Odyssey 2020 The Speaker and Language Recognition Workshop
1-5 November 2020, Tokyo, Japan

A Multi-condition Training Strategy for Countermeasures Against Spoofing Attacks to Speaker Recognizers

João Monteiro[1,2], *Jahangir Alam*[1,2], *Tiago H. Falk*[1]

[1]Institut National de la Recherche Scientifique (INRS-EMT), Quebec, Canada
[2]Centre de Recherche Informatique de Montréal (CRIM), Quebec, Canada

joao.monteiro@emt.inrs.ca, jahangir.alam@crim.ca, falk@emt.inrs.ca

Abstract

In this contribution, we are concerned with the design of effective strategies to train simple-to-use detectors of spoofing attacks to automatic speaker recognizers, i.e., systems able to directly map data into scores indicating the likelihood of an attack, as opposed to complex pipelines involving several independent steps required for training and inference. As such, given that artificial neural networks have been responsible for the shift from pipelines to end-to-end systems within several applications, we specifically target this kind of model. The main challenge in training neural networks for the applications considered herein lies in the fact that openly available spoofing corpora are relatively small due to the inherent difficulty involved in collecting/generating this kind of data. We thus employ a data augmentation strategy enabling the introduction of training examples which significantly improves training data in terms of size and diversity. Neural networks trained on top of augmented training data are shown to be able to attain significant improvement in terms of detection performance when compared to standard GMM-based classifiers.

1. Introduction

The recent rise in the application of artificial neural networks has led to significant performance shifts across several fields, such as object recognition in images [1], generative modeling [2, 3, 4], speech [5, 6] and speaker recognition [7], to name a few. However, such modeling strategy was observed to present properties that might be exploited by attackers at inference time. Specifically, the outputs of such models can greatly change given just subtle variations in the inputs. These so-called adversarial attacks [8] showed that one can leverage inherent properties in order to fool trained models in such a way that attack instances are not detectable by human observers. The described vulnerability constitutes one of the major factors limiting the vast deployment of neural network based technologies into real world scenarios. While most of the recent work on the development of adversarial attacks and defenses have targeted computer vision applications, it was recently shown that the same threat appears in the case of models tailored to speech processing applications, such as speech and speaker recognition [9, 10].

Besides the above described threat of adversarial attacks, which affect applications of neural networks in general, for the specific case of voice biometrics, even simpler attack strategies exist which can be applied to any type of model. One example is replaying someone's voice. As a simple and realistic example, one could record someone's voice saying a command to gain unauthorized access to their portable devices. Another strategy could be with synthesized speech. Recent advances in conditional generative models (e.g., Wavenet [4]) can be used to this end for both text-to-speech or voice conversion settings. The potential consequences of such vulnerabilities are tremendous and range from financial loss to undue incrimination.

Given that these described threats limit the use of high performance systems in real applications, a popular research direction in recent years has been to design countermeasures against such attackers. For the specific case of speaker recognition and voice authentication, for instance, recent spoofing detection challenges [11, 12, 13] were introduced with the goal of pushing forward the state-of-the-art in attack detection. In general, countermeasures can be grouped into two categories: defense and detection methods. While defense techniques try to either improve the model robustness or suppress the success rates of attacks, detection methods take a different approach and attempt to determine if the input is genuine or was somehow manipulated. While both directions are promising, in this paper we focus on the detection approach, as it allows service providers to know when their systems are under attack.

More specifically, we are interested in detection approaches that operate in an end-to-end fashion. Here, end-to-end is referred to systems comprised of a single component able to receive audio as input (or general purpose audio representations) and output scores indicating how likely it is that the input has been tampered. The main advantage of end-to-end settings over conventional pipeline-based methods that rely on several internal blocks, is simplicity. End-to-end systems allow for inference schemes that require a single forward pass, while pipeline methods usually have to deal with several blocks, each one with their specific challenges and limitations. Figure 1 presents a block diagram corresponding to a spoofing detector in tandem with a speaker verification system. In this case, the input corresponds to an audio signal along with a claimed identity. The spoofing detector is only used if the claimed identity is verified as matching the claimed the audio's.

The underlying principle assumed here is that any attack strategy to a voice biometrics systems will generate detectable artifacts, be it room colouration from playback attacks, or unnatural speech characteristics from synthesized speech. Here, we build on previous work [14] by introducing approaches to cleverly augment the available training data in such a manner that does not affect the discriminability of corrupted samples with respect to genuine ones. We show that such augmentation schemes are critical for the task at hand, thus allowing us to effectively train end-to-end detectors using relatively large convolutional models.

In particular, we implement the detector as the well-known time-delay architecture [15] employed within the x-vector setting, as it showed to be effective in summarizing speech into

10.21437/Odyssey.2020-42

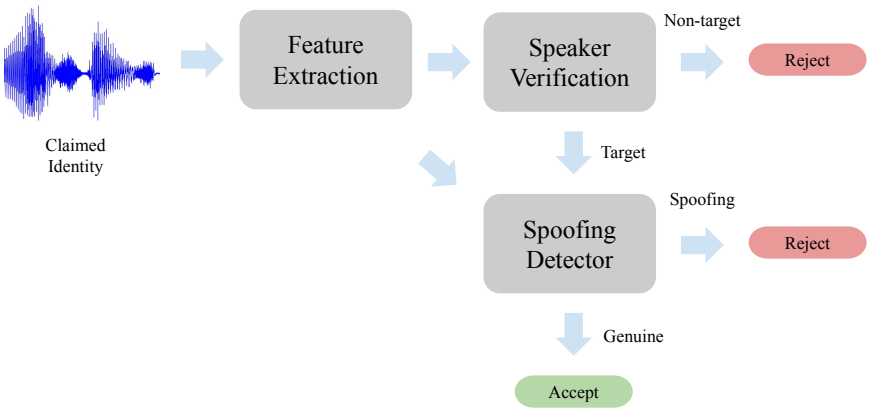

Figure 1: A speaker verification system used alongside with a spoofing detector. Trials accepted by the verification system are further scored by the detector. Only test recordings classified as genuine can be finally accepted.

speaker- and spoken language-dependent representations [7, 16]. The model consists of a sequence of dilated 1-dimensional convolutions across the temporal dimension, followed by a time pooling layer, which simply concatenates element-wise first- and second-order statistics over time. Statistics are finally projected into an output score through fully-connected layers. Speech is represented as general purpose spectral features, which we will describe later on.

The remainder of this paper is organized as follows: Related work is discussed in Section 2. Section 3 describes the approaches employed in order to increase the size and diversity of the available training corpora so as to ensure generalization to unseen test data. Section 4 presents details about training and the model architecture. Evaluation is then presented in Section 5 and conclusions are drawn in Section 6.

2. Related work

Generative classifiers introduced in [17] have been adopted in most countermeasure works. The approach is similar to that of linear discriminant analysis (LDA) so that frame-level generative models are trained one per class. The main difference with respect to LDA is the use of Gaussian Mixture Models (GMM) for modelling the class-conditional features, rather than simple Gaussians with shared covariances as in the LDA case. The focus of the speaker recognizer spoofing attack detection literature then became that of finding optimal speech representations (or features) to capture the salient the artifacts introduced by the various spoofing methods. These representations were then used alongside GMM classifiers.

To this end, several low-level representations of speech signals were explored, roughly falling under two categories: (i) spectral amplitude and phase [18, 19, 20, 21, 22, 23], and (ii) combined amplitude-phase [19, 20, 21]. For example, in [18], a GMM classifier trained on cepstral coefficients computed after a constant Q transform (CQCC) were used. A variant based on infinite impulse response constant Q-transform spectrum (IIR-CQT) was later introduced in [19]. Cochlear filter cepstral coefficients and excitation source-based features were evaluated in [22] and [23], respectively.

Another common strategy used corresponds to the so-called tandem representations, where a frame-level neural network is trained for spoofing detection, and the GMM classifier is then trained on inner layer outputs, predicted posteriors, and speech features [19, 20]. Similarly, in [24] a neural network is trained in a supervised fashion at the frame level and statistics of intermediate representations are used as global descriptors of full recordings. Independently-trained fully-connected neural networks on top of different speech features have been explored in [25] and [26]. Models receive concatenated frames as inputs and classify a full recording as genuine or spoofing. A score-level fusion of the set of detectors is then employed for final decision.

Closer to what is proposed here, recent contributions have tackled the spoofing detection problem in an end-to-end fashion. In [27], a convolutional model is trained on top of raw audio for the detection of replayed attackers so that the model is able to learn representations which will render attackers detectable. In [28], an attentive end-to-end scheme is introduced such that a U-net structure is first employed to map the input features into a set of element-wise importance weights and the weighted input is fed into a set of feed-forward layers to yield scores. In our past contribution, convolutional neural networks were shown to be effective for direct end-to-end detection in [14].

3. Data Augmentation

3.1. Speech representation

We employ two types of general-purpose time-frequency speech representations. Both spectral representations and cepstral coefficients are considered, and the same modeling strategy is used in both cases. Specifically, we report results for models trained on top of the product spectrum (ProdSpec) introduced in [29] and later used in [30], and linear frequency cepstral coefficients [31] (LFCC). In all cases, features are obtained with a short-time Fourier transform with length 512 using a 20ms Hamming window with 50% overlap. The end result are 257 frequency bins for spectral representations and 30 coefficients stacked with *delta* and *delta-delta* coefficients for the LFCC case, thus resulting in a dimensionality of 90. Following our previous findings [14], we employ ProdSpec and LFCC to train detectors of replay and synthetic (generative) attacks

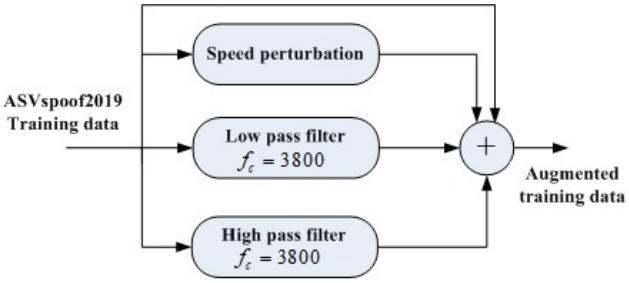

Figure 2: Data augmentation via speed perturbation, low pass, and high pass filtering of ASVspoof 2019 training data.

given that we empirically found those to yield the best detectors in each case.

3.2. Augmentation approach

Data augmentation can be defined as the process of increasing the amount and diversity of existing training data, and it is performed with two goals: (i) as a form of regularization strategy so as to improve the performance measured on test data by directly increasing the number of training examples, and (ii) to compensate possible mismatches between the training and test conditions, which can be seen as a way of inducing robustness across varying noise conditions. For the case of speech processing applications, augmentation is often performed by supplementing a dataset with similar data, artificially created by introducing additive and convolutive noises, for instance. Augmentation schemes became common practice in speech-based applications, such as speaker and speech recognition.

In this work, we perform evaluation using the data introduced for the ASVspoof 2019 challenge [13]. Specifically, we are particularly interested in being able to introduce perturbations in the signals in such a way that the artifacts related to spoofing attacks are preserved, while at the same time the artifacts we introduce with the augmentation process are different from those indicative of attacks. For instance, we empirically found that, depending on the type of attack strategy, this can be achieved through speed perturbations, as well as band-pass filtering, and those offer the extra advantage of not requiring any external corpora to be used, complying with evaluation guidelines of popular challenges such as ASVSpoof. We further observed simple online strategies such as feeding models with random continuous chunks of signals (as opposed to presenting the complete audio recording) to be effective as an augmentation strategy. The approach we evaluate to increase the amount of training data, both bona fide as well as spoofing examples, is to apply speed perturbations, in which case we use perturbation factors of 0.9 and 1.1, to low pass filter with a cut-off frequency of 3.8 kHz, and to high pass filter with the same cut-off frequency set at 3.8 kHz. The described procedure is summarized in Figure 2. By doing so, we are able to increase the size of our corpus by five times with respect to the original training data. Along with that, we perform additional online transformations which will be further described once we detail how mini-batches are constructed at training time.

Table 1: Employed TDNN architecture. T indicates the duration of features in number of frames and N the feature vector dimensionality.

Layer	*Input Dimension*	*Output dimension*
Conv1d+ReLU	N $\times$ T	512 $\times$ T
Conv1d+ReLU	512 $\times$ T	512 $\times$ T
Conv1d+ReLU	512 $\times$ T	512 $\times$ T
Conv1d+ReLU	512 $\times$ T	512 $\times$ T
Conv1d+ReLU	512 $\times$ T	1500 $\times$ T
Statistical Pooling	1500 $\times$ T	3000
Linear+ReLU	3000 $\times$ T	512
Linear+ReLU	512	512
Linear+ReLU	512	1

4. Proposed Method

4.1. Model description

We implement our model using the well-known TDNN architecture employed within the x-vector setting [7] for speaker verification. The model is made up of of a sequence of dilated 1-dimensional convolutions across the temporal dimension, followed by a time pooling layer, which simply concatenates element-wise first- and second-order statistics over time. Concatenated statistics are finally projected into an output vector through two fully-connected layers, and a final fully-connected layer is employed to get the final score. Following the setting introduced in [32] as well as based on the results in [33] showing that classification accuracy is usually better when batch normalization is applied prior to activation, we modified the standard TDNN architecture so that pre-activation batch normalization is performed at each convolution and fully-connected layer. A summary of the employed architecture is shown in Table 1 while the complete end-to-end score computation is illustrated in Figure 3.

4.2. Training details

Training is carried out closely following the setup we introduced in [14], i.e., with Stochastic Gradient Descent performed to minimize the binary cross-entropy loss in a standard binary classification setting. Mini-batches of effective size of 16 (i.e., balanced) are used for both of the cases of ProdSpec and LFCCs. The learning rate and weight decay coefficients were set at $1\mathrm{e}{-3}$ and $5\mathrm{e}{-5}$, and performance is monitored throughout training with a validation set we created by removing 100 and 1000 randomly selected recordings corresponding to clean and attack examples, respectively. Polyak's momentum is also employed with its coefficient set at the default value of 0.9.

We further perform an extra data augmentation strategy in an online fashion, which conveniently also helps to deal with the varying duration across recordings. Each training example is pre-processed such that if it is shorter than 10 seconds, it is repeated up to that length. In the case it is longer, we select a random 10 second segment and this process is repeated whenever a given example is sampled. Since we employ that procedure and ensure all examples within a mini-batch are exactly 10 seconds long, we randomly trim all examples to be within 3-10 seconds, where the exact duration is sampled uniformly from that range for every mini-match. This is done so as to generate diversity in the samples in terms of duration since trained models are expected to be able to perform detection given samples of arbitrary duration at test time.

Additionally, we observed a significant imbalance in the

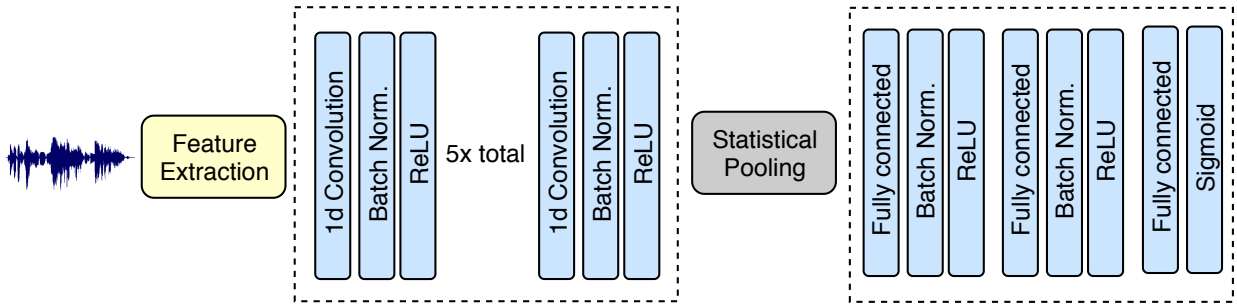

Figure 3: End-to-end detection of spoofing attacks. All convolutions maintain the time resolution. Statistical pooling corresponds to concatenated mean and variance, obtained across the time dimension. N represents the dimension of feature vectors, which correspond to 90 and 257 for LFCC and ProdSpec, respectively.

number of examples corresponding to genuine and spoofing training examples. As such, we oversample genuine examples such that each mini-batch is balanced. We do so by sampling pairs of training examples. We sequentially iterate over the set of recordings corresponding to attacks and, to sample clean recordings, we pick the recording with index $k = j \bmod N_{clean}$, where $j \in \{0, 1, ..., N_{attack} - 1\}$ is the index of attack recordings, and N_{clean} and N_{attack} are the number of clean and spoofing training recordings.

The sequence of indices j is further randomized to provided diverse mini-batches at each epoch. Training proceeds up to a fixed budget of epochs or convergence of the loss measured in the validation set held out of training. This took approximately 12 hours in a single NVIDIA Titan X GPU[1].

5. Experimental Setup and Evaluation

5.1. Dataset and evaluation metrics

The proposed methods are evaluated using the data introduced for the ASVspoof 2019 challenge. Two types of attacks are considered: logical (LA) and physical access (PA) attacks, corresponding to synthetic speech and replayed recordings, respectively. Logical access attacks were created using both voice conversion and text-to-speech systems, while replay attacks are simulated from clean recordings considering exhaustive combinations of three room sizes, three distances to the microphone, and three levels of reverberation. The number of recordings for each case is shown in Table 2 for both training and development partitions. There is no overlap between speakers within train and development data, and no information regarding speaker identities is used in any part of the proposed approach.

5.2. Evaluation

Evaluation on both development and test partitions of ASVSpoof 2019 is performed in terms of the equal error rate (EER) and normalized minimum tandem detection cost function (min-tDCF). EER consists of the value of the *miss rate*, given by the fraction of miss-classified clean test recordings with respect to the number of trials corresponding to genuine samples, at the threshold in which it matches the *false alarm rate*, i.e., the ratio between the number of miss-classified spoofing trials and the number of spoofing test recordings. The min-tDCF, in turn, was recently introduced in [34], and was designed especially for the evaluation of spoofing detection countermeasures when used alongside a speaker recognizer. Refer to [13] for a detailed description regarding both the corpora and evaluation metrics. Since we created our validation sets through removing data from the training partition, the development data is not used in any way during training nor for early stopping or hyperparameter selection. All results are reported for the best performing model we could observe across training in each data partition during a fixed budget of 100 training epochs.

Table 2: Number of bona fide (genuine) and spoofing recordings contained in training and development partitions for both logical and physical access attacks.

	# Speakers	# Recordings			
		Logical Access		Physical Access	
		Bona fide	Spoof	Bona fide	Spoof
Train	20	2580	22800	5400	48600
Dev.	20	2548	22296	5400	24300

In addition to our proposed systems, we further report the performance of baselines for comparison, which include: (i) those provided by the ASVspoof 2019 organizers, which consist of GMM-based classifiers trained on frame level features, reference performance is provided for GMMs trained on both LFCC as well as CQT-based cepstral coefficients (CQCC) [18]. And (ii) our own baselines obtained using the Kaldi toolkit [35] corresponding to a GMM classifier on top of CQCCs, as well as an i-vector system [36] scored with a probabilistic linear discriminant analysis (PLDA) classifier [37]. Both the GMM- and i-vector-based systems use 512-Gaussian components for training bonafide/spoof models and the universal background model (UBM), respectively. I-vectors were finally obtained with total-variability performed on top of UBM's super vectors, yielding 400-dimensional representations of audio clips, using CQCC features. Prior to training PLDA, i-vectors further have their dimensionality reduced to 150 using linear discriminant analysis.

Tables 3 and 4 present the EER and min-tDCF scores obtained by the baselines, as well as our proposed systems for the PA and LA tasks, respectively, considering the development

[1]Code is available at: https://github.com/joaomonteirof/e2e_antispoofing

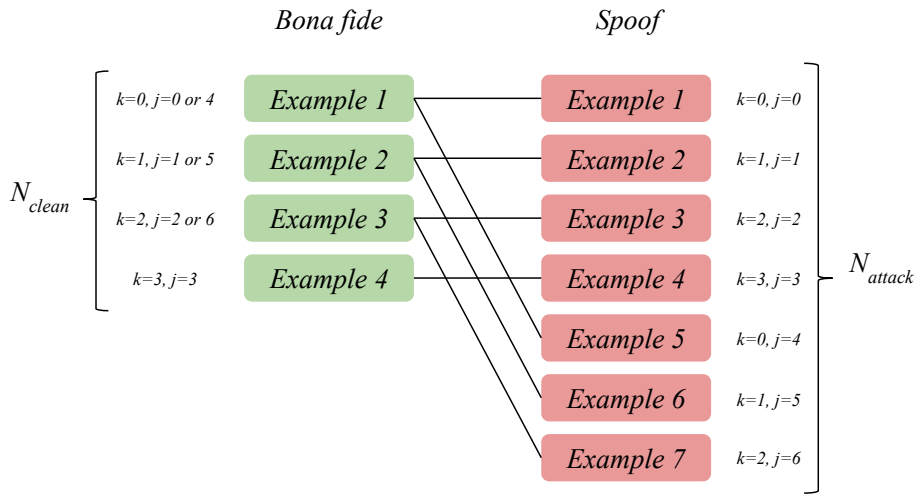

Figure 4: Sampling strategy for constructing mini-batches. Clean examples are sampled several times per epoch so as to ensure mini-batches are balanced.

Table 3: The min-tDCF and EER(%) results for PA task on the development set. Lower values are better.

	System	***EER(%)***	***min-tDCF***
ASVSpoof baselines [13]	CQCC-GMM	9.87	0.1953
	LFCC-GMM	11.96	0.2554
Our baselines	CQCC-GMM	9.70	0.1840
	i-vector/PLDA	9.17	0.2310
Ours	TDNN	2.30	0.0690
	TDNN+SP	0.87	0.0299
	TDNN+Filt.	0.74	0.0244
	TDNN+SP+Filt.	1.34	0.0439

Table 4: The min-tDCF and EER(%) results for the LA task on the development set. Lower values are better.

	System	***EER(%)***	***min-tDCF***
ASVSpoof baselines [13]	CQCC-GMM	0.43	0.0123
	LFCC-GMM	2.71	0.0663
Our baselines	CQCC-GMM	0.39	0.0104
	i-vector/PLDA	0.70	0.0211
Ours	TDNN	0.07	0.0015
	TDNN+SP	0.04	0.0012
	TDNN+Filt.	0.08	0.0011
	TDNN+SP+Filt.	0.08	0.0018

data. Proposed models are indicated by *TDNN*, *TDNN+SP*, *TDNN+Filt*, and *TDNN+SP+Filt.*, which corresponds to: models trained with the original training corpus and the previously described online augementation strategy performed while assembling mini-bacthes, offline augmented train data with speed perturbation only, bandpass filtering only and augmented data with both speed perturbation and bandpass filtering.

In both PA and LA cases, end-to-end approaches are able to outperform more standard GMM-classifers, as well as our i-vector/PLDA system in terms of both EER and min-tDCF. Moreover, specifically for replay attacks as presented in Table 3, we observe that adding speed perturbations yielded an improvement in the detection performance. However, once the complete set of augmentations is employed, we actually observe a degradation in performance. Nevertheless, as will be further discussed, the opposite is observed in the case of evaluation data (c.f. Table 5), which suggests that the mismatch across train and development data is much smaller than that across train and evaluation data. The observed performance degradation is thus an indication of reduced overfitting to train data, which is beneficial given the improved generalization observed when evaluation data is used to assess performance.

Performance on the evaluation data is reported in Tables 5 and 6 for PA and LA attacks, respectively. In this case, we once more observe the proposed end-to-end approaches outperforming the considered baselines. However, for PA attacks specifically, we now observe that the use of more data augmentation consistently implies improved detection performance, which confirms our hypothesis in that the more diverse train data introduces some sort of regularization and avoids overfitting to the types of attack strategies utilized to create train data.

We further stress the observation made above considering the mismatch in results observed between development and evaluation data for all considered systems, baseline or otherwise, for the specific case of LA attacks. In fact, for that particular evaluation case, most systems reach a strong detection performance on development data, while observe a more severe degradation when we move to evaluation data. This is due to the different approaches used to create attacks so as to compose both data partitions; i.e., generative approaches used to create the development partition are similar to those used to create train data. The online augmentation helps in this regard, working as a regularization strategy and enabling better generalization to the new conditions introduced with the evaluation data. However, the artifacts introduced with speed perturbation as well as bandpass filtering appear to overlap with those introduced by the speech synthesis approaches utilized in order to create the attacks, and thus yield a slight degradation in performance, but not due to overfitting in this case. In fact, we hypothesize the opposite happens for LA, since the augmented genuine samples appear alike to synthetic attackers, an effect

Table 5: The min-tDCF and EER(%) results for PA task on the evaluation test set. Lower values are better.

	System	*EER(%)*	*min-tDCF*
ASVSpoof baselines [13]	CQCC-GMM	11.04	0.2454
	LFCC-GMM	13.54	0.3017
Our baselines	CQCC-GMM	11.16	0.2478
	ivector/PLDA	10.18	0.2687
Ours	TDNN	4.46	0.1337
	TDNN+SP	2.18	0.0777
	TDNN+Filt.	1.84	0.0611
	TDNN+SP+Filt.	1.77	0.0597

Table 6: The min-tDCF and EER(%) results for LA task on the evaluation test set. The lower the values of min-tDCF and EER the better is the performance.

	System	*EER(%)*	*min-tDCF*
ASVSpoof baselines [13]	CQCC-GMM	9.57	0.2366
	LFCC-GMM	8.09	0.2116
Our baselines	CQCC-GMM	8.91	0.2157
	ivector/PLDA	16.55	0.4201
Ours	TDNN	7.00	0.1653
	TDNN+SP	8.89	0.1769
	TDNN+Filt.	8.22	0.1769
	TDNN+SP+Filt.	7.12	0.1674

similar to that of label noise is introduced, yielding a too strong regularization strategy for this particular evaluation. The end-to-end models we trained are nevertheless able to outperform considered baselines by a large amount.

6. Conclusion

In this work, we evaluated different strategies aimed at augmenting the training data in order allow for end-to-end spoofing attacks to be detected for speaker recognizers. By doing so, we were able to increase the size of available corpora by a factor of five, while making it more diverse, thus introducing regularization effects that improved generalization to novel conditions.

The main challenge with data augmentation for spoofing lies in the fact that signal transformations employed must preserve the artifacts introduced by the attack strategies, given that those are needed in order to discriminate genuine and spoofing samples. For the spoofing methods tested herein, we observed that speed perturbations and bandpass filtering satisfied these requirements for replay attacks, yielding a simple and efficient approach to greatly increase the amount and diversity of available train data, while simpler trimming across time strategy was more helpful for the case of LA attacks. While we decided to take an offline approach to perform part of the considered audio perturbations, and created corrupted copies of the data prior to actually training models, the proposed transformations are efficient enough to enable a complete online implementation, performed on-the-fly while models are being updated. This can potentially further increase the diversity of train data and make its size virtually unbounded since data instances appear in a different version every time they are sampled.

As opposed to most past contributions, which focused on simple classification pipelines or small neural networks, via data augmentation we were able to effectively train relatively large convolutional models. Experiments with the ASVspoof 2019 Challenge dataset showed that trained TDNNs were able to outperform legacy approaches for both synthetic and replay attack methods.

7. Acknowledgements

The authors wish to acknowledge funding from the Natural Sciences and Engineering Research Council of Canada (NSERC) through contract/grant RGPIN-2019-05381, RGPIN-2016-4175, and RGPAS-493010-2016. The first author was supported by the bourse du CRIM pour etudes supérieures. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the NSERC.

8. References

[1] A. Krizhevsky, I., and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[2] D. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[4] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[5] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio, "End-to-end attention-based large vocabulary speech recognition," in *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2016, pp. 4945–4949.

[6] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4960–4964.

[7] 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.

[8] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," *ArXiv e-prints*, Dec. 2014.

[9] N. Carlini and D. Wagner, "Audio Adversarial Examples: Targeted Attacks on Speech-to-Text," *ArXiv e-prints*, Jan. 2018.

[10] F. Kreuk, Y. Adi, M. Cisse, and J. Keshet, "Fooling End-to-end Speaker Verification by Adversarial Examples," *ArXiv e-prints*, Jan. 2018.

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

[12] 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," *Proc. Interspeech 2017*, pp. 2–6, 2017.

[13] 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.

[14] João Monteiro, Jahangir Alam, and Tiago H Falk, "End-to-end detection of attacks to automatic speaker recognizers with time-attentive light convolutional neural networks," in *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2019, pp. 1–6.

[15] Alex Waibel, Toshiyuki Hanazawa, Geoffrey Hinton, Kiyohiro Shikano, and Kevin J Lang, "Phoneme recognition using time-delay neural networks," *IEEE transactions on acoustics, speech, and signal processing*, vol. 37, no. 3, pp. 328–339, 1989.

[16] David Snyder, Daniel Garcia-Romero, Alan McCree, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "Spoken language recognition using x-vectors.," in *Odyssey*, 2018, pp. 105–111.

[17] 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.

[18] Massimiliano Todisco, Héctor Delgado, and Nicholas Evans, "A new feature for automatic speaker verification anti-spoofing: Constant q cepstral coefficients," in *Speaker Odyssey Workshop, Bilbao, Spain*, 2016, vol. 25, pp. 249–252.

[19] Jahangir Alam and Patrick Kenny, "Spoofing detection employing infinite impulse response—constant q transform-based feature representations," in *2017 25th European Signal Processing Conference (EUSIPCO)*. IEEE, 2017, pp. 101–105.

[20] Md Jahangir Alam, Patrick Kenny, Vishwa Gupta, and Themos Stafylakis, "Spoofing detection on the asvspoof2015 challenge corpus employing deep neural networks," in *Proc. Odyssey*, 2016, pp. 270–276.

[21] Md Jahangir Alam, Patrick Kenny, Gautam Bhattacharya, and Themos Stafylakis, "Development of crim system for the automatic speaker verification spoofing and countermeasures challenge 2015," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[22] Tanvina B Patel and Hemant A Patil, "Combining evidences from mel cepstral, cochlear filter cepstral and instantaneous frequency features for detection of natural vs. spoofed speech," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[23] Tanvina B Patel and Hemant A Patil, "Effectiveness of fundamental frequency (f 0) and strength of excitation (soe) for spoofed speech detection," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5105–5109.

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

[25] Xiong Xiao, Xiaohai Tian, Steven Du, Haihua Xu, Eng Siong Chng, and Haizhou Li, "Spoofing speech detection using high dimensional magnitude and phase features: The ntu approach for asvspoof 2015 challenge," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[26] Xiaohai Tian, Zhizheng Wu, Xiong Xiao, Eng Siong Chng, and Haizhou Li, "Spoofing detection from a feature representation perspective," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 2119–2123.

[27] Hannah Muckenhirn, Mathew Magimai-Doss, and Sébastien Marcel, "End-to-end convolutional neural network-based voice presentation attack detection," in *2017 IEEE international joint conference on biometrics (IJCB)*. IEEE, 2017, pp. 335–341.

[28] 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.

[29] Donglai Zhu and Kuldip K Paliwal, "Product of power spectrum and group delay function for speech recognition," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 2004, vol. 1, pp. I–125.

[30] Md Jahangir Alam, Gautam Bhattacharya, and Patrick Kenny, "Boosting the performance of spoofing detection systems on replay attacks using q-logarithm domain feature normalization," in *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 393–398.

[31] Md Sahidullah, Tomi Kinnunen, and Cemal Hanilçi, "A comparison of features for synthetic speech detection," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[32] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[33] Hossein Zeinali, Luka Burget, Johan Rohdin, Themos Stafylakis, and Jan Honza Cernocky, "How to improve your speaker embeddings extractor in generic toolkits," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6141–6145.

[34] Tomi Kinnunen, Kong Aik Lee, Héctor Delgado, Nicholas Evans, Massimiliano Todisco, Md Sahidullah, Junichi Yamagishi, and Douglas A Reynolds, "t-dcf: a detection cost function for the tandem assessment of spoofing countermeasures and automatic speaker verification," *arXiv preprint arXiv:1804.09618*, 2018.

[35] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011, number CONF.

[36] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[37] Simon JD Prince and James H Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *IEEE 11th International Conference on Computer Vision (ICCV)*, 2007, pp. 1–8.