ADVERSARIAL ATTACKS ON SPOOFING COUNTERMEASURES OF AUTOMATIC SPEAKER VERIFICATION

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

[1]Human-Computer Communications Laboratory, The Chinese University of Hong Kong
[2] Speech Processing and Machine Learning Laboratory, National Taiwan University

ABSTRACT

High-performance spoofing countermeasure systems for automatic speaker verification (ASV) have been proposed in the ASVspoof 2019 challenge. However, the robustness of such systems under adversarial attacks has not been studied yet. In this paper, we investigate the vulnerability of spoofing countermeasures for ASV under both white-box and black-box adversarial attacks with the fast gradient sign method (FGSM) and the projected gradient descent (PGD) method. We implement high-performing countermeasure models in the ASVspoof 2019 challenge and conduct adversarial attacks on them. We compare performance of black-box attacks across spoofing countermeasure models with different network architectures and different amount of model parameters. The experimental results show that all implemented countermeasure models are vulnerable to FGSM and PGD attacks under the scenario of white-box attack. The more dangerous black-box attacks also prove to be effective by the experimental results.

Index Terms— Adversarial attack, anti-spoofing, spoofing countermeasure, white-box attack, black-box attack

1. INTRODUCTION

Automatic speaker verification (ASV) aim to confirm that a given utterance is pronounced by a specified speaker. It is now a mature technology for biometric authentication [1–8]. Modern speaker verification systems harness the combination of several modules to tackle the problem of ASV. In [5], for example, Gaussian mixture models (GMMs) are trained to model the acoustic features and likelihood ratio is used for scoring. Recently, ASV systems based on deep learning models require fewer concepts and heuristics compared to traditional speaker verification systems and have achieved considerable performance improvement. Heigold et al. [6] proposed an integrated model with end-to-end style which directly learns a mapping from a test utterance and a few reference utterances to a verification score, resulting in compact structure and sufficiently good performance. However, past research has shown that ASV systems are vulnerable to malicious attacks using spoofing and fake audios, such as synthetic, converted and replayed speech.

Anti-spoofing countermeasures for speaker verification systems arouse keen interests and several novel studies have been done [9–19]. The ASVspoof 2019 challenge [20], a community-led challenge, attracts more than 60 international industrial and academic teams to investigate spoofing countermeasures for ASV. Both the scenarios of logical access (LA) and physical access (PA) are taken into account. The LA scenario involves fake audios synthesized by modern text-to-speech synthesis (TTS) and voice conversion (VC) models. The PA scenario involves replayed audio recorded in reverberant environment under different acoustic and replay configurations. Several teams achieve excellent performance in detecting spoofing and reinforcing robustness of ASV systems under both LA and PA scenarios [20]. Yet according to [21, 22], machine learning models with impressive performance can be vulnerable to adversarial attacks [23]. Are the spoofing countermeasures for ASV in [20] robust enough to defend against adversarial examples?

An adversarial example $\tilde{x}$ is generated by the combination of a tiny perturbation and a normal instance x. It is very similar to the original normal instance x and may even be visually or acoustically indistinguishable to human, but will lead the neural networks to incorrectly classify it as any target t given a specific perturbation. Szegedy et al. [21] claim that well-trained neural network for image classification can succumb to adversarial attacks. The vulnerability of automatic speech recognition (ASR) neural network model under adversarial attacks is proved in [22], where an adversarial example can be transcribed as any phrase. However, to our best knowledge, the robustness of spoofing countermeasure systems for ASV under the adversarial attacks has not been studied yet. In this paper, we implement several high-performance spoofing countermeasure models in the ASVspoof 2019 challenge and assess the reliability of these models under the attack of adversarial examples.

Adversarial attacks contains two main scenarios: white-box attack and black-box attack. White-box attacks are those where the adversary requires knowledge of the target model internals and adversarial examples are generated by an optimization strategy applied to the input space while fixing the model's parameters. Black-box attacks, such as [24], have no access to target model internals, only to its inputs and outputs. With the paired data acquired from available training data or by testing the online target model, a substitute for the target model can be trained. Then adversarial examples are easily crafted by the substitute and then used to attack the target model. The successful attack in the black-box scenarios, to some extent, guarantees the success of white-box attack because black-box attack requires less information and is more difficult than white-box attack. In this paper, for the sake of completeness, both white-box attacks and black box attacks are adopted to assess the reliability of spoofing countermeasure systems for ASV. There are a lot of adversarial attack approaches [23, 25–28]. In this paper, we adopt the fast gradient sign method (FGSM) [23] and the projected gradient descent (PGD) method [27].

This paper is the first one investigating the vulnerability of spoofing countermeasures for ASV under both white-box and black-box adversarial attacks. We compare performance of black-box attacks across spoofing countermeasure models with different network architectures and different number of parameters. We implement countermeasure models which achieve comparable or even better anti-spoofing performance than some high-performance models in the ASVspoof 2019 challenge and we successfully attack them under both white-box and black-box attack scenarios. All our codes will be made open-source [1].

The paper is organized as follows. Section 2 provides the detailed description of two ASV spoofing countermeasure models. In section 3, we introduce the procedure of adversarial audio generation with two adversarial attack methods, i.e., the FGSM and the PGD method. In section 4, we report the experimental setups. The experiment results and discussion are given in section 5. Finally, we conclude this paper in section 6.

2. ASV SPOOFING COUNTERMEASURE MODELS

Up to the submission time of this paper, only a few top-ranking systems in the ASVspoof 2019 challenge have accessible and complete model description. We choose two kinds of models, proposed by team T44 and team T45, to conduct adversarial attack experiments. According to the results of the ASVspoof 2019 challenge, the overall best performing single system for the LA scenario is proposed by team T45. The authors adopt an angular margin based softmax (A-softmax) [29] loss rather than traditional softmax with cross-entropy loss to train a Light CNN (LCNN) architecture [30]. The system proposed by team T44 is ranked 3-rd and 14-th places for the PA and LA scenarios, respectively. Team T44 adopts a Squeeze-Excitation extended ResNet (SENet) as one of models in their submitted system. We provide brief description of these two kinds of models in the next two subsections.

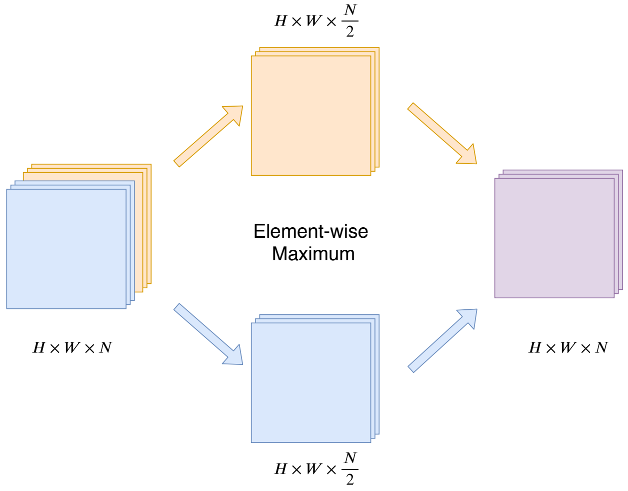

$H \times W \times \frac{N}{2}$
Element-wise Maximum
$H \times W \times N$
$H \times W \times N$
$H \times W \times \frac{N}{2}$

Fig. 1. The Max-Feature-Map (MFM) activation function for convolution layer.

2.1. LCNN model and A-Softmax

The LCNN architecture is successfully adopted for replay attacks detection in [16] and outperformed other proposed systems in ASVspoof 2017 challenge [31]. The LCNN architecture adopt the Max-Feature-Map (MFM) activation based on the Max-Out activation function. The MFM activation function for a convolutional layer is illustrated in Fig. 1, which is defined as

$$\begin{aligned} &y^k_{i,j} = \max(x^k_{i,j}, x^{k+\frac{N}{2}}_{i,j}), \\ &\forall i = \overline{1,H}, j = \overline{1,W}, k = \overline{1,N/2}, \end{aligned} \tag{1}$$

where x is the input feature map of size $H \times W \times N$, y is the output feature map of size $H \times W \times \frac{N}{2}$.

To enhance the anti-spoofing performance of LCNN architecture, team T44 utilizes an A-Softmax loss to train the model. A-Softmax enables the model to learn angularly discriminative features. Geometrically, A-Softmax loss can be viewed as imposing discriminative constraints on a hypersphere manifold. A-Softmax is represented as:

$$L_{ang} = \frac{1}{N}\sum_i -\log\left(\frac{e^{||x_i||\cos(m\theta_{y_i,i})}}{e^{||x_i||\cos(m\theta_{y_i,i})} + \sum_{j\neq y_i} e^{||x_i||\cos(\theta_{y_j,i})}}\right), \tag{2}$$

where N is the number of training samples, $\{x_i\}_{i=1}^N$ and their labels $\{y_i\}_{i=1}^N$ are training pairs, $\theta_{y_i,i}$ is the angle between x_i and the corresponding column y_i of weights W in the fully connected classification layer, and m is an integer that controls the size of an angular margin between classes.

[1] Codes are available at `https://github.com/ano-demo/AdvAttacksASVspoof`

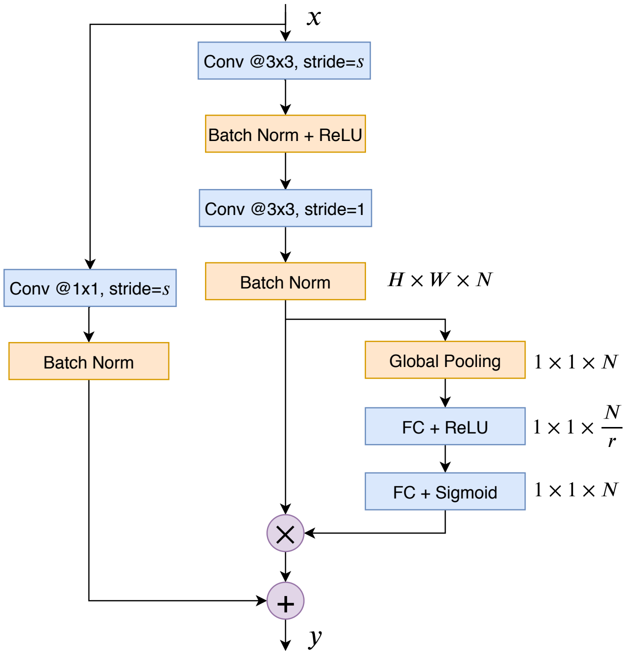


Fig. 2. The Squeeze-Excitation network (SENet) module, where s is a customized variable specifying the stride and r is the reduction factor.

2.2. Squeeze-Excitation ResNet model

Squeeze-Excitation network (SENet) adaptively recalibrates channel-wise feature responses by explicitly modelling dependencies between channels, which has shown great merits in image classification task [32]. In this paper, we implemented a model with network architecture similar to that of SENet34 in [33]. The overall architecture of an SENet moduel is illustrated in Fig. 2.

3. ADVERSARIAL AUDIO GENERATION

To execute an adversarial attack, we consider the model parameters θ as fixed and optimize over the input space. Specifically, in this paper, an adversarial example is a perturbed version of the input spectral feature x.

$$\tilde{x} = x + \delta, \tag{3}$$

where δ is small enough such that the reconstructed speech signal of the perturbed version $\tilde{x}$ is perceptually indistinguishable from the original signal x by humans, but causes the network to make incorrect decision. Searching for a suitable δ can be formulated as solving the following optimization problem:

$$\max_{\delta \in \Delta} \text{Loss}(\theta, x + \delta, y_x), \tag{4}$$

where $\text{Loss}(\cdot)$ is the loss function, y_x is the label of x and Δ is a set of allowed perturbation that formalizes the manipulative power of the adversary. In this paper, Δ is a small l_∞-norm ball, that is, $\Delta = \{\delta | \, ||\delta||_\infty \le \epsilon\}$, $\epsilon \ge 0 \in \mathbb{R}$.

To solve the optimization problem (4), in this paper, we adopt the fast gradient sign method (FGSM) [23] and the projected gradient descent (PGD) method [27].

3.1. Fast gradient sign method

The first adversarial attack method, shorted as the FGSM method, consists of taking a single step along the direction of the gradient, i.e.,

$$\delta = \epsilon \cdot \text{sign}(\nabla_x \text{Loss}(\theta, x, y_x)), \tag{5}$$

where the $\text{sign}(\cdot)$ function simply takes the sign of the gradient $\nabla_x \text{Loss}(\theta, x, y_x)$. Therefore, given an utterance x, the adversarial spectral feature can be simply computed as $\tilde{x} = x + \epsilon \cdot \text{sign}(\nabla_x \text{Loss}(\theta, x, y_x))$. While the FGSM benefits from being the simplest adversarial attack method, it is often relatively inefficient at solving the maximization problem (4).

3.2. Projected gradient descent method

Unlike the FGSM, which is a single-step method, the PGD method is an iterative method. Starting from the original input $x_0 = x$, the input is iteratively updated as follows:

$$x_{k+1} = \text{clip}(x_k + \alpha \cdot \text{sign}(\nabla_{x_k} \text{Loss}(\theta, x_k, y_{x_k}))), \quad \text{for } k = 0, ..., K-1, \tag{6}$$

where α is the step size, K is the number of iterations and the $\text{clip}(\cdot)$ function applies element-wise clipping such that $||x_k - x||_\infty \le \epsilon$, $\epsilon \ge 0 \in \mathbb{R}$. We take x_K as the final perturbed spectral feature. Intuitively, the PGD method can be thought of as iteratively applying small-step FGSM, but forcing the perturbed input stay within the admissible set Δ at every step. The PGD method allows for more effective attacks but is naturally more computationally expensive than the FGSM.

The performance of PGD is still limited by the possibility of sticking at local optima of the loss function. To mitigate this problem, random restarts is incorporated into the PGD method [27]. The PGD method with random restarts will be executed multiple runs. The initial location of the adversarial example is randomly selected within the admissible perturbation set Δ and the PGD method will be executed a certain number of times in one run. The final adversarial example is the one resulting in maximum loss.

4. EXPERIMENTAL SETUPS

This paper uses the ASVspoof 2019 dataset, which encompasses partitions for the assessment of LA and PA scenarios. In this paper, we only utilize the LA partition, which are themselves partitioned into training, development and evaluation sets. We use raw log power magnitude spectrum computed from the signal as acoustic features. Following [34], FFT spectrum is extracted with the Blackman window having size of 1724 and step-size of 0.0081s. Only the first 600 frames for each utterance are used as input for all trained models. No additional preprocessing techniques such as voice activity detection (VAD), pre-emphasis or dereverberation are adopted.

The adversarial attacks are conducted as the following: We train a spoofing countermeasure model using the training set. Hyper-parameters of the countermeasure model are tuned using the development set. We evaluate the anti-spoofing performance and generate adversarial examples using the evaluation set. When generating adversarial examples, we add adversarial perturbation into the acoustic feature vectors using the FGSM or the PGD method introduced in Section 3. The perturbed acoustic features are reconstructed into waveform to attack the well trained countermeasure model.

4.1. Details of countermeasure model implementation

Three countermeasure models are trained, which we term as LCNN-big, LCNN-small and SENet12. LCNN-big has the same network architecture as in [34]. LCNN-small has similar network architecture to LCNN-big, but with less parameters. SENet12 has similar network architecture as in [33] but with less parameters. The number of parameters of these three models are shown as in Table 1. LCNN-big model has larger model capacity than LCNN-small and SENet12 model in terms of number of model parameters, while LCNN-small model and SENet12 model have equal model capacity. The detailed network architecture of LCNN-small and SENet12 model are shown in Table 2 and Table 3, respectively.

LCNN-big and LCNN-small model are trained using the A-Softmax loss function, while SENet12 is trained using the original softmax and cross-entropy loss. The constant m in Eq.(2) is set to 4. To mitigate the overfitting issue, a dropout rate of 0.75 is used when training LCNN models. We found that adding a relatively large L2 regularization is helpful to mitigate the overfitting issue. We set the weight decay rate at 0.001 when training all these three models. We use Adam optimizer with a constant learning rate of 0.001 to update the model parameters in all training cases, where $\beta_1 = 0.9$ and $\beta_2 = 0.999$.

During training stage, we applied early stopping according to the classification accuracy on the development set. During inference stage, we took the cosine similarity between the input and the weight vector in the last FC layer corresponding to the bonafide class as the score of the utterance.

Table 1. Number of parameters of LCNN-Big, LCNN-Small and SENet12 model.

Model	LCNN-big	LCNN-small	SENet12
Num. of parametes	10M	0.51M	0.48M

Table 2. LCNN-small network architecture.

Type	Filter / Stride	Output
Conv_1	5×5 / 1×1	$863 \times 600 \times 16$
MFM_2	–	$863 \times 600 \times 8$
MaxPool_3	$2 \times 2/2 \times 2$	$431 \times 300 \times 8$
Conv_4	$1 \times 1/1 \times 1$	$431 \times 300 \times 16$
MFM_5	–	$431 \times 300 \times 8$
BatchNorm_6	–	$431 \times 300 \times 8$
Conv_7	$3 \times 3/1 \times 1$	$431 \times 300 \times 24$
MFM_8	–	$431 \times 300 \times 12$
MaxPool_9	$2 \times 2/2 \times 2$	$215 \times 150 \times 12$
BatchNorm_10	–	$215 \times 150 \times 12$
Conv_11	$1 \times 1/1 \times 1$	$215 \times 150 \times 24$
MFM_12	–	$215 \times 150 \times 12$
BatchNorm_13	–	$215 \times 150 \times 12$
Conv_14	$3 \times 3/1 \times 1$	$215 \times 150 \times 24$
MFM_15	–	$215 \times 150 \times 12$
MaxPool_16	$2 \times 2/2 \times 2$	$107 \times 75 \times 12$
Conv_17	$1 \times 1/1 \times 1$	$107 \times 75 \times 24$
MFM_18	–	$107 \times 75 \times 12$
BatchNorm_19	–	$107 \times 75 \times 12$
Conv_20	$3 \times 3/1 \times 1$	$107 \times 75 \times 8$
MFM_21	–	$107 \times 75 \times 4$
BatchNorm_22	–	$107 \times 75 \times 4$
Conv_23	$1 \times 1/1 \times 1$	$107 \times 75 \times 8$
MFM_24	–	$107 \times 75 \times 4$
BatchNorm_25	–	$107 \times 75 \times 4$
Conv_26	$3 \times 3/1 \times 1$	$107 \times 75 \times 8$
MFM_27	–	$107 \times 75 \times 4$
MaxPool_28	$2 \times 2/2 \times 2$	$53 \times 37 \times 4$
FC_29	–	64
MFM_30	–	32
BatchNorm_31	–	32
FC_32	–	2

4.2. Adversarial attacks

We apply both white-box and balck-box adversarial attacks on the trained countermeasure models with the FGSM and the PGD method. We investigate adversarial attacks under various levels of manipulative power of the adversary on the audios. In both the FGSM and PGD attack settings, ϵ in Eq.(5) and Eq.(6) is chosen from the set of $\{0.1, 1, 5\}$. To make the level of manipulative power of the PGD attack consistent with the FGSM attack, we make the step-size α and ϵ in the PGD attack scenario satisfy the relationship of

$$\epsilon = \text{number of iterations} \times \alpha. \tag{7}$$

Table 3. SENet12 network architecture.

Type	Filter / Stride	Output
Conv	7×7 / 2×2	$431 \times 300 \times 16$
BatchNorm	–	$431 \times 300 \times 16$
ReLU	–	$431 \times 300 \times 16$
MaxPool	$3 \times 3/2 \times 2$	$215 \times 150 \times 16$
SEResNet Module$\times$1	–	$215 \times 150 \times 16$
SEResNet Module$\times$2	–	$107 \times 75 \times 32$
SEResNet Module$\times$3	–	$53 \times 37 \times 64$
SEResNet Module$\times$1	–	$26 \times 18 \times 128$
Global AvgPool	–	128
FC	–	2

For example, if $\epsilon = 1$ and number of iterations is 10, then α is set to be 0.1. The number of random restarts is 5 in all experiments.

4.3. XAB listening test

To achieve a valid adversarial attack, it is important to make the adversarial audio examples sound indistinguishable from the original audio signals by human ears. We conduct an XAB listensing test, which is a standard way to assess the detectable differences between two choices of sensory stimuli. The adversarial audio signals are generated from the LCNN-big using the PGD method with $\epsilon = 5$. Each of the adversarial audio signal is reconstructed from the perturbed log power magnitude spectrum and the phase spectrum of its corresponding original audio signal. We presented to listeners 50 randomly chosen adversarial-original audio pairs (i.e., A and B), from each of which we randomly choose one as the reference audio (i.e., X). Five listeners take part in the XAB listening test, where they are asked to choose from A and B one audio which sounds more like the reference audio X.

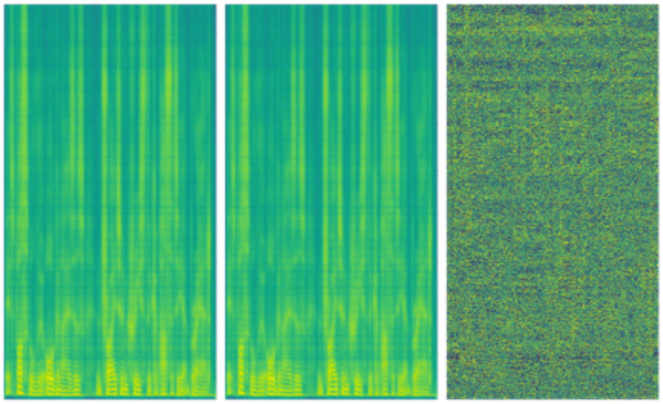
5. RESULTS AND DISCUSSION

5.1. Countermeasure performance

Anti-spoofing performance of the three countermeasure models, LCNN-Big, LCNN-Small and SENet12, is evaluated via the minimum normalized tandem detection cost function (t-DCF) and the equal error rate (EER), as shown in Table 4. Note that the LCNN-Big model achieves comparable performance with that reported in [34], and the SENet12 model has even better performance than the best performing single model reported in [33].

5.2. Adversarial attack results

The subjective XAB listening test in subsection 4.3 results in average classification accuracy of 48.4%, which confirms the validity of the adversarial audio attacks. Spectrograms of original audio, adversarial audio and adversarial perturbation of the utterance LA_E_1001227 is shown in Fig. 3, where the attack is conducted to the LCNN-big model using the PGD method, with $\epsilon = 5$, number of iterations = 10 and number of random restarts = 5. We can see that the adversarial perturbed spectrogram is almost visually indistinguishable from that of the original audio signal. If we set the EER point of the evaluation set as the operating point, the utterance LA_E_1001227 is wrongly classified as "bona fide" in this setting. In the following subsections, we use the EER as metric to evaluate the performance of adversarial attacks by the means of the FGSM or the PGD method, under both white-box and black-box scenarios.



(a) Reference. (b) Adv. example. (c) Adv. perturbation.

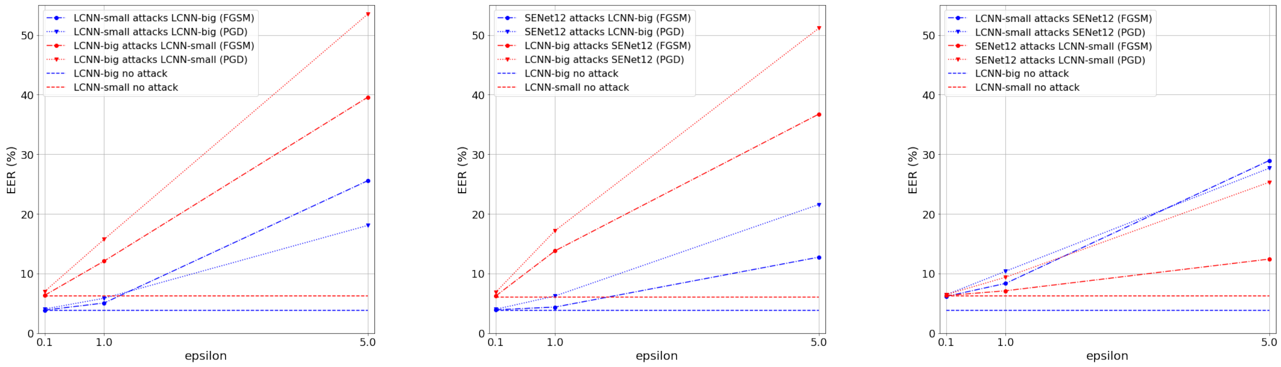
Fig. 3. Sub-figures (a)-(c) are spectrograms of original audio, adversarial audio and adversarial perturbation of the utterance LA_E_1001227. The attack is conducted to the LCNN-big model using the PGD method, where $\epsilon = 5$, number of iterations is 10 and number of random restarts is 5.

Table 4. Anti-spoofing performance of the three countermeasure models, LCNN-big, LCNN-small and SENet12.

Model	Dev		Eval	
	t-DCF$^{min}_{norm}$	EER(%)	t-DCF$^{min}_{norm}$	EER(%)
LCNN-big	0.0010	0.047	0.1052	3.875
LCNN-small	0	0.002	0.1577	6.226
SENet12	0	0	0.1737	6.077

Table 5. White-box attack performance of the FGSM and the PGD method.

EER(%)	$\epsilon = 0.1$		$\epsilon = 1$		$\epsilon = 5$	
	FGSM	PGD	FGSM	PGD	FGSM	PGD
LCNN-big	4.691	**6.256**	36.504	**54.382**	48.457	**93.119**
LCNN-small	7.613	**17.419**	34.670	**73.649**	48.375	**89.845**
SENet12	7.737	**13.896**	24.936	**62.681**	51.626	**87.220**



(a) LCNN-big and LCNN-small.

(b) LCNN-big and SENet12.

(c) LCNN-small and SENet12.

Fig. 4. Black-box attack performance of the FGSM and the PGD method.

5.2.1. *White-box attacks*

The white-box attack performance of the FGSM and the PGD method using different $\epsilon' s$ is shown in Table 5. The EERs of all three countermeasure models increase as ϵ grows. PGD attacks attain larger EERs than FGSM attacks in all three countermeasure models and all the settings of ϵ under the white-box attack scenario. The EERs of all three models under the FGSM attacks reach near 50% when $\epsilon = 5$. As for PGD, the EERs are greater than 50% when $\epsilon = 1$, and are greater than 85% when $\epsilon = 5$, which will result in reversed classification decision if the operating point is pre-defined by the EER point on the evaluation set. We can conclude from the white-box attack results that the reliability of all three countermeasure models are challenged and broken down by FGSM or PGD attacks under the scenario of white-box attack. The PGD method is more effective than the FGSM. Research on more advanced countermeasure models should be done to keep pace with today's white-box adversarial attacks.

5.2.2. *Black-box attacks*

In this part, we study adversarial attacks of the FGSM and the PGD method from the perspective of black-box scenario. As shown in Fig 4, the black-box attacks achieve a resounding success in all the mutual attack settings: LCNN-big with LCNN-small, LCNN-big with SENet12 and LCNN-small with SENet12. In a large fraction of attacking scenarios, the attack method of PGD attains larger EER than the FGSM method, leading to that the PGD method tends to generate more powerful adversarial examples. According to Fig. 4(a) and 4(b), adversarial examples generated by LCNN-big with rather more parameters are more powerful as they can attack smaller models, LCNN-small and SENet12, with larger EERs, while adversarial examples generated by LCNN-small and SENet12 fail to attack LCNN-big with such large EERs. So in our experimental setup, we can safely conclude adversarial examples from small models are outperformed by large models in terms of the performance of attack for both the FGSM and PGD methods. According to the red dotted line with triangle marker and red dash-dot line with circle marker in Fig. 4(a) and Fig. 4(b), the adversarial examples generated by LCNN-big attain greater EER when attack LCNN-small rather than RESNet12, resulting in a conclusion that adversarial attacks are much easier realized under similar model structure. According to Fig 4(c), the attack efficacy of adversarial examples generated by LCNN-small outperforms the adversarial examples from SENet12.

6. CONCLUSIONS

In this paper, we have investigated the vulnerability of spoofing countermeasures for ASV under both white-box and black-box adversarial attacks using the FGSM and PGD method. We have also compared performance of black-box attacks across spoofing countermeasure models with different network architectures and different amount of parameters. We implement three countermeasure models, i.e., LCNN-big, LCNN-small and SENet12, and conduct adversarial attacks on them. The experimental results show that all three models are subject to FGSM and PGD attacks under the scenario of white-box attack. The more dangerous black-box attacks are also proved to be effective by the experimental results. For the future work, we would like to adopt adversarial training methods to improve the robustness of countermeasure models and make them less vulnerable to adversarial attacks.

7. ACKNOWLEDGEMENTS

Songxiang Liu was 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 (Project No. 108-2636-E-002-001).

References

[1] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, “X-vectors: Robust dnn embeddings for speaker recognition,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] Patrick Kenny, Vishwa Gupta, Themos Stafylakis, Pierre Ouellet, and Jahangir Alam, “Deep neural networks for extracting baum-welch statistics for speaker recognition,” in *Proc. Odyssey*, 2014, pp. 293–298.

[9] Nicholas WD Evans, Tomi Kinnunen, and Junichi Yamagishi, “Spoofing and countermeasures for automatic speaker verification.,” in *Interspeech*, 2013, pp. 925–929.

[10] Massimiliano Todisco, Hctor Delgado, and Nicholas Evans, “A new feature for automatic speaker verification anti-spoofing: Constant q cepstral coefficients,” *Odyssey 2016*, Jun 2016.

[11] Giacomo Valenti, Hctor Delgado, Massimiliano Todisco, Nicholas Evans, and Laurent Pilati, “An end-to-end spoofing countermeasure for automatic speaker verification using evolving recurrent neural networks,” *Odyssey 2018 The Speaker and Language Recognition Workshop*, Jun 2018.

[12] Héctor Delgado, Massimiliano Todisco, Md Sahidullah, Nicholas Evans, Tomi Kinnunen, Kong Lee, and Junichi Yamagishi, “Asvspoof 2017 version 2.0: metadata analysis and baseline enhancements,” in *Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018.

[13] Zhizheng Wu, Tomi Kinnunen, Eng Siong Chng, Haizhou Li, and Eliathamby Ambikairajah, “A study on spoofing attack in state-of-the-art speaker verification: the telephone speech case,” in *Proceedings of The 2012 Asia Pacific Signal and Information Processing Association Annual Summit and Conference*. IEEE, 2012, pp. 1–5.

[14] Elie Khoury, Tomi Kinnunen, Aleksandr Sizov, Zhizheng Wu, and Sébastien Marcel, “Introducing i-vectors for joint anti-spoofing and speaker verification,” in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[15] Nanxin Chen, Yanmin Qian, Heinrich Dinkel, Bo Chen, and Kai Yu, “Robust deep feature for spoofing detectionthe sjtu system for asvspoof 2015 challenge,” in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[16] Galina Lavrentyeva, Sergey Novoselov, Egor Malykh, Alexander Kozlov, Oleg Kudashev, and Vadim Shchemelinin, “Audio replay attack detection with deep learning frameworks.,” in *Interspeech*, 2017, pp. 82–86.

[17] Zhuxin Chen, Zhifeng Xie, Weibin Zhang, and Xiangmin Xu, “Resnet and model fusion for automatic spoofing detection.,” in *INTERSPEECH*, 2017, pp. 102–106.

[18] Yanmin Qian, Nanxin Chen, and Kai Yu, “Deep features for automatic spoofing detection,” *Speech Communication*, vol. 85, pp. 43–52, 2016.

[19] Cheng-I Lai, Alberto Abad, Korin Richmond, Junichi Yamagishi, Najim Dehak, and Simon King, “Attentive filtering networks for audio replay attack detection,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6316–6320.

[20] 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.

[21] 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.

[22] Nicholas Carlini and David Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2018, pp. 1–7.

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

[24] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami, "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia conference on computer and communications security*. ACM, 2017, pp. 506–519.

[25] Nicholas Carlini and David Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 39–57.

[26] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2574–2582.

[27] 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.

[28] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, 2019.

[29] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song, "Sphereface: Deep hypersphere embedding for face recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 212–220.

[30] Xiang Wu, Ran He, Zhenan Sun, and Tieniu Tan, "A light cnn for deep face representation with noisy labels," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2884–2896, 2018.

[31] 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.

[32] Jie Hu, Li Shen, and Gang Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[33] 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.

[34] 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.