RECURRENT NEURAL NETWORKS FOR AUTOMATIC REPLAY SPOOFING ATTACK DETECTION

Zhuxin Chen[1]*, Weibin Zhang*[1]*, Zhifeng Xie*[1]*, Xiangmin Xu*[1]*, Dongpeng Chen*[2]

[1]South China University of Technology, GuangZhou, China
[2]Shengyang Technology Company, Shenzhen, China

ABSTRACT

In order to enhance the security of automatic speaker verification (ASV) systems, automatic spoofing attack detection, which discriminates the fake audio recordings from genuine human speech, has gain much attention recently. Among various ways of spoofing attacks, replay attacks are one of the most effective and economical methods. In this paper, we explore using recurrent neural networks for automatic replay spoofing attack detection. More specifically, we focus on recurrent neural networks with more sophisticated recurrent units that involve a gating mechanism, such as a long short term memory (LSTM) unit and a recently proposed gated recurrent unit (GRU). Our experimental results on the ASVspoof 2017 showed that neural networks significantly outperform Gaussian mixture models (GMM). In addition, we achieved the best equal error rate of 9.81% on the ASVspoof 2017 and 1.077% on the BTAS 2016 by using GRU models, which outperform the best feed-forward neural networks by 19% and 46%, relatively and respectively.

Index Terms— Recurrent neural networks, Replay Detection, ASVspoof 2017, BTAS 2016

1. INTRODUCTION

Like other biometric systems such as fingerprint and face recognition, automatic speaker verification (ASV) systems, which determine if a speech recording is generated by a preregistered speaker, are of great use in real life. Unfortunately, ASV systems are still prone to different kinds of attacks. Due to the rapid development in speech technology, voice conversion (VC) [1] and speech synthesis (SS) [2] techniques make it possible to generate synthetic speech that is good enough to deceive an ASV system. In this paper, we focus on another simple yet effective attacking method, the replay spoofing attack. Replay attack can be carried out by replaying recordings of an enrolled speaker's voice to an ASV system in place of genuine speech. No expertise is required to perform replay attacks and common devices such as smart phones can be used as playback devices. Therefore, replay attack poses the greatest threat [3] to ASV systems. Experimental results from 2015 and 2017 ASV spoof challenges [3, 4] also show that the detection of SS and VC spoofing attacks is easier than the detection of replay attacks.

It is very important to develop effective countermeasures to enhance the security of ASV systems. The ASVspoof 2015 challenge [4], which focused on detection of artificial speech generated by VC and SS methods, has stimulated many countermeasures. Most of the efforts were spent on finding new features. Novel speech features such as dynamic coefficients of cepstral features computed by formant-specific block transformation [5] yield almost 0% *equal error rate* (EER) if the spoofing types are known (i.e. homogeneous spoofed speeches are used for training). Constant Q cepstral coefficients (CQCCs), which use the constant Q transform (CQT) instead of Fourier transform to process speech signal, were shown to outperform the previous best result on the ASVspoof 2015 database [6]. In terms of classifiers, Gaussian mixture models (GMMs) are commonly used [5, 7, 8, 9]. Traditional feed-forward neural networks [10] or simple multiple layer perceptron [11] is needed if the input features are high dimensional. Finally, model fusion is found to benefit model robustness [11, 12].

Spoofing attack detection can be viewed as a simple classification problem where the input speech is classified as either genuine human speech or spoofed speech. Therefore, simple classifiers such as GMMs [5, 7, 8, 9], feed-forward neural networks [10] and support vector machines [10] are commonly used. On the other hand, spoofing attack detection can also be viewed as a sequence classification task. Therefore, recurrent neural networks that have been successfully applied to various sequence prediction and sequence labeling tasks such as speech recognition and language modeling can also be used for identifying spoofing attack. In this paper, we are interested in evaluating two closely related variants, i.e., long short-term memory (LSTM [13]) networks and gated recurrent unit (GRU [14]) networks, in spoofing attack detection.

The rest of this paper is organized as follows. In Section 2, we review the related works on spoof detection. In Section 3, we describe the RNN architectures with LSTM unit and GRU unit, followed by the description of dataset, experimen-

E-mail:eeweibin@scut.edu.cn, chen.zhuxin@mail.scut.edu.cn

tal setup and results in Section 4. A conclusion is given in Section 5.

2. RELATED WORKS

Features are crucial to the success of spoof detection. They should provide salient and compact information for the classifiers. Researchers focus on magnitude based features and phase based features [11]. Except CQCCs and MFCCs introduced before, low-level features such as filter bank (Fbank) have been found to lower error rate in speech recognition [15]. In this paper, we will mainly explore CQCCs, MFCCs and Fbank.

As for the classifiers, one popular approach is to use the traditional GMM. In this approach, two different GMMs are trained using the genuine and the spoofed speech sample respectively. During evaluation, the log-likelihood ratio (LLR) score for a given evaluation utterance of T frames is calculated using the following equation:

$$LLR = \frac{1}{T}\left(\sum_{t=1}^{T} logP\left(\mathrm{o}_t|M_g\right) - logP\left(o_t|M_s\right)\right) \quad (1)$$

where $P\left(\mathrm{o}_t|M_g\right)$ denotes the likelihood of genuine speech and $P\left(\mathrm{o}_t|M_s\right)$ denotes the likelihood of spoofed speech. Another efficient approach is to use the feed-forward neural networks. In this approach, the features within a context window are usually spliced together as the inputs of deep neural networks [10]. During testing, it is similar to the case of GMM by using the posterior probabilities given by the network outputs.

3. RECURRENT NEURAL NETWORKS

Recurrent neural networks have achieved tremendous success for sequential modeling. The basic idea of Recurrent Neural Networks (RNNs) is to make use of sequential information. Unlike the feed-forward DNNs where the features within a context window are spliced together as input to make use of dynamic information, RNNs use their internal memory to process arbitrary sequences of inputs, allowing information to persist as they are stored in the memory cell. Therefore, RNNs can capture more information about the sequence.

RNNs have loops in the hidden states, allowing information to be passed from one step of the network to the next. It can be unrolled in its forward computation. However, conventional RNNs suffer from the notorious gradient vanishing problem during backpropagation [16, 17]. The long short-term memory (LSTM [13]) unit and the gated recurrent unit (GRU [14]) are proposed to deal with this problem.

3.1. Long Short-Term Memory

As Figure 1 shows, the LSTM architecture consists of a set of recurrently connected units, known as memory blocks. The memory blocks usually contain one self-connected memory cells to store the temporal state of the network. Three element-wise multiplicative units (namely the input, output and forget gates) are used to control the flow of information. In our implementation, we include peephole connections from its internal cells to the gates in the same cell to learn precise timing of the outputs [18].

At each time step t, the LSTM model can be described using the following equations:

$$i_t = \sigma(W_{ix}x_t + W_{im}m_{t-1} + W_{ic}c_{t-1} + b_i) \quad (2)$$

$$f_t = \sigma(W_{fx}x_t + W_{fm}m_{t-1} + W_{fc}c_{t-1} + b_f) \quad (3)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g(W_{cx}x_t + W_{cm}m_{t-1} + b_c) \quad (4)$$

$$o_t = \sigma(W_{ox}x_t + W_{om}m_{t-1} + W_{oc}c_t + b_o) \quad (5)$$

$$m_t = o_t \odot h(c_t) \quad (6)$$

where i_t, f_t, o_t and c_t denote respectively the activation vectors of input gate, output gate, forget gate, and memory cell at time step t. m_t denotes the cell block output. The W terms denote the weight matrices (e.g., W_{ix} is the matrix of weights from the input gate to the input). The b terms denote the bias vectors (e.g., b_i is the input gate bias vector). $\odot$ denotes the element-wise product of the vectors. g and h are the activation functions of the cell input and cell output. In our experiments, we use the $tanh$ function.

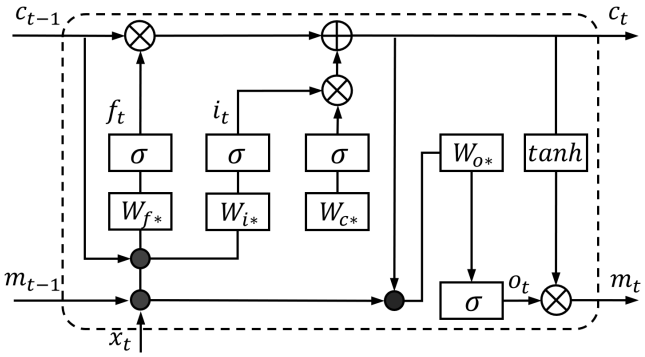

Fig. 1. *A single memory block of the LSTM.*

3.2. Gated Recurrent Unit

Gated Recurrent Units (GRUs), recently proposed by Cho $etal$. [14], are a simpler variant of the LSTM that shares many of the same properties. The GRU has gate units to control the flow of information inside the units, which is similar to LSTM units. Figure 3 shows the architecture of a GRU unit. Unlike the LSTM, a GRU unit only has two gates, the update gate and the reset gate. The GRU fully expose its memory content by eliminating the output gates.

The GRU computes the network activations and outputs at every time step t according to the following equations:

$$z_t = \sigma(W_{zx}x_t + W_{zh}h_{t-1} + b_z) \quad (7)$$

$$r_t = \sigma(W_{rx}x_t + W_{rh}h_{t-1} + b_r) \quad (8)$$

$$\widetilde{h_t} = \tanh(W_{hx}x_t + W_{hh}(r_t \odot h_{t-1}) + b_h) \quad (9)$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \widetilde{h_t} \quad (10)$$

Where x_t, h_t, z_t and r_t are respectively the input vector, output vector, update gate vector and reset gate vector. $\widetilde{h_t}$ is the candidate output.

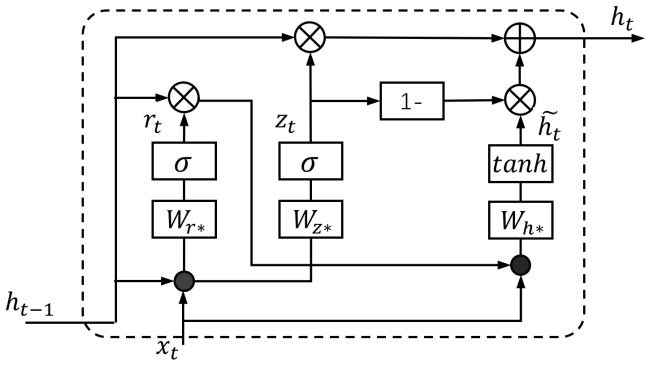

Fig. 2. *A single memory block of the GRU.*

3.3. RNN for Spoof Detection

As shown in Figure 3, the RNN we used has three recurrent layers, followed by a softmax layer that classifies each input frame into spoof or genuine. The RNN block in Figure 3 can be either a LSTM memory block or a GRU block. In our experiments, all the LSTM and GRU blocks have 256 cells.

During training, we cut every utterance into fixed size pieces of length 30 frames. The step size for the pieces is 22. Thus there are eight overlapping frames. The categorical cross-entropy is used as the objective function. In addition, to alleviate the overfitting problem, the dropout technique is applied to the output of RNN blocks with a dropout rate of 0.2.

Given an input frame o_t, the posterior probabilities corresponding to genuine and spoof labels for o_t can be obtained from the output of the network. Similar to the calculation of LLR given in Equation (1), the score for a given utterance of T frames can be calculated as follows:

$$score = \frac{1}{T}\left(\sum_{t=1}^{T} logP\left(genuine|o_t\right) - logP\left(spoof|o_t\right)\right) \quad (11)$$

The system then makes a decision based on the normalized $score$ by comparing with a pre-defined threshold θ.

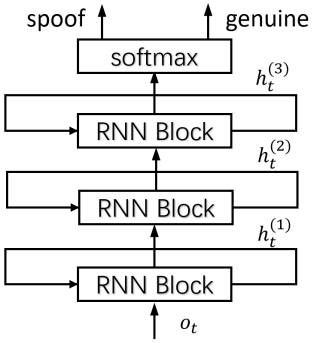

Fig. 3. *The architecture of the RNN for ASV spoof detection in our experiment.*

4. EXPERIMENT

4.1. Dataset

In this paper, we focus on replay spoof detection. ASVspoof 2017 and BTAS 2016 [19] were used. Note that the BTAS 2016 dataset contains SS and VC speech samples which were not included in our experiments. Both of the datasets are separately partitioned into three subsets: training, development and evaluation. Table 1 briefly gives the statics of them. To encourage research towards generalized spoofing countermeasure, only part of the replay conditions in evaluation are seen in the training and development datasets, especially in ASVspoof2017. In our experiments reported below, we used all the training data to train the models and all the development data to tune the model parameters.

Table 1. Number of utterances in both datasets. RE: replay, HQ: high quality speaker, PH1: Samsung Galaxy S4 phone, PH2: iPhone 3GS and PH3 is iPhone 6S.

Database	Types	Train	Dev	Eval
ASVspoof 2017	Genuine data	1508	760	1298
	All replays	1508	950	12922
BTAS 2016	Genuine data	4973	4995	5576
	All replays	2800	2800	4800
	RE-LP-LP	700	700	800
	RE-LP-HQ-LP	700	700	800
	RE-PH1-LP	700	700	800
	RE-PH2-LP	700	700	800
	RE-PH2-PH3	-	-	800
	RE-LPPH2-PH3	-	-	800

Table 2. The EERs(%) for GMM and DNN models on ASVspoof2017.

Model	GMM		DNN	
Feature	CQCC		CQCC	
Dataset	DEV	EVAL	DEV	EVAL
EER	10.83	28.06	5.44	20.36

4.2. Evaluation Metrics

The spoof detection system assigns a score for every audio file (e.g. the score given by Equation (1)). Higher scores means that the system are more confident that the trial is genuine speech while lower scores are assumed to favor the spoofed hypothesis (i.e. replayed speech). Let FAR(θ) and FRR(θ) be the false acceptance and false reject rates defined at the threshold θ, i.e.,

$$\text{FAR}(\theta) = \frac{\text{num. of replay trials with score} > \theta}{N_{spoofed}} \quad (12)$$

$$\text{FRR}(\theta) = \frac{\text{num. of replay trials with score} < \theta}{N_{genuine}} \quad (13)$$

where $N_{spoofed}$ and $N_{genuine}$ are total replay (spoofed) trials and total non-replay (genuine) trials respectively. The metric used in our experiments is the $equal\ error\ rate$ (EER) which

Table 3. The EERs(%) of DNN, LSTM and GRU on ASVspoof 2017.

Model	DNN				LSTM				GRU			
Feature	MFCC		Fbank		MFCC		Fbank		MFCC		Fbank	
Dataset	DEV	EVAL	DEV	EVAL	DEV	EVAL	DEV	EVAL	DEV	EVAL	DEV	EVAL
EER	7.59	12.87	8.09	12.13	10.06	14.42	6.88	10.98	10.39	14.18	6.32	**9.81**

Table 4. The EERs(%) of DNN, LSTM and GRU on BTAS 2016.

Model	DNN				LSTM				GRU			
Feature	MFCC		Fbank		MFCC		Fbank		MFCC		Fbank	
Dataset	DEV	EVAL	DEV	EVAL	DEV	EVAL	DEV	EVAL	DEV	EVAL	DEV	EVAL
ALL	1.153	2.058	0.779	2.007	0.273	2.149	0.052	1.107	0.241	1.912	0.039	**1.077**
RE-LP-LP	0.378	0.773	0.234	0.783	0.215	1.102	0.019	0.528	0.192	2.197	0.019	0.443
RE-LP-HQ-LP	2.905	2.534	2.213	2.308	0.425	1.893	0.122	1.182	0.39	2.038	0.114	0.752
RE-PH1-LP	0.266	1.312	0.254	0.662	0.168	0.698	0.056	0.141	0.235	0.897	0.031	0.191
RE-PH2-LP	0.128	1.002	0.065	0.908	0.063	0.568	0.018	0.209	0.055	1.734	0.019	0.267
RE-PH2-PH3	-	2.521	-	2.517	-	2.461	-	0.495	-	2.364	-	0.53
RE-LPPH2-PH3	-	2.622	-	2.994	-	3.717	-	2.32	-	3.184	-	2.592

corresponds to the threshold θ_{EER} at which the two detection error rates are (approximately) equal. Thus the lower the EER is, the better the system.

4.3. Experimental Setup and Results

Three kinds of features, namely, the CQCCs, MFCCs and Fbank, were used in our experiments. We followed the configuration in [7] to extract 30-dimension static CQCC features. As for MFCC and Fbank, we extract the features every 10ms with a 25ms Hamming window. Unlike the typical settings in speech recognition or speaker recognition, we increased the number of triangular mel-frequency filters to 120. In addition, we used 30-dimension cepstral coefficients. We have found that this can significantly improve the recognition accuracy.

We first evaluated the GMM and DNN approaches. Two GMM models, one for the genuine speech and the other one for the spoofed speech, were trained. Each GMM has 512 Gaussian Components. Static features, together with their delta and double delta, were used as the input for GMMs. As for the input of conventional feed-forward DNNs, we spliced the static features with a context window of 11 frames (i.e. 5 left frames and 5 right frames). There are three hidden layers and each has 512 units. The output layer is a softmax layer with dimension 2. Batch normalization [20] and dropout [21] were used to ease model initialization and prevent the model from over-fitting.

The results of GMM and DNN models with CQCCs as the features are shown in Table 2. As can be seen, DNN significantly outperform the GMM model on both the development and evaluation data sets. We then go on to evaluate DNN models with MFCC and Fbank features. The results are shown in the first part of Table 3. As can be seen, with the recommended settings (i.e. 120 filters and 30 cepstral coefficients), the MFCC and Fbank features significantly outperform the CQCC feature. Therefore, in the following experiments, DNNs with MFCCs and Fbank features will serve as the baseline.

The results of RNN models on ASVspoof 2017 are shown in Table 3. The Fbank feature achieves the best recognition accuracy for RNNs in replay spoofing detection. This coincides with the findings in other applications such as speech recognition [22]. Finally, the GRU model with Fbank feature achieves the best $equal\ error\ rate$ of 9.81%, which outperforms the best feed-forward neural network by 19% relatively.

Table 4 shows the results on BTAS 2016. As we have much more training data on BTAS2016, better results were achieved. Almost the same conclusion can be drawn. The best GRU model outperforms the best DNN model by 46% relatively.

5. CONCLUSION

In this paper, we studied the recurrent neural networks for replay spoof detection. Two widely used recurrent neural networks, namely the LSTM and GRU, were evaluated. We found that both of them significantly outperform the feed-forward neural networks. The GRU models achieve the best results. Finally, by increasing the number Mel-frequency filters and cepstral coefficients, the Fbank feature outperforms the CQCC feature in all our experiments.

6. ACKNOWLEDGEMENT

This work is supported in part by the National Natural Science Founding of China (61601187, U1636218) and the Science and Technology Program of Guangzhou (201704020043).

7. REFERENCES

[1] Tomoki Toda, Hiroshi Saruwatari, and Kiyohiro Shikano, "Voice conversion algorithm based on gaussian mixture model with dynamic frequency warping of straight spectrum," in *ICASSP*, 2001.

[2] Phillip L. De Leon, Michael Pucher, Junichi Yamagishi, Inma Hernaez, and Ibon Saratxaga, "Evaluation of speaker verification security and detection of hmm-based synthetic speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 8, pp. 2280–2290, 2012.

[3] Tomi Kinnunen, Md Sahidullah, Hctor Delgado, Massimiliano Todisco, Nicholas Evans, Junichi Yamagishi, and Aik Lee Kong, "The asvspoof 2017 challenge: Assessing the limits of replay spoofing attack detection," in *INTERSPEECH*, 2017.

[4] Zhizheng Wu, Tomi Kinnunen, Nicholas Evans, Junichi Yamagishi, Cemal Hanilc, and Md Sahidullah Aleksandr Sizov, "Asvspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 4, pp. 588–604, 2015.

[5] Dipjyoti Paul, Monisankha Pal, and Goutam Saha, "Novel speech features for improved detection of spoofing attacks," in *INDICON IEEE*, 2015, pp. 1–6.

[6] Zhizheng Wu, Junichi Yamagishi, Tomi Kinnunen, Cemal Hanili, Mohammed Sahidullah, Aleksandr Sizov, Nicholas Evans, and Massimiliano Todisco, "Asvspoof: The automatic speaker verification spoofing and countermeasures challenge," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 4, pp. 588–604, 2017.

[7] Massimiliano Todisco, Héctor Delgado, and Nicholas Evans, "A new feature for automatic speaker verification anti-spoofing: Constant q cepstral coefficients," in *Speaker Odyssey Workshop*, 2016, vol. 25, pp. 249–252.

[8] Jon Sanchez, Ibon Saratxaga, Inma Hernaez, Eva Navas, and Daniel Erro, "The aholab rps ssd spoofing challenge 2015 submission," in *INTERSPEECH*, 2015.

[9] Kaavya Sriskandaraja, Vidhyasaharan Sethu, Phu Ngoc Le, and Eliathamby Ambikairajah, "Investigation of sub-band discriminative information between spoofed and genuine speech.," in *INTERSPEECH*, 2016.

[10] Jess Villalba, Antonio Miguel, Alfonso Ortega, and Eduardo Lleida, "Spoofing detection with dnn and one-class svm for the asvspoof 2015 challenge," in *INTERSPEECH*, 2015.

[11] Xiong Xiao, Xiaohai Tian, Steven Du, Haihua Xu, Eng Siong Chng, and Haizhou Li, "Spoofing speech detection using high dimensional magnitude and phase features: the ntu system for asvspoof 2015 challenge," in *INTERSPEECH*, 2015.

[12] Zhuxin Chen, Zhifeng Xie, Weibin Zhang, and Xiangmin Xu, "Resnet and model fusion for automatic spoofing detection," in *INTERSPEECH*, 2017.

[13] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[14] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[15] Abdel-rahman Mohamed, George E Dahl, and Geoffrey Hinton, "Acoustic modeling using deep belief networks," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 14–22, 2012.

[16] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[17] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio, "On the difficulty of training recurrent neural networks," in *International Conference on Machine Learning*, 2013, pp. 1310–1318.

[18] Felix A Gers, Nicol N Schraudolph, and Jürgen Schmidhuber, "Learning precise timing with lstm recurrent networks," *Journal of machine learning research*, vol. 3, no. Aug, pp. 115–143, 2002.

[19] Serife Kucur Ergnay, Elie Khoury, Alexandros Lazaridis, and Sbastien Marcel, "On the vulnerability of speaker verification to realistic voice spoofing," in *IEEE International Conference on Biometrics Theory, Applications and Systems*, 2015, pp. 1–6.

[20] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

[21] 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*, vol. 15, no. 1, pp. 1929–1958, 2014.

[22] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton, "Speech recognition with deep recurrent neural networks," in *ICASSP*, 2013.