INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Cross-domain replay spoofing attack detection using domain adversarial training

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

MoE Key Lab of Artificial Intelligence
SpeechLab, Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai, China
{jijijiang77, richman, feixiang121976, yanminqian, kai.yu}@sjtu.edu.cn

Abstract

Replay spoofing attacks are a major threat for speaker verification systems. Although many anti-spoofing systems or countermeasures are proposed to detect dataset-specific replay attacks with promising performance, they generalize poorly when applied on unseen datasets. In this work, the cross-dataset scenario is treated as a domain-mismatch problem and dealt with using a domain adversarial training framework. Compared with previous approaches, features learned from this newly-designed architecture are more discriminative for spoofing detection, but more indistinguishable across different domains. Only labeled source-domain data and unlabeled target-domain data are required during the adversarial training process, which can be regarded as unsupervised domain adaptation. Experiments on the ASVspoof 2017 V.2 dataset as well as the physical access condition part of BTAS 2016 dataset demonstrate that a significant EER reduction of over relative 30% can be obtained after applying the proposed domain adversarial training framework. It is shown that our proposed model can benefit from a large amount of unlabeled target-domain training data to improve detection accuracy.

Index Terms: domain adversarial training, unsupervised domain adaptation, replay spoofing detection, speaker verification

1. Introduction

Automatic speaker verification (ASV) has aroused researchers' attention in the last few decades due to its convenience and reliability for identity authentication. The success of applying deep neural networks further made a significant progress [1, 2, 3, 4], which led to its commercialization for applications in call centers, telephone banking, etc. However, the vulnerability of ASV technologies exposes ASV systems to various spoofing attacks. Depending on whether the spoofing attacks are performed at the sensor level or not, they can be divided into two categories: logical access (LA) condition with Speech synthesis (SS) and voice conversion (VC) attacks, and physical access (PA) condition with replay attacks. Compared with SS and VC attacks, replay attacks pose a greater threat to ASV systems, due to that not only replay audios can be acquired more easily by attackers without any expertise, but also replay attacks are generally more difficult to be detected.

Anti-spoofing technologies are developed to protect ASV systems from malicious spoofing attacks. Recently, some work focused on improving front-end features extracted from audio [5, 6, 7, 8] as well as deep learning models [9, 10, 11, 12] have shown the effect for spoofing detection. Even though the performance of spoofing detection within a specific dataset is promising, generalization towards data unseen in training is still a major problem. Specifically, previous works [13, 14, 15] showed great performance on the in-dataset scenario, such as ASVspoof 2015 dataset [16] and BTAS 2016 dataset [17], but degraded significantly for cross-dataset evaluation. Those results are reasonable since the replay configuration (e.g., recording and playback devices) varies considerably among different spoofing types, spoofing detectors often over-fit to the spoofing types seen in the training set and therefore generalize poorly to unseen ones. Different spoofing types lead to different data distributions, which may explain the poor cross-dataset performance. Here we define this behavior as a domain-mismatch problem for replay attack detection, where the source domain and target domain are defined to represent the distribution of training data and testing data, respectively. For real-world applications, replay spoofing types of unseen data are unpredictable, which makes it impossible to prepare training data of all potential spoofing types in advance. Moreover, recording and labeling new data is costly and therefore often unfeasible, while collecting unlabeled data is relatively easy and affordable. In this paper, we will try to address this domain-mismatch problem for replay attacks where unlabeled target-domain data is available.

The domain-mismatch problem caused by the difference of data distribution between the source domain and target domain occurs in many tasks, such as face recognition [18] and speaker recognition [19]. Approaches to address this problem are often termed as Domain Adaptation (DA), which aims at learning a discriminative predictor in the presence of a distribution shift between two domains. If only unlabeled target domain data is available in the training stage, it will be termed as Unsupervised Domain Adaptation (UDA). One classic UDA method is to adopt domain adversarial training (DAT), which aims at learning features that are discriminative for the main learning task but indistinguishable across domains by using adversarial training between the feature extractor and the domain classifier.

This paper adopts the DAT approach on unsupervised domain adaptation to address the domain-mismatch problem for replay spoofing attack detection. Deep features are first learned by a feature extractor and then passed to two different classification branches. One is the replay spoofing attack detector that judges whether an attempt is a replay attack, the other is the domain classifier which is connected through a gradient reversal layer. An adapted version of a Light CNN (LCNN) model is used as the baseline system, based on which we propose the LCNN-DAT framework. Lastly, the impact of using a different amount of unlabeled target-domain training data is further compared in this paper.

The remainder of this paper is organized as follows. Section 2 illustrates the proposed domain adversarial training architecture for replay spoofing attack detection. In section 3, we

http://dx.doi.org/10.21437/Interspeech.2019-2120

introduce the experimental details as well as present the results and analysis. Finally, conclusions are made in section 4.

2. Domain adversarial training for replay spoofing attack detection

A conventional deep neural network for replay spoofing attack detection usually contains two components: one is the feature extractor that aims at finding discriminative features, the other is the spoofing detector that maps the features into spoofing labels which suggest whether they are spoofing attacks or genuine attempts. Suppose input samples are $\mathbf{x} \in \mathcal{X}$ and output labels are $\mathbf{y} \in \mathcal{Y} = \{[0,1],[1,0]\}$, where $\mathcal{X}$ and $\mathcal{Y}$ are input feature space and output label space, respectively. In a domain-mismatch scenario, source domain data and target domain data share a similar but different data distribution, denoted as $S(\mathbf{x},\mathbf{y})$ and $T(\mathbf{x},\mathbf{y})$, respectively.

In order to alleviate the effect of domain mismatch, we propose a DAT architecture that learns deep features being discriminative for replay spoofing detection but indistinguishable across different domains, which is depicted in Figure 1. Different from a traditional neural network, a new branch is connected after the feature extractor through a gradient reversal layer, serving as the domain classifier. Therefore, the DAT architecture consists of two output layers: one is the spoofing labels $\mathbf{y} \in \mathcal{Y}$ and the other is the domain labels $\mathbf{d} \in \mathcal{D}$. Here $\mathcal{Y} = \mathcal{D} = \{[0,1],[1,0]\}$, because spoofing is commonly modelled as a binary classification task.

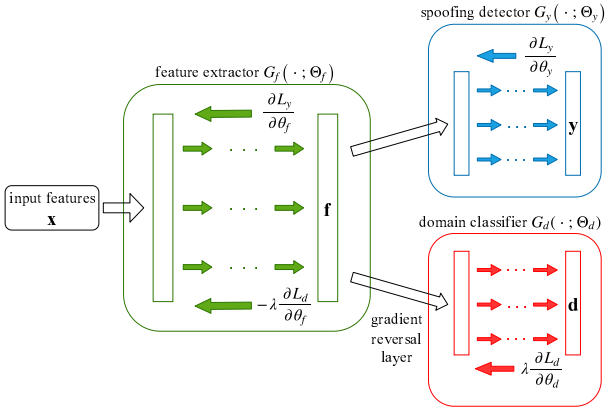


Figure 1: *The proposed domain adversarial training (DAT) architecture for spoofing detection. It includes a feature extractor (green), a spoofing detector (blue) and a domain classifier (red). A gradient reversal layer (GRL), between the feature extractor and the domain classifier, reverses the gradient during back-propagation.*

Specifically, the corresponding mapping functions of the feature extractor $G_f(\cdot;\Theta_f)$, spoofing detector $G_y(\cdot;\Theta_y)$ and domain classifier $G_d(\cdot;\Theta_d)$ are formulated as follows:

$$\mathbf{f} = G_f(\mathbf{x};\Theta_f) \quad (1)$$
$$\mathbf{y} = G_y(\mathbf{f};\Theta_y) \quad (2)$$
$$\mathbf{d} = G_d(\mathbf{f};\Theta_d) \quad (3)$$

Denote $\boldsymbol{x}_i$ as the i-th input sample with labels $\boldsymbol{y}_i$ and $\boldsymbol{d}_i$, which indicates $\boldsymbol{x}_i$ comes from the source domain ($(\boldsymbol{x}_i,\boldsymbol{y}_i) \sim S(\mathbf{x},\mathbf{y})$ if $\boldsymbol{d}_i = [0,1]$) or the target domain ($(\boldsymbol{x}_i,\boldsymbol{y}_i) \sim T(\mathbf{x},\mathbf{y})$ if $\boldsymbol{d}_i = [1,0]$). The spoofing detection loss and domain prediction loss of the i-th input sample are denoted as:

$$\mathcal{L}_y^i(\Theta_f,\Theta_y) = \mathcal{L}_y(G_y(G_f(\boldsymbol{x}_i;\Theta_f);\Theta_y),\boldsymbol{y}_i) \quad (4)$$
$$\mathcal{L}_d^i(\Theta_f,\Theta_d) = \mathcal{L}_d(G_d(G_f(\boldsymbol{x}_i;\Theta_f);\Theta_d),\boldsymbol{d}_i) \quad (5)$$

With the purpose of finding spoofing-discriminative and domain-invariant features, we aim to seek the best parameters Θ_f, Θ_y and Θ_d that minimize the spoofing detection loss and meanwhile maximize the domain prediction loss. Thus the total loss of the whole network for N input samples can be formulated as follows:

$$\begin{aligned} E(\Theta_f,\Theta_y,\Theta_d) &= \sum_{\substack{i=1,\dots,N \\ \boldsymbol{d}_i=[0,1]}} \left(\mathcal{L}_y^i(\Theta_f,\Theta_y) - \lambda\mathcal{L}_d^i(\Theta_f,\Theta_d)\right) \\ &\quad - \sum_{\substack{i=1,\dots,N \\ \boldsymbol{d}_i=[1,0]}} \lambda\mathcal{L}_d^i(\Theta_f,\Theta_d) \\ &= \sum_{\substack{i=1,\dots,N \\ \boldsymbol{d}_i=[0,1]}} \mathcal{L}_y^i(\Theta_f,\Theta_y) - \lambda\sum_{i=1}^{N}\mathcal{L}_d^i(\Theta_f,\Theta_d) \end{aligned} \quad (6)$$

where λ is a positive coefficient that trades off two losses during the process of back-propagation. According to [20], eq. (6) can be optimized theoretically by finding the saddle point $\hat{\Theta}_f$, $\hat{\Theta}_y$ and $\hat{\Theta}_d$ such that

$$\hat{\Theta}_f,\hat{\Theta}_y = \arg\min_{\Theta_f,\Theta_y} E(\Theta_f,\Theta_y,\hat{\Theta}_d) \quad (7)$$
$$\hat{\Theta}_d = \arg\max_{\Theta_d} E(\hat{\Theta}_f,\hat{\Theta}_y,\Theta_d) \quad (8)$$

Using stochastic gradient descent (SGD) with the aid of the gradient reversal layer, the gradients for a source-domain sample update as follows:

$$\theta_f = \theta_f - \alpha\left(\frac{\partial\mathcal{L}_y^i}{\partial\theta_f} - \lambda\frac{\partial\mathcal{L}_d^i}{\partial\theta_f}\right), \quad \forall\,\theta_f \in \Theta_f \quad (9)$$

$$\theta_y = \theta_y - \alpha\frac{\partial\mathcal{L}_y^i}{\partial\theta_y}, \quad \forall\,\theta_y \in \Theta_y \quad (10)$$

$$\theta_d = \theta_d - \alpha\lambda\frac{\partial\mathcal{L}_d^i}{\partial\theta_d}, \quad \forall\,\theta_d \in \Theta_d \quad (11)$$

where α is the learning rate. For a target-domain sample, parameters Θ_y do not update, and parameters Θ_d still update as eq. (11) while parameters Θ_f change their updating rule:

$$\theta_f = \theta_f + \alpha\lambda\frac{\partial\mathcal{L}_d^i}{\partial\theta_f}, \quad \forall\,\theta_f \in \Theta_f \quad (12)$$

3. Experiments

3.1. Datasets

Experiments are conducted on the ASVspoof 2017 V.2 dataset [21] as well as the PA part of BTAS 2016 dataset [17] (only genuine audios and replay attacks, denoted as BTAS-PA 2016 dataset). Detailed statistics on the numbers of utterances of two datasets are shown in Table 1.

For the ASVspoof 2017 V.2 dataset, all genuine audios come from a subset of original RedDots corpus, while the replay

Table 1: *Numbers of utterances in the ASVspoof 2017 V.2 dataset and BTAS-PA 2016 dataset.*

Subset	ASVspoof 2017 V.2			BTAS-PA 2016		
	Train	Dev	Eval	Train	Dev	Eval
Genuine	1507	760	1298	4973	4995	5576
Replay	1507	950	12008	2800	2800	4800
Total	3014	1710	13306	7773	7795	10376

audios are recorded under various replay configurations that include different combinations of acoustic environments, playback devices, and recording devices. The BTAS 2016 dataset is based on the public AVspoof database [22], where surreptitious recordings are also made in different setups and environmental conditions and two more "unknown" types of replay attacks are further added into the evaluation set to make the competition more challenging. Additionally, both development set and evaluation set of the ASVspoof 2017 V.2 dataset and BTAS-PA 2016 dataset are only reserved as testing sets in all experiments. For model selection, we divide 10% of the training set as the validation set.

3.2. Experimental setup

The front-end features are 257-dimension spectrograms that are obtained via computing 512-point Fast Fourier Transform (FFT) every 10 ms with a window size of 25 ms. The Librosa [23] library is used to extract front-end features from raw data, while we employ the Kaldi [24] toolkit to apply cepstral mean and variance normalization (cmvn) per utterance with a 300-frame sliding window. Besides, the mean and standard deviation of the training data are calculated and used for global standardization.

Training is done in utterance fashion, meaning that padding needs to be applied, since utterance lengths differ. In order to process all utterances in parallel within a batch, we pad to the longest length by repeating their features within every batch. The batch size is set to 8 in all experiments.

All neural networks are implemented in PyTorch and Xavier initialization [25] is used for all parametric layers. Cross-entropy loss is adopted as the loss criterion and SGD optimizer with a momentum of 0.9 and a learning rate of 0.0001 is used during the training process of all models. Furthermore, an end-to-end scoring method is adopted, which directly uses score predictions from the neural network to calculate the performance metric (EER). The EER is calculated using the toolkit offered in the ASVspoof 2019 challenge.

3.3. Baseline system

Light CNN (LCNN) was the best system of the ASVspoof 2017 challenge [26], where Max-Feature Map (MFM) activations are used after CNN modules. Since we use batch padding instead of padding all utterances to the maximal length globally, the number of frames (denoted as T) vary from batch to batch. Hence, we adapt the LCNN implemented in [26] into a new version that applies to variable lengths of input features.

The details of the LCNN architecture are described in Table 2. The ceiling mode is used in all max-pooling layers to make it applicable to short utterances with less than 32 frames. Besides, mean pooling is applied in the time dimension after the MaxPool5 layer, thus significantly reducing the number of parameters in the fully-connected (FC) FC6 layer. Dropout layers with a 0.5 ratio are used in both FC7 and FC8.

Table 2: *The architecture of Light CNN model.*

Type	**Filter Size /Stride,Pad**	**Output Size**	**#Params**
Conv1	$5 \times 5/1, 2$	$T \times 257 \times 32$	0.8K
MFM1	-	$T \times 257 \times 16$	-
MaxPool1	$2 \times 2/2, 0$	$T/2 \times 129 \times 16$	-
Conv2a	$1 \times 1/1, 0$	$T/2 \times 129 \times 32$	0.5K
MFM2a	-	$T/2 \times 129 \times 16$	-
Conv2b	$3 \times 3/1, 1$	$T/2 \times 129 \times 48$	6.9K
MFM2b	-	$T/2 \times 129 \times 24$	-
MaxPool2	$2 \times 2/2, 0$	$T/4 \times 65 \times 24$	-
Conv3a	$1 \times 1/1, 0$	$T/4 \times 65 \times 48$	1.2K
MFM3a	-	$T/4 \times 65 \times 24$	-
Conv3b	$3 \times 3/1, 1$	$T/4 \times 65 \times 64$	13.8K
MFM3b	-	$T/4 \times 65 \times 32$	-
MaxPool3	$2 \times 2/2, 0$	$T/8 \times 33 \times 32$	-
Conv4a	$1 \times 1/1, 0$	$T/8 \times 33 \times 64$	2.0K
MFM4a	-	$T/8 \times 33 \times 32$	-
Conv4b	$3 \times 3/1, 1$	$T/8 \times 33 \times 32$	9.2K
MFM4b	-	$T/8 \times 33 \times 16$	-
MaxPool4	$2 \times 2/2, 0$	$T/16 \times 17 \times 16$	-
Conv5a	$1 \times 1/1, 0$	$T/16 \times 17 \times 32$	0.5K
MFM5a	-	$T/16 \times 17 \times 16$	-
Conv5b	$3 \times 3/1, 1$	$T/16 \times 17 \times 32$	4.6K
MFM5b	-	$T/16 \times 17 \times 16$	-
MaxPool5	$2 \times 2/2, 0$	$T/32 \times 9 \times 16$	-
MeanPool5	-	144	-
FC6	-	128	18.4K
MFM6	-	64	-
FC7	-	64	4.1K
FC8	-	2	0.1K
Total	-	-	62.1K

3.4. Evaluation of the proposed LCNN-based domain adversarial training framework

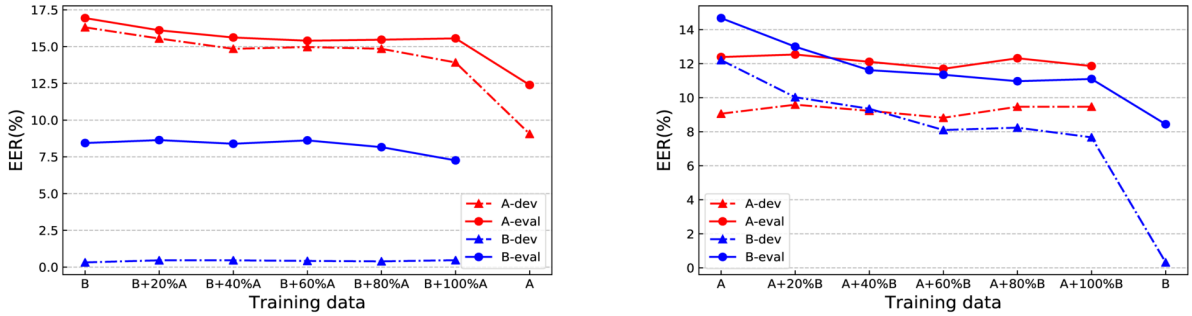
The LCNN-based DAT (LCNN-DAT) framework can be easily obtained from the baseline LCNN model. Specifically, layers from Conv1 to MFM6 are regarded as the feature extractor while the FC7 and FC8 layers compose the spoofing detector. A duplicate copy of the spoofing detector serves as the domain classifier that is connected after the feature extractor through a gradient reversal layer. However, we do not use dropout in the domain classifier.

3.4.1. Domain adversarial training procedure

In order to compensate for the imbalance between the amount of source-domain training data and target-domain training data, we oversample the minority one to match the majority one. Afterward, batches of all source-domain data and batches of all target-domain data are used to train the models in turn. Moreover, to suppress the noisy signals from the domain classifier at the early training stages, we change the adaptation factor λ from 0 to 1 gradually rather than fix it initially, using the following schedule:

$$\lambda = \frac{2}{1 + \exp(-\gamma \cdot e)} - 1 \tag{13}$$

where γ is set to 0.1, and e refers to the number of epochs that have been trained.



(a) *ASVspoof 2017 V.2 dataset (A) as the target domain*

(b) *BTAS-PA 2016 dataset (B) as the target domain*

Figure 2: *EERs of the LCNN or LCNN-DAT models trained on different training data. The training data are A or B meaning that LCNN models are trained on A-train or B-train only, while A+p%B (p=20, 40, 60, 80, 100) refers that LCNN-DAT models are trained on the whole source-domain A-train and p% of target-domain B-train data, and vice versa for B+p%A.*

3.4.2. LCNN-DAT Evaluation

Here, we denote the training set, development set and evaluation set of the ASVspoof 2017 V.2 dataset and BTAS-PA 2016 dataset as A-train, A-dev, A-eval, B-train, B-dev, and B-eval, respectively. Table 3 compares the performance of different systems in terms of EER(%).

Table 3: *EERs (%) of the baseline LCNN models and the proposed LCNN-DAT models on A-dev, A-eval, B-dev, and B-eval. Using A-train+B-train as training data means A-train is source-domain data while B-train is target-domain data, and vice versa for B-train+A-train.*

Models	Training data	Testing datasets			
		A-dev	A-eval	B-dev	B-eval
LCNN	A-train	9.06	12.39	12.21	14.68
LCNN-DAT	A-train+ B-train	9.47	11.86	**7.67**	**11.10**
LCNN	B-train	16.31	16.94	0.32	8.44
LCNN-DAT	B-train+ A-train	**13.92**	**15.56**	0.47	7.26

We achieve 9.06 EER on A-dev and 12.39 EER on A-eval, which suggests our implementation of LCNN is consistent with [11] but generalizes slightly better. Moreover, the LCNN models perform well on both B-dev and B-eval but turn out to overfit on B-train, which explains the significant performance difference. Although the LCNN models perform well within the same domain, they generalize poorly across these two datasets. However, the performance degradation across domains can be effectively reduced by introducing the proposed domain adversarial training architecture, without worsening its overall performance within the original source domain. Specifically, the relative reductions of performance degradation are 38% on B-dev and 57% on B-eval if using LCNN-DAT models trained on A-train+B-train, and are 33% on A-dev and 30% on A-eval if using LCNN-DAT models trained on B-train+A-train. The results show that via introducing domain adversarial training into the LCNN framework, the LCNN-DAT models generalize much better for cross-dataset replay spoofing attack detection than that without DAT.

3.4.3. Effects of target-domain data amount in LCNN-DAT

The whole target-domain training set is used for domain adversarial training in Section 3.4.2. Here we randomly divide it into five folds and then use the first 1, 2, 3, 4 and 5 folds as the unlabeled target-domain training data respectively, which ensures that the smaller training set is a subset of the larger one.

Figure 2 shows the results of all systems. Significant cross-domain performance improvements are obtained regardless of the target-domain data amount used in all cases. However, a tendency is seen that the LCNN-DAT models generalize better across domains using more target-domain training data, without affecting their overall performance within the original source-domain dataset. Furthermore, the relative improvement is more significant when BTAS-PA 2016 dataset is used as the target domain rather than the ASVspoof 2017 V.2 dataset. The reasons are probably that the dataset size of B-train is more than twice of that of A-train, thus effectively helping the LCNN-DAT models to learn better from more target-domain data and achieve better cross-domain performance.

4. Conclusions

In order to address the domain-mismatch problem for replay spoofing attack detection, we propose a domain adversarial training architecture on unsupervised domain adaptation by using extra unlabeled target-domain training data. Via the adversarial training between the feature extractor and the domain classifier, the DAT models learn features that are discriminative in spoofing detection but indistinguishable across different domains. Experiments conducted on the ASVspoof 2017 V.2 dataset and BTAS-PA 2016 dataset show that the proposed LCNN-based DAT (LCNN-DAT) framework generalizes better across datasets than the LCNN model, with an over relative 30% EER reduction if using the whole target-domain training set. Furthermore, better cross-domain performance tends to be obtained by the LCNN-DAT models if more unlabeled target-domain data are used for training.

5. Acknowledgements

This work has been supported by the Major Program of National Social Science Foundation of China (No.18ZDA293). Experiments have been carried out on the PI supercomputer at Shanghai Jiao Tong University.

6. References

[1] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 4052–4056.

[2] Y. Liu, Y. Qian, N. Chen, T. Fu, Y. Zhang, and K. Yu, "Deep feature for text-dependent speaker verification," *Speech Communication*, vol. 73, pp. 1–13, 2015.

[3] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," *Proc. Interspeech 2017*, pp. 999–1003, 2017.

[4] Z. Huang, S. Wang, and K. Yu, "Angular softmax for short-duration text-independent speaker verification," in *Proc. InterSpeech*, September 2018.

[5] H. Sailor, M. Kamble, and H. Patil, "Auditory filterbank learning for temporal modulation features in replay spoof speech detection," in *Proc. Interspeech*, 2018, pp. 666–670.

[6] B. Wickramasinghe, S. Irtza, E. Ambikairajah, and J. Epps, "Frequency domain linear prediction features for replay spoofing attack detection."

[7] T. Gunendradasan, B. Wickramasinghe, N. P. Le, E. Ambikairajah, and J. Epps, "Detection of replay-spoofing attacks using frequency modulation features."

[8] P. A. Tapkir and H. A. Patil, "Novel empirical mode decomposition cepstral features for replay spoof detection."

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

[10] 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.

[11] C.-I. Lai, A. Abad, K. Richmond, J. Yamagishi, N. Dehak, and S. King, "Attentive filtering networks for audio replay attack detection," *CoRR*, vol. abs/1810.13048, 2018.

[12] H. Dinkel, N. Chen, Y. Qian, and K. Yu, "End-to-end spoofing detection with raw waveform cldnns," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 4860–4864.

[13] 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.

[14] 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.

[15] P. Korshunov and S. Marcel, "Cross-database evaluation of audio-based spoofing detection systems," Tech. Rep., 2016.

[16] Z. Wu, T. Kinnunen, N. Evans, J. Yamagishi, C. Hanilçi, M. Sahidullah, and A. Sizov, "Asvspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[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] K. Sohn, S. Liu, G. Zhong, X. Yu, M.-H. Yang, and M. Chandraker, "Unsupervised domain adaptation for face recognition in unlabeled videos," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3210–3218.

[19] Q. Wang, W. Rao, S. Sun, L. Xie, E. S. Chng, and H. Li, "Unsupervised domain adaptation via domain adversarial training for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4889–4893.

[20] 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.

[21] 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.

[22] 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.

[23] B. McFee, M. McVicar, O. Nieto, S. Balke, C. Thome, D. Liang, E. Battenberg, J. Moore, R. Bittner, R. Yamamoto *et al.*, "librosa 0.5. 0," 2017.

[24] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," IEEE Signal Processing Society, Tech. Rep., 2011.

[25] 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.

[26] 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.