DEFENSE AGAINST ADVERSARIAL ATTACKS ON SPOOFING COUNTERMEASURES OF ASV

Haibin Wu[1*], *Songxiang Liu*[2*], *Helen Meng*[2], *Hung-yi Lee*[1]

[1] Graduate Institute of Communication Engineering, National Taiwan University
[2]Human-Computer Communications Laboratory, The Chinese University of Hong Kong

ABSTRACT

Various forefront countermeasure methods for automatic speaker verification (ASV) with considerable performance in anti-spoofing are proposed in the ASVspoof 2019 challenge. However, previous work has shown that countermeasure models are vulnerable to adversarial examples indistinguishable from natural data. A good countermeasure model should not only be robust against spoofing audio, including synthetic, converted, and replayed audios; but counteract deliberately generated examples by malicious adversaries. In this work, we introduce a passive defense method, spatial smoothing, and a proactive defense method, adversarial training, to mitigate the vulnerability of ASV spoofing countermeasure models against adversarial examples. This paper is among the first to use defense methods to improve the robustness of ASV spoofing countermeasure models under adversarial attacks. The experimental results show that these two defense methods positively help spoofing countermeasure models counter adversarial examples.

Index Terms— Adversarial attack, spoofing countermeasure, adversarial training, anti-spoofing, spatial smoothing

1. INTRODUCTION

Automatic speaker verification, abbreviated as ASV, refers to the task of ascertaining whether an utterance was spoken by a specific speaker. ASV is undisputedly a crucial technology for biometric identification, which is broadly applied in real-world applications like banking and home automation. Considerable performance improvements in terms of both accuracy and efficiency of ASV systems have been achieved through active research in a diversity of approaches [1–6]. [4] proposed a method that use the Gaussian mixture model to extract acoustic features and then apply the likelihood ratio for scoring. An end-to-end speaker verification model that directly maps an utterance into a verification score is proposed by [5] to improve verification accuracy and make the ASV model compact and efficient.

Recently, improving the robustness of ASV systems against spoofing audios, such as synthetic, converted, and replayed audios, has attracted increasing attention. The automatic speaker verification spoofing and countermeasures challenge [7–9], which is now in its third edition, aims at developing reliable spoofing countermeasures which can counteract the three kinds of spoofing audios mentioned above. The ASVspoof 2019 takes both logical access (LA) and physical access (PA) into account. The LA scenario contains artificially generated spoofing audios by modern text-to-speech and voice conversion models, and the PA scenario contains replayed audios. A variety of state-of-the-art countermeasure methods that aim at anti-spoofing for ASV models are proposed, and these have achieved considerable performance level for anti-spoofing [10–14]. However, whether these countermeasure models can defend against deliberately generated adversarial examples remain to be investigated.

Adversarial examples [15] are generated by maliciously perturbing the original input with a small noise. The perturbations are almost indistinguishable to humans but can cause a well-trained network to classify incorrectly. Using deliberately generated adversarial examples to attack machine learning models is called adversarial attack. Previous work has shown that image classification models are subject to adversarial attacks [15]. The spoofing countermeasure models for ASV learned by the backpropagation algorithm also have such intrinsic blind spots to adversarial examples [16]. These intrinsic blind spots must be fixed to ensure safety.

To mitigate the vulnerability of spoofing countermeasure models to adversarial attacks, we introduce a passive defense method, namely spatial smoothing, and a proactive defense method, namely adversarial training. Two countermeasure models in ASVspoof 2019 [11, 12] are constructed, and we implement adversarial training and spatial smoothing to improve the reliability of these two models. This work is among the first to explore defense against adversarial attacks for spoofing countermeasure models.

This paper is organized as follows. In section 2, we introduce the procedure of adversarial example generation. Section 3 gives the detailed structure of two countermeasure

[*] Equal contribution. Songxiang Liu and Helen Meng were supported by the General Research Fund from the Research Grants Council of Hong Kong SAR Government (Project No. 14208718). Haibin Wu and Hung-yi Lee were supported by the Ministry of Science and Technology of Taiwan.

models for subsequent experiments. In Section 4, we describe two defense approaches, namely spatial smoothing and adversarial training. The experimental result and analysis are shown in Section 5. Finally, conclusion and future work are given in Section 6.

2. ADVERSARIAL EXAMPLE GENERATION

2.1. Adversarial Example Generation

We can generate adversarial examples by adding a minimally perceptible perturbation to the input space. The perturbation is found by solving an optimization problem. There are two kinds of adversarial attacks: targeted attacks and nontargeted attacks. Targeted attacks aim at maximizing the probability of a targeted class which is not the correct class. Nontargeted attacks aim at minimizing the probability of the correct class. We focus on targeted attacks in this work. Specifically, to generate adversarial examples, we fix the parameters θ of a well-trained model and perform gradient descent to update the input. Mathematically, we want to find a sufficiently small perturbation δ that satisfies (see Equation 1):

$$\begin{aligned} \tilde{x} &= x + \delta, \\ f_\theta(x) &= y, \\ f_\theta(\tilde{x}) &= \tilde{y}, \\ \delta &\in \Delta, \end{aligned} \tag{1}$$

where f is a well-trained neural network parameterized by θ, $x \in R^N$ is the input data with dimensionality N, y is the true label corresponding to x, $\tilde{y}$ is a randomly selected label where $\tilde{y} \neq y$, $\tilde{x} \in R^N$ is the perturbed data, $\delta \in R^N$ is a small perturbation and Δ is the feasible set of δ. Finding a suitable δ is a constrained optimization problem and we can use descent method to solve it. Δ can be a small l_∞-norm ball:

$$\Delta = \{\delta | \, ||\delta||_\infty \leq \epsilon\}, \tag{2}$$

where $\epsilon \geq 0$ and $\epsilon \in R$. The constraint in Equation 2 is a box constrain and clipping is used to make the solution $\tilde{x}$ feasible. We choose the feasible set Δ as shown in Equation 2.

The projected gradient descent method, abbreviated as PGD method is an iterative method for adversarial attack and has shown effective attack performance in various tasks [17]. In this work, the PGD method is introduced to generate adversarial examples. The PGD method is specified in Algorithm 1. In Algorithm 1, x_K is the returned adversarial example, the clip() function applies element-wise clipping to make sure $||x_k - x||_\infty \leq \epsilon$ and $\epsilon \in R^+$.

Algorithm 1 Projected Gradient Descent Method

```
Require: $x$ and $y$, input and its corresponding label. $\tilde{y}$ is a
    selected label and $\tilde{y} \neq y$. $\alpha$, step size. $K$, the number of
    iterations.
Initialize $x_0 = x$;
for $k = 0; k < K; k + +$ do
    $x_{\hat{k+1}} = \text{clip}(x_k + \alpha \cdot \text{sign}(\nabla_{x_k}\text{Loss}(\theta, x_k, \tilde{y})))$;
    if $\text{Loss}(\theta, x_{k+1}, \tilde{y}) < \text{Loss}(\theta, x_k, \tilde{y})$ then
        $x_{k+1} = x_{\hat{k+1}}$
    else
        $x_{k+1} = x_k$
    end if
end for
return $x_K$;
```

3. ASV SPOOFING COUNTERMEASURE MODELS

Inspired by the ASV spoofing countermeasure models in the ASVspoof 2019 challenge [9, 11, 12], we construct two kinds of single models to conduct defense methods. The description of these two models will be given in the subsequent parts.

3.1. VGG-like Network

The VGG network, a model made up of convolution layers and pooling layers, has shown remarkable performance in image classification. [11] studied VGG from the perspective of automatic speaker verification and proposed a VGG-like network with good performance on anti-spoofing for ASV. Based on this finding, we modified VGG to address anti-spoofing and the modified network structure is shown in Table 1.

Table 1. VGG-like network architecture.

Type	Filter	Output
Conv2D-1-1	3×3	$64 \times 600 \times 257$
MaxPool-1	2×2	$64 \times 300 \times 128$
Conv2D-2-1	3×3	$128 \times 300 \times 128$
MaxPool-2	2×2	$128 \times 150 \times 64$
Conv2D-3-1	3×3	$256 \times 150 \times 64$
Conv2D-3-2	3×3	$256 \times 150 \times 64$
MaxPool-3	2×2	$256 \times 75 \times 32$
Conv2D-4-1	3×3	$512 \times 75 \times 32$
Conv2D-4-2	3×3	$512 \times 75 \times 32$
MaxPool-4	2×2	$512 \times 37 \times 16$
Conv2D-5-1	3×3	$512 \times 37 \times 16$
Conv2D-5-2	3×3	$512 \times 37 \times 16$
MaxPool-5	2×2	$512 \times 18 \times 8$
Avgpool	–	$512 \times 7 \times 7$
Flatten	–	25088
FC	–	4096
FC	–	4096
FC(softmax)	–	2

3.2. Squeeze-Excitation ResNet model

Lai et al. [12] proposed the Squeeze-Excitation ResNet model (SENet) to address anti-spoofing for ASV. The system proposed by [12] ranked 3rd and 14th for the PA and LA scenarios respectively in the ASVspoof 2019 challenge. However,

[16] successfully attacked the SENet by deliberately generated adversarial examples. Hence, this work seeks to improve the robustness of SENet with two defense methods elaborated below.

4. DEFENSE METHODS

There are two kinds of defense methods against adversarial attacks: passive defense and proactive defense. Passive defense methods aim at countering adversarial attacks without modifying the model. Proactive defense methods train new models which are robust to adversarial examples. Two defense methods are introduced in this section: spatial smoothing which is inexpensive and complementary to other defense methods and adversarial training.

4.1. Spatial Smoothing

Spatial smoothing (referred as ”filtering”) has been widely used for noise reduction in image processing. It is a method that uses the nearby pixels to smooth the central pixel. There are a variety of smoothing methods based on different weighting mechanisms of nearby pixels, e.g., median filter, mean filter, Gaussian filter, etc. Take the mean filter as an example, a slicing window moves over the picture and the central pixel in the window will be substituted by the mean of the values within the slicing window.

Spatial smoothing was introduced by [18] to harden image classification models by detecting malicious generated adversarial examples. Implementing smoothing does not need extra training effort, so we use this inexpensive strategy to improve the robustness of well-trained ASV models.

4.2. Adversarial Training

Adversarial training, which utilizes adversarial examples and injects them into training data, was introduced in [19] to mitigate the vulnerability of deep neural networks against adversarial examples. Adversarial training can be seen as a combination of an inner optimization problem and an outer optimization problem where the goal of the inner optimization is to find imperceptible adversarial examples and the goal of outer optimization is to fix the blind spots. In this work, we also employ adversarial training. First, we use clean examples to pre-train the countermeasure models for T_1 epochs. Then we do adversarial training for T_2 epochs. The detailed implementation procedure is shown in Algorithm 2.

5. EXPERIMENT

5.1. Experiment Setup

In this paper, we use the LA partition of the ASVspoof 2019 dataset [9]. The LA partition is divided into training, development and evaluation sets. The training and development sets are generated by the same kinds of TTS or VC models while the evaluation set contains examples generated by different kinds of TTS or VC models. We trained and then tested on the development set to ensure similar distributions between the datasets. Raw log power magnitude spectrum computed from raw audio waveform is used as acoustic features. A Hamming window of size 1724 and step-size of 0.001s is used to extract FFT spectrum. We use only the first 600 frames of each utterance for training and testing. We do not employ additional preprocessing methods such as dereverberation or pre-emphasis.

Algorithm 2

```
Require: $X$ and $Y$, set of paired audio and its corresponding
    labels. $\theta$, network parameters. $T_1$, normal training epoch.
    $T_2$, adversarial training epoch. $N$, number of training
    examples, $b$, batch size.
Initialize $\theta$.
for $t = 0; t < T_1; t++$ do
    for $i = 0; i < N/b; i++$ do
        Get $\{(x_i, y_i)\}_{i=1}^{i=b}$ from $\{X, Y\}$;
        Update $\theta$ using gradient decent with respect to
        $\{(x_i, y_i)\}_{i=1}^{i=b}$;
    end for
end for
while $\{t <= T_2$ & $\theta$ not converged$\}$ do
    for $i = 0; i < N/b; i++$ do
        Get $\{(x_i, y_i)\}_{i=1}^{i=b}$ from $\{X, Y\}$;
        Generate adversarial examples $\{(\tilde{x_i})\}_{i=1}^{i=b}$ by PGD
        method;
        Update $\theta$ using gradient decent with respect to
        $\{(\tilde{x_i}, y_i)\}_{i=1}^{i=b}$;
    end for
end while
return $\theta$;
```

The network structures of the two countermeasure models were as described in Section 3. During the experiment, we first use the training data to pre-train the countermeasure models. Then the PGD method as shown in Algorithm 1 is adopted to generate adversarial examples for the well-trained countermeasure models. When we run the PGD method, ϵ is set to 5, K is set to 10 and α is set to 0.5. Then we measure the performance of well-trained countermeasure models by the generated adversarial examples with and without filters. Three kinds of filters including median filter, mean filter and Gaussian filter are implemented. Then we use adversarial training to train the countermeasure model for T_2 epochs as shown in Algorithm 2. After adversarial training, we evaluate the testing accuracy of countermeasure models for adversarial examples.

5.2. Results and Analyses

5.2.1. Spatial Smoothing

After we pre-train VGG and SENet for T_1 epochs, we evaluate the testing accuracy of these two models. According to Table 2, both SENet and VGG achieve high testing accuracy in the testing data which is not perturbed. However, when we test the two models with adversarial examples, the testing accuracy drops drastically. When we apply spatial smoothing to the adversarial examples and then evaluate the performance, the adversarial attack becomes ineffective as there is a great increase in testing accuracy. All three kinds of spatial filters have considerable performance in improving the robustness of countermeasure models against adversarial examples. The improvement obtained with Gaussian filters is much less than the other two filters.

We attempt to explain the contribution of spatial smoothing contributes to spoofing countermeasure model to be robust against adversarial examples. In the adversarial attack scenario, an adversary has full access to a well-trained model but can not alter the parameters of the model. Now, assuming that the adversary is not aware of the existence of spatial smoothing which will be implemented to the input data before the input is thrown into the model. The adversary attempts to find an imperceptible noise which will cause the well-trained model to classify incorrectly by the PGD method and add it to the input. However, the deliberately generated perturbation will be countered by spatial smoothing and the adversarial attack becomes invalid.

Table 2. Testing accuracy of VGG and SENet before adversarial training.

	SENet	VGG
Normal examples	99.97%	99.99%
Adversarial examples	48.32%	37.06%
Adversarial examples + median filter	82.00%	92.72%
Adversarial examples + mean filter	82.39%	93.95%
Adversarial examples + Gaussian filter	78.93%	84.39%

5.2.2. Adversarial Training

As shown in Table 3, the testing accuracy for adversarial examples of SENet increases from 48.32% to 92.40% while the testing accuracy for normal examples changes little after adversarial training. We can see a similar phenomenon for VGG. According to Table 3, adversarial training does improve the robustness of VGG and SENet.

Traditional supervised training does not address the chosen models to be robust to adversarial examples. So the well-trained models by traditional supervised learning may be sensitive to changes in its input space and thus have vulnerable blind spots that can be attacked by a malicious adversary. During the training stage, adversarial attacks should be taken into account by training on a mixture of data which contains not only clean examples but also adversarial examples to regularize and make the model insensitive on all data points within the ϵ max norm box. After doing that, it is hard for malicious adversaries to generate adversarial examples to attack the model. Adversarial training largely samples adversarial examples within the ϵ max norm box to augment the training set. The results in Table 3 illustrate that it is feasible and practical to train a robust countermeasure model using adversarial training.

Table 3. Testing accuracy of VGG and SENet after adversarial training.

	SENet	VGG
Normal examples	99.75%	99.99%
Adversarial examples	92.40%	98.60%
Adversarial examples + median filter	93.74%	98.96%
Adversarial examples + mean filter	93.76%	99.24%
Adversarial examples + Gaussian filter	83.72%	87.22%

5.2.3. Adversarial Training + Spatial Smoothing

We combine spatial smoothing and adversarial training and the experiment results are shown in Table 3. We observe that equipping adversarial training with median filters or mean filters increase the testing accuracy for adversarial examples, as compared to solely using adversarial training. But adding Gaussian filters decreases the testing accuracy. Hence, Median filters and mean filters are more desirable filters than Gaussian filters in our experiment setting.

6. CONCLUSION

In this paper, two kinds of defense methods, namely spatial smoothing and adversarial training, are introduced to improve the robustness of spoofing countermeasure models under adversarial attacks. We implement two countermeasure models, i.e., VGG and SENet and augment them with defense methods. The experiment results show both spatial smoothing and adversarial training enhance robustness of the models against adversarial attacks.

For future work, we will introduce powerful defense methods, such as ensemble adversarial training [20], to make spoofing countermeasure models more robust to adversarial audios generated from testing data having different distribution with training data.

7. REFERENCES

[1] Ahilan Kanagasundaram, Robbie Vogt, David B Dean, Sridha Sridharan, and Michael W Mason, "I-vector based speaker recognition on short utterances," in *Proceedings of the 12th Annual Conference of the International Speech Communication Association*. International Speech Communication Association (ISCA), 2011, pp. 2341–2344.

[2] Daniel Garcia-Romero and Carol Y Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Twelfth annual conference of the international speech communication association*, 2011.

[3] Andrew Senior and Ignacio Lopez-Moreno, "Improving dnn speaker independence with i-vector inputs," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 225–229.

[4] 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.

[5] Georg Heigold, Ignacio Moreno, Samy Bengio, and Noam Shazeer, "End-to-end text-dependent speaker verification," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5115–5119.

[6] Yun Lei, Nicolas Scheffer, Luciana Ferrer, and Mitchell McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 1695–1699.

[7] Zhizheng Wu, Tomi Kinnunen, Nicholas Evans, Junichi Yamagishi, Cemal Hanilçi, Md Sahidullah, and Aleksandr Sizov, "Asvspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[8] Tomi Kinnunen, Md Sahidullah, Héctor Delgado, Massimiliano Todisco, Nicholas Evans, Junichi Yamagishi, and Kong Aik Lee, "The asvspoof 2017 challenge: Assessing the limits of replay spoofing attack detection," 2017.

[9] Massimiliano Todisco, Xin Wang, Ville Vestman, Md Sahidullah, Hector Delgado, Andreas Nautsch, Junichi Yamagishi, Nicholas Evans, Tomi Kinnunen, and Kong Aik Lee, "Asvspoof 2019: Future horizons in spoofed and fake audio detection," *arXiv preprint arXiv:1904.05441*, 2019.

[10] Alejandro Gomez-Alanis, Antonio M Peinado, Jose A Gonzalez, and Angel M Gomez, "A gated recurrent convolutional neural network for robust spoofing detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 12, pp. 1985–1999, 2019.

[11] Hossein Zeinali, Themos Stafylakis, Georgia Athanasopoulou, Johan Rohdin, Ioannis Gkinis, Lukáš Burget, Jan Černockỳ, et al., "Detecting spoofing attacks using vgg and sincnet: but-omilia submission to asvspoof 2019 challenge," *arXiv preprint arXiv:1907.12908*, 2019.

[12] Cheng-I Lai, Nanxin Chen, Jesús Villalba, and Najim Dehak, "Assert: Anti-spoofing with squeeze-excitation and residual networks," *arXiv preprint arXiv:1904.01120*, 2019.

[13] Galina Lavrentyeva, Sergey Novoselov, Andzhukaev Tseren, Marina Volkova, Artem Gorlanov, and Alexandr Kozlov, "Stc antispoofing systems for the asvspoof2019 challenge," *arXiv preprint arXiv:1904.05576*, 2019.

[14] Rohan Kumar Das, Jichen Yang, and Haizhou Li, "Long range acoustic features for spoofed speech detection," in *20th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2019.

[15] 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.

[16] Songxiang Liu, Haibin Wu, Hung-yi Lee, and Helen Meng, "Adversarial attacks on spoofing countermeasure of automatic speaker verification," unpublished.

[17] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[18] Weilin Xu, David Evans, and Yanjun Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," *arXiv preprint arXiv:1704.01155*, 2017.

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

[20] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel, "Ensemble adversarial training: Attacks and defenses," *arXiv preprint arXiv:1705.07204*, 2017.