END-TO-END SPOOFING DETECTION WITH RAW WAVEFORM CLDNNS

Heinrich Dinkel, Nanxin Chen, Yanmin Qian, Kai Yu

Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering
SpeechLab, Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai, China
{heinrich.dinkel,bobchennan}@gmail.com,{yanminqian,kai.yu}@sjtu.edu.cn

ABSTRACT

Albeit recent progress in speaker verification generates powerful models, malicious attacks in the form of spoofed speech, are generally not coped with. Recent results in ASVSpoof2015 and BTAS2016 challenges indicate that spoof-aware features are a possible solution to this problem. Most successful methods in both challenges focus on spoof-aware features, rather than focusing on a powerful classifier. In this paper we present a novel raw waveform based deep model for spoofing detection, which jointly acts as a feature extractor and classifier, thus allowing it to directly classify speech signals. This approach can be considered as an end-to-end classifier, which removes the need for any pre- or post-processing on the data, making training and evaluation a streamlined process, consuming less time than other neural-network based approaches. The experiments on the BTAS2016 dataset show that the system performance is significantly improved by the proposed raw waveform convolutional long short term neural network (CLDNN), from the previous best published 1.26% half total error rate (HTER) to the current 0.82% HTER. Moreover it shows that the proposed system also performs well under the unknown (RE-PH2-PH3,RE-LPPH2-PH3) conditions.

Index Terms— CLDNN, End-to-End, BTAS2016, Spoofing detection

1. INTRODUCTION

Biometric recognition is a broad field which has developed from the classic fingerprint over to face recognition and nowadays speech can be used to naturally to restrict access to a certain medium. The research field which focuses on protecting the integrity of this speech based process is called speaker verification (SV). The main purpose of speaker verification is to detect whether the (real) speaker, who registered himself with the system, produces an utterance to grant access to the system or if that utterance was produced by an impostor. Malignant spoofing attacks mimic real speakers characteristics, thus an unprepared system's performance degrades heavily, when exposed to spoofing attacks [1, 2]. Traditional SV systems are not aware of possible spoofing attacks, which can be threatened by *direct* attacks (also called *spoofing* attacks). These attacks either artificially or naturally produce a spoofed utterance and try to gain access to a system. This work focuses only on the prevention of direct attacks. Overall there are currently four known *direct* attacks (Impersonation, Replay, Synthesis, Voice conversion).

Building an appropriate feature representation and designing a suitable classifier for each of the attack types are seen as separate problems, with different approaches for a suitable solution. One of the general disadvantages is that these features might not be optimal for the classification succeeding the classification task. Our motivation stems from recent advances in anti-spoofing research, which shows that an appropriate feature - independent of the classifier - contributes to prevent spoof attempts of a speaker verification system. In this context, deep neural networks can be seen as a joint classification and feature extraction framework, that aim to generate a feature representation which incorporates all relevant *direct* attack types.

The remainder of this paper is organized as follows. At first Section 2 reviews previous work in the context of spoofing detection. Continuing with Section 3, which describes the CLDNN architecture with raw wave input. Section 4 describes our experimental setup, model parameters, used datasets and demonstrates the results while comparing the CLDNN approach with other neural network anti-spoof techniques. A conclusion is given in Section 5.

2. PREVIOUS WORK

2.1. Model

Previous models include the ever so popular *i*-vector [3, 4, 5], which when used as a standalone model does not perform

This work was supported by the Shanghai Sailing Program No. 16YF1405300, the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning, the China NSFC projects (No. 61573241 and No. 61603252) and the Interdisciplinary Program (14JCZ03) of Shanghai Jiao Tong University in China

well. Another popular approach is to use the traditional GMM model. In this approach two different GMMs are trained representing the *genuine* (M_g) and *spoof* (M_s) labels respectively. Each GMM only uses the respective training data. After training, the score for a given evaluation utterance $\mathbf{x}$ can be calculated as follows:

$$\text{score}(\mathbf{x}) = \log P(\mathbf{x}|M_g) - \log P(\mathbf{x}|M_s) \quad (1)$$

Here $P(\mathbf{x})$ is assumed to be Gaussian distributed. Successful attempts can be seen in [6, 7]. Deep features were also employed and achieved remarkable results. Chen et al. [8, 9], fed PLP features into a neural network and obtained a high-dimensional representation vector. This vector is then used as basis for future classification. In recent work [10], sequence models such as recurrent neural networks - long short term memory (RNN-LSTM) were incorporated to extract features.

2.2. Features

As previous works indicate, feature extraction is crucial in order to detect possibly malicious system accesses. Research towards detecting synthesized spoofing attempts shows that this type of speech generally generates artifacts that can be detected by features that use phase spectrum information. It was seen that phase spectrum based features (e.g. MGDF) [11] seem to discriminate better than common magnitude ones. Moreover, the recently published constant Q cepstral coefficient (CQCC) feature can be seen as the current state-of-the-art, being capable to detect synthesis based attacks [6]. Furthermore, deep feature approaches apply neural networks for feature extraction [12]. Deep feature frameworks use neural networks to achieve a high abstract representation of input frames [8], in order to extract features from one of its hidden layers, which are then scored by using an independent classifier (e.g. GMM, SVM, LDA).

3. RAW WAVEFORM CLDNN ARCHITECTURE

The CLDNN architecture (Convolutional LSTM Deep Neural Network) combines three different types of neural networks into a single model. This model obtains an input in form of a sequence of frames and outputs a likelihood for the whole sequence. The CLDNN performs time-frequency convolution to reduce spectral variance, long-term temporal modeling by using a LSTM, and classification using a DNN.

CLDNN was already successfully employed in automatic speech recognition tasks (ASR) [13]. In contrast to the more common approach to use log mel features, this approach uses raw waveforms as input. We argue that it might be more beneficial for the network to directly learn the time-frequency transformation process on top of being able to retain all information present within the time-domain, instead of having an already preprocessed, abstract, log-mel spectrum domain as input. Thus the network can learn dependencies between adjacent frames in the time domain and it's corresponding frequency domain transformation. In this work, a raw waveform based CLDNN is proposed for the spoofing detection, and the model architecture is specified as Figure 1. In this model the convolution is applied over a sequence of input features $[x_1, \ldots, x_t, \ldots, x_S]$. The convolutional layers are adjusted to share their parameters over the whole sequence with length S.

The first layer in this architecture is a time-convolutional layer over the raw time-domain waveform which can be thought of as a finite impulse-response filter bank followed by a non-linearity (Rectified Linear Unit) [14]. After time convolution, non overlapping max pooling is applied to remove any time variance, thus collapsing all of the input samples (N) into a single value. Having collapsed the time-impulse to a smaller representation vector, frequency convolution follows the time convolution to reduce the phase variations within the time-filtered signal. After frequency convolution, the output for each time-step t is then fed into a two layer LSTM, which outputs a sequence of fixed sized vector representations. In this paper, the last time-step of the sequence is picked to obtain a single vector representation, which then is fed into the neural network classifier, as shown in Figure 1. DNN, CNN and the LSTM are jointly trained within the network. The last layer of the DNN utilizes softmax activation that normalizes the outputs to sum to one. During training a sequence of samples of length S is taken from each utterance and fed it into the network. Evaluation is performed by inputting a whole utterance at a time into the network which produces likelihood scores at the output layer for each class. In this task, four output neurons (each for one attack type (SS, VC, RE) plus the genuine speakers) are used. Thus, the final score is obtained by taking the log-likelihood values which correspond to the genuine class as scores. Thus larger scores correspond to genuine speakers, lower scores correspond to spoofed speech.

3.1. Normalization

The input of the network is directly taken from the raw waveform of a given audio utterance. One utterance with L samples will be cut into same sized pieces of length N. Pieces with length $< N$ will be removed. It is important to note that no voice activity detection is utilized, because artificially created speech sometimes has unusually long silenced segments, which when removed, can degrade the model's performance. In this work, mean and standard deviation normalization are applied onto the extracted raw waves, resulting in unit mean and zero variance within the input data.

3.2. Model specification

The front-end of our framework is comprised of two convolutional layers that aim to invariantly transform the signal

in the time and frequency domain [15]. After each convolution batch normalization is used and followed by a non-overlapping maxpooling. Throughout the network, rectified linear unit (ReLU) is used as activation function. The model is further extended by dropout [16] a probability of 50%, between each linear layer in the classifier, as well as after each LSTM layer. The number of time-kernel convolution filters is set as 39 (comparable to MFCC/PLP), the sequence length S describes the number of following frames which are fed into the CLDNN.

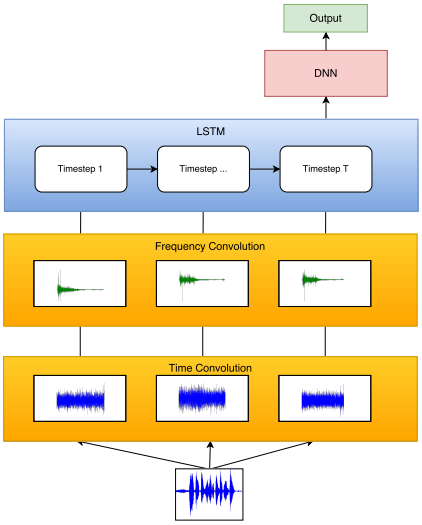

Fig. 1: The architecture of the CLDNN with raw waveform

As optimization method it was decided to use an adaptive learning algorithm, adadelta [17]. As described in [18], using a larger frame size $N = 560$ while keeping the kernel width at 400 is beneficial for the final performance. Moreover we use a stride of 160 (10 ms) as step for the time-convolution filter.

4. EXPERIMENTS

4.1. Dataset

In this paper, the focus lies on the BTAS2016 dataset [19], having overall $43{,}553$ utterances of training, $43{,}575$ utterances of development and $50{,}496$ utterances of evaluation data. The emphasis of this dataset lies on replay attacks, which include low and high quality laptop as well as phone recordings. Unknown replay attacks were also recorded on laptop and phone devices, but differ from these used in the training set.

4.2. Evaluation Protocol

First, a model is trained on the provided training data. After training, the development utterances are inserted into the system and scores for each utterance are obtained, which are consequently used to compute the FAR and FRR. The *false acceptance rate* (FAR) and the *false rejection rate* (FRR) are both metrics, which depend on a certain threshold θ:

$$\mathrm{FAR}(\theta) = \frac{|\mathrm{score}_{\mathrm{attack}} \geq \theta|}{|\mathrm{score}_{\mathrm{attack}}|}, \mathrm{FRR}(\theta) \quad = \frac{|\mathrm{score}_{\mathrm{real}} < \theta|}{|\mathrm{score}_{\mathrm{real}}|} \tag{2}$$

As evaluation metric, the *half total error rate* (HTER) [20] is used. The threshold of the development data θ_{dev} is utilized, in order to calculate the FRR and FAR of the evaluation data:

$$\theta_{dev} = \arg\min_{\theta} \left(\frac{\mathrm{FAR}_{\mathrm{dev}}(\theta) + \mathrm{FRR}_{\mathrm{dev}}(\theta)}{2} \right) \tag{3}$$

$$\mathrm{HTER}_{\mathrm{eval}} = \frac{\mathrm{FAR}_{\mathrm{eval}}(\theta_{\mathrm{dev}}) + \mathrm{FRR}_{\mathrm{eval}}(\theta_{\mathrm{dev}})}{2} \tag{4}$$

4.3. Baseline

The baseline uses a standard GMM approach, which is trained using the procedure defined in Section 2.1. In traditional SV tasks, feature extraction is assisted by voice activity detection (VAD). In light of spoofing detection, VAD is not applied, since SS and VC methods tend to create unnaturally long silent segments. Thus, silence is a key factor in determinating artificially created speech (SS,VC categories).

Parameter	Value
Windowsize	25 ms
Windowshift	10 ms
Static dimension	13 (12 ceps + power)
Normalization	Cepstral mean + var
Dynamic dimension	static + Δ + $\Delta\Delta$ = 39

Table 1: PLP parameters

The GMM baseline uses 512 Gaussian components. The training procedure is the same as described in Section 2.1 and uses the features Table 1. The baseline GMM and the formal published BTAS2016 challenge results are shown in Table 2.

Placing	Model	Feature	HTER
Baseline	GMM	PLP-39	2.96
3rd	BLSTM-DNN	PLP-39	2.20
2nd	LDA	Spectral-M-V	2.04
1st	GMM	MFCC+i-MFCC	**1.26**

Table 2: Previous results for BTAS2016 [19]

4.4. CLDNN - Setup

In our experiments we see that the features extracted from the front-end CNN are generally rich enough in information

content. We adapt the same architecture as seen in [18], but do tune our model to fit the dataset better. Two different setups are presented in Table 3, a large CLDNN-1 model, which acts as the basemodel and a smaller CLDNN-2.

Setup	CNN-Maps	LSTM	DNN
CLDNN-1	39 (time) + 256(frequency)	256	512
CLDNN-2	39 (time) + 128(frequency)	128	256

Table 3: CLDNN Setup. All of the models use a two convolutional layers (time + frequency), two layers of LSTM and a single layer of DNN.

4.5. Sequence length influence

The sequence length plays a crucial role in training RNN-based systems, thus the question arises if the CLDNN performance increase commensurates with a larger sequence length, as it is the case for standard LSTM. It is investigated which sequence length is most likely to be optimal for this task.

Sequence length	FAR	FRR	HTER
25	2.98	0.14	1.56
50	2.62	0.79	1.7
70	3.56	0.8	2.18

Table 4: Sequence length influence, note that FAR and FFR are taken from HTER Error. CLDNN-1 is used as model.

The results using multiple sequence lengths show an uncharacteristic behavior for RNN's. We assume that either the CNN front-end contribution to the final performance is more significant than the LSTM or that by increasing the sequence-length inadvertently decreases the number of samples available in the dataset, which leads to a possible underfit of the data.

4.6. Neural network comparison

For a better comparison to other neural network based methods, a two (LSTM), a two layer bidirectional LSTM (BLSTM) and an improved DNN-BLSTM fusion model, similar to that in [19] were also trained. LSTM and BLSTM models contain in each layer 512 neurons. The DNN-BLSTM model uses the concatenated output vectors of a 7 layer DNN (from the 3rd hidden layer) in addition to the output of three different BLSTM models. The BLSTM output vectors have a size of 512 each, while the DNN uses 1024 dimensional vector representations. Thus, a $1024 + (512 \times 3) = 2560$ dimensional vector representation is obtained. Compared to the proposed end-to-end model, these models make use of a backend LDA classifier, which creates a single score for each vector representation. LSTM, BLSTM and BLSTM-DNN models all use a sequence length of 50.The LSTM, BLSTM and DNN-BLSTM models uniformly use PLP-39 features, similar to these in Table 1 as their input. The results of this paper are compared with other neural networks attempts for antispoof (Table 5). Note that in the competitions paper the categories "LP-LP" were used as a superset of "LP-HQ-LP", thus being non independent, where in this work we assume each category is independent from each other.

Attack *Classifier*	**LSTM** LDA	**BLSTM** LDA	**BLSTM DNN** LDA	**CLDNN-1** Softmax	**CLDNN-2** Softmax	**Best-BTAS** GMM
All	2.99	2.43	1.21	1.56	**0.82**	*1.26*
SS-LP-LP	1.51	1.79	0.5	1.14	**0.38**	0.68
SS-LP-HQ-LP	1.42	1.61	0.95	1.05	**0.64**	0.68
VC-LP-LP	2.64	1.89	0.59	0.97	**0.49**	0.68
VC-LP-HQ-LP	1.64	1.9	0.57	0.55	**0.33**	0.81
RE-LP-LP	4.1	2.85	1.06	0.63	**0.52**	0.87
RE-LP-HQ-LP	11.54	8.54	6.69	2.88	**0.96**	1.81
RE-PH1-LP	5.73	2.04	0.69	0.57	**0.52**	0.68
RE-PH2-LP	3.29	1.54	**0.5**	0.57	1.08	0.68
RE-PH2-PH3	9.48	4.29	3.06	6.63	**1.33**	6.49
RE-LPPH2-PH3	27.35	22.1	26.44	38.07	**21.14**	23.06

Table 5: Comparison between other NN-approaches and the currently best result. Note that "RE-LP-LP","VC-LP-LP" and "SS-LP-LP" do not incorporate the HQ categories (in contrast to the original paper).

As it can be seen in Table 5, the BLSTM-DNN fusion model does outperform the baseline shown in Section 4.3, as well as other neural network based approaches. Additionally, it sets the mark of creating the currently best result on this corpus. Furthermore, the CLDNN-2 model performs well on unknown attacks (11.64% compared to 14.78%).

5. CONCLUSION

This paper successfully introduces an end-to-end framework using a raw waveform based CLDNN model for spoofing detection. Compared to the previous deep feature based methods, which builds the front-end and back-end separately, the new end-to-end raw waveform CLDNN makes the whole detection process more flexible, while at the same time being able to use the rich speech information from raw waveform by simultaneously optimizing feature extraction and classification accuracy. Surprisingly, performance increases by using these unprocessed raw waveform, indicating that raw signal might be a valid start point for this task, and the joint optimization on both front-end and back-end also makes this new architecture advanced.

In future research, we would like to focus on more complex front-end feature extraction using more suitable time and frequency filtering networks.

6. REFERENCES

[1] Serife Kucur Ergnay, Elie Khoury, Alexandros Lazaridis, and Sebastien Marcel, "On the vulnerability of speaker verification to realistic voice spoofing," *IEEE International Conference on Biometrics: Theory, Applications and Systems*, 2015.

[2] Nicholas Evans, Tomi Kinnunen, Junichi Yamagishi, Zhizheng Wu, Federico Alegre, and Phillip De Leon, "Speaker recognition anti-spoofing," *Handbook of Biometric Anti-Spoofing*, , no. Springer, pp. 1–25, 2014.

[3] Shitao Weng, Shushan Chen, Lei Yu, Xuewei Wu, Weicheng Cai, Zhi Liu, and Ming Li, "The sysu system for the interspeech 2015 automatic speaker verification spoofing and countermeasures challenge," *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[4] Sergey Novoselov, Alexandr Kozlov, Galina Lavrentyeva, Konstantin Simonchik, and Vadim Shchemelinin, "Stc anti-spoofing systems for the asvspoof 2015 challenge itmo university , st. petersburg , russia," *Icassp 2016*, pp. 5475–5479, 2016.

[5] Elie Khoury, Tomi Kinnunen, Aleksandr Sizov, Zhizheng Wu, and Sebastien Marcel, "Introducing i-vectors for joint anti-spoofing and speaker verification," *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, , no. Cm, pp. 61–65, 2014.

[6] Massimiliano Todisco, Hctor Delgado, and Nicholas Evans, "A new feature for automatic speaker verification anti-spoofing: Constant q cepstral coefficients," , no. July, 2016.

[7] Md Sahidullah, Tomi Kinnunen, and Cemal Hanili, "A comparison of features for synthetic speech detection," *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. 2015, no. July 2016, pp. 2087–2091, 2015.

[8] Nanxin Chen, Yanmin Qian, and Kai Yu, "Multi-task learning for text-dependent speaker verification," in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2015, vol. 2015, pp. 185–189.

[9] Nanxin Chen, Yanmin Qian, Heinrich Dinkel, Bo Chen, and Kai Yu, "Robust deep feature for spoofing detection - the sjtu system for asvspoof 2015 challenge," *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pp. 2097–2101, 2015.

[10] Yanmin Qian, Nanxin Chen, and Kai Yu, "Deep features for automatic spoong detection," *Speech Communication*, 2016.

[11] Zhi-zheng Wu, Eng Siong Chng, and Haizhou Li, "Detecting converted speech and natural speech for anti-spoofing attack in speaker recognition," *Proceedings of the 13th Annual Conference of the International Speech Communication Association, Interspeech 2012*, pp. 1700–1703, 2012.

[12] Yuan Liu, Yanmin Qian, Nanxin Chen, Tianfan Fu, Ya Zhang, and Kai Yu, "Deep feature for text-dependent speaker verification," *Speech Communication*, vol. 73, pp. 1–13, 2015.

[13] David R. Tobergte and Shirley Curtis, "Convolutional, long short-term memory, fully connected deep neural networks," *Journal of Chemical Information and Modeling*, vol. 53, no. 9, pp. 1689–1699, 2013.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," *Arxiv.Org*, vol. 7, no. 3, pp. 171–180, 2015.

[15] Yedid Hoshen, Ron J. Weiss, and Kevin W. Wilson, "Speech acoustic modeling from raw multichannel waveforms," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2015, vol. 2015-Augus, pp. 4624–4628.

[16] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout : A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research (JMLR)*, vol. 15, pp. 1929–1958, 2014.

[17] Matthew D. Zeiler, "Adadelta: An adaptive learning rate method," *arXiv*, p. 6, 2012.

[18] Tara N. Sainath, Ron J. Weiss, Andrew Senior, Kevin W. Wilson, and Oriol Vinyals, "Learning the speech front-end with raw waveform cldnns," in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2015, vol. 2015, pp. 1–5.

[19] P Korshunov, S Marcel, and H Muckenhirn, "Overview of btas 2016 speaker anti-spoofing competition," in *IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, Niagara Falls, NY, USA, 9 2016.

[20] Ivana Chingovska, Andre Rabello Dos Anjos, and Sebastien Marcel, "Biometrics evaluation under spoofing attacks," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 12, pp. 2264–2276, 2014.