Overview of BTAS 2016 Speaker Anti-spoofing Competition

P. Korshunov,* S. Marcel, H. Muckenhirn (Idiap team),
Idiap Research Institute, Martigny, Switzerland
{pavel.korshunov,sebastien.marcel,hannah.muckenhirn}@idiap.ch

A. R. Gonçalves, A. G. Souza Mello, R. P. Velloso Violato,
F. O. Simões, M. U. Neto, M. de Assis Angeloni, J. A. Stuchi (CPqD team),
Center for Research and Development in Telecommunications, Brazil
{andrerg,amello,rviolato,simoes,uliani,massis,jastuchi}@cpqd.com.br

H. Dinkel, N. Chen, Y. Qian (SJTUSpeech team),
Shanghai Jiao Tong University, China
{heinrich.dinkel,yanminqian}@sjtu.edu.cn, bobchennan@jhu.edu

D. Paul, G. Saha (IITKGP_ABSP team),
Indian Institute of Technology, Kharagpur, India
{dipjyotipaul,gsaha}@ece.iitkgp.ernet.in

Md Sahidullah (IITKGP_ABSP team)
University of Eastern Finland, Finland
sahid@cs.uef.fi

Abstract

This paper provides an overview of the Speaker Anti-spoofing Competition organized by Biometric group at Idiap Research Institute for the IEEE International Conference on Biometrics: Theory, Applications, and Systems (BTAS 2016). The competition used AVspoof database, which contains a comprehensive set of presentation attacks, including, (i) direct replay attacks when a genuine data is played back using a laptop and two phones (Samsung Galaxy S4 and iPhone 3G), (ii) synthesized speech replayed with a laptop, and (iii) speech created with a voice conversion algorithm, also replayed with a laptop.

The paper states competition goals, describes the database and the evaluation protocol, discusses solutions for spoofing or presentation attack detection submitted by the participants, and presents the results of the evaluation.

1. Introduction

Despite the growing usage and increasing reliability of automatic speaker verification (ASV) systems, they are shown to be vulnerable to presentation or spoofing attacks. In such attack, an invalid user attempts to gain access to the system by presenting counterfeit (fake) speech sample(s) as the evidence of a valid user. Counterfeit speech can be synthesized from text, converted using speech of another person, or simply replayed using some playback device, e.g., a mobile phone.

The participants in this anti-spoofing competition proposed presentation attack detection (PAD) techniques to protect an ASV system against presentation attacks. Essentially, these techniques should effectively separate real (genuine) speech recordings from spoofed speech (attacks).

Compared to the previously conducted ASVspoof [14] anti-spoofing challenge, which considered only synthetic speech attacks that bypass the microphone (coined as 'logical access' attacks), this competition focuses on a more practical *replay attacks*, which are broadly defined as *presentation attacks* by ISO standardization committee [7]. Although presentation attacks are considered important by the industry, they received considerably less attention, since, until now, there was no dataset with such attacks.

In this competition, we used a recent publicly available AVspoof [9] database[1], which provides a comprehensive variety of presentation attacks, including attacks when a genuine data is played back to an ASV system using laptop speakers, high quality speakers, and two mobile phones. Synthetic speech attacks, such as speech synthesis and voice conversion replayed with laptop speakers, are also included.

The participants were provided with two non-overlapping sets (each containing real and spoofed

*Except for the first two organizers, authors are in no particular order.

[1] https://www.idiap.ch/dataset/avspoof

data subsets) for training and calibration of their PAD systems. The submitted systems were evaluated on a separate independent testing set, which, besides the attacks present in AVspoof database, also included additional 'unknown' attacks.

Table 1: Number of utterances in AVspoof subsets. 'SS' stands for speech synthesis, 'VC' for voice conversion, and 'RE' for replay. 'LP' indicates laptop, 'PH1' is Samsung Galaxy S4 phone, 'PH2' is iPhone 3GS, 'PH3' is iPhone 6S, and 'HQ' means high quality speakers were used during replay.

Type of data	**Train**	**Dev**	**Test**
genuine data	4973	4995	5576
all attacks	38580	38580	44920
SS-LP-LP	490	490	560
SS-LP-HQ-LP	490	490	560
VC-LP-LP	17400	17400	19500
VC-LP-HQ-LP	17400	17400	19500
RE-LP-LP	700	700	800
RE-LP-HQ-LP	700	700	800
RE-PH1-LP	700	700	800
RE-PH2-LP	700	700	800
RE-PH2-PH3	-	-	800
RE-LPPH2-PH3	-	-	800

2. Database

AVspoof, a publicly available database[1] used in the competition, contains real (genuine) speech samples from 44 participants (31 males and 13 females) recorded over the period of two months in four sessions, each scheduled several days apart in different setups and environmental conditions such as background noises. The first session was recorded in the most controlled conditions. Speech samples were recorded using three devices: laptop using microphone AT2020USB+, Samsung Galaxy S4 phone, and iPhone 3GS, with the following types of samples recorded: (1) reading part of 10 or 40 pre-defined sentences read by subjects, (2) pass-phrases part of 5 short prompts read by subjects, and (3) free speech part of a free speech about any topic for 3 to 10 minutes.

To have an unbiased evaluation, the samples of the database are split into three non-overlapping subsets: training, development, and test. Each subset consists of two main parts: (i) real or genuine data and (ii) several different presentation attacks (see Table 1).

When generating presentation attacks, the assumption is that a verification system is installed on a laptop (with an internal built-in microphone) and an attacker is trying to gain access to this system by playing back to it a pre-recorded genuine data or an automatically generated synthetic data using some playback device. In AVspoof database, presentation attacks consist of (i) direct replay attacks when a genuine data is played back using a laptop with internal speakers, a laptop with external high quality speakers, Samsung Galaxy S4 phone, and iPhone 3G, (ii) synthesized speech replayed with a laptop and HQ speakers, and (iii) converted voice attacks replayed with a laptop and HQ speakers.

New attacks in Test set To make competition more challenging, additional attacks were recorded for the test set to introduce 'unknown' types of attacks, which allow to access submissions for a scenario when not all attacks are known *a priori* (a common situation in practice).

Therefore, two types of replay attacks were recorded: (i) the data recorded with iPhone 3G was replayed to an iPhone 6S and (ii) the original data from the test set was replayed to an iPhone 6S. These types of attacks simulate two different practical scenarios, in the first case, it is assumed that the attacker obtained the required audio samples by secretly recording them with a mobile phone (iPhone 3G) and, in the second case, the attacker simply has stolen the original data. These attacks were played to iPhone 6S to simulate the situation when the verification system is installed on a mobile device.

Table 1 presents the detailed view of the database and how it was split into training, development, and test sets. The rows of the table correspond to different types of data, including different attacks. For instance, 'SS-LP-HQ-LP' means that synthesized samples were played back using laptop with high quality speakers connected to it and the target verification system is assumed to be running on a laptop as well. Similarly, attack 'RE-PH2-PH3' means that data was first recorded with iPhone 3G and then played back using the same phone to iPhone 6S, where the verification system is assumed to be running. However, in 'RE-LPPH2-PH3', the original data that was recorded with a laptop is replayed using iPhone 3G (i.e., it was stolen by the attacker) to iPhone 6S.

3. Evaluation protocol

Training and development sets were released to the participants at the start of the competition, so they had enough time to train their proposed PAD systems on the training set and tune it on the development set. Once the scores for the development set were submitted, the test set was released, with all data anonymized (randomized file names with no information on what is real data or attacks), to assess and rank the accuracy of the proposed systems. In addition to having two new attacks, test set also contained 5473 *anchor* samples (with randomized file names) from the development set. Since participants were not aware about the existence of the anchor files, it allowed us to determine whether each final submitted system was the same that was used to generate the development set scores, by checking that score values for anchor files in test and development sets match exactly.

The evaluation of the PAD systems was done based on the *false acceptance rate* (FAR) and the *false rejection rate* (FRR), which depend on a certain *threshold* θ:

$$\begin{aligned} \text{FAR}(\theta) &= \frac{|\{h_{attack} \mid h_{attack} \geq \theta\}|}{|\{h_{attack}\}|} \\ \text{FRR}(\theta) &= \frac{|\{h_{real} \mid h_{real} < \theta\}|}{|\{h_{real}\}|}, \end{aligned} \tag{1}$$

where h_{real} is a score for real or genuine data and h_{attack} is a score for the attack or spoofed data.

We use the development set to determine threshold θ_{dev}, based on the *equal error rate* (EER) of the evaluated system. The final evaluation performance is then computed as the *half total error rate* (HTER) (more details about the proposed evaluation can be found in [4]):

$$\begin{aligned} \theta_{dev} &= \arg\min_{\theta} \frac{\text{FAR}_{dev}(\theta) + \text{FRR}_{dev}(\theta)}{2} \\ \text{HTER}_{eval}(\theta_{dev}) &= \frac{\text{FAR}_{eval}(\theta_{dev}) + \text{FRR}_{eval}(\theta_{dev})}{2} \end{aligned} \tag{2}$$

The main goal for using the proposed evaluation protocol is to apply the same evaluation conditions to all participants. Such approach allows a fair and objective comparison between different submissions.

4. Baseline system

A baseline PAD system[2] was provided to participants as an example of the working system with EER to beat. The baseline system is based on the open source Bob toolbox [1]. The provided system uses simple spectrogram-based ratios as features and logistic regression as a classifier, which is a relatively simple setup that should be easy to beat.

Prior to computing features, a given audio sample is first split into overlapping 20ms-long speech frames with a 10ms overlap. The frames are pre-emphasized with 0.97 coefficient and pre-processed by applying Hamming window. A power spectrum is computed from the preprocessed signal and is filtered with 40 Mel-scale triangular filters, resulting in 40 spectral bands of the signal. These bands are split into 10 sub-bands, and one average value is computed for each sub-band in both dimensions: within 4 bands and across signal length. The ratios between these 10 average values result in 10-values feature vector. Vectors from the training set are used to train a logical regression classifier, while features from development set are used to compute development scores and determine the threshold.

The EER of the baseline system is 5.91% on the development set, so the goal of participants was to develop a system that can beat this value and also perform well on the test set with added ‘unknown’ attacks.

5. Submitted approaches

Seven different teams from around the world registered for the competition and four teams submitted their results for development and test sets.

5.1. Submission by ‘CPqD’ team

This system relies on two types of features that are extracted from each speech signal. First type is 20 *Mel-frequency cepstral coefficients* (MFCCs) [5] and their deltas that were computed on frame-by-frame basis and then averaged. Then, another set of *cepstrum* coefficients was computed using the following steps: given a windowed frame y_k of the signal, compute its Fourier transform and consider only its magnitude $Y_k = |F(y_k)|$. Compute the average of Y for the whole signal, normalize it and take the log of the result, according to the following equation:

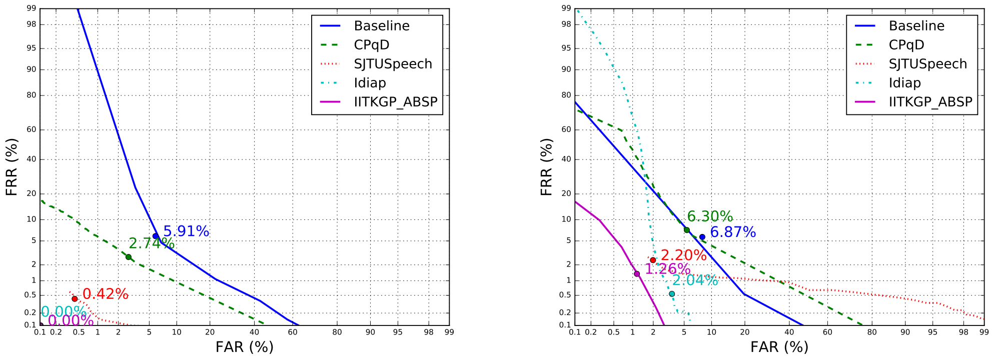
$$m_Y = \log\left(\frac{\sum_k Y_k}{\sqrt{\left(\sum_k Y_k\right)^t \left(\sum_k Y_k\right)}}\right) \tag{3}$$

The log of the magnitude of the inverse Fourier transform of m_Y lead to the *cepstrum* coefficients.

Model description As a classifier model, the system uses a neural network for binary classification. It aggregates all attacks in an “attack” class. The neural network has the following architecture: (i) two hidden layers (10 and 64 neurons each) with *relu* activation function, (ii) dropout in the first hidden layer, and (iii) data was scaled before passing to the neural network.

Auxiliary training data In addition to the provided dataset, an ASVspoof [14] dataset from the Interspeech 2015 challenge was also used to enhance the training data and, hopefully, avoid model over-fitting to the given dataset. The reasoning is that machine learning-based models tend to look at specific features of the loudspeakers and microphones used in the training audios, and have problems when audios provided from other devices or that do not contain such patterns (different attacks, for example).

[2] Source code: https://pypi.python.org/pypi/bob.paper.btas_c2016



(a) Dev set with EER values

(b) Test set with HTER values

Figure 1: DET curves for all submissions computed for development and test sets.

Table 2: Evaluation results (error rates are in percentages) for the development and test sets.

Team	**EER (Dev)**	**FAR (Test)**	**FRR (Test)**	**HTER (Test)**
Baseline	5.91	5.74	8.01	6.87
CPqD	2.74	7.23	5.38	6.30
SJTUSpeech	0.42	2.42	1.98	*2.20*[5]
Idiap	0.00	0.54	3.54	2.04
IITKGP_ABSP	0.00	1.36	1.16	1.26

5.2. Submission by 'SJTUSpeech' team

This system is closely related to the authors' previous work in speaker verification [10]. A traditional 39-dimensional CMVN was used with normalized PLP features [6] in a 10ms frame-window as the base features. These features are fed into the respective classifier and produce a fixed size representation from a given hidden layer. Two base classifiers consist of a 7 Layer DNN and a 4 Layer BLSTM. Similarly to the approach in [3], all networks use four output neurons, representing the three spoofing types (voice conversion, speech synthesis, and replay) in addition to the genuine class.

DNN The features are extended with a 15×15 context window, resulting in a 1209-dimensional feature vector, which is fed into the network. DNN contains 1024 neurons in every hidden layer. After each linear layer, in addition to dropout, batch normalization is applied. During training, 60 iterations of adadelta optimization is also run[3].

BLSTM The second classifier contains a 4 Layer Network with two sigmoid activations surrounding two layers of LSTM. The 39-dimensional PLP features are fed into the model and it is then trained for 10 iterations with adam as the optimization method[3]. As the result, the output of the last hidden layer is used as a valid 512 dimensional representation of the spoofing type.

Fusion Based on the development set, the relation between utterance length and error rate was analyzed with conclusion that the BLSTM's error distribution is shifted towards short utterances. The same analysis on the DNN revealed that it is less error-prone to short utterances. Thus a combination of the DNN and BLSTM feature vectors was done[3] to create a 1560 dimensional representation. These vectors are used as the input to our scoring method.

Scores are obtained by using a Gaussian function as a classifier by running scikit linear discriminant analysis (LDA) framework[4] implementation to model the genuine utterances as a single model with a Gaussian distribution and directly calculate log likelihoods by scoring these models against all utterances in the dataset. Thus, for each utterance, the score value is near 0 ($\log(1)$) if the model and the utterance do match with each other (i.e., genuine) and the score is lower than zero if they do not match (i.e., spoofed utterances).

[3] https://gitlab.com/Richy/BTAS2016

[4] http://scikit-learn.org/0.16/modules/generated/sklearn.lda.LDA.html

[5] For SJTUSpeech, 1% of scores from anchor files planted in test set differed from scores compared to development set. The team could not explain this discrepancy.

Table 3: Per attack results (HTER %) from all submissions for test set.

Attacks	Baseline	CPqD	SJTUSpeech	Idiap	IITKGP_ABSP
All together	6.87	6.30	*2.20*[5]	2.04	1.26
SS-LP-LP	2.87	5.89	1.88	0.27	0.68
SS-LP-HQ-LP	2.87	7.81	1.75	0.27	0.68
VC-LP-LP	3.58	4.92	1.73	0.33	0.74
VC-LP-HQ-LP	3.39	6.16	1.81	0.27	0.81
RE-LP-LP	17.02	16.53	10.34	15.83	8.58
RE-LP-HQ-LP	11.24	9.80	10.02	0.58	1.81
RE-PH1-LP	52.24	9.30	1.52	0.33	0.68
RE-PH2-LP	51.96	23.46	2.05	25.18	3.59
RE-PH2-PH3	51.56	36.43	2.84	50.08	6.49
RE-LPPH2-PH3	20.62	31.30	18.09	46.64	23.06

5.3. Submission by 'Idiap' team

Please note that this team was treated in the same way as other participants and was not aware about the details of the competition and differences in data and protocol. The submitted system is based on long-term spectral mean and standard deviation. Each utterance is split into frames of 32ms with a shift of 10ms. Each frame is first pre-emphasized to enhance the high frequencies and a discrete Fourier transform (DFT) is taken. Then, the mean $\mu[k]$ and the standard deviation $\sigma[k]$ are computed over all the frames of the log magnitude of the DFT coefficients:

$$\begin{aligned} \mu[k] &= \frac{1}{M}\sum_{m=1}^{M} \log|X_m[k]|, \\ \sigma^2[k] &= \frac{1}{M}\sum_{m=1}^{M} \left(\log|X_m[k]| - \mu[k]\right)^2, \end{aligned} \tag{4}$$

where $X_m[k]$ is the k^{th} coefficient of the DFT of the m^{th} frame, $k = 0 \ldots \frac{N}{2} - 1$, $m = 1 \ldots M$, with N the length of a frame, and M is the number of frames extracted from the utterance.

The resulting feature vector is the concatenation of the mean vector with the standard deviation vector. Given that the frames are computed over 32ms and the sampling frequency is 16kHz, a feature vector of 512 components per utterance is obtained.

The feature vectors are classified using a classifier based on LDA. The input features are projected onto one dimension with an LDA and the obtained values are directly used as scores. The performance achieved with this simple linear classifier shows that long-term spectral statistics are highly discriminative features.

The motivation for using these features is twofold. First, a long-term spectral mean (or a long-term average spectrum) is widely used as a measure of voice quality [8], which is meaningful in presentation attacks detection, since it captures information about channel degradation and speech "naturalness". Although cepstral mean and standard deviation (directly related to the long-term spectral mean and standard deviation) are traditionally removed from the signal in speech and speaker recognition systems, they are more robust to channel variability. Hence, this information is useful to detect presentation attacks as they are played back to the system, leading to channel degradation.

5.4. Submission by 'IITKGP_ABSP' team

This anti-spoofing system is based on the score-level fusion of two sub-systems. It uses two different spectral features: (MFCCs) [5] and *inverted* MFCCs (IMFCCs) [2], respectively. The feature extraction of both the sub-systems is followed by *Gaussian mixture model-maximum likelihood* (GMM-ML) [12] classifier with the log-likelihood ratio as the scores. Feature vectors are computed with a frame size of 20ms and overlap of 50%. Speech activity detector (SAD) is not employed as non-speech frames could be helpful for spoofing detection [13]. The feature extractors are optimized on the development data. Both features are extracted using 20 filters in the filter bank. Energy coefficient is retained to formulate 20-dimensional feature vector with only *static coefficients*. There are 10 iterations of *expectation-maximization* (EM) algorithm to estimate parameters of two GMMs (real/genuine and spoofed) with ML criteria [11]. The number of mixture components is set at 512. The log-likelihood score is calculated as,

$$\Lambda(\mathbf{X}) = \mathcal{L}(\mathbf{X}|\boldsymbol{\lambda}_{\text{real}}) - \mathcal{L}(\mathbf{X}|\boldsymbol{\lambda}_{\text{attack}}), \tag{5}$$

where $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$ is the feature matrix of the test utterance, T is the number of frames and $\mathcal{L}(\mathbf{X}|\boldsymbol{\lambda}) = (1/T)\sum_{t=1}^{T} \log p(\mathbf{x}_t|\boldsymbol{\lambda})$ is the average log-likelihood of $\mathbf{X}$ given GMM model $\boldsymbol{\lambda}$. $\boldsymbol{\lambda}_{\text{real}}$ and $\boldsymbol{\lambda}_{\text{attack}}$ correspond to the

GMM models that are generated from genuine and replay spoofed speech samples respectively. The final score is computed as the *linear fusion* of scores for MFCCs and IMFCCs, and it is expressed as,

$$\Lambda_{\text{fused}}(\mathbf{X}) = (1-\alpha)\Lambda_{\text{mfcc}}(\mathbf{X}) + \alpha\Lambda_{\text{imfcc}}(\mathbf{X}) \qquad (6)$$

where, $\Lambda_{\text{mfcc}}(\mathbf{X})$ and $\Lambda_{\text{imfcc}}(\mathbf{X})$ are two log-likelihood ratio scores for two features, respectively, and $\alpha = 0.5$.

6. Evaluation of submissions

Prior to computing error rates, the test scores were checked for consistency with the development scores using the anchor files present in both sets (see Section 3 for details). All participants, except for *SJTUSpeech* team, had exact matching scores for all 5473 anchor files. For SJTUSpeech, 56 anchor files (about 1%) had mismatched scores in test and development sets, but since the mismatched values would not affect the resulted error rates, the team's results were accepted in the competition.

The evaluation results[2] for the submitted systems are presented in Table 2, where EER values characterize performances on the development set and FAR, FRR, and HTER values summarize the performances on the test set. Figure 1 shows DET curves, presenting a more detailed overview of the submissions performances on both sets.

To demonstrate how different types of attacks affect different systems, we show HTER value for each separate attack from the test set in Table 3, which can be related to the number of samples in each attack given in Table 1. Please note that the last two rows of the tables correspond to new attacks (see Section 2) that are added in the test set but were not available in training or development sets.

From Table 3, it can be noted that all submitted systems demonstrate difficulty in detecting new attacks added in the test set. Also, since database contained disproportionally more samples generated with voice conversion, the overall HTER results were lower for those systems that were more successful in detecting this specific type of attack. Therefore, a more unbiased way to compare systems would be to look at the performance for each different attack.

7. Conclusion

The evaluation results of the competition demonstrated that presentation attacks still pose a serious challenge to some of the most advanced presentation attack detection systems. The attacks for which the systems were not specifically trained were particularly challenging. These 'unknown' attacks are often expected in a practical scenario. Therefore, new PAD methods need to be developed, which can capture general spoofing information to generalize well for different types of attacks. New datasets with larger variety of practical setups and capturing conditions (noise, microphones, speakers, etc.) are also necessary.

Acknowledgments

This work was conducted in the framework of the EU funded BEAT project (#284989), Norwegian SWAN project, and was partially funded by Swiss Centre for Biometrics Research and Testing.

References

[1] A. Anjos, L. E. Shafey, R. Wallace, M. Günther, C. McCool, and S. Marcel. Bob: a free signal processing and machine learning toolbox for researchers. In *ACM international conference on Multimedia*, pages 1449–1452, Oct. 2012. 3

[2] S. Chakroborty, A. Roy, and G. Saha. Improved closed set text-independent speaker identification by combining MFCC with evidence from flipped filter banks. *International Journal of Signal Processing*, 4(2):114–122, 2007. 5

[3] N. Chen, Y. Qian, H. Dinkel, B. Chen, and K. Yu. Robust deep feature for spoofing detection – the SJTU system for ASVspoof 2015 challenge. In *Interspeech*, pages 2097–2101, Dresden, Germany, Sept. 2015. 4

[4] I. Chingovska, A. Anjos, and S. Marcel. Biometrics evaluation under spoofing attacks. *IEEE Transactions on Information Forensics and Security*, 9(12):2264–2276, Dec. 2014. 3

[5] S. B. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 28(4):357–366, 1980. 3, 5

[6] H. Hermansky. Perceptual linear predictive (PLP) analysis of speech. *The Journal of the Acoustical Society of America*, 87(4):1738–1752, 1990. 4

[7] ISO/IEC JTC 1/SC 37 Biometrics. DIS 30107-1, information technology – biometrics presentation attack detection. American National Standards Institute, Jan. 2016. 1

[8] P. Kitzing. LTAS criteria pertinent to the measurement of voice quality. *Journal of phonetics*, 14:477–482, 1986. 5

[9] S. Kucur Ergunay, E. Khoury, A. Lazaridis, and S. Marcel. On the vulnerability of speaker verification to realistic voice spoofing. In *IEEE International Conference on Biometrics: Theory, Applications and Systems*, pages 1–6, Sept. 2015. 1

[10] Y. Liu, Y. Qian, N. Chen, T. Fu, Y. Zhang, and K. Yu. Deep feature for text-dependent speaker verification. *Speech Communication*, 73:1–13, 2015. 4

[11] D. Paul, M. Pal, and G. Saha. Novel speech features for improved detection of spoofing attacks. In *Annual IEEE India Conference (INDICON)*, pages 1–6. IEEE, 2015. 5

[12] D. Reynolds and R. C. Rose. Robust text-independent speaker identification using Gaussian mixture speaker models. *IEEE Transactions on Speech and Audio Processing*, 3(1):72–83, 1995. 5

[13] M. Sahidullah, T. Kinnunen, and C. Hanilçi. A comparison of features for synthetic speech detection. In *Interspeech*, pages 2087–2091, Dresden, Germany, Sept. 2015. 5

[14] Z. Wu, T. Kinnunen, N. Evans, J. Yamagishi, C. Hanilçi, M. Sahidullah, and A. Sizov. ASVspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge. In *Interspeech*, pages 2037–2041, Dresden, Germany, Sept. 2015. 1, 3