The 2017 Conference on Computational Linguistics and Speech Processing
ROCLING 2017, pp. 264-275

SUT System Description for Anti-Spoofing 2017 Challenge

Department of Computer Engineering
Sharif University of Technology, Tehran, Iran
Mohammad Adiban, Hossein Sameti, Nooshin Maghsoodi, Sajjad Shahsavari
adiban@ce.sharif.edu, sameti@sharif.edu, nmaghsoodi@ce.sharif.edu,
mrshahsavari@ce.sharif.edu

Abstract

Reliability of Automatic Speaker Verification (ASV) systems has always been a concern in dealing with spoofing attacks. Among these attacks, replay attack is the simplest and the easiest accessible method. This paper describes a replay spoofing detection system applied to ASVspoof2017 corpus. To reach this goal, features such as Constant-Q Cepstral Coefficients (CQCC), Modified Group Delay (MGD), Mel Frequency Cepstral Coefficients (MFCC), Relative Spectral Perceptual Linear Predictive (RASTA-PLP) and Linear Prediction Cepstral Coefficients (LPCC), and different classifiers including Gaussian Mixture Models (GMM), Multi-Layer Perceptron (MLP), Support Vector Machine (SVM) and Linear Gaussian (LG) classifier have been employed. We also used identity vector (i-vector) based utterance representation. Finally, scores of different subsystems have been fused to construct the proposed system. The results show that the best performance is attained using this score level fusion.

Keywords: Spoofing Attack, Automatic Speaker Verification, Replay, ASVspoof2017

1. Introduction

In recent years, there has been growing interest to develop Automatic Speaker Verification systems. ASV systems aim to verify the speaker's identity using his/her speech. There are many mass-market applications of ASV systems, such as phone banking and trading, password resetting, access control in smart phones, credit card activation etc. [1, 2]. Due to the development of ASV systems, concerns about spoofing attacks and security of these systems are increasing. To study spoofing approaches and their threats, the ASVspoof2015 [3] and ASVspoof2017 [4] challenges are introduced. Spoofing attacks include four main approaches: impersonation, speech synthesis, voice conversion and replay [3, 5].

Among them, replay attack is the most easily accessible approach available with low technology recording devices, such as voice recorder, laptop, smartphone etc. [5]. These devices may incur different noises during spoofing attacks such as channel or convolutional noise and quantization noise. The channel noise could be originated from the different recording devices, different recording environment and changes in the distance to the microphone. Therefore, in spoofing attacks context, this noise occurs owing to the fact that replayed speech is recorded by two

devices and one loudspeaker [2, 6]. Quantization noise is due to analog-to-digital conversion. These distortions cause a mismatch between the genuine and replayed speech pattern. This mismatch can be detected by training a classifier using cepstral-based or spectral-based features. According to the replayed speech threats, the ASVspoof2017 challenge is introduced concentrating on the replay spoofing attacks. This challenge aims to analyze vulnerability of ASV systems and its countermeasures in the face of replay attack [4]. In this study, we focus on the replay spoofing attacks and countermeasures based on the ASVspoof2017 dataset.

In this work, we utilized six different features, Comprising Constant Q Cepstral Coefficient (CQCC), Modified Group Delay (MGD), Mel-Frequency Cepstral Coefficient (MFCC), Relative Spectral Perceptual Prediction (RASTA-PLP), Linear Prediction Cepstral Coefficient (LPCC) and identity vector (i-vector) [7]. We used four types of classifiers: Gaussian Mixture Model (GMM), Multi-Layer Perceptron (MLP), Support Vector Machine (SVM), and Linear Gaussian (LG) Model. Within this framework, we investigated fusion of these different subsystems. The rest of this paper is organized as follows: in Section 2 ASVspoof2017 dataset and metrics are introduced. An overview of the above mentioned features is provided in Section 3 and the classifiers are described in Section 4. Eventually, the experimental results are reported in Section 5 and the conclusion is presented in Section 6.

2. Dataset and Metrics

The ASVspoof2017 challenge is introduced to provide a standard database containing genuine and spoofed speech by replay attack [4, 8]. Spoofed speech is created in 179 sessions with 125 different configurations by applying different replay techniques to a given utterance as described in [9]. This corpus is based on the RedDots [10] corpus and consists three subsets: training, development and evaluation. The training subset is generated from 10 male speakers and contains 1508 genuine utterances and 1508 spoofed speech trials by replay attack. Spoofed speech is created in six sessions with three different configurations. The development subset is comprised of 760 genuine and 950 spoofed utterances and generated by 8 speakers. Furthermore, the spoofed utterances are generated from 10 different replay sessions with different playback and recording devices. The evaluation data includes 13306 utterances generated from 1298 genuine and 12008 spoofed trials. The statistics of each subset are summarized and illustrated in Table 1. It should be noted that the number of spoofed trials was 14420 originally and it was decreased to 12008 after modification by organizer. There are six evaluation conditions in this corpus. For each there is a disjoint set of replay trials. Replay trials in condition C1 have a remarkable amount of background noise or channel distortion so that they are almost easily detectable, whereas replay trials in condition C6 are of high quality thus they are relatively more difficult to detect. More details about the corpus and the ASVspoof2017 challenge can be found in [4, 11].

Table 1: Statistic of the ASVspoof 2017 database.

Subset	#spk	#Replay sessions	#Replay config	#Utterance	
				Non-replay	Replay
Training	10	6	3	1508	1508
Devel.	8	10	10	760	950
Eval.	24	163	112	1298	12008
Total	42	179	125	3566	14466

2.1. Evaluation Metrics

In this task the metric of evaluation is based on Equal Error Rate (EER). Therefore, we assign a score to each trial, then let define $P_{fa}(\theta)$ as the false alarm and $P_{miss}(\theta)$ as the miss rates at threshold θ:

$$P_{fa}(\theta) = \frac{\#\{replay - trials > \theta\}}{\#\{Total - replay - trials\}} \tag{1}$$

$$P_{miss}(\theta) = \frac{\#\{non - replay - trials \leq \theta\}}{\#\{Total - non - replay - trials\}} \tag{2}$$

Now EER is computed [4]:

$$EER = P_{fa}(\theta_{EER}) = P_{miss}(\theta_{EER}) \tag{3}$$

where θ_{EER} is the value of the parameter θ when P_{fa} equals P_{miss}.

3. Features

3.1. Constant Q Cepstral Coefficients

We first used the Constant Q Cepstral Coefficients (CQCC) feature [12]. This feature is based upon the Constant Q Transform (CQT), initially proposed in the field of music processing [13]. In the recent years the CQT has been widely used to analyze, classification and separation of audio signals, and has achieved significant results [14, 15, 15]. The CQT uses geometrically spaced frequency bins [12]. Considering Fourier-based approaches, using regular spaced frequency, makes them variable in the shape and width (the Q-factor) of the filter in the frequency domain [16], while CQT engages a constant Q factor along the entire spectrum. One advantage of CQT over Fourier-transform is related to their frequency and temporal resolution. In other word, Fourier-transform yields low frequency resolution in lower frequency and low temporal resolution in higher frequency while CQT has high resolution in both cases [17]. More details about CQT are given in [12]. For extracting CQCC, first we compute the CQT for the discrete time domain signal $x(n)$. Then in the case of speech signals, the spectrum is usually obtained using the discrete Fourier transform (DFT). In the next step the cepstrum in a time sequence $x(n)$ is obtained using the inverse transformation in the spectrum logarithm whereas the inverse transformation is normally implemented with the discrete cosine transform (DCT). The steps for extracting the CQCC are depicted in Fig. 1.

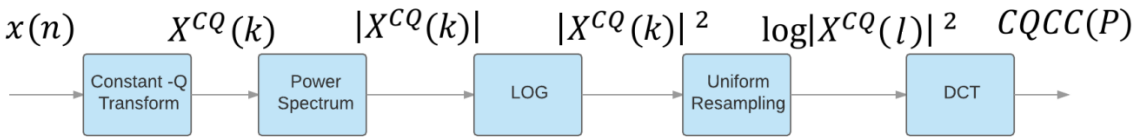

Figure 1: Block diagram of CQCC feature extraction [12].

3.2. Modified Group Delay

The Modified Group Delay (MGD) [18] has been used in phoneme recognition for many years [19]. MGD contains both magnitude and phase information [20] and can be used as an informative cue for replay speech detection. Phase information is vital factor in speech coding. Its role is contributed to keeping the balance between redundant information reduction and quality of the received speech. Accordingly, perceptual threshold value is fixed and we try to keep phase quantization error below the mentioned value. This value is experimentally determined by listener(s) given signals with noticeable difference in their face but not perceptually recognizable. Several methods of spoofing countermeasures based on features using magnitude and phase information are introduced in [21]. Modified group delay is originated from the group delay (GD) which is the derivative of phase spectrum with respect to frequency:

$$GD(t,\omega) = princ\{\theta(t,\omega) - \theta(t,\omega-1)\} \tag{4}$$

where princ is the mapping function and adds integer numbers of 2π to map the input onto $[-\pi;\pi]$ interval. The robustness of the group delay in the face of transmission channel and ambient noise has been studied in [22, 23, 24]. Considering GD, the modified group delay can be computed according to:

$$MGD = \frac{\tau(t,\omega)}{|\tau(t,\omega)|}|\tau(t,\omega)|^{\alpha} \tag{5}$$

$$\tau(t,\omega) = \frac{X_R(t,\omega)Y_R(t,\omega) + X_I(t,\omega)Y_I(t,\omega)}{|S(t,\omega)|^{2\gamma}} \tag{6}$$

here, $X(t;\omega)$ is the fast Fourier transform (FFT) of speech signal $x(n)$ and $Y(t;\omega)$ is the fast Fourier transform of the $nx(n)$ where $x(n)$ is re-scaled form of $x(n)$, R is the real part of spectrum, and I is the imaginary part of spectrum. Furthermore, $S(t;\omega)$ is a smoothed configuration of $X(t;\omega)$.

3.3. Mel Frequency Cepstral Coefficients

Commonly, phase information remains in the original utterance in replayed speech [5]. Therefore, amplitude-based features such as MFCC provides the important information and a reliable means of detecting replay spoofing attack. Experimental result on ASVspoof2017 as are described in Section 5 shows fusion of MFCC and phase-based features like MGD achieves significant performance to spoofing detection.

3.4. Relative Spectral Perceptual Linear Predictive

The Perceptual Linear Predictive (PLP) [25] is a feature extraction method based on short-term spectrum. As less sensitivity of human perception to the speech spectral factors [25], such features are very informative on spectral analysis domain. RASTA-PLP is the modified version of PLP making it more robust to linear spectral distortions [26]. It should be noted that there is almost no preference regarding the performance in applying RASTA-processing in PLP comparing with applying PLP for the clean data, however, the recognizer would be much robust respect to the factors such as microphone quality or its position to the mouth in the case of employing RASTA-PLP. Therefore, the efficiency of RASTA-PLP in the face of convolutional noise is clear [26].

3.5. Linear Prediction Cepstral Coefficient

Linear Prediction Coding (LPC) is one of the most popular and powerful methods that gives us the basic parameters of the speech signals and is widely used in speaker recognition [27]. LPCC coefficients can be obtained from LPC using autocorrelation method. One of the properties of LPCC is its high sensitivity to quantization noise. Speech processing systems using LPCC feature have achieved high performance dealing with speech recorded in the noise-free conditions [27]. This feature can be useful for replay spoofed speech detection especially when it is fused with MFCC and RASTAPLP.

3.6. Identity Vector(i-vector)

As mentioned before, in this work we have also used i-vector as a representation for each utterance. For i-vector extraction, firstly, a large GMM (*e.g.* with 2048 components), called Universal Background Model (UBM), is trained on the sufficient data. Using UBM, each utterance is modeled by a super vector which is produced by concatenating mean vectors of Gaussian components in the UBM. In the factor analysis viewpoint, this super vector, M, can be modeled as:

$$M = m + Ty \tag{7}$$

where T is a low rank matrix representing speaker and channel variability jointly, m is the speaker and channel independent super vector defined by the means of UBM components and y is a latent variable with standard normal distribution. In this model, T is denoted as i-vector extractor or total variability space, and y is representing i-vector. The next stage is training the i-vector extractor, and then, using i-vector extractor, the super vector corresponding to each utterance will be mapped to a vector with lower dimension (*e.g.* 100 or 200), the i-vector.

4. Classifiers

4.1. Gaussian Mixture Models

In this work, Gaussian mixture model (GMM) is used as classifier. We trained two GMMs by EM iterations, one for the genuine speech and the other for the spoofed speech. In the next step, the score for each trial is obtained by computing log likelihood ratio:

$$LLR(S) = logP(S|\theta_{genuine}) - logP(S|\theta_{spoof}) \tag{8}$$

where S is a feature vector corresponding to the test utterance and $\theta_{genuine}$ and θ_{spoof} denote the GMMs for genuine and spoof speech, respectively.

4.2. Multi-Layer Perceptron

We also used Multi-Layer Perceptron (MLP) as a discriminative classifier to compute the posterior probability of each genuine and spoof class for the given input feature vector as named before. We trained the networks with mini-batch stochastic gradient descent (MSGD) optimization algorithm for minimizing cross entropy objective function. The output layer consists of two neurons with softmax activation function that represent posterior probabilities of the genuine and spoof classes. In the following, the output score for each given speech signal sequence is obtained by computing log likelihood ratio (LLR):

$$LLR(S) = logP(genuine|S) - logP(spoof|S) \tag{9}$$

Where S is the input sequence and $P(genuine|S)$ and $P(spoof|S)$ are posteriors of genuine and spoofed trials, respectively.

4.3. Support Vector Machine

In this paper, Gaussian Support Vector Machine (SVM) is applied to the problem of spoofed/non-spoofed classification. In this manner, a binary SVM is trained using i-vector features to discriminate genuine from replayed speech. Accordingly, we obtain a binary function that determines the posterior probability of spoofed and genuine trials based on Logistic regression.

4.4. Linear Gaussian

A linear Gaussian model [28] is a Bayes net model. The popularity of linear Gaussian models comes from analytical properties of Gaussian processes [9]. In this model all the variables are Gaussian and the output of the linear system obtained by Gaussian distributed input is also Gaussian distributed. In this work we have used a linear Gaussian model for classification of genuine and spoofed speech. For the linear Gaussian, we used system identification methods based on Expectation Maximization (EM) algorithm to maximize the likelihood of the observed data, and we have used i-vector as the input to the linear Gaussian.

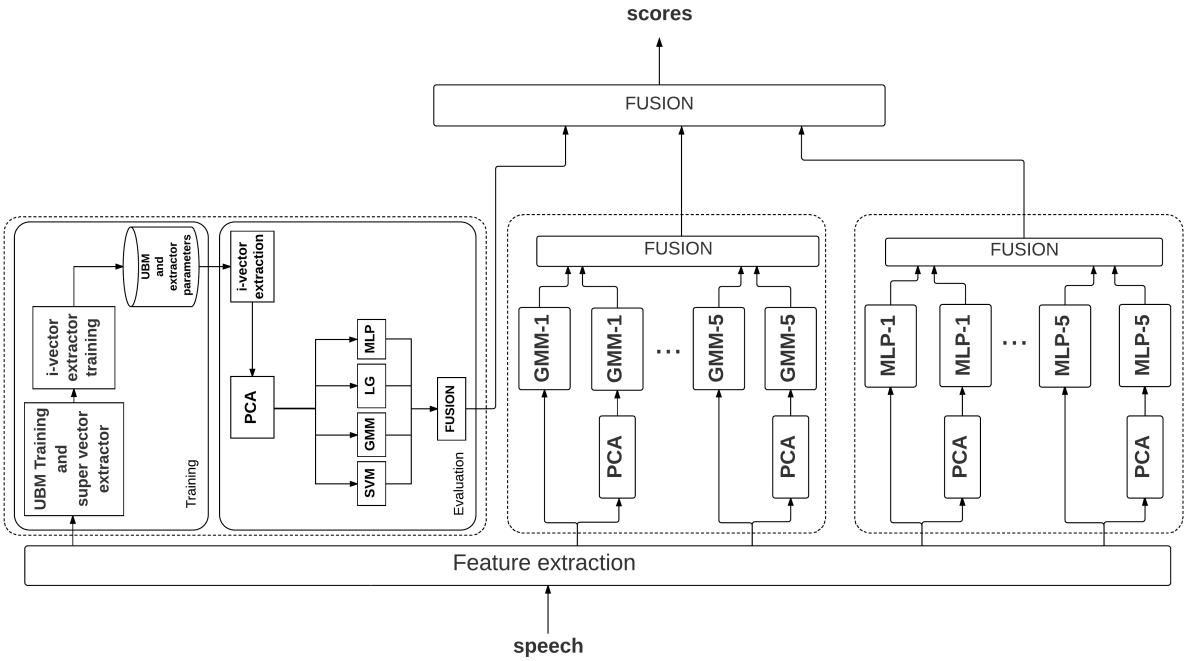

Figure 2: Proposed SUT system structure

5. Experimental Results

5.1. Baseline System

ASVspoof2017 challenge has introduced a baseline system to detect replay/non-replay trials [11]. This system is used as our baseline in this paper. This system uses a common GMM back-end classifier and CQCC features. The attained EER of the baseline was reported 24.65%. It is worth mentioning that 20 participants from all 49 participants could achieve better performance in terms of EER compared to the baseline system, in this challenge. Obviously, the reported performance shows the difficulty level of the challenge.

5.2. SUT System Structure

In this work, we have applied 90-dimensional CQCC, 427-dimensional MGD (with two tuning parameters γ and α set to 0.9 and 0.4, respectively), 24-dimensional MFCC, 13-dimensional RASTA-PLP and 12-dimensional LPCC for each GMM and MLP. The GMM components were set to be 512 (except for the GMM used in the i-vector based system). Furthermore, each MLP (except for the MLP used in the i-vector based system) was trained with 2 hidden layers each containing 256 neurons with rectified linear unit (ReLU) activation function. In i-vector based part the number of UBM components are 2048, and the dimensionality of the i-vetor extractor was set to 200. In addition, for training the classifiers in i-vector based system, we used 4-components GMM, one layer MLP with 30 neurons and applied one Gaussian in linear Gaussian.

Table 2: EER (%) for different systems in *development* set.

Systems	**EER%**
GMM1 (CQCC + PCA)	**6.12**
GMM2 (MGD + PCA)	31.33
GMM3 (MFCC + PCA)	20.45
GMM4 (RASTA-PLP + PCA)	17.69
GMM5 (LPCC + PCA)	29.87
FUSION (GMM-all features)	**4.23**
MLP1 (CQCC + PCA)	**21.36**
MLP2 (MGD + PCA)	41.70
MLP3 (MFCC + PCA)	28.15
MLP4 (RASTA-PLP + PCA)	25.48
MLP5 (LPCC + PCA)	39.17
FUSION (MLP-all features)	**19.02**
i-vector	**5.38**
FUSION (GMM + i-vector)	3.81
FUSION (MLP + i-vector)	5.96
FUSION (GMM + MLP)	4.66
FUSION (GMM + MLP + i-vector)	**3.10**

5.3. Development Set

Table 2 represents the EER obtained by our different countermeasure systems and their fusion of the development set. Consequently, we trained 5 GMMs and 5 MLPs as classifiers, then the obtained scores by each classifier were fused. As the input of each classifier we used one feature vector and its Principal Component Analysis (PCA) vector. Experimental results show that applying feature PCA transformation, especially for CQCC, significantly improves the results. Fig. 2 illustrates our proposed system. We used all features and their PCA transformation for training GMMs and MLPs. The results obtained from GMM classifiers and their fusion are presented in the first part of the Table 2. It shows CQCC and its PCA performs better than others. As shown in Table 2, the best EER is achieved by fusion of CQCC-based GMM classifier with the others features and it is about 4.23%. The second part of the Table 2 denotes the results of MLP classifiers. It is observed that the EER values of MLP are not as good as of GMM values are, however, MLP improves the EER when it is fused with GMM classifiers. Like GMM, MLP can get lower EER in the fusion condition.

The third part of Table 2 presents i-vectors result. In this step, after i-vector extraction, PCA without dimension reduction is applied to them and they are centered by their mean and length normalized. Then, i-vectors are directly used to train four different classifiers separately. The classifiers are two-class classifier and include GMM, SVM, Linear Gaussian Classifier and MLP. Finally, we used BOSARIS toolkit [29] to train logistic regression for fusion. It is obvious that the identity vector shows significantly better results respect to the neural network and GMM except in the case of fusing the GMMs trained with all the features. The last part of Table 2 shows results with different systems fusion. It shows that i-vector performs better when

it is fused whether with only GMM or with both GMM and MLP. However, fusing with MLP alone cannot improve the performance. Also, it is observed that fusion of the systems reduces the EER significantly and the lowest EER is obtained when all systems are fused. It should be noted that this score is attained by linear fusion of subsystems scores.

5.4. Evaluation Set

Table 3 shows the EER values of the evaluation set obtained by the different fusion systems. In the evaluation phase, we have used the same system as for the development data. The first part of Table 3 denotes the results of GMM classifiers. As same as the results in the development part, between the different types of features the lowest EER has been obtained by CQCC and it is about 16.21% while fusing CQCC with other features considerably reduces EER by approximately 3 percent. In the second trying, in contrast with the DEV results, an EER of 27.23% is obtained in MLP-based system using MFCC feature. Unexpectedly, using CQCC feature was not resulted in the best EER, however, the fusion of features improved the results like prior system. We speculate that the low amount of training data causes low performance of MLP. However, as shown in the Table 3, the fusion of the MLP with other systems improves EER. Furthermore, an EER of 16.69% has been measured when i-vector has been applied to the system. Although i-vector feature can help us to reach a considerable EER, the results would be more attractive in the case that i-vector have been fused with other system scores. The fusion sets are mentioned in the fourth part of Table 3.

Table 3: EER (%) for different systems in *evaluation* set.

Systems	**EER%**
Baseline	**26.65%**
GMM1 (CQCC + PCA)	**16.21**
GMM2 (MGD + PCA)	30.24
GMM3 (MFCC + PCA)	26.41
GMM4 (RASTA-PLP + PCA)	29.13
GMM5 (LPCC + PCA)	35.60
FUSION (GMM-all features)	**13.36**
MLP1 (CQCC + PCA)	37.76
MLP2 (MGD + PCA)	39.53
MLP3 (MFCC + PCA)	**27.23**
MLP4 (RASTA-PLP + PCA)	36.34
MLP5 (LPCC + PCA)	31.44
FUSION (MLP-all features)	**26.12**
i-vector	**16.69**
FUSION (GMM + i-vector)	10.88
FUSION (MLP + i-vector)	13.75
FUSION (GMM + MLP)	13.24
FUSION (GMM + MLP + i-vector)	**10.31**

Like previous results achieved in the development set, the lowest ERR for evaluation data belongs to the fusion of all the three systems (GMM + MLP + i-vector). The final score is obtained by linear fusion of sub-systems scores. The coefficients of this linear fusion are determined by tuning parameters on development data. An overview of SUT system is graphically illustrated in Fig. 2.

6. Conclusion

Replayed speech can be used as a spoofing speech attack and provides wide threats for the automatic speaker verification (ASV) systems. To study these threats and countermeasures ASVspoof2017 corpus is presented by NIST. In this work, we described an anti-replay spoof system based on ASVspoof2017. Our proposed system used various types of classifiers and features. Since the replayed speech may be distorted, these distortions cause a mismatch between the genuine and the replay speech pattern. This mismatch can be detected by training a classifier using cepstral-based features such as CQCC and MFCC or spectral based features like PLP and RASTA-PLP. To obtain final scores, each classifier computes the posterior probability of each genuine and spoof class for the given input utterance. To achieve better results, we fused the scores computed by systems. Experimental results show that the best result in terms of EER is attained by fusion of all systems (GMM + MLP + i-vector).

7. Acknowledgment

The authors of the paper would like to thank Mohammad Elmi and Hossein Zeinali for their assistance and helpful comments.

References

[1] M. Hakan, "Automatic speaker verification on site and by telephone: methods, applications and assessment," PhD diss., KTH, 2006.

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

[3] Y. Huang Q. Li, "An auditory-based feature extraction algorithm for robust speaker identification under mismatched conditions," *EEE transactions on audio, speech, and language processing*, pp. 1791–1801, 2011.

[4] X. Xiao, X. Tian, S. Du, H. Xu, S Chng, and H Li, "Spoofing speech detection using high dimensional magnitude and phase features: The ntu approach for asvspoof 2015 challenge," *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[5] Z. Wu, T. Kinnunen, N. Evans, J. Yamagishi, C. Hanilci, M. Sahidullah, and A. Sizov, "Asvspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge," *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[6] T. Kinnunen, M. Sahidullah, M. Falcone, L. Costantini, R.G. Hautamäki, D. Thomsen, A. Sarkar, Z.H. Tan, H. Delgado, M. Todisco, and N. Evans, "Reddots replayed: A new replay spoofing attack corpus for text-dependent speaker verification research," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017.

[7] Z. Wu, N. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, and H Li, "Spoofing and countermeasures for speaker verification: a survey," *Speech Communication*, vol. 66, pp. 130–153, 2015.

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

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

[10] H. Murthy, V.Gadde, and N. Evans, "The modified group delay function and its application to phoneme recognition," *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference*, vol. 1, pp. I–68, 2003.

[11] Z. Wu, P. L. De Leon, C. Demiroglu, A. Khodabakhsh, S. King, Z. Ling, and D.saito, "Anti-spoofing for text-independent speaker verification: An initial database, comparison of countermeasures, and human performance," *IEEE/ACM Transactions on Audio, Speech, and Language Processing 24*, pp. 768–783, no. 4 ,2016.

[12] T. Kinnunen, N. Evans, J. Yamagishi, K. A. Lee, M. Sahidullah, M. Todisco, and H. Delgado, "Asvspoof 2017: automatic speaker verification spoofing and countermeasures challenge evaluation plan," *Training*, vol. 10, n. 1508, 2017.

[13] K.R. Ghule and R. R. Deshmukh, "Feature extraction techniques for speech recognition: A review," *International Journal of Scientific & Engineering Research 6*, pp. 2229–5518, no. 5 ,2015.

[14] N.Desai, K. Dhameliya, and V. Desai, "Feature extraction and classification techniques for speech recognition: A review," *International Journal of Emerging Technology and Advanced Engineering 3*, pp. 367–371, no. 12 ,2013.

[15] S. M. Van, "Handling convolutional noise in missing data automatic speech recognition," *Proceedings International conference on spoken language processing*, 2006.

[16] N. Morgan, N. Bayya, A. Kohn, and P. Hermansky, "Rasta-plp speech analysis," *ICSI Technical Report TR-91-969*, 1991.

[17] Y. Liu, Y. Tian, L. He, J. Liu, and M. T . Johnson, "Simultaneous utilization of spectral magnitude and phase information to extract supervectors for speaker verification anti-spoofing," *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[18] M. R. Hedge, , H. A. Murthy, and V. R. R. Gadde, "Significance of the modified group delay feature in speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing 15*, pp. 190–202, no. 1 ,2007.

[19] S. H. K. Parthasarathi, R. Padmanabhan, and H. A. Murthy, "Robustness of group delay representations for noisy speech signals," *IEEE Transactions on Audio, Speech, and Language Processing 15*, pp. 190–202, no. 1 ,2007.

[20] B. Egnanarayana and H. A. Murthy, "Significance of group delay functions in spectrum estimation," *IEEE Transactions on signal processing40*, pp. 2281–2289, no. 9 ,1992.

[21] M. Todisco, H. Delgado, and N. Evans, "Constant q cepstral coefficients: A spoofing countermeasure for automatic speaker verification," *Computer Speech and Language 45*, pp. 516–535, 2017.

[22] B. Yegnanarayana and H.A. Murthy, "Significance of group delay functions in spectrum estimation," *IEEE Transactions on signal processing40*, pp. 2281–2289, no. 9 ,1992.

[23] K. A. Lee, A. Larcherand, G. Wang, P. Kenny, N. Brümmer, D. V. Leeuwen, H. Aronowitz, M. Kockmannand, C. Vaquero, B. Ma, and H. Li, "The reddots data collection for speaker recognition," *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[24] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing 19*, pp. 788–798, no. 4 ,2011.

[25] N. Cristianini and J. Shawe-Taylor, "An introduction to support vector machines and other kernel-based learning methods," *Cambridge university press*, 2000.

[26] C. Y. Peng, K.L. Lee, and G. M. Ingersoll, "An introduction to logistic regression analysis and reporting," *IEEE Transactions on Audio, Speech, and Language Processing 19*, pp. 3–14, no. 1 ,2002.

[27] D. Koller and N. Friedman, "Probabilistic graphical models: principles and techniques," *MIT press*, 2002.

[28] R. Jaiswal, D. Fitzgerald, E. Coyle, and S. Rickard, "Towards shifted nmf for improved monaural separation," pp. 19–19, 2013.

[29] C. Schörkhuber, A. Klapuri, and A. Sontacchi, "Audio pitch shifting using the constant-q transform," *Journal of the Audio Engineering Society 61*, no. 7/8 ,2013.