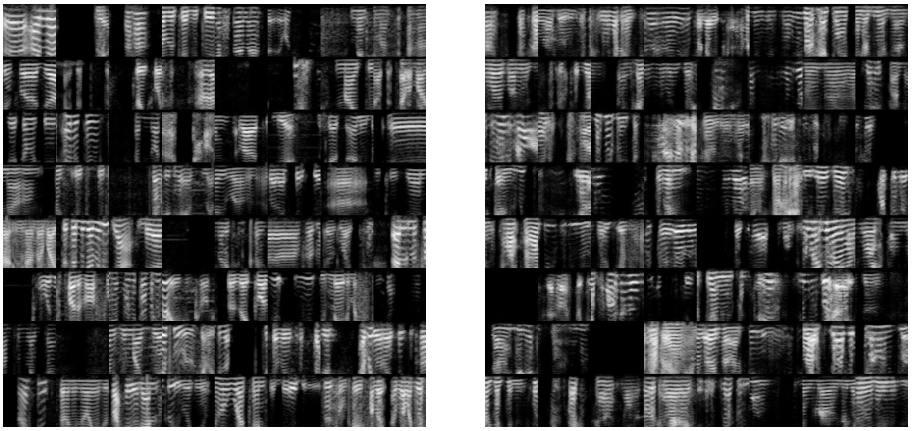
6 Results on adversarial methods

6.1 GAN Mel-Spectrogram

Using the improved Wasserstein GANs framework, we trained generators to construct 64x64 Mel-Spectrogram images from a noise vector. We used two popular architectures for generator/discriminator pairs:

- *DCGAN* [15] 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* [9] 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.

Visual results are demonstrated below in Figure 2. We generally [3] used the same parameters as [7], namely 5 critic iterations per generator iteration, a gradient penalty weight of 10, and batch size of 64. 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.

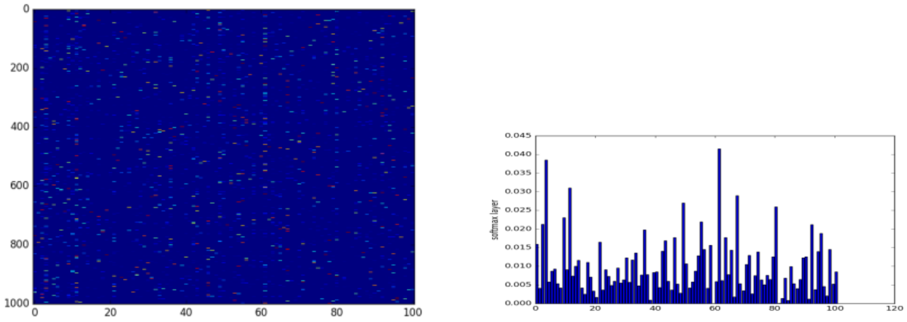
6.2 GAN Adversarial attacks

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 6.2.1. Figure 3b depicts that our GAN-trained generator successfully learns all speakers across the dataset, without mode collapsing.

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 [13]; 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[4]. properties of speech. In the targeted attacks subsection 6.2.2 we propose a new objective function that allows using the data from all speakers.

[3] We use different parameters and loss function for targeted attacks.

[4] 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.

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

6.2.1 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 4a. Although not included in the figure, **neither WaveNet samples nor SampleRNN samples were able to attack the recognition model in the same way.** [5]

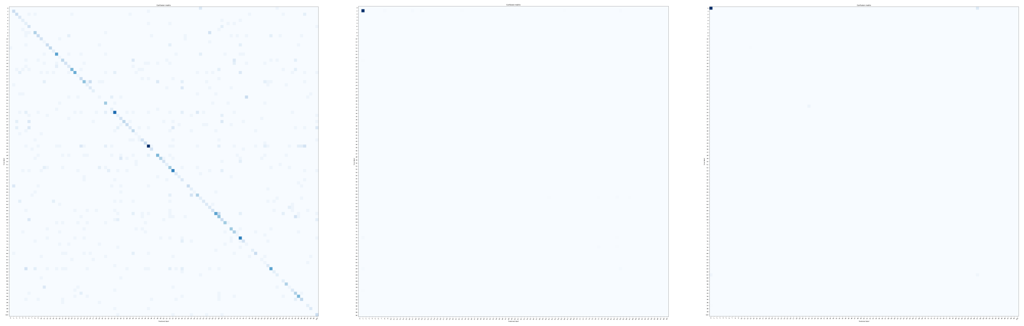
6.2.2 Targeted attacks

However, the most natural attack is one in which we train a GAN to directly fool a speaker recognition system, i.e., to produce samples that the system classifies as matching a target speaker with reasonable confidence. We attempted using data from the target speaker only, but the generated samples did not fool the speaker recognition systems. To circumvent this, 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. The critic's loss L_C becomes:

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

where $P_{\hat{x}}$ is the distribution of samples from other speakers, and α is a tunable scaling factor. In our experiments, we trained this GAN with 1 target speaker and a set of over 100 'other' speakers. On each critic iteration, we would feed it with a batch of samples from one target speaker, and a batch of data uniformly sampled from the other speakers. During training we used two approaches. The first one used an α of 0.1 (we found that not including this scaling factor led to serious overfitting and poor convergence of the GAN). The second one used an α of 1 combined with modifications to the network parameters, including different weight initialization, learning rates and addition of gaussian noise to the target speaker data. We invite readers to our github repo for details. Results are demonstrated in Figure 4: the confusion matrices in 4b and 4c show that all energy is concentrated in the targeted speaker ids, 1 and 0 respectively.

[5] Unlike our method, neither were they successful in attacking a GMM-UBM speaker recognizer.



(a) Confusion matrix of untargeted model

(b) Confusion matrix of targeted attacks with $\alpha = 0.1$

(c) Confusion matrix of targeted attacks with $\alpha = 1$

Figure 4: Confusion matrix of targeted attacks

7 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.

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.

Acknowledgments

References

[1] Adobe. Adobe voco, 2017.

[2] 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.

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

[4] 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.

[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] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

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

[14] 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.

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

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.