A Robust Method for Speech Replay Attack Detection

Lang Lin, Rangding Wang*, Diqun Yan and Li Dong
Faculty of Electrical Engineering and Computer Science, Ningbo University
Ningbo 315211, China
[e-mail: ll_linlang@163.com]
*Corresponding author: Rangding Wang
[e-mail: wangrangding@nbu.edu.cn]

Received January 14, 2019; revised Septebmer 20, 2019; accepted October 23, 2019; published January 31, 2020

Abstract

Spoofing attacks, especially replay attacks, pose great security challenges to automatic speaker verification (ASV) systems. Current works on replay attacks detection primarily focused on either developing new features or improving classifier performance, ignoring the effects of feature variability, e.g., the channel variability. In this paper, we first establish a mathematical model for replay speech and introduce a method for eliminating the negative interference of the channel. Then a novel feature is proposed to detect the replay attacks. To further boost the detection performance, four post-processing methods using normalization techniques are investigated. We evaluate our proposed method on the ASVspoof 2017 dataset. The experimental results show that our approach outperforms the competing methods in terms of detection accuracy. More interestingly, we find that the proposed normalization strategy could also improve the performance of the existing algorithms.

Keywords: Automatic speaker verification, replay attacks, channel effect, robustness, post-processing

Foundation Items：This work was supported by the National Natural Science Foundation of China (Grant No. U1736215, 61672302), Zhejiang Natural Science Foundation (Grant No. LZ15F020002, LY17F020010), Ningbo Natural Science Foundation (Grant No. 2017A610123), Mobile Network Application Technology Key Laboratory of Zhejiang Province (Grant No. F2018001), Ningbo University Fund (Grant No. XKXL1509, XKXL1503) and K.C. Wong Magna Fund in Ningbo University.

http://doi.org/10.3837/tiis.2020.01.010 **ISSN : 1976-7277**

1. Introduction

Automatic speaker verification system (ASV) has been widely used in finance and life applications due to its convenience, high security, and remote operability [1]. While the ASV technology is constantly evolving, various spoofing attacks on ASV systems emerge [2]. Spoofing attacks can be generally classified into four categories: voice conversion impersonation, replay speech and synthesis speech. We focus on replay attacks, which is regarded as the most flexible, easiest spoofing attacks. The main reason is that with the widespread use of high-fidelity recording and playback device, it is easy to record the voice of the target speaker. No signal processing expertise is needed, making replay spoofing attacks easy to implement.

In the last decades, Shang *et al.* [3] and Jakub *et al.* [4] proposed a replay attacks detection algorithm by comparing a test recording with the recordings that exist in the database. Wang *et al.* [5] developed a method by using channel information to detect replay attacks. However, these works used a database collected with a small set of recording and playback devices. Recently, the ASVspoof 2017 challenge [6] put its focus on replay attacks, which has received extensive attention from researchers. Constant Q cepstral coefficients (CQCC), which is proposed by Todisco M *et al.* [7], was adopted in the baseline system for this challenge. After that, various features were used in recent literature to improve the performance of replay detection, such as the inverted Mel-frequency cepstral coefficients (IMFCC) [8], single frequency filtering coefficients (SFFCC) [9], high-frequency cepstral coefficients (HFCC) [10], and linear frequency cepstral coefficients (LFCC) [11]. All these works used CQCC features as baseline features and a Gaussian Mixture Model (GMM) classifier for the final classification.

Current research on replay detection has concentrated on either developing new features or improving the classifier, implicitly ignoring the variability of features. The variability of features includes acoustic variability, channel variability, speaker variability, etc. The most influential on replay attacks detection is the channel variability, which is most likely to change in the practical terms. The main reason is that the recording device and playback device used by the attacker are usually unknown to the detector. If we cannot eliminate the channel effects brought by the device, the robustness of the feature will be significantly reduced. However, few works noticed that eliminating channel effects could improve detection performance.

In our previous research on ASVspoof 2017 dataset, we performed a detailed analysis of the differences between genuine speech and the replay speech on the frequency sub-bands. Our research showed that the discriminative information of genuine speech and replay speech is mainly distributed in two sub-bands, i.e., 0-1 kHz and 7-8 kHz [12][13]. one plausible explanation for this observation can be as follows. In the replay speech, the reverberation information and channel noise caused by the recording and playback devices usually have low-frequency components, whereas the environment noises are often in the high-frequency bands [14]. Based on this observation, in this paper, we first establish a mathematical model for genuine speech and replay speech. Then we propose a method by using a band-stop filter on speech signals to emphasize discriminant sub-bands. Finally, the cepstrum feature named band-stop filter cepstral coefficient (SFCC) is then extracted based on the residual signal. SFCC can not only effectively extract the spectral information of the high-frequency region, but also can describe the low-frequency spectrum information in detail. To improve the detection performance of our method, four normalization techniques for eliminating channel effects are also adopted to our features. Experimental results show that our algorithm can effectively detect replay speech. It also proves that eliminating channel effects can actually improve the detection performance of the existing algorithms.

The contribution of this paper can be summarized as follows:

● By analyzing the influence of channel variability on features, a simple but effective solution to discriminate genuine speech and replay speech is proposed.

● We propose a novel feature, SFCC, that can capture low-frequency information and high-frequency information of the spectrum. Experimental results show that our algorithm can effectively detect replay speech.

●We establish a mathematical model for genuine speech and replay speech, and proposed a method that uses normalization techniques to eliminate the effects of the channel. This approach significantly boosts the robustness of the feature, which could benefit the existing algorithm as well.

The rest of this paper is organized as follows. Section 2 briefly reviews the related works. Our method is introduced in Section 3. The experimental results are presented in Section 4. Section 5 concludes this work.

2. Related Works

2.1 Replay Attacks

Replay attacks are considered one of the easiest and most effective way of spoofing attacks. The main reason is that it does not require any special signal processing knowledge for an attacker, only a recording device can be implemented. Replay attacks are exemplified by a scenario in **Fig. 1**. It can be seen that the genuine speech is directly from the target speaker's voice, and the replay speech is obtained by the attacker recording the target speaker's voice and then playing it back. By denoting the speech signal by s, and replay channel response is h, the replay speech signal r is can be represented by the following linear convolution process:

$$r = s \otimes h. \tag{1}$$

It can be seen from (1) that the difference between genuine and replay speech is mainly caused by different channels (i.e., h). Therefore, how to effectively extract the essential discriminative information between replay speech and genuine speech due to different channels is the key to detecting replay attacks. Currently, researchers have proposed some effective replay attacks detection algorithms. In the next, we briefly review some related papers.

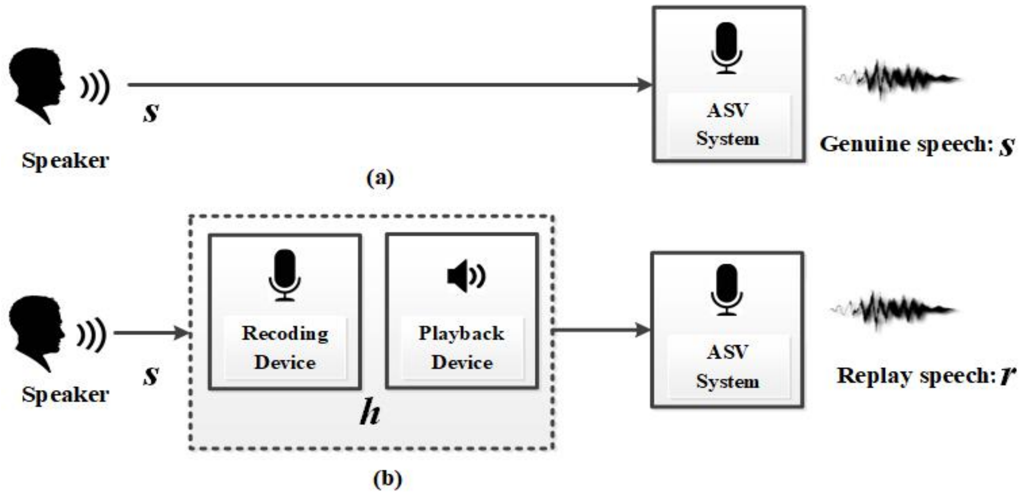


Fig. 1. An illustration of the replay attacks. (a) is the genuine speech generation process, genuine speech is a recording of a bona fide access to the microphone of the ASV system. (b) is the replay speech generation process. Replay speech is a spoofing recording in which an attacker records the target speaker's voice and later plays it back to an ASV system

2.2 Methods for Detecting Replay Attacks

Features based on the short-term spectrum are widely used in spoofing detection. In this paper, we focus on several cepstral features that perform well in replay attacks detection, namely, CQCC, LFCC, IMFCC and HFCC.

The recently proposed CQCC feature has proven to be effective in the replay attacks detection [7]. CQCC is a spectro-temporal resolution variable feature that is derived from a constant Q transform (CQT). Although the CQCC feature provides more spectral detail information in the low-frequency region, the discriminative information in the high-frequency region is totally neglected.

LFCC feature is a cepstrum-based feature. Unlike CQCC, LFCC employs a discrete Fourier transform(DFT). It is an efficient tool for time-frequency analysis, which imposes regularly spaced frequency bins. The LFCC feature extracts the information of the entire frequency band. In fact, the spoofing information is mainly distributed in the low- frequency and high-frequency sub-bands. Therefore the LFCC feature is not able to provide more spectral detail in the discriminative frequency bands.

IMFCC is another cepstrum-based feature. The processing steps of the IMFCC feature are similar to LFCC feature extraction chain with the exception of the filters. In IMFCC feature, filters have denser spacing in the high-frequency region. Therefore, IMFCC feature can capture more spectrum information in the high-frequency region, but inevitably ignores low-frequency details.

Similar to I-MFCC, HFCC feature also concentrates on high-frequency information. In the pre-processing step of the HFCC feature, the speech signal is filtered using a high-pass filter. Only the high-frequency information is retained for the extraction of cepstral features. However, this feature also ignores the difference between genuine speech and replay speech in the low-frequency region.

In the following, we will first establish a mathematical model of the replayed speech and then analyze how to remove the effects of the channel. Finally, we will present our method for detecting replay attacks.

3. Proposed method

3.1 Mathematical Model of Replay Speech

In this subsection, we first establish a mathematical model of the replay speech and then analyze how to remove the influence of the channel. We know that the replay speech signal r is a linear convolution of a genuine speech signal s and impulse response of the channel h. To analyze the impact of the channel, the convolutive relationship between s and h can be transformed into a multiplicative relationship in the frequency domain by taking DFT given by：

$$\begin{aligned} F(r) &= F(s \otimes h) \\ &= F(s) \times F(h) \end{aligned}, \tag{2}$$

where $F(.)$ is the Fourier transform function, $F(r)$, $F(s)$ is the spectrum vectors of replay speech and genuine speech, respectively, and $F(h)$ is impulse response of channel in the frequency domain. Further, we transform the multiplicative relationship between $F(s)$ and $F(h)$ into the additive relationship of cepstrum by taking logarithm, which is：

$$\begin{aligned} R &= \log[\mathrm{F}(r)] \\ &= \log[F(s) \times F(h)] \\ &= \log F(s) + \log F(h) \\ &= S + H \end{aligned}, \tag{3}$$

where R, S, H are the cepstral vectors of replay speech, genuine speech and impulse response of the channel, respectively. In an utterance, the channel change is extremely weak, so we can reasonably assume that channel response H does not change [15].

Although the information of the replay channel is helpful for replay attacks detection. We prefer to use the influences other than those caused by channel differences to discriminate between genuine speech and replay speech. The main reason is that in the practical scenario, one does not know the recording and playback devices used by the attacker. As a result, it is difficult to establish an accurate model of the channel used by the attacker. Therefore, how to effectively remove the channel effect is crucial for detecting replay attacks.

3.2 Remove the Channel Effects

It is well known that in the cepstral domain any convolutional distortions are represented by addition. As we have mentioned before, we intend to use the influences other than those caused by channel differences to discriminate between genuine speech and replay speech. Therefore, we in this section discuss how to remove the influence of the channel.

In the preprocessing stage, the speech signal is split up into overlapping frames. We can observe that for every n-th frame cepstral coefficient is:

$$R_n = S_n + H \ , \tag{4}$$

where R_n, S_n are the cepstral coefficients of each frame of replay speech and genuine speech, respectively. It should be emphasized that in the analysis, we assume that channel response is not changing in an utterance. By taking the average over all frames, we get:

$$\mu_{R_n} = \frac{1}{N}\sum_n R_n = \frac{1}{N}\sum_n S_n + H \ , \tag{5}$$

where μ_{R_n} is the mean of the cepstral coefficient. By further subtracting the average from each coefficient we can obtain:

$$\begin{aligned} C_n &= R_n - \mu_{R_n} \\ &= (S_n + H) - (\frac{1}{N}\sum_n S_n + H), \\ &= S_n - \frac{1}{N}\sum_n S_n \end{aligned} \tag{6}$$

where C_n is the final signal with channel influences removed.

In the next, we will introduce the features proposed in this paper and the four post-processing methods used to eliminate channel effects

3.3 Feature Extraction

In our preliminary studies [12][13], we found that the spoofing information is mainly distributed in the low-frequency region (0-1kHz) and the high-frequency region (7-8 kHz) of the spectrum. Therefore, the SFCC feature proposed in this paper is based on these two discriminative sub-bands. SFCC is a cepstrum feature that captures low frequency and high-frequency information of the spectrum through band-stop filtering. **Fig. 2**. shows the extraction process of SFCC features.

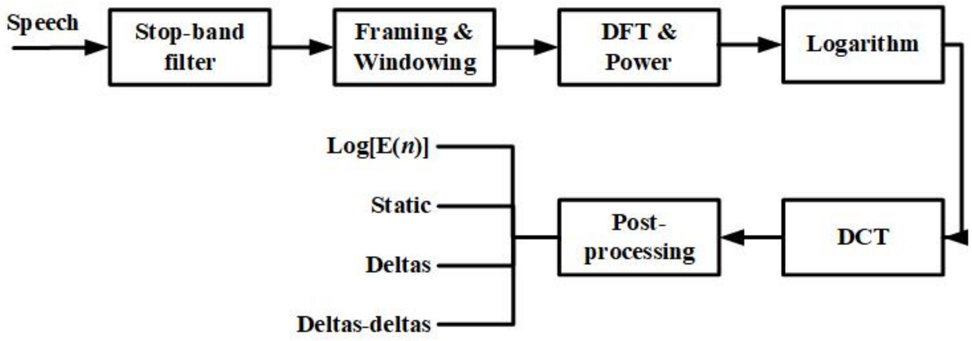


Fig. 2. SFCC Feature Extraction Process. $\log E(n)$ is the log-energy coefficient. Static represents the coefficients of the DCT. Delta and acceleration represent the delta coefficients and delta-delta coefficients of the static coefficients.

First, the speech signal is filtered using a Chebyshev band-stop filter. Then, the residual signal is split up into overlapping frames. After that, the power spectrum of each frame is derived from a DFT, given by

$$SF(k,n) = |X^{DFT}(k,n)|^2, \tag{7}$$

where $k = 1, 2..., K$ represents the frequency bin index, n represents the frame index, and $X^{DFT}(k,n)$ is the spectral coefficient derived by DFT, and $SF(k,n)$ is the power spectral coefficient. Finally, the power spectrum coefficient is logarithmically computed and converted to cepstral coefficients by adopting the discrete cosine transform (DCT), which is:

$$C(p,n) = \sum_k^K \log[SF(k,n)]\cos\left(\frac{p(k-\frac{1}{2})p}{K}\right), \tag{8}$$

where p is the dimension of features, $C(p,n)$ is the cepstral coefficient. In addition, we also added a log-energy coefficient in the SFCC feature vector. The log-energy coefficient $\log\left[E(n)\right]$ can be calculated according to:

$$\log[E(n)] = \log\left[\sum_{k=1}^{K} |\mathrm{X}^{DFT}(k,n)|^2\right] - \log[\mathrm{K}], \tag{9}$$

The final feature vector consists of 121 dimensions, including 40-dimensional static coefficients, 40-dimensional delta coefficients, 40-dimensional acceleration (delta-delta) coefficients, and 1-dimensional log-energy coefficient.

3.4 Normalization as Post-processing

In this subsection, we will perform a post-processing method on the extracted cepstral features to eliminate channel effects. The post-processing of features mainly includes four normalization methods, namely, cepstral mean subtraction (CMS), cepstral mean and variance normalization (CMVN), cepstral gain normalization (CGN) and quantile-based cepstral dynamics normalization (QCN). In the following, we will briefly describe each of these methods.

CMS [16] is efficient normalization technique for ASV system. It normalizes the cepstrum feature by subtracting the mean of the cepstrum, which can be expressed as

$$C_{p,n}^{CMS} = C_{p,n} - \overline{\mu}_{C_{p,n}} \ , \tag{10}$$

where $C_{p,n}$ is the cepstral vector, $C_{p,n}^{CMS}$ is the cepstral vector performed to CMS, $\overline{u}_{C_{p,n}}$ is the mean of each cepstral vector, p represents the dimension of cepstral and n is the frame index.

CMVN not only subtracts the mean for each dimension cepstrum but also normalizes the variance of the cepstrum features [17]. The process of CMVN can be described as

$$C_{p,n}^{CMVN} = \frac{C_{p,n} - \overline{\mu}_{C_{p,n}}}{\hat{\sigma}_{C_{p,n}}} \ , \tag{11}$$

where $C_{p,n}^{CMVN}$ is the cepstral vector performed to CMVN, $\hat{\sigma}_{C_{p,n}}$ represents the variance of each cepstral vector.

CMN and CMVN assume that the distributions of cepstral coefficients are Gaussian. However, the distribution of cepstrums in practice is not the case. Therefore, a new normalization method CGN [18] was proposed to solve this issue. The CGN can be expressed as

$$C_{p,n}^{CGN} = \frac{C_{p,n} - \overline{\mu}_{C_{p,n}}}{C_{p,max} - C_{p,min}} \ , \tag{12}$$

where $C_{p,n}^{CGN}$ is the cepstral vector performed to CGN, $C_{p,max}$ and $C_{p,min}$ are the maximum and minimum values of each dimension.

QCN [19], which is proposed by H. Boril *et al.*, mainly used to reduce the mismatch between training and test sample distribution. The QCN determines the dynamic range of the cepstral feature by cepstrum histogram quantile [20]. QCN first subtracts the quantile mean from all samples and then normalizes it according to the dynamic range of the quantile. The process of QCN can be described as

$$C_{p,n}^{QCN_j} = \frac{C_{p,n} - (q_j^{C_{p,n}} + q_{100-j}^{C_{p,n}})/2}{q_{100-j}^{C_{p,n}} - q_j^{C_{p,n}}}, \tag{13}$$

where $C_{n,i}^{QCN_j}$ is the cepstral vector performed to QCN, j is in percent, $q_j^{C_n}$ and $q_{100-j}^{C_n}$ are low and high quantiles of cepstral distributions for each cepstral dimension.

4. Experimental Results

4.1 Experimental Setup

4.1.1 Dataset

The detection performance of the replay attacks method is evaluated on ASVspoof 2017 Challenge dataset [21] [22]. This database contains three non-overlapping subsets: train (Tra) set, development (Dev) set and evaluation (Eval) set. In the ASVspoof 2017 Challenge, the train subset (i.e., Tra) and the development subset (i.e., Dev) were provided in the early stage. The Dev subset is a small dataset that is used primarily for the team to debug algorithm parameters, while the Evaluation (i.e., Eval) subset is released later by the organizer to evaluate the team's algorithms. The details of the dataset are shown in **Table 1**. In this paper, we use the Tra set to train the classifier, and the Dev set and Eval set is used for testing.

Table 1. Experimental setup for data from the ASV spoof 2017.

Dataset	**# Speaker**	**# Replay session**	**# Replay Configuration**	**# Replay speech**	**# Genuine speech**
Train (Tra)	10	6	3	1508	1508
Development (Dev)	8	10	10	760	950
Evaluation (Eval)	24	161	57	1298	12008
Total	42	177	61	3565	14465

4.1.2 Feature Parameters

Features based on the short-term spectrum are widely used in spoofing detection. In this work, we focus on several cepstral features that perform well in replay attacks detection [11], namely, IMFCC, LFCC, HFCC, and CQCC. The features and their parameters used in this study are the same as given reference. A summary of the features and their parameters used in this study is shown in **Table 2**. Δ and Δ^2 is the delta and acceleration (delta-delta) of the static coefficients. "√" represents the use of parameters, and "-" represents not use.

Table 2. Experimental setup for features and their parameters

Feature	Frame length /shift	Window function	DFT or CQT bins	coefficients			Post-processing
				Static	Log-energy	deltas	
CQCC	1728/864	Hanning	863	$c_0 - c_{19}$	√	$\Delta + \Delta^2$	CMVN
IMFCC	512/256	Hamming	512	$c_0 - c_{13}$	–	Δ	–
HFCC	512/256	Hamming	512	$c_0 - c_{29}$	–	$\Delta + \Delta^2$	–
LFCC	512/256	Hamming	512	$c_0 - c_{69}$	–	$\Delta + \Delta^2$	–
SFCC	256/128	Hanning	256	$c_0 - c_{29}$	√	$\Delta + \Delta^2$	QCN

4.1.3 Classifier and Metrics

The classifier used in this paper is a GMM model of 256 components. First, different models for genuine speech (λ_g) and replay speech (λ_s) are learned using an expectation maximization (EM) algorithm with random initialization. Then classifier scores of test-utterance are calculated by log-likelihood ratio, which is

$$LLR(X) = \log[L(X \mid \lambda_g)] - \log[L(X \mid \lambda_s)], \tag{14}$$

where $LLR(X)$ represents the log-likelihood ratio of X , X is a sequence of feature vectors, L denotes the likelihood function, and λ_g , λ_s represent the GMMs for genuine speech and replay speech, respectively. Replay detection accuracy is measured by computing equal error rate (EER) [7]. Denoting the false alarm rates and miss rates at the threshold θ by $P_{false}(\theta)$ and $P_{miss}(\theta)$:

$$P_{false}(\theta) = \frac{\#\{\text{spoof trials with score} > \theta\}}{\#\{\text{Total spoof trials}\}} \quad , \tag{15}$$

$$P_{miss}(\theta) = \frac{\#\{\text{genuine trials with score} < \theta\}}{\#\{\text{Total genuine trials}\}} \quad . \tag{16}$$

The false alarm rates and miss rates depend on the threshold θ . When the two rates are equal at the threshold θ_{EER} , the value is called EER, i.e., $EER = P_{false}(\theta_{EER}) = P_{miss}(\theta_{EER})$.

4.2 Experimental Results

We conduct the experiments separately on the Dev set and Eval set of ASVspoof 2017 dataset. The first experiment shows differences in performance for various features. The second experiment assesses improvements to the performance delivered by various post-processing methods. In the third experiment, we evaluate the effect of the size of the training set on the detection performance.

4.2.1 Comparison of Features

In our first experiment, we compare the performance of various features. A summary of the features and their parameters used in this paper is shown in **Table 2**. The experimental results are shown in **Table 3**.

It can be observed that on the Dev set, the IMFCC without any post-processing method achieves the best performance. However, on the Eval set, the detection performance of this feature is very unsatisfactory. Overfitting may be a reasonable explanation for this problem.

One knows that Dev set only includes a small number of samples, while the Eval data set contains more diverse samples, so the detection performance on the Eval set is more able to measure the detection ability of the algorithm. As can be seen from **Table 3**, our method using QCN normalization shows superior performance than the other competing methods. Our method has an EER of 10.11%, which is a 27% improvement over the baseline system. Further, CQCC and LFCC also show good performance.

Table 3. Experimental Result of different features. (The best results on Eval set are highlighted in boldface, "-" represents no use of this technique).

Features	**Post-processing**	**Training on Train Testing on Dev (EER %)**	**Training on Train Testing on Eval (EER %)**
CQCC [7]	CMVN	9.24	13.92
IMFCC [8]	-	5.63	34.87
HFCC [10]	No	9.59	28.09
LFCC [11]	CMN	8.64	18.82
SFCC (Ours)	QCN	8.38	**10.11**

4.2.2 Effect of Feature Normalization

In our second experiment, we investigate the effect of feature normalization on detection performance. Normalization is one of the most effective methods in post-processing, mainly to eliminate channel effects. Specifically, we study four types of normalization: CMS, CMVN, CGN, and QCN. The performance of the normalization technology for replay spoof detection is summarized in **Table 4**.

This Experiment demonstrates that our method achieves optimal performance which is highlighted in boldface in **Table 4**. It is obvious that there is a significant improvement in replay detection performance when using normalization techniques. The two normalization methods with stable performance are CMVN or QCN, respectively. One possible explanation is that both normalization techniques use the mean subtraction and variance normalization, which are effective methods to compensate for channel variability.

In addition, for the same cepstrum feature, different normalization methods have different effects on detection performance. For CQCC, MFCC, and LFCC, the CMVN normalization method greatly improves the performance of the algorithm. Compared with HFCC and our features, QCN shows better performance. Therefore, when we detect replay attacks, how to extract effective features is important, and proper post-processing of features also benefit existing algorithms. Finally, as previously mentioned, the I-MFCC without any post-processing method may be overfitting. Our experiment shows that overfitting could be avoided by using normalization.

Table 4. Experimental results of different post-processing (The best results on Eval set are highlighted in boldface, "-" represents no use of this technique).

Feature	**Feature Normalization**	**Training on Tra set Testing on Dev set (EER %)**	**Training on Tra set Testing on Eval set (EER %)**
CQCC [7] (Baseline)	--	10.81	34.18
	CMN	9.62	15.22
	CMVN	9.24	13.92
	CGN	9.05	33.48
	QCN	10.59	15.48
IMFCC [8]	--	5.63	34.87
	CMN	14.72	24.38
	CMVN	12.23	21.25
	CGN	22.68	31.10
	QCN	13.61	22.92
HFCC [10]	--	9.61	28.40
	CMN	12.90	21.84
	CMVN	16.69	21.67
	CGN	20.87	24.16
	QCN	16.80	20.72
LFCC [11]	--	10.73	27.01
	CMN	8.64	18.82
	CMVN	13.00	17.17
	CGN	11.75	25.60
	QCN	15.17	20.58
SFCC (Ours)	--	9.18	40.15
	CMN	7.37	13.88
	CMVN	9.36	10.33
	CGN	11.63	24.02
	QCN	**8.38**	**10.11**

4.2.3 Comparison of Modified Features

In Section 4.2.2, for a fair comparison, the parameters for different features were set as recommended by the corresponding references. These parameter settings, obtained by the original author with extensive experiments, can be regarded as empirically optimal. **Table 2** shows that even compared with the results with those empirically optimal parameter settings, our algorithm still achieves the best performance.

To further evaluate the effect of feature post-processing methods on the detection performance, we conducted an additional experiment (i.e., the third experiment). In this third experiment, all the features were tested under the same conditions (when applicable): the frame length and shift, window function, DFT or CQT bins and dimensionality are all the same. As shown below, feature post-processing techniques use QCN and CMVN with better performance in the second experiment. A summary of the modified features and their parameters is shown in **Table 5**. Results for different modified features are summarized in **Table 6**.

Table 5. Features and their parameters

Feature	Frame length /shift	Window function	DFT or CQT bins	Coefficients		
				Static	Log-energy	deltas
CQCC	1728/864	Hanning	863	$c_0 - c_{29}$	√	$\Delta + \Delta^2$
IMFCC	256/128	Hanning	256	$c_0 - c_{29}$	√	$\Delta + \Delta^2$
HFCC	256/128	Hanning	256	$c_0 - c_{29}$	√	$\Delta + \Delta^2$
LFCC	256/128	Hanning	256	$c_0 - c_{29}$	√	$\Delta + \Delta^2$
SFCC	256/128	Hanning	256	$c_0 - c_{29}$	√	$\Delta + \Delta^2$

Table 6. Experimental results of modified features (‘Original ’ represents that the feature uses the parameters set by the reference, and the results are highlighted in bold).

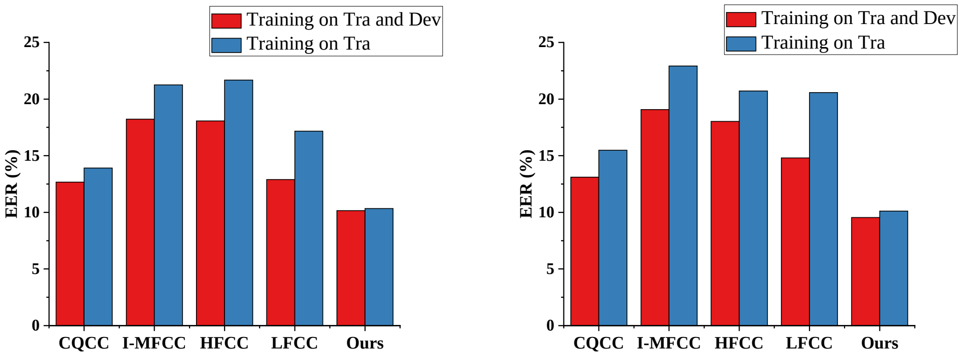
Modified Feature	Post-processing	Training on Tra set Testing on Dev set (EER %)	Training on Tra set Testing on Eval set (EER %)
CQCC	Original	9.24	13.92
	CMVN	9.08	13.88
	QCN	10.99	15.22
IMFCC	Original	5.63	34.87
	CMVN	9.82	11.35
	QCN	10.71	12.38
HFCC	Original	9.59	28.09
	CMVN	14.27	18.85
	QCN	14.22	19.11
LFCC	Original	8.64	18.82
	CMVN	12.96	15.44
	QCN	13.88	17.82
SFCC(Ours)	CMVN	9.36	10.33
	QCN	8.38	10.11

Note that, the results obtained from the Dev set are shown in **Table 6**, third column. One can see that the detection performance of the original features outperforms the modified features for almost all cases. The main reason is that the parameters settings for those original features are obtained by the original author with extensive experiments on DEV set. Therefore, compared to the modified features, the original features can achieve the best detection performance on the DEV set. However, on the Eval set containing more replay configurations, the modified features yield the lowest EER. Certainly, post-processing methods were beneficial for replay attack detection for most of the cases. Although other modified features have varying degrees of performance improvement, our proposed SFCC feature could achieve the best performance.

4.2.4 Effect of the Training set

In the previous experiment, we only used Tra set as a training set to evaluate our proposed algorithm. One knows that the size of the training set is also an important factor affecting the performance of the algorithm. Therefore, in the last experiments, we used a combination of Tra set and Dev set for training and tested on Eval set. For the post-processing method of the feature, we only used QCN and CMVN that performed better in the above experiments. The results for different training set are shown in **Fig. 3**.

It is clear that the performance of all algorithms using Tra set and Dev set as a training set can be improved. Therefore, we can conclude that the extension of the training set does improve the algorithm performance. This will encourage us to improve the detection performance from the perspective of data-augmentation. No matter what subset is used as the training set, our method is obviously superior to other methods. Our method yields lower EERs of 9.55% when using Tra and Dev as the training set.



(a) Features with CMVN (b) Features with QCN

Fig. 3. Comparison of the different training set

5. Conclusion

Replay attacks pose great security challenges to ASV systems. Many detection methods were developed to combat such attacks. In this work, we observe that the channel variability could harm detection accuracy of a detection method. In this work, a novel approach to replay attacks detection using low-frequency and high-frequency information of the spectrum is proposed. We first establish a mathematical model for replay speech and then propose a method to eliminate the influence of the channel. Finally, a novel feature is proposed. It is empirically found that using normalization as post-processing methods could improve detection performance. To evaluate the detection performance of our method and the effect of channel variability, we conduct experiments on the ASVspoof 2017 dataset. The results show that our approach outperforms other competing methods. Our results suggest that the normalization plays an indispensable role in the replay detection task.

References

[1] D. Zhu, B. Ma and H. Li, “Speaker verification with feature-space MAPLR parameters,” *IEEE Transactions on Audio Speech &Language Processing*, vol. 19, no. 3, pp. 505-515, April, 2011. Article (CrossRef Link).

[2] Z. Wu, T. Kinnunen, S. Chng and H.Li, “A study on spoofing attack in state-of-the-art speaker verification: the telephone speech case,” in *Proc. of Signal & Information Processing Association Annual Summit and Conference*, pp. 1-5, December 03-06, 2012. Article (CrossRef Link).

[3] W. Shang and M. Stevenson, “A Playback Attack Detector for Speaker Verification Systems,” in *Proc. of International Symposium on Communications Control and Signal Processing*, pp.1144-1149, March 12-14, 2008. Article (CrossRef Link).

[4] G. Jakub, G. Marcin and S. Rafal, “Playback attack detection for text-dependent speaker verification over telephone channels,” *Speech Communication*, vol.67, pp.143-153, March, 2015. Article (CrossRef Link).

[5] Z. Wang, G. Wei and H. He, “Channel pattern noise based playback attack detection algorithm for speaker recognition,” in *Proc. of International Conference on Machine Learning and Cybernetics*, pp.1708-1713, July10-13, 2011. Article (CrossRef Link).

[6] H. Delgado, M. Todisco and M. Sahidullah, “ASVspoof 2017 Version 2.0: meta-data analysis and baseline enhancements,” in *Proc. of Odyssey 2018 - The Speaker and Language Recognition Workshop*, pp.296-303, June 26-29, 2018. Article (CrossRef Link).

[7] M. Todisco, H. Delgado and N. Evans, “A New Feature for Automatic Speaker Verification Anti-Spoofing: Constant Q Cepstral Coefficients,” in *Proc. of Odyssey 2016 - The Speaker and Language Recognition Workshop*, pp.283-290, June 21-24, 2016. Article (CrossRef Link).

[8] L. Li, Y. Chen and D. Wang, “A Study on Replay Attack and Anti-Spoofing for Automatic Speaker Verification,” in *Proc. of INTERSPEECH 2017*, pp. 92-96, August 22-24, 2017. Article (CrossRef Link).

[9] K. Alluri, S. Achanta and S. Kadiri, “SFF Anti-Spoofer: IIIT-H Submission for Automatic Speaker Verification Spoofing and Countermeasures Challenge 2017,” in *Proc. of INTERSPEECH 2017*, pp. 107–111, August 22-24, 2017. Article (CrossRef Link).

[10] P. Nagarsheth, E. Khoury, and K. Patil, “Replay attack detection using DNN for channel discrimination,” in *Proc. of INTERSPEECH 2017*, pp. 97–101, August 22-24, 2017. Article (CrossRef Link).

[11] R. Font1, J. Espm, M. Cano, R. Font, “Experimental analysis of features for replay attack detection–Results on the ASVspoof 2017 Challenge,” in *Proc. of INTERSPEECH 2017*, pp. 7–11, August 22-24, 2017. Article (CrossRef Link).

[12] L. Lin, R Wang, D. Yan and C, Li, “A Replay Voice Detection Algorithm Based on Multi-feature Fusion,” in *Proc. of International Conference on Cloud Computing and Security*, pp. 289-299, June 8-10, 2018. Article (CrossRef Link).

[13] L. Lin, R. Wang and D. Yan, “A Replay Speech Detection Algorithm Based on Sub-band Analysis,” in *Proc. of Intelligent Information Processing*, pp. 337-345, October 19-22, 2018. Article (CrossRef Link) .

[14] M. Saranya, R. Padmanabhan and H. Murthy, “Replay Attack Detection in Speaker Verification Using non-voiced segments and Decision Level Feature Switching,” in *Proc. of SPCOM 2018*, pp. 332-336, June 2018. Article (CrossRef Link).

[15] B. Rafi, R. Murty and S. Naya, “A new approach for robust replay spoof detection in ASV systems,” in *Proc. of 2017 IEEE Global Conference on Signal and Information Processing*, pp. 1–5, November 14-16, 2018. Article (CrossRef Link).

[16] B. Atal, “Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification,” *Journal of the Acoustical Society of America*, vol. 55, no. 6, pp. 1304–1312, June 1974. Article (CrossRef Link).

[17] O. Viikki and K. Laurila, “Noise robust HMM-based speech recognition using segmental cepstral feature vector normalization,” in *Proc. of Esca Nato Workshop on Robust Speech Recognition for Unknown Communication Channels*, pp. 107–110. January 1997. Article (CrossRef Link).

[18] S. Yoshizawa, N. Hayasaka, N.Wada and Y. Miyanaga, “Cepstral gain normalization for noise robust speech recognition,” in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 209-212, May,17-21,2004. Article (CrossRef Link).

[19] H. Boril and L. Hansen, “Unsupervised equalization of Lombard effect for speech recognition in noisy adverse environment,” in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 3937-3940, April 19-24, 2009. Article (CrossRef Link).

[20] H. Boril and L. Hansen, “UT-Scope: Towards LVCSR under Lombard effect induced by varying types and levels of noisy background,” in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp.4472- 4475, May 22-27, 2011. Article (CrossRef Link).

[21] T. Kinnunen, M. Sahidullah, M. Falcone, “RedDots replayed: A new replay spoofing attack corpus for text-dependent speaker verification research,” in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5395-5399, March 5-9, 2017. Article (CrossRef Link).

[22] K. Lee, A. Larcher and G. Wang, “The reddots data collection for speaker recognition,” in *Proc. of INTERSPEECH 2015*, pp. 2996–3000, September 6-10, 2015 Article (CrossRef Link).

Lang Lin received the M.S.degree from Ningbo University in 2019, He is currently an engineer at Southeast Digital Economic Development Research Institute. His research interests include signal processing, multimedia security and forensic.

Rangding Wang is a professor at Ningbo University, China. He received the Ph. D. from Tongji University in 2004. His research interests mainly include multimedia security, digital watermarking for digital rights management, data hiding, and steganography.

Diqun Yan is an associate professor at Ningbo University, China. He received the Ph. D. from Ningbo University in 2012. His research interests include multimedia forensics and security.

Li, Dong (S’14–M’18) received the B.Eng. degree from Chongqing University in 2012, and the M.S. and Ph.D. degrees from the University of Macau in 2014 and 2018, respectively. He is currently an Assistant Professor with the Department of Computer Science, Faculty of Electrical Engineering and Computer Science, Ningbo University. His research interests include statistical image modeling and processing, multimedia security and forensic, and machine learning.