Spoofing Detection on the ASVspoof2015 Challenge Corpus Employing Deep Neural Networks

Md Jahangir Alam, Patrick Kenny, Vishwa Gupta, Themos Stafylakis

Computer research Institute of Montreal (CRIM)
Montreal, Quebec, Canada
{jahangir.alam, vishwa.gupta, patrick.kenny, themos.stafylakis}@crim.ca

Abstract

This paper describes the application of deep neural networks (DNN), trained to discriminate between human and spoofed speech signals, to improve the performance of spoofing detection. In this work we use amplitude, phase, linear prediction residual, and combined amplitude - phase-based acoustic level features. First we train a DNN on the spoofing challenge training data to discriminate between human and spoofed speech signals. Delta filterbank spectra (DFB), delta plus double delta Mel-frequency cepstral coefficients (DMCC), delta plus double delta linear prediction cepstral coefficients (DLPCC) and product spectrum-based cepstral coefficients (DPSCC) features are used as inputs to the DNN. For each feature, posteriors and bottleneck features (BNF) are then generated for all the spoofing challenge data using the trained DNN. The DNN posteriors are directly used to decide if a test recording is spoofed or human. For spoofing detection with the acoustic level features and the bottleneck features we build a standard Gaussian Mixture Model (GMM) classifier. When tested on the spoofing attacks (S1-S10) of ASVspoof2015 challenge evaluation corpus, DFB-BNF, DMCC-BNF, DLPCC-BNF, DPSCC-BNF and DPSCC-DNN systems provided equal error rates (EERs) of 0.013%, 0.007%, 0.0%, 0.022%, and 1.00% respectively, on the S1-S9 spoofing attacks. On the all ten spoofing attacks (S1-S10) the EERs obtained by these five systems are 3.23%, 2.15%, 3.3%, 3.28 and 2.18%, respectively.

1. Introduction

A spoofing attack is a situation in which one person or program successfully impersonates a legitimate user. Since spoofing attacks are easy to implement it is the greatest threat to speaker verification systems [1, 3, 31]. Some examples of spoofing attacks are: impersonation, replay, voice conversion and speech synthesis. Among these the last two attacks have gained a lot of research attentions due to the availability of many online open-source toolkits. The automatic speaker verification spoofing and countermeasures challenge 2015 (ASVspoof2015) provides a common framework for the evaluation of spoofing countermeasures or anti-spoofing techniques in the presence of various seen and unseen spoofing attacks [1]. This challenge focused on a spoofing detection task which mainly includes voice converted and speech synthesis attacks [1]. Various frameworks with different types of countermeasures were proposed in the first ASVspoof2015 challenge [2-12].

Most of the successful spoofing countermeasures reported in the literature are based on phase [28-31]. In ASVspoof2015 amplitude-, phase- and joint amplitude - phase-based features [2-12] provided comparable spoofing detection performance [2].

Inspired by the impressive gains in performance obtained by the deep neural networks (DNN) for speech, speaker and language recognition applications, several participants [3, 9, 12] in the ASVspoof2015 challenge incorporated DNN feature representations and posterior probabilities for the detection of spoofing attacks.

The bottleneck features supplied by a DNN are increasingly being used for reducing the recognition errors in speech related applications [13-17, 32]. The bottleneck here refers to a linear hidden layer with relatively small number of nodes placed in between the input and output layers of a DNN. Bottleneck features can be seen as a compact low-dimensional representation of inputs which contains discriminative information for classification [13].

DNN-based feature representations as a countermeasure for spoofing detection have been proposed in [3], [9], and [12]. In [3] a spoofing-discriminant neural network is trained and a *s* (spoofing) -vector is calculated for each utterance. Then a Mahalanobis distance measure with score normalization is applied to *s*-vectors for the detection of spoofing. In [9] both DNN posteriors and bottleneck feature are used for spoofing detection. Posterior probabilities are transformed into log likelihood ratios to obtain the scores of the system and a one-class SVM is employed with the DNN bottleneck feature to discriminate spoofed speech from human speech. Multiple countermeasures are used in [12] and for each system, an MLP is trained to predict the posterior probability of spoofing. Final scores were obtained by fusing the scores of all systems.

In the present work, we use a DNN to generate a bottleneck feature (BNF) representation and frame level posteriors for the spoofing detection task. The DNN front-ends use delta filterbank spectra (DFB), delta plus double delta Mel-frequency cepstral coefficients (DMCC), delta plus double delta linear prediction cepstral coefficients (DLPCC) and product spectrum-based cepstral coefficients (DPSCC) features. The frame level posterior probabilities are transformed into log likelihood ratio to obtain the scores of the system [9]. A GMM is trained on the human speech BNF and another on the spoofed speech BNF. For each test segment, the bottleneck features are used to calculate a log likelihood ratio for the human and spoofed speech hypotheses. We found that bottleneck features-based systems (DFB-BNF, DMCC-BNF, DPSCC-BNF, and DLPCC-BNF) provided improved spoofing detection performance when evaluated on the S1-S9 spoofing attacks. The DMCC-BNF system and DNN posteriors-based system (DPSCC-DNN) yielded improved spoofing detection performance when evaluated on all (S1-S10) spoofing attacks of ASVspoof2015 challenge evaluation data.

doi: 10.21437/Odyssey.2016-39

2. DNN-based Spoofing Detection

A DNN can be used as a classifier to estimate the posterior probability of a class given the input data or as a feature generator to extract relevant features, known as bottleneck features (BNF), for used by a classifier. Extraction of BNFs is done by placing a hidden layer, which has relatively small number of nodes compared to the size of other layers, in between the input and output layers [13-14]. In speech related applications these features are widely employed for improving recognition accuracy [13-17].

In order to discriminate between the human and spoofed speech signals we train a DNN on the spoofing challenge training data for each input feature. The input feature to the DNN are either delta filter-bank (DFB) or delta + double delta Mel-frequency cepstral coefficients (DMCC) or delta + double delta linear prediction cepstral coefficients (DLPCC) or delta + double delta product spectrum-based cepstral coefficients (DPSCC). The input layer consists of a sliding window with the current frame in the center and a context of 7 left and right frames. Only global mean and variance normalization is applied to the input features. The DNN has 5 hidden layers and the 5-th layer is the bottleneck layer. Each hidden layer has 1000 neurons and uses sigmoid activation with the exception of 5-th layer which is linear and has 64 nodes. The output layer, which is the classification layer, is a softmax of dimension 2 i.e., one output for human speech signal and one for spoof signal by considering the five spoofing attacks in the challenge training data as one class. With the DFB features the final frame accuracy on the validation set was 91.9% and for the DLPCC and DPSCC features they were 94.03% and 90.0%, respectively.

With BNF feature a Gaussian Mixture Models (GMM) is used as classifier for spoofing detection whereas with DNN posteriors the log likelihood ratio is obtained by transforming the posterior probabilities given by a DNN as follows [9]:

$$LLR = \log\left(p\left(\text{human}\,|O\right)\right) - \log\left(p\left(\text{spoof}\,|O\right)\right).$$

3. Acoustic Features for Spoofing Detection

Several amplitude-, phase-, and combined amplitude - phase-based are chosen for spoofing detection with a GMM and as input to a deep neural network for feature representation and classification. These features are briefly described below.

3.1. MFCC features

The Mel-frequency cepstral coefficients (MFCC) are computed from the speech amplitude spectrum with the following steps as shown in figure 1:

- After pre-emphasizing and framing of the speech signal discrete Fourier transform (DFT) is applied to estimate the short-time power spectrum.
- Apply Mel-filterbank to the estimated power spectrum to compute Mel-filterbank energies (FBE).
- Apply discrete cosine transform (DCT) to the log (FBE) to obtain MFCC features.

3.2. MFCC - CNPCC features

This feature is the concatenation of MFCC and cosine normalized phase cepstral coefficients (CNPCC) [2, 20], denoted here as MFCC-CNPCC [2]. This feature can be obtained with the following steps:

- Compute MFCC feature using the procedure mentioned in section 3.1.
- Compute the CNPCC features by applying a DCT transform on the unwrapped and cosine normalized phase spectra [2, 20].
- Concatenate the MFCC and CNPCC features to get MFCC-CNPCC.

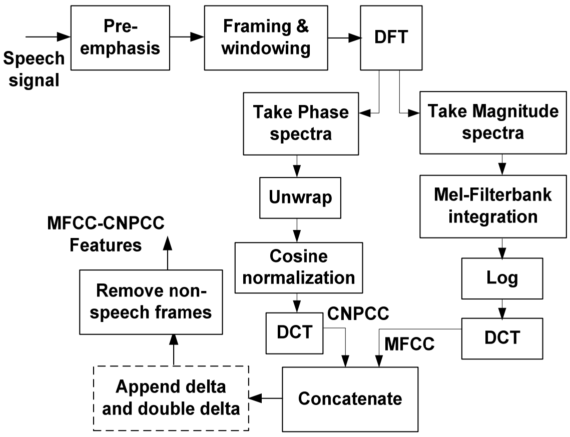

Figure 1: Conventional MFCC, phase-based feature CNPCC and joint Amplitude & phase-based countermeasures MFCC-CNPCC extraction steps [2].

3.3. Product spectrum-based MFCC features

The product spectrum, introduced in [22] for a speech recognition task, helps to mitigate the effect of zeros in the group delay function. The group delay is defined as the negative derivative of the phase spectrum. The product of speech power spectrum $\left|X(\omega)\right|^2$ and the group delay function $\tau_g(\omega)$ is known as product spectrum $P(\omega)$ and is expressed as:

$$P(\omega) = \left|X(\omega)\right|^2 \tau_g(\omega) = X_R(\omega)Y_R(\omega) + X_I(\omega)Y_I(\omega), \quad (1)$$

where $\tau_g(\omega) = \left(X_R(\omega)Y_R(\omega) + X_I(\omega)Y_I(\omega)\right) \big/ \left|X(\omega)\right|^2$, $X(\omega)$ is the Fourier transform of speech signal $x(n)$, $Y(\omega)$ is the Fourier transform of $y(n) = nx(n)$, and the subscripts R and I denote the real and imaginary parts, respectively. The product spectrum incorporates information from both the amplitude and phase spectra and therefore, this feature may be a good candidate for spoofing detection and speaker verification [2, 23]. Figure 2 presents an overview of the product spectrum-based MFCC (PSCC) feature extraction procedure.

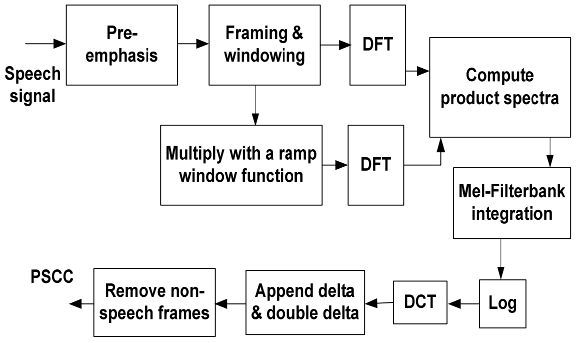

Figure 2: Schematic diagram showing various steps to extract countermeasures for spoofing detection task based on product spectrum cepstral coefficients (PSCC) [2].

3.4. Linear prediction (LP)-based features

In LP analysis each sample is predicted as a linear weighted sum of the past *p* samples as:

$$\hat{x}(n) = \sum_{k=1}^{p} a_k x(n-k) , \tag{2}$$

where p is prediction order, $x(n)$ is current speech sample, and $\{a_k\}$ are LP coefficients. In this work we use *p* = 20. The prediction error $e(n)$ is obtained as the difference between the predicted speech sample $\hat{x}(n)$ and the actual speech sample as:

$$e(n) = x(n) - \hat{x}(n) = x(n) - \sum_{k=1}^{p} a_k x(n-k) . \tag{3}$$

The prediction error $e(n)$ might contain information which has not been captured by the LP coefficients of the actual signal and which can be used for speaker recognition [24] and spoofing detection tasks [2, 4]. The linear prediction cepstral coefficients (LPCC) and linear prediction residual cepstral coefficients (LPRC) are obtained by performing LP analysis of $x(n)$ and $e(n)$, respectively, and then converting the LP coefficients into cepstral coefficients by the Levinson Durbin recursion.

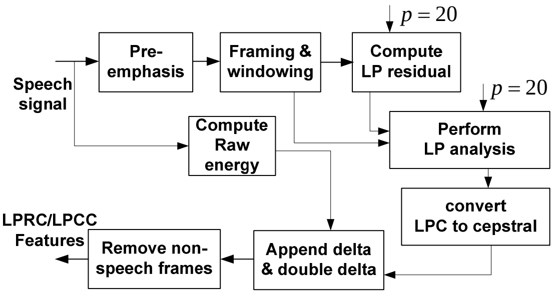

Figure 3: *Block diagram showing various steps to extract linear prediction cepstral coefficients (LPCC) and linear prediction residual cepstral coefficients (LPRC) by performing LP analysis of the actual speech signal and the residual signal, respectively.*

4. GMM classifier

We train a 512-component GMM for the human speech model M_h on all human training features and another for spoofed speech models M_s on all training spoofed (S1-S5) features. Then, for each test feature frame O the log likelihood ratio (*LLR*) is computed for the two hypotheses as:

$$LLR = \log\left(p\left(O\middle|M_h\right)\right) - \log\left(p\left(O\middle|M_s\right)\right) . \tag{4}$$

5. Experiments

5.1. Experimental setup

Features described in sections 2 & 3 are used as spoofing countermeasures. Three variants of bottleneck features (BNF) are extracted, namely, DFB-BNF (extracted from delta filterbank features), DLPCC-BNF (extracted from delta + double delta LPCC features), and DPSCC-BNF (extracted from delta + double delta PSCC features). The motivation for excluding the static feature and using only delta or delta + double delta features for this task is the findings of [27]. The feature dimension is 60 (including log energy, delta and double deltas) for all systems with the exception of the bottleneck features-based systems which have a dimension of 64. The analysis frame length is 25 ms with a frame shift of 10 ms. No feature normalization was applied as it was found to degrade spoofing detection performance [2]. Only global mean and variance normalization is applied to the feature which is used as input to the DNN. Non-speech frames were removed using VAD (voice activity detector) segmentations generated by a GMM-based VAD [25-26].

5.2. ASVspoof2015 challenge corpus

The ASVspoof2015 challenge corpus is comprised of human and spoofed speech signals. Human (or genuine) speech is recorded from 45 male and 61 female speakers without significant channel or background noise effects. The spoofed speech is obtained from the human data by applying several voice conversion and speech synthesis methods, as described in [1], are given briefly below:
S1: Frame selection based voice conversion [29].
S2 : Voice conversion based on modifying the first cepstrum of the MFCC.
S3: Speech synthesis using the Hidden Markov model toolkit (HTS) and models are adapted to the target speaker with 20 utterances [30].
S4: Same as S3 but instead of 20 utterances 40 utterances are used to adapt models to the target speakers.
S5: Voice conversion using Festvox toolkit [37].
S6: Voice conversion using joint density GMM and maximum likelihood parameter generation [34].
S7: Similar to S6 spoofing attack but using line spectrum pair (SLP) instead of MFCC.
S8: Tensor based voice conversion [35].

S9: Voice conversion based spoofing attack using kernel partial least square (KPLS) to implement a non-linear transformation [36].
S10: Speech synthesis spoofing attack using MaryTTS toolkit [38]. Here, the models are trained with 40 utterances per target speaker.
The entire corpus is divided into three subsets: training, development and evaluation. The training set includes 3750 genuine and 12625 spoofed recordings. In the development set there are 497 genuine and 49875 spoofed trials. The evaluation set is comprised of 9404 genuine and 184000 spoofed trials. There is no speaker overlap across the three subsets regarding target speakers used in voice conversion or speech synthesis adaptation [1]. The spoofing attack S1 to S5 are common to train, development and evaluation sets and are known as known spoofing attacks. The evaluation set includes additional five spoofing attacks (S6 to S10) and are referred to as unknown spoofing attacks. More details about the challenge protocols and corpus can be found in [1, 24].

5.3. Results and Discussion

The performance of the deep neural network-based systems (i.e., DFB-BNF, DMCC-BNF, DPSCC-BNF, DLPCC-BNF, and DPSCC-DNN) and other systems are evaluated on the ASVspoof2015 challenge evaluation data. The parameters of each system were tuned on development test data. For convenient, a brief description of each of 13 systems is presented in Table 1. The equal error rate (EER) metric is used for evaluation of system performance.

Spoofing detection results on the nine vocoded spoofing attacks (S1-S9) [1] of the evaluation data, in terms of percentage EER, obtained with different countermeasures are reported in Table 2. Results from four participants (CRIM [2], SJTU [3], NTU [12], and UZ (University of Zaragoza) [9]) primary submitted fused systems are also included in Table 2 for comparison purposes. The results of ASVspoof2015 challenge and the participants' system descriptions can be found on the challenge website (http://www.spoofingchallenge.org/) [24, 33].

It is observed from Table 2 that the bottleneck features-based systems depicted excellent performance on all nine spoofing attacks which use a vocoder for speech synthesis or voice conversion i.e., vocoded spoofing attacks. The DLPCC-BNF system outperformed other systems in *known*, *unknown* and *All* conditions (against spoofing attacks S1-S9). Comparing the performances of DLPCC and DLPCC-BNF systems and the excellent results obtained with the DFB-BNF, DMCC-BNF, DPSCC-BNF, & DLPCC-BNF systems proves the effectiveness of using deep neural network - bottleneck countermeasures to reduce spoofing detection error rates on the vocoded spoofing attacks (S1-S9). Observing the EERs of the DLPCC and LPRC systems it can be concluded that the LPRC features are more discriminative than the LPCC features for distinguishing human speech from spoofed speech [2] against the vocoded spoofing attacks. This is because prediction error will be more in the natural speech than that of the spoofed (synthesized or voice converted) speech [2].

Table 3 presents the EERs obtained with the DFB-BNF, DMCC-BNF, DLPCC-BNF, DPSCC-BNF, DPSCC-DNN and our primary system, which is fusion of several countermeasures-based systems, for the ASVspoof2015 challenge. It is evident from Tables 1 & 2 that if the bottleneck features are extracted using acoustic features appropriate for a specific task, a significant reduction in spoofing detection error can be obtained with a single system. Considering all spoofing attacks (vocoded and non-vocoded, i.e., S1-S10) DMCC-BNF and DPSCC-DNN systems performed the best. Both systems provided an average relative improvements of 20.0% and 19.0%, respectively, in terms of EER, compared to our primary system (fused system, EER = 2.69%) submitted to the spoofing challenge 2015.
Since no vocoder was used in spoofing technique S10 synthesis [1], vocoder mismatch between the training and evaluation data resulted in significantly higher EERs for all participants on the S10 attack.

6. Conclusions

In this paper we employed deep neural network (DNN) for spoofing detection task. We used the bottleneck feature representations supplied by a DNN as a spoofing countermeasure to defend speaker verification systems against various spoofing attacks. We also used the DNN posteriors for discriminating between human and spoof speech signals. The delta filterbank, delta + double delta MFCC, LPCC and PSCC (product spectrum-based cepstral coefficients) features were used as inputs to the DNN. For all reported systems except the DPSCC-DNN system, a standard GMM classifier was used for classification. For the DPSCC-DNN system the output of the DNN i.e., posterior probabilities are directly transformed into log likelihood ratio (LLR). It was observed from the reported results on the ASVspoof2015 evaluation data that the bottleneck features are very effective to reduce the spoofing detection error rates on the vocoded spoofing attacks (S1-S9). The DLPCC-BNF countermeasure demonstrated excellent performance with an EER of 0.0% on all (S1-S9) vocoded spoofing attacks. Considering all spoofing attacks (vocoded and non-vocoded, i.e., S1-S10) the DPSCC-DNN and DMCC-BNF systems performed the best. The DPSCC-DNN and DMCC-BNF systems provided an average relative improvements of 19.0%, and 20.0%, respectively, in terms of EER, compared to our primary system (fused system, EER = 2.69%) submitted to the spoofing challenge 2015.

7. References

[1] Zhizheng Wu, Tomi Kinnunen, Nicolas Evans, Junichi Yamagishi, Cemal Hanilci, Md Sahidullah, Aleksandr Sizov, "ASVspoof 2015: the First Automatic Speaker Verification Spoofing and Countermeasures Challenge," Interspeech 2015.

[2] Md Jahangir Alam, Patrick Kenny, Gautam Bhattacharya, Themos Stafylakis, "Development of CRIM System for the Automatic Speaker Verification Spoofing and Countermeasures Challenge 2015", Interspeech 2015.

[3] Nanxin Chen, Yanmin Qian, Heinrich Dinkel, Bo Chen, Kai Yu, "Robust Deep Feature for Spoofing Detection - The SJTU System for ASVspoof 2015 Challenge", Interspeech 2015.

[4] Artur Janicki, "Spoofing Countermeasure Based on Analysis of Linear Prediction Error", Interspeech 2015.

[5] Yi Liu, Yao Tian, Liang He, Jia Liu, Michael T. Johnson, "Simultaneous Utilization of Spectral Magnitude and Phase Information to Extract Supervectors for Speaker Verification Anti-spoofing", Interspeech 2015.

[6] Sergey Novoselov, Alexandr Kozlov, Galina Lavrentyeva, Konstantin Simonchik, Vadim Shchemelinin, "STC Anti-spoofing Systems for the ASVspoof 2015 Challenge", arXiv:1507.08074, 2015.

[7] Tanvina B. Patel, Hemant A. Patil, "Combining Evidences from Mel Cepstral, Cochlear Filter Cepstral and Instantaneous Frequency Features for Detection of Natural vs. Spoofed Speech", Interspeech 2015.

[8] Jon Sanchez, Ibon Saratxaga, Inma Hernaez, Eva Navas, Daniel Erro, "The AHOLAB RPS SSD Spoofing Challenge 2015 submission", Interspeech 2015.

[9] Jesus Villalba, Antonio Miguel, Alfonso Ortega, Eduardo Lleida, "Spoofing Detection with DNN and One-class SVM for the ASVspoof 2015 Challenge", Interspeech 2015.

[10] Longbiao Wang , Yohei Yoshida, Yuta Kawakami, Seiichi Nakagawa, "Relative phase information for detecting human speech and spoofed speech", Interspeech 2015.

[11] Shitao Weng, Shushan Chen, Lei Yu, Xuewei Wu, Weicheng Cai, Zhi Liu, Ming Li, "The SYSU System for the Interspeech 2015 Automatic Speaker Verification Spoofing and Countermeasures Challenge", arXiv:1507.06711, 2015.

[12] Xiong Xiao, Xiaohai Tian, Steven Du, Haihua Xu, Eng Siong Chng, Haizhou Li, "Spoofing Speech Detection Using High Dimensional Magnitude and Phase Features: the NTU Approach for ASVspoof 2015 Challenge", Interspeech 2015.

[13] Yao Tian, Meng Cai, Liang He, Jia Liu, "Investigation of Bottleneck Features and Multilingual Deep Neural Networks for Speaker Verification," Proc. Interspeech 2015.

[14] F. Richardson, D. Reynolds, and N. Dehak, "Deep Neural Network Approaches to Speaker and Language Recognition," *IEEE Signal Processing Letters*, Vol. 22, No. 10, pp. 1671-1675, October 2015.

[15] G. Hinton et al., “Deep neural networks for acoustic modeling in speech recognition,” IEEE Signal Process. Mag. , pp. 82–97, Nov. 2012.

[16] Y. Song, B. Jiang, Y. Bao, S. Wei, and L.-R. Dai, “I-vector representation based on bottleneck features for language identification,” Electron. Lett. , pp. 1569–1580, 2013.

[17] P. Matejka, L. Zhang, T. Ng, H. S. Mallidi, O. Glembek, J. Ma, and B. Zhang, “Neural network bottleneck features for language identification,” in Proc. IEEE Odyssey , pp. 299–304, 2014.

[18] Zhizheng Wu, Tuomas Virtanen, Eng Siong Chng, Haizhou Li, "Exemplar-based sparse representation with residual compensation for voice conversion", IEEE/ACM Transactions on Audio, Speech and Language Processing, Vol. 22, Issue 10, pp. 1506-1521, 2014.

[19] P. L. De Leon, M. Pucher, J. Yamagishi, I. Hernaez, and I. Saratxaga, “Evaluation of speaker verification security and detection of HMM-based synthetic speech,” IEEE Trans. Audio, Speech and Language Processing , vol. 20, no. 8, pp. 2280–2290, 2012.

[20] Z. Wu, X. Xiao, E. S. Chng, and H. Li, “Synthetic speech detection using temporal modulation feature,” in Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP) , 2013.

[21] Z. Wu, E.S. Chng, and H. Li, “Detecting converted speech and natural speech for anti-spoofing attack in speaker recognition,” in Interspeech, 2012.

[22] D. Zhu and K. Paliwal, “Product of power spectrum and group delay function for speech recognition,” in Proc. Int. Conf. Acoust., Speech, Signal Process. , pp. 125–128, 2004.

[23] Md. Jahangir Alam, Patrick Kenny and Themos Stafylakis, "Combining Amplitude and Phase-Based Features for Speaker Verification with Short Duration Utterances," Proc. Interspeech, Dresden Germany, Sept. 2015.

[24] Z. Wu, T. Kinnunen, N. Evans, J. Yamagishi, ASVspoof 2015: Automatic Speaker Verification Spoofing and Countermeasures Challenge Evaluation plan, 2015. http://www.spoofingchallenge.org/asvSpoof.pdf

[25] T. Kinnunen, P. Rajan, ”A practical, self-adaptive voice activity detector for speaker verification with noisy telephone and microphone data”, *Proc. of* ICASSP, pp. 7229-7233, Vancouver, Canada, May 2013.

[26] Alam, J., Kenny, P., Ouellet, P., Stafylakis, T. and Dumouchel, P., "Supervised/Unsupervised Voice Activity Detectors for Text-Dependent Speaker Recognition on the RSR2015 Corpus," Proc. Odyssey Speaker and Langauge Recognition Workshop, Joensuu, Finland June 2014.

[27] M. Sahidullah, T. Kinnunen, C. Hanilçi, ”A Comparison of Features for Synthetic Speech Detection”, Proc. of Interspeech, 2015.

[28] P. L. De Leon, M. Pucher, J. Yamagishi, I. Hernaez, and I. Saratxaga, “Evaluation of speaker verification security and detection of HMM-based synthetic speech,” IEEE Trans. Audio, Speech and Language Processing , vol. 20, no. 8, pp. 2280–2290, 2012.

[29] Z. Wu, T. Virtanen, T. Kinnunen, E. S. Chng, and H. Li, “Exemplar-Based Unit Selection for Voice Conversion Utilizing Temporal Information,” in Proc. of Interspeech, pp. 950–954, Lyon, France, Aug. 2013.

[30] J. Yamagishi, T. Kobayashi, Y. Nakano, K. Ogata, and J. Isogai, “Analysis of Speaker Adaptation Algorithms for HMM-Based Speech Synthesis and a Constrained SMAPLR Adaptation Algorithm,” IEEE Transactions on Audio, Speech and Language Processing, vol. 17, no. 1, pp. 66–83, Jan. 2009.

[31] Z. Wu, N. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, and H. Li, “Spoofing and countermeasures for speaker verification: A survey,” Speech Communication, vol. 66, no. 0, pp. 130 – 153, 2015.

[32] F. Richardson, D. Reynolds, and N. Dehak, "A Unified Deep Neural Network for Speaker and Language Recognition," Proc. Interspeech, Dresden, Germany, September 2015.

[33] ASVspoof2015 challenge results. http://www.zhizheng.org/slides/is2015_overview_talk.pdf

[34] T. Toda, A. W. Black, and K. Tokuda, “Voice Conversion Based on Maximum-Likelihood Estimation of Spectral Parameter Trajectory,” IEEE Transactions on Audio, Speech and Language Processing, vol. 15, no. 8, pp. 2222–2235, Nov. 2007.

[35] D. Saito, K. Yamamoto, N. Minematsu, and K. Hirose, “One-to-many voice conversion based on tensor representation of speaker space,” in Proc. of Interspeech, pp. 653–656, Florence, Italy.

[36] E. Helander, H. Silen, T. Virtanen, and M. Gabbouj, “Voice Conversion Using Dynamic Kernel Partial Least Squares Regression,” IEEE Transactions on Audio, Speech, and Language Processing, vol. 20, no. 3, pp. 806–817, Mar. 2012.

[37] The Festvox speech synthesis system: http://www.festvox.org

[38] The MARY TTS - an open-source, multilingual text-to-speech synthesis system, http://mary.dfki.de

Table 1: A brief description of the systems used for the spoofing detection in this work.

MFCC	Spoofing detection with MFCC feature using a GMM classifier
MFCC-CNPCC	Spoofing detection with combined MFCC -CNPCC feature using a GMM classifier
PSCC	Product spectrum-based MFCC (PSCC) + standard GMM classifier
DLPCC	Delta + double delta LPCC features with a GMM classifier
LPRC	LP residual cepstra (LPRC)-based spoofing detection with a GMM classifier
DMCC	40-dimensional delta + double delta MFCC features with a GMM classifier
DFB-BNF	Bottleneck features (BNF) + GMM classifier, input feature to the DNN is delta filterbank (DFB) features.
DLPCC-BNF	Bottleneck features (BNF) + GMM classifier, input feature to the DNN is DLPCC features.
DPSCC-BNF	Bottleneck features (BNF) + GMM classifier, input feature to the DNN is delta PSCC (DPSCC) features.
DPSCC-DNN	DNN posteriors were directly transformed in likelihood ratio (LLR), input feature to the DNN is delta PSCC (DPSCC) features.
CRIM [2]	Our primary (fused) system for the spoofing challenge 2015 [2]
SJTU [3]	Primary submitted system of SJTU for the spoofing challenge 2015 [3]
NTU [12]	Primary submitted system of NTU for the spoofing challenge 2015 [12]
UZ [9]	Primary system of University of Zaragoza for spoofing challenge 2015 [9]

Table 2: *Spoofing detection performance on the challenge evaluation data using a standard GMM classifier with various features as countermeasures and with the DNN posteriors and bottleneck features. The lowest EERs are highlighted in bold face.*

	EER (%)				
	Known (S1-S5)	**Unknown (S6-S9)**	**All (S1-S9)**	**Unknown S10**	**All (S1-S10)**
MFCC	0.46	0.39	0.43		
MFCC-CNPCC	0.93	0.55	0.77		
PSCC	0.390	0.337	0.366		
DLPCC	0.489	0.099	0.316		
LPRC	0.278	0.179	0.234		
DFB-BNF	**0.0088**	**0.0183**	**0.013**	32.28	3.24
DLPCC-BNF	**0.00**	**0.00**	**0.00**	33.0	3.3
DPSCC-BNF	0.019	0.025	0.022	32.69	3.28
DMCC-BNF	**0.0087**	**0.005**	**0.007**	21.47	**2.15**
DPSCC-DNN	1.16	0.79	1.0	**12.86**	**2.18**
CRIM [2]	0.041	0.085	0.060	26.39	2.69
SJTU [3]	0.058	0.098	0.076	24.601	2.528
NTU [12]	**0.003**	**0.003**	**0.003**	26.14	2.617
UZ [9]	0.025	0.033	0.029	40.708	4.097

Table 3: *Spoofing detection performance of the DNN posterior and bottleneck features-based systems against various known (S1-S5) and unknown (S6-S10) spoofing attacks on the challenge evaluation corpus. Our primary fused system results in the ASVspoof2015 challenge are also included for comparison. The lowest EERs are highlighted in bold face.*

EER (%)					
Known					
S1	**S2**	**S3**	**S4**	**S5**	
0.024	0.105	0.025	0.017	0.033	CRIM [2]
0.00	0.03	**0.00**	**0.00**	0.01	DFB-BNF
0.00	**0.00**	**0.00**	**0.00**	**0.00**	DLPCC-BNF
0.00	0.08	**0.00**	**0.00**	0.01	DPSCC-BNF
0.00	0.04	**0.00**	**0.00**	0.01	DMCC-BNF
0.04	4.3	**0.0**	0.01	1.47	DPSCC-DNN
Unknown					
S6	**S7**	**S8**	**S9**	**S10**	
0.093	0.011	0.24	0.00	26.39	CRIM [2]
0.01	0.00	0.05	0.01	32.28	DFB-BNF
0.00	**0.00**	**0.00**	**0.00**	33.00	DLPCC-BNF
0.03	**0.00**	0.07	**0.00**	32.69	DPSCC-BNF
0.00	**0.00**	**0.02**	**0.00**	21.47	DMCC-BNF
0.94	0.21	1.85	0.19	**12.86**	DPSCC-DNN