Integrated Presentation Attack Detection and Automatic Speaker Verification: Common Features and Gaussian Back-end Fusion

Massimiliano Todisco, Héctor Delgado, Kong Aik Lee, Md Sahidullah, Nicholas Evans, Tomi Kinnunen, Junichi Yamagishi

Integrated Presentation Attack Detection and Automatic Speaker Verification: Common Features and Gaussian Back-end Fusion

Massimiliano Todisco[1], *Héctor Delgado*[1], *Kong Aik Lee*[2], *Md Sahidullah*[3], *Nicholas Evans*[1], *Tomi Kinnunen*[4] *and Junichi Yamagishi*[5,6]

[1]Department of Digital Security, EURECOM, France
[2]Data Science Research Laboratories, NEC Corporation, Japan
[3]MULTISPEECH, Inria, France
[4]School of Computing, University of Eastern Finland, Finland
[5]Digital Content and Media Sciences Research Division, National Institute of Informatics, Japan
[6]Centre of Speech Technology Research, University of Edinburgh, U.K.

{todisco,delgado,evans}@eurecom.fr,k-lee@ax.jp.nec.com,
md.sahidullah@inria.fr, tkinnu@cs.uef.fi, jyamagis@nii.ac.jp

Abstract

The vulnerability of automatic speaker verification (ASV) systems to spoofing is widely acknowledged. Recent years have seen an intensification in research efforts to develop spoofing countermeasures, also known as presentation attack detection (PAD) systems. Much of this work has involved the exploration of features that discriminate reliably between bona fide and spoofed speech. While there are grounds to use different front-ends for ASV and PAD systems (they are different tasks) the use of a single front-end has obvious benefits, not least convenience and computational efficiency, especially when ASV and PAD are combined. This paper investigates the performance of a variety of different features used previously for both ASV and PAD and assesses their performance when combined for both tasks. The paper also presents a Gaussian back-end fusion approach to system combination. In contrast to cascaded architectures, it relies upon the modelling of the two-dimensional score distribution stemming from the combination of ASV and PAD in parallel. This approach to combination is shown to generalise particularly well across independent ASVspoof 2017 v2.0 development and evaluation datasets.

Index Terms: automatic speaker verification, spoofing, countermeasures, presentation attack detection

1. Introduction

Presentation attack detection (PAD) systems capable of detecting and deflecting so-called spoofing attacks, or presentation attack (PA) in ISO/IEC 30107[1] nomenclature, leveled at automatic speaker verification (ASV) systems have been under development for a number of years. While ASV systems aim to verify the identity claimed by a speaker, PAD systems aim to verify the authenticity of the speech signal itself, namely whether it is bona fide speech or whether, instead, it is artificially created or somehow manipulated, i.e. *spoofed*.

While early PAD systems used features similar to those used for ASV, being distinctly different tasks, most efforts to develop effective PAD systems have focused on the design of new features tailored to discriminate between bona fide and spoofed speech. While the use of features designed specifically for PAD have been shown to give better performance than systems that use features designed for ASV, the use of different front-ends augments computational complexity.

It can hence be convenient to use a single front-end. The use of such a single front-end avoids redundant processing and can also simplify the combination of ASV and PAD decisions. The search for features which perform well for a combined ASV and PAD task is the subject of this paper.

A second contribution relates to the manner in which ASV and PAD systems scores can be combined. It extends previous work [1] which proposed cascade and parallel approaches to system combination and is similar in nature to the combination architecture reported in [2]. New to this paper is a two-dimensional score modelling technique which avoids the joint optimisation of separate ASV and PAD decision thresholds. The explicit modelling of target and impostor trial scores encompassing genuine, bona fide trials in addition to both zero-effort and spoofed impostor trials provides for greater flexibility in decision boundaries and hence more reliable decisions. The merits of these two contributions are assessed through experiments with the ASVspoof 2017 database of bona fide and spoofed speech signals and protocols for the assessment of combined ASV and PAD systems.

The remainder of the paper is organised as follows. Section 2 describes the different front-ends used in this work. The approach to system combination is presented in Section 3. Experiments are reported in Section 4 whereas results are reported in Section 5. Conclusions are presented in Section 6.

2. Front-end processing

This paper aims to determine a common front-end for both ASV and PAD tasks. While ASV calls for features that capture speaker-discriminant information, PAD systems rely on features that capture the tell-tale signs of spoofing. The study includes four different front-ends, each of which is described here.

Mel-frequency cepstral coefficients (MFCCs): MFCCs are used widely in speech and speaker recognition and have been explored extensively as features for spoofing detection [3]. MFCCs are usually derived from short-time Fourier transform (STFT) decompositions, the application of perceptually motivated Mel-frequency scaled filterbank [4] and standard cepstral analysis.

[1]https://www.iso.org/standard/67381.html

Table 1: *Classification of trials in PAD and ASV tasks. By "–" we assume that ASV has zero or no capability to reject spoofing imposter trials, similarly for PAD that cannot differentiate zero-effort imposter and target trials. Alternatively "–" means* arbitrary*: since the last row corresponds to the logical* AND *operator to combine the two systems, choosing either +1 or -1 in the place of "–" gives rise to the same trial classifications for the integrated system.*

class	C_1	C_2	C_3
system/trial	target	zero-effort nontarget	spoof nontarget
PAD	+1	–	-1
ASV	+1	-1	–
ASV + PAD	+1	-1	-1

Linear frequency cepstral coefficients (LFCCs): LFCCs are similar to MFCC except for the use of a linear-scaled in place of a Mel-scaled filterbank, thereby giving a constant spectral resolution. LFCCs have also been applied to both speech and speaker recognition, in addition to spoofing detection [3].

Infinite impulse response constant Q Mel cepstral coefficients (ICMCs): ICMCs have been applied successfully to ASV [5], utterance verification (UV) [5] and speaker diarization [6]. Features are based upon the perceptually-motivated infinite impulse response constant Q transform (IIR-CQT) approach to spectro-temporal decomposition [7]. In contrast to the STFT, the spectral resolution has a constant Q factor which reflects filter selectivity, defined as the ratio between the centre frequency and bandwidth. Efficient feature extraction is obtained from the IIR filtering of the fast Fourier transform (FFT) giving a variable-resolution decomposition with greater frequency resolution at low frequencies and greater time resolution at higher frequencies. ICMCs are then obtained from Mel-scaling and standard cepstral analysis.

Constant Q cepstral coefficients (CQCCs): CQCC features were designed specifically for spoofing detection [8, 9] and applied subsequently to ASV [10]. CQCCs rely on the same CQT approach used in ICMCs but are extracted according to the more computationally demanding approach described in [11, 12]. Resampling [9] is applied to warp the geometric scale of the CQT to the linear scale of the discrete Cosine transform (DCT).

3. Integration of PAD with ASV

The integration of ASV and PAD can be achieved at the model/feature level [13] or at the score level [1]. This paper focuses on the latter. Dedicated classifiers are developed for ASV and PAD, and scores produced by each system are combined (*i.e.*, late fusion). Even at this score level, there are different approaches to combination including both cascaded and parallel combinations [1]. We describe below the cascade approach and then the proposed Gaussian back-end fusion approach.

3.1. Task and trial definitions

Whereas ASV and PAD systems are both binary classifiers, they tackle different tasks. Table 1 defines the three different types of trial that ASV and PAD systems may encounter: (1) *target*, (2) *zero-effort nontarget* and (3) *spoof nontarget* trials. Also illustrated in Table 1 are the ground-truth labels for each task

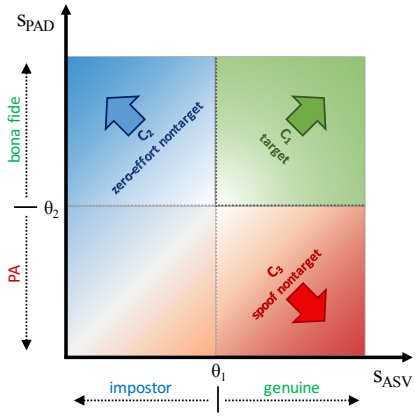

Figure 1: *A two-dimensional view of the score space formed by PAD and ASV scores.*

and trial combination. PAD systems aim to distinguish bona fide speech from spoofed speech, while ASV systems aim to verify a claimed identity.

This same idea is illustrated in the two-dimensional score space of Fig. 1. For a threshold θ_2 applied to PAD scores s_{PAD}, all trials for which the score is greater than the threshold, $s_{\text{PAD}} > \theta_2$, are classified as bona fide speech. Those for which the score is below the threshold are classified as presentation attack speech. Similarly, for a threshold θ_1 applied to ASV scores s_{ASV}, all trials for which the score is greater than the threshold, $s_{\text{ASV}} > \theta_1$, are classified as genuine speaker trials. Those for which the score is below the threshold are classified as impostor trials. The green-colored region to the upper-right-hand corner of Fig. 1, where $\{s_{\text{PAD}} > \theta_2\} \cap \{s_{\text{ASV}} > \theta_1\}$, corresponds to target trials (class C_1). Class C_2 (zero-effort nontarget) is represented by the blue-colored region, which is the resultant of the impostor and bona fide axes. Similarly, class C_3 (spoof nontarget) is represented by the red-colored region, which is the resultant of the target and spoofed axes. As shown in Table 1, only target trials should be positively verified. Both forms of nontarget trial should be rejected.

3.2. Cascaded/tandem combination

ASV and PAD systems can be cascaded in either order – PAD followed by ASV, or ASV followed by PAD. ASV and PAD systems can be optimised independently or jointly, *e.g.* considering an architecture whereby PAD precedes ASV, then the PAD threshold θ_2 can be optimised, for instance, to minimise the *equal error rate* (EER) of the ASV system. In order to estimate the performance of the integrated system, trials classified as spoofs are assigned arbitrarily $-\infty$ scores and are thereby rejected automatically by the ASV system that follows.

The cascaded approach relies on two thresholds, θ_2 and θ_1, applied to PAD and ASV scores, respectively. The cascading of ASV and PAD in such a way is equivalent to the partitioning of the score space into rigid decision regions with the vertical and horizontal decision boundaries being those illustrated in Fig. 1.

3.3. Gaussian back-end fusion

In contrast to the cascaded approach to combination involving the optimisation of two different thresholds for PAD and ASV, the parallel approach to combination requires the optimisation of only a single threshold. Recall the $N = 3$ classes of trial

illustrated in Table 1, namely target (C_1), zero-effort nontarget (C_2) and spoof nontarget (C_3). Let $\mathbf{s} = [s_{\text{PAD}}, s_{\text{ASV}}]^{\text{T}} \in \mathbb{R}^2$ be a two-dimensional score vector of PAD and ASV scores corresponding to a single ASV trial.

We treat $\boldsymbol{s}$ as a 2D feature vector and model the class-conditional probability density of $\boldsymbol{s}$ using a Gaussian

$$p(\mathbf{s}|C_l) = \mathcal{N}(\mathbf{s}|\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l),$$

where $\boldsymbol{\mu}_l \in \mathbb{R}^2$ and $\boldsymbol{\Sigma}_l \in \mathbb{R}^{2\times 2}$ are the mean vector and the covariance matrix corresponding to class C_l where $l \in \{1, 2, 3\}$. For each trial t, a fused score is computed as the log-likelihood ratio according to:

$$\begin{aligned} \tilde{s}_t &= \log \frac{p(\boldsymbol{s}_t|\text{H}_0)}{p(\boldsymbol{s}_t|\text{H}_1)} \\ &= \log \frac{p(\mathbf{s}_t|C_1)}{\alpha \times p(\mathbf{s}_t|C_2) + (1-\alpha) \times p(\mathbf{s}_t|C_3)}, \end{aligned} \tag{1}$$

where the null hypothesis H_0 represents the likelihood that $\boldsymbol{s}_t$ is a target speaker trial (from class C_1) while the alternative hypothesis H_1 signifies that $\boldsymbol{s}_t$ belongs to either C_2 or C_3. Therefore, (1) converts the 2D detection score vector $\boldsymbol{s}_t$ into a log-likelihood ratio, a scalar where higher relative values are associated with a stronger support for the null hypothesis. Classification decisions corresponding to the last line of Table 1 may then be made upon the application of a single threshold θ to $\tilde{s}_t$.

Note that the alternative hypothesis likelihood in the denominator of (1) represents, in fact, a 2-component GMM with mixing weight α. The hyper-parameter $\alpha \in [0, 1]$, determines the weight of C_2 and C_3 in the denominator of Eq. 1 (the alternative hypothesis). Parameters μ_l, Σ_l and threshold θ can be learned from development data.

3.4. Modeling capacity of the two integration approaches

The cascaded approach has two parameters that require optimization — the thresholds θ_1 and θ_2. The Gaussian back-end approach, in turn, has $3 \times 2 = 6$ parameters to specify the three mean vectors, $3 \times 3 = 9$ parameters to specify the three covariance matrices (they are symmetric so 3 parameters are sufficient), one combination parameter α, plus the final decision threshold, *i.e.* a total of $6 + 9 + 1 = 16$ parameters.

While the number of parameters in the Gaussian back-end fusion approach is greater than that for the cascaded approach, generative score modelling offer better potential for generalization. In specific, the decision regions in the cascaded approach have a limited modeling capacity: as illustrated in Fig. 1, the decision region for the target class is an infinite rectangular region, with the lower left corner at (θ_1, θ_2). Since the joint distribution of the PAD and ASV scores is certainly not defined by such regions, the cascaded approach has limited modeling power. In the Gaussian back-end fusion, in turn, we use three Gaussians with arbitrary (non-shared) covariance matrices, which yields generally a more complex nonlinear decision boundary [14] and makes the model better adaptable to arbitrary scores.

Similar 2D back-end fusions have been used earlier in the context of general biometrics and joint operation of PAD and biometric modalities, *e.g.* [2] and [15]. The authors of [2, 15] used a *linear* classifier whose parameters (i.e. slope and intercept) were trained using a logistic loss.

4. Experimental setup

This section describes the database used in this work with implementation details for the individual PAD and ASV systems.

4.1. ASVspoof 2017 v2.0 database and protocols

Experiments relate to the ASVspoof 2017 v2.0 database [16] of bona-fide and replayed, spoofed short utterances of about 1 to 5 seconds each. PAD systems are trained using the official training partition containing 1507 bona fide and 1507 replayed speech segments. Joint PAD and ASV experiments were performed using a specifically designed protocol encompassing target segments and both zero-effort and spoofed speech segments[2]. The number of speakers and trials in the development and evaluation sets are illustrated in Table 3.

4.2. Front-ends

Experiments were conducted using the four front-ends described in Section 2 using pre- and post-processing comprising the addition of log-energy parameters, cepstral mean and variance normalization (CMVN) [17], relative spectral (RASTA) filtering [18] and articulation rate (ARTE) filtering [10]. Dynamic coefficients up to double deltas are also considered.

The configuration of MFCCs and LFCCs is standard: 19 (S)tatic coefficients (excluding the 0-th), RASTA filtering with appended (D)elta and (A)cceleration coefficients. The ICMC configuration is that reported in [19] and is the same as for MFCCs. Based on configurations used previously for text-dependent ASV [10] and PAD [16], CQCCs includes 29 S coefficients with appended D coefficients, ARTE filtering and log-energy coefficients. None of the experiments reported here use speech activity detection (SAD). Any single experiment reported in this paper involve PAD and ASV systems that use the exact same front-end configurations.

4.3. PAD and ASV systems

Both the PAD and the ASV classifiers are conventional Gaussian mixture models (GMMs). The PAD classifier uses models of 512 components. Models are learned for bona fide and spoofed speech with an expectation-maximisation (EM) algorithm with random initialisation. Classifier scores for a given test utterance are computed as the log-likelihood ratio between the GMMs for bona fide and spoofed speech. The ASV classifier also uses models of 512 components and learns speaker specific models from the maximum a posteriori (MAP) adaptation of a universal background model (UBM) trained on the RSR2015 database [20]. Scores are the log-likelihood ratio given the target model and the UBM.

4.4. Integration of PAD and ASV

Concerning the Gaussian back-end fusion described in Section 3.3, we use maximum likelihood to obtain the means and covariances of all the three classes (C_1, C_2 and C_3). The value of α is set empirically to 0.96 using a grid search on the development set.

5. Experimental results and discussion

Results are presented in Table 2 for development (D) and evaluation (E) partitions of the joint PAD-ASV protocol (see Table 3). Results are presented for each front-end and for the cascaded combination and the proposed Gaussian backend fusion (bottom part of Table 2). The performance of two alternative methods, namely linear regression (LR) and polynomial linear

[2]Note to the Interspeech reviewers: this custom protocol will be made public later on, to ensure reproducibility.

Table 2: *Speaker verification performance in terms of EER using linear regression fusion, polynomial logistic regression fusion, cascade/tandem combination and proposed Gaussian back-end fusion of PAD and ASV scores for the ASVspoof 2017 v2.0 database.* ***D:*** *development set,* ***E****: evaluation set. The best average results for development set are shown in boldface.*

impostor type	zero-effort		spoof		average		zero-effort		spoof		average	
feat. config / tested on	D	E	D	E	D	E	D	E	D	E	D	E
	Logistic regression fusion [2]						**Polynomial logistic regression fusion [2]**					
MFCC	3.78	2.42	42.72	31.02	23.25	16.72	3.86	2.50	43.81	35.14	23.84	18.82
LFCC	5.72	2.11	46.41	35.71	26.06	18.91	5.47	2.20	37.64	26.99	21.55	14.60
ICMC	2.67	2.16	43.60	33.59	23.14	17.88	2.60	2.08	37.58	29.31	**20.09**	**15.69**
CQCC	6.02	3.52	38.76	33.17	**22.39**	**18.34**	6.02	7.93	42.67	47.96	24.34	27.94
	Cascaded/tandem combination						**Gaussian back-end fusion**					
MFCC	5.19	5.36	23.07	24.65	14.13	15.00	3.99	3.26	21.02	24.35	12.51	13.81
LFCC	7.28	4.96	22.41	21.28	14.84	13.12	5.71	2.90	21.26	17.98	13.48	10.43
ICMC	4.30	4.92	27.08	27.82	15.69	16.37	3.06	3.71	20.90	22.51	11.98	13.11
CQCC	7.31	8.30	15.71	25.26	**11.51**	**16.78**	6.04	4.71	17.34	18.11	**11.69**	**11.41**

Table 3: *Statistics of the ASVspoof 2017 joint PAD+ASV protocol.*

	#spk	target	zero-effort	spoof
Dev.	8	742	5186	940
Eval.	24	1106	18624	10878

regression (PLR) fusion approaches [2], is also reported (top part of Table 2). These contrastive methods also aim to split the 2D score space into two classes. Results are also presented separately for target trials combined with zero-effort and spoofed impostor trials, and the average.

LFCC, ICMC and CQCC features perform marginally better than MFCC features. LFCC features generalise better across development and evaluation subsets, giving the best average (zero-effort and spoofing impostor) results of 13% and 10% EER for each approach to combination. This observation shows that other features that give better performance in terms of spoofing detection do not necessarily give the best performance when ASV and PAD are combined. This is probably due to use of single features which avoids redundant processing and simplifies the combination of ASV-PAD decisions.

These same results also show that the Gaussian back-end fusion approach proposed in this paper outperforms the cascade/tandem combination and the LR and PLR approaches. The improvement in performance is attributed to use of a single, flexible and jointly-optimised, instead of independently optimised, rigid thresholds. The former gives better capacity to reject spoofed trials with less impact on the rejection of target trials. The Gaussian back-end fusion approach reported in this paper therefore offers better robustness to spoofing and better usability. Results for LR and PLR fusion approaches are globally worse with respect to the other two approaches. Moreover, they exhibit similar performance for development and evaluation sets, but with a lack of generalisation.

Finally, Fig. 2 shows the 2D score space representation for the LFCC features for development and evaluation set. By analysing the data distribution, it is clear that a decision can not be taken using approaches with rigid thresholds optimised on development set.

Figure 2: *Two-dimensional PAD-ASV score representation corresponding to LFCC features for development (left) and evaluation (right) set. Green dots represent target trials, while blue stars represent zero-effort trials and red crosses are spoof trials.*

6. Conclusions

This paper presents a comparative study of the performance of different front-ends when used for both automatic speaker verification (ASV) and presentation attack detection (PAD). Performance is assessed with difference approaches to system integration including cascaded combination, linear and polynomial logistic regression and a Gaussian back-end fusion approach. The use of a single front-end for both ASV and PAD systems simplifies integration in terms of convenience and efficiency; computational effort is reduced by avoiding redundant processing.

Performance is assessed using the ASVspoof 2017 v2.0 database. Results show that feature that achieve the best results for independent ASV or PAD tasks do not give the best performance when systems are combined. The cascaded approach to ASV and PAD combination, as well as the logistic and polynomial logistic regression approach, improve reliability in the case where the nature of non-target trials is known. When faced with unknown, or previously unseen forms of non-target trials and spoofing attacks, then performance degrades significantly; these approaches to ASV and PAD combination fail to generalise. In contrast, the Gaussian back-end approach to integration is shown to generalise well and gives the lowest equal error rate for the independent evaluation set.

7. References

[1] M. Sahidullah, H. Delgado, M. Todisco, H. Yu, T. Kinnunen, N. Evans, and Z.-H. Tan, "Integrated spoofing countermeasures and automatic speaker verification: an evaluation on ASVspoof 2015," in *INTERSPEECH 2016, Annual Conference of the International Speech Communication Association, September 8-12, 2016, San Francisco, USA*, San Francisco, Sept. 2016.

[2] I. Chingovska, A. Anjos, and S. Marcel, "Anti-spoofing in action: Joint operation with a verification system," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2013, Portland, OR, USA, June 23-28, 2013*, 2013, pp. 98–104. [Online]. Available: https://doi.org/10.1109/CVPRW.2013.22

[3] M. Sahidullah, T. Kinnunen, and C. Hanilçi, "A comparison of features for synthetic speech detection," in *INTERSPEECH*, 2015, pp. 2087–2091.

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

[5] H. Delgado, M. Todisco, M. Sahidullah, A. K. Sarkar, N. Evans, T. Kinnunen, and Z.-H. Tan, "Further optimisations of constant Q cepstral processing for integrated utterance verification and text-dependent speaker verification," in *SLT 2016, IEEE Workshop on Spoken Language Technology*, San Diego, Dec. 2016.

[6] J. Patino, H. Delgado, N. Evans, and X. Anguera, "EURECOM submission to the Albayzin 2016 Speaker Diarization Evaluation," in *Proc. IberSPEECH*, 2016.

[7] P. Cancela, M. Rocamora, and E. López, "An efficient multi-resolution spectral transform for music analysis," in *Proceedings of ISMIR*, 2009, pp. 309–314.

[8] M. Todisco, H. Delgado, and N. Evans, "A new feature for automatic speaker verification anti-spoofing: Constant Q cepstral coefficients," in *Proc. Odysssey*, Bilbao, Spain, 2016.

[9] M. Todisco, H. Delgado, and N. Evans, "Constant Q cepstral coefficients: A spoofing countermeasure for automatic speaker verification," *Computer Speech & Language*, 2017.

[10] M. Todisco, H. Delgado, and N. Evans, "Articulation rate filtering of CQCC features for automatic speaker verification," in *Proc. INTERSPEECH*, 2016.

[11] J. Brown, "Calculation of a constant Q spectral transform," *Journal of the Acoustical Society of America*, vol. 89, no. 1, pp. 425–434, January 1991.

[12] J. Brown and M. S. Puckette, "An efficient algorithm for the calculation of a constant q transform," *The Journal of the Acoustical Society of America*, vol. 92, no. 5, pp. 2698–2701, 1992.

[13] A. Sizov, E. Khoury, T. Kinnunen *et al.*, "Joint speaker verification and antispoofing in the i-vector space," *IEEE Trans. on Information Forensics and Security*, vol. 10, no. 4, pp. 821–832, 2015.

[14] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, 2nd ed., New York, 2000.

[15] P. Korshunov and S. Marcel, "Joint operation of voice biometrics and presentation attack detection," in *8th IEEE International Conference on Biometrics Theory, Applications and Systems, BTAS 2016, Niagara Falls, NY, USA, September 6-9, 2016*, 2016, pp. 1–6. [Online]. Available: https://doi.org/10.1109/BTAS.2016.7791179

[16] H. Delgado, M. Todisco, M. Sahidullah, N. Evans, T. Kinnunen, K. A. Lee, and J. Yamagishi, "ASVspoof 2017 Version 2.0: meta-data analysis and baseline enhancements," in *Proc. Odysssey*, Les Sables D'Olonne, France, 2018. [Online]. Available: http://www.asvspoof.org/data2017/ASVspoof2017_V2_Odyssey_2018.pdf

[17] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas, "Score normalization for text-independent speaker verification systems," *Digit. Signal Process.*, vol. 10, no. 1, pp. 42–54, Jan. 2000. [Online]. Available: http://dx.doi.org/10.1006/dspr.1999.0360

[18] D. Ellis, "PLP and RASTA (and MFCC, and inversion) in MATLAB," http://www.ee.columbia.edu/~dpwe/resources/matlab/rastamat/, accessed: 2016-03-22.

[19] H. Delgado, M. Todisco, M. Sahidullah *et al.*, "Further optimisations of constant q cepstral processing for integrated utterance and text-dependent speaker verification," in *2016 IEEE Spoken Language Technology Workshop (SLT)*, 2016, pp. 179–185.

[20] A. Larcher, K. A. Lee, B. Ma, and H. Li, "Text-dependent Speaker Verification: Classifiers, Databases and RSR2015," vol. 60, pp. 56–77, 2014.