INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Representation Learning to Classify and Detect Adversarial Attacks against Speaker and Speech Recognition Systems

Jesús Villalba[1,2], *Sonal Joshi*[1], *Piotr Żelasko*[1,2], *Najim Dehak*[1,2]

[1]Center for Language and Speech Processing, Johns Hopkins University, USA
[2]Human Language Technology Center of Excellence, Johns Hopkins University, USA
{jvillal7,sjoshi12,pzelask2,ndehak3}@jhu.edu

Abstract

Adversarial attacks have become a major threat for machine learning applications. There is a growing interest in studying these attacks in the audio domain, e.g, speech and speaker recognition; and find defenses against them. In this work, we focus on using representation learning to classify/detect attacks w.r.t. the attack algorithm, threat model or signal-to-adversarial-noise ratio. We found that common attacks in the literature can be classified with accuracies as high as 90%. Also, representations trained to classify attacks against speaker identification can be used also to classify attacks against speaker verification and speech recognition. We also tested an attack verification task, where we need to decide whether two speech utterances contain the same attack. We observed that our models did not generalize well to attack algorithms not included in the attack representation model training. Motivated by this, we evaluated an unknown attack detection task. We were able to detect unknown attacks with equal error rates of about 19%, which is promising.

Index Terms: adversarial attacks, speaker recognition, speech recognition, x-vectors

1. Introduction

Adversarial attacks [1] have become a major threat for machine learning systems including those based on audio such as speaker and speech recognition. These attacks consist of adding an imperceptible perturbation to the speech waveform, which is optimized to change the output of the systems.

Though adversarial attacks research started in the image domain [2, 3, 4], it is gaining attention in the audio domain. There are already works that study the effectiveness of adversarial attacks on speech recognition [5, 6, 7, 8, 9, 10], and speaker recognition [11, 12, 13, 14, 15]. Furthermore, some recent works propose pre-processing defenses [16, 17], intended to remove the adversarial noise from the audio; or using adversarial training to robustify speaker identification networks [18].

In this work, we investigate how to detect and classify adversarial attacks against audio systems. This research has been done in the context of the DARPA RED (Reverse Engineering of Deceptions) program [19]. The goal of the program is to produce algorithms for automatically identifying the toolchains behind attacks. Obtaining such information could help to identify the attackers, their intentions, and decide which defenses might be most effective against those attacks. Encouraged by the success of deep learned representations like x-vectors in speaker verification [20, 21], emotion recognition [22], or speech pathology detection [23]; we applied the same class of network architectures to compute embeddings, which can help us decide whether a test recording has been attacked and extract information about the attack. We will denote these embeddings as *attack signatures* to follow DARPA's nomenclature.

We experimented on three tasks: attack classification, verification, and unknown attack detection. Attack classification is the task of deciding whether a test utterance is benign or whether it belongs to a set of known attacks. This classification can be done w.r.t. either the attack algorithm, the threat model, or the perturbation size –e.g., Signal-to-Noise Ratio (SNR). Similarly to speaker verification, the attack verification task is deciding whether two utterances have been attacked in the same way. This can be useful when we have a few samples from an attacker using an unknown deception toolchain, and we want to detect new attacks from this attacker. Finally, the unknown attack detection task is deciding whether an utterance contains an unknown attack, meaning attack which is not included in our training set. This can be useful to detect new attacks so that later we can add them to our database of attacks.

2. Adversarial Attacks

2.1. Threat models

Suppose $\mathbf{x} \in \mathbb{R}^T$ is a benign audio waveform of length T, also called clean, or bonafide. Let y^{benign} be its true labels. An attacker could craft an adversarial example $\mathbf{x}' = \mathbf{x} + \boldsymbol{\delta}$ by adding an imperceptible perturbation $\boldsymbol{\delta}$ to the speech waveform. The adversarial perturbation $\boldsymbol{\delta}$ is optimized to alter the decision of speech or speaker recognition systems. To enforce imperceptibility of the perturbation, some distance metric is minimized or bounded $D(\mathbf{x}, \mathbf{x}') < \varepsilon$. Typically, this is the L_p norm of the perturbation, $D(\mathbf{x}, \mathbf{x}') = \|\boldsymbol{\delta}\|_p$. The choosing of this metric is usually known in the literature as the *threat model* of the attack. In this work, we consider L_0, L_1, L_2 and L_∞ threat models.

2.2. Attack algorithms

2.2.1. PGD and FGSM attacks

The projected gradient descent (PGD) algorithm [24] takes the benign audio waveform of length T, $\mathbf{x} \in \mathbb{R}^T$ and computes an adversarial example $\mathbf{x}' = \mathbf{x} + \boldsymbol{\delta}$, by iteratively optimizing $\boldsymbol{\delta}$ to maximize the misclassification error as

$$\boldsymbol{\delta}_{i+1} = \mathcal{P}_\varepsilon(\boldsymbol{\delta}_i + \alpha\,\text{sign}(\nabla_{\mathbf{x}'_i} L(g(\mathbf{x}'_i), y^{\text{benign}}))) \;, \quad (1)$$

where function $g(\mathbf{x})$ is the speaker/ASR classifier, L is cross-entropy loss, y^{benign} is the true label, and i is the iteration number. The function $\mathcal{P}_\varepsilon$ projects $\boldsymbol{\delta}$ into the l_p ball with radius ε. Usually, $p = \{1, 2, \infty\}$ are used with this attack. Typically, $\boldsymbol{\delta}$ is initialized randomly, and the attacker tries several random initializations and uses the one providing the highest loss.

There are simplified versions of this attack like the fast gradient sign method (FGSM) [2] and Iterative FGSM (Iter-FGSM) [4]. Iter-FGSM is PGD-L_∞ attack without random initializations, i.e., $\boldsymbol{\delta}$ is initialized to zero. FGSM is a a single iteration Iter-FGSM.

http://dx.doi.org/10.21437/Interspeech.2021-1759

2.2.2. *Carlini-Wagner*

The Carlini-Wagner (CW) attack [3] is computed by finding the minimum perturbation $\boldsymbol{\delta}$ that fools the classifier while maintaining imperceptibility. $\boldsymbol{\delta}$ is obtained by minimizing the loss,

$$C(\boldsymbol{\delta}) \triangleq D(\mathbf{x}, \mathbf{x}+\boldsymbol{\delta}) + c\, f(\mathbf{x}+\boldsymbol{\delta}) \qquad (2)$$

where, D is the distance metric of the threat model. L_0, L_2 or L_∞ are used in the literature. By minimizing D, we minimize the perceptibility of the perturbation. f is defined in such a way that the system fails if and only if $f(\mathbf{x}+\boldsymbol{\delta}) \leq 0$. The precise definition for f can be found in [15] and [16] for attacks against speaker verification and classification respectively.

3. Attack Representation Learning

3.1. x-Vectors

In order to learn representations for attack classification/detection, we propose to use the same kind of x-vector architectures that we use for speaker recognition. The x-vector approach uses a neural network to encode the identity/attack information in each speech utterance into a single embedding vector [20]. The x-vector network consists of three parts. First, an encoder network extracts frame-level representations from acoustic features (MFCC, filter-banks). This is followed by a global temporal pooling layer that produces a single vector per utterance–we used mean and standard deviation pooling. Finally, a feed-forward network computes attack class posteriors. The network is trained on a large set of attacks, using some form of cross-entropy loss. We employed additive angular margin softmax (AAM-softmax) [25] in this work. In the evaluation phase, the x-vector embedding is obtained from the first affine transform after pooling, while the last layers of the network are discarded. Different x-vector systems are characterized by different encoder architectures and pooling methods. In this work, we used a Thin-ResNet34 encoder similar to the one in [26] with 16 to 128 channels in the residual blocks. In the context of adversarial attacks, we will denote the embeddings extracted from the x-vector network as *attack signatures*.

3.2. Applications of attack signatures/embeddings

3.2.1. *Attack classification*

This task entails classifying a test utterance into one of the known attack classes or the benign (unattacked) class. We can classify attacks attending to different criteria like optimization algorithm used to compute the adversarial perturbation, threat model, signal-to-noise ratio between the benign signal and the adversarial perturbation, etc. If the classes used to train the signature extractor network match our target classes, we can use the network output logits to classify the test sample. Otherwise, we can train another classifier, e.g., linear-Gaussian, PLDA, or logistic regression, on top of the signature vectors.

3.2.2. *Attack verification/detection*

Similar to speaker verification, attack verification is the task of deciding whether a test utterance contains the same attack as the enrollment utterance(s). In this case, we may have unknown attack types, i.e. attacks that are not included in the training of the signature extractor. We hypothesized that known attacks will define a manifold where unknown attacks also live. We used probabilistic linear discriminant analysis (PLDA) [27] to evaluate the log-likelihood ratio between the *same* vs *different* attack hypothesis.

Table 1: *Probability distributions used to generate attacks with random hyperparameters. In PGD-L_1, norms are normalized by the number of samples n, and in PGD-L_2 by $\sqrt{n}$, to make norm values comparable across utterances of different lengths*

Algorithm	Hyper-parameter	Distribution
(Iter-)FGSM/PGD-L_x	max. L_x (ε)	log-Uniform(3×10^{-6},0.03)
Iter-FGSM/PGD-L_x	learn. rate (α)	log-Uniform(10^{-6}, 0.005)
Iter-FGSM	num. iters.	$1.25\ \varepsilon/\alpha$
PGD-L_x	num. iters.	Uniform(10, 100)
	num. random inits	Uniform(2, 5)
CW-L_x	learn. rate	Uniform(10^{-5},10^{-3})
	confidence	Uniform(0,3)
CW-L_2/L_∞	num. iters.	Uniform(10, 200)
CW-L_0	num. iters.	Uniform(10, 100)

3.2.3. *Unknown attack detection*

Unknown attack detection is the task of deciding whether a test utterance contains an attack not included in our training set. We used a PLDA model to compute the likelihood ratio between the *unknown* vs *known* attack hypothesis. We could prove that this is given by

$$\text{LLR} = -\log \frac{1}{N} \sum_{i=1}^{N} \frac{P\left(\mathbf{x}_{\text{test}}, \mathbf{X}_i | \text{same}\right)}{P\left(\mathbf{x}_{\text{test}}, \mathbf{X}_i | \text{diff}\right)} \qquad (3)$$

where $\mathbf{x}_{\text{test}}$ is the test signature; and $\mathbf{X}_i$ are the signatures of the known attacks of class i.

4. Experiments

4.1. Speaker and speech recognition tasks

For speaker recognition, we created an experimental setup based on the VoxCeleb 1 and 2 datasets [28]. VoxCeleb2-dev set was split into VoxCeleb2-dev-train and VoxCeleb2-dev-test. For each speaker, we put 90% of its utterances in dev-train and 10% in dev-test. VoxCeleb2-dev-train was used to train speaker recognition networks. Meanwhile, VoxCeleb2-dev-test was used as the test set for a speaker classification task. The logits outputs of the speaker recognition network were used as scores for this task. We also used the standard VoxCeleb1-test-original-clean speaker verification task. To evaluate speaker verification trials, we compared enrollment and test x-vector embeddings using cosine scoring. Scores were calibrated into log-likelihood ratios using logistic regression trained on benign trials. The x-Vector architecture was based on a Thin-ResNet34 [26, 15], with 256 embedding dimension, additive angular margin=0.3. As a reference, these systems provided 1.94%, 1.91%, and 3.2% EER in the VoxCeleb1 original, entire and hard tasks. We did not use a larger x-vector architecture to alleviate the huge computing cost of generating adversarial attacks. In [15], we showed that x-vector EER can degrade to 50% (the worst possible) under adversarial attacks, even with very high signal-to-perturbation-noise ratios ($>$ 20 dB).

For automatic speech recognition (ASR), we trained our systems on the LibriSpeech [29] 960 hours train split. We tested on the first 100 utterances of the test-clean split, also to limit the computing cost. We evaluated Espresso [30] ASR system, based on the Transformer encoder-decoder architecture [31]. This setup is the same as [17].

4.2. Adversarial attack generation

We generated attacks against VoxCeleb2-dev-train and VoxCeleb2-dev-test speaker classification; VoxCeleb1 speaker verification; and LibriSpeech-test ASR. For speaker recognition, we generated FGSM, Iter-FGSM, PGD-L_∞/L_1/L_2 and CW-L_∞/L_0/L_2 attacks using our own implementations.

Table 2: *Normalized confusion matrix (%) for attack algorithm classification in speaker recognition.*

	Benign	CW-L0	CW-L2	CW-Linf	FGSM	Iter-FGSM	PGD-L1	PGD-L2	PGD-Linf
VoxCeleb2-dev-test speaker classification attacks									
Benign	**98.8**	0.3	0.5	0.3	0.0	0.0	0.0	0.0	0.1
CW-L0	1.2	**98.6**	0.1	0.2	0.0	0.0	0.0	0.0	0.0
CW-L2	2.8	0.1	**82.9**	11.4	0.0	0.3	0.3	0.6	1.5
CW-Linf	0.9	0.0	3.5	**95.0**	0.0	0.0	0.0	0.0	0.6
FGSM	1.5	0.1	1.1	0.1	**94.3**	0.7	0.5	1.3	0.3
Iter-FGSM	1.7	0.0	1.9	0.1	2.6	**82.4**	0.0	0.3	10.9
PGD-L1	1.4	0.0	0.9	0.0	0.1	0.0	**91.4**	6.2	0.0
PGD-L2	1.2	0.0	2.6	0.0	0.6	0.9	**46.5**	42.7	5.6
PGD-Linf	1.3	0.0	6.9	1.4	0.1	12.5	0.0	3.0	**74.6**
VoxCeleb1-test speaker verification attacks									
Benign	**94.1**	0.3	1.3	4.1	0.0	0.1	0.0	0.1	0.1
CW-L0	1.7	**97.9**	0.1	0.3	0.0	0.0	0.0	0.0	0.0
CW-L2	4.5	0.2	**75.4**	12.4	0.2	0.8	0.3	1.2	5.0
CW-Linf	1.4	0.1	4.1	**93.5**	0.0	0.0	0.0	0.0	0.9
FGSM	3.0	0.1	2.7	0.2	**90.4**	1.3	0.2	1.7	0.4
Iter-FGSM	2.0	0.0	5.8	0.1	2.3	**74.9**	0.0	1.6	13.3
PGD-L1	2.7	0.0	2.8	0.3	0.3	0.1	**85.2**	8.6	0.1
PGD-L2	2.4	0.1	4.0	0.2	1.1	1.0	33.5	**49.8**	7.8
PGD-Linf	2.7	0.1	9.5	1.5	0.2	3.8	0.0	3.3	**79.0**

Table 3: *Normalized confusion matrix (%) for attack threat model classification in speaker recognition*

	VoxCeleb2-dev-test speaker classification attacks					VoxCeleb1-test speaker verification attacks				
	Benign	L0	L1	L2	Linf	Benign	L0	L1	L2	Linf
Benign	**99.3**	0.2	0.0	0.4	0.1	**96.1**	0.2	0.0	1.4	2.3
L0	1.3	**98.6**	0.0	0.0	0.2	1.7	**98.1**	0.0	0.1	0.1
L1	1.4	0.0	**92.0**	6.3	0.2	2.6	0.0	**88.0**	9.0	0.3
L2	3.1	0.0	17.6	**71.4**	7.8	4.2	0.1	12.9	**71.2**	11.6
Linf	1.7	0.0	0.2	9.6	**88.4**	2.5	0.0	0.0	12.6	**84.8**

To better model the variability of the attack manifold, the attack hyper-parameters were randomly sampled from different probability distributions as shown in Table 1. For each attack algorithm-threat model, we generated attacks for 25% of utterances in the datasets. However, we only kept those attacks that were successful in changing the classification label from correct to incorrect. For speaker verification, we used a decision threshold–$\theta = \text{logit} P_{\mathcal{T}}$ with target prior $P_{\mathcal{T}} = 0.05$, to classify trials as target or non-target.

For ASR, we generated FGSM and PGD-$L_\infty/L_1/L_2$ attacks with norm values $\varepsilon = \{0.0001, 0.001, 0.01, 0.1, 0.2\}$. The experimental setup to generate ASR attacks is publicly available in the Armory toolkit [32], which uses the attack algorithms in the Adversarial Robustness Toolbox (ART) [33].

4.3. Attack signature extraction networks

The signature extraction networks were built with the same Thin-ResNet34 x-Vector architecture used for speaker recognition, but with embedding dimension empirically set to 10 and margin=0.2. We did not use noise and reverberation augmentation since it did not improve the attack classification performance. The attacked and benign samples in VoxCeleb2-dev-train were used to train three extractors. Each extractor detects a different attack property: attack algorithm+threat-model; threat-model; and signal-to-adversarial-noise ratio (SNR). Additionally, three extra extractors were trained leaving out the CW attacks to evaluate the effect of having unknown attacks in the test set. These signature extractors, trained on speaker classification attacks, were evaluated on attacks against speaker classification and verification; and ASR.

4.4. Attack classification in speaker recognition

For these experiments, we used the three networks trained on all the attack algorithms. Table 2 shows confusion matrices[1] for algorithm+threat-model classification. For attacks against VoxCeleb2-dev-test speaker classification, we obtained 90.2% attack classification accuracy. Benign samples were correctly classified with 98.8% Acc.; and most attack algorithms obtained more than 74% Acc. The largest confusion was between PGD-L_2 and L_1. 33% of PGD-L_2 attacks were assigned to PGD-L_1. Thus, we found that is difficult to discriminate between L_1 and L_2 threat models. There was also significant confusion between PGD-L_∞ and Iter-FGSM. However, this was expected since Iter-FGSM is just a PGD-L_∞ without random re-starts. The T-SNE plot in Figure 1a presents similar evidence. We observe a significant overlap between PGD-L_1/L_2; and PGD-L_∞ and Iter-FGSM; while the other classes are fairly well separated. For the attacks against VoxCeleb1 speaker verification, we obtained 84.6% accuracy, and the confusion matrix follows a pattern similar to the previous one. These results suggest that an attack classification model trained on a speaker classification task–closet-set of speakers, multi-class objective–, can be transferred to a verification task–open-set, unknown speakers, binary decision–, with small performance degradation.

Table 4: *Normalized confusion matrix (%) for attack SNR(dB) classification in speaker recognition*

SNR(dB)	Benign	0	10	20	30	40	50	60
Attack on VoxCeleb2-dev-test Speaker Classification Task								
Benign	**98.7**	0.0	0.0	0.0	0.2	0.5	0.5	0.3
0	0.0	**87.3**	12.7	0.0	0.0	0.0	0.0	0.0
10	0.0	2.7	**90.8**	6.4	0.0	0.1	0.0	0.0
20	0.2	0.0	5.3	**84.3**	10.1	0.2	0.0	0.0
30	0.5	0.0	0.0	7.7	**78.8**	12.9	0.0	0.0
40	1.5	0.0	0.0	0.0	15.4	**66.9**	16.1	0.1
50	3.7	0.0	0.0	0.0	0.2	11.7	**80.7**	3.7
60	20.3	0.0	0.0	0.0	0.0	2.2	**50.6**	26.9
Attack on VoxCeleb1-test Speaker Verification Task								
Benign	**93.4**	0.0	0.0	0.0	0.9	2.8	2.4	0.4
0	0.0	**79.4**	20.6	0.0	0.0	0.0	0.0	0.0
10	0.0	1.3	**87.4**	11.3	0.0	0.0	0.0	0.0
20	0.1	0.0	4.1	**78.5**	17.1	0.2	0.0	0.0
30	0.3	0.0	0.0	5.4	**69.6**	24.3	0.4	0.0
40	2.3	0.0	0.0	0.0	10.9	**64.7**	21.8	0.2
50	5.0	0.0	0.0	0.0	0.4	11.9	**76.5**	6.3
60	28.9	0.0	0.0	0.1	0.7	4.1	**45.0**	21.1

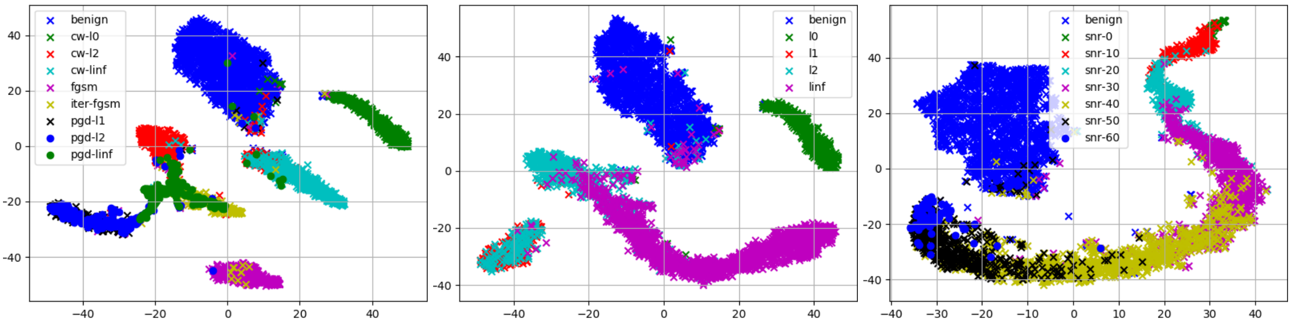
Second, we look at the threat-model classification task. Here, accuracies were 90.7% and 86% for attacks against speaker classification and verification respectively. Confusion matrices in Table 3 show accuracies larger than 84% for all threat models, except L_2, which is again 17% misclassified with L_1. The T-SNE plot in Figure 1b shows us two clusters for L_2 threat model, one for CW-L_2 and another for PGD-L_2. The PGD-L_2 cluster is overlapped with PGD-L_1 cluster. The Benign, L_0 and L_∞ clusters are well separated. Again, we observe that the threat-model classifier transferred well from the speaker classification to the verification task.

Finally, Table 4 and Figure 1c present results for SNR classification. Here, we note that most samples are classified in the correct SNR bin or in the ones immediately next to it. The largest error was for attacks at SNR=60dB, which were 20-28% misclassified as benign samples. However, few attacks were successful in fooling our speaker classifier at 60 dB (316), compared to 50 dB (1738) and 40 dB (3475).

4.5. Attack classification in speech recognition

We evaluated our attack classifier on attacks against ASR, obtained 5% accuracy with about 50% of attack files classified as

[1] Confusion matrices have ground truth in rows, predictions in columns and are row normalized



(a) *Attack algorithm*

(b) *Threat model*

(c) *SNR(dB)*

Figure 1: *T-SNE plots for attack signatures extracted from adversarial attacks on VoxCeleb2 dev-test speaker classification task. Signature extractors trained on our VoxCeleb2 dev-train set to discriminate attack algorithm, threat model or SNR.*

Table 5: *Normalized confusion matrices (%) for attack algorithm classification in ASR*

	With Benign					Without Benign			
	Benign	FGSM	PGD-L1	PGD-L2	PGD-Linf	FGSM	PGD-L1	PGD-L2	PGD-Linf
Benign	31.2	1.0	**43.7**	24.0	0.0	-	-	-	-
FGSM	8.8	**71.8**	1.5	14.3	3.6	**73.7**	1.7	21.1	3.6
PGD-L1	33.1	2.5	**46.3**	18.2	0.0	3.7	47.1	**49.2**	0.0
PGD-L2	29.7	6.6	29.1	**30.1**	4.5	8.5	29.7	**57.3**	4.5
PGD-Linf	11.1	3.2	1.1	21.9	**62.8**	4.0	1.1	32.2	**62.8**

Table 6: *EER(%) for attack detection tasks. Attacks are against VoxCeleb2-dev-test speaker classification. Benign, FGSM and PGD are known attacks, CW attacks are unknown.*

	Attack group		
Detection Task	Known+Unknown	Known	Unknown
Algorithm+Threat-Model	**19.6**	12.6	41.5
Threat-Model	24.8	**7.8**	**36.6**
SNR(dB)	24.8	7.9	44.2

benign. Then, we trained a PLDA classifier on top of the attack signatures using 100 ASR attack samples (20 per attack class), and tested on the remaining 1900 samples. When we consider benign samples in the experiment, accuracy was 50%. In Table 5 (*with benign*), we observe significant confusion between benign, PGD-L_1/L_2, while FGSM and PGD-L_∞ had better accuracies. If we do not consider the benign class–assuming that we know that we are under attack–, accuracy raises to 60%, but still with large confusion between L_1 and L_2 threat models as we observed in the speaker recognition attacks.

We also evaluated our SNR classifier on the ASR attacks, obtaining promising results. Here, 82% of benign samples were correctly classified and $> 88\%$ attacks with SNR$\in [0, 40]$ were classified with error $\leq$ 10 dB. High SNR attacks were more problematic. 50 dB and 60 dB attacks were classified as benign with 30% and 76% probabilities respectively.

4.6. Attack detection

In this section, we approach the attack detection/verification task in a similar way as the speaker verification task. That is, given two utterances, we decide whether both contain the same or different attacks. We used a PLDA model to compute the log-likelihood ratio between those two hypotheses. For this experiment, we used attack signatures and PLDA trained without CW attacks. Thus, benign, FGSM and PGD versions are known and CW versions are unknown. We created attack algorithm/threat-model/SNR verification trials using the adversarial samples obtained by attacking the VoxCeleb2-dev-test speaker classification task. In total, we obtained about 10M trials for each one of the attack verification tasks. Table 6 shows the results in terms of equal error rate (EER). When considering only trials between known attacks, EER was low (8-12%). However, for trials between unknown attacks, EER raised to 36-44%–still better than chance. The best result was for threat-model detection with 36% EER. This result indicates that signature extractors need to improve to be robust to unknown attacks. One possible direction is to train our signatures with a wider range of attack algorithms from those available in the literature.

4.7. Unknown attack detection

Here, we decide whether a test utterance contains an attack seen in training or not. For this experiment, we used the signature extractor trained to discriminate attack algorithms on FGSM and PGD versions. CW versions were unknown attacks. A PLDA model was used to compute the likelihood ratio between the unknown and known attack hypothesis. We considered two conditions: with and without the benign class. In the former, the test utterance can be benign or attacked and we obtained 37.3% EER. In the latter, all test utterances are adversarial–we assume that we have an oracle detector that tells us that we are under attack–, and we obtained 19% EER. We think that this is a promising result that can be improved with further research.

5. Conclusions

In this paper, we applied representation learning based on x-vector architectures to several tasks related to the classification and detection of adversarial attacks against speaker and speech recognition. First, we showed that the most common adversarial attacks in the literature–FGSM, PGD, Carlini-Wagner–can be detected and classified with accuracies as high as 90%. We also showed that neural networks trained to classify attacks against speaker identification can be used to classify attacks against speaker verification with small performance degradation. Furthermore, our learned representations termed as attack signatures can be transferred to classify attacks against speech recognition. We also performed an attack verification task finding that attack signatures did generalize poorly to unknown attack algorithms. Finally, we evaluated an unknown attack detection task. We found that unknown attacks can be detected with equal error rates of about 19%, which is promising.

6. Acknowledgements

Research supported by DARPA RED HR00112090132.

7. References

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *ICLR 2014*, 2014.

[2] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," in *ICLR 2015*, dec 2015.

[3] N. Carlini and D. Wagner, "Towards Evaluating the Robustness of Neural Networks," in *IEEE Symposium on Security and Privacy, 2017*, aug 2016.

[4] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *CoRR 2017*, jul 2017.

[5] M. Cisse, Y. Adi, N. Neverova, and J. Keshet, "Houdini: Fooling Deep Structured Prediction Models," in *NIPS 2017*, jul 2017, pp. 6977—-6987.

[6] D. Iter, J. Huang, and M. Jermann, "Generating adversarial examples for speech recognition," Tech. Rep., 2017.

[7] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *SPW 2018*, 2018.

[8] P. Neekhara, S. Hussain, P. Pandey, S. Dubnov, J. McAuley, and F. Koushanfar, "Universal Adversarial Perturbations for Speech Recognition Systems," in *INTERSPEECH 2019*, Graz, Austria, sep 2019, pp. 481–485.

[9] L. Schonherr, K. Kohls, S. Zeiler, T. Holz, and D. Kolossa, "Adversarial Attacks Against Automatic Speech Recognition Systems via Psychoacoustic Hiding," in *NDSS 2019*, Reston, VA, 2019.

[10] Y. Qin, N. Carlini, I. Goodfellow, G. Cottrell, and C. Raffel, "Imperceptible, Robust, and targeted adversarial examples for automatic speech recognition," in *ICML 2019*, 2019, pp. 9141–9150.

[11] F. Kreuk, Y. Adi, M. Cisse, and J. Keshet, "Fooling End-To-End Speaker Verification With Adversarial Examples," in *ICASSP 2018*, apr 2018, pp. 1962–1966.

[12] Y. Gong and C. Poellabauer, "Crafting Adversarial Examples For Speech Paralinguistics Applications," in *DYnamic and Novel Advances in Machine Learning and Intelligent Cyber Security (DYNAMICS) Workshop*, San Juan, Puerto Rico, dec 2018.

[13] Y. Xie, C. Shi, Z. Li, J. Liu, Y. Chen, and B. Yuan, "Real-Time, Universal, and Robust Adversarial Attacks Against Speaker Recognition Systems," in *ICASSP 2020*, may 2020, pp. 1738–1742.

[14] X. Li, J. Zhong, X. Wu, J. Yu, X. Liu, and H. Meng, "Adversarial Attacks on GMM I-Vector Based Speaker Verification Systems," in *ICASSP 2020*, Barcelona, Spain, may 2020, pp. 6579–6583.

[15] J. Villalba, Y. Zhang, and N. Dehak, "x-Vectors Meet Adversarial Attacks : Benchmarking Adversarial Robustness in Speaker Verification," in *Interspeech 2020*, Shanghai, China, sep 2020.

[16] S. Joshi, J. Villalba, P. Żelasko, L. Moro-Velázquez, and N. Dehak, "Adversarial attacks and defenses for speaker identification systems," *arXiv preprint arXiv:2101.08909*, 2021.

[17] P. Żelasko, S. Joshi, Y. Shao, J. Villalba, J. Trmal, N. Dehak, and S. Khudanpur, "Adversarial attacks and defenses for speech recognition systems," *arXiv preprint arXiv:2103.17122*, 2021.

[18] A. Jati, C.-C. Hsu, M. Pal, R. Peri, W. AbdAlmageed, and S. Narayanan, "Adversarial attack and defense strategies for deep speaker recognition systems," *Computer Speech and Language*, vol. 68, p. 101199, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0885230821000061

[19] "Reverse Engineering of Deceptions(GARD)," http://aiweb.techfak.uni-bielefeld.de/content/bworld-robot-control-software/, 2021, [Online; accessed 18-July-2021].

[20] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-Vectors : Robust DNN Embeddings for Speaker Recognition," in *ICASSP 2018*, Alberta, Canada, apr 2018, pp. 5329–5333.

[21] J. Villalba, D. Garcia-Romero, N. Chen, G. Sell, J. Borgstrom, A. McCree, L. P. Garcia-Perera, S. Kataria, P. S. Nidadavolu, P. A. Torres-Carrasquillo, and N. Dehak, "Advances in Speaker Recognition for Telephone and Audio-Visual Data : the JHU-MIT Submission for NIST SRE19," in *Odyssey 2020*, Tokyo, Japan, 2020.

[22] R. Pappagari, T. Wang, J. Villalba, N. Chen, and N. Dehak, "X-Vectors Meet Emotions: A Study On Dependencies Between Emotion and Speaker Recognition," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Barcelona, Spain: IEEE, may 2020, pp. 7169–7173. [Online]. Available: https://ieeexplore.ieee.org/document/9054317/

[23] L. Moro-Velazquez, J. Villalba, and N. Dehak, "Using X-Vectors to Automatically Detect Parkinson's Disease from Speech," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Barcelona, Spain: IEEE, may 2020, pp. 1155–1159. [Online]. Available: https://ieeexplore.ieee.org/document/9053770/

[24] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations*, 2018.

[25] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," in *CVPR 2019*, 2019.

[26] H. Zeinali, S. Wang, A. Silnova, P. Matějka, and O. Plchot, "BUT System Description to VoxCeleb Speaker Recognition Challenge 2019," in *The VoxSRC Workhsop 2019*, oct 2019.

[27] P. Kenny, "Bayesian Speaker Verification with Heavy-Tailed Priors," in *Odyssey 2010*, Brno, Czech Republic, jul 2010.

[28] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Computer Speech and Language*, vol. 60, 2020.

[29] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[30] Y. Wang, T. Chen, H. Xu, S. Ding, H. Lv, Y. Shao, N. Peng, L. Xie, S. Watanabe, and S. Khudanpur, "Espresso: A fast end-to-end neural speech recognition toolkit," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 136–143.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[32] "Armory Toolkit," https://github.com/twosixlabs/armory/blob/master/docs/scenarios.md, 2021, [Online; accessed 18-July-2021].

[33] "Adversarial Robustness Toolkit (ART)," https://github.com/Trusted-AI/adversarial-robustness-toolbox, 2021, [Online; accessed 18-July-2021].