End-to-End Convolutional Neural Network-based Voice Presentation Attack Detection

Hannah Muckenhirn
Idiap Research Institute, Martigny, Switzerland
École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland
hannah.muckenhirn@idiap.ch

Mathew Magimai-Doss and Sébastien Marcel
Idiap Research Institute, Martigny, Switzerland
{mathew,sebastien.marcel}@idiap.ch

Abstract

Development of countermeasures to detect attacks performed on speaker verification systems through presentation of forged or altered speech samples is a challenging and open research problem. Typically, this problem is approached by extracting features through conventional short-term speech processing and feeding them to a binary classifier. In this article, we develop a convolutional neural network-based approach that learns in an end-to-end manner both the features and the binary classifier from the raw signal. Through investigations on two publicly available databases, namely, ASVspoof and AVspoof, we show that it yields systems comparable to or better than the state-of-the-art approaches for both physical access attacks and logical access attacks. Furthermore, the approach is shown to be complementary to a spectral statistics-based approach, which, similarly to the proposed approach, does not use prior assumptions related to speech signals.

1. Introduction

Speaker verification (SV) systems aim to verify an identity claim based on an individual's voice. One of the potential applications of SV systems lies in the area of access control through user authentication. While current state-of-the-art SV systems are robust to *zero-effort* impostors, they are vulnerable to more sophisticated attacks, called presentation or spoofing attacks, consisting in presenting forged or altered speech samples as input to the system [5, 24]. The forged speech samples can be obtained by recording the target speaker's voice; synthesizing speech that carries the target speaker characteristics; or applying voice conversion methods to convert an impostor speech into the target speaker speech. Depending upon how the forged samples are presented to the SV system, there are two types of attacks: (a) physical access attacks, where the sample is fed as input to the SV system through the sensor, i.e., the microphone and (b) logical access attacks, where the sample is injected into the SV system software process, bypassing the sensor. Development of countermeasures to detect such presentation attacks is of paramount interest.

In the literature, the research has mainly focused on logical access attacks, where different approaches have been proposed to build binary classifiers that can detect attacks generated via speech synthesis and voice conversion systems. These approaches mainly differ in terms of the features and the type of classifiers used. More precisely, different magnitude spectrum based features [15, 20, 21], phase spectrum based features [15, 17, 20] and modulation spectrum based features [26], which are all computed on short-term spectrum, have been investigated in conjunction with classifiers such as Gaussian mixture models [15, 17, 21, 26], support vector machines [15] and artificial neural networks [20]. While these approaches have led to advancements in detecting attacks, there is an open issue. Standard feature extraction methods typically incorporate task specific prior speech production and speech perception knowledge on the short-term Fourier spectrum. However, unlike tasks such as speech recognition and speaker recognition, there is little or no prior knowledge about the characteristics of the signal that differentiates a forged signal from a genuine signal. Furthermore, standard speech related assumptions, such as the source filter modeling and the auditory filtering may hold well for both genuine and forged signals.

A potential approach would be to make minimum assumptions, i.e., not to rely on prior knowledge related to speech production and perception. In that direction, leveraging from recent findings in machine learning, deep architectures are being employed to learn automatically the features by using short-term processing based intermediate representations as input, such as log-scale spectrograms [29] or filter-banks [2, 14, 23]. More recently, an approach was proposed in [7], which simply uses spectral statistics estimated from the Fourier magnitude spectrum of the signal to detect presentation attacks, making thus minimum prior assumptions about the signal. Investigations on both physical access attacks and logical access attacks have shown that such an approach yields comparable or better systems than the conventional short-term speech processing based systems.

In this paper, we go one step further where, rather than transforming the speech signal from time domain to frequency domain through Fourier transform and then building classifiers, the transformation of the speech signal, the features and the classifier are learned jointly from the raw speech signal. The approach is fundamentally motivated by recent advances in deep learning-based speech processing, where both the features and the classifiers are jointly learned from the raw speech signal [9, 16, 22, 28]. Specifically, in the proposed approach the suitable segmental processing of the input signal is determined during training and the features and classifier are learned in a data- and task-driven manner for presentation attack detection (PAD). Through experimental studies on two databases, namely, AVspoof and ASVspoof databases, we show that the proposed approach with a single convolution layer is able to detect both physical access attacks and logical access attacks well, and yields performance comparable to or better than the approaches proposed in the literature.

Section 2 presents the CNN-based approach. Section 3 and Section 4 presents the experimental setup and the results on the two databases, respectively. Section 5 presents an analysis of the information learned by the CNN and Section 6 concludes the paper.

2. Proposed approach

We follow the CNN-based end-to-end acoustic modeling approach originally proposed for automatic speech recognition in [9] and developed further in [10, 11]. In this approach, as illustrated in Fig. 1, the CNN consists of a feature stage modeled by N convolution layers followed by a classification stage modeled by a multilayer perceptron (MLP). In our studies, the feature stage consists of one convolution layer (N = 1) and the classifier stage consists of an MLP with a single hidden layer or a single layer perceptron (SLP), i.e., no hidden layer. The motivation behind such a simple architecture choice comes from the work in [7], where the speech is transformed once through Fourier transform and the first order and second order statistics of the magnitude spectrum estimated over the utterance are classified using a linear discriminant analysis classifier or a MLP with a single hidden layer. In comparison to that, we could interpret the convolution layer as a transformation of the signal that is learned from the data in a task driven manner, as opposed to the Fourier transform, and the classification stage as a linear classifier in the case of SLP and as a non-linear classifier in the case of MLP. Furthermore, we do not perform any max-pooling as we experimentally observed that it did not improve the performance of the system.

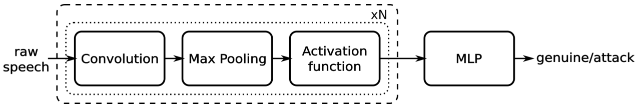


Figure 1: Diagram of a Convolutional Neural Network.

Each speech sample is split into blocks of length w_{seq} ms and shifted by w_{shift} ms. Each block is fed successively to the CNN, i.e., the CNN outputs one score per block. Fig. 2 shows the processing carried out in the convolution layer. Specifically, the convolution layer consisting of n_f filters processes a block of signal of length w_{seq} ms in short segments based on the length of the filters kW (kernel width) and shift dW. The output of the filters are fed to an activation function, which is a hard hyperbolic tangent function in this case, i.e.,

$$f(x) = \begin{cases} 1, & \text{if } x > 1 \\ -1, & \text{if } x < -1 \\ x, & \text{otherwise.} \end{cases}$$

The output of the activation function is subsequently fed to the classifier stage, which in the case of MLP has a hidden layer composed of n_{hu} hidden units followed by hard hyperbolic tangent activation function. The output layer of the MLP is a softmax layer composed of two units corresponding to the genuine class and the attack class. The parameters of the classifier and feature stages are randomly initialized and trained via the stochastic gradient descent algorithm using a cross entropy optimization criterion. The hyper parameters w_{shift}, w_{seq}, kW, dW, n_f and n_{hu} are determined based on the frame-level error rate computed over a development set.

3. Experimental setup

3.1. Databases and protocols

We validate the proposed approach on two databases: the Audio-Visual Spoofing (AVspoof) database, which contains both logical and physical attacks and the Automatic Speaker Verification Spoofing (ASVspoof) database, which contains only logical attacks. In the remainder of this section, we describe these databases as well as their evaluation protocols.

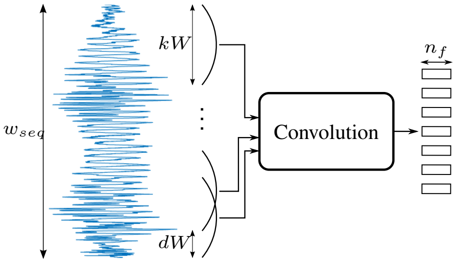


Figure 2: Illustration of the convolution layer processing.

3.1.1 AVspoof

The AVspoof database[1] contains replay attacks, as well as speech synthesis and voice conversion attacks both produced via logical and physical access. In the remainder of the paper, we call "AVspoof-LA" the subset containing genuine samples and logical access attacks and "AVspoof-PA" the subset containing genuine samples and physical access attacks. This database contains the recording of 31 male and 13 female participants divided into four sessions. Each session is recorded in different environments and different setups. The attacks are played with four different loudspeakers. For the replay attacks, the original samples are recorded with four different microphones.

3.1.2 ASVspoof

The ASVspoof[2] database contains genuine and spoofed samples from 45 male and 61 female speakers. This database contains only speech synthesis and voice conversion attacks produced via logical access, i.e., directly injected in the system. The attacks in this database were generated with 10 different speech synthesis and voice conversion algorithms. Only 5 types of attacks are in the training and development set (S1 to S5), while 10 types are in the evaluation set (S1 to S10). This allows to evaluate the systems on known and unknown attacks. The full description of the database and the evaluation protocol are given in [25]. This database was used for the ASVspoof 2015 Challenge and is a good basis for system comparison as several systems have already been tested on it.

3.1.3 Evaluation protocols

Both databases are divided into three subsets, each containing a set of non-overlapping speakers: the training, development and evaluation set, presented in Table 1. However, we do not use the same evaluation protocol for the two databases. On the AVspoof database, the development set is used to choose the threshold as to obtain an Equal Error Rate (EER), i.e., the false alarm rate and the miss rate are equal. Then, the performance is evaluated by computing the Half Total Error Rate (HTER) on the evaluation set. On the other hand, to evaluate our system on the ASVspoof database, we follow the evaluation protocol used during the ASVspoof 2015 Challenge to be able to compare to other systems. In both the development and evaluation set, the threshold is fixed independently for each type of attack with the EER criterion. Then, the performance of the system is evaluated by averaging the EER over the known attacks (S1-S5), the unknown attacks (S6-S10) and all the attacks (S1-S10).

Table 1: Number of speakers and utterances for each set of AVspoof and ASVspoof databases: training, development, evaluation.

Database	**set**	**speakers**		**utterances**		
		male	female	genuine	LA attacks	PA attacks
AVspoof	train	10	4	4973	17890	38580
	dev	10	4	4995	17890	38580
	eval	11	5	5576	20060	43320
ASVspoof	train	10	15	3750	12625	–
	dev	15	20	3497	49875	–
	eval	20	26	9404	184000	–

Table 2: Hyper-parameters of the CNN trained on the three datasets: AVspoof-PA, AVspoof-LA and ASVspoof.

	w_{shift} (ms)	w_{seq} (ms)	kW (samples)	dW (samples)	n_f	n_{hu}
AVspoof-PA	10	310	300	100	20	–
	10	310	300	10	20	100
AVspoof-LA	10	310	300	100	100	–
	10	310	300	100	20	20
ASVspoof	10	310	300	100	100	–
	10	310	300	100	20	2000

3.2. Systems

Before feeding the raw speech signal to the CNN, we (a) perform an energy-based voice activity detection to remove the silence parts at the beginning and end of the utterances and (b) normalize the signal in each segment of the kernel width kW by its mean and variance, as done in the earlier work on speech recognition [9, 11]. As detailed in Section 2, there are six hyper-parameters that need to be set: w_{shift}, w_{seq}, kW, dW, n_f and n_{hu}. These hyper-parameters are chosen based on the frame-level accuracy achieved on the development set during the training phase. Table 2 presents the values of these hyper-parameters for each dataset: AVspoof-LA, AVspoof-PA and ASVspoof, found through a coarse grid search. In the case of SLPs, there is no n_{hu}.

In addition to studying the proposed CNN-based approach as a stand alone system, we also study the score

[1] https://www.idiap.ch/dataset/avspoof
[2] http://dx.doi.org/10.7488/ds/298

Table 3: HTER (%) of PAD systems on AVspoof, separately trained for the detection of Physical Access (PA) and Logical Access (LA) attacks. Evaluation set.

	LFCC	RFCC	LTSS	CNN	CNN	Comb	Comb
	[4]	[4]	[6]	SLP	MLP	×	+
LA	**0.00**	0.03	0.04	0.07	0.02	**0.00**	**0.00**
PA	5.00	2.70	0.18	0.11	**0.09**	**0.09**	**0.09**

level combination of the CNN-based system with *off-the-shelf* long term spectral statistics (LTSS)-based system developed in [7] and further investigated in [6]. More precisely, we combine the probabilities of genuine and attack classes estimated by the two systems through the product and the sum combination rule [19].

The development of the CNN-based system was done using Torch7 toolkit [3]. All the experiments are reproducible[3].

4. Results

In this section, we present the results obtained on the three datasets: AVspoof-PA, AVspoof-LA and ASVspoof.

4.1. AVspoof database

Table 3 presents the results of the proposed approach compared to the best systems reported in [4] and [6]. These systems correspond respectively to standard spectral features-based approaches using GMMs (denoted as LFCC and RFCC) and a spectral statistics-based approach using LDA classifier (denoted as LTSS) on AVspoof-PA and AVspoof-LA. LFCC and RFCC stand respectively for Linear and Rectangular Frequency Cesptral Coefficients and refer to cepstral features estimated by placing triangular shaped filters and rectangular shaped filters on a linear scale [15]. Comb (×) and Comb (+) denote the combination of the LTSS system and the CNN MLP system based on the product rule combination and the sum rule combination, respectively.

We can observe that the CNN-based approach yields performance comparable to or better than systems reported in the literature. The CNN MLP system performs slightly better than the CNN SLP system on both physical and logical access attacks. The combination with LTSS leads to a slight improvement for logical access attacks.

4.2. ASVspoof database

Table 4 presents the results per type of attack, over the known attacks (S1-S5), the unknown attacks (S6-S10) and all the attacks, and compares the performance achieved with the proposed end-to-end CNN-based approach with the performances reported in the literature. Systems A-E correspond to the five better performing systems of Interspeech 2015 ASVspoof challenge. LFCC features resulted in the best system in [4]. Constant Q Cepstral Coefficients (CQCC)-based system resulted in the best overall system in [21]. $\{$DNN,RNN$\}$ corresponds to the best system obtained in [14], which is a score-level fusion of features learned with a Deep Neural Network (DNN) and classified with a LDA and features learned with a Recurrent Neural Network (RNN) and classified with a support vector machine. In both cases the features are learned from filter bank energies. The system $\{$CNN,RNN,CNN+RNN$\}$ was developed in [29] and is a score-level fusion of a CNN, a RNN and a combined CNN and RNN, all trained on the log-scale spectrogram of the speech utterances. In the LTSS-based approach, the LDA based system yielded the best system for the known attacks condition and the MLP-based system performed the best on the unknown attacks condition [6]. Thus, we present the two LTSS-based systems.

On S1-S9 conditions, the CNN SLP system yields a performance comparable to the systems reported in the literature. On S10 condition, the performance is worse than the systems compared. With CNN MLP, i.e., with the use of a hidden layer, the S10 performance improves but is still significantly high. Overall, CNN SLP system performs comparable to or better than CNN MLP system, except for S10. A classifier fusion of CNN SLP and LTSS MLP using product combination rule, denoted as $\{$CNN SLP $\times$ LTSS MLP$\}$, and sum combination rule, denoted as $\{$CNN SLP $+$ LTSS MLP$\}$, yields one of the best systems on both known and unknown attacks. Together with the investigations on AVspoof database, this indicates that the proposed CNN-based approach is complementary to the LTSS-based approach.

5. Analysis and discussion

In this section, we first analyze the frequency response of the filters learned by the CNN on the three datasets. We then compare our system to the one developed in [9] for speech recognition. Finally, we compare our approach to PAD systems using deep architectures as a feature stage classifier.

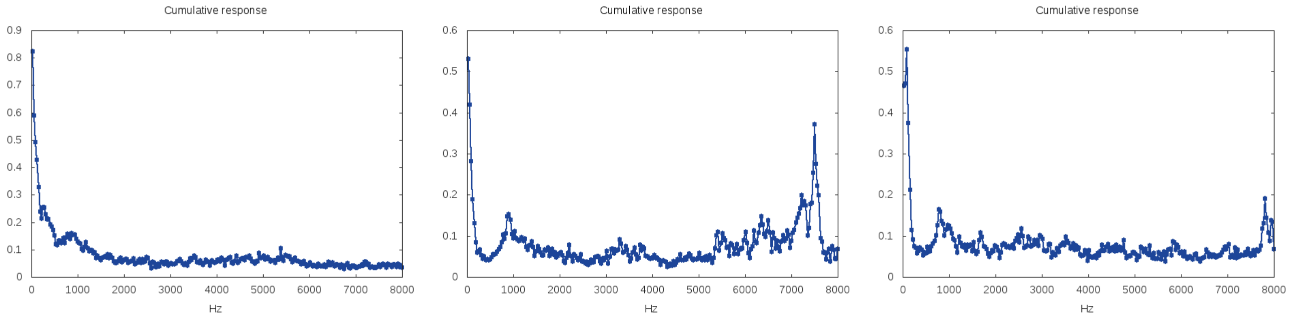
5.1. Analysis of convolution filters

The proposed CNN-based approach performs well on AVspoof-PA, AVspoof-LA and ASVspoof (except for S10). One of the question that arises is: what is being learned by the filters in the convolution layer? In the earlier work on automatic speech recognition, it was found that the filters in the first convolution layer model the spectrum “in-parts”. One way to understand the manner in which different parts of the spectrum are modeled is to observe the cumulative frequency response of the learned filters [10, 11]. Thus, we analyzed the filters by computing the 512-points FFT of each filter in the CNN-based system and calculating the

[3] https://gitlab.idiap.ch/hmuckenhirn/CNN-voice-PAD

Table 4: EER (%) per type of attack computed on the ASVspoof database. Evaluation set.

System	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	known	unknown	all
A [12]	0.10	0.86	0.00	0.00	1.08	0.85	0.24	0.14	0.35	8.49	0.408	2.013	1.211
B [8]	0.00	0.02	0.00	0.00	0.01	0.02	0.00	0.02	0.00	19.57	0.008	3.922	1.965
C [2]	-	-	-	-	-	–	-	-	-	-	0.058	4.998	2.528
D [27]	0.0	0.0	0.0	0.0	0.01	0.01	0.0	0.0	0.0	26.1	0.003	5.231	2.617
E [1]	0.024	0.105	0.025	0.017	0.033	0.093	0.011	0.236	0.000	26.393	0.041	5.347	5.694
LFCC [4]	0.032	0.500	0.000	0.000	0.126	0.151	0.011	0.234	0.032	5.561	0.132	1.198	0.665
CQCC [21]	0.005	0.106	0.000	0.000	0.130	0.098	0.064	1.033	0.053	1.065	0.048	0.463	0.256
{DNN,RNN} [14]	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	10.7	0.0	2.2	1.1
{CNN,RNN,CNN+RNN} [29]	0.09	0.29	0.00	0.00	0.99	0.64	0.71	0.00	0.29	11.67	0.27	2.66	1.47
LTSS, LDA [6]	0.000	0.043	0.000	0.000	0.086	0.086	0.022	0.086	0.032	10.218	0.026	2.089	1.058
LTSS, MLP [6]	0.011	0.151	0.000	0.000	0.352	0.288	0.054	0.043	0.065	1.564	0.103	0.403	0.253
CNN SLP	0.011	0.032	0.032	0.032	0.032	0.032	0.032	0.032	0.032	30.276	0.028	6.081	3.054
CNN MLP	0.011	0.043	0.043	0.043	0.043	0.043	0.043	0.054	0.043	28.572	0.037	5.751	2.894
{CNN SLP × LTSS MLP}	0.000	0.000	0.000	0.000	0.000	0.011	0.000	0.000	0.008	1.553	**0.000**	**0.314**	**0.157**
{CNN SLP + LTSS MLP}	0.000	0.022	0.000	0.000	0.032	0.022	0.011	0.008	0.022	1.540	0.011	0.320	0.166



(a) ASVspoof

(b) AVspoof-LA

(c) AVspoof-PA

Figure 3: Cumulative frequency response of the convolution filters learned on the ASVspoof, AVspoof-LA and AVspoof-PA databases.

cumulative frequency response by summing the magnitude spectra.

Fig. 3a and 3b shows the gain normalized cumulative frequency response of CNN MLP systems for ASVspoof and AVspoof-LA, which contain both logical access attacks. It can be observed that in both cases the filters lay emphasis on very low frequencies, with the maximum response lying at 0 Hz. This contradicts the work in [13, 18], in which the authors observed on the ASVspoof database that high frequency regions were more discriminative. However, this is consistent with observations made in earlier works [6, 21]. Specifically, in [6] the analysis of the LDA weights of the LTSS-based system showed that more importance is given to very low frequency regions than high frequency regions. Furthermore, in [6] and [21] it was found that high frequency resolution is needed to generalize the system to the unseen S10 attack condition, which is based on concatenative speech synthesis as opposed to statistical parametric speech synthesis systems used in seen conditions. So one plausible reasoning for our CNN-based approach to perform worse on S10 condition is that the filters that model about 20ms signal (kW = 300 samples) are not able to capture the highly localized discriminative information for the S10 condition in the frequency domain. The CNN-MLP studies indicate that the generalization may not be improved by just increasing the classifier stage complexity. A potential solution would be to find alternative CNN architectures, e.g., longer kernel widths, more convolution layers, or to combine with complementary approaches, as demonstrated in this article.

Fig. 3c shows the gain normalized cumulative frequency response of the filters in the CNN MLP system for AVspoof-PA. In comparison to AVspoof-LA, AVspoof-PA training mainly differs on two aspects: (a) training with replay attacks in addition to synthetic speech and voice conversion attacks and (b) the speech is input through the sensor, which can have an additional channel effect. As it can be seen from Table 2, in the case of both AVspoof-PA and AVspoof-LA, $kW = 300$ and $n_f = 20$. As a consequence, there are similarities between cumulative frequency response of

AVspoof-PA and AVspoof-LA, see Fig. 3 (b), with subtle differences at low frequencies and high frequencies. In particular, the maximum response lies at a non-zero frequency ($\approx$ 60 Hz). Taken together, these observations indicate that the CNN is capturing somewhat different discriminative information in the frequency domain for physical access attacks and logical access attacks. Understanding these differences together with the role of dW and kW and the output of the filters is part of our future work.

5.2. Comparison to deep learning based approaches

The proposed end-to-end PAD approach is inspired from the works of Palaz et al. [9, 11], originally developed for automatic speech recognition. So a fundamental question that arises is: are there any differences? At the architectural level, there are several differences. Firstly, for PAD we observe that the kernel width kW is longer than the kW for speech recognition. Palaz et al. found that using $kW = 30$ samples, corresponding to $\approx$ 2ms, at the first convolution layer was optimal for speech recognition while for our task the optimal value is $kW = 300$ samples, corresponding to $\approx$ 20ms, which is in the order of the frame size conventionally used to derive short-term spectral features. This indicates that the speech signal needs to be processed differently for PAD and speech recognition in the time domain. Secondly, the architecture employed in this paper is much simpler: only one convolution layer, no max-pooling while for speech recognition Palaz et al. found that at least three convolution layers along with max-pooling at each layer was needed. We performed experiments by adding more convolution layers and employing max-pooling at the output of each convolution layer, we did not observe any gains. This suggests that an architecture with low complexity is sufficient for PAD.

Another point of comparison is the information captured by the convolution filters. As presented in Section 5.1, the filters for PAD are modeling highly localized frequency information. On the other hand, Palaz et al. found that the filters learned at the first convolution layer give emphasis to the telephone bandwidth and frequencies above 6000 Hz [11]. This indicates that, even though the end-to-end methodology is inspired from speech recognition, what is learned by the CNN is indeed task specific.

The approaches proposed in [2, 14, 23, 29] employ deep learning methods for PAD. The main difference with our approach is that all these approaches use an intermediate representation of the speech signal as input: log-scale spectrograms or filter-banks output. Furthermore, these approaches employ deep architectures with multiple hidden or convolution layers to extract and model the information suitable from the intermediate representations for PAD. In contrast, our system contains only 1 convolution layer, with 20 or 100 filters that directly operates on the raw speech signal and has at the maximum one hidden layer. Table 4 presents the results of [2, 14, 29]. It is interesting to observe that our approach with a SLP (in the classifier stage) performs comparable or better than these systems, except on the S10 attack. This indicates that learning feature and classifier directly from the raw speech signal in an end-to-end manner for PAD is beneficial and is worth pursuing.

6. Conclusions

In this paper, we proposed an approach that makes minimal assumptions, and learns the relevant features and classifier from the raw speech signal in an end-to-end manner for PAD using CNNs. Our investigations showed that with a simple architecture, a single convolution layer in the feature stage and a SLP or a MLP with one hidden layer in the classifier stage, the approach performs well for both physical access and logical access attacks and yields systems comparable to or better than the state-of-the-art approaches, which are largely based on standard short-term speech processing. Furthermore, the proposed approach is complementary to the LTSS-based approach. In the present study our analysis was limited to filter frequency responses. Our future work will focus on analyzing how these filters respond together to the input speech signal to gain better understanding about the signal characteristics that differentiate genuine speech from attacks.

Acknowledgment

This work was partially funded by the Swiss National Science Foundation through the project UniTS and the Research Council of Norway through the project SWAN. The authors would like to thank their ex-colleague Dimitri Palaz for his help in setting up the Torch based system for presentation attack detection.

References

[1] M. J. Alam, P. Kenny, G. Bhattacharya, and T. Stafylakis. Development of CRIM system for the automatic speaker verification spoofing and countermeasures challenge 2015. In *Proc. of Interspeech*, 2015.

[2] N. Chen, Y. Qian, H. Dinkel, B. Chen, and K. Yu. Robust deep feature for spoofing detection - the SJTU system for ASVspoof 2015 challenge. In *Proc. of Interspeech*, 2015.

[3] R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A Matlab-like Environment for Machine Learning. In *BigLearn, NIPS Workshop*, 2011.

[4] P. Korshunov and S. Marcel. Cross-database evaluation of audio-based spoofing detection systems. In *Proc. of Interspeech*, 2016.

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

[6] H. Muckenhirn, P. Korshunov, M. Magimai.-Doss, and S. Marcel. Long term spectral statistics for voice presentation attack detection. *To appear in IEEE Transactions on Audio, Speech, and Language Processing*, 2017.

[7] H. Muckenhirn, M. Magimai.-Doss, and S. Marcel. Presentation attack detection using long-term spectral statistics for trustworthy speaker verification. In *Proc. of International Conference of the Biometrics Special Interest Group (BIOSIG)*, Sept. 2016.

[8] S. Novoselov, A. Kozlov, G. Lavrentyeva, K. Simonchik, and V. Shchemelinin. STC anti-spoofing systems for the ASVspoof 2015 challenge. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[9] D. Palaz, R. Collobert, and M. Magimai.-Doss. Estimating phoneme class conditional probabilities from raw speech signal using convolutional neural networks. In *Proc. of Interspeech*, 2013.

[10] D. Palaz, M. Magimai.-Doss, and R. Collobert. Analysis of CNN-based speech recognition system using raw speech as input. In *Proc. of Interspeech*, 2015.

[11] D. Palaz, M. Magimai.-Doss, and R. Collobert. End-to-end acoustic modeling using convolutional neural networks for automatic speech recognition. Idiap-RR Idiap-RR-18-2016, Idiap, 6 2016. http://publications.idiap.ch/downloads/reports/2016/Palaz_Idiap-RR-18-2016.pdf.

[12] 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. In *Proc. of Interspeech*, 2015.

[13] D. Paul, M. Pal, and G. Saha. Spectral features for synthetic speech detection. *IEEE Journal of Selected Topics in Signal Processing*, 2017.

[14] Y. Qian, N. Chen, and K. Yu. Deep features for automatic spoofing detection. *Speech Communication*, 85:43–52, 2016.

[15] M. Sahidullah, T. Kinnunen, and C. Hanilçi. A comparison of features for synthetic speech detection. In *Proc. of Interspeech*, 2015.

[16] T. N. Sainath, R. J. Weiss, A. Senior, K. W. Wilson, and O. Vinyals. Learning the speech front-end with raw waveform CLDNNs. In *Proc. of Interspeech*, 2015.

[17] J. Sanchez, I. Saratxaga, I. Hernaez, E. Navas, D. Erro, and T. Raitio. Toward a universal synthetic speech spoofing detection using phase information. *IEEE Transactions on Information Forensics and Security*, 10(4):810–820, 2015.

[18] K. Sriskandaraja, V. Sethu, P. N. Le, and E. Ambikairajah. Investigation of sub-band discriminative information between spoofed and genuine speech. In *Proc. of Interspeech*, 2016.

[19] D. M. Tax, M. van Breukelen, R. P. Duin, and J. Kittler. Combining Multiple Classifiers by Averaging or by Multiplying? *Pattern Recognition*, 33(9):1475 – 1485, 2000.

[20] X. Tian, Z. Wu, X. Xiao, E. S. Chng, and H. Li. Spoofing detection from a feature representation perspective. In *Proc. of ICASSP*, 2016.

[21] M. Todisco, H. Delgado, and N. Evans. A new feature for automatic speaker verification anti-spoofing: Constant Q cepstral coefficients. In *Proc. of Odyssey*, 2016.

[22] G. Trigeorgis, F. Ringeval, R. Brueckner, E. Marchi, M. A. Nicolaou, B. W. Schuller, and S. Zafeiriou. Adieu features? End-to-end speech emotion recognition using a deep convolutional recurrent network. In *Proc. of ICASSP*, 2016.

[23] J. Villalba, A. Miguel, A. Ortega, and E. Lleida. Spoofing detection with DNN and one-class SVM for the ASVspoof 2015 challenge. In *Proc. of Interspeech*, 2015.

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

[25] Z. Wu, T. Kinnunen, N. W. D. Evans, J. Yamagishi, C. Hanilçi, M. Sahidullah, and A. Sizov. ASVspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge. In *Proc. of Interspeech*, 2015.

[26] Z. Wu, X. Xiao, E. S. Chng, and H. Li. Synthetic speech detection using temporal modulation feature. In *Proc. of ICASSP*, 2013.

[27] X. Xiao, X. Tian, S. Du, H. Xu, E. S. Chng, and H. Li. Spoofing speech detection using high dimensional magnitude and phase features: The NTU approach for ASVspoof 2015 challenge. In *Proc. of Interspeech*, 2015.

[28] R. Zazo, T. N. Sainath, G. Simko, and C. Parada. Feature learning with raw-waveform CLDNNs for voice activity detection. In *Proc. of Interspeech*, 2016.

[29] C. Zhang, C. Yu, and J. H. Hansen. An investigation of deep learning frameworks for speaker verification anti-spoofing. *IEEE Journal of Selected Topics in Signal Processing*, 11(4):684–694, 2017.