Interspeech 2018
2-6 September 2018, Hyderabad

Decision-level feature switching as a paradigm for replay attack detection

Saranya M. S.[1], *Hema A. Murthy*[1]

[1] Indian Institute of Technology Madras
{saranms,hema}@cse.iitm.ac.in

Abstract

A pre-recorded audio sample of an authentic speaker presented to a voice-based biometric system is termed as a replay attack. Such attacks can be detected by identifying the characteristics of the recording device and environment. An analysis of different recording devices indicates that each recording device affects the spectrum differently. It is also observed that each feature captures specific characteristics of recording devices. In particular, Mel Filterbank Slope (MFS) captures low-frequency information corresponding to that of the low-quality recording devices, while Linear Filterbank Slope (LFS) captures high-frequency information which corresponds to that of a high-quality recording device. The proposed approach uses MFS and LFS along with Mel Frequency Cepstral Coefficients (MFCC) and Constant-Q Cepstral Coefficients (CQCC) in a Decision-level Feature Switching (DLFS) paradigm to determine whether a given utterance is spoofed. The obtained results surpass the state-of-the-art Light Convolutional Neural Network (LCNN) based replay detection system with a relative improvement of 7.43% on the ASV-spoof-2017 evaluation dataset.
Index Terms: Replay attack detection, Filterbanks, MFS, LFS, Anti-spoofing, ASV-spoof-2017, Feature-switching, DLFS

1. Introduction

Automatic speaker verification (ASV) is the process of verifying an audio sample given a claim. ASV systems are robust to acoustic variations and zero-impostor threats [1] to a great extent, while the presentation attack detection (PAD) poses a real threat for commercial exploitation of voice as a biometric. ISO/IEC 30107-1:2016[1] defines PAD as the presentation of fake biometric sample to a biometric detection system. The process of this intentional circumvention of ASV systems is referred to as *spoofing*. Spoofing an ASV system can occur in any of the eight different stages as mentioned in [2]. Among the eight stages, the sample acquisition stage is the easiest to spoof. Four types of spoofing attacks that can occur at the sample acquisition stage are (i) speech synthesis (ii) voice conversion (iii) impersonation and (iv) replay attacks. The state-of-the-art ASV systems are robust against impersonation [3, 4] but fail when subjected to PAD attacks. The ASV-spoof-challenge was first proposed in 2015 to detect attacks based on speech synthesis, and voice conversion. Many spoof detection algorithms have been able to counter such attacks [5–7]. The replay attack is a process where the pre-recorded utterance of a genuine speaker is replayed by an impostor to access the ASV system. The ASV-spoof-2017 challenge was conducted to develop countermeasures to detect replayed utterances collected from widely varying acoustic conditions. A large variety of techniques have been proposed to detect replay attacks in the recent literature. A summary of the results of all the teams that participated in the challenge can be found in [8].

A Light Convolutional Neural Network (LCNN) with Max-Feat-Map (MFM) activation function was used for feature extraction in [9]. The Discrete Fourier Transform (DFT) spectrogram of the data is used as the input to the LCNN and features are extracted from the penultimate layer. A Gaussian Mixture Model (GMM) classifier built using these features resulted in an EER of 7.37% on the evaluation data. Late fusion of this GMM system along with other classifiers improved the EER to 6.73%. This system is considered as the state-of-the-art today [9]. In [10], systems trained on the development data and tested on the training data give better performance than any other system in the literature (Section 2). This system cannot be considered as the state-of-the-art as it violates the evaluation protocol.

While MFM-LCNN focuses on the design of an appropriate neural network architecture to learn the spoofed information, we propose the use of features that are relevant to capture the spectral characteristics of replayed utterances. The nature of the recording device plays a vital role in distinguishing genuine and replayed instances. A keen observation of the spectrogram of the genuine and replayed utterances from a low-quality microphone and high-quality microphone shows visible distortions in low- and high-frequency regions of the spectrum respectively (as highlighted by the ellipse in the insets in Figure 1).

The objective of this paper is to choose the feature that best identifies a replayed instance in a given environment. MFS features use the Mel scale which emphasizes low-frequency regions [11, 12], while LFS uses the linear frequency scale where the resolution at high-frequencies is better than that of MFS [13]. Similarly, other features namely, MFCC and CQCC may be useful for detection of other spectral artifacts. It is to be noted that MFS and LFS are used for the first time for replay attack detection[2], and it is shown that the performance of the replay attack detection system is far better than that of the MFCC/CQCC based systems. Since each of these features may contribute to the detection performance, instead of score fusion, a novel approach called Decision-level Feature Switching (DLFS) is proposed. DLFS chooses the feature that reposes maximum confidence during decision making. Standard two class GMM based replay detection systems are built using each of the features, namely, MFS, LFS, CQCC and MFCC.

The rest of the paper is organized as follows. Section 2 briefly discusses a variety of features used in the literature for replay detection. Sections 3 and 4 analyze the dataset and the proposed features respectively. Section 5 explains the DLFS approach for replay detection system. Section 6 describes the experimental setup and result analysis followed by the conclusion in Section 7.

[1] https://www.iso.org/standard/53227.html

[2] These two features are extensively used in the speaker verification literature [11–13].

10.21437/Interspeech.2018-1494

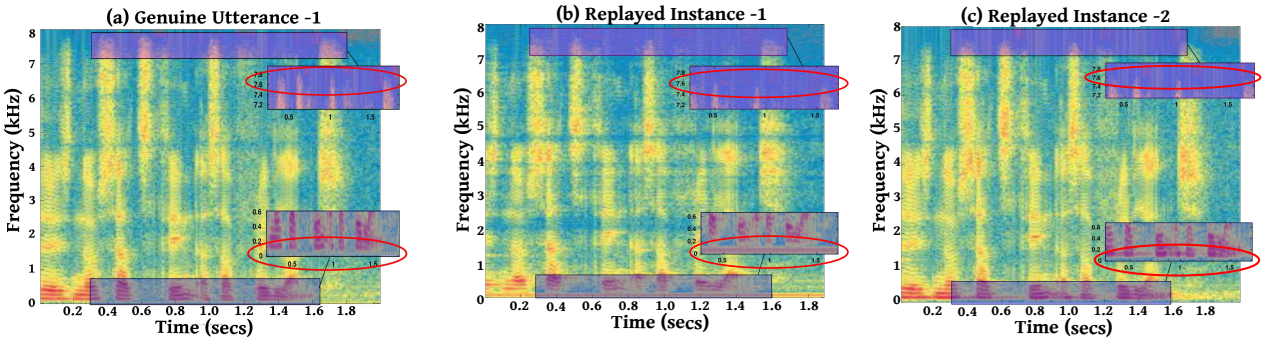

Figure 1: *Subplot (a) is the spectrogram of a genuine utterance. Subplots (b) and (c) are the spectrograms of the replayed instances of (a) recorded using the microphone of Samsung Galaxy S7 and a high-quality Zoom-H6 handy recorder. The high and low-frequency regions are magnified and encircled in the insets to highlight the key difference between the instances.*

2. Features used in prior work

An F-ratio probing tool is proposed in [14] to avoid the over-fitting problem of model training by using the mean and covariance of trained models. The effectiveness of this tool was demonstrated using Linear Frequency Cepstral Coefficients (LFCC) and Inverse MFCC (IMFCC) features. Authors in [15] claimed that the replayed utterances have imperfections at high-frequencies near the Nyquist rate due to the effect of anti-aliasing. Sub-band analysis using IMFCC and LPCC residual features was used to illustrate this effect. Data-augmentation was implemented in [16] to handle unseen real-time data in test conditions. Single Frequency Filtering Cepstral Coefficients (SFFCC) and High-frequency cepstral coefficients (HFCC) were proposed for replay detection task in [17] and [18] respectively. Experiments in [17] have shown that higher dimensional cepstral coefficients have more cues to detect the replayed utterances than the lower dimensional coefficients.

In [19] the authors proposed a feature selection approach where a ReliefF algorithm [20] and Minimum Redundancy-Maximum Relevance (MRMR) approach [21] is used together to detect the most discriminative and less redundant feature information. Authors also show that the inclusion of five static pitch related features detects the replayed utterance better. A comprehensive study of the relevance of different features for replay utterance detection can be found in [22]. The features considered include spectral sub-band based features along with IMFCC, LFCC, and LPCC. Instantaneous frequency (IF) was used along with Variable length Teager Energy Operator (VTEO) based Energy Separation Algorithm (ESA) in [23] to identify the replayed utterances. In [24] two new source features, namely peak-to-side-lobe ratio and epoch strength were proposed. An ensemble learning approach was proposed in [25] with different classifiers and various features like CQCC, MFCC, and Perceptual Linear Prediction coefficients (PLP). In all these efforts, GMM classifier or a Deep Neural Networks (DNN) or a Support Vector Machine (SVM) or a combination of one or more of these three classifiers were used for final classification. MFCC and CQCC were used as the baseline systems.

3. ASV-spoof-2017 Dataset

A subset of RedDots data collection [26] and its replayed derivatives constitute the ASV-spoof-2017 corpus [27]. The replayed derivatives were generated in natural conditions from different environments (E), using different recording devices (R) and playback devices (P). The ASV-Spoof-2017 dataset has three subsets namely training (train), development (dev), and evaluation (eval). Table 1 shows the number of utterances and the unique number of E-R-P (Environment-Recording device-Playback device) combinations in each subset.

Table 1: *Dataset Description.*

Subset	No. of Speakers	Total no. utterances	No. of Utterances		No. of Unique E-R-P	Total duration
			Genuine	Replayed		
Train	10	3016	1508	1508	3	2.22 hrs
Dev	8	1710	760	950	10	1.44 hrs
Eval	24	13306	1298	12008	57	11.94 hrs

The train-dev subset and train-eval subset have only one common E-R-P condition each whereas dev-eval subset has seven common E-R-P conditions. The remaining set of E-R-P conditions are not seen in the development or training data. This poses a real challenge for handling unseen E-R-P conditions.

4. MFS and LFS Features

The effectiveness of MFS and LFS features is evaluated in this Section. As MFS and LFS are used for the first time for replay attack detection, a brief description of these features is also presented. A replayed utterance contains both the genuine speaker's information and that of the E-R-P condition. A recorded utterance inevitably captures the reverberation of the recording environment unless a dedicated line-in and line-out are used. Apart from the reverberation and noise information, the *impact of the recording device* is the key factor that distinguishes the replayed instance from the genuine utterance. Figure 1(a) shows the spectrogram of a genuine utterance by Speaker-A, whereas Figures 1(b) and 1(c) show the replayed instances of the same utterance through a Samsung Galaxy S7 microphone and high-quality Zoom-H6 Handy recorder respectively. The spectrograms look more or less similar but (as indicated in Section 1), minor differences can be seen at both low- and high-frequency regions (as highlighted in the inset). Cepstral coefficients derived from the spectrum will simply average away these differences. MFS and LFS are spectral slope features and are therefore likely to magnify the artifacts in the spectrum as a function of frequency [11–13]. These features have been exploited extensively in speaker verification [11, 12] and diarization [13]. The block diagram of the feature extraction process for MFS and LFS is shown in Figure 2.

4.1. Mel Filterbank Slope Features

Mel filterbanks are based on the Mel scale, which gives higher resolution at low-frequencies when compared to that of high-frequencies [28]. MFS features are therefore expected to capture the variation in low-frequency regions in the replayed in-

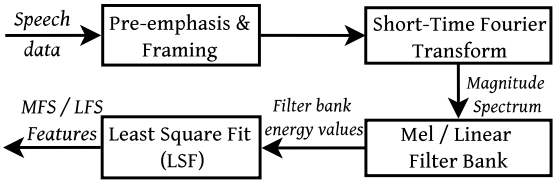

Figure 2: *Extraction of MFS/LFS features*

stances. Figure 1(b) is the spectrogram of a replayed instance, recorded using a Samsung Galaxy S7 microphone. The spectrogram shows the presence of constant low-frequency component throughout the replayed instance.

4.2. Linear Filterbank Slope Features

A digital recording device is typically associated with low-pass anti-aliasing filter (AAF) at a particular cut-off frequency. A replayed utterance undergoes this AAF at least twice which leads to imperfections in the spectrum at the Nyquist frequency [15]. Although the effect of noise and reverberation are compensated to a certain extent while using a high-quality recording device, the effect of AAF persists. Hence features extracted from filterbanks with low resolution in the high-frequency range, namely, CQCC, MFCC, and MFS, may not be able to identify these imperfections. Figure 1(c) is another replayed instance of the same genuine utterance. The replayed instance is recorded using a Samsung Galaxy S7 microphone that uses an active noise canceler for recording. Comparing Figure 1(a) and 1(c), it can be seen that the spectrogram is more or less identical to that of the genuine utterance (Figure 1(a)) except for artifacts near the Nyquist frequency. This effect is better emphasized by a linear filterbank rather than a Mel filterbank owing to its higher resolving power at high-frequencies.

5. DLFS Replay Detection System

As discussed in Section 4, MFS may capture almost all variations in low-frequency components and gross variation in high-frequency components. On the other hand, LFS may not capture the subtle low-frequency component variations as the resolution is the same at all frequencies. Thus using the feature that better identifies the difference will be a more suitable solution for replay detection task. DLFS essentially uses the appropriate feature for every utterance. Four individual GMM based replay detection systems are built using CQCC, MFCC, MFS and LFS features. The set of these four features are referred to as *candidate features*, and the feature-specific systems are referred to as *baseline systems* in the rest of the paper. DLFS primarily uses the scores from these GMM systems effectively to improve replay detection performance. The process of developing the baseline and DLFS systems is elaborated in the following subsections.

5.1. Baseline Systems

The genuine and replayed utterances from the training data are used to form genuine GMM ($\lambda_{\mathsf{G}f}$) and spoofed GMM ($\lambda_{\mathsf{S}f}$) respectively where $f \in \mathcal{F}$, and $\mathcal{F}$={MFS, LFS, CQCC, MFCC}. Development data is not used for model training. Final score $\mathcal{S}_f(t)$ for every test utterance t, is computed using log-likelihood scores ($\wedge$) as follows:

$$\wedge_{\mathsf{G}f}(\mathcal{X}) = \mathrm{p}(\mathcal{X}|\lambda_{\mathsf{G}f}) \text{ and } \wedge_{\mathsf{S}f}(\mathcal{X}) = \mathrm{p}(\mathcal{X}|\lambda_{\mathsf{S}f}) \quad (1)$$

$$\mathcal{S}_f(t) = \wedge_{\mathsf{G}f}(\mathcal{X}) - \wedge_{\mathsf{S}f}(\mathcal{X}) \quad (2)$$

where $\mathcal{X} = \{\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_n}\}$ is the set of n feature vectors that make up the test utterance t. Based on the score $\mathcal{S}_f(t)$, a label is assigned to every t as

$$l_f = \begin{cases} \text{genuine}, & \text{if } \mathcal{S}_f(t) > 0 \\ \text{spoofed}, & \text{if } \mathcal{S}_f(t) < 0 \end{cases} \quad (3)$$

5.2. Cohort Normalization for Replay Detection

T-norm based score normalization is performed using cohorts *within the same feature space* [29]. Since replay detection task is a two-class problem, a test utterance classified as genuine (class-1) will have cohorts from the spoofed class (class-2) and vice-versa. The cohorts for the test utterances are chosen from development data[3]. Scores of dev-data trials are pre-computed using Equations (1) and (2). The score normalization with cohort scores is performed as follows:

1. The score $\mathcal{S}_f(t)$ and label l_f of a test utterance t in a feature stream f is obtained using Equations (2) and (3).
2. If the predicted label l_f is *genuine*, the C closest pre-computed scores from the spoofed class of the dev-data are considered as cohort scores for t in the feature stream f and vice-versa.
3. Using the mean (μ_{Cf}) and standard deviation (σ_{Cf}) of the C cohort scores from feature stream f chosen in Step-2, the normalized score $\mathcal{N}_f(t)$ is determined as

$$\mathcal{N}_f(t) = \frac{\mathcal{S}_f(t) - \mu_{Cf}}{\sigma_{Cf}}, \ f \in \mathcal{F} \quad (4)$$

Score normalization is implemented on the eval-data using the dev-data. The optimization of parameters for score normalization is performed by dividing the dev-data in the ratio of 70:30, where 70% trials are used for testing, and 30% trials are used to choose cohort scores. The value of C is estimated empirically for every trial t, and it is found that $C > 30$ does not change μ_{Cf} and σ_{Cf} significantly. This process is repeated for every feature $f \in \mathcal{F}$ and these normalized scores of the baseline systems are used for DLFS.

5.3. DLFS Architecture

LFS and MFS emphasize different portions of the spectrum as shown in Section 4. Similarly, MFCC and CQCC may be relevant for other types of environments. In speaker verification, feature switching [30–32] identifies the optimal feature space for every enrolled speaker during training, and testing is performed in the optimal feature space of the claimed speaker. For replay detection, a variant of feature switching termed as DLFS is proposed. In the replay detection task, since the test utterances do not have any claim, the optimal feature cannot be estimated apriori. The optimal feature is, therefore, determined during the test using the scores and labels of all candidate features $\mathcal{F}$:

1. The trial utterance (t) is tested against all the baseline systems, and the correponding normalized scores $\mathcal{N}_f(t)$ and labels l_f are obtained using Equations (1)-(4).
2. Voting is used to determine the final label ($\hat{l}$) of the utterance. Using $\hat{l}$, the final score $\hat{\mathcal{N}}(t)$ is computed as

$$\hat{\mathcal{N}}(t) = \begin{cases} \max\{\mathcal{N}_f(t)\}, & \text{if } \hat{l} = \text{genuine}, f \in \mathcal{F} \\ \min\{\mathcal{N}_f(t)\}, & \text{if } \hat{l} = \text{spoofed}, f \in \mathcal{F} \end{cases} \quad (5)$$

[3] Since dev-data is not used for model training, the protocol of ASV-spoof-2017 challenge is not violated [27].

The f that corresponds to the final score $\hat{\mathcal{N}}(t)$ is chosen as the optimal feature for t. Selection of maximum or minimum score increases the confidence in classification.

3. If the number of candidate features is even, the votes can get evenly distributed for both the classes. In such cases, $\hat{\mathcal{N}}(t)$ and $\hat{l}$ are calculated as follows:

$$\text{if } (|a| > |b|) \implies \hat{l} = \text{genuine} \; ; \hat{\mathcal{N}}(t) = a \quad (6)$$

$$\text{if } (|a| < |b|) \implies \hat{l} = \text{spoofed} \; ; \hat{\mathcal{N}}(t) = b \quad (7)$$

where $a = \max\{\mathcal{N}_i(t)\}$ and $b = \min\{\mathcal{N}_j(t)\}$. i and j are the subsets of $\mathcal{F}$ with labels genuine and spoofed respectively : $(i \cap j) = \emptyset$ and $(i \cup j) = \mathcal{F}$. Choosing the best feature increases the discriminability of the classifier.

6. Experiments and Analysis

Features are extracted from genuine and replayed utterances of the training data. The standard 25 ms frame size and 10 ms frame shift are used to extract all four candidate features ($\mathcal{F}$). No voice activity detection (VAD) is performed on the data since non-speech segments and silence regions are more likely to contain E-R-P information [33]. A filterbank of 100 filters and 70 filters are chosen empirically to extract MFS and LFS features respectively. As mentioned in [25] our experiments show poor performance with cepstral mean subtraction and variance normalization (CMVN). Hence only cepstral mean subtraction (CMS) is applied. Separate GMM classifiers are trained for every feature space f, and the number of mixture components in each GMM is identified empirically, such that the performance on dev-data is enhanced. $\mathcal{N}_f(t)$ for every test utterance in the feature space f is calculated, and the EER is computed using Bosaris toolkit [34]. The accuracy of every system is calculated based on the ratio between the total number of hits and the total number of utterances in the dataset. The performance of all four baseline systems after score normalization is listed in Table 2.

Table 2: *Results of baseline systems (in %).*

System	Feature	Development Data		Evaluation Data	
		EER	Accuracy	EER	Accuracy
CQC_BL	CQCC	4.49	83.27	26.40	29.99
MFC_BL	MFCC	7.56	92.22	11.28	72.53
MFS_BL	MFS	**3.58**	**95.55**	**7.82**	**77.16**
LFS_BL	LFS	5.13	95.38	9.82	71.77

As observed in all prior work, using information from more than one feature space improves the performance. DLFS approach spontaneously chooses the optimal feature that better discriminates an utterance from the other class (Section 5.3). The DLFS systems are made with all combinations of features from $\mathcal{F}$, and the three best-performing systems are reported in Table 3. Unlike score fusion systems, the DLFS systems do not require a separate weight learning process. Since the performance of score fusion systems is not superior to the DLFS systems, the results are not reported in the paper.

MFM in the state-of-the-art system acts as a feature selector, training the MFM-LCNN with 25 different layers and 371K parameters [9] is computationally expensive. Every test utterance is passed through the LCNN and features are extracted. On the other hand, domain information along with a set of handcrafted features with GMMs seems to perform equally well. DLFS chooses the most suitable feature for each trial from this feature set. DLFS uses the trials' scores from individual feature-specific systems and does not require separate training or weight learning algorithms.

Table 3: *EERs of DLFS systems (in %).*

System	Feature[3]	Dev-Data	Eval-Data
DLFS_LS	LFS | MFS	4.13	6.65
DLFS_MLS	MFCC | LFS | MFS	3.98	**6.23**
DLFS_CMLS	CQCC | MFCC | LFS | MFS	**3.30**	6.60

The distribution of optimal features of development and evaluation data of the DLFS_MLS system is shown in Table 4. The misses and false alarms of the DLFS_MLS and DLFS_CMLS systems in all the candidate feature spaces are shown in Table 5. The statistics reported in the table are calculated with respect to the total number of trials in the evaluation data. From this table, it is evident that MFS and LFS detect both genuine and replayed instances better than MFCC for DLFS_MLS system. Although LFS has less number of hits in DLFS_CMLS system, the number of false alarms is also less compared to other features.

Table 4: *Distribution of features (the number of trials in each feature space) in DLFS_MLS system*

Dataset	MFCC	LFS	MFS
Dev	247	1020	443
Eval	2832	4477	5997

Table 5: *Statistics of two best DLFS systems on evaluation data.*

Opt. Feature	DLFS_MLS			DLFS_CMLS		
	Hits	Misses	False Alarms	Hits	Misses	False Alarms
CQCC	-	-	-	1372	15	2582
MFCC	1778	6	1048	1897	4	866
MFS	5373	8	616	5022	7	579
LFS	2999	54	1424	693	9	260

7. Conclusion

Replayed utterances contain perceptible recording device information in either the high- or low-frequency regions. Features that emphasize these characteristics are required. Two alternatives features namely, MFS and LFS are proposed for the first time for the task of replay attack detection. MFS resolves low-frequency regions better than LFS, while LFS resolves high-frequency regions better than MFS. The proposed features in tandem with the feature switching paradigm outperform the state-of-the-art LCNN based system with a relative improvement of 7.43%. The training and evaluation E-R-P hardly overlap; nevertheless, the system does scale well.

8. Acknowledgements

We would like to thank the ASV-Spoof-2017 organizers for providing the new dataset for the replay detection task.

9. References

[1] S. J. Elliott, *Zero Effort Forgery*. Boston, MA: Springer US, 2009, pp. 1411–1414.

[2] Z. Wu *et al.*, "Spoofing and countermeasures for speaker verification: A survey," *Speech Communication*, vol. 66, pp. 130 – 153, 2015.

[3] The symbol '|' represents *exclusive OR*. Feature A *(OR)* B will be the optimal feature.

[3] J. Mariéthoz and S. Bengio, "Can a Professional Imitator Fool a GMM-Based Speaker Verification System?" IDIAP, Tech. Rep. IDIAP-RR-61-2005, 2005.

[4] R. G. Hautamäki, T. Kinnunen *et al.*, "i-vectors meet imitators: on vulnerability of speaker verification systems against voice mimicry." in *INTERSPEECH*, 2013, pp. 930–934.

[5] T. B. Patel and H. A. Patil, "Significance of Source-Filter Interaction for Classification of Natural vs. Spoofed Speech," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 4, pp. 644–659, June 2017.

[6] Z. Wu *et al.*, "ASVspoof: The Automatic Speaker Verification Spoofing and Countermeasures Challenge," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 4, pp. 588–604, June 2017.

[7] T. B. Patel and H. A. Patil, "Cochlear Filter and Instantaneous Frequency Based Features for Spoofed Speech Detection," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 4, pp. 618–631, June 2017.

[8] T. Kinnunen *et al.*, "The ASVspoof 2017 challenge: Assessing the limits of replay spoofing attack detection," in *INTERSPEECH*, Aug 2017, pp. 1–6.

[9] G. Lavrentyeva *et al.*, "Audio replay attack detection with deep learning frameworks," in *INTERSPEECH*, Aug 2017, pp. 82–86.

[10] H. Delgado *et al.* (2018) ASVspoof 2017 Version 2.0: meta-data analysis and baseline enhancements. [Online]. Available: http://www.asvspoof.org/data2017/ASVspoof2017_V2_Odyssey_2018.pdf

[11] Hema A. Murthy *et al.*, "Robust text-independent speaker identification over telephone channels," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 5, pp. 554–568, Sep 1999.

[12] S. Madikeri and H. A. Murthy, "Mel Filter Bank energy-based Slope feature and its application to speaker recognition," in *National Conference on Communications (NCC)*, Jan 2011, pp. 1–4.

[13] S. Madikeri and H. Bourlard, "Filterbank slope based features for speaker diarization," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 111–115.

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

[15] M. Witkowski *et al.*, "Audio Replay Attack Detection Using High-Frequency Features," in *INTERSPEECH*, Aug 2017, pp. 27–31.

[16] W. Cai *et al.*, "Countermeasures for Automatic Speaker Verification Replay Spoofing Attack : On Data Augmentation, Feature Representation, Classification and Fusion," in *INTERSPEECH*, 2017, pp. 17–21.

[17] K. N. R. K. R. Alluri *et al.*, "SFF Anti-Spoofer: IIIT-H Submission for Automatic Speaker Verification Spoofing and Countermeasures Challenge 2017," in *INTERSPEECH*, Aug 2017, pp. 107–111.

[18] P. Nagarsheth *et al.*, "Replay attack detection using DNN for channel discrimination," *INTERSPEECH*, pp. 97–101, Aug 2017.

[19] X. Wang, Y. Xiao, and X. Zhu, "Feature selection based on CQCCs for automatic speaker verification spoofing," *INTERSPEECH*, pp. 32–36, Aug 2017.

[20] I. Kononenko, "Estimating attributes: Analysis and extensions of RELIEF," in *Machine Learning: ECML-94*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, pp. 171–182.

[21] H. Peng *et al.*, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[22] R. Font *et al.*, "Experimental analysis of features for replay attack detection–Results on the ASVspoof 2017 Challenge," *INTERSPEECH*, pp. 7–11, Aug 2017.

[23] H. A. Patil *et al.*, "Novel Variable Length Teager Energy Separation Based Instantaneous Frequency Features for Replay Detection," *INTERSPEECH*, pp. 12–16, Aug 2017.

[24] S. Jelil *et al.*, "Spoof detection using source, instantaneous frequency and cepstral features," *INTERSPEECH*, pp. 22–26, Aug 2017.

[25] Z. Ji *et al.*, "Ensemble learning for countermeasure of audio replay spoofing attack in asvspoof2017," pp. 87–91, Aug 2017.

[26] K.-A. Lee *et al.*, "The RedDots data collection for speaker recognition," in *INTERSPEECH*. ISCA, 2015, pp. 2996–3000.

[27] T. Kinnunen *et al.*, "Automatic Speaker Verification Spoofing and Countermeasures Challenge (ASVspoof)," Feb 2017. [Online]. Available: http://www.spoofingchallenge.org/index2017.html

[28] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, Aug 1980.

[29] R. Auckenthaler *et al.*, "Score normalization for text-independent speaker verification systems," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 42–54, 2000.

[30] Saranya M. S. *et al.*, "Feature-switching: Dynamic feature selection for an i-vector based speaker verification system," *Speech Communication*, vol. 93, pp. 53–62, 2017.

[31] T. Asha *et al.*, "Feature switching in the i-vector framework for speaker verification," in *INTERSPEECH*, Sep 2014, pp. 1125–1129.

[32] R. Padmanabhan *et al.*, "Acoustic feature diversity and speaker verification." in *INTERSPEECH*, 2010, pp. 2110–2113.

[33] Z. Chen *et al.*, "ResNet and Model Fusion for Automatic Spoofing Detection," in *INTERSPEECH*, Aug 2017, pp. 102–106.

[34] N. Brümmer *et al.* (2013) The BOSARIS Toolkit: Theory, Algorithms and Code for Surviving the New DCF.