Speech Communication 85 (2016) 83–97

Contents lists available at ScienceDirect

Speech Communication

journal homepage: www.elsevier.com/locate/specom

Spoofing detection goes noisy: An analysis of synthetic speech detection in the presence of additive noise

Cemal Hanilçi[a,b,*], Tomi Kinnunen[a], Md Sahidullah[a], Aleksandr Sizov[a]

[a] School Of Computing, University of Eastern Finland, Joensuu, Finland
[b] Department of Electrical-Electronic Engineering, Bursa Technical University, Bursa, Turkey

ARTICLE INFO

Article history:
Available online 10 October 2016

Keywords:
Speaker recognition
Anti spoofing
Countermeasures
Additive noise

ABSTRACT

Automatic speaker verification (ASV) technology is recently finding its way to end-user applications for secure access to personal data, smart services or physical facilities. Similar to other biometric technologies, speaker verification is vulnerable to spoofing attacks where an attacker masquerades as a particular target speaker via impersonation, replay, text-to-speech (TTS) or voice conversion (VC) techniques to gain illegitimate access to the system. We focus on TTS and VC that represent the most flexible, high-end spoofing attacks. Most of the prior studies on synthesized or converted speech detection report their findings using high-quality clean recordings. Meanwhile, the performance of spoofing detectors in the presence of additive noise, an important consideration in practical ASV implementations, remains largely unknown. To this end, our study provides a comparative analysis of existing state-of-the-art, off-the-shelf synthetic speech detectors under additive noise contamination with a special focus on front-end processing that has been found critical. Our comparison includes eight acoustic feature sets, five related to spectral magnitude and three to spectral phase information. All the methods contain a number of internal control parameters. Except for feature post-processing steps (deltas and cepstral mean normalization) that we optimized for each method, we fix the internal control parameters to their default values based on literature, and compare all the variants using the exact same dimensionality and back-end system. In addition to the eight feature sets, we consider two alternative classifier back-ends: Gaussian mixture model (GMM) and i-vector, the latter with both cosine scoring and probabilistic linear discriminant analysis (PLDA) scoring. Our extensive analysis on the recent ASVspoof 2015 challenge provides new insights to the robustness of the spoofing detectors. Firstly, unlike in most other speech processing tasks, all the compared spoofing detectors break down even at relatively high signal-to-noise ratios (SNRs) and fail to generalize to noisy conditions even if performing excellently on clean data. This indicates both difficulty of the task, as well as potential to over-fit the methods easily. Secondly, speech enhancement preprocessing is not found helpful. Thirdly, GMM back-end generally outperforms the more involved i-vector back-end. Fourthly, concerning the compared features, the Mel-frequency cepstral coefficient (MFCC) and subband spectral centroid magnitude coefficient (SCMC) features perform the best on average though the winner method depends on SNR and noise type. Finally, a study with two score fusion strategies shows that combining different feature based systems improves recognition accuracy for known and unknown attacks in both clean and noisy conditions. In particular, simple score averaging fusion, as opposed to weighted fusion with logistic loss weight optimization, was found to work better, on average. For clean speech, it provides 88% and 28% relative improvements over the best standalone features for known and unknown spoofing techniques, respectively. If we consider the best score fusion of just two features, then RPS serves as a complementary agent to one of the magnitude features. To sum up, our study reveals a significant gap between the performance of state-of-the-art spoofing detectors between clean and noisy conditions.

1. Introduction

Automatic speaker verification (ASV) (Reynolds and Rose, 1995) is the task of authenticating users based on their voices. Traditionally, ASV has mostly been applied in specialized surveillance and

* Corresponding author.
E-mail addresses: cemal.hanilci@btu.edu.tr (C. Hanilçi), tkinnu@uef.fi (T. Kinnunen), sahid@uef.fi (M. Sahidullah), sizov@uef.fi (A. Sizov).

http://dx.doi.org/10.1016/j.specom.2016.10.002
0167-6393/

forensics applications but recent methodological advances have greatly increased interest in mass-market adoption to secure personal data. For instance, in 2013 a smartphone voice unlock feature was introduced to a Baidu-Lenovo phone,[1] and similar activities are being pursued by Google to their Android devices.[2] Some of the favorable points of ASV over other popular biometric identifiers are wide applicability (no other sensors except microphone required), natural integration with face authentication in smartphones, as well as *revocability*: if a voice token is compromised or stolen, another user pass-phrase can be selected.

A speech-based authentication system to control access to personal data or physical site will be useful only if it helps to improve the overall system security. A now well-recognized security concern with any biometric modality — including fingerprints, face, and speech — is that they are vulnerable to circumvention by *spoofing attacks* (Jain et al., 2006), whereby an attacker attempts to gain unauthorized access to the system by masquerading herself as another user. Attacks can naturally be executed at any parts of the system (Ratha et al., 2001), including software, biometric templates or features. However, *direct* attacks, involving an injection of forged biometric data to the sensor or the transmission point, are arguably most accessible to even less technology-aware attackers. Consequently, direct spoofing attacks are under active research across all the major biometric modalities. Specific to ASV, four currently known types of direct attacks have been identified (Evans et al., 2014; Wu et al., 2015a): (i) *replay* (Ergünay et al., 2015; Galka et al., 2015; Villalba and Lleida, 2010), representation of a prerecorded target speaker utterance; (ii) *impersonation* (Farrús et al., 2008; Hautamäki et al., 2013), human-based mimicry of a target voice; (iii) *text-to-speech synthesis* (TTS), artificially generated target voice from an arbitrary text input (Leon et al., 2010a); and (iv) *voice conversion* (VC), modification of source speech towards target speaker characteristics (Jin et al., 2008).

In this study, we focus on VC and TTS as they are arguably more flexible and consistent for spoofing both text-independent and -dependent ASV systems (Wu et al., 2015a). The effectiveness of VC and TTS spoofing attacks were first demonstrated nearly two decades ago in Pellom and Hansen (1999) and Masuko et al. (1999). Further recent studies (Alegre et al., 2012; Bonastre et al., 2007; Kons and Aronowitz, 2013; Leon et al., 2010b; Matrouf et al., 2006; Wu et al., 2015b; Wu and Li, 2014) affirm that even state-of-the-art ASV systems remain highly vulnerable to modern VC and TTS attacks. State-of-the-art VC and TTS can produce high-quality target speech using a relatively small amount of training data (Toda et al., 2006; Yamagishi et al., 2009). Even if implementing such attacks in practice would currently require a dedicated effort or special skill-set from the attacker, anytime in near future one should expect advanced voice transformation tools to be readily available for end-users in smartphones or other portable devices, thereby greatly increasing the threats imposed by advanced VC and TTS spoofing attacks.

Having recognized the vulnerability problem caused by spoofing attacks, a few first steps to develop various *countermeasures* (CMs) have been taken (Wu et al., 2015a). The most common approach (for an exception, see Sizov et al. (2015)) is to equip an off-the-shelf ASV system with a stand-alone spoofing attack detector module. In our case, a classifier that will assign a *human* or *synthetic* label (or a likelihood score) to a given utterance.[3]

The novel contribution of this work, which is placed into a wider ASV context in Section 2, is briefly stated as follows. We provide a detailed analysis on synthetic speech detection under acoustically degraded conditions, namely, additive noise, whose effects to spoofing detection are so far poorly understood. We do *not* introduce new methods but analyze the state-of-the-art methods with respect to their potential robustness bottlenecks under as comparable parameter settings and evaluation data as possible. In specific, we adopt the now widely-adopted ASVspoof 2015 challenge data (Wu et al., 2015c) to our experiments, so as to assess the joint effect of varied attacks *and* additive noise. By focusing on the key part of synthetic spoofing detectors, the feature extractor, our aim is to gain improved understanding on generalization capability of the feature extractors in this task. Our study, being the most comprehensive comparative analysis on the topic to date, is targeted especially for practitioners, such as ASV vendors, and researchers new to ASV spoofing research. The material throughout the manuscript is intended to be tutorial-like and as self-contained as possible.

2. Related work, motivation and contributions

2.1. Methods for detecting synthetic speech

Synthetic speech detection is enabled by imperfections of the VC or TTS systems. For instance, voice coders (vocoders) used for speech parameterization in VC and TTS systems use greatly simplified models of human voice production, such as all-pole synthesis filters driven by impulse train excitation (SPTK, 2014). Processing artifacts affect the spectral, temporal and prosody characteristics of synthetic speech. Similar to ASV, synthetic speech detectors consist of front-end (feature extraction) and back-end (classifier) components. Most of the work on synthetic speech detection focus on the former, including specific/tailored features combined with a simple Gaussian mixture model (GMM) or support vector machine (SVM) back-end. A substantially different approach, using standard MFCC features but focusing on i-vectors and advanced back-end modeling ideas, was carried out in Sizov et al. (2015) with promising results on the voice-converted version of NIST 2006 SRE data (though not performing well on ASVspoof 2015 (Hanilçi et al., 2015)).

In Wu et al. (2012b), standard Mel-frequency cepstral coefficients (MFCCs), cosine phase and modified group delay features were compared for the detection of Gaussian mixture model (GMM) and unit selection based synthetic speech, cosine phase features leading to the lowest error rates. In Wu et al. (2013), MFCCs, modified group delay, phase, and amplitude modulation features were compared for detecting synthetic speech, the group delay features yielding the highest accuracy. One of the most popular feature sets used for synthetic speech detection are the so-called *relative phase shift* (RPS) features (Leon et al., 2011; 2012; Sánchez et al., 2015). They are calculated based on the phase shift of the harmonic components of the signal with respect to fundamental frequency (F0), and have been reported to be effective in detecting synthetic speech (Leon et al., 2012; Sánchez et al., 2015). However, for instance Leon et al. (2012) suggests that RPS-based synthetic speech detection might be sensitive to vocoder mismatch across training and test sets, leading to degraded performance. More recently in Sánchez et al. (2015), the RPS features were used to detect synthetic speech signals provided by Blizzard Challenge. The authors found out that RPS features outperformed MFCCs on detecting speech generated by statistical parametric speech synthesis whereas MFCCs yielded higher accuracy when synthetic signals were generated by unit selection, diphone or hybrid methods. Similar, inconsistent observations were found in our recent study (Sahidullah et al., 2015) where RPS features performed the best

[1] http://www.signalprocessingsociety.org/technical-committees/list/sl-tc/spl-nl/2013-02/SpeakerVerificationMakesItsDebutinSmartphone

[2] http://thehackernews.com/2015/04/android-trusted-voice.html

[3] For brevity, we use “synthetic speech detection” to refer to detection of both VC and TTS attacks. In the present context, such umbrella term is justified as TTS and VC systems often employ similar methods for voice coding

out of 17 compared feature extraction techniques when vocoders between training and test were matched, but yielded the highest error rates in the opposite case.

In Wang et al. (2015), another robust phase-related feature similar to RPS, termed *relative phase information* (RPI) (Nakagawa et al., 2012), was used for synthetic speech detection using ASVspoof 2015 database. It was found to outperform both MFCC and MGD features. RPI processing aims at normalizing the phase changes resulting from frame positioning. In specific, with the aid of discrete Fourier transform (DFT), phase information is estimated relative to a *fixed* base frequency ($\omega_b = 2\pi \times 1000$ Hz was used in Wang et al. (2015) and Nakagawa et al. (2012)) in contrast to the RPS representation that is based on sinusoidal modeling with phase shifts computed relative to estimated F0.

2.2. Towards varied spoofing attacks: SAS corpus and ASVspoof 2015 challenge

As the above review indicates, a large number of potentially useful methods to detect synthetic speech have been investigated. The *user's dilemma*, however, is that their relative performances are either incomparable or under-representative of real-world deployment, for many reasons. Firstly, no single study compares the various methods on a common set of data or using a unified objective evaluation metric, making unbiased performance assessment challenging, if not impossible. Secondly, the studies usually contain only a handful of attacks, making conclusions attack-dependent. Thirdly, most studies involve a closed-world evaluation setting where the synthetic test samples originate from the same methods, channels and environments as used in training. This corresponds to a scenario where the ASV system administrator (defender) knows in advance what spoofing technique the attacker will employ. While such an oracle evaluation scenario may provide experimental bounds to the highest performance achievable using a specific attack detector, it is unlikely to be representative of an actual attack scenario where the attacker may employ novel (presently unknown) attacks. Fourthly, differently from the traditional NIST speaker verification scenarios involving channel- and condition-mismatched data, most of the datasets used for synthetic speech detection have consisted of high-quality (wideband) noise-free signals. As a result, it is largely unknown how well the state-of-the-art synthetic speech detectors generalize to non-ideal conditions involving not only varied spoofing materials but extrinsic distortions induced by the environment or channel, important factors in any real-world deployment of ASV technology.

To address the first three concerns — incomparability of results, limited set of attacks and closed-world evaluation bias — a new speaker verification spoofing and anti-spoofing (SAS) corpus was introduced recently in Wu et al. (2015b) and used in *ASVspoof 2015: Automatic Speaker Verification Spoofing and Countermeasures Challenge* (Wu et al., 2015c),[4] that focused on stand-alone synthetic speech detection involving both known and unknown attacks. The findings of ASVspoof 2015 were disseminated at a special session of the latest edition of *Interspeech* conference in Dresden, Germany[5].

During the special session, several participating sites reported independently that spectral phase-based features (such as cosine phase (Wu et al., 2012b), modified group delay (Wu et al., 2012b) and RPS (Sánchez et al., 2015)) outperformed spectral magnitude-based features in synthetic speech detection (Novoselov et al., 2015; Villalba et al., 2015; Wang et al., 2015; Xiao et al., 2015). GMM-based system (Reynolds and Rose, 1995) was used for modeling both natural and synthetic speech classes in most of the studies presented at the special session (Sanchez et al., 2015; Villalba et al., 2015; Wang et al., 2015). Though in Villalba et al. (2015), more advanced support vector machines (SVM) and deep neural networks (DNN) are utilized as their back-ends, the performance of GMM systems was found to be similar or better. Similar observation was made in our preliminary study on ASVspoof 2015 data (Hanilçi et al., 2015). An i-vector with Gaussian back-end and DNN based approach was also investigated in Zhang et al. (2016) without improvement in performance compared to GMM. In most recent studies using ASVspoof 2015 data, fundamental frequency (F0) contour and strength of excitation (SoE) were also used in combination with MFCCs and *cochlear filter cepstral coefficients and instantaneous frequency* (CFCCIF) features (Patel and Patil, 2016). In Todisco et al. (2016), *constant Q cepstral coefficient* (CQCC) was proposed for synthetic speech detection.

2.3. Contribution of the present study: joint effect of varied attacks and noise

In our two preliminary studies on ASVspoof 2015 data, we did extensive comparative evaluation of several front-end (Sahidullah et al., 2015) and back-end (Hanilçi et al., 2015) synthetic speech detectors. In our experiments, the simplest ideas tended to outperform more elaborate ones. For instance, raw power spectrum features and maximum likelihood (ML) trained Gaussian mixture models (GMMs) did a decent job both in detecting both unknown and known attacks, while i-vector (Dehak et al., 2011) based spoofing detection (Khoury et al., 2014; Sizov et al., 2015) yielded much higher error rates.

The present study extends Sahidullah et al. (2015) and Hanilçi et al. (2015) towards an extended and self-contained comparative evaluation of synthetic speech detectors. Unlike Sahidullah et al. (2015) and Hanilçi et al. (2015), where we used the original high-quality ASVspoof 2015 samples, in this study, we address the fourth concern missing from most of the prior studies: robustness of synthetic speech detection under acoustically degraded conditions. In general, an acoustic signal reaching a recognizer can be subjected to many extrinsic imperfections, induced by additive noise, transmission channel (including compression artifacts and low bandwidth), and reverberation, to name a few. A limited number of earlier studies have executed spoofing experiments on 8 kHz telephony data (Khoury et al., 2014; Wu et al., 2012a), though under somewhat artificial scenario in which an existing telephone-quality corpus has been post-processed through voice conversion attacks, as opposed to the more likely case of spoofing attacks taking place *before* signal transmission. We argue that it is difficult, if not impossible, to isolate the relative impact of spoofing artifacts and extrinsic distortions without an access to the original, undistorted signal. Therefore, there is a clear need to examine spoofing attacks under *controlled* extrinsic distortions to gain improved insight as to what might be the important considerations in developing practical countermeasures. A recent study (Wester et al., 2015) addressed the impact of bandwidth to synthetic speech detection accuracy on the same ASVspoof 2015 corpus as used in the present study.

In contrast to the above prior studies, we focus solely on arguably one of the most common and relevant sources of distortions, additive noise. It has received almost no prior attention to the best of our knowledge.[6] Specifically, using the ASVspoof 2015 corpus, we provide a detailed performance assessment of several

[4] www.spoofingchallenge.org

[5] http://www.signalprocessingsociety.org/technical-committees/list/sl-tc/spl-nl/2015-11/2015-11-ASVspoof/

[6] An independent study, made publicly available almost in parallel to ours Tian et al. (2016), considers the same ASVspoof2015 database under additive noise contamination. Their noise contamination design is similar to ours though spoofing

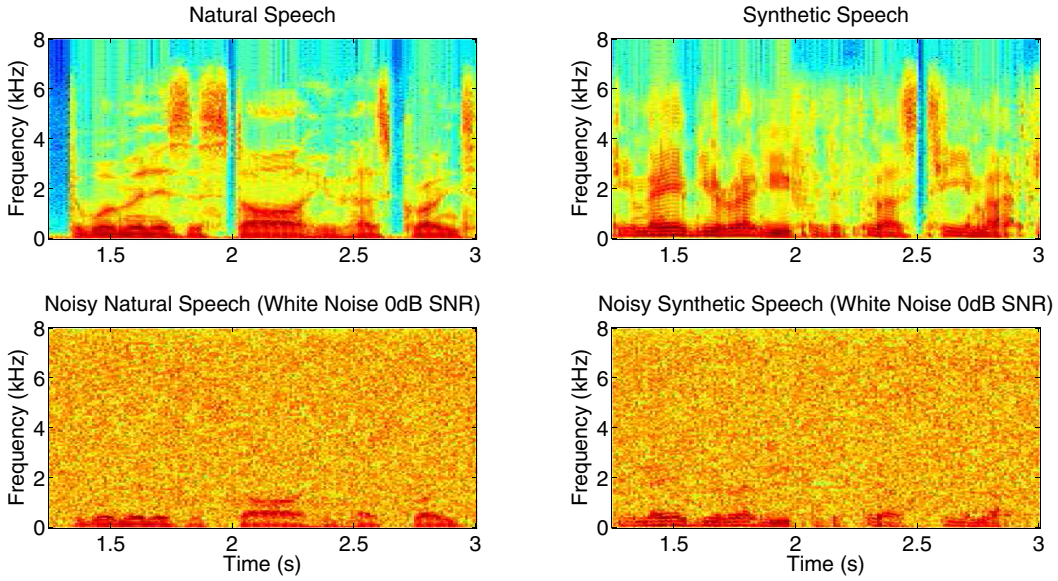

Fig. 1. Natural and synthetic speech signals of the same speaker and their noisy counterparts.

spoofing detectors under additive noise contamination. Special attention is paid in making the compared methods as comparable as possible with respect to feature dimensionalities, frame rate and other control parameters.

We expect this to be a notoriously difficult task that could serve as a useful evaluation test-bench for developing new robust countermeasures more relevant for end-user applications. As state-of-the-art TTS and VC methods can produce high-quality speech, sometimes close to or indistinguishable from authentic human speech (*unit selection* Sündermann et al. (2006) is a good example), we expect additive noise to mask further the already small differences between human and synthetic speech. As a motivation, Fig. 1 displays spectrograms of natural and synthetic speech signals of the same speaker and their noisy counterparts. While differences of natural and synthetic speech are apparent for the clean data, additive noise makes it difficult to tell the difference.

It is not obvious, for instance, whether standard speech enhancement techniques as a pre-processing method will be helpful: as noise suppression is always traded-off with speech distortion (Benesty et al., 2008), processing artifacts due to speech enhancement could be confused with artifacts due to synthesis vocoders. Similarly, as indicated above, the popular RPS (Leon et al., 2012; Sánchez et al., 2015) feature requires fundamental frequency tracking whose performance is affected by additive noise (Rabiner et al., 1976). For these reasons, it is not obvious what type of front-end or back-end modeling ideas will work comparatively better for synthetic speech detection under noisy conditions. To answer these questions, we have selected state-of-the-art or otherwise popular feature extraction methods based on both our preliminary results (Sahidullah et al., 2015) and those of the ASVspoof 2015 participants. Our eight feature sets, detailed below, include both magnitude- and phrase-related features. From the classifier side, we use GMMs trained via maximum likelihood (ML), reported as the best-performing one in Hanilçi et al. (2015), as well as the i-vector approach (Khoury et al., 2014; Sizov et al., 2015).

detection features are mostly different, and our manuscript provides a more thorough analysis.

3. Spoofing detection

Given a speech signal s, synthetic speech detection task is to decide whether s belongs to a natural speech class – hypothesis $\mathcal{H}_0$, or a synthetic speech class – hypothesis $\mathcal{H}_1$. The decision is based upon the log-likelihood ratio score, Λ:

$$\Lambda(s) = \log \mathrm{p}(\mathrm{s}|\mathcal{H}_0) - \log \mathrm{p}(\mathrm{s}|\mathcal{H}_1). \tag{1}$$

To estimate the probabilities $\mathrm{p}(\mathrm{s}|\mathcal{H}_0)$ and $\mathrm{p}(\mathrm{s}|\mathcal{H}_1)$ we need to train an acoustic model for each hypothesis. In our recent anti-spoofing study on ASVspoof 2015 (Hanilçi et al., 2015), we evaluated a number of different classification techniques. Gaussian mixture models (GMM), trained with maximum likelihood (ML) principle, was found the best choice.

GMM is a well-known probabilistic model that is extensively used for speaker recognition ever since it was introduced for the task (Reynolds and Rose, 1995). We separately use natural and synthetic training data to train two GMMs. Each GMM consists of a mixture weight w_i, a mean vector $\boldsymbol{\mu}_i$ and a covariance matrix $\boldsymbol{\Sigma}_i$ for each mixture component i. We use expectation-maximization (EM) algorithm to estimate the model parameters $\lambda = \{w_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=1}^{M}$, where M is the number of mixture components.

After the two acoustical models are trained, the log-likelihood for each hypothesis and a sequence of feature vectors $\mathbf{X} = \{\mathbf{x}_1, \ldots \mathbf{x}_T\}$, that represent the speech signal s, takes the following form

$$\log \mathrm{p}(\mathrm{s}|\mathcal{H}_k) = \frac{1}{T} \log \mathrm{p}(\{\mathbf{x}_1, \ldots \mathbf{x}_T\}|\lambda_k) = \frac{1}{T} \sum_{t=1}^{T} \log (\mathbf{x}_t|\lambda_k).$$

Besides GMM, we also consider the i-vector paradigm (Dehak et al., 2011), that became state-of-the-art technique for text-independent speaker verification. Recently, it was also used to perform speaker verification and anti-spoofing jointly in the i-vector space (Sizov et al., 2015). In essence, i-vector $\mathbf{w}$ is a fixed-sized low-dimensional vector per utterance that contains both speaker- and channel-specific variability. To extract an i-vector, we factorize a GMM mean supervector $\boldsymbol{\mu}$ as $\boldsymbol{\mu} = \mathbf{m} + \mathbf{Tw}$, where $\mathbf{T}$ is a low-rank rectangular matrix, $\mathbf{m}$ is a speaker-independent mean vector and $\mathbf{w}$ has a standard normal prior distribution. Refer to Dehak et al. (2011) for more details.

We use two different i-vector based classifiers to compute the final score (1): *cosine similarity measure* and *probabilistic linear discriminant analysis* (PLDA) (Prince and Elder, 2007). Given two i-vectors, extracted from target ($\mathbf{w}_{\text{tgt}}$) and test ($\mathbf{w}_{\text{tst}}$) utterances, we compute cosine similarity between them using

$$\text{cosine}(\mathbf{w}_{\text{tgt}}, \mathbf{w}_{\text{tst}}) = \frac{\mathbf{w}_{\text{tgt}}^{\text{T}} \mathbf{w}_{\text{tst}}}{\|\mathbf{w}_{\text{tgt}}\| \|\mathbf{w}_{\text{tst}}\|}. \quad (2)$$

As the cosine similarity measure does not compute likelihoods, instead of Eq. (1) we form the detection score as follows:

$$\text{score} = \text{cosine}(\hat{\mathbf{w}}_{\text{nat}}, \mathbf{w}_{\text{tst}}) - \text{cosine}(\hat{\mathbf{w}}_{\text{synth}}, \mathbf{w}_{\text{tst}}), \quad (3)$$

where $\hat{\mathbf{w}}_{\text{nat}}$ and $\hat{\mathbf{w}}_{\text{synth}}$ represent the average training i-vectors for natural and synthetic speech classes, respectively.

Besides cosine scoring, we also consider the so-called simplified PLDA (Kenny, 2010). The idea behind PLDA is to split total i-vector variability into speaker and channel components, which allows efficient inference during a test stage. To train the model, we grouped together i-vectors from each synthesis method and from a natural speech which gave us 6 classes ("speakers"). For more details on the data, refer to Section 6.1.

4. Natural vs. synthetic/converted speech

Before proceeding to recognition experiments, we first wish to understand the acoustic signal properties of the natural and synthetic speech signals. To analyze the characteristics of natural and synthetic speech, long-term average spectra (LTAS) is utilized. LTAS somewhat represents the physical characteristics of the speaker related the vocal tract resonances (Linville and Rens, 2001) and is mostly used in audio forensics (Grigoras, 2010) and for measuring the audibility of speech to compute speech intelligibility index (Byrne et al., 1994). LTAS is computed by time averaging the short-term Fourier magnitude spectra of all frames:

$$\text{LTAS}(k) = \frac{1}{T} \sum_{t=1}^{T} |S_t(k)|^2, \quad (4)$$

where $S_t(k)$ denotes the windowed discrete Fourier transform of tth speech frame of the signal, s, at DFT bin k and T is the total number of speech frames after voice activity detection (VAD). We compute the average LTAS of human and synthetic speech signals using the training portion of the ASVspoof 2015 dataset for each synthesis/conversion technique (S1-S5) to visualize their differences in frequency domain. Fig. 2 displays the LTAS computed using synthetic and natural speech signals (average LTAS is computed using 2525 speech files per method). Synthetic speech power is attenuated below 4 kHz compared to natural speech. For $f > 4$ kHz, the opposite happens and the difference between human and synthetic speech signals are larger. Especially for S3 and S4, hidden Markov model (HMM)-based speech synthesis techniques, the relative difference between human and synthetic speech are higher than for the other synthesis/conversion techniques. Interestingly, when $f > 7$ kHz, larger differences occur between other conversion techniques and natural speech.

It is well known that additive noise drastically reduces the speaker, language and speech recognition performances. Several methods to cope with the adverse effects of additive noise contamination have been proposed. Speech enhancement techniques aim to improve the quality of the signal corrupted by noise in the signal level. Cepstral mean subtraction (CMS) (Atal, 1974), cepstral mean and variance normalization (CMVN) and RASTA filtering (Hermansky and Morgan, 1994) are the popular feature level methods to suppress linear channel bias in cepstral features, often yielding increased speaker recognition accuracy. Speaker, language and speech recognition under additive noise and mismatched channel

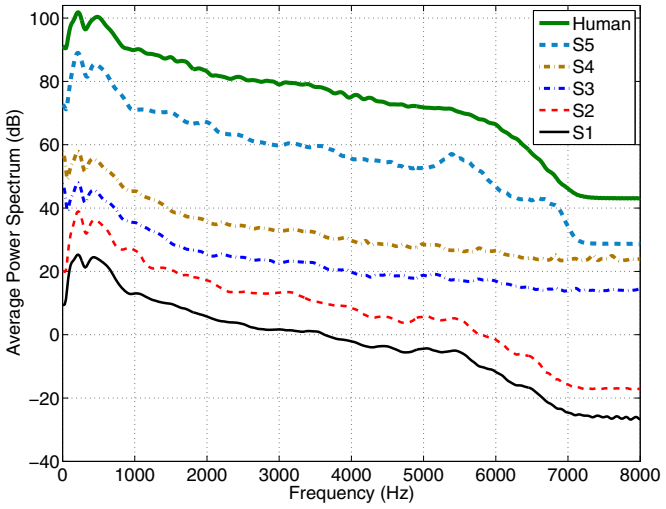

Fig. 2. Long-term average power spectra of synthetic and human speech signals (we used 2525 speech files per each method to compute an average). The spectra have been shifted by 10 dB with respect to each other.

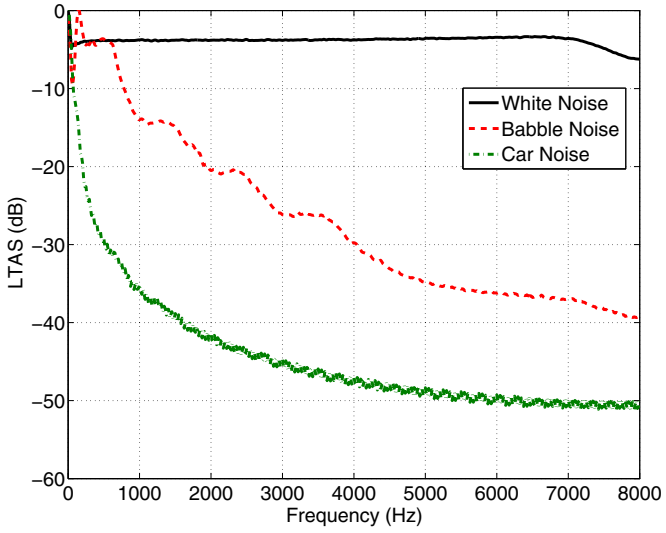

Fig. 3. Long-term average power spectra of different noise types used in the experiments.

conditions are well-studied and several techniques have been proposed to improve the performance. However, since spoofing detection has only recently been drawn attention, its performance under degradation and possible solutions for mismatched conditions are unknown. Thus, a thorough analysis on the effect of noise is necessary for the anti-spoofing research.

In this study, we consider three noise types: (i) white noise, (ii) babble noise and (iii) car noise. The LTAS variations of each noise type are shown in Fig. 3.

5. Feature extraction methods

Speech features representing short-term spectral features, which are mostly used for speech and speaker recognition, are also employed in speech-based spoofing detection. A comparative evaluation of a large number of speech features for this task is available in Sahidullah et al. (2015). In this paper, we focus on the most promising (or otherwise popular) features for noise-robust spoofing detection, namely, mel-frequency cepstral coefficients (MFCCs), inverted mel-frequency cepstral coefficients (IMFCCs) (Chakroborty et al., 2007), spectral centroid magnitude coefficients (SCMCs)

Table 1
Summary of the features and their parameters used in this study. Check marks represents corresponding post processing is applied whereas empty entries correspond to opposite.

Features	Frame length/shift	# DFT bins	Filters			Coefficients	Post processing		
			#	Type	Scale		Δ	ΔΔ	CMS
MFCC	20 ms/10 ms	512	32	Triangular	Mel	$c_0 - c_{31}$	✓	✓	✓
IMFCC	20 ms/10 ms	512	32	Triangular	Mel	$c_0 - c_{31}$	✓	✓	✓
SCMC	20 ms/10 ms	512	32	Rectangular	Linear	$c_0 - c_{31}$	✓	✓	✓
MHEC	20 ms/10 ms	-	32	Gammatone	ERB	$c_0 - c_{31}$	✓	✓	✓
RPS	20 ms/10 ms	512	32	Triangular	Mel	$c_0 - c_{31}$			
MGD	20 ms/10 ms	512	-	-	-	$c_0 - c_{31}$	✓	✓	✓
CosPhase	20 ms/10 ms	512	-	-	-	$c_0 - c_{31}$			

(Kua et al., 2010), recently proposed constant Q cepstral coefficients (CQCC) (Todisco et al., 2016) and relative phase shift (RPS) (Leon et al., 2011; 2012; Sánchez et al., 2015), modified group delay (MGD) (Murthy and Gadde, 2003) and cosine phase (CosPhase) (Wu et al., 2012b) features. MFCC and IMFCC are based on filter bank analysis, SCMC contains detailed information of subband while RPS, MGD and CosPhase carry phase-related information. In addition to magnitude and phase based features, we also evaluate recently proposed mean Hilbert envelope coefficient (MHEC) feature used successfully for robust speaker and language recognition (Sadjadi and Hansen, 2015).

The features and their parameters used in this study are summarized in Table 1. All the features have been made as comparable as possible: their frame rates, DFT sizes, number of filters and dimensionality are the same (where applicable). Feature post-processing techniques (none or deltas followed by cepstral mean subtraction) were optimized for each feature set separately. In the following, we briefly describe each of the features.

5.1. Mel-frequency cepstral coefficients (MFCCs)

In short-term speech processing, the speech signal is first divided into short overlapping frames (here 20 ms frames with 10 ms overlap is used). Then, the power spectrum of each Hamming windowed frame is computed using discrete Fourier transform (DFT) by

$$|X[k]|^2 = \left| \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N} \right|^2 \quad 0 \le k \le K-1, \tag{5}$$

where, k is the DFT bin and $\mathbf{x} = [x[0], \ldots, x[N-1]]$ is a windowed speech frame (assumed to be zero outside of the interval $[0, N-1]$). In standard filterbank based feature extraction schemes, the power spectrum is processed using a set of overlapping band-pass filters. Logarithmic filter bank outputs are then converted into cepstral coefficients by applying discrete Cosine transform (DCT). Generally, triangular filters spaced in mel-scale are used as filterbank and the resulting features are the mel-frequency cepstral coefficients (MFCCs).

5.2. Inverted Mel-frequency cepstral coefficients (IMFCCs)

In MFCCs, filters have denser spacing in low-frequency region. The IMFCC features are extracted using an *inverted* Mel scale (Chakroborty et al., 2007), implemented in practice by flipping the Mel-scaled filter bank in frequency axis giving more emphasis on the high-frequency region. Fig. 4 shows an example of triangular filters spaced on Mel and inverted Mel scales. Otherwise, all the processing steps remain the same as in MFCC extraction.

5.3. Spectral centroid magnitude coefficients (SCMCs)

Spectral centroid magnitude contains speech information similar to magnitude at the formant frequencies (Kua et al., 2010). The spectral centroid magnitude coefficients (SCMCs) are computed as follows. First, spectral centroid magnitude (SCM) for the ith subband of speech frame is computed as:

$$\mathrm{SCM}_i = \frac{\sum_{k=0}^{K/2} f[k]|X[k]|w_i[k]}{\sum_{k=0}^{K/2} f[k]w_i[k]}, \tag{6}$$

where $f[k]$ is the normalized frequency ($0 \le f[k] \le 1$) and $w_i[k]$ is a window function in the frequency domain (here rectangular window is used) for computing the centroid of the ith subband. In the next step, the logarithm of SCM values are computed and converted into feature coefficients (SCMCs) by using DCT. This subband feature outperformed other related features in our preliminary comparison (Sahidullah et al., 2015).

5.4. Constant Q cepstral coefficients (CQCCs)

CQCC is another magnitude-based feature proposed very recently to spoofing detection Todisco et al. (2016). It was reported to achieve the lowest EERs for known and unknown attacks on the ASVspoof 2015 corpus. CQCC uses a wavelet-like, perceptually motivated time-frequency analysis known as the *constant Q transform* (CQT) Brown (1991). In contrast to the fixed time-frequency resolution of the short-term Fourier transform, CQT provides a higher frequency resolution for the lower frequencies and a higher temporal resolution for the higher frequencies. In order to compute the cepstrum, the CQT-based power spectrum is first uniformly sampled in linear frequency scale. Finally, CQCCs are computed by performing DCT. In this work, we have used the implementation of CQCC made publicly available by EURECOM.[7] The default values of the control parameters were used in our experiments.[8]

5.5. Mean Hilbert envelope coefficients (MHECs)

Gammatone filterbank based features are sometimes used in speech and speaker recognition especially under mismatched and reverberated speech conditions (Mitra et al., 2014; Sadjadi and Hansen, 2015; Yin et al., 2011). In general, the speech signal is first processed by a bank of Gammatone filters that are equally spaced on the equivalent rectangular bandwidth (ERB) scale between 100 and 8000 Hz (assuming the speech signal is sampled at 16 kHz). In this study, we used the Gammatone filterbank implementation provided by Auditory toolbox (Slaney, 1998).

[7] http://audio.eurecom.fr/software/CQCC_v1.0.zip

[8] For the CQCCs, the number of feature vectors implied by the default parameters used in Todisco et al. (2016) is slightly different from the other features. On average, CQCCs produces about 17% more feature frames.

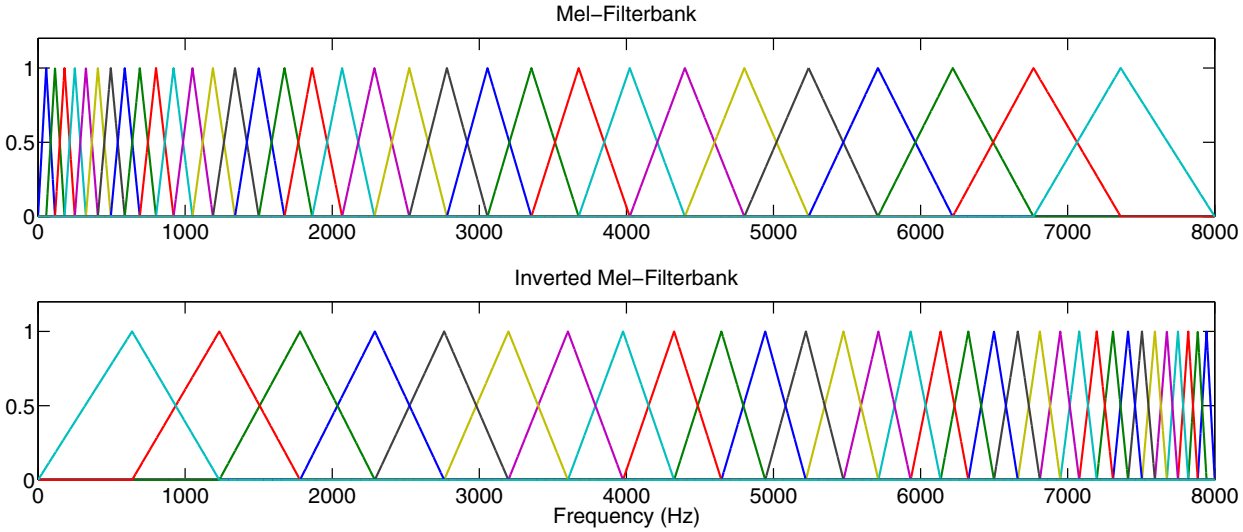

Fig. 4. Triangular filters spaced on Mel and inverted-Mel scale.

Mean Hilbert envelope coefficients (MHECs) were recently proposed for noise robust speech, speaker, and language recognition (Sadjadi et al., 2012; Sadjadi and Hansen, 2015). It uses the output of each filter in the filterbank. Calculation of MHEC features is performed through the following steps:

1. First, the speech signal is passed through a Gammatone filterbank consisting of 32 filters and for each Gammatone filter output, the temporal envelope, the squared magnitude of the analytical signal is obtained using the Hilbert transform.
2. The envelope is smoothed by applying a low pass filter with cut-off frequency of $f_c = 20$ Hz.
3. Short-term energy is computed from each smoothed envelope by framing and windowing.
4. MHECs are computed from the energies using logarithmic compression followed by DCT.

5.6. Relative-Phase shift (RPS) features

The relative phase shift (RPS) features Leon et al. (2011); 2012); Sánchez et al. (2015) are based on harmonic modeling of the speech signal. In harmonic modeling, each frame is approximated as the sum of sinusoids in the form:

$$x[n] = \sum_k A_k[n]\cos(\phi_k[n]), \tag{7}$$

where $A_k[n]$ is the amplitude and

$$\phi_k[n] = 2\pi k F_0 n + \theta_k \tag{8}$$

is the instantaneous phase of the kth harmonic. F_0 is the fundamental frequency and θ_k is the initial phase of the kth harmonic. The instantaneous phase depends on the time instant n and harmonic, k, whereas the initial phase, θ_k, is independent of the time instant. The RPS value is the *phase shift* of the kth harmonic component with respect to fundamental frequency (Leon et al., 2011; 2012; Sánchez et al., 2015). It is calculated by solving for θ_k by equating the time instants n_i in (8) between the kth harmonic and the reference fundamental frequency, assuming $\theta_1 = 0$:

$$\theta_k = \phi_k[n_i] - k\phi_1[n_i], \tag{9}$$

We used COVAREP tool (Degottex et al., 2014) to compute the RPS values. COVAREP tool uses 100 ms frames with 10 ms frame shift for computing the F_0. The RPS features are computed from the RPS values by performing phase unwrapping and then differentiation followed by Mel-scale integration and DCT as in Leon et al. (2011); 2012). Similar to other front-end configurations, the 0th coefficient is included.

5.7. Modified group delay function

Group delay function representing phase information shows spurious high amplitude spikes at zeros of short-term magnitude spectrum due to excitation sources. Modified group delay function (MGDF) Murthy and Gadde (2003) is formulated by suppressing zeros of the magnitude spectrum. It is defined as,

$$\tau(k) = \mathrm{sgn} \times \left| \frac{[X_R(k)Y_R(k) + X_I(k)Y_I(k)]}{H(k)^{2\gamma}} \right|^{\alpha} \tag{10}$$

where sgn is the sign of $X_R(k)Y_R(k) + X_I(k)Y_I(k)$, $X_R(k)$ and $X_I(k)$ represent real and imaginary part of DFT for a speech frame $x(n)$ and $Y_R(k)$ and $Y_I(k)$ represent the real and the imaginary parts of DFT for $nx(n)$. $H(k)$ is the speech spectrum after cepstral smoothing, while α and γ are two control parameters. Cepstral like features are computed from MGDF using DCT. This feature was used for synthetic speech detection in Wu et al. (2012b). In the experiments, the parameters α and γ are set to 0.3 and 0.1, respectively.

5.8. Cosine phase (CosPhase) features

The phase spectrum computed using short-time Fourier transform can be used for speech feature extraction. Since the phase spectrum calculated directly from the complex STFT parameters is discontinuous with respect to frequency, we first unwrap the phase spectrum. The cosine function is then applied to the unwrapped phase spectrum to normalize the range in $[-1.0, 1.0]$. Then discrete cosine transform (DCT) is applied to the cosine normalized phase spectrum. This feature is called as CosPhase and used in spoofing detection (Wu et al., 2012b).

6. Experimental setup

6.1. Database

The experiments are conducted on the ASVspoof 2015 database (Wu et al., 2015c) which consists of speech data with no channel or background noise collected from 106 speakers (45 male and 61 female) and three subsets with non-overlapping speakers:

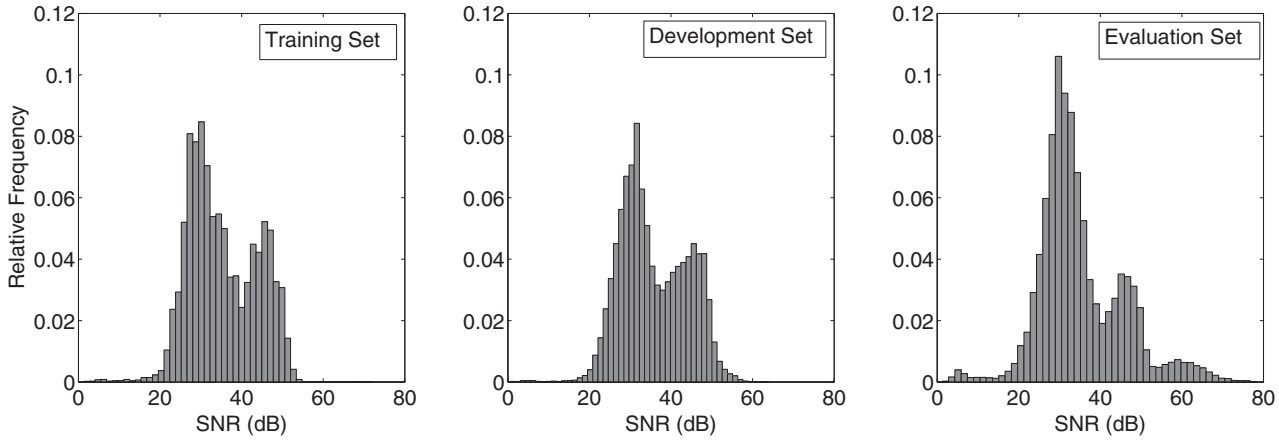

Fig. 5. Distributions of estimated SNR levels for each subset of ASVspoof 2015 dataset.

Table 2
Statistics of the ASVspoof 2015 database, used in the experiments Wu et al. (2015c).

Subset	Number of speakers		Number of utterances	
	Male	*Female*	*Natural*	*Synthetic*
Training	10	15	3750	12,625
Development	15	20	3497	49,875
Evaluation	20	26	9404	184,000

- **Training** subset is used to train genuine and spoofed classes for spoofing detection. It contains natural and five different types of spoofed speech: three are generated using voice conversion and the rest using speech synthesis. Voice conversion algorithms are (i) frame-selection (S1), (ii) spectral slope shifting (S2) and (iii) Festvox (S5) system[9] whereas the speech synthesis spoofs are based on hidden Markov model-based methods (S3 and S4).
- **Development** subset is used to optimize spoofing detectors. It contains the same five spoofing methods (S1-S5) as the training subset.
- **Evaluation** subset is used for evaluating the final performance of the system. It contains five "known" algorithms seen in the training and development subsets (S1-S5) as well as five "unknown" algorithms (S6-S10).

Table 2 summarizes speaker and utterance information for each subset.

To analyze the original ASVspoof 2015 data regarding noise level and to show the quality of recordings in the database before interpreting the results, we computed the SNR level of recordings. Fig. 5 shows the histograms of estimated SNR levels[10] for each subset of the original ASVspoof 2015 dataset. All the speech files from the training set are used to plot the histogram for this subset, whereas randomly selected 30,000 speech signals are used to generate histograms for Evaluation and Development subsets. A vast majority of the signals have a relatively high SNR exceeding 20 dB. The evaluation subset contains also signals with very high SNRs (approximately 8% of 30,000 files have SNR $>$ 50 dB).

We use *Filtering and Noise Adding Tool* (FaNT)[11] to corrupt the original ASVspoof 2015 signals with noise for introducing controlled degradation. FaNT is an open-source tool which follows the ITU recommendations for noise adding and filtering. To be more precise, it uses psychoacoustic speech level computation based on the ITU recommendation P.56 (*objective measurement of active speech level*). We digitally add white, babble and car noises from NOISEX-92 database (Varga and Steeneken, 1993). For each noise type we consider 3 SNR levels: 0, 10 and 20 dB. The reasons for selecting these types of noise are the following: (i) White noise has a flat spectral density and it masks all the frequency components uniformly. Although it rarely represents a real-case scenario, it is a commonly used control noise in studying robust speech processing methods. (ii) Babble noise is one of the most difficult noise types in speech applications containing a mixture of multiple speakers — a situation that occurs on a daily basis in any crowded place (Krishnamurthy and Hansen, 2009). (iii) Car noise is another noise type that may frequently occur in our daily life such as making a phone call while driving.

In the experiments, we consider **noise mismatched** condition by training the natural and synthetic speech models using the original clean training files, but test them on noisy files. The reason for this choice is practicality: in a real-world deployment of ASV technology in smartphones or other portable devices, the operation environment of the user would be rarely known precisely.

6.2. Classifier and features

We use 32 coefficients (including c_0) and 32 filters in filterbank for every method. This is done to have comparable results for different feature extraction methods. We apply energy-based voice activity detection (VAD) (Kinnunen and Li, 2010, p. 24) on clean data to get speech/non-speech labels. Using clean VAD labels allows us to focus merely on the effect of noise on synthetic speech detection rather than mixed effects of VAD and feature set. These labels are used to discard non-speech frames for both clean and noisy speech.

For GMM-based classification, we use two models to represent natural and synthetic speech classes (see details in Section 3). GMM for each class has 512 components and is trained using 5 EM iterations (the performance differences for larger number of components were negligible in our initial experiments).

For i-vector based classification, we train a gender-independent universal background model (UBM) consisting of 512 Gaussians using 9000 utterances from 150 male and 150 female speakers from the WSJ0 & 1 corpora Wall Street Journal Corpus (2015). To train the **T**-matrix, we select 8945 utterances from 178 male and

[9] http://www.festvox.org
[10] SNREval Toolkit from http://labrosa.ee.columbia.edu/projects/snreval/ is used to estimate the SNR levels.
[11] http://dnt.kr.hsnr.de/

177 female speakers from the WSJ0 & 1 databases[12] and run EM-algorithm for 5 iterations. The extracted 600 dimensional i-vectors are further processed by applying *within-class covariance normalization* (WCCN) Hatch et al. (2006), followed by projection to the unit sphere Garcia-Romero and Espy-Wilson (2011). The logic behind WCCN is not to normalize within-speaker variation Dehak et al. (2011), like it is done for speaker recognition, but to normalize within-class (natural or synthetic) variation. To this end, we separate the training data into natural and synthetic classes and use them to compute WCCN transformation matrix **B** (Dehak et al., 2011, p. 791). PLDA model trained on original (clean) data is used in noisy spoofing detection experiments.

6.3. Combined countermeasures via score fusion

Given the wide diversity and varied difficulty of existing and future spoofing attacks, it might be difficult to come up with a single feature set to detect all possible attacks. As an example, phase-related features might be suited to detect attacks whose vocoder discards natural phase information while other methods may possess superior noise robustness. This motivates exploration towards countermeasures that includes a bank of different front-ends, some being potentially specialized to detect particular types of attacks. To this end, here we consider two score level fusion strategies to maximally benefit from the complementarity of our features: 1) **Fusion 1: Score averaging** – a simple technique, which does not require any training, 2) **Fusion 2: weighted sum**, where fusion weights and a bias term are estimated using logistic regression (Brümmer et al., 2007). We use the development data to train the parameters for each noise type and SNR level.

6.4. Performance measure

Following the evaluation plan of ASVspoof 2015, equal error rate (EER) is used as the objective performance criterion in the experiments. EER corresponds to the threshold at which false acceptance (P_{fa}) and miss rate (P_{miss}) are equal. P_{fa} is the ratio of number of spoof trials detected as genuine speech to the total number of spoof trials and P_{miss} is the ratio of number of genuine trials detected as spoofed to the total number of genuine trials. The EERs reported in this work were computed using the bosaris toolkit[13] which computes the EER on receiver operating characteristics (ROC) convex hull (ROCCH) that is an interpolated version of standard ROC.

7. Results

We conduct the experiments separately on the development and evaluation parts of ASVspoof 2015. The development part is first used for optimizing the system parameters and configurations. The feature extraction method that yield the lowest EERs is then selected for further experiments on the evaluation part.

7.1. Effect of feature post-processing

In our first experiment on the development set, we study the effect of feature post-processing. Specifically, we study the appending Δ and $\Delta\Delta$ features and cepstral mean subtraction (CMS). The results on MFCC and CosPhase features are shown in Fig. 6. Here, MFCC and CosPhase features are selected as representatives of magnitude and phase-based features, respectively. The upper row corresponds to the MFCC and the lower row to the CosPhase features. For the original (clean) case, 2.24% EER is obtained using only the base MFCCs. Appending Δ and $\Delta\Delta$ features to the MFCCs reduces the EER to 0.49%. Applying CMS slightly reduces the performance for the clean case (0.84% EER). For the CosPhase features, in turn, the lowest EER (1.09%) is obtained with the base features on clean data in contrast to MFCCs. Appending Δ features to the base CosPhase features almost doubles the EER (2.16%). Appending $\Delta\Delta$ or applying CMS does not help to increase the synthetic speech detection performance with CosPhase features.

For the noisy case, appending the Δ and $\Delta\Delta$ coefficients considerably improves the accuracy in most cases. For example, we see 78% relative improvement over the base MFCCs for babble noise at 20 dB SNR (EER 16.29% → 3.61%). Similarly, applying CMS on top of the dynamic features improves performance considerably. Whereas, post-processing shows an opposite effect with CosPhase features where the smallest EERs are obtained with the base features independent of the noise type and SNR.

The results in Fig. 6 are for the MFCC and CosPhase features. The results were similar for the other studied features. Namely, for the magnitude (MFCCs, IMFCCs, SCMC and CQCC) and MHEC features, the best performance is obtained with the full post-processing (included deltas followed by CMS) whereas for the phase-based features the raw features yield the smallest EERs except for MGD. Out from the 10 conditions evaluated (3 SNRs × 3 noise types plus the clean data), MGD features with deltas and feature normalization yielded the lowest EERs in 6 cases. Thus, in all the remaining experiments, we will adopt the raw RPS and CosPhase features. For all the rest of the features, we include deltas and CMS.

7.2. Comparison of features

The results on development set for different features using GMM are summarized in Table 3. For the clean (original) case, the RPS features yield the lowest EER. However, under additive noise, especially for white noise and at low SNR levels of car and babble noises, the performance of RPS is relatively poor. This could be because RPS requires estimated F_0 values that are difficult to extract reliably from noisy data. For babble and car noises at high SNRs (20 dB), RPS yields reasonable accuracy. The SCMC features perform well for the babble and car noises, whereas for white noise, MHEC yields lower EERs. To sum up Table 3, none of the feature sets is consistently superior to others. In most cases, SCMC outperforms the other features. Out of the three phase features (RPS, MGD and CosPhase), CosPhase features are superior to RPS and MGD under white noise case. However, RPS outperforms MGD and CosPhase for babble and car noises. In general, magnitude features outperform phase-related features independent of noise type and SNR.

Applying score fusion to the eight feature extraction methods considerably improves the accuracy for all cases including the original (clean) condition as Table 3 indicates. Weighted sum technique where the weights of each individual system are estimated with logistic regression (indicated as Fusion2 in Table 3) yield lower EERs than score averaging fusion (Fusion1). The effect of each individual feature set on the fusion performance has been investigated and it was found that excluding RPS from the fusion (applying score fusion to the six remaining feature sets) dramatically increases the EERs irrespective of noise and SNR. This suggests that RPS consists of complementary information even though it gives poor stand-alone performance compared to other features.

[12] Usually 283 speakers from WSJ0 & 1 databases are used in most studies which is the official training set of the corpora. In our experiments, we included test sets of WSJ0 & 1 corpora in addition to training set which yields a total of 177 male and 178 female speakers.

[13] https://sites.google.com/site/bosaristoolkit/

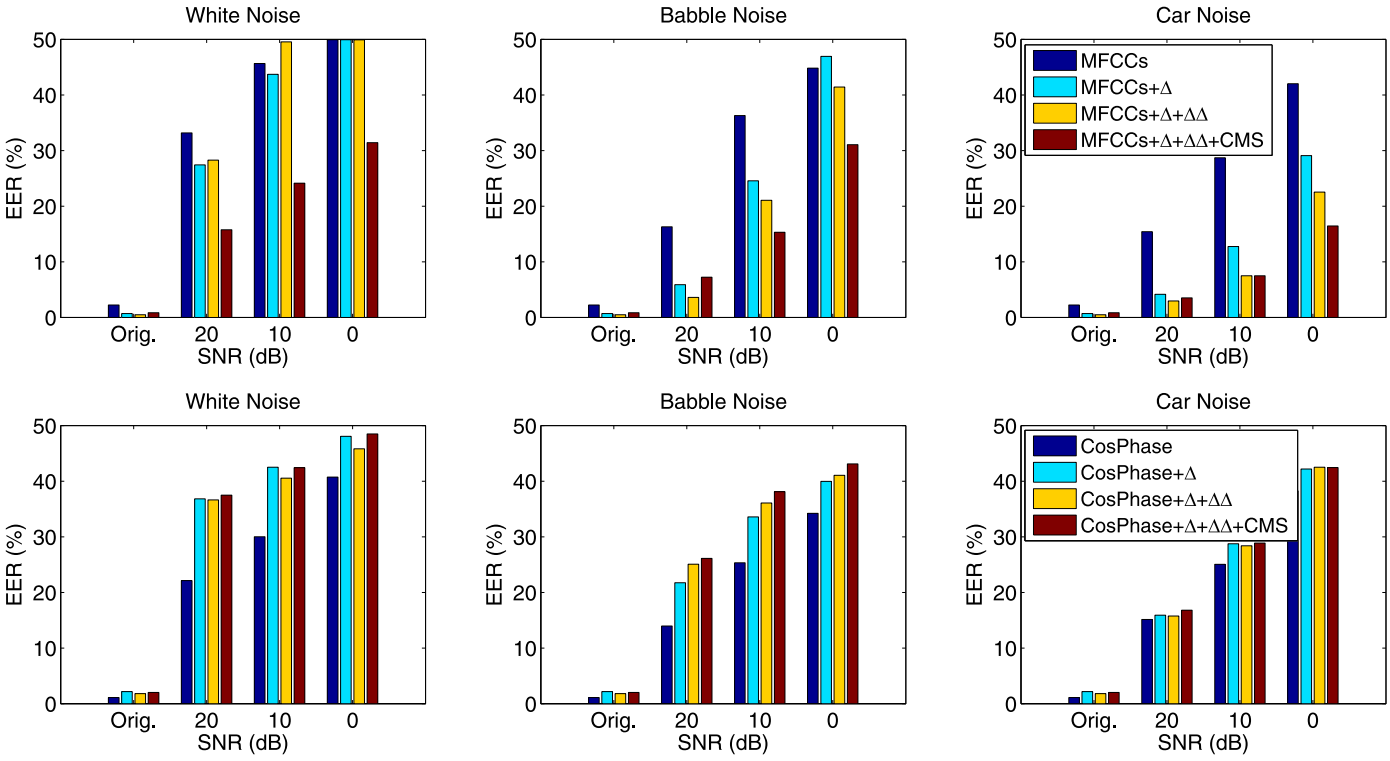

Fig. 6. Effects of Δ and ΔΔ MFCC features and Cepstral Mean Subtraction on synthetic speech detection. First row, MFCC features. Second row, CosPhase Features.

Table 3
Comparison (EER, %) of different front-end features in noisy conditions on development set using Gaussian Mixture Model classifier. The results for clean original condition are presented as well as the average results for all noisy sub-conditions.

Noise type	SNR (dB)	MFCC	IMFCC	SCMC	CQCC	MHEC	RPS	MGD	CosPhase	Fusion1	Fusion2
Original		0.84	0.91	0.38	0.44	3.92	**0.15**	1.25	1.09	0.02	**0.00**
White	20	15.75	34.17	21.91	33.41	**12.08**	37.64	28.35	22.12	12.17	**8.45**
	10	24.13	44.56	32.19	38.13	**22.2**	41.37	39.23	30.02	18.84	**16.09**
	0	**31.42**	48.86	39.86	45.55	33.37	43.61	46.45	40.73	29.42	**27.69**
Babble	20	7.23	5.66	**2.71**	18.07	11.06	5.26	13.77	13.97	1.89	**0.56**
	10	15.32	15.4	**9.36**	29.49	25.58	20.04	26.26	25.33	7.72	**4.96**
	0	31.05	37.73	**30.09**	41.60	40.87	39.90	40.12	34.22	26.58	**22.85**
Car	20	3.51	1.94	0.87	9.26	8.96	**0.74**	9.30	15.14	0.39	**0.03**
	10	7.48	4.69	**2.48**	18.04	19.47	5.75	15.84	25.05	2.56	**0.67**
	0	16.44	14.27	**8.74**	29.42	33.12	24.03	29.72	38.23	11.83	**7.12**
Average		16.92	23.03	**14.85**	26.34	21.06	21.84	25.02	24.58	11.14	**8.84**

7.3. *Effect of speech enhancement*

Next, we study the effect of speech enhancement techniques. To this end, magnitude and power spectral subtraction algorithms (Berouti et al., 1979; Boll, 1979) and Wiener filtering (Lim and Oppenheim, 1979) approaches are adopted. Detection error trade-off (DET) curves for different speech enhancement methods for each noise type, at 0 dB SNR and using MFCC features with deltas and CMS as well as CosPhase features, are shown in Fig. 7. Here, the DET curves are generated by pooling the scores of all the individual attacks.[14] Fig. 7 indicates that the attempted speech enhancement techniques do not yield any performance gains for MFCC features. For CosPhase features, magnitude spectral subtraction slightly improves the performance for white noise whereas for babble and car noises speech enhancement methods do not improve the performance. These three methods were applied to SCMC features as well in order to analyze the effect of speech enhancement on different features and to check whether the observations can be generalized and the similar results have been obtained. Apart from these three popular methods, other methods including minimum mean square error (MMSE), logarithmic MMSE (logMMSE) and iterative Wiener filtering techniques (as available in the Appendix of Loizou (2007)) were studied, without success. The reduction on the performance after speech enhancement might be because speech enhancement introduces musical noise and other processing artifacts that mask the synthesis or conversion artifacts.

A recent independent study (Yu et al., 2016) confirms the ineffectiveness of traditional unsupervised speech enhancement techniques for spoofing detection in noisy condition. Currently, similar to most speech processing tasks, the use of deep neural network (DNN) based techniques is extensively studied on speech enhancement (Han et al., 2015; Xu et al., 2014; 2015) and could be an interesting approach. However, as DNNs require large amounts of

[14] Although in ASVspoof 2015 the evaluation metric is averaged EER over different attacks, producing a single DET curve that would coincidence with this operating point is not obvious. Thus, here the scores are pooled to generate the DET plot and to compute the corresponding EERs in Fig. 7 legends.

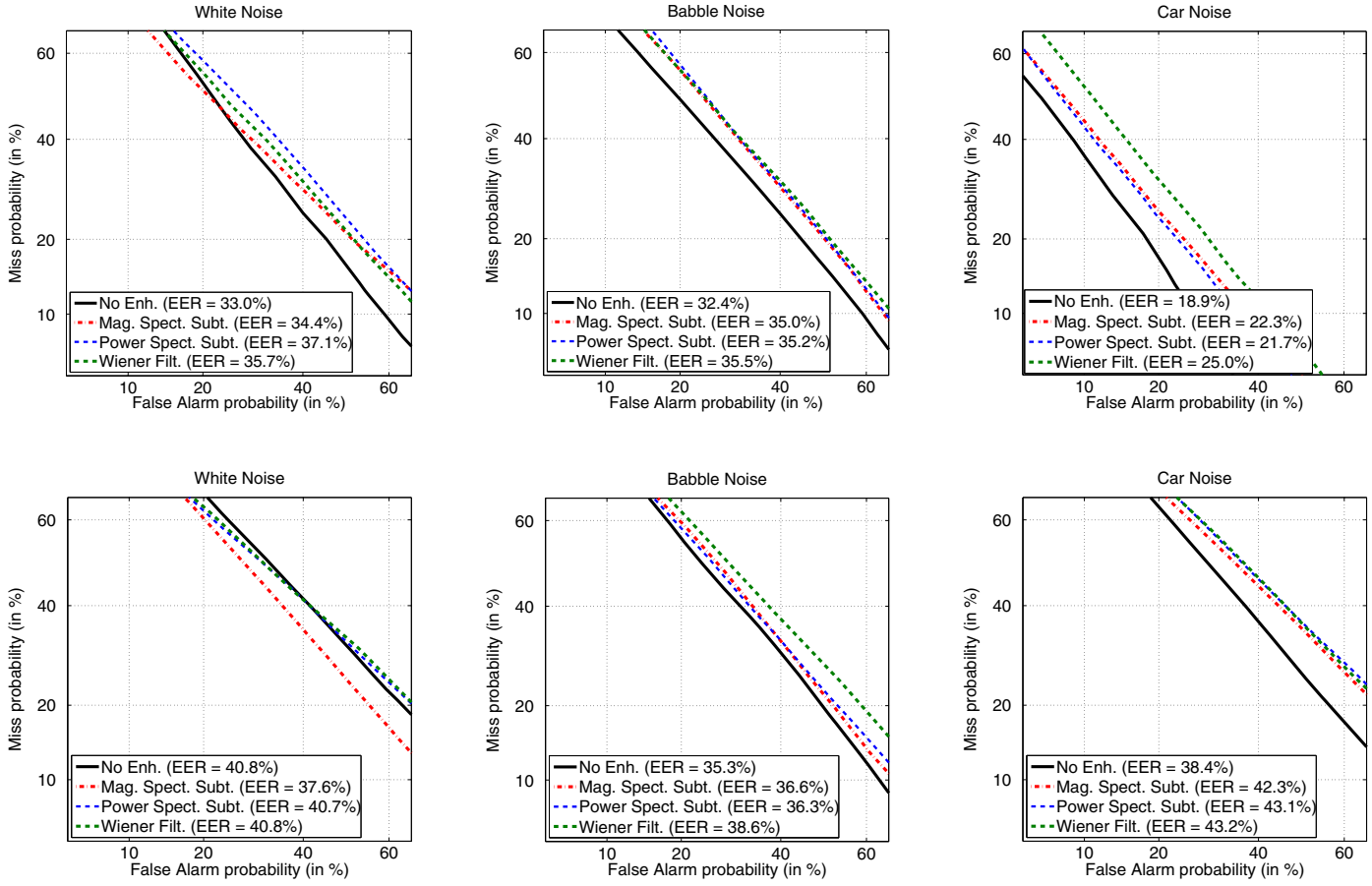

Fig. 7. DET curves for different speech enhancement techniques under additive noise (0 dB). First row, MFCC features. Second row, CosPhase features.

additional training data from different noisy conditions for supervised training, they are not addressed in this study that focuses on DSP-based unsupervised speech enhancement techniques. Further, achieving performance improvement in unseen noisy condition appears challenging even with DNN-based speech enhancement methods (Sun et al., 2016).

7.4. i-vector countermeasures from different features

Up to this point, we have utilized the computationally light GMM classifier to study different feature configurations. In our last experiments with the development set, we study an i-vector based countermeasure. To this end, i-vector extractors are trained from scratch for all the seven acoustic feature sets. The results are provided in Table 4 for both cosine and PLDA scoring. For clean (original) case, the recently proposed CQCC features yield the smallest EER among the eight methods. While the performance of CQCC features with i-vector back-end is superior to GMM classifier on clean data, for the remaining seven feature extraction methods, GMM back-end outperforms the i-vector back-end. For additive noise cases, i-vector is inferior to GMM independent of the noise type and feature extraction method. Similar results for GMM and i-vector techniques were found in our recent comparative study of classifiers for synthetic speech detection (Hanilçi et al., 2015). This could be because of the short duration of recordings (approximately 3 seconds) that ASVspoof 2015 consists of. Similar observation for i-vector performance on short utterances were found in Li et al. (2016) where GMM and i-vector systems were compared for speaker verification task using short data and it was found GMM recognizer outperforms i-vector system.

Similar to GMM experiments under additive noise (Table 3), none of the features are systematically superior to others. The features that yield the lowest EERs are different for each noise type and SNR level. MHEC yields the highest performance for white noise whereas, for the babble and car noises, RPS is superior to other features at high SNRs (20 and 10 dB). Concerning the two i-vector back-end variants, PLDA does not bring substantial improvements in comparison to cosine scoring. The most considerable performance improvement with PLDA is obtained with CosPhase features using original (clean) data (EER reduced from 11.80% to 4.54% with PLDA). Similar to the results with GMM classifier, CosPhase features outperform the other phase features (RPS and MGD) under white noise. However, for the babble and car noises, RPS outperforms other phase features. The performance of MGD features, in turn, lies between RPS and CosPhase. In the next experiments on Evaluation set, MFCC and SCMC features as two magnitude and RPS and MGD features as phase based features using GMM and i-vector techniques will be considered.

7.5. Results on evaluation set

In the experiments with the evaluation portion of ASVspoof 2015, we first study the performance of each individual attack using clean data with two magnitude (MFCC and SCMC) and two phase (RPS and MGD) based features. The EERs obtained with GMM and i-vector techniques for the individual attacks are summarized in Table 5. Similar to observations found on the development set, GMM outperforms both i-vector scoring variants independent of the attack type and the features.

Table 4
Comparison (EER, %) of different front-end features in noisy conditions on development set using Cosine/Probabilistic Linear Discriminant Analysis i-vector classifiers. The results for the clean (original) condition are presented as well as the average results for all noisy sub-conditions. The lower half of the table presents a difference between the corresponding EERs for PLDA and cosine scoring. Blue cells indicate conditions where PLDA scoring is advantageous to cosine scoring, whereas red cells indicate the opposite.

Cosine Scoring

	SNR (dB)	MFCC	IMFCC	SCMC	CQCC	MHEC	RPS	MGD	CosPhase	Fusion1	Fusion2
	Original	5.12	**3.24**	5.30	0.26	12.31	5.18	8.40	11.80	0.01	**0.00**
White	20	26.48	45.51	39.97	41.55	**26.05**	39.97	39.34	32.61	20.37	**17.27**
	10	36.35	47.60	44.15	44.76	**30.71**	45.24	45.70	35.98	31.4	**26.26**
	0	43.47	48.26	46.68	48.27	**39.20**	47.60	48.04	47.98	41.55	**37.53**
Babble	20	20.94	28.07	24.44	27.63	25.58	**19.10**	25.12	33.11	6.15	**5.75**
	10	33.59	40.54	34.97	39.21	33.54	**31.03**	36.37	41.23	18.71	**18.13**
	0	45.71	48.15	45.02	46.20	43.65	43.73	45.59	**43.47**	38.88	**37.57**
Car	20	24.00	13.56	14.34	13.46	22.53	**11.88**	21.42	33.84	2.04	**1.86**
	10	33.67	26.28	22.61	25.53	27.76	**22.14**	29.78	43.91	8.02	**8.01**
	0	39.62	40.39	**33.12**	37.84	34.76	38.34	38.82	48.45	23.18	**22.24**
	Average	30.89	34.16	31.06	32.47	29.60	30.42	33.85	36.75	19.03	**17.46**

PLDA - Cosine

	SNR (dB)	MFCC	IMFCC	SCMC	CQCC	MHEC	RPS	MGD	CosPhase	Fusion1	Fusion2
	Original	-0.09	0.81	0.73	-0.01	0.39	-0.15	0.08	-7.26	0.01	0.00
White	20	-1.16	-0.87	-0.93	2.21	2.18	3.12	0.06	3.12	1.69	1.88
	10	-1.48	0.14	-0.04	1.25	2.49	1.53	-0.27	0.7	-1.40	2.44
	0	0.2	0.32	0.27	0.3	1.29	-0.16	-0.15	-5.69	-1.99	0.65
Babble	20	-0.29	-0.04	0.04	0.84	0.49	4.05	0.7	2.81	2.87	1.35
	10	-0.46	-0.92	-0.42	0.94	0.12	3.99	-0.13	1.18	4.83	1.52
	0	0.11	0.08	-0.36	-0.19	-0.12	1.69	-0.31	0.77	0.58	3.01
Car	20	0.35	0.97	0.55	0.61	0.24	1.91	12.42	2.17	1.59	0.72
	10	-1.17	-0.23	0.53	1.34	0.51	3.97	0.28	0.54	3.79	0.65
	0	-1.61	-1.32	-0.61	0.22	0.82	3.85	-0.38	-0.34	5.76	0.56
	Average	-0.56	-0.11	-0.03	0.75	0.85	2.38	0.11	0.28	1.77	1.27

Table 5
Comparison (EER, %) of Gaussian Mixture Model classifier and two i-vector based classifiers: Cosine scoring and Probabilistic Linear Discriminant Analysis. We consider individual attacks on clean evaluation set using selected two magnitude (MFCCs and SCMC) and two phase (RPS and MGD) based features.

Features	Classifier	Known attacks					Unknown attacks					Avg.
		S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	(S6-S9)
MFCC	GMM	**0.00**	3.54	**0.00**	**0.00**	0.70	1.10	0.80	0.53	0.11	27.34	0.63
	Cosine	2.89	9.26	2.67	2.66	6.01	8.07	3.64	5.03	3.07	46.49	4.95
	PLDA	3.26	9.67	2.16	2.39	5.84	8.23	3.55	6.97	3.29	47.11	5.51
SCMC	GMM	**0.00**	1.22	0.05	**0.02**	0.60	0.46	**0.07**	0.31	**0.02**	29.92	**0.21**
	Cosine	4.24	12.31	2.08	2.27	5.46	7.64	3.03	2.73	2.45	44.17	3.96
	PLDA	5.29	12.72	2.61	2.90	5.76	8.33	3.76	4.72	3.14	46.47	4.98
RPS	GMM	**0.00**	**0.02**	0.10	0.10	**0.04**	2.00	**0.01**	0.92	**0.00**	45.18	0.73
	Cosine	3.73	3.32	5.06	4.90	6.25	10.62	9.03	17.21	3.79	46.11	10.16
	PLDA	4.20	3.74	4.46	4.12	4.49	11.11	14.38	17.03	4.53	46.93	11.76
MGD	GMM	0.10	3.45	0.08	0.11	2.42	4.26	0.96	2.42	1.74	24.32	2.34
	Cosine	7.19	14.74	5.04	5.48	11.42	12.42	11.82	13.00	11.09	36.59	12.08
	PLDA	8.17	15.33	4.88	5.33	11.74	13.37	13.01	13.53	11.03	38.94	12.73
Fusion1	GMM	**0.00**	**0.02**	**0.00**	**0.00**	**0.02**	**0.11**	**0.04**	**0.01**	**0.00**	21.44	**0.04**
	Cosine	0.29	1.33	0.24	0.27	1.13	2.11	0.88	1.39	0.38	41.50	1.19
	PLDA	0.51	2.26	0.24	0.26	0.94	2.34	1.25	2.08	0.50	44.39	1.54
Fusion2	GMM	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	8.36	8.48	8.61	8.35	**8.27**	8.45
	Cosine	0.96	0.91	0.87	0.92	0.91	14.26	14.41	14.53	14.26	14.45	14.36
	PLDA	0.76	0.80	0.70	0.76	0.73	15.00	15.02	15.21	14.89	15.08	15.03

Independent of the classifier and features, S10 —the speech synthesis algorithm that uses MARY text-to-speech system[15]— is the most difficult attack type to detect in comparison to the other unknown attacks (S6-S9). This could be because S10 does not use any vocoder in generating the synthetic speech signals whereas the popular STRAIGHT vocoder Kawahara et al. (1999) is used in most of the remaining attacks. Thus, spoofing detectors trained with a STRAIGHT vocoder but tested without it will induce a mismatch between the training and the test samples (Wu et al., 2015c), making detection of S10 relatively more difficult.

In general, the SCMC features yield lower EERs than MFCCs with the GMM classifier except for S10. Concerning the two phase-based features, RPS outperforms MGD in most cases. Notably, MGD yields considerably better performance than RPS for S10, therefore for unknown attacks, on average. For the unknown attacks, MFCCs are superior to phase based MGD features. However, for known attacks RPS yields better accuracy than magnitude based MFCCs. For the two scoring variants of i-vector, in turn, MFCCs outperform the

[15] http://mary.dfki.de/

Table 6
Comparison (EER, %) of known and unknown attacks for Gaussian Mixture model classifier on evaluation set. In each row, the lowest EERs for the known (K) and unknown (U) attacks (S6-S9 attacks) are bolded and underlined, respectively.

Noise	SNR	MFCC		SCMC		RPS		MGD		Fusion1		Fusion2	
type	(dB)	K.	U.	K.	U.	K.	U.	K.	U.	K.	U.	K.	U.
Original		0.85	0.63	0.38	0.22	0.05	0.73	1.23	2.35	0.01	0.04	**0.00**	0.04
White	20	16.43	17.94	19.92	15.40	38.53	40.62	27.25	36.24	**13.39**	13.93	16.76	16.40
	10	25.45	29.78	33.36	32.14	42.16	44.98	37.42	38.66	**22.78**	26.13	25.91	27.71
	0	35.07	39.66	43.73	42.27	44.56	46.64	44.42	45.88	**34.29**	38.53	34.96	38.90
Babble	20	7.48	6.49	2.15	1.39	6.09	10.62	14.20	23.55	1.13	1.81	**0.69**	1.97
	10	15.59	12.76	8.32	5.30	21.17	23.71	26.30	35.65	5.81	6.52	**5.36**	8.08
	0	33.54	28.40	**29.74**	25.13	40.66	40.81	37.59	40.77	**24.90**	23.75	25.23	23.95
Car	20	3.57	2.83	0.79	0.52	0.74	3.67	9.39	16.12	0.11	0.45	**0.05**	0.38
	10	7.31	6.03	2.16	1.67	5.28	9.93	15.99	24.44	1.00	2.07	**0.72**	1.95
	0	17.33	14.69	8.59	7.36	24.66	25.67	30.32	36.63	8.17	9.38	**7.11**	8.03

SCMC features, except for S10. Overall, S10 yields extremely high EERs while reasonable accuracies are obtained for the other attacks. In most studies that report their findings on the ASVspoof 2015 data, the performance of countermeasures is reported by averaging the EER of individual unknown attacks (S1-S10), which was the official evaluation metric of the challenge. However, the average EER of unknown attacks becomes highly dependent on the performance of S10 attack. Therefore, in Table 5, the performance of unknown conditions are reported by averaging the S6-S9 attacks rather than S6-S10. Since GMM outperformed i-vectors systematically, only the GMM results are presented in the remaining experiments on the Evaluation set.

Note that in Table 5, simple score averaging (Fusion 1) performs considerably better than fusion with weights optimized using logistic regression (Fusion 2). This stems from the fact that, during the training of Fusion 2, we pool all scores together and look for a joint transformation for all the attack types. This results in almost equal performance of the system to each attack type. Unfortunately, due to a very high EER for S10, this performance could be called as being "equally bad".

The results for the noise-contaminated evaluation set obtained with GMM using selected magnitude and phase based features are given in Table 6. MFCCs yield lower EERs than SCMCs under white noise for both known and unknown attacks. For the babble and car noises, in turn, SCMCs outperform MFCCs. Similar to results on Development Set (Table 4), a considerable reduction in EERs is obtained using SCMC features over MFCCs under car and babble noise cases. For phase features, RPS is superior to MGD features for both known and unknown attacks under babble and car noises whereas MGD shows better performance than RPS under white noise case. In general, magnitude features (MFCCs and SCMCs) yield lower EERs than phase features independent of noise and SNR.

8. Conclusion

In this study, our goal was to analyze the robustness of existing state-of-the-art countermeasure systems for synthetic speech detection in the presence of additive noise. Extensive experiments were conducted using different front-ends and back-ends for three type of noises (white, babble and car) with three different noise levels (20 dB, 10 dB, and 0 dB). We evaluated the performance with five different short-term magnitude features (MFCC, IMFCC, SCMC, CQCC and MHEC) and three short-term phase features (RPS, MGD, and CosPhase). These features have successfully been used for spoofing detection in clean conditions whereas our study addresses their performance under additive noise backgrounds. As a back-end, we have experimented with two well-known approaches: Gaussian mixture model (GMM) and i-vector. We also explored the effect of various speech enhancement techniques as well as the impact of different feature post-processing methods. Finally, we have investigated fusion techniques to combine the strength of multiple systems.

Our extensive results on ASVspoof 2015 dataset indicate that additive noise contamination considerably complicates the task of synthetic speech detection. Applying standard speech enhancement techniques, such as magnitude spectral subtraction, power spectral subtraction, and Wiener filtering were not found helpful in improving the accuracy. In recent studies, it was reported that DNN-based speech enhancement techniques outperforms standard methods such as MMSE and Wiener filtering (Sun et al., 2016). Therefore, applying DNN-based speech enhancement for anti-spoofing under additive noise would be interesting for the future work. We also found that phase-based features, RPS, and CosPhase, perform better in the absence of any feature post-processing schemes like delta features or cepstral mean subtraction (CMS). But those post-processing steps were found crucial for the other features.

White noise degrades the accuracy the most. For example, in an experiment on the development set, EER increased from 0.84% to 31.42% with MFCC features and GMM back-end in the presence of white noise with 0 dB SNR. The severity of white noise can be explained with the help of comparative long-term average spectra (LTAS) of different noises. We have shown that it has a considerable effect on the entire speech spectrum unlike other two types of noises where the effect on speech spectrum is mostly partial.

Concerning the back-ends, we have observed the GMM-based classifier to consistently outperform the more sophisticated i-vector method. Poor results for the i-vector systems could be explained by short utterances or possibly suboptimal data selection to train UBM and T-matrix. Our findings on spoofing detection task also agree with the results from the previously conducted independent studies, but on the clean condition.

In the study of features using GMM as the classifier, MFCCs give best recognition accuracy in most cases in the presence of white noise while SCMCs perform better for babble and car. However, this observation is not consistent when we take i-vector systems into account. For example in an experiment with the development set, MHEC feature outperforms the other features for the i-vector-cosine system whereas MFCCs win when PLDA scoring is employed. We have also observed that RPS feature — which was successfully used in many spoofing detection studies and outperforms other features such as MFCC in clean conditions — generally yield higher EERs than standard MFCC features in the presence of additive noise. However, its performance is still superior to the other two phase based features compared: MGD and CosPhase.

The results on the evaluation section of ASVspoof 2015 further reveals that detecting *unknown* attacks is much harder than detecting *known* attacks in noisy condition. Moreover, from a detailed

study on attack-specific performances with clean speech data, we find that the notable performance difference between known and unknown attacks is mostly due to one specific spoofing attach, S10 (i.e., MARY TTS) which does not use any vocoder as the other synthetic speech generation techniques used in ASVspoof 2015. This was the general observation regarding different systems submitted to ASVspoof 2015 (Wu et al., 2015c).

Finally, we have observed considerable gain in spoofing detection performance due to fusion of multiple front-ends. For example, in the presence of 10 dB car noise, the EERs of known and unknown attack using score-average fused system are 0.99% and 7.82%, respectively, whereas best individual system (here, SCMC) gives 2.16% and 8.49%. We have also noticed that improvement for the known attack condition is relatively higher than the improvement in unknown attack. We further observe that logistic regression based fusion scheme is better for known attacks, however, score average based method is more appropriate for unknown attacks. This is because for the logistic regression approach the fusion parameters are optimized on development set, i.e., for known attacks, and those optimized parameters are used for fusion of evaluation set scores consisting known and unknown attack. Applying score average based fusion strategy is a compromise to reduce the generalization error. Preventing fusion overfitting is an important practical consideration and clearly deserves further attention.

Our results suggest that synthetic speech detection becomes more challenging in noisy conditions, similar to speaker verification in a noisy environment. This study opens a few potential directions for future work. The first one is a development of robust approaches for both front-end and back-end sides of spoofing detection systems. In front-end side, we used the most promising (or otherwise popular) features in this study. Other phase based techniques, such as RPI that was reported to perform well under noisy conditions in other speech processing tasks, would be interesting to study in antispoofing under additive noise. The other direction is a study of trustworthiness of voice biometric systems under a joint presence of spoofing attacks and noise that calls for joint optimization and evaluation of ASV and spoofing countermeasure systems.

Acknowledgments

This project has been primarily supported by the Academy of Finland (projects 253120 and 283256). The paper also reflects some results from the OCTAVE Project (#647850), funded by the Research European Agency (REA) of the European Commission, in its framework programme Horizon 2020. The views expressed in this paper are those of the authors and do not engage any official position of the European Commission. The work of Cemal Hanilçi is supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) under project #115E916

References

Alegre, F., Vipperla, R., Evans, N.W.D., Fauve, B.G.B., 2012. On the vulnerability of automatic speaker recognition to spoofing attacks with artificial signals. In: Proc. EUSIPCO, pp. 36–40.

Atal, B.S., 1974. Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification. J. Acoust. Soc. Am. 55 (6), 1304–1312.

Benesty, J., Sondhi, M.M., Huang, Y. (Eds.), 2008, Springer Handbook of Speech Processing. Springer, Berlin.

Berouti, N., Schwartz, R., Makhoul, J., 1979. Enhancement of speech corrupted by acoustic noise. In: Proc. ICASSP, pp. 208–211.

Boll, S., 1979. Suppression of acoustic noise in speech using spectral subtraction. IEEE Trans. Acoust. Speech Signal Process. 27, 113–120.

Bonastre, J., Matrouf, D., Fredouille, C., 2007. Artificial impostor voice transformation effects on false acceptance rates. In: Proc. INTERSPEECH, pp. 2053–2056.

Brown, J., 1991. Calculation of a constant Q spectral transform. J. Acoust. Soc. Am. 89 (1), 425–434.

Brümmer, N., Burget, L., Černockỳ, J., Glembek, O., Grezl, F., Karafiat, M., Van Leeuwen, D., Matě, P., Schwarz, P., Strasheim, A., 2007. Fusion of heterogeneous speaker recognition systems in the STBU submission for the NIST speaker recognition evaluation 2006. IEEE Trans. Audio Speech Lang. Process. 15 (7), 2072–2084.

Byrne, D., Dillon, H., Tran, K., Arlinger, S., Wilbraham, K., Cox, R., Hagerman, B., Hetu, R., Kei, J., Lui, C., Kiessling, J., Kotby, N.M., Nasser, N.H.A., Wafaa, Nakanishi, Y., Oyer, H., Powell, R., Stephens, D., Meredith, R., Sirimanna, T., Tavartkiladze, G., Frolenkov, G.I., Westerman, S., Ludvigsen, C., 1994. An international comparison of long-term average speech spectra. J. Acoust. Soc. Am. 96 (4), 2108–2120.

Chakroborty, S., Roy, A., Saha, G., 2007. Improved closed set text-independent speaker identification by combining MFCC with evidence from flipped filter banks. Int. J. Signal Process. 4 (2), 114–122.

Degottex, G., Kane, J., Drugman, T., Raitio, T., Scherer, S., 2014. COVAREP – a collaborative voice analysis repository for speech technologies. In: Proc. ICASSP. Florence, Italy, pp. 960–964.

Dehak, N., Kenny, P., Dehak, R., Dumouchel, P., Ouellet, P., 2011. Front-end factor analysis for speaker verification. IEEE Trans. Audio Speech Lang. Process. 19 (4), 788–798.

Ergünay, S.K., Khoury, E., Lazaridis, A., Marcel, S., 2015. On the vulnerability of speaker verification to realistic voice spoofing. In: Proc. BTAS, pp. 1–6.

Evans, N.W.D., Kinnunen, T., Yamagishi, J., Wu, Z., Alegre, F., Leon, P.L.D., 2014. Speaker recognition anti-spoofing. In: Handbook of Biometric Anti-Spoofing - Trusted Biometrics under Spoofing Attacks, pp. 125–146.

Farrús, M., Wagner, M., Anguita, J., Hernando, J., 2008. How vulnerable are prosodic features to professional imitators? In: Proc. Odyssey, p. 2.

Galka, J., Grzywacz, M., Samborski, R., 2015. Playback attack detection for text-dependent speaker verification over telephone channels. Speech Commun. 67, 143–153.

Garcia-Romero, D., Espy-Wilson, C.Y., 2011. Analysis of i-vector length normalization in speaker recognition systems. In: Proc. INTERSPEECH, pp. 249–252.

Grigoras, C., 2010. Statistical tools for multimedia forensics. In: Proc. Audio Engineering Society Conference: 39th International Conference: Audio Forensics: Practices and Challenges, pp. 27–32.

Han, K., Wang, Y., Wang, D., Woods, W.S., Merks, I., Zhang, T., 2015. Learning spectral mapping for speech dereverberation and denoising. IEEE/ACM Trans. Audio Speech Lang. Process. 23 (6), 982–992.

Hanilçi, C., Kinnunen, T., Sahidullah, M., 2015. Classifiers for synthetic speech detection: a comparison. In: Proc. INTERSPEECH, pp. 2057–2061.

Hatch, A.O., Kajarekar, S.S., Stolcke, A., 2006. Within-class covariance normalization for SVM-based speaker recognition. In: Proc. ICSLP.

Hautamäki, R.G., Kinnunen, T., Hautamäki, V., Leino, T., Laukkanen, A., 2013. I-vectors meet imitators: on vulnerability of speaker verification systems against voice mimicry. In: Proc. INTERSPEECH, pp. 930–934.

Hermansky, H., Morgan, N., 1994. RASTA processing of speech. IEEE Trans. Speech Audio Process. 2 (4), 578–589.

Jain, A.K., Ross, A., Pankanti, S., 2006. Biometrics: a tool for information security. IEEE Trans. Inf. Forensics Sec. 1 (2), 125–143.

Jin, Q., Toth, A.R., Black, A.W., Schultz, T., 2008. Is voice transformation a threat to speaker identification? In: Proc. ICASSP, pp. 4845–4848.

Kawahara, H., Masuda-Katsuse, I., de Cheveigné, A., 1999. Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: possible role of a repetitive structure in sounds. Speech Commun. 27 (3–4), 187–207.

Kenny, P., 2010. Bayesian speaker verification with heavy-tailed priors. In: Proc. Odyssey, p. 14.

Khoury, E., Kinnunen, T., Sizov, A., Wu, Z., Marcel, S., 2014. Introducing i-vectors for joint anti-spoofing and speaker verification. In: Proc. INTERSPEECH, pp. 61–65.

Kinnunen, T., Li, H., 2010. An overview of text-independent speaker recognition: from features to supervectors. Speech Commun. 52 (1), 12–40.

Kons, Z., Aronowitz, H., 2013. Voice transformation-based spoofing of text-dependent speaker verification systems. In: Proc. INTERSPEECH, pp. 945–949.

Krishnamurthy, N., Hansen, J.H.L., 2009. Babble noise: modeling, analysis, and applications. IEEE Trans Audio Speech Lang. Process. 17 (7), 1394–1407.

Kua, J.M.K., Thiruvaran, T., Nosratighods, M., Ambikairajah, E., Epps, J., 2010. Investigation of spectral centroid magnitude and frequency for speaker recognition. In: Proc. Odyssey, p. 7.

Leon, P.L.D., Apsingekar, V.R., Pucher, M., Yamagishi, J., 2010a. Revisiting the security of speaker verification systems against imposture using synthetic speech. In: Proc. ICASSP, pp. 1798–1801.

Leon, P.L.D., Hernez, I., Saratxaga, I., Pucher, M., Yamagishi, J., 2011. Detection of synthetic speech for the problem of imposture. In: Proc. ICASSP. IEEE, pp. 4844–4847.

Leon, P.L.D., Pucher, M., Yamagishi, J., 2010b. Evaluation of the vulnerability of speaker verification to synthetic speech. In: Proc. Odyssey, p. 28.

Leon, P.L.D., Pucher, M., Yamagishi, J., Hernáez, I., Saratxaga, I., 2012. Evaluation of speaker verification security and detection of HMM-based synthetic speech. IEEE Trans. Audio Speech Lang. Process. 20 (8), 2280–2290.

Li, L., Wang, D., Zhang, C., Zheng, T.F., 2016. Improving short utterance speaker recognition by modeling speech unit classes. IEEE/ACM Trans. Audio Speech Lang. Process. PP (99). doi:10.1109/TASLP.2016.2544660. 1–1.

Lim, J.S., Oppenheim, A.V., 1979. Enhancement and bandwidth compression of noisy speech. Proc. IEEE 67 (12), 1586–1604.

Linville, S.E., Rens, J., 2001. Vocal tract resonance analysis of aging voice using long-term average spectra.. J. Voice 15 (3), 323–330.

Loizou, P.C., 2007. Speech Enhancement: Theory and Practice, first ed. CRC Press, Inc.
Masuko, T., Hitotsumatsu, T., Tokuda, K., Kobyashi, T., 1999. On the security of HMM-based speaker verification systems against imposture using synthetic speech. In: Proc. Eurospeech.
Matrouf, D., Bonastre, J., Fredouille, C., 2006. Effect of speech transformation on impostor acceptance. In: Proc. ICASSP, pp. 933–936.
Mitra, V., Wang, W., Franco, H., Lei, Y., Bartels, C., Graciarena, M., 2014. Evaluating robust features on deep neural networks for speech recognition in noisy and channel mismatched conditions. In: Proc. INTERSPEECH, pp. 895–899.
Murthy, H., Gadde, V., 2003. The modified group delay function and its application to phoneme recognition. In: Proc. ICASSP, 1, pp. I–68–71 vol.1.
Nakagawa, S., Wang, L., Ohtsuka, S., 2012. Speaker identification and verification by combining mfcc and phase information. IEEE Trans. Audio Speech Lang. Process. 20 (4), 1085–1095.
Novoselov, S., Kozlov, A., Lavrentyeva, G., Simonchik, K., Shchemelinin, V., 2015. STC anti-spoofing systems for the ASVspoof 2015 challenge. http://arxiv.org/ftp/arxiv/papers/1507/1507.08074.pdf.
Patel, T., Patil, H., 2016. Effectiveness of fundamental frequency (F0) and strength of exictation (SOE) for spoofed speech detection. In: Proc. ICASSP, pp. 5105–5109.
Pellom, B.L., Hansen, J.H.L., 1999. An experimental study of speaker verification sensitivity to computer voice-altered imposters. In: Proc. ICASSP, pp. 837–840.
Prince, S.J.D., Elder, J.H., 2007. Probabilistic linear discriminant analysis for inferences about identity. In: Proc. ICCV, pp. 1–8.
Rabiner, L.R., Cheng, M.J., Rosenberg, A.E., McGonegal, C.A., 1976. A comparative performance study of several pitch detection algorithms. IEEE Trans. Acoust. Speech Signal Process. 24 (5), 399–418.
Ratha, N.K., Connell, J.H., Bolle, R.M., 2001. Enhancing security and privacy in biometrics-based authentication systems. IBM Syst. J. 40 (3), 614–634.
Reynolds, D.A., Rose, R.C., 1995. Robust text-independent speaker identification using Gaussian mixture speaker models. IEEE Trans. Speech Audio Process. 3 (1), 72–83.
Sadjadi, S.O., Boril, H., Hansen, J.H.L., 2012. A comparison of front-end compensation strategies for robust LVCSR under room reverberation and increased vocal effort. In: Proc. ICASSP, pp. 4701–4704.
Sadjadi, S.O., Hansen, J.H., 2015. Mean Hilbert envelope coefficients (MHEC) for robust speaker and language identification. Speech Commun. 72, 138–148.
Sahidullah, M., Kinnunen, T., Hanilçi, C., 2015. A comparison of features for synthetic speech detection. In: Proc. INTERSPEECH, pp. 2087–2091.
Sanchez, J., Saratxaga, I., Hernaez, I., Navas, E., Erro, D., 2015. The AHOLAB RPS SSD spoofing challenge 2015 submission. In: Proc. INTERSPEECH, pp. 2042–2046.
Sánchez, J., Saratxaga, I., Hernáez, I., Navas, E., Erro, D., Raitio, T., 2015. Toward a universal synthetic speech spoofing detection using phase information. IEEE Trans. Inf. Forensics Secur. 10 (4), 810–820.
Sizov, A., Khoury, E., Kinnunen, T., Wu, Z., Marcel, S., 2015. Joint speaker verification and anti-spoofing in the i-vector space. IEEE Trans. Inf. Forensics Secur. (99).
Slaney, M., 1998. Auditory Toolbox (version 2). Interval Research Corporation Technical Report #1998-10.
SPTK: Speech signal processing toolkit. 2014. Version 3.8, http://sp-tk.sourceforge.net/.
Sun, M., Zhang, X., hamme, H.V., Zheng, T.F., 2016. Unseen noise estimation using separable deep auto encoder for speech enhancement. IEEE/ACM Trans. Audio Speech Lang. Process. 24 (1), 93–104.
Sündermann, D., Höge, H., Bonafonte, A., Ney, H., Black, A.W., Narayanan, S.S., 2006. Text-independent voice conversion based on unit selection. In: Proc. ICASSP, pp. 81–84.
Tian, X., Wu, Z., Xiao, X., Chng, E. S., Li, H., 2016. Spoofing detection under noisy conditions: a preliminary investigation and an initial database. http://arxiv.org/pdf/1602.02950v1.pdf.
Toda, J., Ohtani, Y., Shikano, K., 2006. Eigenvoice conversion based on Gaussian mixture model. In: Proc. INTERSPEECH.
Todisco, M., Delgado, H., Evans, N., 2016. A new feature for automatic speaker verification anti-spoofing: Constant Q cepstral coefficients. In: Proc. Odyssey.
Varga, A., Steeneken, H.J.M., 1993. Assessment for automatic speech recognition ii: NOISEX-92: a database and an experiment to study the effect of additive noise on speech recognition systems. Speech Commun. 12 (3), 247–251.
Villalba, J.A., Lleida, E., 2010. Speaker verification performance degradation against spoofing and tampering attacks. In: Proc. FALA, pp. 131–134.
Villalba, J.A., Miguel, A., Ortega, A., Lleida, E., 2015. Spoofing detection with DNN and one-class SVM for the ASVspoof 2015 challenge. In: Proc. INTERSPEECH, pp. 2067–2071.
Wang, L., Yoshida, Y., Kawakami, Y., Nakagawa, S., 2015. Relative phase information for detecting human speech and spoofed speech. In: Proc. INTERSPEECH, pp. 2092–2096.
Wester, M., Wu, Z., Yamagishi, J., 2015. Human vs machine spoofing detection on wideband and narrowband data. In: Proc. INTERSPEECH, pp. 2047–2051.
Wall Street Journal Corpus. 2015. [Online:] http://www.ldc.upenn.edu.
Wu, Z., Evans, N., Kinnunen, T., Yamagishi, J., Alegre, F., Li, H., 2015a. Spoofing and countermeasures for speaker verification: a survey. Speech Commun. 66, 130–153.
Wu, Z., Khodabakhsh, A., Demiroğlu, C., Yamagishi, J., Saito, D., Toda, T., King, S., 2015b. SAS: A speaker verification spoofing database containing diverse attacks. In: Proc. ICASSP, pp. 4440–4444.
Wu, Z., Kinnunen, T., Chng, E.S., Li, H., 2012a. A study on spoofing attack in state-of-the-art speaker verification: the telephone speech case. In: Proc. APSIPA ASC, pp. 1–5.
Wu, Z., Kinnunen, T., Evans, N., Yamagishi, J., Hanilçi, C., Sahidullah, M., Sizov, A., 2015. ASVspoof 2015: The first automatic speaker verification spoofing and countermeasures challenge. In: Proc. INTERSPEECH, pp. 2037–2041.
Wu, Z., Li, H., 2014. Voice conversion versus speaker verification: an overview. APSIPA Trans. Audio Signal Inf. Process. 3 (e17).
Wu, Z., Siong, C.E., Li, H., 2012b. Detecting converted speech and natural speech for anti-spoofing attack in speaker recognition. In: Proc. INTERSPEECH.
Wu, Z., Xiao, X., Chng, E., Li, H., 2013. Synthetic speech detection using temporal modulation feature. In: Proc. ICASSP, pp. 7234–7238.
Xiao, X., Tian, X., Du, S., Xu, H., Chng, E.S., Li, H., 2015. Spoofing speech detection using high dimensional magnitude and phase features: The NTU approach for ASVspoof 2015 challenge. In: Proc. INTERSPEECH, pp. 2052–2056.
Xu, Y., Du, J., Dai, L.R., Lee, C.H., 2014. An experimental study on speech enhancement based on deep neural networks. IEEE Signal Process. Lett. 21 (1), 65–68.
Xu, Y., Du, J., Dai, L.R., Lee, C.H., 2015. A regression approach to speech enhancement based on deep neural networks. IEEE/ACM Trans. Audio Speech Lang. Process. 23 (1), 7–19.
Yamagishi, J., Kobayashi, T., Nakano, Y., Ogata, K., Isogai, J., 2009. Analysis of speaker adaptation algorithms for HMM-based speech synthesis and a constrained SMAPLR adaptation algorithm. Trans. Audio Speech Lang. Process. 17 (1), 66–83.
Yin, H., Hohmann, V., Nadeu, C., 2011. Acoustic features for speech recognition based on Gammatone filterbank and instantaneous frequency. Speech Commun. 53 (5), 707–715.
Yu, H., Sarkar, A., Thomsen, D.A.L., Tan, Z.H., Ma, Z., Guo, J., 2016. Effect of multi-condition training and speech enhancement methods on spoofing detection. In: 2016 First International Workshop on Sensing, Processing and Learning for Intelligent Machines (SPLINE), pp. 1–5.
Zhang, C., Ranjan, S., Nandwana, M., Zhang, Q., Misra, A., Liu, G., Kelly, F., Hansen, J., 2016. Joint information from nonlinear and linear features for spoofing detection: an i-vector/DNN based approach. In: Proc. ICASSP, pp. 5035–5039.