Device Feature Extraction Based on Parallel Neural Network Training for Replay Spoofing Detection

Chang Huai You, *Member, IEEE,* Jichen Yang, *Member, IEEE*

***Abstract*—A device feature, which contains information from both the recording channel and playback channel, is of paramount importance in replay spoofing detection. Presently, no technical reports, pertaining to the usage of device information for spoofing detection to achieve speaker verification, have been published. In this paper, we propose to build a replay device feature (RDF) extractor on the basis of the genuine-replay-pair parallel neural network training database. The target of the neural network for such an RDF extractor is constrained in a spectrum domain such as discrete Fourier transform or constant-Q transform. Different types of neural networks are investigated for the parallel training framework. Finally the RDF extractor is formed by applying discrete cosine transform to the output vector of neural network. The bidirectional long short term memory gives the best performance among the investigated types of neural networks. The experimental results on ASVspoof 2017 corpus version 2.0 indicate that the equal error rate (EER) of a replay detection system with the proposed RDF has a value of 15.08%. Furthermore, by combining the RDF with constant-Q cepstral coefficients plus the log energy in score level, the EER of the detection system can be further reduced to 8.99%. In addition, the experimental results also reveal that the RDF exhibits great complementarity with many handcrafted features.**

Index Terms: replay speech detection, recording device, playback device, speaker verification

I. Introduction

A series of techniques starting from Gaussian mixture model (GMM)-universal background model (UBM) [1], hidden Markov Model (HMM)-UBM [2], support vector machine (SVM) [3] [4], joint factor analysis (JFA) [5] [6] [7], i-vector [8] [9] [10] [11] to the deep neural network based x-vector [12] [13] have been evidencing the progress of speaker recognition. As a result speaker verification technology has been widely applied in multitudinous commercial applications such as credit card, telephone banking, e-commerce, secure building access and suspect identification. However, these techniques become vulnerable while facing intentional attacks. False acceptance rates of state-of-the-art automatic speaker verification (ASV) systems significantly increased with the spoofing attacks using voice conversion technology [14] [15] [16], speech synthesis [17] technology, replay tool [18] [19] [20] [21] and impersonation skill [22]. Being provoked by the severity of these fraudulent attacks, the Automatic Speaker Verification Spoofing and Countermeasures Challenge (ASVspoof) was launched in 2015 with the objective of enhancing the security of ASV systems against spoofing attacks. Among the above-mentioned attacks, replay spoofing poses a significant threat to the reliability of speaker verification system because replay spoofing is the most difficult to be detected among the four kinds of attacks and it can be easily applied with very low technical cost [23] [25]. Following the ASVspoof 2015 challenge [26] which is mainly focusing on speech synthesis and voice conversion attacks, the ASVspoof 2017 challenge targets replay attack.

Replay attack happens easily as they do not need any profound technology or specialized tools. A serious instance is to use high-quality mobile-phone to arouse the replay attack to the speaker verification system. Detecting replay attack using acoustic signal processing is considered hard, due to the unpredictable variation in the quality of a replay attack. The naturally occurring factors such as reverberation may be confusable in some cases with the artifacts introduced by replay. Researchers also used machine learning for detecting replay attacks, and found it to perform poorly, mainly due to the overfitting caused by the variability of speech signals [27]. Such models do not generalize well to unseen acoustic environments that may be encountered in practice. Audio recordings using high-quality microphones in ideal acoustic environments can be indistinguishable from genuine speech signals.

In the case of replay attack, prerecorded utterance of the target speaker is played to a speaker verification system to gain unauthorized access. Obviously the replayed speech signal itself contains the characteristics of the intermediate replay devices, including the recording device and playback device. The earliest research pertaining to the vulnerabilities of a speaker verification system to replay attacks was performed in [28]. In this research, replay attacks were simulated by concatenating isolated digits from the recordings of a genuine user and played back to the speaker verification system. An aberrant increase in both equal error rate (EER) and false acceptance rate (FAR) was observed in the presence of the replay attacks. Similar trends reported in [29] indicate that the EER of a joint factor analysis (JFA) based speaker verification system was 0.71% when only non-spoofing impostor trials were tested but when this EER operating point was selected as the decision threshold, 68% of the replayed trials were wrongly accepted by the system.

A replay speech detector usually consists of a front-

This work was supported in part by the Programmatic Grant A1687b0033 through the Singapore Governments Research, Innovation and Enterprise 2020 Plan (Advanced Manufacturing and Engineering domain) and in part by the National Research Foundation Singapore through the AI Singapore Programme under Award AISG-100E-2018-006 (Chang Huai You and Jichen Yang are joint first authors). *(Corresponding author: Jichen Yang)*

Chang Huai You is with Institute for Infocomm Research, A*STAR, Singapore. (e-mail:echyou@i2r.a-star.edu.sg).

Jichen Yang is with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. (e-mail:eleyji@nus.edu.sg).

end feature extraction and back-end classification. Regarding the front-end feature extraction, many features for replay speech detection have been studied since the ASVspoof 2017 challenge [18] [19]. These features are categorized into two groups: handcrafted features and deep features. Typical handcrafted features used for spoofing detection include the constant-Q cepstral coefficient (CQCC) [30], Mel-frequency cepstral coefficient (MFCC) [30], linear frequency cepstral coefficient (LFCC) [31], rectangular filter cepstral coefficient (RFCC) [32] and constant-Q statistics-plus-principle information coefficient (CQSPIC) [33]. While the handcrafted features are obtained via certain transformations which convert signals from the time domain into variants of the transform domain, the deep features are usually obtained through deep learning of various neural networks, which include the deep neural network (DNN) [23], recurrent neural network (RNN) [23] and convolutional neural network (CNN) [34].

The deep feature is generally extracted from a hidden layer of deep model. There have been several works extracting deep feature for replay spoofing detection. In [35], both DNN-based frame-level and RNN-based sequence-level are developed to extract their respective deep features. For DNN-based deep feature, an utterance-level identity representation is obtained by averaging the deep features across the entire utterance, while for RNN-based deep feature the output of hidden layer at the last time step is used to generate the final identity representation for the entire utterance. Moreover, both DNN-based and RNN-based spoofing identity representation can be combined to achieve better performance of spoofing detection. In [34], log power magnitude from CQT and fast Fourier transform (FFT) are used as the input of a light CNN, then the deep feature is generated from the CNN. In [36], the deep feature is generated via a CNN-DNN neural network architecture. The neural network is trained with the input from tandem feature of high-frequency cepstral coefficient (HFCC) and CQCC, the target setting for neural network training is built to classify the genuine speech and replay speech, or different channel conditions. After the model is trained, the last fully connected layer is used to generate the deep feature. In [37], CNN plus RNN are trained for deep feature extraction, where the RNN is fed with the output of the CNN to learn long-term dependencies, so that all deep feature vectors from CNN are processed by the RNN to obtain the spoofing identity representation for the entire utterance.

Different from the above-mentioned deep features which are based on non-parallel training ways, in this paper, we will propose a parallel training method to extract feature which is targeted to device information extraction. Although device information is important for replay spoofing detection, there have never been any reports on how to extract device information in the speech signal processing field. In this paper, we focus on device information study and propose a deep feature extraction method to depict the trait of replay device [38]. We found that the replay device feature (RDF) is highly effective on determining the device-involved (replay) speech and non-device-involved (genuine) speech. We adopt a neural network [39] to model the replay device, and the neural network is parallel trained by using the genuine-replay-pair database. We investigated different neural network architectures in this study such as DNN, long short term memory (LSTM) and bidirectional LSTM (BLSTM). We found the BLSTM exhibits very good performance for the device feature representation. Finally the device feature is generated via discrete cosine transform (DCT) which is applied to the output layer of the trained neural network.

Regardless of the feature adopted, it is always followed by the final spoofing detection through the back-end classifier. In back-end classification part, most models for the spoofing detection are based on GMM [40] [41] [42], linear discriminative analysis (LDA) [35], support vector machine (SVM) [34] [43] [44], residual network (ResNet) [27], DNN [23] [24] and BLSTM [45] [46]. Neural network classifiers are reported to give better performance than GMM [47].

The remainder of the paper is organized as follows. Section II briefs the related methods through which we study the device feature. In Section III, we introduce the channel system of replay device and propose a device feature extraction framework. In Section IV, the experimental results and corresponding analysis on ASVspoof 2017 corpus version 2.0 are described. In Section V, the conclusions of the work are presented.

II. Related Methods

A. Constant-Q Transform versus Fourier Transform

Discrete Fourier transform (DFT) is the traditional approach widely used for the analysis of speech signals. However, it is not necessarily ideal for spoofing detection because it imposes the regular spaced frequency bins which leads to unbalance distribution of the frequency bin (bank) resolution.

Constant Q transform (CQT) is closely pertinent to the complex Morlet wavelet transform [48] [49]. Like DFT, CQT transforms the time-domain signal into the spectral domain. In DFT, frequency bin is distributed uniformly along with the frequency axis, so that every bin has the same bandwidth; it implies that the ratio of center frequency to bandwidth is increased while the frequency bin increases. In contrast to DFT, CQT is of constant ratio of center frequency to bandwidth. In particular, CQT has geometrically spaced center frequencies $f_k = 2^{\frac{k-1}{B}} f_1$, where $k = 1, 2, ..., K$ denotes the frequency bin, f_1 is the centre frequency of the lowest-frequency bin, and B is the number of bins per octave-band. It is the geometrically spaced distribution of the center frequencies that ensures a constant Q factor across the entire spectrum. The ratio of center frequency to bandwidth, Q, is formulated as follows:

$$Q = \frac{f_k}{\Delta f_k} = \frac{f_k}{f_{k+1} - f_k} \tag{1}$$

Thus we have Q value of CQT [1]

$$Q = \frac{f_1 2^{\frac{k-1}{B}}}{f_1 2^{\frac{k-1}{B}} (2^{\frac{1}{B}} - 1)} = \frac{1}{2^{\frac{1}{B}} - 1} \tag{2}$$

[1]The Q value of DFT for the k-th frequency bin is k.

(2) indicates that Q is a constant independent of frequency bin k. Obviously, the window length for the kth frequency bin is $N_k = \frac{f_s}{\Delta f_k}$ where f_s is the sampling frequency.

A standard DFT uses a constant bin size throughout all frequencies. This typically leads to a reasonably consistent, fully continuous transform. However, the constant bin size for all frequencies leads to some problems when frequency is mapped on a logarithmic scale. Specifically, the peaks of speech signal on the low frequency region are incredibly wide and DFT may result in lack of detail in the lower end. This is an issue in emulating human perception because humans perceive frequency on a logarithmic scale. In contrast to DFT, CQT seeks to solve this problem above by increasing the resolution for lower frequencies and alleviating some of the computational strains by reducing the bin size used for high frequencies. It allows CQT to yield better results where low frequencies and logarithmic frequency mapping are concerned, but it is slightly less detailed in the far upper frequencies.

B. Neural Network for Deep Feature

Neural network is net W hooking together many simple neurons so that the output of a neuron can be an input factor to another, where W denotes the whole set of trainable parameters of the neural network.

1) DNN: Generally, a neural network is formed by neurons which are distributed on every layer; the value of a neuron is given by applying an activation function to the input from the neurons of its preceding layer. Suppose we have a set of training samples $(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), ..., (\mathbf{x}^{(r)}, \mathbf{y}^{(r)}), ..., (\mathbf{x}^{(m)}, \mathbf{y}^{(m)})$ where $(\mathbf{x}^{(r)}, \mathbf{y}^{(r)})$ forms a pair with the r-th input vector and its corresponding labelled target. We can train the neural network using batch gradient descent. Having the loss function with respect to that single sample $J(W, \mathbf{x}^{(r)}, \mathbf{y}^{(r)})$, the batch cost function may be $\mathbf{J}(W) = \frac{1}{m}\sum[J(W, \mathbf{x}^{(r)}, \mathbf{y}^{(r)})] + \Phi(W)$, here $\Phi(W)$ indicates the weight decay term. Training the neural network may be just to repeatedly take gradient descent by using partial derivatives of the cost function with respect to the W through the back propagation way, so that the $W_{jk}^{(l)}$ in l-th layer can be updated by $W_{jk}^{(l)} = W_{jk}^{(l)} - \alpha\frac{\partial \mathbf{J}(W)}{\partial W_{jk}^{(l)}}$ for a particular weight connecting node j of current layer to node k of the next layer. Deep neural network is the forward neural network with more than two hidden layers. Given a trained W, output of DNN is produced from an input that goes through a number of layers.

2) LSTM: In general neural network, two successive inputs are assumed to be independent of each other. This assumption, however, is not true in real-life scenarios. RNN can use their internal state (memory) to memorize the sequence information of successive inputs. LSTM [50] is a variant architecture of RNN and it has shown to be successful in many speech processing applications.

The standard LSTM cell architecture contains one self-connected cell and three controlling gates. The equations of an LSTM cell are given as follows

$$\begin{aligned} i_t &= \sigma(W_{ij}x_t + b_{ii} + W_{hi}h_{(t-1)} + b_{hi}) \\ f_t &= \sigma(W_{if}x_t + b_{if} + W_{hf}h_{(t-1)} + b_{hf}) \\ g_t &= \tanh(W_{ig}x_t + b_{gf} + W_{hg}h_{(t-1)} + b_{hg}) \\ c_t &= f_t * c(t-1) + i_t * g_t \\ o_t &= \sigma(W_{io}x_t + b_{io} + W_{ho}h_{(t-1)} + b_{ho}) \\ h_t &= o_t * \tanh(c_t) \end{aligned} \tag{3}$$

where h_t is the hidden state at time t, c_t is the cell state at time t; x_t is the input at time t, $h(t-1)$ is the hidden state of the layer at time $t-1$ (or the initial hidden state at time t); and i_t, f_t, g_t, o_t are the input , forget, cell and output gates respectively; and σ is the sigmoid function, $*$ denotes Hadamard product.

In multilayer LSTM, the input $x_t^{(l)}$ of the l-th layer ($l >= 2$) is the hidden state $h_t^{(l-1)}$ of the previous layer by drop out $\sigma_t^{(l-1)}$ where each $\sigma_t^{(l-1)}$ is a Bernoulli random variable which is $\mathbf{0}$ with probability dropout.

An LSTM neural network is trained on a set of training sequences using the gradient descent method, it adopts back-propagation through time (BPTT) to compute the gradients needed during the optimization process, in order to change each weight of the LSTM network in proportion to the derivative of the error (at the output layer of the LSTM network) with respect to corresponding weight. When error values are back-propagated from the output layer, the error remains in the LSTM cell. This error carousel continuously feeds error back to each of the LSTM unit's gates, until they learn to cut off the value.

3) BLSTM: BLSTM is a typical example of bidirectional recurrent neural networks that connect two hidden layers of opposite directions to the same output. Unidirectional LSTM only makes use of past context. In contrast, BLSTM utilizes both the past and future context by processing the input in both forward and backward directions with two separated hidden layers, which are then concatenated as the final output. In BLSTM, the two directional neurons do not have any interactions. The process of the BLSTM layer can be depicted in Fig. 1.

BLSTM acoustic models often achieve better performance in speech recognition than LSTM. It comes with a cost where a significant latency needs to be introduced since decoding needs to wait until a whole sequence is completed.

III. Proposed Replay Device Feature Extractor

In the traditional training of deep feature, the generative neural network for deep feature of spoofing detection is non-parallel trained, the non-parallel training can be used for unsupervised training or semi-supervised training. For the unsupervised training, the target is the input feature itself; the cost function may be related to the squared error between the reconstructed feature (vector of output layer of the neural network) and the original input feature (e.g. CQCC, MFCC, LPCC etc). For semi-supervised training, the target can be spoofing identity (ID) or the joint trait with the speaker ID and spoofing ID; for this situation, the cost function can be

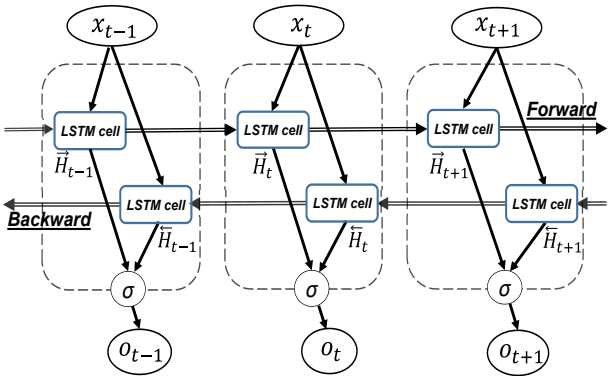

Fig. 1. A BLSTM cell process where $\vec{H}_t$ and $\bar{H}_t$ denote the LSTM cell parameters at time t instance for forward and backward layers respectively. Both forward and backward networks connect to a common activation layer to produce outputs.

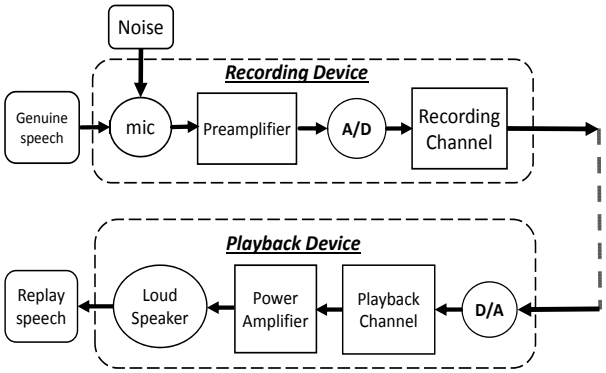

Fig. 2. Diagram of the replay speech generation process.

the squared error between the target and the vector of output layer of the deep feature neural network. After training the neural network, the final deep feature can be extracted from a certain hidden layer of the neural network by inputting the original feature.

Different from the traditional deep feature training method which is targeted to the type of attack, in this paper, we aim to develop a parallel training of neural network to represent the characteristics of replay device. In this deep feature training, the output of neural network is targeted to be the log-spectral-magnitude difference between the input speech and its corresponding genuine speech. We start our proposed RDF extraction from the analysis of the device system, then we introduce the training of the neural network, finally the RDF extractor is developed.

A. Analysis of Replay Device System

This section introduces replay speech generation process and the relationship between genuine speech and replay speech.

The replay device is composed of a recording device and playback device. The recording device is used to record genuine speech and the playback device is to play the recorded speech. The recording device contains three modules: preamplifier, analog-to-digital converter (A/D), and recording channel. On the other side, the playback device contains another three modules: digital-to-analog converter (D/A), power amplifier and loudspeaker. Fig.2 shows the process of replay speech generation.

It may be reasonable to assume A/D converter in recording device and D/A converter in playback device are ideal, since the quantization error can be negligible. The preamplifier in recording device and amplifier in playback device are used to amplify signal linearly.

Let $s_g(t)$ be a genuine speech, k_r the recording preamplifier coefficient, and $h_f(t)$ the impulse response of recording channel, the recording output $s_r(t)$ is given by:

$$s_r(t) = k_1(s_g(t) + n(t)) * h_f(t) \tag{4}$$

where $n(t)$ denotes the environmental noise in the recording environment.

Let k_2 denote the power amplification gain in playback device, $h_p(t)$ the impulse response of playback channel, and $h_l(t)$ the impulse response of loudspeaker. Given input $s_r(t)$, we have the output of the playback speech $s_\rho(t)$ as follows

$$s_\rho(t) = [k_2 s_r(t) * h_p(t)] * h_l(t) \tag{5}$$

Consequently, we have the replay speech as follows

$$\begin{aligned} s_\rho(t) &= k_2[k_1(s_g(t) + n(t)) * h_f(t)] * h_p(t) * h_l(t) \\ &= k_1 k_2 s_g(t) * h_f(t) * h_p(t) * h_l(t) \\ &+ \ k_1 k_2 n(t) * h_f(t) * h_p(t) * h_l(t) \end{aligned} \tag{6}$$

Applying a spectral domain transform such as DFT or CQT to Eq.(6), we have

$$\begin{aligned} S_\rho(jw) &= k_1 k_2 S_g(jw) H_f(jw) H_p(jw) H_l(jw) \\ &+ \ k_1 k_2 N(jw) H_f(jw) H_p(jw) H_l(jw) \end{aligned} \tag{7}$$

where $S_\rho(jw)$, $S_g(jw)$, $H_f(jw)$, $H_p(jw)$, $H_l(jw)$, and $N(jw)$ are the spectral transforms of $s_\rho(t)$, $s_g(t)$, $h_f(t)$, $h_p(t)$, $h_l(t)$, and $n(t)$ respectively.

Ignoring the input noise to the microphone, which is the environmental interference, we consider only the replay device affecting to the replay speech. The replay speech can be simplified as below

$$\tilde{S}_\rho(jw) = S_g(jw) H_\varpi(jw) \tag{8}$$

where $H_\varpi(jw) = k_1 k_2 H_f(jw) H_p(jw) H_l(jw)$. It is clear that $H_\varpi(jw)$ represents the generic replay device system function in any variant of spectrum domain, e.g. DFT, CQT or wavelet transform domain.

From Eq. (8), the log-spectral-magnitude of $H_\varpi(jw)$ can be obtained by:

$$\log|H_\varpi(jw)| = \log|\tilde{S}_\rho(jw)| - \log|S_g(jw)| \tag{9}$$

Eq. (9) describes the relationship among the input genuine speech, replay device system and the replay spoofing speech.

Usually device information hidden behind the utterance is relatively weak as compared to those non-device signals such as the content of speech, information of speaker, and even the noise that contaminated it. In some sense, the replay device information is actually submerged under these non-device signals. Without device information hidden inside a speech

utterance, there is nothing used to detect the replay spoofing. This equation provides a way to reveal the device information from the spectral aspect; it brings the motivation of the device feature extraction – as long as we have a parallel training speech dataset, we can adopt neural network architecture to extract the device information.

B. Neural Network for Deep Feature Generation

Actually it can be assumed that there is a virtual replay device system $\breve{H}$. For the case of replay speech, the virtual device system is the real device defined by Eq. (9); for the case of genuine speech, it is a full-pass filter system with its gain to be 1, i.e. $\breve{H}_{\varpi}(jw) = 1$.

Now let's define the virtual device system for any types of speech signal (either replay or genuine speech) as follows:

$$\log|\breve{H}_{\varpi}(jw)| = \begin{cases} \log|\tilde{S}_{\rho}(jw)| - \log|S_g(jw)|, \text{ if replay} \\ 0, \text{ if genuine} \end{cases} \quad (10)$$

Obviously, the higher is the quality of replay device, the closer to zero is $\log|\breve{H}_{\varpi}(jw)|$ and the flatter is it with respect to w. Inspired from Eq. (10), a neural network architecture, which can extract device information, is proposed by using the virtual device function $\log|\breve{H}_{\varpi}(jw)|$: Regardless what type of input speech (replay or genuine), there is a corresponding virtual device function $\log|\breve{H}_{\varpi}(jw)|$. Therefore we propose to design the neural network architecture as follows. The neural network parameters are trained using the genuine-replay-pair training database, so that a parallel training structure is applied by setting up the pair of input speech and its corresponding virtual device. (a) If the input is the feature of replay speech, the corresponding virtual device is actually the real device system, which can be computed using the difference between log-spectral-magnitudes of replay speech and its corresponding genuine speech. Thus the neural network output is targeted to the real device system that is equivalent to $\log|S_{\rho}(jw)| - \log|S_g(jw)|$. (b) If the input is the feature of genuine speech, the corresponding virtual device is a full-pass filter system with its gain set as 1, the output of the neural network is targeted to zero to indicate no real device applied. As a result, the neural network system becomes a virtual replay device function generator.

Obviously it is a regression problem of the neural network. The loss function, ψ, is the square error between the output $\mathbf{y}$ of the neural network and the virtual device function $\log|\breve{H}_{\varpi}(jw)|$ corresponding to the input speech.

$$\begin{aligned} \psi &= \left|\left|\mathbf{y} - \breve{\mathbf{H}}_{log}\right|\right|^2 \\ \mathbf{y} &= [y_0, y_1, ..., y_{b-1}] \\ \breve{\mathbf{H}}_{log} &= [\log|\breve{H}_{\varpi}(jw_0)|, \log|\breve{H}_{\varpi}(jw_1)|, ..., \log|\breve{H}_{\varpi}(jw_{b-1})|] \end{aligned} \quad (11)$$

where b denotes the number of frequency bins. As per Eq. (10), the output of neural network are constrained in the log magnitude spectrum domain such as DFT, CQT and wavelet transform.

Fig. 3 shows how to train the neural network. The input layer is the feature of speech utterance, the feature can be either MFCC or log-spectral-magnitude or its extension; and the output layer is constrained to target to the corresponding virtual device according to Eq. (10). The hidden layers define the type of neural network architecture. The mean-square-error (MSE) cost function given by $\frac{1}{n}\sum_{n=1}^{N}\psi_n$ is used as objective function for the neural network weight parameter update, where N denotes the number of frames. In this paper, we investigate different neural networks for the device feature extraction. They are DNN, LSTM, and BLSTM.

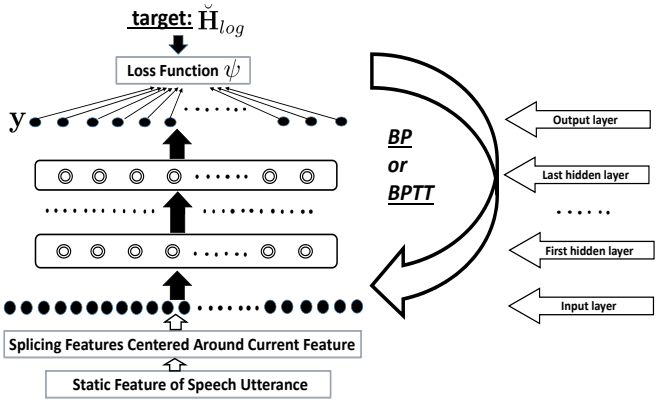

Fig. 3. Schematic diagram of neural network training for device feature extraction. BP: Backpropagation for DNN; BPTT: Backpropagation through Time for LSTM or BLSTM.

C. RDF Extraction

Given a speech utterance, depending on the configuration of the trained neural network, the relevant handcrafted feature such as CQCC, log-spectral-magnitude of CQT, MFCC or LPCC is extracted. The feature is used as input vector to neural network and go through the hidden layers of neural network to obtain the output vector. Finally DCT is applied on the output vector of the neural network to generate the RDF.

In this study, three neural networks (DNN, LSTM and BLSTM) were investigated. Two hidden layers are used for both DNN and LSTM respectively. The number of nodes in the hidden layers are the same for both DNN and LSTM for a fair comparison. They are 3072 and 1536 for the first hidden layer and second hidden layer respectively. And the BLSTM uses two LSTMs with same parametric structures, one for forward while another for backward. In order to minimize the computational complexity for feature extraction, the feature to the input layer of neural network is chosen to be the log-spectral-magnitude, which is consistent with the output layer of the neural network. Here CQT is mainly studied and adopted for the output target of the neural network. Considering CQT, the number of neurons in the input layer is 9493 (which is composed of eleven spliced frames), and the one in output layer is 863. In our experiments, BLSTM performs better than both DNN and LSTM.

D. An Example of Spoofing Detection System with RDF Feature Extraction

For the detection of spoofing attacks, GMM is commonly used for the classification while DNN is effective for high dimension input features [23].

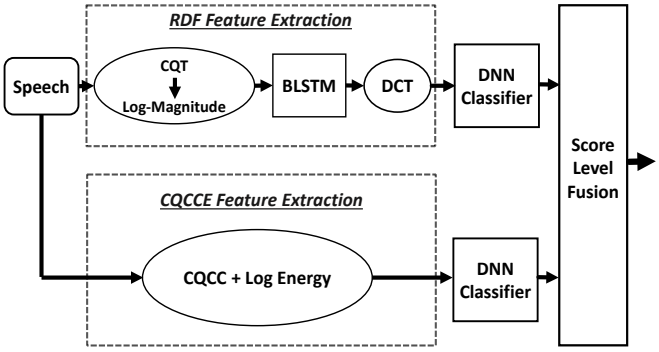

Fig. 4. An example of replay spoofing detection system with RDF feature extraction.

In this paper, we show an example of the RDF application by developing a hybrid spoofing detection system. In the hybrid system, RDF detection system is combined with CQCC plus the log energy (CQCCE) detection system to give a significant improvement on replay detection. In the RDF detection system, CQT is applied first to convert the signal from the time domain into CQT spectral domain, it is followed by magnitude and logarithm computation. The two systems go through fusion at the score level. Fig. 4 gives a brief flow chat to describe the example of the spoofing detection system with RDF feature extraction.

In this system, BLSTM is trained for RDF deep feature extraction, and DNN is trained as a classifier to detect the replay spoofing speech. The input layer of the DNN is connected to a big dimension from eleven adjacent frames of the RDF feature, there are two hidden layers with 512 neurons per layer. The output layer is composed of two neurons to indicate spoofing or genuine speech.

We apply the similar DNN network configuration to the CQCCE feature to implement another spoofing detection system, and finally the two systems are merged together at the score level to give the final score of the proposed hybrid spoofing detection system.

IV. Performance Evaluation

We conducted the evaluation on the ASVspoof 2017 Challenge platform that focuses on replay attacks [51]. The ASVspoof organizers used RedDots corpus to serve as genuine speech, and the speech from the subset of the RedDots corpus was replayed through a variety of replay configurations involving different recording devices, playback devices and different environments. In this paper, we adopt the improved version, i.e. ASVspoof 2017 corpus version 2.0 (ASVspoof2017-V2) [51].

TABLE I
Some details of ASVspoof2017-V2.

Subset	Num			
	#Speakers	#Utterances	#Genuine	#Spoofed
Training	10	3,014	1,507	1,507
Development	8	1,710	760	950
Evaluation	24	13,306	1,298	12,008

ASVspoof2017-V2 is constituted by three subsets: training data, development and evaluation data, Table I gives some details of ASVspoof2017-V2. The sampling rate of the speech signal is 16 kHz. The training database provides parallel training condition to train the replay device feature extractor. In particular, for the case of spoofing speech used as input to the neural network, there are 1507 genuine-replay pairs of the speech utterances from the training dataset to train neural network parameters; for the case of genuine speech used as input to the neural network, there are another 1507 genuine-genuine pairs of speech utterances from the training dataset to train the neural network parameters.

To examine the performance of our proposed device feature, we used a series of four-layer DNNs for classification. The performance of the entire system is measured in terms of EER. The input layer of the DNNs are constituted by splicing eleven frames centralized at the current frame. The DNNs have two hidden layers with 512 nodes in each layer and its output layer has two nodes.

A. Deployment of RDF Spoofing Detection on Development Dataset of ASVspoof2017-V2

Considering the factor of computational complexity, we keep the transform (either DFT or CQT) consistent for input and output of the neural network. In this paper, the log-spectral-magnitude of DFT or CQT is selected to be the input feature to the neural network for RDF feature extraction. Upon the investigation of DFT and CQT in the experiments, it was observed that CQT is much more advanced than DFT. In this section, we shall focus on CQT only. In our experiment, the static dimension of the device feature for CQT is set to 863 to give the best performance in terms of EER for 16 kHz sampling rate. The deployment of RDF-based spoofing detection is composed of two parts: neural network configuration for RDF feature extraction and dynamic feature input to detection classifier. The number of neurons in each layer of the neural network is the critical factor for the performance of the RDF extraction. In the other side, dynamic feature selection is an important factor for spoofing detection. Our previous works [33] [42] [52] have shown static feature for CQT degrades the performance in replay speech detection, therefore only dynamic features including delta (D), acceleration (A) and delta-acceleration concatenation (DA) are selected for investigation in this paper. The CQT parameters are configured as follows: number of frequency bins in each octave is set to 96, number of octaves is set to 9, sampling period is 16, and gamma is chosen as 3.3026. This settings are consistent with those in [53].

We used the same 3,014 genuin/replay utterances from the training set for the DNN model on the development set for the classification in replay spoofing detection. We show the experimental results of CQT:RDF on ASVspoof2017-V2 development dataset in Table II. In the table, different BLSTM neural network architectures for the device feature extractor are used as the front end of the detection system, and different dynamic features (i.e. D, A, and DA) as input to DNN classification are used as back-end of spoofing detection,

where ILN, HL1N, HL2N and OLN denote the numbers of neurons in input layer, first hidden layer, second hidden layer and output layer respectively.

With respect to RDF neural network configuration, the CQT:RDF-A with network structure 9493:3072:1536:863 performs best on ASVspoof2017-V2 development set among the four network structures in terms of EER [2]. Subsequently we used the CQT:RDF-A and CQT:RDF-DA with network structure 9493:3072:1536:863 to evaluate the RDF performance on ASVspoof2017-V2 evaluation set. In the respect of dynamic feature comparison for detection, no matter which network structure of BLSTM is, CQT:RDF-A gives the best performance on ASVspoof2017-V2 development set. CQT:RDF-D gives the second best performance for the first network structure, while the CQT:RDF-DA gives the second best performance for the last three network structures.

B. Experimental Analysis

ASVspoof 2017 focused on the development of replay attack countermeasures, its evaluation entailed highly heterogeneous acoustic recording and playback conditions which increased the EER of a baseline ASV system extremely [19].

To evaluate the device feature on the evaluation set, the RDF neural networks were kept unchanged [3], 4,724 utterances from training and development sets were used to re-train only the DNN models for the classification in spoofing detection. Our experiment on ASVspoof2017-V2 evaluation set shows that the EERs using the CQT:RDF-A and CQT:RDF-DA are 17.00% and 15.08% respectively. It indicates the CQT:RDF-DA performs better than the CQT:RDF-A. In contrast to the EER result in Table II, we found an interesting phenomenon, the CQT:RDF-A performs better than the CQT:RDF-DA on the ASVspoof2017-V2 development set while the CQT:RDF-DA performs better than the CQT:RDF-A on the ASVspoof2017-V2 evaluation set. The reason may be certain differences between the ASVspoof2017-V2 development set and the ASVspoof2017-V2 evaluation set, for example, some types of replay speech in evaluation set do not appear in development set. Consequently, in the remainder of the paper, we only use DA dynamic feature for all features on the evaluation set.

In handcrafted features (e.g. MFCC and CQCC), DCT [54] has three functions: de-correlation of the feature dimensions [55], energy concentration and dimension reduction. In this paper, we investigate whether DCT has the same functions in RDF extraction as in handcrafted feature extraction. We also investigate the effects of DCT in the RDF feature using different types of neural networks including DNN, LSTM and BLSTM as RDF extractors. From Table III, it can be seen that the performance of the RDF with DCT is much better than the RDF without DCT for both DFT and CQT, which means that DCT plays a significant role in the RDF.

[2]The dimension of 9493 in the neural network input layer is the outcome of splice of eleven frames with feature dimension 863 for CQT situation.

[3]i.e. the neural network models are the same as those used in the above development section.

TABLE II
EXPERIMENTAL RESULTS (EER (%)) ON ASVSPOOF2017-V2 DEVELOPMENT SET USING DIFFERENT BLSTM CONFIGURATIONS AND DIFFERENT DYNAMIC FEATURES.

BLSTM Configuration	Dynamic Feature		
ILN : HL1N: HL2N: OLN	D	A	DA
9493 : 863 : 1536 : 863	40.21	37.38	41.92
9493 : 2048 : 1536 : 863	41.95	39.97	41.21
9493 : 3072 : 1536 : 863	40.95	37.19	40.44
9493 : 4096 : 1536 : 863	40.31	38.00	39.82

TABLE III
EXPERIMENTAL RESULTS IN TERMS OF EER (%) ON ASVSPOOF2017-V2 EVALUATION SET COMPARISON BETWEEN RDF-WITH-DCT AND RDF-WITHOUT-DCT.

Feature — Neural Network	DNN	LSTM	BLSTM
DFT: RDF-with-DCT	30.21	30.72	30.18
DFT: RDF-without-DCT	31.69	32.84	32.04
CQT: RDF-with-DCT	20.73	15.68	15.08
CQT: RDF-without-DCT	23.22	18.61	18.41

In our experiments, the output layer of CQT:RDF neural networks was always of 863 neurons for DNN, LSTM and BLSTM; while the output layer of DFT:RDF neural networks was always of 257 neurons for DNN, LSTM and BLSTM. From the above experiments, we can observe that CQT:BLSTM is the best among the three types of neural networks.

In traditional empirical knowledge, DCT can be used to reduce the dimensions, like MFCC. The dimension of static feature is usually selected to be low such as 13, 20 or 30 because the low dimension is able to concentrate most of energy and thus reduce the sensitivity to some interference. Consequently, in conventional classification, the low dimension selection gives the best performance since the robustness is strengthened. For RDF neural network training, it is necessary to investigate the dimension selection after applying DCT.

In order to investigate the size of the CQT:RDF-with-DCT dimension, not only the low dimensions but also high dimensions were surveyed. From Table IV, it can be seen that low static dimension such as 13, 20 and 30 gives poor performance. For this point, it is very different from handcrafted feature. When static dimension equals 863, the RDF performance reaches the best. The reason may be that the discriminative information of the RDF is hidden not only in low dimensions but also in high dimensions after DCT. In other words, DCT is helpful for de-correlation, but cannot be used for dimension reduction for the device feature.

TABLE IV
EXPERIMENTS FOR DCT DIMENSION SELECTION BASED ON PERFORMANCE IN TERMS OF EER (%) CONDUCTED ON ASVSPOOF2017-V2 EVALUATION SET.

DCT dimension	13	20	30	256	512	863
CQT: DNN:DCT	33.69	29.18	26.42	19.78	19.34	20.73
CQT: LSTM:DCT	25.17	21.67	19.83	15.72	15.76	15.68
CQT:BLSTM:DCT	36.25	35.59	37.06	33.80	29.41	15.08

C. Performance Measurement

In this performance evaluation, we adopt CQT:BLSTM-with-DCT as the RDF feature extractor to compare with other features.

TABLE V
COMPARISON WITH SOME COMMONLY USED ACOUSTIC FEATURES ON ASVSPOOF2017-V2 EVALUATION SET IN TERMS OF EER (%).

Features	EER	Features	EER
IFCC	34.61	RDF+IFCC	14.83
MFCC	23.79	RDF+MFCC	12.50
CQLM	16.61	RDF+CQLM	10.78
LFCC	16.60	RDF+LFCC	10.12
CQCC	15.46	RDF+CQCC	10.62
RDF	15.08		
CQCCE	10.61	RDF+CQCCE	8.99
CQCCE+CQCC	10.61	RDF+CQCCE+CQCC	8.99
CQCCE+CQLM	10.61	RDF+CQCCE+CQLM	8.99
CQCCE+LFCC	10.56	RDF+CQCCE+LFCC	8.98
CQCCE+IFCC	10.25	RDF+CQCCE+IFCC	8.81
CQCCE+MFCC	9.58	RDF+CQCCE+MFCC	8.67
CQCCE+LFCC+CQLM	10.56		
CQCCE+LFCC+IFCC	10.25		
CQCCE+IFCC+CQLM	10.24		
CQCCE+MFCC+CQLM	9.59		
CQCCE+MFCC+LFCC	9.58		
CQCCE+MFCC+IFCC	9.02		

1) Comparison with acoustic features: Recently, CQCC was reported to be sensitive to the general forms of spoofing attacks so that it becomes an effective spoofing countermeasure [56]. There are five modules in CQCC extraction: *CQT*, *Power Spectrum*, *Log*, *Uniform Resampling* and *DCT*. The role of each module is briefed as follows: *CQT* is used to convert speech from the time domain to the CQT spectral domain; it is followed by *Power Spectrum* which is used to calculate the octave power spectrum; afterwards *Log* is applied to obtain octave power spectrum in logarithm scale; then *Uniform Resampling* is adopted to convert the logarithm octave power spectrum into the linear logarithm power spectrum; and finally *DCT* is used to extract principal information by attempting to decorrelate the intermediate coefficients to deduce the dimensions. Since the log energy has been found to improve the performance of CQCC [51], CQCCE is selected for comparison.

In this paper, we shall show the following five conventional features for comparison. They are CQCC, MFCC, instantaneous frequency cosine coefficient (IFCC) [57], LFCC [31] [58], and CQT log-spectral-magnitude (CQLM). The different DNN classifiers are trained for the spoofing detection with different features using the same training database as that with RDF. The static dimensions of 13, 20, 20, 30, 31 and 863 are properly selected for MFCC, IFCC, LFCC, CQCC, CQCCE and CQLM respectively.

Table V gives the performance comparison between the proposed feature and typical conventional features on the ASVspoof2017-V2 evaluation set. It can be seen that the performance of CQLM, CQCC, CQCCE and RDF are better than MFCC. The reason may be that CQT is a long-term transform and it can provide more frequency details than the discrete Fourier transform that is a short-term transformation. It also can be seen that though RDF only consists of devices information, its performance is better than those of CQLM and CQCC which have multi-aspect information. The performance of CQCCE is much better than the RDF, which means that log energy is very helpful and is of much complementary space with CQCC for replay speech detection.

In Table V, 'α + β' denotes the two systems 'α' and 'β' to be fused at the score level. It can be seen that: (1) By score fusion, the EER of RDF + MFCC drops down to 12.50% from its original EER values of 15.08% (RDF) and 23.79% (MFCC). It means that RDF has much complementary space with MFCC. (2) Though RDF is built up from CQLM, the score level fusion of RDF + CQLM improves much from its original respective systems. The reason may be that most of the information in RDF is device and environment information while CQLM has multi-aspect information, and the information in RDF and in CQLM is complementary with each other. (3) RDF also has much complementary information with CQCC and CQCCE, it is found that the EER of RDF + CQCCE reaches 8.99%. (4) The fusion of three systems (RDF+CQCCE+MFCC) gives the best performance with an EER of 8.67%.

TABLE VI
COMPARISON WITH DIFFERENT DEEP FEATURE BASED REPLAY DETECTION SYSTEM ON ASVSPOOF 2017 EVALUATION SET IN TERMS OF EER (%).

Deep Feature	Tr. target	Detect.	Fusion	EER
FFT-CNN [34]	discrim.	RNN	-	10.69
MFCC-DNN [23]	discrim.	RNN	-	12.87
MFCC-LSTM [23]	discrim.	RNN	-	14.42
DFT-DNN	discrim.	DNN	-	15.96
DFT-DNN	discrim.	DNN	CQCCE	10.34
DFT-BLSTM	discrim.	DNN	-	38.23
DFT-BLSTM	discrim.	DNN	CQCCE	10.57
MFCC-DNN	discrim.	DNN	-	17.05
MFCC-DNN	discrim.	DNN	CQCCE	9.68
MFCC-BLSTM	discrim.	DNN	-	38.75
MFCC-BLSTM	discrim.	DNN	CQCCE	10.61
CQT-BLSTM	discrim.	DNN	-	27.65
CQT-BLSTM	discrim.	DNN	CQCCE	10.05
RDF:CQT-BLSTM	regression	DNN	-	15.08
RDF:CQT-BLSTM	regression	DNN	CQCCE	8.99

2) Comparison with deep features: In this session, we investigate the performances among different deep features for the spoofing detection in terms of EER through ultimate detection systems. So far, almost all the neural networks for the deep feature extraction are trained under supervised learning style where the targets of neural networks are set to discriminative labels with spoofing classes. And the deep features are extracted from the last hidden layer of the neural networks. In this paper, we state the training target for these deep features as discriminative modeling. Under the discriminative modeling, the training databases are not required to contain parallel speech signals to mapping with the input speech signals; the only requirement is the categories of the speech signals. As long as we have a label of a speech utterance, the nonparallel training data can be adopted. All the above mentioned deep features are based on discriminative output of neural networks which target to the clustering labels. The neural networks

aim to bring the solution of the classification problem using nonparallel training databases.

Table VI demonstrates the performance of different deep features in terms of EER via certain detection systems conducted on the ASVspoof2017-V2 evaluation set. It consists of the above-mentioned nonparallel trained deep features including **FFT-CNN**, **MFCC-DNN**, **MFCC-LSTM**, **DFT-DNN**, **DFT-BLSTM**, **MFCC-BLSTM** and **CQT-BLSTM**.

In the table [4], **FFT-CNN** represents log power magnitude of FFT as the input acoustic feature for CNN to learn deep feature; **MFCC-DNN** and **MFCC-LSTM** represent MFCC as the acoustic feature input to DNN and LSTM respectively to learn respective deep features; The experiments reported from [23] and [34] used the deep features as input to train respective RNNs for spoofing classification in the replay detection systems; while in our experiments, we used the deep features as input to DNNs for the purpose of spoofing classification.

DFT-DNN denotes the log-spectral-magnitude of DFT as acoustic feature input to DNN to learn the deep feature generator; and **DFT-BLSTM** represents the log-spectral-magnitude of DFT as acoustic feature input to BLSTM network to learn the deep feature generator. The dimension of the DFT acoustic feature is 257 and the output layer contains two neurons to represent spoofing or genuine, the number of neurons in the last hidden layers of DNN and BLSTM are set to 257 respectively. **MFCC-BLSTM** is the deep feature extracted from the last hidden layer of BLSTM whose input layer is from the 13 dimension MFCC feature and the output is targeted to two classes indicating spoofing and genuine probabilities. **CQT-DNN** denotes log-spectral-magnitude of CQT as input to DNN to generate deep feature; and **CQT-BLSTM** denotes log-spectral-magnitude of CQT as input to BLSTM neural network to generate deep feature. The dimension of the CQT acoustic feature is 863 and the output layer contains two neurons, the number of neurons in the last hidden layers of DNN and BLSTM are set to 863 respectively. Except the spoofing detection systems reported from [23] and [34], the classification parts of the other detection systems listed in the table are built using DNN with two hidden layers. From the table, it can be seen that the detection systems with DNN-based discriminative deep features outperforms those with BLSTM-based discriminative deep features; and the CQT with BLSTM discriminative modeling demonstrates better deep feature performance than the DFT and MFCC features with BLSTM do.

In contrast to the traditional discriminative deep feature extractors that are trained using nonparallel training data, our proposed RDF deep feature extractor is generative modeling and trained using parallel training data. Different from the discriminative deep features which are extracted from the last hidden layers of neural networks, the RDF feature is extracted from the output layer of the neural network which is set for the solution of regression problem targeting to depict the characteristics of virtual device feature. The key point of the idea is to prove that the device feature can be depicted by using the information from the input speech and its corresponding genuine speech in log-spectral-magnitude domain. For the neural network structure, the output layer should be constrained to the spectral domain as indicated by Eq. (10) in session III-B. However, the input speech feature is not limited to certain transformation, i.e. handcrafted features such as CQCC and MFCC can also be used as input feature to the neural network for RDF generator. **RDF:CQT-BLSTM** in Table VI denotes our proposed RDF feature. In **RDF:CQT-BLSTM**, the log-spectral-magnitude in CQT domain is selected as input feature to the input layer of BLSTM neural network to generate the RDF deep feature.

The single RDF detection system performs not so good as compared to some nonparallel trained models reported in [23] and [34]. It may be due to the limited amount of parallel pairs used for training and the different detection-purpose neural network architectures where RNNs were used in the three precedent systems, i.e. **FFT-CNN** [34], **MFCC-DNN** [23], and **MFCC-LSTM** [23]. It is also observed that, with the similar DNN used for detection-purpose classification, the proposed **RDF:CQT-BLSTM** performs the best among the deep features with BLSTM feature generators. From the table, it can be seen that the single **RDF:CQT-BLSTM** based detection system reaches EER of 15.08% while the discriminative modeling based detection system **CQT-BLSTM** gives EER of 27.65%.

In the experiments, we investigate the feature complementarity by using the score fusion with CQCCE detection system. The CQCCE detection system was briefly described in sessions III-D and IV-C1. The resulting EERs after the score level fusion indicate that the improvement from the fusion with CQCCE detection system is more or less for all of the investigated deep feature systems by notice of 10.61% EER from single CQCCE system. From this CQCCE perspectives, the improvement of discriminative deep feature **CQT-BLSTM** is about 5.27% (EER only drops to 10.05% from 10.61%). It can be seen that the proposed **RDF:CQT-BLSTM** shows strong complementarity with CQCCE at score level comparing with others in terms of EER. And the **RDF:CQT-BLSTM** fused with CQCCE performs the best among all the fused systems. The EER improvement of CQCCE system after fusion with **RDF:CQT-BLSTM** is about 15.27% (EER drops from 10.61% to 8.99%).

Usually, it is a fact that the parallel data is relatively hard for collection as compared to the nonparallel data. Fortunately, ASVspoof2017-V2 provides quite amount of replay pair data, on which we give the evidence for the RDF effectiveness. However, ASVspoof2017-V2 data is still not big enough to convince the consistency of the proposed replay countermeasure. In order to give further inference, we conducted another testing beyond the experiments reported in this paper; by adding more genuine pair (genuine-to-genuine) from ASVspoof2017-V2 development dataset, we observe that the RDF performance is improved apparently. In our future study, the RDF method will be further assessed with larger database covering more devices, speakers and environments.

[4] "Tr. target" and "discrim." mean the training target is set to discriminative modelling while "regression" means the neural network is targeted to regress to the characteristics of virtual device property.

V. Conclusion

In this paper, we have proposed a parallel neural network training architecture for replay device feature extraction which aims to detect the replay spoofing attack for automatic speaker verification. Device feature provides a way to depict the characteristics of the replay spoofing device. In contrast with existing methods for replay speech detection, this paper addresses the problem in extracting device information for replay spoofing detection. In this parallel training framework, different neural networks have been investigated for the RDF extraction. A method to extract RDF based on CQT, BLSTM and DCT shows the effectiveness of the RDF deep feature architecture. In addition, DCT only plays the role of decorrelation of the feature coefficients but cannot be used to reduce the dimensionality for the RDF. The experimental results show that the hybrid RDF related spoofing detection system yields satisfactory results. It has been observed that the proposed RDF has much complementary capability with typical conventional features at score level.

References

[1] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digit. Signal Process.*, vol. 10, pp. 19-41, 2000.

[2] A. Larcher, K. A. Lee, B. Ma, and H. Li, "Imposture classification for text-dependent speaker verification," *IEEE Intern. Conf. on Acoust., Speech, and Sig. Proce.*, ICASSP, 2014, pp. 739–743.

[3] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Sig. Process. Lett.*, vol. 13, pp. 308-311, 2006.

[4] C. H. You, K. A. Lee and H. Li, "GMM-SVM Kernel with a Bhattacharyya-based distance for speaker recognition," *IEEE Trans. Audio, Speech and Lang. Process.*, vol. 18, no. 6, pp. 1300-1312, Aug. 2010.

[5] P. Kenny, "Joint factor analysis of speaker and session variability: theory and algorithms," CRIM, Montreal, *Technical Report, CRIM-06/08-13*, 2005.

[6] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice Modeling with Sparse Training Data," *IEEE Trans. Speech and Audio Process.*, vol. 13, no.3, pp. 345-359, 2005.

[7] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Trans. Audio, Speech and Lang. Process.*, vol. 15, no. 4, pp. 1435-1447, May 2007.

[8] M. McLaren, and D. van Leeuwen, "Source-normalised-and-weighted LDA for robust speaker recognition using i-vectors," *IEEE Intern. Conf. on Acoust., Speech, and Sig. Proce.*, ICASSP, May. 2011, pp. 5456-5459.

[9] S.J.D. Prince, "Probabilistic linear discriminant analysis for inferences about identity," in *Proc. Intern. Conf. on Computer Vision*, Oct. 2007, pp. 1-8.

[10] P. Kenny, "Bayesian speaker verification with heavy-tailed priors," in *Proc. Odyssey 2010 - The Speak. and Lang. Recog. Workshop*, ODYSSEY, 2010.

[11] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," *Conf. of the Intern. Speech Comm. Asso.*, INTERSPEECH, Aug. 2011, pp. 249–252.

[12] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: robust DNN embeddings for speaker recognition," *IEEE Intern. Conf. on Acoust., Speech, and Sig. Proce.*, ICASSP, April. 2018, pp. 5329–5333.

[13] L. Xu, R. K. Das, E. Ylmaz, J. Yang, H. Li, "Generative x-vectors for text-independent speaker verification," *IEEE Spoken Language Technology Workshop*, SLT 2018, Dec. 2018, pp. 1014–1020.

[14] Z. Wu, T. Virtanen, E. S. Chng, and H. Li, "Exemplar-based sparse representation with residual compensation for voice conversion," *IEEE/ACM Trans. Audio, Speech and Lang. Process.*, vol. 22, no. 10, pp. 1506–1521, 2014.

[15] X. Tian, S. W. Lee, Z. Wu, E. S. Chng, and H. Li, "An example-based approach to frequency warping for voice conversation," *IEEE/ACM Trans. Audio, Speech and Lang. Process.*, vol. 25, no. 10, pp. 1863–1875, 2017.

[16] J. Yang and R. K. Das, "Improving anti-spoofing with octave spectrum and short-term spectral statistics information," *Applied Acoustics*, vol. 157, pp. 30–39, 2020.

[17] M. Shannon, H. Zen, and W. Byrne, "Autoregressive models for statical parametric speech synthesis," *IEEE Trans. Audio, Speech and Lang. Process.*, vol. 21, no. 3, pp. 587–597, 2013.

[18] T. Kinnunen, M. Sahidullah, M. Falcone, L. Costantini, R. G. Hautamaki, D. Thomsen, A. Sarkar, Z.-H. Tan, H. Delgado, M. Todisco, N. Evans, V. Hautamaki, and K. A. Lee, "RedDots replayed: a new replay spoofing attack corpus for text-dependent speaker verification research," *IEEE Intern. Conf. on Acoust., Speech, and Sig. Proce.*, ICASSP, 2017, pp. 5395–5399.

[19] T. Kinnunen, and H. D. Md Sahidullah, M. Todisco, N. Evans, J. Yamagishi, and K. A. Lee, "The ASVspoof 2017 challenge: assessing the limits of replay spoofing attack detection," *Conf. of the Intern. Speech Comm. Asso.*, INTERSPEECH, Aug. 2017, pp. 2–6.

[20] J. Yang and R. K. Das, "Low frequency frame-wise normalization over constant-Q transform for playback speech detection," *Digit. Signal Process.*, vol. 80, pp. 30–39, 2019.

[21] J. Yang, L. Liu and Q. He, "Discriminative feature based on FWMW for playback speech detection," Electronics Letters, vol. 55, no. 15, pp. 861–864, 2019.

[22] R. G. Hautamäki, T. Kinnunen, V. Hautamäki, and A. M. Laukkanen, "Automatic versus human speaker verification:the case of voice mimicrys," *Speech Comm.*, vol. 72, pp. 13–31, 2015.

[23] Z. Chen, W. Zhang, Z. Xie, X. Xu, and D. Chen, "Recurrent neural networks for automatic replay spoofing attack detection," *IEEE Intern. Conf. on Acoust., Speech, and Sig. Proce.*, ICASSP, 2018, pp. 2052–2056.

[24] J. Yang, R. K. Das and N. Zhou, "Extraction of octave spectra information for spoofing attack detection," *IEEE/ACM Trans. Audio, Speech and Lang. Process.*, vol.27, no.12, pp. 2373-2384, 2019.

[25] Z. Wang, Q. He, X. Zhang, H. Luo, and Z. Su, "Playback attack detection based on channel pattern noise," *Journal of South China University of Technology (Natural Science Edition)*, pp. 1708–1713, 2011.

[26] Z. Wu, T. Kinnunen, N. Evans, J. Yamagishi, C. Hanilc, M. Sahidullah, and S. Aleksandr, "ASVspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge," *Conf. of the Intern. Speech Comm. Asso.*, INTERSPEECH, 2015, pp.2037-2041.

[27] F. Tom, M. Jain, and P. Dey, "End-to-end audio replay attack detection using deep convolutional networks with attention," *Conf. of the Intern. Speech Comm. Asso.*, INTERSPEECH, Sep. 2018, pp. 681–685.

[28] J. Lindberg and M. Blomberg, "Vulnerability in speaker verification: A study of technical impostor techniques," *6th Euro. Conf. on Speech Comm. and Tech.*, EUROSPEECH, 1999, pp. 5–9.

[29] J. Villalba and E. Lleida, "Speaker verification performance degradation against spoofing and tampering attacks," *FALA 10 Workshop*, 2010, pp. 131–134.

[30] M. Todisco, H. Delgado, and N. Evans, "Constant Q cepstral coefficients: a spoofing countermeasure for automatic speaker verification," *Computer Speech Language*, vol. 45, pp. 516–535, 2017.

[31] R. Font, J. M. Espin, M. J. Cano, "Experimental analysis of features for replay attack detectionResults on the ASVspoof 2017 Challenge," *Conf. of the Intern. Speech Comm. Asso.*, INTERSPEECH, Aug. 2017, pp. 7-11.

[32] T. hasan, S. O. Sadjadi, G. Liu, N. Shokouhi, H. Boril, and J. H. L. Hansen, "CRSS systems for 2012 NIST speaker recognition evaluation," *IEEE Intern. Conf. on Acoust., Speech, and Sig. Proce.*, ICASSP, 2013, pp. 6783-6787.

[33] J. Yang, C. H. You, and Q. He, "Feature with complementarity of statistics and principal information for spoofing detection," *Conf. of the Intern. Speech Comm. Asso.*, INTERSPEECH, Sep. 2018, pp. 651–655.

[34] G. Lavrentyeva, S. Novoselov, E. Malykh, A. Kozlov, O. Kudasher, and V. Shchemelinin, "Audio replay attack detection with deep learning framework," *Conf. of the Intern. Speech Comm. Asso.*, INTERSPEECH, Aug. 2017, pp. 82–86.

[35] Y. M. Qian, N. X. Chen, and K. Yu, "Deep features for automatic spoofing detection," *Speech Comm.*, vol. 85, pp 43-52, Dec 2016.

[36] P. Nagarsheth, E. Khoury, K. Patil, and M. Garland, "Replay attack detection using DNN for channel discrimination," *Conf. of the Intern. Speech Comm. Asso.*, INTERSPEECH, Aug. 2017, pp. 97–101.

[37] A. Gomez-Alanis, A. M. Peinado, J. A. Gonzalez, and A. M. Gomez, "A deep identity representation for noise robust spoofing detection," *Conf. of the Intern. Speech Comm. Asso.*, INTERSPEECH, Sep. 2018, pp. 676-680.

[38] C. H You, J. Chen, and H. D. Tran, "Device feature extractor for replay spoofing detection," *Conf. of the Intern. Speech Comm. Asso.*, INTERSPEECH, Sep. 2019, pp. 2933–2937.

[39] T. Thireou and M. Reczko, "Bidirectional long short-term networks for predicting the subcellular localization of eukaryotic proteins," *IEEE/ACM Trans. on Comput. Biology and Bioinform.*, vol. 4, no. 3, pp. 441–446, 2007.

[40] M. Withowski, S. Kacprasko, P. Zelasko, K. Kowlczyk, and J. Galka, "Audio replay attack detection using high-frequency features," *Conf. of the Intern. Speech Comm. Asso.*, INTERSPEECH, Aug. 2017, pp. 27–31.

[41] H. A. Patil, M. R. Kamble, T. B. Patel, and M. Soni, "Novel variable length teager engery separation based on instantaneous frequency features for replay detection," *Conf. of the Intern. Speech Comm. Asso.*, INTERSPEECH, Aug. 2017, pp. 12–16.

[42] J. Yang, R. K. Das, and H. Li, "Extended constant-Q cepstral coefficients for detection of spoofing attacks," *Asia-Pacif. Sig. and Inform. Process. Assoc.)*, APSIPA, Nov. 2018, pp. 1024-1029.

[43] P. Nagarshenth, E. Khoury, K. Patil, and M. Garland, "Replay attack detection using DNN for channel discrimination," *Conf. of the Intern. Speech Comm. Asso.*, INTERSPEECH, Aug. 2017, pp. 97–101.

[44] X. Wang, Y. Xiao, and X. Zhu, "Feature selection based on CQCCs for automatic speaker verification spoofing," *Conf. of the Intern. Speech Comm. Asso.*, INTERSPEECH, Aug. 2017, pp. 32–36.

[45] W. Cai, D. Cai, W. Liu, G. Li, and M. Li, "Countermeasures for automatic speaker verification repaly spoofing attack: on data augmentation, feature representation, classification and fusion," *Conf. of the Intern. Speech Comm. Asso.*, INTERSPEECH, Aug. 2017, pp. 17–21.

[46] K N R K R Alluri, S. Achanta, Sudarsana, S. R. Kadiri, S. V Gangasheetty, and J. Vuppala, "SFF anti-spoofer: IIIT-H submission for automatic speaker verificatio spoofing and countermeasures channlenge 2017," *Conf. of the Intern. Speech Comm. Asso.*, INTERSPEECH, Aug. 2017, pp. 107–111.

[47] F. Seide and A. Agarwal, "CNTK: Microsoft's open-source deep learning toolkit," *Proc. of the 22nd ACM SIGKDD Intern. Conf. on Know. Disc. and Data Min.*, 2016, pp. 2135–2135.

[48] J. Youngberg and S. Boll, "Constant-Q signal analysis and synthesis," *IEEE Intern. Conf. on Acoust., Speech, and Sig. Proce.*, ICASSP, 1978, pp. 375–378.

[49] J C. Brown, "Calculation of a constant Q spectral transform," *Journ. of Acous. Soc. of Amer.*, vol. 89, no. 1, 425–434, 1991.

[50] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*. vol. 9, no. 8, pp. 1735–1780, 1997.

[51] H. Delgado, M. Todisco, M. Sahidullah, N. Evans, T. Kinnunen, K. A. Lee, and J. Yamagishi, "ASVspoof 2017 version 2.0: meta-data analysis and baseline enhancements," *Odyssey 2018: The Speak. and Lang. Recog. Workshop*, ODYSSEY, 2018, pp. 296–303.

[52] J. Yang and L. Liu, "Playback speech detection based on magnitude-phase spectrum," *Electronics Letters*, vol. 54, no. 14, pp. 901–903, 2018.

[53] M. Todisco, H. Delgado, and N. Evans, "Constant Q cepstral coefficients: A spoofing countermeasure for automatic speaker verification," *Computer Speech & Language*, vol. 45, pp. 516–535, 2017.

[54] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. on Computers*, vol. C-23, pp. 90–93, 1974.

[55] Q. Li and Y. Huang, "An auditory-based feature extraction algorithm for robust speaker identification under mismatched conditions," *IEEE Trans. Audio, Speech and Lang. Process.*, vol. 19, no. 6, pp. 1791–1801, 2011.

[56] M. Todisco, H. Delgado, and N. Evans, "A new feature for automatic speaker verification anti-spoofing:constant Q cepstral coefficients," *Odyssey 2016: The Speak. and Lang. Recog. Workshop*, ODYSSEY, 2016, pp. 283–290.

[57] K. Vijayan, P. R. Reddy, and K. S. R. Murty, "Significance of analytic phase of speech signals in speaker verification," *Speech Comm.*, vol. 81, pp. 54-71, 2016.

[58] F. Alegre, A. Amehraye, and N. Evans, "A one-class classification approach to generalised speaker verification spoofing countermeasures using local binary patterns," *IEEE Sixth Intern. Conf. on Biometrics: Theory, App. and Sys.*, BTAS, 2013, pp. 1–8.

Chang Huai You received the B.Sc. degree in physics and wireless from Xiamen University, Xiamen, China, in 1986, the M.Eng. degree in communication and electronics engineering from Shanghai University of Science and Technology, Shanghai, China, in 1989, and the Ph.D. degree in electrical and electronic engineering from Nanyang Technological University (NTU), Singapore, in 2006. From 1989 to 1992, he was an Engineer with Fujian Hitachi Television Corporation, Ltd., Fuzhou City, China. From 1992 to 1998, he was an Engineering Specialist with Seagate International, Singapore. He joined the Centre for Signal Processing, NTU, as a Research Engineer in 1998 and became a Senior Research Engineer in 2001. In 2002, he joined the Agency for Science, Technology, and Research, Singapore, and was appointed as a Member of Associate Research Staff. Since 2003, he has been a Scientist with the Institute for Infocomm Research, Singapore. His research interests include speaker and language recognition, speech recognition, speech synthesis, speech enhancement, microphone array beamforming, audio signal processing, and image processing. He is the reviewer of many journals and international conferences.

Jichen Yang received Ph.D degree in Communication and Information System from South China University of Technology (SCUT), Guangzhou, China in 2010. He was a Postdoctoral Research Fellow from October 2011 to March 2016 in SCUT. Since April 2016, he has been a Postdoctoral Researcher Fellow initially at the Department of Human Language Technology, Institute for Infocomm Research (I^2R), A*STAR, Singapore and then in the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. His research interests mainly include anti-spoofing and forensics, speaker recognition and speaker diarization.