IberSPEECH 2018
21-23 November 2018, Barcelona, Spain

Performance evaluation of front- and back-end techniques for ASV spoofing detection systems based on deep features

Alejandro Gomez-Alanis[1], Antonio M. Peinado[1], Jose A. Gonzalez[2], and Angel M. Gomez[1]

[1]University of Granada, Granada, Spain
[2]University of Malaga, Malaga, Spain
{agomezalanis,amp,amgg}@ugr.es, j.gonzalez@uma.es

Abstract

As Automatic Speaker Verification (ASV) becomes more popular, so do the ways impostors can use to gain illegal access to speech-based biometric systems. For instance, impostors can use Text-to-Speech (TTS) and Voice Conversion (VC) techniques to generate speech acoustics resembling the voice of a genuine user and, hence, gain fraudulent access to the system. To prevent this, a number of anti-spoofing countermeasures have been developed for detecting these high technology attacks. However, the detection of previously unforeseen spoofing attacks remains challenging. To address this issue, in this work we perform an extensive empirical investigation on the speech features and back-end classifiers providing the best overall performance for an antispoofing system based on a deep learning framework. In this architecture, a deep neural network is used to extract a single identity spoofing vector per utterance from the speech features. Then, the extracted vectors are passed to a classifier in order to make the final detection decision. Experimental evaluation is carried out on the standard ASVSpoof2015 data corpus. The results show that classical FBANK features and Linear Discriminant Analysis (LDA) obtain the best performance for the proposed system.
Index Terms: Automatic speaker verification, spoofing detection, deep neural networks, deep features, classifier.

1. Introduction

Automatic Speaker Verification (ASV) aims to authenticate the identity claimed by a given individual [1]. However, most ASV systems are vulnerable to spoofing attacks, in which an impostor try to gain fraudulent access to the system by presenting to the ASV system speech acoustics resembling the voice of a genuine user. Four types of spoofing attacks have been identified [2]: (i) replay (i.e. using pre-recorded voice of the target user), (ii) impersonation (i.e. mimicking the voice of the target voice), and also either (iii) text-to-speech synthesis (TTS) or (iv) voice conversion (VC) systems to generate artificial speech resembling the voice of a legitimate user. The aim of this work is to develop robust anti-spoofing countermeasures for either VC or TTS based attacks.

The performance of anti-spoofing systems can meaningfully vary depending on the voice features used to feed them. Due to this, voice features have attracted the attention of a number of researchers [8, 9, 10]. However, anti-spoofing systems based on neural networks usually use classical voice features, such as FBANKs, and to the best of our knowledge, the new popular CQCC features have not been employed yet to feed these types of systems.

In the last years, the technique of deep features extraction have been explored to obtain more discriminative and effective features for spoofing detection [6, 7, 11]. This technique consists of employing deep neural networks in the front-end of the anti-spoofing system which are fed by speech features, so that the deep features extracted by the neural network are passed to a classifier in order to make the final detection decision (genuine or spoof). The core idea is to take advantage of the nonlinear modeling and discriminative capabilites of deep neural networks which have shown to be suitable for feature engineering [3], not only for spoofing detection, but also for speech recognition [4], speaker recognition [3], and speech synthesis [5].

In this work, we compare the performance of different features and back-ends in an anti-spoofing system which extracts deep features [6] in order to detect VC and TTS attacks. This anti-spoofing system employs a convolutional neural network (CNN) plus a recurrent neural network (RNN) and gets a single spoofing identity representation per utterance. Although a similar comparison has already been studied in [7], our study presents three important differences: (1) our anti-spoofing system employs a CNN to extract convolutional features at the speech frame level, (2) we compare the performance of classical features, such as FBANKs and MFCCs, with the performance of the recent popular CQCC features [8], and (3) we combinate different features and classifiers in order to find the combination which offers the best performance.

This paper is organized as follows. Section 2 describes the features and back-ends we are going to compare in a CNN + RNN anti-spoofing system. Then, in Section 3, we outline the speech corpora, the network training, and the performance evaluation details. Section 4 discusses the results of the different features and back-ends in the deep neural network based anti-spoofing system. Finally, we present the conclusions derived from this research in Section 5.

2. System description

This section is devoted to the description of the anti-spoofing system. First, Section 2.1 describes different voice features: FBANK, MFCC and CQCC. The neural network architecture for deep feature extraction is detailed in Section 2.2. Furthermore, Section 2.3 describes different classifiers (back-ends): Linear Discriminant Analysis (LDA), Support Vector Machine (SVM), and One-Class Support Vector Machine (One-Class SVM).

2.1. Speech features

As demonstrated in [11], traditional log MEL filterbank features (FBANK) are effective for detecting spoofing attacks with systems based on neural networks. These features are obtained by passing the Short Time Fourier Transform (STFT) magnitude spectrum through a Mel-filterbank and applying a log opera-

10.21437/IberSPEECH.2018-10

tion. However, FBANK features are usually high-correlated. One way to decorrelate these features is to apply the Discrete Cosine Transform (DCT) to get the classical Mel Frequency Cepstral Coefficient (MFCC) features.

In [8], CQCC features are proposed for spoofing detection, which are obtained using the Constant Q Transform (CQT). The Q factor is a measure of the selectivity of each filter and is defined as the ratio between the center frequency and the bandwidth of the filter. In contrast to the STFT, whose Q factor increases when moving from low to high frequencies as the bandwidth is the same for all filters, the bandwidth of the filters employed in the CQT is not constant, and this results in getting a higher frequency resolution for low frequencies and a higher temporal resolution for high frequencies. In this manner, the CQCC features try to imitate the human perception system which is known to approximate a constant Q factor between 500Hz and 20kHz [20].

In this work, we employ the classical FBANK and MFCC features, as well as the popular CQCC features, to feed the anti-spoofing system.

2.2. Front-end

The front-end architecture of the anti-spoofing system is shown in Fig. 1. A context window of W frames (centered at the frame being processed) is used to obtain the input signal spectral features which are fed into the system. Then, the CNN provides a deep feature vector per window, and all deep features vectors of the considered utterance are processed by the RNN which computes an embedding vector for the whole utterance. We call this the spoofing identity vector. Since the front-end is trained to perform utterance-level classification of the attacks, this embedding vector should provide more discriminative information for spoofing detection than the raw speech features.

In this architecture, the CNN plays the role of a frame-level deep feature extractor providing one feature vector for each context window of W frames. In order to this, the CNN acts as a classifier whose task consists of determining whether the input feature are either genuine or belong to one of the K spoofing attacks (S1, S2, ..., SK) present in the training set. This CNN uses 2 convolutional and pooling layers as feature extractors, followed by 2 fully connected layers with a softmax layer of $K+1$ neurons as classification layer. To prevent overfitting, we used an annealed dropout training procedure [17]. In annealed dropout, the dropout probability of the nodes in the network is decreased as training progresses. In this work, the annealed function reduces the dropout rate from an initial rate $prob[0]$ to zero over N steps with constant rate. The dropout probability $prob[t]$ at epoch t is given as:

$$prob[t] = max\Big(0, 1 - \frac{t}{N}\Big) prob[0]. \quad (1)$$

As shown in Fig. 1, the deep features obtained from the CNN are fed into an RNN, which computes the anti-spoofing identity vector of the utterance. The main advantage of using an RNN, based on gated recurrent units (GRU) [16], is its ability for learning the long-term dependencies of the subsequent deep feature vectors. Finally, a fully connected layer containing $K+1$ neurons (one per class: genuine, S1, S2, ..., SK) is connected to the output of the last time step, followed by a softmax layer. The state of the last time step represents the single identity spoofing vector of the whole utterance.

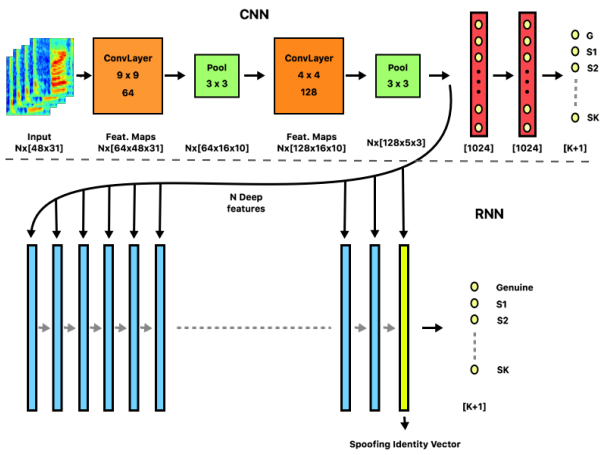

Figure 1: *Front-end architecture of the anti-spoofing system which extracts a spoofing identity vector per utterance (N represents the number of context windows per utterance). This system is proposed in [6].*

2.3. Back-end

After deep feature extraction, every utterance is represented by a single spoofing identity vector. A back-end classifier is then applied on these vectors to do the final detection decision. In this section three classifiers will be tested: LDA, SVM and one-class SVM.

A. *Linear Discriminant Analysis*

LDA assumes that each class density can be modeled as a multivariate Gaussian

$$N(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi)^{\frac{p}{2}}|\boldsymbol{\Sigma}_k|^{\frac{1}{2}}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)}, \quad (2)$$

where $\boldsymbol{\Sigma}_k$ and $\boldsymbol{\mu}_k$ is the covariance and mean for class k, and p is the dimension of the identity vectors. Moreover, the LDA model assumes every class shares the same covariance, that is, $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}, \forall k$. The goal of LDA is to find a transformation which maximizes the distance between classes while minimizing the spreading within each class. This can be formulated as a diagonalization problem where matrix $\boldsymbol{\Sigma}_b\boldsymbol{\Sigma}$ ($\boldsymbol{\Sigma}_b$ is the between-class covariance) is diagonalized, so the transformation can be built from the resulting eigenvectors.

Our LDA classifier uses $K+1$ classes which represent genuine speech and the K known spoofing attacks considered in the training set. In this way, the LDA assigns a genuine speech confidence score to each utterance, which is then used for binary decision (spoof or genuine) during the evaluation.

B. *Support Vector Machine*

A support vector machine (SVM) separates data points in a high dimensional space defined by a kernel function. In this manner, we first obtain a binary function that describes the probability density function where the genuine data lives. This function returns +1 in the small region corresponding to the genuine speech data and -1 elsewhere. Thus, the core idea of SVM is to estimate the hyperplane with the largest separation margin between the two classes.

Table 1: *Structure of the ASVspoof2015 data corpus divided by the training, development and evaluation sets [14].*

Subset	# Speakers		# Utterances	
	Male	Female	Genuine	Spoofed
Training	10	15	3750	12,625
Development	15	20	3497	49,875
Evaluation	20	26	9404	184,000

In this work, this classifier is used to classify the spoofing identity vectors obtained by the front-end system, where +1 indicates genuine speech and -1 indicates spoofed speech.

C. One-Class Support Vector Machine

Complex classifiers may overfit the training spoofed data. To create a spoof-independent system, we also test a derivative model that can only be trained on genuine speech data. This is a type of one-class SVMs [12], usually employed to find abnormal data. This was first tried in spoofing detection with phase-based features in [13]. This kind of SVM is also applied here to classify the spoofing identity vectors, and only genuine speech data has been used to train the one-class SVM model.

3. Experimental framework

To evaluate the performance of several features and back-ends in an anti-spoofing system based on neural networks, the ASVspoof 2015 dataset [14], a standard data corpus for research on spoofing detection, was employed. Details about the methodology followed for training and testing are also given in this section.

3.1. Speech corpus

The ASVspoof 2015 corpus [14] defines three datasets (training, development and evaluation), each one containing a mix of genuine and spoofed speech. The structure of these three datasets are shown in Table 1. Spoofing attacks were generated either by TTS or VC. A total of 10 types of spoofing attacks (S1 to S10) are defined: three of them are implemented using TTS (S3, S4 and S10), and the remaining seven ones (S1, S2, S5, S6, S7, S8 and S9) using different VC systems. Attacks S1 to S5 are referred to as *known attacks*, since the training and development sets contain data for these types of attacks, while attacks S6 to S10 are referred to as *unknown attacks*, because they only appear in the evaluation set. More details about this corpus can be found in [14].

3.2. Spectral Analysis

The frame window size is 25 ms with 10 ms of frame shift. Moreover, the size of the context window is $W = 31$ frames, and the number of filters used to get the spectral features is $M = 48$ filters. In contrast to [7] and [11], we use a 48-dim static spectral features without delta and acceleration coefficients, as we have realized that the context window of 31 frames is already exploiting the correlations between consecutive frames. Therefore, a higher spectral resolution is achieved while the size of the spectral feature vector is smaller than in [7].

3.3. Training

The CNN and RNN networks are trained using Adam optimizer [18]. As there are $K = 5$ known spoofing attacks in the data corpus, the softmax layer of both CNN and RNN contains $K + 1 = 6$ neurons (one per class). The two fully connected layers of the CNN have 1024 sigmoid neurons, and the layer of the RNN has 1920 GRUs, which is the length of the identity spoofing vector of the whole utterance. To prevent the problem of overfitting, the initial dropout probabilities are 50% and 40% from the first to the last fully connected layer, respectively. Also, early stopping is applied in order to stop the training process when no improvement of the cross entropy is obtained after 15 iterations. All the specified parameters of the system have been optimized using the validation set of the data corpus [14].

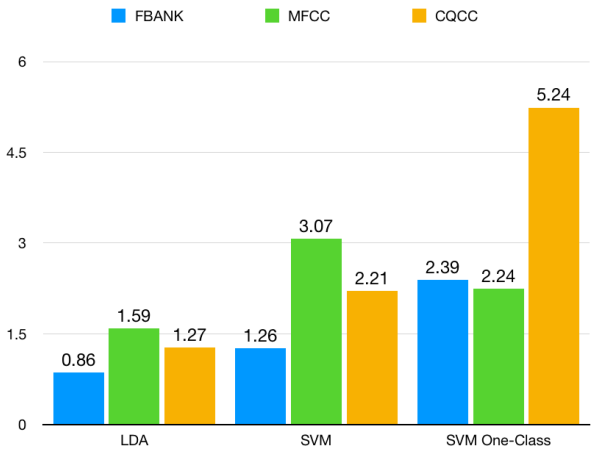

Figure 2: *Comparison of average EERs (%) between known and unknown spoofing attacks on evaluation dataset for different features and back-ends, including FBANK, MFCC, CQCC, LDA, SVM and SVM One-Class.*

3.4. Performance evaluation

The equal error rate (EER) is used to evaluate the system performance. As described in the ASVspoof 2015 challenge evaluation plan [14], the EER was computed independently for each spoofing algorithm and then the average EER across all attacks was used. To compute the average EER, we used the Bosaris toolkit [15].

4. Experimental results

4.1. Comparison of features and back-ends

Table 2 shows the detailed results of the different features (FBANK, MFCC and CQCC) and classifiers (LDA, SVM and SVM One-Class) in the described CNN + RNN anti-spoofing system. Furthermore, a summary of these results is shown in Fig. 2. The best performance is obtained with the combination of FBANK features and the LDA classifier. In average, the FBANK features obtain the best performance independently of the back-end, although MFCC features perform better on the SVM One-Class considering all the attacks. The CQCC features achieve the best average performance in the known attacks with LDA and SVM back-ends, but these two combinations perform very poorly in the S10 attack.

Regarding the back-ends, the LDA outperforms the other 2 classifiers in the known and unknown attacks. Moreover, the binary SVM classifier performs much better than SVM One-Class using FBANK and CQCC features.

Table 2: *Comparison on evaluation dataset for each spoofing attack in terms of (%) EER*

Features	Back-end	Known Attacks						Unknown Attacks						Total
		S1	S2	S3	S4	S5	Avg.	S6	S7	S8	S9	S10	Avg.	Avg.
FBANK	LDA	**0.01**	0.09	**0.00**	**0.00**	0.11	0.04	0.66	0.21	**0.00**	0.36	**7.16**	**1.68**	**0.86**
	SVM	0.03	0.13	**0.00**	0.01	0.22	0.08	0.77	0.34	0.18	0.48	10.46	2.44	1.26
	SVMOne	0.36	2.07	0.17	0.12	4.37	1.42	5.44	1.34	0.34	1.53	8.23	3.38	2.40
MFCC	LDA	0.06	0.08	**0.00**	**0.00**	0.06	0.04	0.11	**0.12**	**0.00**	0.05	15.43	3.14	1.59
	SVM	0.05	0.19	0.01	0.01	0.23	0.10	0.22	0.21	0.05	0.15	29.58	6.04	3.07
	SVMOne	0.43	1.97	0.12	0.12	2.11	0.95	3.38	2.07	0.06	1.03	11.09	3.53	2.24
CQCC	LDA	0.04	0.04	**0.00**	**0.00**	0.04	0.02	0.13	0.51	0.05	0.08	11.76	2.51	1.27
	SVM	0.03	**0.01**	0.01	0.01	**0.02**	**0.01**	**0.06**	0.37	0.07	**0.02**	21.52	4.41	2.21
	SVMOne	1.72	6.14	0.49	0.47	7.34	3.23	10.13	9.67	1.39	6.50	8.54	7.25	5.24

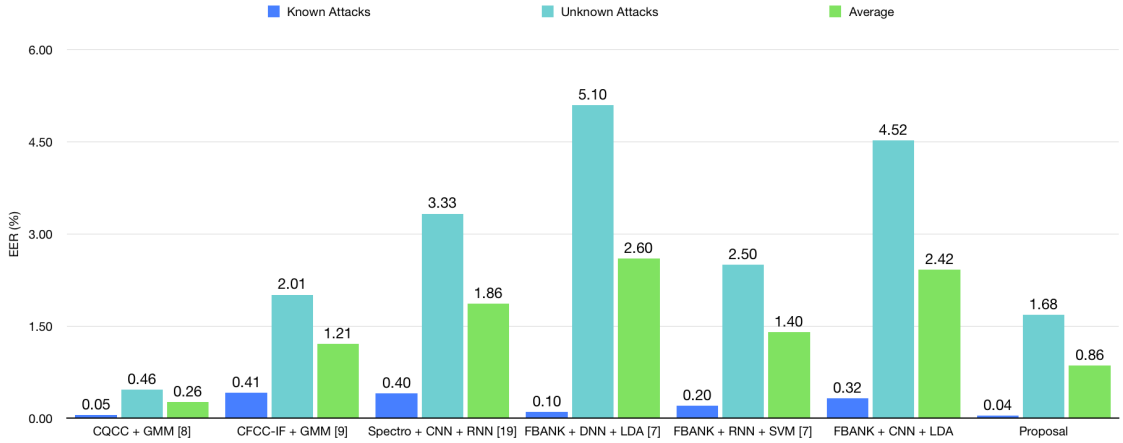

Figure 3: *Comparison on evaluation dataset for known and unknown spoofing attacks in terms of average (%) EER*

According to these results, we propose an anti-spoofing system which employs FBANK features, a CNN + RNN architecture to get the spoofing identity vector of an utterance, and an LDA classifier to make the final detection decision (spoof or genuine).

4.2. Comparative performance

A comparison of our proposal with other popular techniques from the literature are presented in Fig. 3.

The first two systems CQCC + GMM [8] and CFCC-IF + GMM [9] employ the features which perform best for spoofing detection using a Gaussian Mixture Model (GMM) as back-end. The other four systems are the most popular anti-spoofing systems based on deep learning frameworks. The FBANK + CNN + LDA system has been proposed in [11], but as its performance is not provided in this reference for the clean scenario, we have evaluated instead our proposed system removing the RNN and averaging the deep features for getting the spoofing identity vector of the utterance as in [11].

The CQCC + GMM system achieves the best average performance, although our proposed system (FBANK + CNN + RNN + LDA) achieves the best results for the known attacks. Compared to the rest of deep learning systems (Spectro + CNN + RNN [19], FBANK + DNN + LDA [7], FBANK + RNN + SVM [7] and FBANK + CNN + LDA), our proposal outperforms all of them for the known and unknown attacks. In particular, the result of our proposal for the S10 attack is quite noteworthy. Furthermore, our proposed system also achieves a lower EER in almost all attacks than the CFCC-IF system [9], performing 0.45% better on average when considering all the attacks.

Despite that the CQCC + GMM system outperforms all the systems in a clean condition training scenario, our previous work [6] and reference [11] demonstrate that CQCC + GMM performs worse than the systems based on deep features when a noisy scenario is considered. Moreover, reference [6] shows that, when a typical multicondition training for noise robustness is applied, our system, based on deep identity vectors, clearly outperforms CQCC + GMM even in clean conditions.

5. Conclusions

This paper has evaluated different features and classifiers in order to find the combination which offers the best performance for an anti-spoofing system based on a deep learning framework. The experimental results have shown that FBANK features and an LDA obtain the best performance for systems based on the extraction of deep features, rather than the popular CQCC features and other types of classifiers, such as binary SVM and SVM One-Class. Furthermore, the proposed system (FBANK + CNN + RNN + LDA) outperforms the rest of deep learning systems of the literature.

6. Acknowledgements

This work has been supported by the Spanish MINECO/FEDER Project TEC2016-80141-P and the Spanish Ministry of Education through the National Program FPU (grant reference FPU16/05490). We also acknowledge the support of NVIDIA Corporation with the donation of a Titan X GPU.

7. References

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

[2] Z. Wu et al., "Anti-spoofing for text-independent speaker verification: An initial database, comparison of countermeasures, and human performance," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 4, pp. 768–783, 2016.

[3] Liu, Y., Qian, Y., Chen, N., Fu, T., Zhang, Y., Yu, K., "Deep feature for text-dependent speaker verification", in *Speech Communication*, vol. 13, pp. 1–13, 2015.

[4] Grzl, F., Karafit, M., Kontr, S., Cernocky, J., "Probabilistic and bottle-neck feature for LVCSR of meetings," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2007, pp. 757–760.

[5] Wu, Z., King, S., "Improving trajectory modelling for dnn-based speech synthesis by using stacked bottleneck features and minimum trajectory error training," in *IEEE/ACM Trans. on Audio, Speech and Language Processing*, vol. 24, pp. 1255–1265, 2016.

[6] Alejandro Gomez-Alanis, Antonio M. Peinado, Jose A. Gonzalez, and Angel M. Gomez, "A Deep Identity Representation for Noise Robust Spoofing Detection," in *Proc. InterSpeech*, 2018.

[7] Y. Qian, N. Chen, and K. Yu, "Deep features for automatic spoofing detection," in *Speech Communication*, vol. 85, pp. 43–52, 2016.

[8] M. Todisco, H. Delgado, and N. Evans, "A new feature for automatic speaker verification anti-spoofing: Constant Q cepstral coefficients," *Proc. Odyssey*, 2016, pp. 249–252.

[9] T. B. Patel and H. A. Patil, "Combining evidences from mel cepstral, cochlear filter cepstral and instantaneous frequency features for detection of natural vs. spoofed speech," *Proc. Interspeech*, 2015, pp. 2062–2066.

[10] Muckenhirn, H., Korshunov, P., Magimai-Doss, M., Marcel, S., "Long-Term Spectral Statistics for Voice Presentation Attack Detection," in *IEEE/ACM Trans. on Audio, Speech and Language Processing*, vol. 25, pp. 2098–2111, 2017.

[11] Y. Qian, N. Chen, H. Dinkel, and Z. Wu, "Deep Feature Engineering for Noise Robust Spoofing Detection," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 25, no. 10, pp. 1942–1955, 2017.

[12] Scholkopf, B., Williamson, R. C., Smola, A. J., et al., "Support vector method for novelty detection," in *Proc. NIPS*, 2000, pp. 582–588.

[13] Jess Villalba, Antonio Miguel, Alfonso Ortega, and Eduardo Lleida, "Spoofing detection with dnn and one-class svm for the asvspoof 2015 challenge," in *Proc. InterSpeech*, 2015, pp. 2067–2071.

[14] Z. Wu, T. Kinnunen, N. Evans, J. Yamagishi, C. Hanilçi, M. Sahidullah, and A. Sizov, "ASVspoof 2015: The first automatic speaker verification spoofing and countermeasures challenge," in *Proc. InterSpeech*, 2015, pp. 2037–2041.

[15] N. Brümmer and E. deVilliers, "The BOSARIS toolkit: Theory, algorithms and code for surviving the new DCF," in *NIST SRE11 Speaker Recognition Workshop*, Atlanta, Georgia, USA, Dec. 2011, pp. 1–23.

[16] Kyunghyun Cho, et al., "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," *Proc. Empirical Methods in Natural Language Processing*, 2014, pp. 1724–1734.

[17] S. Rennie, V. Goel, and S. Thomas, "Annealed dropout training of deep networks," in *Proc. Spoken Language Technology Workshop*, 2014, pp. 159–164.

[18] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6890*, 2014.

[19] Chunlei Zhang, Chengzhu Yu, and John H. L. Hansen, "An investigation of Deep-Learning Frameworks for Speaker Verification Antispoofing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, pp. 684–694, 2017.

[20] B. C. J. Moore, "An Introduction to the Psychology of Hearing", BRILL, 2003.