FOOLING END-TO-END SPEAKER VERIFICATION WITH ADVERSARIAL EXAMPLES

Felix Kreuk[1]*, Yossi Adi*[1]*, Moustapha Cisse*[2]*, Joseph Keshet*[1]

[1]Bar-Ilan University, Israel [2]Facebook AI Research

ABSTRACT

Automatic speaker verification systems are increasingly used as the primary means to authenticate costumers. Recently, it has been proposed to train speaker verification systems using end-to-end deep neural models. In this paper, we show that such systems are vulnerable to adversarial example attacks. Adversarial examples are generated by adding a peculiar noise to original speaker examples, in such a way that they are almost indistinguishable, by a human listener. Yet, the generated waveforms, which sound as speaker A can be used to fool such a system by claiming as if the waveforms were uttered by speaker B. We present white-box attacks on a deep end-to-end network that was either trained on YOHO or NTIMIT. We also present two black-box attacks. In the first one, we generate adversarial examples with a system trained on NTIMIT and perform the attack on a system that trained on YOHO. In the second one, we generate the adversarial examples with a system trained using Mel-spectrum features and perform the attack on a system trained using MFCCs. Our results show that one can significantly decrease the accuracy of a target system even when the adversarial examples are generated with different system potentially using different features.

Index Terms— Automatic speaker verification, adversarial examples

1. INTRODUCTION

Automatic speaker verification is the task of verifying that a spoken utterance has been produced by a claimed speaker. It is one of the most mature technologies for biometric authentication deployed by banks and e-commerce as the primary means to authenticate customers online and over the phone. The vulnerability of such a system is a real threat to these applications.

The standard verification protocol comprises the following three steps: training, enrollment, and evaluation. In the training stage, one learns a suitable internal speaker representation from a set of utterances and builds a simple scoring function. In the enrollment stage, a speaker provides a few utterances which are used to estimate the speaker model. During the evaluation stage, the verification task is performed by scoring a new unknown utterance against the speaker model. If the resulted score is greater than a pre-defined threshold, the system predicts that the unknown utterance produced by the claimed speaker. When the authentication is based on the voice of the speaker, irrespective of what the speaker said, the system is a text-independent speaker verification system.

Most of the modern speaker verification systems have several components. For example, the combination of i-vector for speaker representation and probabilistic linear discriminant analysis (PLDA) for a scoring function has become the dominant approach, both for text-dependent and text-independent speaker verification [1, 2, 3]. Recently, Heigold et al. [4] proposed to train speaker verification systems in an end-to-end fashion using deep neural models. This approach allows to directly learn from utterances, which improves capturing long-range context and reduces the complexity (one vs. number of frames evaluations per utterance), and the direct and joint estimation, which can lead to better and more compact models. Moreover, this approach often results in considerably simplified systems requiring fewer concepts and heuristics.

Although deep neural networks have enabled several breakthroughs in notoriously difficult problems such as image classification [5, 6], speech recognition [7, 8], speech processing [9, 10] and machine translation [11], it has been shown [12] that they are not robust to tiny perturbations in the input space. Indeed, adding a well-chosen small perturbation to the input of a network can change its prediction. When the difference between the perturbed image and the original image is indistinguishable by the human eye the example, the perturbed image is called an *adversarial example*.

Adversarial examples were first introduced in [13]. Their study first demonstrated that deep neural networks could achieve high accuracy on previously unseen examples while being vulnerable to small adversarial perturbations. This finding has recently aroused keen interest in the community [12, 14, 13, 15]. Several studies have subsequently analyzed the phenomenon [16, 17] and various approaches have been proposed to improve the robustness of neural networks [18, 19]. However, most of the previous works on adversarial examples are focused on the vision domain.

In this work, we investigate the generation of adversarial examples to attack an end-to-end neural based speaker verification model. We demonstrate the generation of adversarial examples to attack a text-dependent speaker verification system while using the architecture proposed in [4].

To the best of our knowledge, the only work that applied adversarial attacks to speech data is [20] where the authors present an adversarial attack on an end-to-end automatic speech recognition system. We are unaware of any previous study on adversarial examples for fool speaker verification systems. A different approach for attacking speaker verification systems is known as *spoofing attack* [21, 22]. In that type of attack, an adversary may use a falsifying voice, such as the recorded file of another person, as input for the speaker verification system. Our approach is different since our goal is to generate acoustic utterances which sound as speaker A (at least to the human ear) but can be used to fool the system by claiming that utterances were produced by speaker B.

This paper is organized as follows. In Section 2 we formally set the notation and definitions used throughout the paper. Section 3 provides a detailed description of the speaker verification model. In Section 4 we describe the mechanism behind the generation of adversarial examples. In Section 5 we report the results of attacking the speaker verification model in various ways. We conclude the paper with a discussion in Section 6.

2. NOTATIONS AND DEFINITIONS

In this section, we formulate the task of speaker verification rigorously and set the notation for the rest of the paper. We denote the domain of the acoustic feature vectors by $\mathcal{X} \subset \mathbb{R}^d$. The acoustic feature representation of a speech signal is therefore a sequence of vectors $\mathbf{x} = (x_1, x_2, ...x_T)$, where $x_i \in \mathcal{X}$ for all $1 \leq i \leq T$. The length of the input signal varies from one signal to another. Thus T is not fixed. We denote by $\mathcal{X}^*$ the set of all finite-length sequences over $\mathcal{X}$.

Recall that in speaker verification the goal is to assert if the claimed speaker spoke the input utterance. Specifically, the speaker verification system is a function that gets as input an utterance $\mathbf{x}$ produced by an unknown speaker, and a set of n enrollment utterances produced by speaker k denoted as $\mathbf{X}^k = \{\mathbf{x}_1^k, ..., \mathbf{x}_n^k\}$. The output of the system is a real number in the simplex $p \in [0, 1]$ estimating the probability that the utterance $\mathbf{x}$ produced by speaker k.

Let $g_\theta : \mathcal{X}^* \times (\mathcal{X}^*)^n \rightarrow [0, 1]$ be the speaker verification function implemented as neural network with a set of parameters θ. Given a set of training examples, the parameters θ are found by minimizing the negative log likelihood loss function. Each example in the training set is a tuple $(\mathbf{x}, \mathbf{X}^k, y)$ of a spoken utterance $\mathbf{x} \in \mathcal{X}$, an enrollment set of a speaker k, $\mathbf{X}^k$, and a binary label $y \in \{0, 1\}$ indicating whether the utterance $\mathbf{x}$ was produced speaker k. We denote by $\ell : [0, 1] \times \{0, 1\} \rightarrow \mathbb{R}$ the log-likehood loss function.

3. END-TO-END DEEP NETWORK MODEL

In this section, we describe the network architecture used as our speaker verification function. The architecture was initially proposed in [4], and serves as a baseline in recent work on end-to-end models for speaker verification [23, 24]. It is depicted in Figure 1.

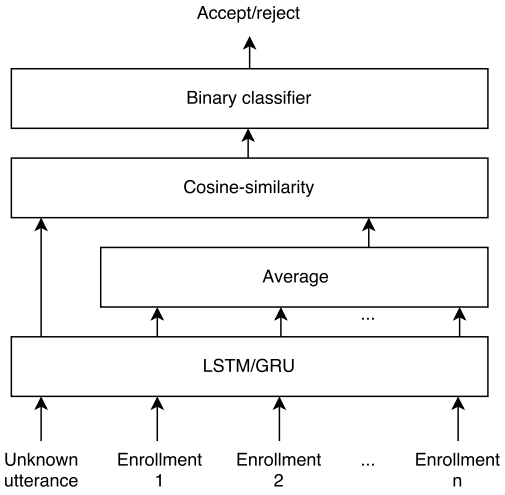

Fig. 1: End-to-end deep network model for speaker verification. The architecture is based on [4].

Recall that the input to the verification function is an unknown utterance and a set of n enrollment utterances. Each of the $n+1$ utterances is fed into a recurrent LSTM network and is represented as an embedding vector of size D. Denote by $\mathbf{u}$ and by $\mathbf{U}^k = (\mathbf{u}_1^k, \ldots, \mathbf{u}_n^k)$ the embeddings of the unknown utterance x and the enrollment set $\mathbf{X}^k$. The embeddings of the enrollments set are averaged to a single vector:

$$\mathbf{v}^k = \frac{1}{n} \sum_i^n \mathbf{u}_i^k.$$

Then the resemblance between the embedding of the unknown utterance and the average embedding of the enrollment is computed using the cosine-similarity function:

$$sim(\mathbf{u}, \mathbf{v}^k) = \frac{\mathbf{u} \cdot \mathbf{v}^k}{\|\mathbf{u}\| \|\mathbf{v}^k\|}.$$

The final stage takes the cosine-similarity, multiplies it by a scalar and add a bias to generate a probability estimation. One can think of this layer as an automatic setting of the threshold detection. The whole network is trained using with negative log-likelihood loss function.

4. GENERATING ADVERSARIAL EXAMPLES

Given an input utterance $\mathbf{x}$, an adversarial example is a perturbed version of the original pattern

$$\tilde{\mathbf{x}} = \mathbf{x} + \delta_{\mathbf{x}},$$

where $\delta_{\mathbf{x}} \in \mathbb{R}^d$ is small enough for $\tilde{\mathbf{x}}$ to be undistinguishable from $\mathbf{x}$ by a human, but causes the network to predict an incorrect label.

Formally, given a trained network g_θ and a p-norm, the adversarial example is generated by solving the following optimization problem:

$$\tilde{\mathbf{x}} = \underset{\tilde{\mathbf{x}}:\|\tilde{\mathbf{x}}-\mathbf{x}\|_p \leq \epsilon}{\operatorname{argmax}} \ell\big(g_\theta(\tilde{\mathbf{x}}, \mathbf{X}^k), y\big),$$

where ϵ represents the strength of the adversary, and p is the norm value. In words, we would like to maximize, rather than minimize, the loss function between the prediction of g_θ on the adversarial example and the correct label under the constraint that the adversarial example is similar to the original example in p-norm.

Assuming the loss function ℓ is differentiable, the authors of [16] proposed to take the first order Taylor expansion of $\mathbf{x} \mapsto \ell(g_\theta(\mathbf{x}, \mathbf{X}^k), y)$ to compute $\delta_\mathbf{x}$ by solving the following problem:

$$\tilde{\mathbf{x}} = \underset{\tilde{\mathbf{x}}:\|\tilde{\mathbf{x}}-\mathbf{x}\|_p \leq \epsilon}{\operatorname{argmax}} \big(\nabla_\mathbf{x} \ell(g_\theta(\mathbf{x}, \mathbf{X}^k), y)\big)^T (\tilde{\mathbf{x}} - \mathbf{x})$$

When $p = \infty$ the solution to the optimization problem is

$$\tilde{\mathbf{x}} = \mathbf{x} + \epsilon \operatorname{sign}(\nabla_\mathbf{x} \ell(g_\theta(\mathbf{x}, \mathbf{X}^k), y)),$$

which corresponds to the *fast gradient sign method* proposed in [12]. In other words, generating adversarial examples following the fast sign gradient method involves with taking the sign of the gradients of the loss function with respect to the input, multiply it by a small fraction so it will be indistinguishable to a human and add it to the original example.

Note that a single training example is composed of two parts; an enrollment set and a test utterance. To mimic a realistic model attack, we add the adversarial noise only to the test utterance and leave the enrollment set unchanged.

5. EXPERIMENTS

The setting was similar in all our experiments. We represented the speech utterances by acoustic features and trained a speaker verification model on that representation. Then we generated adversarial examples by adding noise to the feature vectors, and finally, acoustic waveforms were *reconstructed* from the adversarial example. For a reference, we also reconstruct waveforms of the original examples. The waveforms corresponding to the adversarial examples were used to fool the trained model as well as a different model that was trained under different conditions. We now describe the experimental setting in detail.

We evaluated the effectiveness of our method on two datasets: YOHO [25] and NTIMIT [26], each sampled at 8kHz. We used two sets of acoustic features. The first set of features was the Mel-spectrum. The acoustic signal was split into frames of 64 milliseconds with a shift of 4 milliseconds[1]. Then we applied Hamming window and computed STFT. We used a 65 Mel frequency channels that yielded $d = 65$ acoustic features. The second set we used was the Mel-Frequency Cepstrum Coefficients (MFCCs), extracted from the Mel-spectrum. Overall we trained four models; for each of the datasets (NTIMIT and YOHO), we used both feature sets.

For the YOHO corpus, each training example was generated by picking one of the verification utterances and associating it with a set of 10 random enrollment utterances. Ten speakers were excluded from the training set and used as a test set. For the NTIMIT corpus, since there are only ten utterances per speaker, we generate each of the training examples by picking one utterance to be the verification example and four other utterances to be an enrollment set. In both datasets, we swapped the enrollment set in half of the examples to generate negative instances.

We generated adversarial examples using the fast gradient sign method by adding adversarial perturbation to the test utterance vector $\mathbf{x}$ using several epsilon values. We found that $\epsilon \in (0.2, 0.3)$ caused the classifier to misclassify the adversarial examples with a high probability while keeping them remarkably similar to the original ones, a description of the adversarial examples evaluation process can be found in the next subsection. The models‘ performance was evaluated using the precision of correct classifications, and not using the standard equal error-rate (EER) since we aimed to show the effectiveness of adversarial attacks rather than comparing our model to other speaker verification systems.

5.1. ABX Testing

To validate that the generated adversarial examples are indeed indistinguishable by humans we performed an ABX test. An ABX test is a standard way to assess the detectable differences between two choices of sensory stimuli. We presented to listeners two audio samples A and B; each being either the original (reconstructed) waveform or an adversarial waveform of the same example. These two samples are followed by a third sound X which was randomly chosen to be either A or B. The listener was instructed to decide whether X is more similar to sample A or sample B. We randomly sampled 50 pairs of audio examples, original and adversarial ones. All waveforms were reconstructed from Mel-spectrogram using the Griffin-Lim algorithm [27]. eight different listeners tested each audio pair. Overall, on average 54% of the examples were correctly classified by the human listeners. Subsequently, we use such indistinguishable adversarial examples to test the robustness of speaker verification system.

5.2. A white-box attack

In the setting of a white-box attack, we assume that the adversary has access to the internals of the model to be attacked. In other words, the attacker has complete knowledge and control of the network and can access the networks‘ gradients.

[1]The high shift frequency allows us a better reconstruction of the signal.

Table 1: System accuracy under white-box attacks.

	YOHO		
	Original test	Adversarial test	Diff
Mel-spectrum	85.50%	37.50%	48.00%
MFCC	87.50%	25.75%	61.75%
	NTIMIT		
	Original test	Adversarial test	Diff
Mel-spectrum	84.26%	24.40%	59.86%
MFCC	82.14%	10.20%	69.94%

Table 2: False-positive rate (FPR) under white-box attacks.

	YOHO	
	Original test	Adversarial test
Mel-Spectrum	1.46%	69.76%
MFCC	4.88%	94.63%
	NTIMIT	
	Original test	Adversarial test
Mel-Spectrum	10.98%	79.19%
MFCC	1.73%	82.08%

An adversary can use these gradients to perturb the original input to become adversarial. The adversarial examples were crafted directly on the inputs that fed to the network.

Table 1 summarizes the results. The upper panel and the lower panel describes the results for the YOHO and the NTIMIT corpus, respectively. For each dataset, the accuracy on the test set is given in the first column. Even though one could achieve better results, our focus is to demonstrate the effectiveness of adversarial attacks on this model and to propose alternative ways (in addition to the traditional ones) of evaluating speaker verification systems. We assume the performance difference observed here is due to limited training data (YOHO has around 15.6 hours of speech, while the "OK Google" dataset contains 333 hours off speech [4]). In the second column, we presented the accuracy of the adversarial examples (generated from the test set examples). The last column is the degradation in performance.

In speech verification systems it might be considered more important to perform well regarding *false-positive rate* (FPR), this is the case where an adversary claims to be someone else and is wrongfully accepted. Results of FPR are given in Table 2, where we can see a significant degradation regarding FPR performance during the adversarial attack.

5.3. Cross-dataset

In the setting of black-box attack, the adversary has no access the model internals, only to its inputs and outputs. This setup is the strongest since it assumes no knowledge of the adversary regarding the type of model, its architecture or parameters. Moreover, the success of a black-box attack almost assures that a white-box will succeed equally or better. In our work, we performed two back-box attacks: a *cross-dataset* attack and a *cross-feature* attack. In the case of cross-dataset attach, the adversarial examples were crafted using a model trained on dataset A, is used to attack a model trained on dataset B.

We trained two models: model A was trained on the YOHO dataset using MFCC features, while model B was trained on the NTIMIT dataset using Mel-spectrum features. Then, model B was used to create adversarial examples on NTIMIT; these examples were used to attack model A. In other words, models A and B were trained on two different datasets. The examples created by model B were used to attack model A.

We found that model A reached an accuracy of 81.55% on NTIMIT reconstructed clean test set and 58.93% on NTIMIT reconstructed adversarial test set, a difference of 22.62%. The FPR was degraded from 12% to 46%.

5.4. Cross-features

A different type of back-box attack is done by creating adversarial examples using one set of acoustic features and attacking a model that was trained on a different set of acoustic features. More specifically, we trained a model on the YOHO corpus where the features were Mel-spectrum. We created adversarial examples and reconstructed waveforms. We then attacked a model that was trained on YOHO, but the features were MFCC.

We found that the MFCC model reached an accuracy of 81% on the reconstructed clean test set and accuracy of 62.25% on the reconstructed adversarial test set with a difference of 18.75%. The FPR degraded from 16% to 46%.

6. CONCLUSION

While deep neural networks have shown to improve the accuracy compared to the traditional speech verification components [28], it becomes critical to revisit the evaluation protocol of those models and design new ways to assess their reliability beyond the traditional metrics.

For future work, we would like to evaluate the robustness of the traditional speaker verification systems to adversarial examples as well as to apply adversarial training techniques to make the neural-based ones more robust to these type of attacks.

7. REFERENCES

[1] Patrick Kenny, "Bayesian speaker verification with heavy-tailed priors.," in *Odyssey*, 2010, p. 14.

[2] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on*

Audio, Speech, and Language Processing, 19.4, p 788–798, 2011.

[3] Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn, "Speaker verification using adapted gaussian mixture models," *Digital signal processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[4] Georg Heigold, Ignacio Moreno, Samy Bengio, and Noam Shazeer, "End-to-end text-dependent speaker verification," in *ICASSP*, 2016.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[7] Dario Amodei et al., "Deep speech 2: End-to-end speech recognition in english and mandarin," in *ICML*, 2016.

[8] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *ICML*, 2006.

[9] Yossi Adi, Joseph Keshet, and Matthew Goldrick, "Vowel duration measurement using deep neural networks," in *MLSP*. IEEE, 2015.

[10] Yossi Adi, Joseph Keshet, Emily Cibelli, and Matthew Goldrick, "Sequence segmentation using joint rnn and structured prediction models," in *ICASSP*, 2017.

[11] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint 1409.0473*, 2014.

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

[13] Christian Szegedy et al., "Intriguing properties of neural networks," *arXiv preprint 1312.6199*, 2013.

[14] Nicolas Papernot et al., "Practical black-box attacks against deep learning systems using adversarial examples," *arXiv preprint*, 2016.

[15] Pedro Tabacof and Eduardo Valle, "Exploring the space of adversarial images," in *IJCNN*. IEEE, 2016.

[16] Uri Shaham, Yutaro Yamada, and Sahand Negahban, "Understanding adversarial training: Increasing local stability of neural nets through robust optimization," *arXiv preprint 1511.05432*, 2015.

[17] Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard, "Robustness of classifiers: from adversarial to random noise," in *NIPS*, 2016.

[18] Nicolas Papernot et al., "Distillation as a defense to adversarial perturbations against deep neural networks," in *SP*. IEEE, 2016.

[19] Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier, "Parseval networks: Improving robustness to adversarial examples," in *ICML*, 2017.

[20] Moustapha M Cisse, Yossi Adi, Natalia Neverova, and Joseph Keshet, "Houdini: Fooling deep structured visual and speech recognition models with adversarial examples," in *NIPS*, 2017.

[21] Zhizheng Wu, Junichi othersYamagishi, Tomi Kinnunen, Cemal Hanilçi, Mohammed Sahidullah, Aleksandr Sizov, Nicholas Evans, and Massimiliano Todisco, "Asvspoof: the automatic speaker verification spoofing and countermeasures challenge," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 4, pp. 588–604, 2017.

[22] Abdenour Hadid, Nicholas Evans, Sébastien Marcel, and Julian Fierrez, "Biometrics systems under spoofing attack: an evaluation methodology and lessons learned," *IEEE Signal Processing Magazine, 32.5, p 20–30, 2015*.

[23] David Snyder, Pegah Ghahremani, Daniel Povey, Daniel Garcia-Romero, Yishay Carmiel, and Sanjeev Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *SLT*, 2016.

[24] Shi-Xiong Zhang, Zhuo Chen, Yong Zhao, Jinyu Li, and Yifan Gong, "End-to-end attention based text-dependent speaker verification," in *SLT*, 2016.

[25] Joseph P Campbell, "Testing with the yoho cd-rom voice verification corpus," in *ICASSP*, 1995.

[26] Charles Jankowski, Ashok Kalyanswamy, Sara Basson, and Judith Spitz, "Ntimit: A phonetically balanced, continuous speech, telephone bandwidth speech database," in *ICASSP*, 1990.

[27] Daniel Griffin and Jae Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[28] David Snyder, Daniel Garcia-Romero, Daniel Povey, and Sanjeev Khudanpur, "Deep neural network embeddings for text-independent speaker verification," *Proc. Interspeech*, 2017.