INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Replay Attack Detection using DNN for Channel Discrimination

Parav Nagarsheth, Elie Khoury, Kailash Patil, Matt Garland

Pindrop, Atlanta, USA

{pnagarsheth,ekhoury,kpatil,matt.garland}@pindrop.com

Abstract

Voice is projected to be the next input interface for portable devices. The increased use of audio interfaces can be mainly attributed to the success of speech and speaker recognition technologies. With these advances comes the risk of criminal threats where attackers are reportedly trying to access sensitive information using diverse voice spoofing techniques. Among them, replay attacks pose a real challenge to voice biometrics. This paper addresses the problem by proposing a deep learning architecture in tandem with low-level cepstral features. We investigate the use of a deep neural network (DNN) to discriminate between the different channel conditions available in the ASVSpoof 2017 dataset, namely recording, playback and session conditions. The high-level feature vectors derived from this network are used to discriminate between genuine and spoofed audio. Two kinds of low-level features are utilized: state-of-the-art constant-Q cepstral coefficients (CQCC), and our proposed high-frequency cepstral coefficients (HFCC) that derive from the high-frequency spectrum of the audio. The fusion of both features proved to be effective in generalizing well across diverse replay attacks seen in the evaluation of the ASVSpoof 2017 challenge, with an equal error rate of 11.5%, that is 53% better than the baseline Gaussian Mixture Model (GMM) applied on CQCC.

Index Terms: replay attacks, speaker verification, spoofing challenge

1. Introduction

Automatic Speaker Verification (ASV) systems are being increasingly challenged by spoofing techniques like voice conversion, speech synthesis and replay attacks. Among these, replay attacks, also known as *presentation attacks*, have been shown to reduce the accuracy of ASV systems by significant margins. Replay attacks are easy to generate with no expertise required in speech processing and machine learning. With use of high-quality playback and recording devices, it is conceivable to make replay attacks indistinguishable from a genuine access. Under replay attacks, the false acceptance rate of state-of-the-art ASV systems evaluated in [1] and [2] reach 77% and 66%, respectively.

In the last decade, several researchers have tackled the problem of spoofing attacks and have proposed different countermeasures targeting either text-dependent or text-independent ASV systems.

In [3], the problem of replay attacks detection was reduced to the problem of far-field detection. They found that the spectral ratio, low-frequency ratio, and modulation index carry relevant information to distinguish between genuine and far-field speech. They address this problem for the sub-category of text-independent ASV.

More countermeasures for replay attacks on text-independent ASV systems were proposed in [4] and evaluated on the AVspoof dataset [1]. Their baseline system uses spectrogram-based ratios and a logistic regression classifier. The second system uses two different cepstral features and a binary neural network classifier that classifies between genuine and spoofing attacks. The third system uses normalized PLP and two deep network architectures: feed forward DNN and bidirectional long short term memory network (BLSTM). Both architectures have four output classes: one for genuine speech and the remaining three for attacks present in the dataset, namely voice conversion, speech synthesis and replay attacks. The fourth system uses long-term spectral mean and standard deviation [5] to generate a feature vector that is in turn used in a linear discriminant analysis framework to discriminate between genuine speech and spoofing attacks. The last system in [4] uses mel frequency cepstrum coefficients (MFCC) and inverted MFCC (IMFCC) features to train two GMMs: one for genuine speech and one for spoofing attacks.

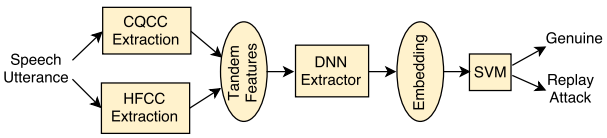

Figure 1: *Proposed Replay Attack Detection.*

In [6], a method based on spectral bitmap [7] was proposed to detect replay attacks presented for a text-dependent ASV system. Similarly, the ASVspoof 2017 challenge[1] [8] focuses on text-dependent speaker recognition. The challenge aims to assess and enhance the security of ASV systems against wide variety of replay attacks. In fact, the organizers have used the RedDots [9] dataset to generate replay attacks under different playback, recording and environmental conditions [10]. The speech utterances are of short duration and variable phonetic content, and were previously used to evaluate text-dependent speaker recognition systems [9]. When replacing the zero-effort impostors with replay attacks, empirical results in [8] have shown that the accuracy of a state-of-the-art text-dependent ASV system drops dramatically with an EER reaching 42%. A baseline system is provided as part of the challenge. This system uses Constant-Q Cepstral Coefficients (CQCCs) [11], that are multi-resolution cepstral features derived from the perceptually inspired Constant-Q transform (CQT). The classification is done using GMM modeling.

Although the text is provided as part of the ASVspoof challenge, and thus, can be used as an additional cue to detect replay attacks, we have investigated a general countermeasure solution that works for both text-dependent and text-independent speaker recognition systems.

The proposed replay attack detection is illustrated in Fig 1. Our contribution is two-fold: the first is at the signal level,

[1] http://www.spoofingchallenge.org

http://dx.doi.org/10.21437/Interspeech.2017-1377

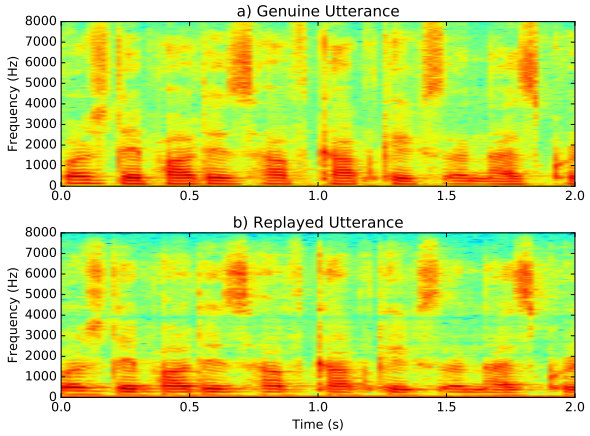

Figure 2: *Spectrogram of genuine and replayed audio of the utterance "Birthday parties have cupcakes and ice-cream" from the same speaker. The higher frequency bands characteristically have lower power in the replayed utterance, compared to the original utterance. Additionally, there are artifacts present in low and mid-range frequencies.*

where we propose a new set of features, coined "high-frequency cepstral coefficients" (HFCC), that capture channel characteristics in the non-speech region of the spectrum and that were found to provide superior accuracy when used in tandem with CQCC. The second is at the modeling level, where we propose a DNN feature extractor that is trained to discriminate between different conditions due to changes in playback, recording and environmental conditions.

The remainder of this paper is as follows: Section 2 presents HFCC and the tandem features. Section 3 introduces our proposed front-end feature extractor. Section 4 details the experimental setup and results. Section 5 concludes this paper.

2. Low-Level Feature Extraction

In search for a new set of features for replay attacks, we analyzed the spectral differences between genuine and spoofed audio. In Figure 2, the spectrograms of genuine and replayed samples of the same utterance from the same speaker can be visually inspected. The spectral differences are particularly pronounced in the spectrum traditionally thought to have low speech content in both, higher as well as lower frequencies. In the following sections, we describe our attempt to capture these variations.

2.1. High-Frequency Cepstral Coefficients

The short-time Fourier transform (STFT) followed by cepstral extraction is commonly used as the low-level feature vector that feeds most modeling and classification approaches in general speech processing tasks. The power-spectrum of the framed speech signal is often transformed by a filter bank for dimensionality reduction. While a Mel-scale filter bank is the most popular approach for the speech recognition and speaker verification applications, it has been found that linear triangular filters [12, 13, 14] or inverse gammatone filters [14] are more suited to detect voice conversion and text-to-speech spoofing artifacts.

When designing features for replay attack detection, we reasoned that recording and playback devices designed for telephony are likely to exhibit channel artifacts in the form of attenuation or emphasis of certain frequencies outside the voice band (300-3400 Hz). These artifacts are more pronounced in low-quality recording or playback devices, while a high-quality recording-playback device pair tends to have a flatter frequency response. In this paper, we focus our investigation for new features on the high-frequency region of the spectrum outside the voiced speech frequencies.

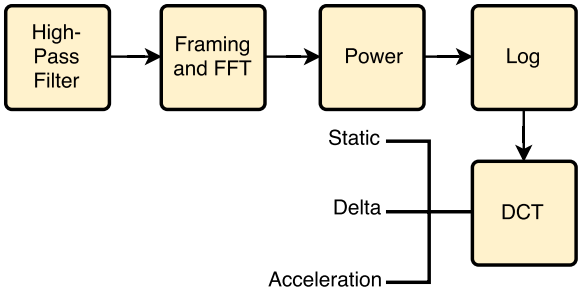

Figure 3: *HFCC Feature Extraction Pipeline.*

Figure 3 shows the feature extraction pipeline of HFCC features. In the preprocessing step, the signal is filtered using a second order high-pass butterworth filter with a suitable cutoff. The rest of the pipeline is similar to a typical cepstral extraction chain with the exception of the filterbank. The filterbank has been eliminated in favor of a high-resolution representation of all frequency bins.

The final feature vector has static, delta and acceleration (delta-delta) of the first 30 coefficients of the DCT. We found empirically that the optimal cutoff is at 3500 Hz, which attenuates lower frequencies at 12 dB per octave. This eliminates the contribution of fundamental and harmonic speech frequencies in the feature vector.

2.2. Tandem features

HFCC are designed to focus only on the high frequency region, that is the most affected by spoofing artifacts. However, our empirical results have showed that low and mid-range frequencies carry minor yet additional information to distinguish between genuine and replayed speech. Hence, we chose to use HFCC in tandem with CQCC [11] features to exploit possible complementarity in feature space. CQCCs were first introduced as a countermeasure to voice conversion and text-to-speech (TTS) attacks to ASV systems in the context of the ASVspoof 2015 challenge [15]. They were found also very useful for replay attacks, getting superior results compared to LFCC [10]. They are based on the constant-Q transform (CQT), which results in high time resolution at high frequencies and low time resolution at low frequencies.

In contrast to the original paper, we used zero-mean and unit variance normalized 30-dimensional CQCCs, along with their first and second derivatives. The means and variances are computed on the full training data. The same normalization is applied to HFCC features. We empirically found that it slightly improves the accuracy of the DNN-based system.

3. Front-End DNN Feature Extractor

Many voice spoofing countermeasures have found success in using a two GMM model (one each for genuine and spoofed classes) to generate likelihood scores. GMM modeling assumes that the data follow a mixture of Gaussians. However, this assumption is often not valid in practice unless the number of Gaussians is infinite. Thus, we propose to replace the back-end GMM modeling with a DNN. DNNs are very good at modeling the complex and non-linear shape of the data. In addition, once the DNN is trained, the inference step is relatively very fast compared to the likelihood computation in the GMM.

Following the practical success of i-vectors that benefit from recent advances in discriminative machine learning approaches and their low memory and I/O footprints, we use a DNN model as a high-level feature extractor.

While training the DNN, two different classification strategies were considered:

- The first is a binary classification strategy that distinguishes between genuine speech and replay attacks.
- The second is a multi-class classification strategy that distinguishes between various available channel conditions. In case of the ASVspoof challenge, the number of output units is equal to the number of unique replay configurations (*Playback-Recording-Environment*) that are available in the training data[2].

We found empirically that the second strategy works the best and generalizes better to unseen channel conditions. We think the reason is that the corresponding DNN embeddings capture more channel information than the binary-classification ones.

Figure 4 illustrates the DNN architecture used in this paper. The input layer to the network is an 2D images of size d x N, where d is the dimension of the feature vectors and N is the number of frames. We choose N to correspond approximately to 1 second of audio[3].

The first three hidden layers are convolutional layers. Each layer has 128 filters. The size of filters of the first convolutional layer is d x 3, while the filters of the second and third layers are of size 1 x 3. The third convolutional layer is followed by a max pooling layer that is done over the time axis.

The second set of hidden layers are three fully connected layers with 256 units each. During training, we apply dropout [16] to these three layers with a ratio of 30%. This is to prevent the DNN from overfitting. The DNN is trained with 2,000 epochs. Once the DNN is trained, the last fully connected layer will be used to generate the embeddings (or the feature representations) of the audio utterance.

4. Experimental Results

In this section, we describe the database used to evaluate our proposed features and DNN architecture. In the following subsections, we discuss the parameters chosen to extract the features and train the models.

4.1. ASVspoof 2017 Corpus

The ASVspoof 2017 corpus is derived from the RedDots database. As described in [8] and [10], the genuine utterances are a subset of the original RedDots database. The replayed data has been collected from a pool of volunteers with access to a diverse array of replay and playback devices, in different session conditions. As shown in Table 1, the corpus is partitioned into a training (Train), development (Dev) and evaluation (Eval) sets. Ground-truth labels of the speaker, sentence, playback device, replay device, session information and the target class (spoof or genuine) are available for both Train and Dev sets. The Eval set has been provided with labels only for the sentence spoken for each test utterance. While there is no overlap in speakers between the data partitions, there is some overlap in replay configurations between the partitions. Replay configurations are chosen among 15 playback devices, 16 recording devices and an unspecified number of environmental conditions.

Table 1: *Partitioning of the ASVspoof 2017 Corpus.*

Subset	# Speakers	# Replay Sessions	Utterances	
			Non-Replay	Replay
Train	10	6	1,508	1,508
Dev	8	10	760	950
Eval	24	163	1,298	12,922
Total	42	179	3,566	15,380

4.2. CQCC Baseline

Our baseline model is similar to the one used by the organizers of the ASVspoof 2017 challenge [8]. The front-end features have the same configuration as the best performing system on ASVspoof 2015 data [11]. The number of bins per octave (B) is set to 96 and the first 30 coefficients of the DCT are used as the feature vector along with their first and second derivatives. This choice of B results in a hop-size of 8.5 ms. The back-end is based on two 512-component GMM models which are trained using the EM algorithm, on genuine and spoof utterances, respectively. Scores are derived using the difference between the log-likelihoods of the genuine and spoof models:

$$\hat{s} = \log(P(X|\theta_g)) - \log(P(X|\theta_s)) \quad (1)$$

where P is the likelihood function, X is the sequence of feature vectors, θ_g and θ_s are parameters for the genuine and spoof models, respectively.

As shown in Table 2, the EER on Dev and Eval are 11.0% and 24.7%, respectively. This suggests that the Eval data are more challenging than the Dev data and that the baseline system lacks generalizability.

When normalizing the CQCC using zero-mean and unit-variance, the EERs drop to 13.7% and 28.5%, respectively.

4.3. HFCC with GMM Back-end

For the framing step, a window length of 30 ms is chosen with a 50% overlap resulting in a hop-size of 15 ms. The final feature vector contains the first 30 coefficients of the DCT with the first and second derivatives (delta and acceleration).

To evaluate the performance of these features in comparison to the baseline, the back-end configuration is exactly the same as that described in the previous section. A 512-component GMM is trained for each of the genuine and spoof classes and the score for test utterances is generated using the log-likelihoods of the two models.

Table 2 shows that this system outperforms the CQCC baseline on both Dev and Eval sets with EERs of 5.9% and 23.9%,

[2]The number of output units is equal to four when using "Train" data, and 13 when using "Train + Dev" data.

[3]For a hop size of 8.5 ms, N was fixed to 125.

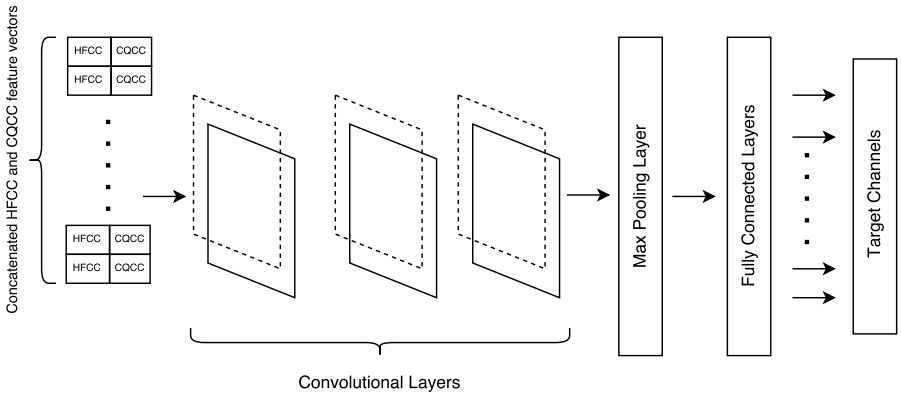

Figure 4: *Our Proposed DNN Architecture. The hidden layers consist of three convolutional layers, one max-pooling layer, and three fully connected layers.*

respecitvely. It is worth noting that the improvement is less pronounced on the Eval set. This might be attributed to over-tuning the hyper-parameters on the Dev set.

4.4. Score Fusion of HFCC and CQCC GMM-based Systems

To evaluate complementarity of CQCC and HFCC features, we initially perform a score-level fusion by a weighted sum of the scores from CQCC-GMM and HFCC-GMM systems. The weights are learned using logistic regression [17]. This system acheived the best performance on the Dev set among our proposed methods with an EER of 3.2%. This finding indicates that CQCC and HFCC features are indeed complementary. This score fusion system performs better than the single systems on the Eval set with and EER of 18.1%. However, this improvement is relatively less impressive than on the Dev set. We think that this is due to overfitting of fusion parameters.

4.5. DNN with CQCC Features

To assess the accuracy of our proposed multi-class DNN system over GMM baseline, we ran experiments with normalized CQCC as input features to the DNN. At training time, the embeddings extracted from each training utterance were then used to train a linear support vector machine (SVM) model that discriminates between genuine and spoof classes. At test time, the embeddings are used to compute the distance to the hyperplane, and generate a score for each test utterance. This system got very good results with EERs of 6.9% and 16.5% on the Dev and Eval sets, respecitvely. This shows that our proposed DNN back-end has better generalization power than GMM back-end.

4.6. DNN with Tandem HFCC and CQCC Features

Our final experiment involved a feature-level fusion of HFCC and CQCC features. To align the features in the time-domain, the parameters for the HFCC features were changed to match the hop-size for the CQCC features. The new window length for HFCC features was chosen to be 25.5 ms with a hop-size of 8.5 ms, resulting in an overlap of 66.7%. Training and scoring of the system are done in the same fashion as above using SVM. This system got an EER of 7.6% on the Dev set. Interestingly, this system achieved the best performance on the Eval set with an EER of 11.5%. This EER is 53% better than that of the CQCC-GMM baseline. These results also show the benefit of combining CQCC and HFCC features. Even though the gap between Dev and Eval results is reduced, there is an absolute difference of 3.9%. We are planning to do additional investigation when the Eval ground-truth labels are made available.

Table 2: *Comparison of our proposed systems. The best EER on each partition are highlighted.*

Sys	Description	Dev	Eval
S1	CQCC - GMM baseline	11.0%	24.7%
S2	Norm. CQCC - GMM	13.7%	28.5%
S3	HFCC - GMM	5.9%	23.9%
S4	Score fusion of S1 and S3	**3.2%**	18.1%
S5	CQCC - DNN-SVM back-end	6.9%	16.5%
S6	HFCC+CQCC - DNN-SVM back-end	7.6%	**11.5%**

5. Conclusions

We investigated a new countermeasure that benefits from the recent advances in both machine learning and signal processing. The proposed DNN front-end extractor uses CQCC features in tandem with our proposed HFCC features. Unlike existing approaches, the DNN is trained to maximize the channel variability. The proposed system outperforms the baseline CQCC-GMM system with 53% reaching a pooled EER of 11.5%. Future work may focus on a deep architecture that generalizes to different classes of spoofing attacks.

6. References

[1] S. Kucur Ergunay, E. Khoury, A. Lazaridis, and S. Marcel, "On the vulnerability of speaker verification to realistic voice spoofing," in *IEEE International Conference on Biometrics: Theory, Applications and Systems*. IEEE, Sep. 2015, pp. 1–8.

[2] 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)*, Sept 2014, pp. 1–6.

[3] J. Villalba and E. Lleida, "Preventing replay attacks on speaker verification systems," in *2011 Carnahan Conference on Security Technology*, Oct 2011, pp. 1–8.

[4] P. Korshunov, S. Marcel, H. Muckenhirn, A. R. Gonalves, A. G. S. Mello, R. P. V. Violato, F. O. Simoes, M. U. Neto, M. de Assis Angeloni, J. A. Stuchi, H. Dinkel, N. Chen, Y. Qian, D. Paul, G. Saha, and M. Sahidullah, "Overview of btas 2016 speaker

anti-spoofing competition," in *2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, Sept 2016, pp. 1–6.

[5] H. Muckenhirn, M. Magimai.-Doss, and S. Marcel, "Presentation attack detection using long-term spectral statistics for trustworthy speaker verification," in *Proceedings of International Conference of the Biometrics Special Interest Group (BIOSIG)*, Sep. 2016.

[6] Z. Wu, S. Gao, E. S. Cling, and H. Li, "A study on replay attack and anti-spoofing for text-dependent speaker verification," in *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific*, Dec 2014, pp. 1–5.

[7] A. L. chun Wang, "An industrial-strength audio search algorithm," in *Proceedings of the 4 th International Conference on Music Information Retrieval*, 2003.

[8] T. Kinnunen, M. Sahidullah, H. Delgado, M. Todisco, N. Evans, J. Yamagishi, and K. A. Lee, "The ASVspoof 2017 challenge: Assessing the limits of replay spoofing attack detection," in *INTERSPEECH*, 2017.

[9] K.-A. Lee, A. Larcher, G. Wang, P. Kenny, N. Brümmer, D. A. van Leeuwen, H. Aronowitz, M. Kockmann, C. Vaquero, B. Ma *et al.*, "The reddots data collection for speaker recognition." in *INTERSPEECH*, 2015, pp. 2996–3000.

[10] T. Kinnunen, M. Sahidullah, M. Falcone, L. Costantini, R. Gonzalez Hautamäki, D. Thomsen, A. Sarkar, Z.-H. Tan, H. Delgado, M. Todisco, N. Evans, V. Hautamäki, and K. Aik Lee, "RedDots replayed: A new replay spoofing attack corpus for text-dependent speaker verification research," in *ICASSP*, 2017.

[11] M. Todisco, H. Delgado, and N. Evans, "A new feature for automatic speaker verification anti-spoofing: Constant q cepstral coefficients," in *Odyssey 2016, The Speaker and Language Recognition Workshop*, 2016.

[12] P. Korshunov and S. Marcel, "Cross-database evaluation of audio-based spoofing detection systems," in *Interspeech*, San Francisco, CA, USA, Sep. 2016.

[13] M. Sahidullah, T. Kinnunen, and C. Hanilçi, "A comparison of features for synthetic speech detection," in *Interspeech 2015*, 2015.

[14] M. Sahidullah, H. Delgado, M. Todisco, H. Yu, T. Kinnunen, N. Evans, and Z.-H. Tan, "Integrated spoofing countermeasures and automatic speaker verification: an evaluation on asvspoof 2015," *Interspeech 2016*, pp. 1700–1704, 2016.

[15] Z. Wu, J. Yamagishi, T. Kinnunen, C. Hanilci, M. Sahidullah, A. Sizov, N. Evans, M. Todisco, and H. Delgado, "Asvspoof: the automatic speaker verification spoofing and countermeasures challenge," *IEEE Journal of Selected Topics in Signal Processing*, vol. PP, no. 99, pp. 1–1, 2017.

[16] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting." *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[17] A. Sizov, E. Khoury, T. Kinnunen, Z. Wu, and S. Marcel, "Joint speaker verification and anti-spoofing in the i-vector space," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 821–832, Apr. 2015.