Dual-adversarial domain adaptation for generalized replay attack detection

Hongji Wang, Heinrich Dinkel, Shuai Wang, Yanmin Qian, Kai Yu

MoE Key Lab of Artificial Intelligence
SpeechLab, Department of Computer Science and Engineering
AI Institute, Shanghai Jiao Tong University, Shanghai

{jijijiang77, richman, feixiang121976, yanminqian, kai.yu}@sjtu.edu.cn

Abstract

Despite tremendous progress in speaker verification recently, replay spoofing attacks are still a major threat to these systems. Focusing on dataset-specific scenarios, anti-spoofing systems have achieved promising in-domain performance at the cost of poor generalization towards unseen out-of-domain datasets. This is treated as a domain mismatch problem with a domain adversarial training (DAT) framework, which has previously been applied to enhance generalization. However, since only one domain discriminator is adopted, DAT suffers from the false alignment of cross-domain spoofed and genuine pairs, thus failing to acquire a strong spoofing-discriminative capability. In this work, we propose the dual-adversarial domain adaptation (DADA) framework to enable fine-grained alignment of spoofed and genuine data separately by using two domain discriminators, which effectively alleviates the above problem and further improves spoofing detection performance. Experiments on the ASVspoof 2017 V.2 dataset and the physical access portion of BTAS 2016 dataset demonstrate that the proposed DADA framework significantly outperforms the baseline model and DAT framework in cross-domain evaluation scenarios. It is shown that the newly proposed DADA architecture is more robust and effective for generalized replay attack detection.

Index Terms: dual-adversarial domain adaptation, domain invariant, replay spoofing attack detection, speaker verification

1. Introduction

As a more and more mature technology in identity authentication, Automatic Speaker Verification (ASV) has been deployed into many real-world applications in telephone banking, call centers, surveillance, etc. However, ASV systems are acknowledgedly vulnerable to various spoofing attacks, including impersonation, speech synthesis (SS), voice conversion (VC), and replay attacks [1, 2]. Compared with SS and VC attacks (Logical Access, LA), replay attacks (Physical Access, PA) pose a greater threat to ASV systems, for the reason that not only replay audios can be obtained with greater ease using consumer-grade devices, but also replay attacks are generally more difficult to be detected [3, 4, 5]. Although recent progress in replay spoofing detection has shown promising performance within a specific dataset, generalization towards unseen data in training is still very poor, especially for cross-dataset evaluation scenarios [6, 7, 8, 9]. Those results make sense due to the significant difference in speakers, accents, text, and especially replay configurations (e.g., acoustic environment, recording and playback devices) across datasets, which indeed lead to different data distributions and cause the spoofing detectors to over-fit seriously.

In [10], we defined this behavior as a domain-mismatch problem in replay spoofing detection and addressed it by introducing a domain adversarial training (DAT) framework. Specifically, a traditional neural-network-based anti-spoofing model is adapted by adding a new domain discriminator branch and then trained using the standard DAT strategy. Therefore, the DAT framework can learn better deep representations that are still spoofing-discriminative but domain-invariant. Note that there are two critical assumptions here:

- Different spoofing datasets are regarded as different domains because the replay configurations and spoofing types vary across them.
- Labeled source-domain data and unlabeled target-domain data are used for training, which can be termed as Unsupervised Domain Adaptation (UDA).

Obviously, the key point of DAT is to reduce the domain discrepancy by aligning the whole data distribution between the source domain and target domain using a single domain discriminator. According to [11], the DAT method does not consider complex multi-mode structures underlying the data distributions, which may lead to false alignments among different classes and further mix up the discriminative structure of the main learning task. Similarly, the spoofing-discriminative capability of the aforementioned DAT framework could be somewhat weakened or even sacrificed owing to the false alignment of cross-domain spoofed and genuine (bona fide) pairs.

Motivated by this, we present the dual-adversarial domain adaptation (DADA) approach for replay attack detection, which enables fine-grained alignment of spoofed and genuine data separately based on two domain discriminators: one for the spoofed class and the other for the genuine class. To validate the effectiveness of the DADA framework, three neural-network-based anti-spoofing models are evaluated: the adapted Light CNN (LCNN) model [10], the 10-layer ResNet (ResNet10) model [12], and our Context-Gate CNN (CGCNN) model presented in ASVspoof 2019 [13], based on which we propose the LCNN-DADA, ResNet10-DADA, and CGCNN-DADA frameworks. It is shown that each DADA framework outperforms the corresponding baseline model and DAT framework, with better generalization performance on unseen cross-domain data.

The rest of this paper is organized as follows. Section 2 illustrates the proposed dual-adversarial domain adaptation framework for replay spoofing attack detection. In Section 3, we present the experimental details as well as analyze the results. Finally, we conclude this paper in Section 4.

2. Dual-adversarial domain adaptation for replay spoofing attack detection

Based on deep neural networks, conventional anti-spoofing models can be decomposed into two components: the feature

Yanmin Qian and Kai Yu are the corresponding authors.

extractor aiming at learning deep spoofing-discriminative embeddings as well as the spoofing detector mapping the embeddings into spoofing labels (spoofed or genuine). In the DAT framework, a domain discriminator is additionally connected after the feature extractor through a gradient reversal layer (GRL) [10]. Similarly, the dual-adversarial domain adaptation framework for spoofing detection can be constructed by adding two domain discriminators: one for the spoofed class and the other for the genuine class. Ideally, the spoofed-class domain discriminator distinguishes the source domain from the target domain within spoofed data, while the genuine-class domain discriminator differentiates them within genuine data. Nevertheless, since the target-domain data is unlabeled in spoofing, it is not easy to decide which domain discriminator is responsible for each target-domain training sample. Fortunately, the outputs of the spoofing detector exactly convey strong label signals, which can be used as soft spoofing labels.

Figure 1 depicts the proposed DADA architecture. Firstly, an input feature $\mathbf{x}$ is fed into the feature extractor to learn a deep embedding $\mathbf{f}$. Afterward, for a labeled source-domain sample, we train the spoofing detector and its corresponding domain discriminator (spoofed-class or genuine-class). For an unlabeled target-domain sample, however, we first forward it through the spoofing detector to obtain its soft label, then we train both domain discriminators together by multiplying the losses with the corresponding class probabilities. The DADA architecture consists of three outputs: the spoofing label $\mathbf{y} \in \mathcal{Y}$, the spoofed-class domain label $\mathbf{d}^s \in \mathcal{D}^s$, and the genuine-class domain label $\mathbf{d}^g \in \mathcal{D}^g$, where $\mathcal{Y} = \mathcal{D}^s = \mathcal{D}^g = \{[0,1],[1,0]\}$.

Suppose a source domain $S = \{(\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{d}_i^s, \boldsymbol{d}_i^g)\}_{i=1}^{n_s}$ and a target domain $T = \{(\boldsymbol{x}_i, \boldsymbol{d}_i^s, \boldsymbol{d}_i^g)\}_{i=1}^{n_t}$ are given as training data. Furthermore, for a training sample $\boldsymbol{x}_i$, the spoofing label $\boldsymbol{y}_i = [y_i^s, y_i^g]$ is defined as follows:

$$[y_i^s, y_i^g] = \begin{cases} [1,0], & \boldsymbol{x}_i \in S \text{ and } \boldsymbol{x}_i \text{ is spoof}. \\ [0,1], & \boldsymbol{x}_i \in S \text{ and } \boldsymbol{x}_i \text{ is genuine}. \\ [\hat{y}_i^s, \hat{y}_i^g], & \boldsymbol{x}_i \in T. \end{cases} \tag{1}$$

where $[\hat{y}_i^s, \hat{y}_i^g]$ is the softmax output of the spoofing detector.

Note that the original losses of the spoofed-class and genuine-class domain predictions can be denoted as:

$$\mathcal{L}_d^s(\boldsymbol{x}_i) = \mathcal{L}_d^s(G_d^s(G_f(\boldsymbol{x}_i;\Theta_f);\Theta_d^s), \boldsymbol{d}_i^s) \tag{2}$$

$$\mathcal{L}_d^g(\boldsymbol{x}_i) = \mathcal{L}_d^g(G_d^g(G_f(\boldsymbol{x}_i;\Theta_f);\Theta_d^g), \boldsymbol{d}_i^g) \tag{3}$$

Hence, the unified domain prediction loss for any training sample $\boldsymbol{x}_i$ can be denoted as:

$$\mathcal{L}_d(\boldsymbol{x}_i) = y_i^s \mathcal{L}_d^s(\boldsymbol{x}_i) + y_i^g \mathcal{L}_d^g(\boldsymbol{x}_i) \tag{4}$$

Besides, if $\boldsymbol{x}_i$ is source-domain, we can calculate the spoofing detection loss:

$$\mathcal{L}_y(\boldsymbol{x}_i) = \mathcal{L}_y(G_y(G_f(\boldsymbol{x}_i;\Theta_f);\Theta_y), \boldsymbol{y}_i) \tag{5}$$

With the aim of seeking the best parameters Θ_f, Θ_y, Θ_d^s, and Θ_d^g that minimize the spoofing detection loss and meanwhile maximize the domain prediction loss, the cost function of the DADA framework can be formulated as follows:

$$C(\Theta_f, \Theta_y, \Theta_d^s, \Theta_d^g) = \frac{1}{n_s}\sum_{\boldsymbol{x}_i \in S} \mathcal{L}_y(\boldsymbol{x}_i) - \frac{\lambda}{n}\sum_{\boldsymbol{x}_i \in S \cup T} \mathcal{L}_d(\boldsymbol{x}_i) \tag{6}$$

where $n = n_s + n_t$, and λ is a positive coefficient that trades off two losses during back-propagation. Theoretically, Equation (6) can be optimized by seeking the saddle point $\hat{\Theta}_f$, $\hat{\Theta}_y$, $\hat{\Theta}_d^s$, and $\hat{\Theta}_d^g$ such that

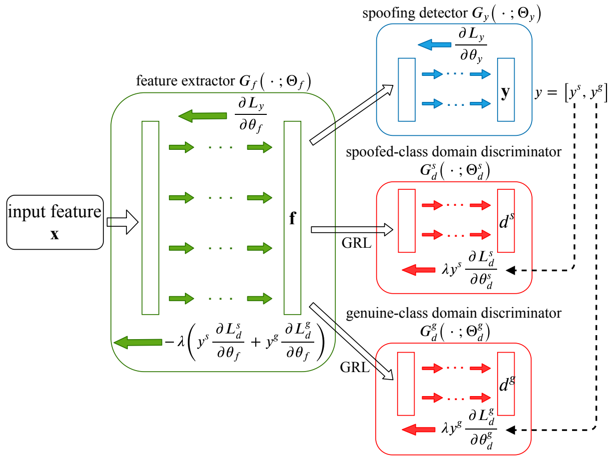


Figure 1: *The proposed dual-adversarial domain adaptation (DADA) architecture for replay attack detection. It includes a feature extractor (green), a spoofing detector (blue), as well as two domain discriminators (red): one for the spoofed class and the other for the genuine class. GRL stands for Gradient Reversal Layer, which reverses the gradient during back-propagation.*

$$\hat{\Theta}_f, \hat{\Theta}_y = \arg\min_{\Theta_f, \Theta_y} C(\Theta_f, \Theta_y, \hat{\Theta}_d^s, \hat{\Theta}_d^g) \tag{7}$$

$$\hat{\Theta}_d^s, \hat{\Theta}_d^g = \arg\max_{\Theta_d^s, \Theta_d^g} C(\hat{\Theta}_f, \hat{\Theta}_y, \Theta_d^s, \Theta_d^g) \tag{8}$$

Similar to [10, 14], the stochastic gradient descent (SGD) optimizer can be used to update the model parameters with the aid of the gradient reversal layer.

3. Experiments

3.1. Datasets

According to [15], anti-spoofing systems trained on simulated data cannot detect real-world spoofing attacks. Thus our work discards the ASVspoof 2019 PA dataset where the spoofed data is artificially simulated. All experiments are conducted on the ASVspoof 2017 V.2 dataset [16] as well as the PA portion of BTAS 2016 dataset [17] (denoted as BTAS-PA 2016 dataset). Detailed statistics of two datasets are shown in Table 1.

Table 1: *Overall duration (in hours) as well as the numbers of utterances and replay configurations (RC) for each subset.*

Subset	ASVspoof 2017 V.2			BTAS-PA 2016		
	Train	Dev	Eval	Train	Dev	Eval
dur (h)	2.22	1.44	11.94	20.86	19.95	21.50
# utts	3014	1710	13306	7773	7795	10376
# RCs	3	10	57	4	4	6

Covering ten different fixed pass-phrases, the genuine utterances in the ASVspoof 2017 V.2 dataset come from a subset of the RedDots corpus [18] that is commonly-used in text-dependent ASV research. They are further replayed and recorded using a variety of heterogeneous devices and acoustic environments. The BTAS 2016 dataset is based on the public AVspoof database [19], where recording and replay conditions cover different types of microphones/speakers with varying sound quality. For each dataset, we use the evaluation set as the testing set and *pool the training set and development set as the actual training data*, 10% of which are further divided as the validation set for model selection.

3.2. Experimental setup

Most of the experimental setups in our previous work [10] are reserved here. Firstly, we extract 257-dimensional log power spectrograms as front-end features by computing 512-point Short-Time Fourier Transform (STFT) every 10 ms with a window size of 25 ms. Afterward, we apply 300-frame sliding-window cepstral mean and variance normalization (cmvn) per utterance as well as global standardization. Since utterance lengths differ, we pad all utterances to the maximum length by repeating their features within every batch, which enables them to be processed in parallel. Due to the GPU memory limitation, the batch size is set to 8, and the maximum utterance length should not exceed 1500 during the training process.

PyTorch is used to implement all neural networks, whose parametric layers are initialized with Xavier initialization [20]. We adopt the cross-entropy loss criterion as well as the SGD optimizer with a learning rate of 0.001 and a momentum of 0.9 for all models. Furthermore, the evaluation metric is Equal Error Rate (EER), which is calculated with the score predictions directly from the spoofing detector.

Lastly, since the amount of training data is relatively small, especially in the ASVspoof 2017 V.2 dataset, we fix the seed for all pseudo-random generators (both CPU and GPU) and run each model for five times by enumerating the seed from one to five, which makes our results more convincing and easily-reproduced. The final EER of each model is the average of five corresponding EERs.

3.3. Model configurations

Backbones: In order to validate the effectiveness and robustness of the proposed DADA framework, besides the Light CNN (LCNN) model used in our previous work [10], this paper further investigates three model structures:

- Adapted Light CNN (LCNN): LCNN was the best system in ASVspoof 2017 [21], where a Max-Feature Map (MFM) activation is used after each convolution operation. It also performed well in ASVspoof 2019 [22]. Therefore, we reserve the adapted LCNN as a baseline, which applies to variable lengths of input features.
- 10-layer ResNet (ResNet10): The ResNet variations used in ASVspoof 2019 achieved great performance in the PA subtask [23, 24, 25]. ResNet10 comprised of only 4 residual blocks $\{1, 1, 1, 1\}$ [12] is comparable with LCNN (9-layer CNNs) in parameter size. Similarly, we remove all batchnorm layers inside.
- Context-Gate CNN (CGCNN): CGCNN was our main proposal in ASVspoof 2019, with promising performance in both PA and LA subtasks [13]. Specifically, gated linear unit (GLU) activations are used to replace the MFM activations in LCNN. Except for that, CGCNN shares a similar structure with LCNN.

DAT and DADA frameworks: As mentioned in Section 2, compared to the baseline models, the corresponding DAT and DADA frameworks are constructed by adding one and two domain discriminator branches, respectively. In our experiments, each domain discriminator is a 2-layer perceptron (input size: 64, hidden size: 64, output size:2), mapping the 64-dimensional output from the feature extractor to 2 classes (source and target domains). All model definitions are open-source [1].

Table 2: *EERs (%) of the baseline models as well as the corresponding DAT and DADA frameworks. $\overline{A_{train}}$ means that A_{train} is used without labels, and similarly for $\overline{B_{train}}$.*

Models	Training data	Testing sets		
		A_{eval}	B_{eval}	Avg
LCNN	$A_{train} + B_{train}$	14.15	5.87	10.01
LCNN	A_{train}	10.13	12.43	11.28
LCNN-DAT	$A_{train} + \overline{B_{train}}$	10.21	11.51	10.86
LCNN-DADA	$A_{train} + \overline{B_{train}}$	10.05	**10.07**	**10.06**
LCNN	B_{train}	18.65	8.83	13.74
LCNN-DAT	$B_{train} + \overline{A_{train}}$	18.37	8.94	13.65
LCNN-DADA	$B_{train} + \overline{A_{train}}$	**16.60**	9.34	**12.97**
ResNet10	$A_{train} + B_{train}$	15.49	5.92	10.70
ResNet10	A_{train}	13.36	16.77	15.06
ResNet10-DAT	$A_{train} + \overline{B_{train}}$	13.35	17.00	15.17
ResNet10-DADA	$A_{train} + \overline{B_{train}}$	14.72	**12.83**	**13.77**
ResNet10	B_{train}	22.21	6.11	14.16
ResNet10-DAT	$B_{train} + \overline{A_{train}}$	22.74	7.02	14.88
ResNet10-DADA	$B_{train} + \overline{A_{train}}$	**15.74**	5.69	**10.71**
CGCNN	$A_{train} + B_{train}$	13.39	4.58	8.98
CGCNN	A_{train}	12.49	18.21	15.35
CGCNN-DAT	$A_{train} + \overline{B_{train}}$	10.87	17.84	14.35
CGCNN-DADA	$A_{train} + \overline{B_{train}}$	11.27	**13.64**	**12.45**
CGCNN	B_{train}	20.60	6.59	13.59
CGCNN-DAT	$B_{train} + \overline{A_{train}}$	19.79	6.86	13.32
CGCNN-DADA	$B_{train} + \overline{A_{train}}$	**16.20**	7.34	**11.77**

DAT and DADA training strategies: To compensate for the data imbalance between the source and target domains, we oversample the minority one to match the majority one. Afterward, we update the model parameters every two batches: one is source-domain and the other is target-domain. Moreover, to suppress the noisy domain signals at the early training stages, the trade-off factor λ adapts from 0 to 1 gradually, following the schedule:

$$\lambda = \frac{2}{1 + \exp(-\gamma \cdot e)} - 1 \tag{9}$$

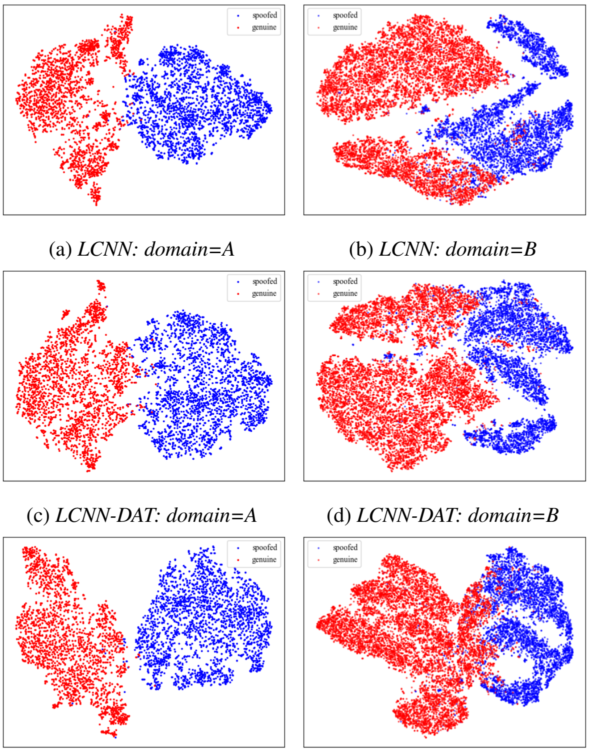
where γ is set as 0.01 (after fine-tuning), and e refers to the number of epochs that have been trained.

3.4. Results and analysis

Here, we denote the training data (*both training set and development set*) and the testing set (*evaluation set*) of the ASVspoof 2017 V.2 dataset and BTAS-PA 2016 dataset as A_{train}, A_{eval}, B_{train}, and B_{eval}, respectively. EERs (%) of different systems are compared in Table 2.

Although the baseline models achieve great performance on in-domain testing sets, they generalize poorly on cross-domain testing sets, with significant performance degradation. For example, ResNet10 trained on B_{train} achieves 6.11% EER on B_{eval}, while only 22.21% EER on A_{eval}. By adopting the DAT framework, the performance degradation can be slightly reduced for both LCNN and CGCNN, but increased for ResNet10.

[1] https://github.com/JiJiJiang/ASV-Anti-Spoofing-DADA.git



(a) *LCNN: domain=A* (b) *LCNN: domain=B*

(c) *LCNN-DAT: domain=A* (d) *LCNN-DAT: domain=B*

(e) *LCNN-DADA: domain=A* (f) *LCNN-DADA: domain=B*

Figure 2: *The t-SNE visualization of all training data embeddings in each domain that are extracted by LCNN trained on A_{train}, LCNN-DAT trained on "$A_{train}+\overline{B_{train}}$", and LCNN-DADA trained on "$A_{train}+\overline{B_{train}}$", respectively. The ASVspoof 2017 V.2 dataset (A) and the BTAS-PA 2016 dataset (B) are the source domain and target domain, respectively.*

However, each DADA framework significantly outperforms the corresponding baseline model and DAT framework in cross-domain evaluation scenarios. In addition, for each DADA framework, it achieves comparable performance with the corresponding baseline model as well as the DAT framework within the original source domain. Considering both testing sets, the new DADA approach achieves the best overall generalization performance, as seen in the "Avg" column.

We also train the baseline model on "A_{train} + B_{train}" to investigate the upper bound of the proposed DADA framework. It is shown that the proposed DADA framework can achieve averagely very close performance to the corresponding baseline model trained on "A_{train} + B_{train}", especially for ResNet10-DADA trained on "B_{train} + $\overline{A_{train}}$". Interestingly, although the baseline model trained on "A_{train} + B_{train}" outperforms that trained on only B_{train} when tested on B_{eval}, it performs worse on A_{eval} compared with that trained on only A_{train}. The reason is probably that A_{train} is much smaller than B_{train}, making the baseline model over-fit to the B domain severely.

3.4.1. t-SNE visualization

To better understand the mechanism of the new DADA framework, we use t-SNE projection [26] to visualize embedding distributions of the models. An example is shown in Figure 2. LCNN trained on A_{train} cannot generalize well on B domain because of the obvious domain mismatch. With the DAT framework, although the whole domain discrepancy is mitigated slightly, LCNN-DAT fails to discriminate spoofed samples from genuine ones well. However, not only LCNN-DADA aligns the data distributions better in fine grain, but also it acquires a stronger spoofing-discriminative capability, which shows that the proposed DADA approach is more effective and generalizes better on unseen cross-domain data.

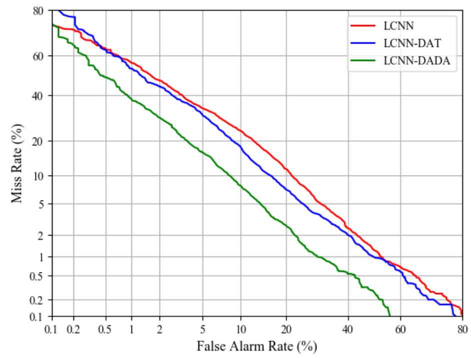


Figure 3: *The DET curves for the LCNN, LCNN-DAT, and LCNN-DADA models, respectively, when tested on B_{eval}.*

3.4.2. Detection Error Trade-off curve

Since EER only corresponds to the threshold where the miss rate equals to the false alarm rate, the detection error trade-off (DET) curve is adopted to intuitively show the system performance at each threshold. Figure 3 compares the DET curves for the same models mentioned above. Obviously, the new LCNN-DADA model achieves both lower miss rate and false alarm rate at any threshold in comparison with the LCNN and LCNN-DAT models, which reveals the robustness of the proposed DADA framework for replay spoofing attack detection.

4. Conclusions

Although the domain adversarial training (DAT) framework mitigates the domain mismatch for replay attack detection, it cannot acquire a strong spoofing-discriminative capability due to the false alignment of spoofed and genuine pairs across domains. This paper proposes the dual-adversarial domain adaptation (DADA) framework to enable fine-grained alignment of spoofed and genuine data separately by using two domain discriminators, thus effectively alleviating the false alignment problem and further improving generalization performance for replay spoofing detection. Experiments conducted on the ASVspoof 2017 V.2 dataset as well as the BTAS-PA 2016 dataset show that the newly proposed DADA framework significantly outperforms the corresponding baseline model (LCNN, ResNet10 or CGCNN) and our previous DAT framework in cross-domain evaluation scenarios, with the best overall generalization performance. Furthermore, examples are given to show the effectiveness and robustness of the DADA framework for generalized replay attack detection.

5. Acknowledgements

This work has been supported by the National Key Research and Development Program of China (Grant No.2017YFB1302402). Experiments have been carried out on the PI supercomputers at Shanghai Jiao Tong University.

6. References

[1] Z. Wu, N. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, and H. Li, "Spoofing and countermeasures for speaker verification: A survey," *speech communication*, vol. 66, pp. 130–153, 2015.

[2] Y. Qian, N. Chen, H. Dinkel, and Z. Wu, "Deep feature engineering for noise robust spoofing detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 10, pp. 1942–1955, 2017.

[3] F. Alegre, A. Janicki, and N. Evans, "Re-assessing the threat of replay spoofing attacks against automatic speaker verification," in *2014 International Conference of the Biometrics Special Interest Group (BIOSIG)*. IEEE, 2014, pp. 1–6.

[4] S. Yoon and H. Yu, "Multiple points input for convolutional neural networks in replay attack detection," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6444–6448.

[5] Y. Qian, N. Chen, and K. Yu, "Deep features for automatic spoofing detection," *Speech Communication*, vol. 85, pp. 43–52, 2016.

[6] P. Korshunov and S. Marcel, "Cross-database evaluation of audio-based spoofing detection systems," in *Interspeech*, no. CONF, 2016.

[7] D. Paul, M. Sahidullah, and G. Saha, "Generalization of spoofing countermeasures: A case study with asvspoof 2015 and btas 2016 corpora," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 2047–2051.

[8] H. Dinkel, Y. Qian, and K. Yu, "Investigating raw wave deep neural networks for end-to-end speaker spoofing detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 11, pp. 2002–2014, Nov 2018.

[9] R. K. Das, J. Yang, and H. Li, "Assessing the scope of generalized countermeasures for anti-spoofing," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6589–6593.

[10] H. Wang, H. Dinkel, S. Wang, Y. Qian, and K. Yu, "Cross-domain replay spoofing attack detection using domain adversarial training," *Proc. Interspeech 2019*, pp. 2938–2942, 2019.

[11] Z. Pei, Z. Cao, M. Long, and J. Wang, "Multi-adversarial domain adaptation," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[12] S. Wang, Y. Yang, T. Wang, Y. Qian, and K. Yu, "Knowledge distillation for small foot-print deep speaker embedding," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6021–6025.

[13] Y. Yang, H. Wang, H. Dinkel, Z. Chen, S. Wang, Y. Qian, and K. Yu, "The sjtu robust anti-spoofing system for the asvspoof 2019 challenge," *Proc. Interspeech 2019*, pp. 1038–1042, 2019.

[14] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[15] G. Lavrentyeva, S. Novoselov, M. Volkova, Y. Matveev, and M. De Marsico, "Phonespoof: A new dataset for spoofing attack detection in telephone channel," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 2572–2576.

[16] H. Delgado, M. Todisco, M. Sahidullah, N. Evans, T. Kinnunen, K. Lee, and J. Yamagishi, "Asvspoof 2017 version 2.0: meta-data analysis and baseline enhancements," in *Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018.

[17] P. Korshunov, S. Marcel, H. Muckenhirn, A. R. Gonçalves, A. S. Mello, R. V. Violato, F. O. Simões, M. U. Neto, M. de Assis Angeloni, J. A. Stuchi *et al.*, "Overview of btas 2016 speaker anti-spoofing competition," in *2016 IEEE 8th international conference on biometrics theory, applications and systems (BTAS)*. IEEE, 2016, pp. 1–6.

[18] T. Kinnunen, M. Sahidullah, M. Falcone, L. Costantini, R. G. Hautamäki, D. Thomsen, A. Sarkar, Z. Tan, H. Delgado, M. Todisco, N. Evans, V. Hautamäki, and K. A. Lee, "Reddots replayed: A new replay spoofing attack corpus for text-dependent speaker verification research," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5395–5399.

[19] S. K. Ergünay, E. Khoury, A. Lazaridis, and S. Marcel, "On the vulnerability of speaker verification to realistic voice spoofing," in *2015 IEEE 7th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. IEEE, 2015, pp. 1–6.

[20] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[21] G. Lavrentyeva, S. Novoselov, E. Malykh, A. Kozlov, O. Kudashev, and V. Shchemelinin, "Audio replay attack detection with deep learning frameworks." in *Interspeech*, 2017, pp. 82–86.

[22] G. Lavrentyeva, S. Novoselov, A. Tseren, M. Volkova, A. Gorlanov, and A. Kozlov, "Stc antispoofing systems for the asvspoof2019 challenge," *arXiv preprint arXiv:1904.05576*, 2019.

[23] X. Cheng, M. Xu, and T. F. Zheng, "Replay detection using cqt-based modified group delay feature and resnewt network in asvspoof 2019," in *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2019, pp. 540–545.

[24] C.-I. Lai, N. Chen, J. Villalba, and N. Dehak, "Assert: Anti-spoofing with squeeze-excitation and residual networks," *arXiv preprint arXiv:1904.01120*, 2019.

[25] W. Cai, H. Wu, D. Cai, and M. Li, "The dku replay detection system for the asvspoof 2019 challenge: On data augmentation, feature representation, classification, and fusion," *arXiv preprint arXiv:1907.02663*, 2019.

[26] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.