A Comparison of Features for Replay Attack Detection

Zhifeng Xie\textsuperscript{a}, Weibin Zhang\textsuperscript{b}, Zhuxin Chen and Xiangmin Xu
South China University of Technology, Guangzhou, China

Email: \textsuperscript{a}xie.zhifeng@mail.scut.edu.cn, \textsuperscript{b}zhang.weibin@mail.scut.edu.cn

Abstract: Speaker verification (ASV) systems are still vulnerable to different kinds of spoofing attacks, especially replay attack due to high-quality playback devices. Many countermeasures have been developed recently. Most of the efforts focus on the search for more salient features and many new features have been proposed. Five kinds of features, namely Mel-frequency cepstral coefficients (MFCCs), linear frequency cepstral coefficients (LFCCs), inverted Mel-frequency cepstral coefficients (IMFCCs), constant Q cepstral coefficients (CQCCs) and bottleneck features were compared on the public ASVspoof 2017 and BTAS 2016 datasets in this paper. Our experimental results show that MFCCs and bottleneck features yield comparable results. Both of them significantly outperform others (including the recently proposed CQCCs). However, the number of filters and cepstral bins are essential to the success of MFCCs.

1. Introduction

Automatic speaker verification (ASV), a convenient biometric authentication technology that recognizes the speakers’ identification depended on sound recordings, has gain much attention recently. However, even the most advanced ASV systems are still vulnerable to certain spoofing attacks, such as replay attacks, voice conversion and speech synthesis. Replay attacks have the greatest threat to ASV systems due to high-quality playback devices [1].

Great efforts have been made to develop countermeasures. Except the traditional Mel-frequency cepstral coefficients (MFCCs [2]) that have been widely applied in speaker recognition and speech recognition, many new features have been proposed, including phase spectrum [3], magnitude spectrum [4], linear prediction error [5], long-term features like F0 statistics [6], inverted Mel-frequency cepstral coefficients (IMFCCs [7]), linear frequency cepstral coefficients (LFCCs [8]) and constant Q cepstral coefficients (CQCCs [9]). As for the back-end classifiers, the traditional Gaussian Mixture Model (GMM [3]) and deep neural network (DNN [10]) are the most frequently used classifiers.

Given so many features proposed, it is necessary to compare the performance of different features for spoofing detection. Some work has been done before. The authors in [11] found that high-dimensional and dynamic features are useful. Similar conclusions were also drawn in [12]. However, only dynamic features are not good enough to detect spoofing attack. It is useful for detecting spoofing attacks in real-life condition using static and dynamic coefficients of spectral features[13]. In addition, it is found that high-frequency spectral information is more important than the low-frequency region, especially for IMFCC [14]. On the ASVspoof dataset, the newly proposed CQCCs have been shown that it is 72% higher than the best results ever. [9].
However, previous comparisons have some limitations. It has been shown that IMFCCs perform better than traditional MFCCs [14]. In addition, by increasing the resolution of MFCCs, we can significantly increase the recognition accuracy of MFCCs [15, 16]. However, in experiments reported in [9], only traditional MFCCs are compared with CQCCs. In [7, 11, 14], the newly proposed CQCCs are not included in the comparisons. In terms of the classifiers, simple classifiers such as Gaussian mixture models (GMMs) are commonly used [7, 14]. More advanced models such as deep neural networks that are more powerful in processing high-dimension input are not used. Last but not least, The experimental results of ASV deception challenge in 2015 and 2017 [1,17] also show that detection of replay attacks is more difficult than SS and VC deception attacks. It is also easier to perform replay attacks. Unfortunately, there is no comparison of features for replay attack detection.

In this paper, we focus on replay attack. Five kinds of features, namely MFCCs, IMFCCs, LFCCs, CQCCs, and bottleneck features were compared on the public ASVspoof 2017 and BTAS 2016 datasets. Deep neural networks were used as the classifiers.

The rest of this paper is organized according to the following. We will describe different features used in our systems in Section 2. In Section 3, the classifiers will be presented. In Section 4, we will give our experimental settings and results. Lastly, we draw conclusions in Section 5.

2. Features
As stated in the Introduction, many different feature representations have been proposed. Here we investigate MFCCs, LFCCs, IMFCCs, CQCCs, and bottleneck features.

2.1. Short-Term Power Spectrum Features
MFCCs have been commonly used in speaker verification. Firstly, we use Fourier transform to get the power spectrum of each frame. Then we calculate the energy in each Mel-frequency filter bank of the power spectrum. The cepstral coefficients will then be given by the DCT of the log energies of filter banks.

Linear Frequency Cepstral Coefficients (LFCCs) and Inverted Mel Frequency Cepstral Coefficients (IMFCCs) are extracted mostly in the same way as MFCCs. The only difference is that they give emphasis to different area of frequencies by using different filters. The filters in MFCCs space in Mel scale, while the LFCCs space in linear scale, and the IMFCCs space in inverted-Mel scale. The differences are illustrated in Figure 1.

![Figure 1. Filter used by different features. From top to the bottom: IMFCCs, MFCCs and LFCCs](image)

However, some researchers argue that MFCCs may perform poorly due to discriminative information missing. In our preliminary experiments, we also got the same conclusion. Researchers in [15] proposed to allocate different numbers of filters to different sub-bands to overcome this problem. Our previous work [16] showed that we can simply increase the number of filters to keep more
discriminative information. In this paper we further explored the effect of cepstral bins and the number of filters. Maintaining the Integrity of the Specifications

2.2. Constant Q Cepstral Coefficients
Constant Q Cepstral Coefficients (CQCCs) achieved state-of-art performance in ASVspoof 2015 Challenge for spoof detection. The results show that compared with other methods, the cqccs feature has a higher possibility of capturing deception attacks and manipulating artificial traces. [9]. Constant Q transform (CQT) is used to obtain the spectrum instead of using short-term Fourier transform. More details about CQCCs can be found in [9].

2.3. Bottleneck feature
It is shown that bottleneck features [18] have achieved good performance in improving the accuracy of automatic speech recognition (ASR) systems. Thus we explored using bottleneck feature in replay spoof detection.

We extract bottleneck features from a well-trained DNN network. We used a bottleneck layer as the last hidden layer of the DNN. After training, we expect the bottleneck layer will contain enough discriminative information to enable distinguish between spoofed and genuine speech. We extract the bottleneck feature from the output of the last hidden layer. Figure 2 shows our extracting process.

3. Methodology
Two classifiers were used in our experiments, i.e. the Gaussian mixture model and deep neural networks. We will introduce them briefly in this Section.

3.1. Gaussian Mixture Models
Gaussian mixture models (GMMs) is a conventional classifier in speaker recognition. Speech were separated into genuine and spoofed. All samples from a class are used to train a class-depend model. During testing, we calculate the log-likelihood ratio (LLR) for each speech frame $o_i$ by using the Equation (1). The final score for an utterance is calculated by summing all LLR scores, and normalized by the number of frames.

$$LLR = \log P(o_i \mid M_{\text{genuine}}) - \log P(o_i \mid M_{\text{spoof}})$$

3.2. Deep Neural Networks
DNN have obtained great success in many fields including spoof detection [10]. Our results shows that deep neural network performed better than GMM in replay detection. Usually the features within a
context windows are spliced together as the input of deep neural networks. Backpropagation is used to optimized the model parameters. In the process of testing, it is similar to GMM except for the posterior probability given by network output.

4. Database
The publically available datasets, the ASVspoof2017 [1] and the BTAS2016 [19], were used in our experiments. Since we focus on replay spoof detection, the voice conversion and speech synthesis data were removed from the BTAS2016 dataset. The two data sets are divided into three subsets: training, development and evaluation. Table 1 gives a brief description of the data set.

| Database       | Types     | Train | Dev | Eval |
|----------------|-----------|-------|-----|------|
| ASVspoof2017   | Genuine data | 1508  | 760 | 1298 |
|                | All replay | 1508  | 950 | 12922 |
| BTAS2016       | Genuine data | 4973  | 4995| 5576 |
|                | All replay | 2800  | 2800| 4800 |
|                | RE-LP-LP   | 700   | 700 | 800  |
|                | RE-LP-HQ-LP| 700   | 700 | 800  |
|                | RE-PH1-LP  | 700   | 700 | 800  |
|                | RE-PH2-LP  | 700   | 700 | 800  |
|                | RE-PH2-PH3 | -     | -   | 800  |
|                | RE-LPPH2-PH3| -    | -  | 800  |

The cepstrum of the speech signal is obtained from the inverse transformation of the logarithm of the spectrum. For MFCCs, IMFCCs and LFCCs, the spectrum is obtained by using the short-term Fourier transform (STFT). The constant Q transform is used to get the spectrum of CQCCs instead. In addition, filters are applied before the inverse transform to get MFCCs, IMFCCs and LFCCs while uniform resampling is used for CQCCs.

For MFCCs, IMFCCs and LFCCs, the speech signal is examined every 10ms with a 25ms overlapping window. In order to provide more discriminative information for spoof attack detection, we used different numbers of filters, ranging from 23 (typical setting in speech recognition) to 126 (the maximum value if 512 points of STFT is used). Different numbers of cepstral bins (from 13 to 40) were also examined. As mentioned above, dynamic features are very useful for spoof detection. Therefore, they were appended to the static features, resulting in 39- to 120- dimension features. For CQCCs, the Matlab tool introduced in [9] was used to extract the features. CQCCs used uniform resampling instead of bandpass filters before the inverse transformations. Same as other cepstral features, we investigated CQCCs with different numbers of cepstral bins (from 13 to 40). In addition, the dynamic features are appended with the static features.

To extract bottleneck features, we used a traditional feed-forward deep neural network (DNN). We inserted a bottleneck layer before the last layer of the DNN. There are four hidden layers (including the bottleneck layer) in the DNN. Every hidden layer has 512 nodes except the bottleneck layer. For the input of the DNN, we used a context window of 11 frames. Since the context information has been included in the input, we did not used dynamic features for bottleneck features. In addition, the number of nodes for the bottleneck layer was chosen to be the same with the dimension of cepstral coefficients for fair comparison.

As stated before, we used DNNs and GMMs as the classifiers. For the GMMs, each GMM was trained with 512 Gaussian components. We trained three hidden layers of DNN, each with 512 hidden units. We used a context window of 11 feature frames as input. Besides, there is a batch normalization layer behind each hidden layer, which makes the learning process insensitive to the learning rate and initial parameters. We use cross-entropy as loss function. Dropout are used to train networks and prevent them from over-fitting. The output layer is softmax of two dimension, one for real voice hypothesis and the other for deceiving voice.
10-fold cross-validation is used in our experiments. Data is randomly divided into 10 subsamples. One subsample is picked as the testing data, and the rest of data are used for training. This process is then repeated 10 times. The advantage of this method is that all data are used for both training and validation.

5. Experimental Results

We first perform experiments to compare the classifiers of GMMs and DNNs. The results are shown in Table 2. As can be seen, the DNN models perform significantly better than the GMM models. Thus in the following experiments we will mainly focus on using DNN models.

Table 2. Equal error rates (EER%) of GMMs and DNNs on development/evaluation data sets

| feature | GMM         | DNN         |
|---------|-------------|-------------|
| MFCC    | 7.47/16.91  | 7.73/10.14  |
| CQCC    | 10.83/28.06 | 5.18/19.41  |

Table 3 and table 4 show the experimental results on the development and evaluation data sets respectively, with various settings for feature extraction. The DNN classifiers were used. We can see that, as the number of filters increased, the EERs of MFCCs and LFCCs decreased significantly. MFCCs get the best result 7.73% EER on development data set and 10.14% on evaluation data. LFCCs get the best results 6.20% and 12.91% EER respectively on development and evaluation data sets.

Table 3. Equal error rates (EER%) on the development data set with various settings for feature extraction.

| feature | filter numbers | features dimension |
|---------|----------------|--------------------|
|         |                | 39 | 60 | 90 | 120 |
| MFCC    | 23             | 21.14 | 18.96 | - | - |
|         | 40             | 20.33 | 15.98 | - | - |
|         | 60             | 17.11 | 12.23 | 10.60 | 11.88 |
|         | 80             | 15.65 | 11.43 | 9.94 | 9.46 |
|         | 100            | 14.52 | 12.13 | 7.59 | 8.02 |
|         | 126            | 10.23 | 8.56 | 7.61 | 7.73 |
| IMFCC   | 23             | 8.21 | 8.84 | - | - |
|         | 40             | 8.69 | 8.84 | 8.73 | 8.95 |
|         | 60             | 8.68 | 8.78 | 8.45 | 8.60 |
|         | 80             | 8.65 | 8.83 | 8.44 | 8.74 |
|         | 100            | 8.35/ | 8.78 | 8.42 | 8.81 |
| LFCC    | 23             | 10.61 | 9.95 | - | - |
|         | 40             | 9.54 | 9.59 | 9.34 | 8.31 |
|         | 60             | 7.36/ | 8.46 | 7.45 | 7.30 |
|         | 80             | 8.38 | 7.34 | 6.54 | 6.56 |
|         | 100            | 8.13 | 7.55 | 6.29 | 6.20 |
| CQCC    | -              | 5.29 | 5.36 | 4.92 | 5.13 |
| bottleneck | -          | 7.36 | 6.43 | 6.94 | 7.03 |
Table 4. Equal error rates (EER%) on the development data set with various settings for feature extraction. features dimension

| feature   | filter number s | features dimension |
|-----------|-----------------|--------------------|
|           | 39              | 60                 | 90                 | 120                |
| MFCC      |                 |                    |                    |                    |
| 23        | 28.06           | 27.97              | -                  | -                  |
| 40        | 25.49           | 23.15              | -                  | -                  |
| 60        | 23.02           | 18.57              | 15.25              | 13.84              |
| 80        | 20.46           | 15.27              | 13.97              | 14.07              |
| 100       | 17.56           | 14.63              | 12.87              | 11.56              |
| 126       | 14.56           | 14.63              | 12.33              | 10.14              |
| IMFCC     |                 |                    |                    |                    |
| 23        | 16.83           | 16.11              | -                  | -                  |
| 40        | 15.58           | 15.01              | 15.45              | 15.19              |
| 60        | 15.51           | 15.37              | 16.12              | 16.04              |
| 80        | 15.43           | 15.27              | 15.97              | 15.07              |
| 100       | 15.81           | 15.18              | 14.79              | 15.07              |
| LFCC      |                 |                    |                    |                    |
| 23        | 18.48           | 17.92              | -                  | -                  |
| 40        | 15.91           | 14.8               | 15.88              | 15.45              |
| 60        | 13.99           | 13.76              | 14.18              | 13.84              |
| 80        | 15.35           | 13.06              | 13.14              | 12.22              |
| 100       | 14.04           | 14.08              | 13.62              | 12.91              |
| CQCC      | -               | 22.63              | 20.34              | 18.41              | 19.42              |
| bottleneck| -               | 11.45              | 10.41              | 10.57              | 11.27              |

As for the CQCCs and bottleneck features, different dimensions of features were also used to find the best results. Results shows that 90-dimension CQCCs features and 120- dimension bottleneck features perform the best.

As can be seen from Table 3 and 4, if the typical settings (i.e., 23 filters and 13 cepstral bins) was used for cepstral extraction, MFCCs perform the worst. This coincide with previous findings [16]. However, by increasing the number of filters and also the number of cepstral bins, the performance of MFCCs can be significantly improved. The best MFCC significantly outperform the newly proposed CQCC feature. The best result is achieved by using the MFCCs. The best results of bottleneck features and MFCCs are comparable.

Table 5 gives the results of 10-fold experiments in ASVspoof 2017. MFCCs and Bottleneck features also perform the best in these features.

Table 5. Equal error rates (EER%) on ASVspoof 2017 using ten-fold.

|        | MFCC | IMFCC | LFCC | CQCC | Bottleneck |
|--------|------|-------|------|------|------------|
| dev    | 2.38%| 2.85% | 2.73%| 2.89%| 2.41%      |
| eval   | 2.52%| 3.23% | 3.31%| 3.41%| 2.75%      |

Table 6 gives the results on BTAS 2016 replay detection. Results show that MFCCs have the best results of 0.22% in development set and 1.24% in evaluation set. IMFCCs also have the same performance with MFCC in development set. However, IMFFCs perform bad in evaluation set. This means IMFCCs can not performed well in unknown attacks detection. Table 7 gives more analysis of the MFCCs results. Different replay devices are separately calculated the EER on evaluation set. Obviously the HQ(high quality) devices is more difficult in this task.

Table 6. Equal error rates (EER%) on dev and eval set of BTAS 2016

|        | MFCC | IMFCC | LFCC | CQCC | Bottleneck |
|--------|------|-------|------|------|------------|
| dev    | 0.22%| 0.22% | 0.06%| 1.22%| 0.43%      |
| eval   | 1.24%| 11.05%| 2.04%| 8.32%| 1.75%      |
Table 7. Equal error rates (EER%) on different replay devices using MFCC features

| Replay devices | Eval (EER %) |
|----------------|-------------|
| All replay     | 1.24        |
| RE-LP-LP       | 1.506       |
| RE-LP-HQ-LP    | 1.758       |
| RE-PH1-LP      | 0.3587      |
| RE-PH2-LP      | 0.6277      |
| RE-PH2-PH3     | 0.7353      |

6. Conclusion

This paper presents a comparison of features for replay attack detection. By using the released ASVspoof 2017 and BTAS 2016 data set that is designed for replay attack detection, we compared five features: MFCCs, LFCCs, IMFCCs, CQCCs and bottleneck features. Our results show that the performance of spoof detection systems can be significantly improved by increasing the number of filters and cepstral bins. MFCCs and bottleneck features yield comparable results. Both of them significantly outperform the other features, including the newly proposed CQCC feature.

References

[1] T. KinmuneN, M. Sahidullah, H. Delgado, M. Todisco, N. Evans., J. Yamagishi, K. A. Lee, “The ASVspoof 2017 Challenge: Assessing the Limits of Replay Spoofing Attack Detection”, Interspeech 2017, in press.

[2] S. Davis and P. MerMelstein, “Comparison of parametric repre- sentations for monosyllabic word recognition in continuously spo- ken sentences”, in IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 28, no. 4, pp. 357366, Aug 1980.

[3] L. Wang, Y. Yoshida, Y. Kawakami and S. Nakagawa, “Relative phase information for detecting human speech and spoofed speech”, in INTERSPEECH, pp. 2092-209, September 2015.

[4] M. J. Alam, P. Kenny, G. Bhattacharya, and T. Stafylakis, “Development of CRIM system for the automatic speaker verification spoofing and countermeasures challenge”, in INTERSPEECH, pp. 2072-2076, 2015.

[5] A. Janicki, “Spoofing countermeasure based on analysis of linear prediction error”, in INTERSPEECH, pp. 2077-2081, September 2015.

[6] P. L. De Leon, B. Stewart, and J. Yamagishi, “Synthetic speech discrimination using pitch pattern statistics derived from image analysis”, in INTERSPEECH, pp. 370-373, 2012.

[7] P. Korshunov, and S. Marcel, “Cross-database evaluation of audio-based spoofing detection systems”, in INTERSPEECH, pp. 1705-1709, 2016.

[8] S. Furui, “Cepstral analysis technique for automatic speaker verification”, In IEEE Transactions on Acoustics, Speech and Signal Processing (ICASSP), vol. 29, no. 2, pp. 254272, Apr. 1981.

[9] M. Todisco, H. Delgado, and N. Evans, “A new feature for automatic speaker verification antispoofer: Constant Q cepstral coefficients,” in Speaker Odyssey Workshop, Vol. 25, pp. 249252, 2016.

[10] J. Villalba, A. Miguel, A. Ortega, and E. Lleida, “Spoofing detection with dnn and one-class svm for the asvspoof 2015 challenge”, in INTERSPEECH, pp. 2067-2071, 2015.

[11] X. Tian, Z. Wu, X. Xiao, E. S. Chng, and H. Li, “ Spoofing detection from a feature representation perspective”, In IEEE Transactions on Acoustics, Speech and Signal Processing (ICASSP), pp. 2119-2123, March 2016.

[12] D. Paul, M. Pal, and G. Saha, “Novel speech features for improved detection of spoofing attacks”, In IEEE 2015 Annual IEEE India Conference (INDICON), pp. 1-6. December, 2016

[13] D. Paul, M. Sahidullah, and G. Saha, “Generalization of spoofing countermeasures: A case study with ASVspoof 2015 and BTAS 2016 corpora”, In IEEE Transactions on Acoustics, Speech and Signal Processing (ICASSP), 2017.
[14] M. Sahidullah, T. Kinnunen, and C. Hanili, “A comparison of features for synthetic speech detection”. In Sixteenth Annual Conference of the International Speech Communication Association, 2015.

[15] K. Sriskandaraja, V. Sethu, P. N. Le, and E. Ambikairajah, “Investigation of sub-band discriminative information between spoofed and genuine speech”, in INTERSPEECH, pp. 1710-1714, 2016.

[16] Z. Chen, Z. Xie, W. Zhang, X. Xu, “ResNet and Model Fusion for Automatic Spoofing Detection”, INTERSPEECH 2017, in press.

[17] Z. Wu, J. Yamagishi, T. Kinnunen, C. Hanilci, M. Sahidulla, A. Sizov et al, “Asvspoof: the automatic speaker verification spoofing and countermeasures challenge”, in IEEE Journal of Selected Topics in Signal Processing, 2017.

[18] Y. Du, and M. L. Seltze. ”Improved Bottleneck Features Using Pretrained Deep Neural Networks”. in INTERSPEECH, pp. 237-240, August 2011.

[19] Ergünay, Serife Kucur and Khoury, Elie and Lazaridis, “On the vulnerability of speaker verification to realistic voice spoofing”, in IEEE International Conference on Biometrics Theory, Applications and Systems, 2015.