Masking Speech Feature to Detect Adversarial Examples for Speaker Verification

Xing Chen, Jiadi Yao and Xiao-Lei Zhang †
† School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an, China
E-mail: {xing.chen, yaojiadi}@mail.nwpu.edu.cn, xiaolei.zhang@nwpu.edu.cn

***Abstract*—Adversarial examples of speaker verification (SV) systems are the clean audio recordings added with imperceptible perturbation. They are generated to manipulate the decision of SV, which poses a serious threat to the security of SV. Therefore, many adversarial example detection methods have been proposed to defend against such adversarial attacks. However, existing methods either require additional training of detection models or are time-consuming. In this paper, we propose a non-training and effective method to detect adversarial examples. It simply masks the parts of the input speech features (e.g. LogFBank) that contain less speaker information. The masked parts will inevitably have a small impact on genuine examples, and large impact on adversarial examples. Therefore, the adversarial examples can be detected by analyzing the absolute alteration of scores before and after masking. Experimental results on ResNet34 showed that our method outperforms the training-dependent Parallel-Wave-GAN baseline, and only consumes 1/10 of the detection time of the baseline.**

I. Introduction

Speaker verification (SV) is a task of determining whether two utterances are from the same speaker [1]. With the widespread application of SV, such as authentication, bank transaction, and forensics, its security has become increasingly important. However, the adversarial attacks proposed in recent years can defeat a SV system at a very high signal-to-noise ratio (SNR) [2], which brings great challenges to the application of SV.

Adversarial attack refers to adding imperceptible perturbation to the input test utterance of a SV system, which leads the SV system to an expected wrong decision of the attacker. The goal of the attacker has the following two situations. The first situation classifies non-target trials as targets, which is called impersonation attack. The second situation classifies target trials as non-targets, which is called evasion attack. We call the perturbed test speech as adversarial examples, which has been previously studied in speech processing systems such as automatic speech recognition and SV. For example, Villalba *et al.* [2] found that the state-of-the-art (SOTA) SV models are vulnerable to adversarial examples in white-box scenarios where the attacker knows the model structure and parameters, even in high SNR levels of 30-60 dB [2]. Kreuk *et al.* [3] studied the vulnerability of SV against transferable adversarial attacks under the condition of cross-datasets and cross-features. It indicated that the adversarial examples are still effective in black box scenarios where the attacker can only query the decision result of the SV system. In addition, many works have contributed to exploring robust adversarial examples in the aspects of universality [4, 5], realistic scenarios [5, 6], and imperceptibility [7].

In order to defend against adversarial attacks, a number of countermeasures including proactive defense and reactive defense have been proposed. Proactive defense approaches need to modify the SV model, which is inconvenient to deploy. For example, Wang *et al.* [8] added an adversarial regularization loss term to improve the robustness of the model against adversarial examples. Reactive defense approaches which are subdivided into mitigation-based and detection-based, can be deployed directly in front of the SV system. For example, Zhang *et al.* [9] proposed an adversarial separation network to restore the clean speech. Wu *et al.* [10] used a cascaded self-supervised module as a filter to remove adversarial noise. Note that the works in [9, 10] are all mitigation-based defense methods. In addition, Li *et al.* [11] trained a VGG-like binary detector to detect adversarial examples. Wu *et al.* [12] proposed to detect adversarial examples by the absolute discrepancy score before and after the phase reconstruction of the spectrograms, using either the Griffin-Lim algorithm or the Parallel-Wave-GAN (PWG) model. Villalba *et al.* [13] used an x-vector model to train an embedding feature extractor, and detected adversarial examples by comparing the similarity of the embedding features. Note that the works in [11–13] are all detection-based defense methods.

However, we found that existing detection-based defense methods on SV either require additional training or are time-consuming. To address the shortcomings simultaneously, we intend to directly utilize the fragility of adversarial examples after feature-level transformation. In this paper, we propose to detect adversarial examples by Masking the input LogFBank at High frequencies (MLFB-H) or Masking the input LogFBank using one-order Difference (MLFB-D). The proposed methods are training-independent and have a fast detection speed. Specifically, we assume that the adversarial perturbation is evenly added to the input LogFBank. The foundation of the proposed methods is that masking the parts of the LogFBank feature that contain less speaker information has a small impact on genuine examples and a large impact on adversarial examples respectively. Hence, we can detect adversarial examples by comparing the score variation of the

978-616-590-477-3

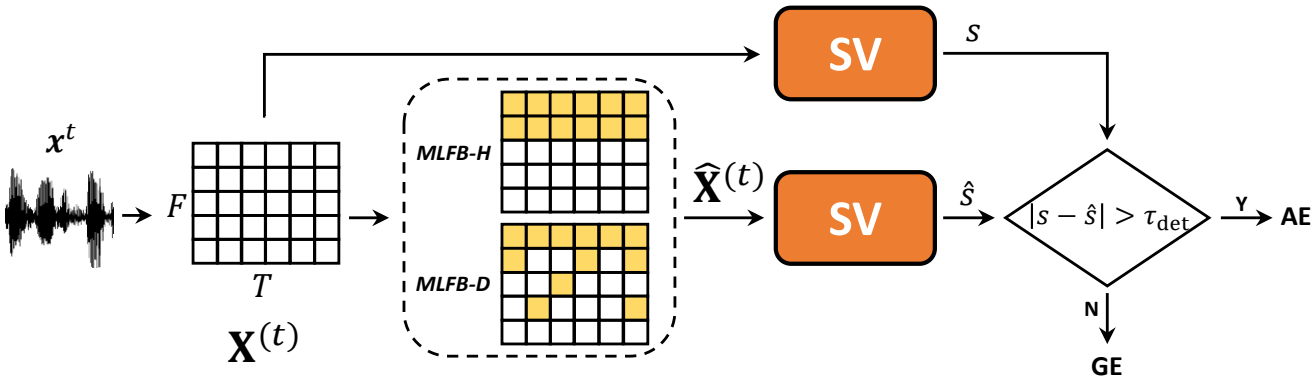


Fig. 1: Pipeline of the proposed detection methods. The symbols $\mathbf{X}^{(t)}$ and $\hat{\mathbf{X}}^{(t)}$ denote the original and masked acoustic features (e.g. LogFBank) respectively. The SV score variation $|s - \hat{s}|$ after the masking operation is used to distinguish adversarial examples (AE) and genuine examples (GE).

examples before and after the masking operation, where the score is the cosine similarity between the test utterance and the enrollment utterance. We evaluate the effectiveness of the proposed methods in scenarios that adversarial examples of multiple intensities are mixed on two SOTA SV models. Experimental results show that the proposed methods achieve a comparable detection equal error rate (EER) with the PWG baseline, with an empirical computational complexity of only 1/10 of the latter.

II. Preliminaries

A. Speaker Verification

SV aims at verifying whether an utterance is pronounced by a hypothesized speaker. In the test stage of SV, given an enrollment utterance $\boldsymbol{x}^e$ and a test utterance $\boldsymbol{x}^t$, we first transform the utterances into acoustic features (e.g. LogFBank or MFCC), denoted as $\mathbf{X}^{(e)}$ and $\mathbf{X}^{(t)}$ respectively. Then, an SV model, which consists of a frame-level feature extractor, a pooling layer, and a segment-level feature extractor from bottom-up, extracts speaker embeddings from the acoustic features. Finally, we determine whether the two utterances come from the same speaker by comparing the similarity of their speaker embeddings with a predefined threshold, which is formulated as follows:

$$s = S_\theta\Big(f\left(\boldsymbol{x}^e\right), f\left(\boldsymbol{x}^t\right)\Big) \underset{H_0}{\overset{H_1}{\gtrless}} \eta, \tag{1}$$

where s is the similarity of the two embeddings, $S_\theta(\cdot)$ denotes the well-trained SV model $S(\cdot)$ with parameters θ, $f(\cdot)$ is used to extract the acoustic features, and η is the predefined threshold, H_1 represents the hypothesis of $\boldsymbol{x}^e$ and $\boldsymbol{x}^t$ belonging to the same speaker, and H_0 is the diametrical hypothesis of H_1.

B. Audio Adversarial Attack

Audio adversarial examples are crafted by deliberately adding subtle noise to speech signals, which ultimately makes a speech processing system generate special errors that are expected by the attacker. In the case of white-box attacks where the attacker has complete access to all components of the victim model, many algorithms obtain the gradient of the loss with respect to the input audio for an efficient search of the adversarial noise. In this paper, we use the white-box attack algorithm *Basic Iterative Method* (BIM) [14] as the attacker. It iteratively searches for adversarial examples via the following formula:

$$\boldsymbol{x}_{n+1} = \mathrm{Clip}_{\boldsymbol{x}^t,\epsilon}\Big(\boldsymbol{x}_n + k \times \alpha\, \mathrm{sign}\big(\nabla_{\boldsymbol{x}_n} S_{\theta,f}\left(\boldsymbol{x}^e, \boldsymbol{x}_n\right)\big)\Big), \tag{2}$$

where

$$k = \begin{cases} -1, & \text{if } \boldsymbol{x}^e \text{ and } \boldsymbol{x}^t \text{ contribute to a target trial} \\ 1, & \text{if } \boldsymbol{x}^e \text{ and } \boldsymbol{x}^t \text{ contribute to a non-target trial} \end{cases}$$

and $n = 0, 1, \cdots, N$, with $N = \lceil \epsilon/\alpha \rceil$ as the number of iterations, $\lceil \cdot \rceil$ is the function of taking the upper bound, $\boldsymbol{x}_n$ is initialized by the test utterance, i.e. $\boldsymbol{x}_0 = \boldsymbol{x}^t$ (note that, $\boldsymbol{x}^t$ is not normalized), ϵ constrains the maximum magnitude of the perturbation, and $\mathrm{Clip}_{\boldsymbol{x}^t,\epsilon}(\cdot)$ denotes an element-wise clipping function which ensures $\|\boldsymbol{x}_n - \boldsymbol{x}^t\|_\infty \leq \epsilon$, k controls the direction of step, α is the step size, and the function $\mathrm{sign}(\cdot)$ only gets the direction of the gradient of the similarity score against the input $\boldsymbol{x}_n$. When the adversarial attack algorithm ends up with N iterations, an adversarial example $\tilde{\boldsymbol{x}}^t$ is found as $\boldsymbol{x}_N$.

III. Methods

A. Overview

The detection process is shown in Fig. 1. Given the test utterance $\boldsymbol{x}^t$, we first extract LogFBank feature $\mathbf{X}^{(t)} = f(\boldsymbol{x}^t)$, where $\mathbf{X}^{(t)} \in \mathbb{R}^{F \times T}$ with F and T representing the number of mel-filters and frames respectively.

Then, we calculate a mask matrix $\mathbf{M}$ to transform the speech feature $\mathbf{X}^{(t)}$ to another masked speech feature $\hat{\mathbf{X}}^{(t)}$ by:

$$\hat{\mathbf{X}}^{(t)} = \mathbf{M} \odot \mathbf{X}^{(t)} \tag{3}$$

where $\odot$ denotes the element-wise product operator. Here we propose two methods to obtain $\mathbf{M}$.

The first method, named MLFB-H, obtains the mask matrix by:

$$\mathbf{M} = \begin{bmatrix} \mathbf{1}_{(F-l)\times T} \\ \mathbf{0}_{l\times T} \end{bmatrix}, \tag{4}$$

where l is the length of masking, and the symbols $\mathbf{1}_{a\times b}$ (or $\mathbf{0}_{a\times b}$) denotes an all one (or zero) matrix with a rows and b columns.

The second method, named MLFB-D, masks the time-frequency bins whose absolute values of their one-order difference along the frequency axis is smaller than a threshold:

$$\mathbf{M}_{i,j} = \begin{cases} 1, & \text{if } \left|\mathbf{X}^{(t)}_{i+1,j} - \mathbf{X}^{(t)}_{i,j}\right| > \xi \\ 0, & \text{if } \left|\mathbf{X}^{(t)}_{i+1,j} - \mathbf{X}^{(t)}_{i,j}\right| \le \xi \end{cases}, \tag{5}$$

$$\forall\, i = 0, 1, \cdots, F-2, \quad \forall\, j = 0, 1, \cdots, T-1$$

where the subscripts i and j represent the frequency dimension and the time dimension respectively, ξ is the threshold of the masking, $\mathbf{M}_{i,j}$ is an element of the mask matrix $\mathbf{M}$. Note that, we splice an all-zero matrix $\mathbf{0}_{1\times T}$ at high frequencies to ensure that $\mathbf{M}$ and $\mathbf{X}^{(t)}$ have the same dimension, i.e., $\mathbf{M}_{F-1,j} = 0$.

After obtaining the masked speech feature $\hat{\mathbf{X}}^{(t)}$, two similarity scores are calculated by

$$s = S_\theta\Big(\mathbf{X}^{(e)}, \mathbf{X}^{(t)}\Big), \text{ and } \hat{s} = S_\theta\Big(\mathbf{X}^{(e)}, \hat{\mathbf{X}}^{(t)}\Big). \tag{6}$$

Finally, we compare the score variation $\upsilon = |s - \hat{s}|$ with a detection threshold τ_{det}. When $\upsilon > \tau_{\text{det}}$, the test utterance $\boldsymbol{x}^t$ is detected as an adversarial example, and vice versa.

B. Evaluation

We use EER and detection success rate (DSR) to evaluate the performance of the proposed method following the evaluation method in [12]. For the set of genuine examples $\mathcal{G} = \{(\boldsymbol{x}^t_i, \boldsymbol{x}^e_i) \mid i = 0, 1, \cdots, I\}$, a score variation set $\mathcal{V}_{\text{gen}}$ after masking can be obtained by:

$$\upsilon_i = \left|S_\theta\Big(f\left(\boldsymbol{x}^e_i\right), f\left(\boldsymbol{x}^t_i\right)\Big) - S_\theta\Big(f\left(\boldsymbol{x}^e_i\right), \hat{f}\left(\boldsymbol{x}^t_i\right)\Big)\right| \tag{7}$$

where $\upsilon_i \in \mathcal{V}_{\text{gen}}$ with $i = 0, 1, \cdots, I$, and $\hat{f}\left(\boldsymbol{x}^t_i\right)$ represents that the test utterance $\boldsymbol{x}^t_i$ is transformed by a cascade of the acoustic feature extractor and the masking operation. For the set of adversarial examples $\mathcal{A} = \{(\tilde{\boldsymbol{x}}^t_i, \boldsymbol{x}^e_i) \mid i = 0, 1, \cdots, I\}$, similarly, a score variation set $\mathcal{V}_{\text{adv}}$ is calculated by (7), except that $\boldsymbol{x}^t_i$ is replaced by the adversarial example $\tilde{\boldsymbol{x}}^t_i$.

The evaluation metric EER is defined by the following formulas:

$$\begin{aligned} \text{FAR}_{\text{det}}(\tau) &= \frac{|\{\upsilon_i > \tau \mid \upsilon_i \in \mathcal{V}_{\text{gen}}\}|}{|\mathcal{V}_{\text{gen}}|}, \\ \text{FRR}_{\text{det}}(\tau) &= \frac{|\{\upsilon_i \le \tau \mid \upsilon_i \in \mathcal{V}_{\text{adv}}\}|}{|\mathcal{V}_{\text{adv}}|}, \\ \text{EER}_{\text{det}} &= \text{FAR}_{\text{det}}(\tau_{\text{eer}}) = \text{FRR}_{\text{det}}(\tau_{\text{eer}}), \end{aligned} \tag{8}$$

where $\text{FAR}_{\text{det}}(\tau)$ and $\text{FRR}_{\text{det}}(\tau)$ are the false accept rate (FAR) and false reject rate (FRR) respectively of the detection given a threshold τ, $|\mathcal{S}|$ represents the number of elements in set $\mathcal{S}$. After manually given a tolerable FAR of detection, denoted as $\text{FAR}_{\text{given}}$, we define the evaluation metric DSR as:

$$\text{DSR} = \frac{|\{\upsilon_i > \tau_{\text{det}} \mid \upsilon_i \in \mathcal{V}_{\text{adv}}\}|}{|\mathcal{V}_{\text{adv}}|}, \tag{9}$$

$$\tau_{\text{det}} = \underset{\tau}{\text{argmin}} \left|\text{FAR}_{\text{det}}(\tau) - \text{FAR}_{\text{given}}\right|, \tag{10}$$

where τ_{det} is the detection threshold for $\text{FAR}_{\text{given}}$. Since the detection threshold in (10) depends only on the score variation set $\mathcal{V}_{\text{gen}}$ and $\text{FAR}_{\text{given}}$, we evaluated the DSR of detection method under the hybrid sets of adversarial examples with multiple perturbation intensities.

IV. Experimental Settings

A. Dataset

All of our experiments were conducted on the VoxCeleb dataset [15], which contains over one million utterances from more than 7,000 speakers of different ethnicities, accents, professions and ages. The two SOTA SV models were trained on VoxCeleb2, and tested on the original trials. Because it is very time-consuming to generate adversarial examples for all 37,611 trials, we randomly selected 1,000 trials from the original ones, and conducted the attack and detection experiments on the 1,000 trials.

B. SV Models

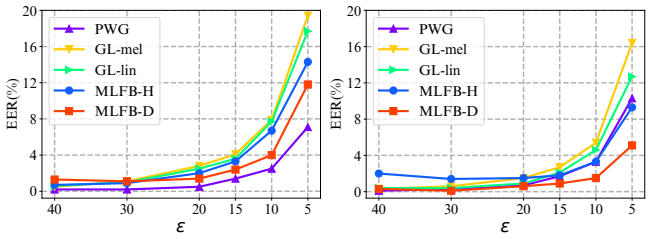
Different x-vector SV models are characterized by different network structures and pooling strategies. In this study, we used ECAPA_TDNN[1] and ResNet34[1] x-vector models, with the attentive statistical pooling and temporal statistical pooling strategy respectively. A hamming window of width 25ms and step 10ms is used to partition speech signals into frames. A 80-dimensional LogFBank followed by cepstral mean normalization were used to extract the acoustic features. Data augmentation, such as perturbing speed, superimposed disturbance and simulating reverberation were adopted in training the SV models. In addition, they all used additive angular margin softmax (AAM-Softmax) as the training loss and cosine similarity as the back-end scoring.

C. Detection Settings

We followed the settings in [12][2] to generate adversarial examples. With the step size $\alpha = 1$ fixed, we crafted adversarial examples set $\mathcal{A}_\epsilon$ for each value of perturbation constraint, where $\epsilon = 5, 10, 25, 20, 30, 40$. The genuine examples set, denoted as $\mathcal{G}_\epsilon$, was generated by adding the gaussian white noise to obtain a targeted SNR at the sentence level. In addition, the masking length $l = 8$ and the masking threshold $\xi = 0.05$ were adopted in MLFB-H and MLFB-D respectively.

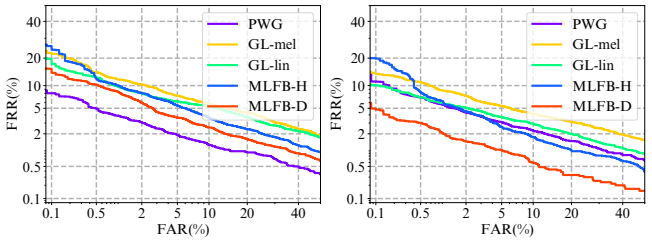
[1] https://github.com/wenet-e2e/wespeaker

[2] https://github.com/hbwu-ntu/spot-adv-by-vocoder



(a) ECAPA_TDNN (b) ResNet34

Fig. 2: EER performance of the five detection methods with different perturbation constraints ϵ.



(a) ECAPA_TDNN (b) ResNet34

Fig. 3: DET performance of the five detection methods in the condition that multiple adversarial examples are mixed.

TABLE I
ATTACK PERFORMANCE OF THE BIM ATTACKER ON TWO SOTA SV MODELS. THE EERS OF THE ECAPA_TDNN AND RESNET34 SV MODELS ON GENUINE EXAMPLES ARE 1.20% AND 1.07% RESPECTIVELY.

	ϵ	5	10	15	20	30	40
ECAPA_TDNN	EER (%)	52.00	81.80	91.80	96.20	98.80	99.80
	ASR (%)	51.30	80.80	90.90	95.70	98.70	99.30
	SNR (in dB)	51.70	46.38	43.72	41.91	39.53	37.95
ResNet34	EER (%)	56.80	85.80	94.00	97.60	98.80	99.60
	ASR (%)	56.10	83.90	93.00	96.80	99.00	99.30
	SNR (in dB)	52.00	47.49	44.96	43.23	40.93	39.42

TABLE II
DETECTION SUCCESS RATE AND DETECTION TIME OF THE FIVE DETECTION METHODS WITH DIFFERENT GIVEN FARS ON TWO SOTA SV MODELS.

DSR (%)	$\text{FAR}_{\text{given}}$ (%)	5.0	1.0	0.5	0.1	Time/ms
ECAPA_TDNN	PWG	**98.10**	**96.10**	**95.03**	**91.98**	181.6
	GL-mel	92.55	88.18	85.70	77.58	76.1
	GL-lin	93.77	90.27	87.50	80.37	52.3
	MLFB-H	94.50	89.95	87.67	74.17	**18.0**
	MLFB-D	96.37	91.60	89.72	84.42	18.6
ResNet34	PWG	96.88	94.28	92.92	88.80	165.9
	GL-mel	75.90	62.80	55.60	40.00	69.5
	GL-lin	95.40	92.70	89.60	86.90	44.4
	MLFB-H	97.40	93.68	91.62	80.03	16.0
	MLFB-D	**98.97**	**97.92**	**96.98**	**95.05**	**15.9**

V. RESULTS AND DISCUSSION

Table I shows the attack results of adversarial examples on the two SOTA SV models, where the EER of the victim model, attack success rate (ASR) and SNR are used to evaluate attack performance with different threat models and different perturbation strengths. From the table, it can be seen that ECAPA_TDNN and ResNet34 achieve an EER of 1.20% and 1.07% respectively on the standard entire trials before being attacked, which indicates that the SV models are SOTA. However, with the increase of the attack intensity, both EER and ASR are increased from 50%+ to 99%+, while the average SNR remains above 37dB, which shows the strong threat of the generated adversarial examples. The following experiments were conducted on such threatening adversarial examples.

In Fig. 2, EER is used to evaluate the detection performance of five methods on two SV models. According to the definition in Section IV-C, the detection EER was calculated on $\mathcal{A}_\epsilon$ and $\mathcal{G}_\epsilon$ by (8), where $\epsilon = 5, 10, 15, 20, 30, 40$. Experimental results on the ECAPA_TDNN SV model show that the detection performance of MLFB-H/D is better than GL-mel and GL-lin methods, and slightly worse than that of PWG. Experimental results on the ResNet34 SV model show that the detection performance of MLFB-D is superior to all other four methods. Although the detection performance of MLFB-H is the worst after the perturbation intensity is higher than 20, its EER is still less than 4%.

In Fig. 3, the detection error tradeoff (DET) curve is used to evaluate the detection performance. The score variations are obtained from $\mathcal{A}_{\text{all}}$ and $\mathcal{G}_{\text{all}}$, where $\mathcal{A}_{\text{all}}$ is the set of the adversarial examples mixed by multiple perturbation intensities, and $\mathcal{G}_{\text{all}}$ is obtained in the same way. The results are basically consistent with that in Fig. 2. From the figure, we see that MLFB-D is slightly inferior to PWG on ECAPA_TDNN, but is superior to the other three methods. MLFB-D achieves the SOTA performance on the ResNet34 SV model with a detection EER of less than 2%.

Table II shows the DSR of the adversarial examples with a given FAR as well as the average detection time. Consistently with the condition of Fig. 3, here we mixed adversarial examples of various strengths for the evaluation. Because the score variation sets $\mathcal{V}_{\text{adv}}$ and $\mathcal{V}_{\text{gen}}$ can be obtained by the sets of the hybrid examples $\mathcal{A}_{\text{all}}$ and $\mathcal{G}_{\text{all}}$ respectively using (7), we calculate DSR by (9) with a given tolerable FAR. From the table, we observe that (i) DSR decreases when FAR drops from 5% to 0.1%. For example, the DSR of MLFB-D decreases from 98.97% to 95.05% on ResNet34; (ii) when the FAR is 0.1%, the proposed MLFB-D method can still reach a DSR of 84.42% on ECAPA_TDNN and 95.05% on ResNet34 respectively; (iii) the average detection time of the proposed two methods under 1000 examples is only about 18ms, which is only 1/10 of that consumed by the PWG method.

To explain why the detection methods are effective, we summarized the statistics of the score variations $v = |s - \hat{s}|$ between the adversarial examples and genuine examples for the five detection methods in Fig. 4. Substantially, both the baseline method in [12] and the proposed MLFB-H/D methods are seeking for a transformation that has the maximum impact on the adversarial examples and minimal impact on the genuine examples respectively. To achieve the best detection performance, the proposed MLFB-H adopts the masking length l to make a trade-off between the two impacts, so as to the

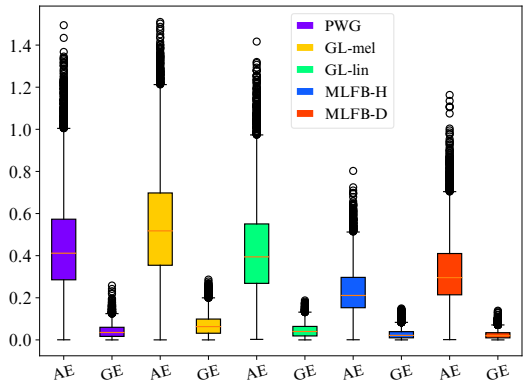


Fig. 4: Boxline of the score variations of the adversarial examples (AE) and genuine examples (GE) for the five detection methods on the ResNet34 SV model.

effect of the masking threshold ξ adopted by MLFB-D.

VI. Conclusion

In this paper, we have proposed to detect adversarial examples by masking the input acoustic features of SV. The proposed MLFB-H and MLFB-D differs in how the masks are generated. The foundation for the success of the proposed methods is that masking the parts of the acoustic features incorporating less speaker information will inevitably have a small impact on genuine examples, and large impact on adversarial examples. Experimental results show that the proposed adversarial example detection method, MLFB-D, outperforms the baseline models and achieves the SOTA detection performance on ResNet34. Moreover, the proposed two methods do not need model training and have a low cost of detection time.

VII. Acknowledgement

This work was supported in part by National Science Foundation of China under Grant No. 62176211, in part by Project of the Science, Technology, and Innovation Commission of Shenzhen Municipality under grant No. JSGG20210802152546026 and JCYJ20210324143006016.

References

[1] Z. Bai and X.-L. Zhang, "Speaker recognition based on deep learning: An overview," *Neural Networks*, vol. 140, pp. 65–99, 2021.

[2] J. Villalba, Y. Zhang, and N. Dehak, "x-vectors meet adversarial attacks: Benchmarking adversarial robustness in speaker verification." in *INTERSPEECH*, 2020, pp. 4233–4237.

[3] F. Kreuk, Y. Adi, M. Cisse, and J. Keshet, "Fooling end-to-end speaker verification with adversarial examples," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 1962–1966.

[4] J. Li, X. Zhang, C. Jia, J. Xu, L. Zhang, Y. Wang, S. Ma, and W. Gao, "Universal adversarial perturbations generative network for speaker recognition," in *2020 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2020, pp. 1–6.

[5] W. Zhang, S. Zhao, L. Liu, J. Li, X. Cheng, T. F. Zheng, and X. Hu, "Attack on practical speaker verification system using universal adversarial perturbations," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 2575–2579.

[6] Z. Li, C. Shi, Y. Xie, J. Liu, B. Yuan, and Y. Chen, "Practical adversarial attacks against speaker recognition systems," in *Proceedings of the 21st international workshop on mobile computing systems and applications*, 2020, pp. 9–14.

[7] Q. Wang, P. Guo, and L. Xie, "Inaudible adversarial perturbations for targeted attack in speaker recognition," *arXiv preprint arXiv:2005.10637*, 2020.

[8] Q. Wang, P. Guo, S. Sun, L. Xie, and J. H. Hansen, "Adversarial regularization for end-to-end robust speaker verification." in *Interspeech*, 2019, pp. 4010–4014.

[9] H. Zhang, L. Wang, Y. Zhang, M. Liu, K. A. Lee, and J. Wei, "Adversarial separation network for speaker recognition." in *INTERSPEECH*, 2020, pp. 951–955.

[10] H. Wu, X. Li, A. T. Liu, Z. Wu, H. Meng, and H.-y. Lee, "Adversarial defense for automatic speaker verification by cascaded self-supervised learning models," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6718–6722.

[11] X. Li, N. Li, J. Zhong, X. Wu, X. Liu, D. Su, D. Yu, and H. Meng, "Investigating robustness of adversarial samples detection for automatic speaker verification," *arXiv preprint arXiv:2006.06186*, 2020.

[12] H. Wu, P.-C. Hsu, J. Gao, S. Zhang, S. Huang, J. Kang, Z. Wu, H. Meng, and H.-Y. Lee, "Adversarial sample detection for speaker verification by neural vocoders," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 236–240.

[13] J. Villalba, S. Joshi, P. Żelasko, and N. Dehak, "Representation learning to classify and detect adversarial attacks against speaker and speech recognition systems," *arXiv preprint arXiv:2107.04448*, 2021.

[14] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," *arXiv preprint arXiv:1611.01236*, 2016.

[15] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.