Speech Communication 85 (2016) 43–52

Contents lists available at ScienceDirect

Speech Communication

journal homepage: www.elsevier.com/locate/specom

# Deep features for automatic spoofing detection

Yanmin Qian*, Nanxin Chen, Kai Yu*

*Key Lab. of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering SpeechLab, Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China*

ARTICLE INFO

*Article history:*
Received 15 May 2016
Revised 15 October 2016
Accepted 17 October 2016
Available online 20 October 2016

*Keywords:*
Automatic spoofing detection
Deep features
Deep neural network
Recurrent neural network

ABSTRACT

Recently biometric authentication has made progress in areas, such as speaker verification. However, some evidence shows that the technology is susceptible to malicious spoofing attacks, and thus dedicated countermeasures are needed to detect a variety of specific attack types. Inspired by the great success of deep learning in automatic speech recognition, we propose a detailed deep learning based feature engineering framework for spoofing detection in this paper. To incorporate deep learning into spoofing detection, this work proposes novel approaches for extracting and using features from deep learning models. In contrast to the traditional short-term spectral features, such as MFCC or PLP, outputs from the hidden layer of various deep models are employed as *deep features* for spoofing detection. Two frameworks are developed to extract deep features, including DNN-based frame-level feature extraction and RNN-based sequence-level feature extraction, and several structures are explored within each framework. Once the deep features are extracted, they can be used as a spoofing identity representation for each utterance, and the appropriate back-end classifier is then applied to make the final detection decision. These approaches were evaluated on the ASVspoof2015 Challenge data corpus. Experiments show that deep feature based systems achieve good performance, even without using any designed features such as phase and cochlea features common in spoofing detection, and obtain significant performance improvements compared to the traditional baselines. The EER of the best deep feature system achieves nearly 0.0% for all attack types from S1 to S9, and gets 1.1% on all averaged conditions (plus S10), which is very promising performance in ASVspoof2015 Challenge task.

## 1. Introduction

Recently there has been substantial progress on biometric recognition systems, of which automatic speaker verification (ASV) system is one important type. ASV systems take claimed identity and speech samples as input and decide whether to accept or reject the claim. *Text-dependent* and *text-independent* are two categories of typical ASV systems. Text-dependent ASV systems use the fixed phrases within the enrollment and verification stages, while text-independent accepts any speech. Benefiting from a number of technical advances (Chen et al., 2015b; Dehak et al., 2011; Liu et al., 2015a), such as progress on channel and noise compensation (Burget et al., 2007; Hubeika et al., 2008; Solomonoff et al., 2005; Vair et al., 2006), current ASV systems give impressive results on many tasks (Greenberg et al., 2013).

When applying automatic speaker verification to real scenarios, security and robustness become more important. To be applied in scenarios such as mobile payment systems, automobiles or mobile phone unlocking systems, it is crucial to know the robustness of the ASV system against spoofed attacks. As with other biometric recognition systems, these attacks can be generally grouped into two categories:

- **Direct attacks**, also referred to as *spoofing attacks*, can be directly applied without any prior knowledge from the ASV system. For example, the attacker may use generated or altered speech to login as the target speaker.
- **Indirect attacks**, generally require system-level access. For instance, the attacker may attempt to change the running state of any component in the system, such feature extraction, models and decisions.

### 1.1. Spoofing approaches

Since indirect attacks require system-level access, they are relatively difficult to implement. Accordingly, direct attacks are the greatest threat and are the main focus of this paper. Direct spoofing attacks can be commonly categorized into the following four types:

* Corresponding authors.
*E-mail addresses:* yanminqian@sjtu.edu.cn, yanminqian@gmail.com (Y. Qian), bobchennan@gmail.com (N. Chen), kai.yu@sjtu.edu.cn (K. Yu).

http://dx.doi.org/10.1016/j.specom.2016.10.007
0167-6393/© 2016 Elsevier B.V. All rights reserved.

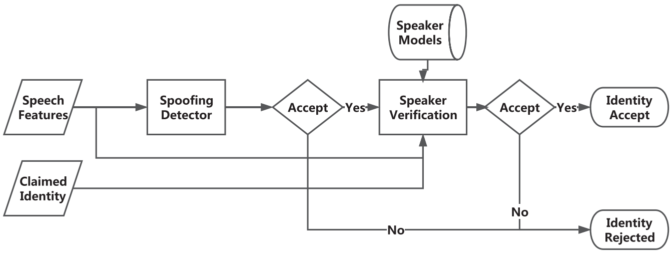

**Fig. 1.** Typical Automatic Speaker Verification (ASV) system.

*Impersonation*: a speaker mimics another person, which requires professional expertise. These attacks have seen little interest in recent research and are generally seen as being less potent. Skilled impersonators, although being potentially serious threats, generally do not occur often. Recent work in Wu et al. (2015a) suggested that there are no consistent results on the detection of impersonation.

*Replay*: an already recorded utterance of a speaker is played into the speaker verification system. Since smartphones are widely available and can easily be used to record and replay speech, this approach is an increasingly practical attack. The attack may be concatenated speech, extracted from a number of shorter segments (Villalba et al., 2015).

*Speech synthesis*: a speech synthesis system trained on speech given by the target speaker is used to generate spoofed utterances from written text. Synthesis has become an increasingly big threat to ASV systems due to the increasing availability of open source toolkits. In addition, the recent advances in synthesis technology make this attack a serious threat for ASV systems.

*Voice conversion*: similar to speech synthesis, this attack type focuses on converting pre-recorded speech from any person to mimic specific vocal features of a target speaker. In contrast to speech synthesis systems which require text input, the input in a voice conversion system is natural speech. Recent progress on voice conversion also makes this easily to implement without help from experts.

### 1.2. Countermeasures

Countermeasures are designed to protect the ASV systems from attacks. Some work has been proposed that use special features to detect specific spoofing attacks. For instance, similarities within stored access attempts (Shang and Stevenson, 2010), spectral ratio and modulation indexes (Villalba and Lleida, 2011a, 2011b), channel noise (Wang et al., 2011) has been demonstrated to be efficient in detecting replay attacks. Phase spectra such as cosine normalised phase and modified group delay phase features (Wu et al., 2012a, 2012b), F0 statistics (De˜Leon et al., 2012; Ogihara et al., 2005), higher order Mel-cepstral coefficients (Chen et al., 2010), intra-frame differences (Satoh et al., 2001) were reported to protect ASV systems from speech synthesis and voice conversion attacks. Other work has focused on modeling, including supervector-based SVM (Alegre et al., 2012; Liu et al., 2015b), i-vector (Khoury et al., 2014; Novoselov et al., 2015; Weng et al., 2015), and have been reported to be robust against artificial signal attacks.

## 2. Spoofing detection

A typical Automatic Speaker Verification (ASV) system is shown in Fig. 1. It contains two parts: the spoofing detector which detects whether the audio was given by the real speaker or not, and speaker verification which verifies the audio against the claimed identity. The spoofing detector is very important because a well-designed spoofing detector can filter most attacks to the ASV system. Traditionally there are two main approaches for spoofing detection.

### 2.1. GMM framework

The GMM–UBM approach has been widely used in speaker verification, and can also be easily extended to spoofing detection. The spoofing detection task can be restated as a basic hypothesis test between

- $H_0$: The utterance $X$ is from one true speaker $S$
- $H_1$: The utterance $X$ is not from any true speakers

Then the decision is made according to the likelihood ratio shown as:

$$\frac{p(X|H_0)}{p(X|H_1)}\begin{cases}\geq\theta & Accept\\ <\theta & Reject\end{cases} \tag{1}$$

where $p$ is the probability density function, also referred as the *likelihood*. The decision threshold for accepting or rejecting is $\theta$, which is obtained from the development set.

Normally genuine utterances and spoofing utterances can be used to train a individual GMM, and then $p(X|H_0)$ and $p(X|H_1)$ are obtained separately, giving a log-likelihood ratio:

$$\text{Score}(X) = \log p(X|\text{GMM}_{\text{genuine}}) - \log p(X|\text{GMM}_{\text{spoofing}}) \tag{2}$$

Based on this basic GMM model, other extended approaches using GMM have also been proposed, such as the supervector method which concatenates all mean vectors from different mixtures in the GMM (Alegre et al., 2012; Liu et al., 2015b).

### 2.2. I-vector framework

The identity vector (*i-vector*) is developed from Joint Factor Analysis (JFA) (Kenny et al., 2007), which is a model representing both speaker and session variability in GMM (Dehak, 2009). A single subspace called total variability is proposed in i-vector approach (Dehak et al., 2011), with speaker and session-dependent GMM supervector $\mathbf{M}$ is defined as:

$$\mathbf{M} = \mathbf{m} + \mathbf{T}\mathbf{w} \tag{3}$$

where $\mathbf{M}$ is the supervector for the utterance, $\mathbf{m}$ is the speaker- and channel-independent supervector, and $\mathbf{T}$ is a low rank matrix of speaker and session variability. The total factor $\mathbf{w}$ is called the identity vector or i-vector, and it usually has a very low dimension compared to $\mathbf{M}$ and $\mathbf{m}$. Normally the cosine similarity classifier is used to do fast scoring and decision within the i-vectors $\mathbf{w_1}$ for enrollment speech and $\mathbf{w_2}$ for the test utterance:

$$\text{Score}(\mathbf{w_1}, \mathbf{w_2}) = \frac{<\mathbf{w_1}, \mathbf{w_2}>}{||\mathbf{w_1}||\ ||\mathbf{w_2}||} \tag{4}$$

where $<\mathbf{w_1}, \mathbf{w_2}>$ is the inner product of two i-vectors, and $||\mathbf{w_1}||$ or $||\mathbf{w_2}||$ is the length of the respective i-vector. In addition, other more advanced back-end classifiers can also be applied with i-vectors to obtain better performance in spoofing detection, including Support Vector Machine (SVM) (Novoselov et al., 2015; Weng et al., 2015) and Deep Neural Networks (DNN) (Tian et al., 2016; Zhang et al., 2016).

## 3. Deep features extraction using deep models

Both in speaker recognition and spoofing detection, feature extraction is very important to system construction. Specially designed features usually have some advantages on certain attacks but may not work well in other conditions. Accordingly, it is important to explore new technologies to obtain more discriminative and effective features for automatic spoofing detection.

Neural networks, especially deep neural networks, have powerful discriminative abilities. The nonlinear modeling ability makes the DNN not only a powerful back-end classifier (Tian et al., 2016; Zhang et al., 2016) but also superior for feature engineering (Liu et al., 2015a), and it has been successful in several speech-based applications, such as speech recognition (Grézl et al., 2007), speaker recognition (Liu et al., 2015a), and speech synthesis (Wu and King, 2016). There have been several works using DNN for spoofing detection, however most of them utilized the DNN as a back-end classifier (Tian et al., 2016; Zhang et al., 2016), and there is still no completed and detailed work using the DNN as a feature engineering for spoofing detection.

Inspired from the previous work using DNN for a feature engineering in speech recognition (Grézl et al., 2007) and speaker recognition (Liu et al., 2015a; Variani et al., 2014), DNN-based features are developed here for spoofing detection. Traditional spectral features, such as Filter Banks (Fbank), Mel Frequency Cepstral Coefficient (MFCC) or Perceptual Linear Predictive (PLP), pass through deep models and the outputs derived from a specific hidden layer can be obtained. These new derived features from the DNN are named *deep features*. It is believed that deep features should be more robust and effective than the conventional spectral-based features due to the powerful ability of deep models. Focusing on deep feature extraction, this paper will develop and investigate several types of deep models to extract effective features representation for spoofing detection. As far as we know, this is also the first comprehensive work on the DNN-based features engineering for spoofing detection.

Considering that spoofing detection focuses on the classification of one entire utterance instead of just each independent frame, an utterance-level identity representation is needed. Aiming at the final utterance-level identity representation for each utterance, two frameworks are proposed: Feed-forward DNN-based frame-level feature extraction and RNN-based sequence-level feature extraction. Diagrams of these two frameworks are shown in Fig. 2, and described in the following sections.

### 3.1. DNN based frame-level feature extraction

The normal feed-forward DNNs take a context window as the input and predict the probability of each class on each frame. To obtain the utterance-level identity representation, the outputs from the hidden layers are averaged across the entire utterance duration, and post-normalization is applied to get the final identity representative vector. Three types of feed-forward DNNs are introduced within this framework.

#### 3.1.1. Deep stacked autoencoder

In order to represent information efficiently, a natural idea is to compress the information in a relatively low dimensional space. Principal component analysis (PCA) is the traditional method, but its linearity property limits its ability. Generative stacked autoencoder (SAE) is an alternate (Bengio et al., 2007), which uses an activation function, such as *sigmoid* to make the whole transformation nonlinear. The stochastic gradient descent (SGD) algorithm is used to train parameters. Another advantage is that no labels are required, and SAE is unsupervised learning, which can utilize a large amount of training data.

Squared error is used as an objective function to measure the difference within reconstructed feature and original input feature. As shown in Fig. 3, at each time, two layers including the encoder and decoder are trained. Then the decoder is removed and the compressed encoder output is utilized as the new input. This approach shows great advantages in feature representations for several tasks (Feng et al., 2014; Gehring et al., 2013; Vincent et al., 2008, 2010).

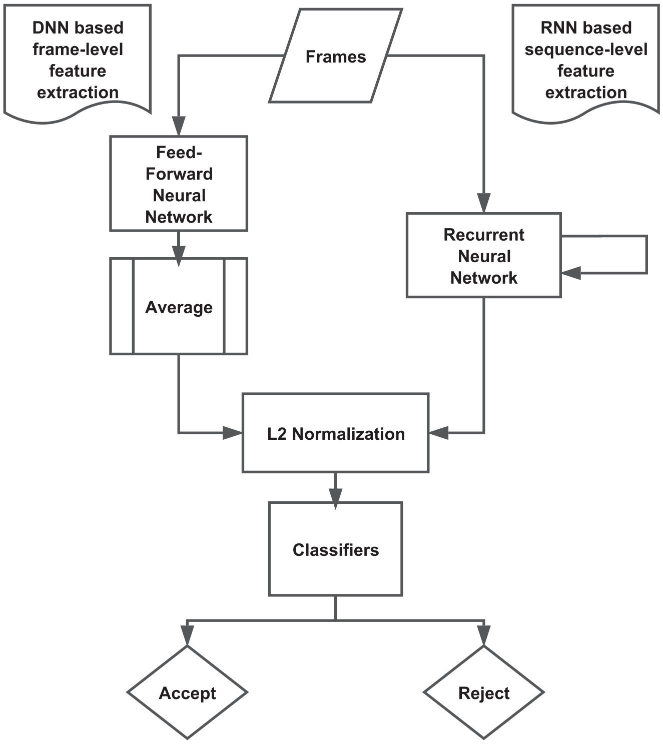
DNN based frame-level feature extraction
Frames
RNN based sequence-level feature extraction
Feed-Forward Neural Network
Recurrent Neural Network
Average
L2 Normalization
Classifiers
Accept
Reject

**Fig. 2.** Deep model based frameworks for spoofing detection: DNN based frame-level feature extraction (left) and RNN based sequence-level feature extraction (right).

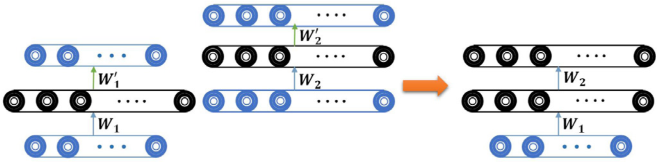

**Fig. 3.** Deep stacked autoencoder with 2 hidden layers.

Once the SAE is trained, the original spectral features are fed through the neural network and the outputs of a particular hidden layer are extracted. The context information can also be encoded using extended inputs across $n$ left and right frames. The final deep features are obtained from the individual hidden layers of the unsupervised trained SAE, and used as inputs for detecting spoofing attacks.

#### 3.1.2. Spoofing-discriminant deep neural network

Spoofing-discriminant deep neural network is a natural choice for spoofing discrimination, as shown in the left part of Fig. 4. This approach was first adopted in our previous work for ASVspoof2015 Challenge (Chen et al., 2015a). The DNN is trained to discriminate the spoofing types known in the corpus, constraining other information, such as phone variability and channel variability at a relative lower level. Spoofing-discriminant DNN, which is targeted to classify spoofing types, is more powerful in information reconstruction and must be more reliable in spoofing information extraction. In the ASVspoof2015 Challenge, there are five spoofing algorithms included in the training data, so for spoofing discrimination the output of the DNN is designed with six classes, including five known spoofing attacks plus the genuine (human) speech class.

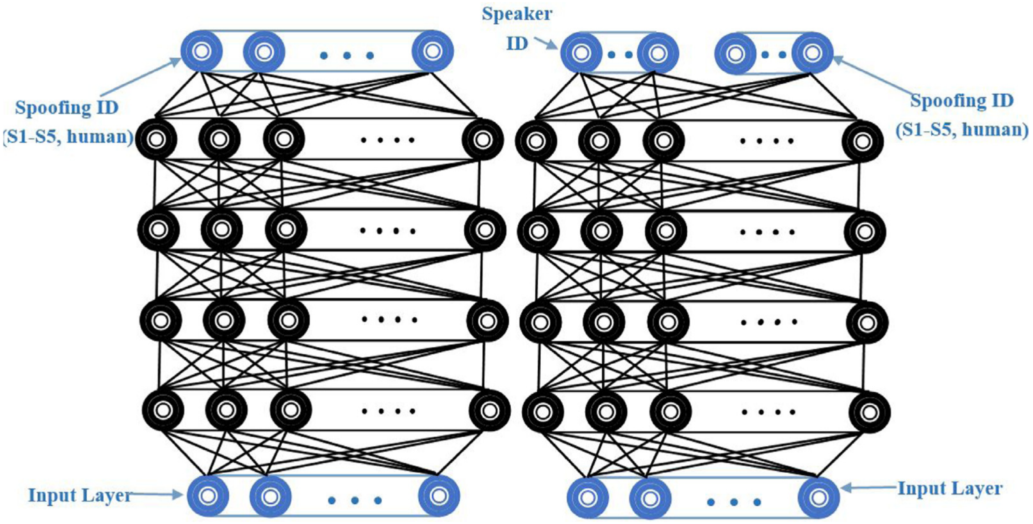

**Fig. 4.** Left: spoofing-discriminant DNN; Right: multi-task joint-learned DNN.

### 3.1.3. *Multi-task joint-learned deep neural network*

Although the spoofing task aims to detect spoofed speech, the speaker information can still be utilized in the genuine speech. In this work, we extend the spoofing-discriminant DNN to utilize speaker information within a multi-task joint-learning framework. Multi-task joint-learning is a method which optimizes the model parameters with multiple criteria simultaneously, and has been implemented in several speech related applications, such as speech recognition (Chen et al., 2014; Seltzer and Droppo, 2013), speaker verification (Chen et al., 2015b; Liu et al., 2015a) and speech activity detection (Jaitly et al., 2014), etc. In this work we apply multi-task joint-learning on the spoofing task, with two output layers to discriminate $\text{Spoof}_i$ and $\text{Speaker}_i$ individually, shown as the right part of Fig. 4. One-hot encoding is used individually and the objective function is the sum of two cross entropy functions. During the training, the gradients are calculated as the sum of two parts given by different output layers. Joint learning avoids over-fitting for DNN training, and also takes knowledge from both spoofing-discrimination and speaker-discrimination, so it is interesting to compare these results and investigate whether speaker-discriminant information can improve the performance of spoofing detection.

Once the multi-task joint-learned DNN is trained, the two output layers can be removed and the remaining hidden layers are used to extract the spoofing-speaker joint representative deep features.

## 3.2. *RNN based sequence-level feature extraction*

The above feed-forward DNN-based framework is trained for classification on a frame-level. An averaging within the whole utterance assumes the independence of each frame and the equal contribution from all the frames. However, this assumption is not very accurate since not all parts have equal importance and the frame sequence is not independent. Accordingly we want to use another model which could utilize all frames dependently, e.g. the whole utterance can be taken into the consideration as the input, and a single representation is created after processing the whole utterance. In order to encode the sequential knowledge in the model, recurrent neural networks (RNN) are used. RNNs have several advantages compared to DNNs, such as the flexibility to process sequential data, and the ability to memory preceding information internally.

Typical RNNs have two categories for sequence labelling:

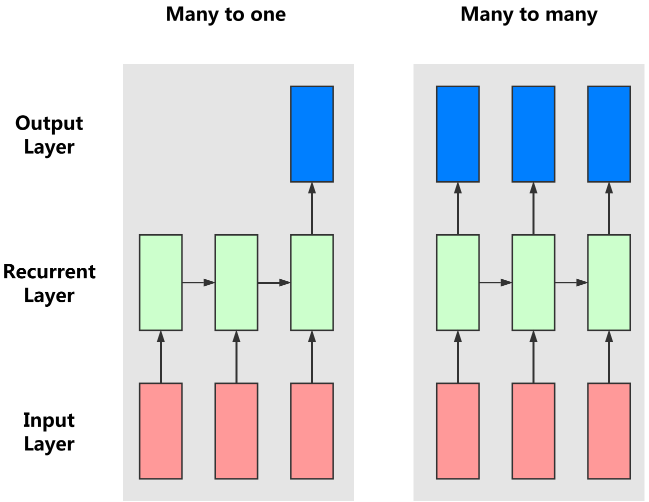

**Fig. 5.** Recurrent neural networks: Left: many input to one output & Right: many input to many output.

- Many to one shown to the left of Fig. 5
- Many to many shown to the right of Fig. 5

The second mode is broadly used in speech recognition and machine translation (Graves et al., 2013; Sutskever et al., 2014), to decode the information at every time step. Spoofing detection can be regarded as an entire-sequence labelling problem, which makes the decision at the final time step, and the outputs from the beginning part of the sequence may not be accurate. Accordingly the first category (many to one) is used in our RNN-based implementation for spoofing detection. The network is spoofing-discriminant, so the output layer contains six classes including the five known spoofing attacks and one genuine (human) speech class. When finishing the RNN training, the output of the hidden layer at the last time step will be used, and shown as Fig. 4 the later L2-normalization is applied to generate the final identity representation for the entire utterance. In this paper, two recurrent models are applied.

### 3.2.1. *Uni-directional long short term memory network*

Long short term memory network (LSTM) (Hochreiter and Schmidhuber, 1997) is one type of recurrent neural network that

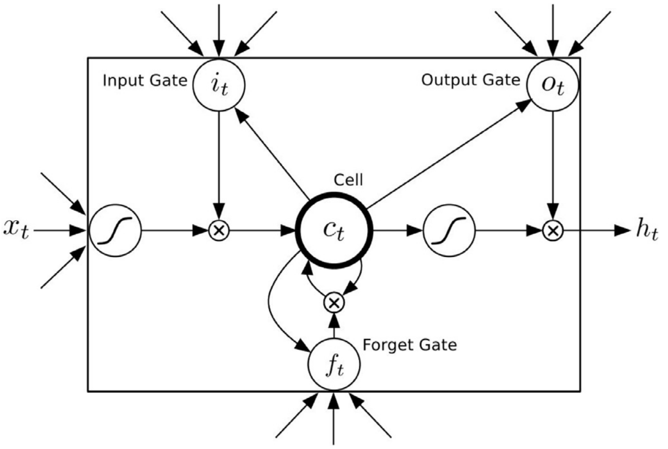

**Fig. 6.** One LSTM memory block (Hochreiter and Schmidhuber, 1997; Srivastava et al., 2015).

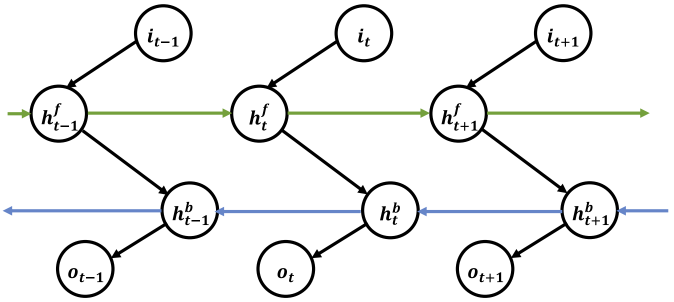

**Fig. 7.** A bi-directional LSTM layer.

has been proven more powerful on sequence generating and sequence labelling than the basic RNN. Compared to the normal RNN model with basic nodes, each node in the LSTM–RNN hidden layer is represented using a LSTM structure. Shown in Fig. 6, a LSTM layer is composed of recurrently connected memory blocks and different gates. These gates perform activities such as read, write, and reset operations (Wöllmer et al., 2011). The overall effect is to allow the network to have abilities to remember and recall information over a long period. The LSTM–RNN was shown better than the basic RNN on many tasks (Gers and Schmidhuber, 2001; Sundermeyer et al., 2012) due to its ability to model information over a long period more accurately and effectively. It is able to overcome the gradient vanishing and exploration problems in training, which is beneficial for model optimization (Hochreiter et al., 2001). The LSTM–RNN is investigated here for the spoofing detection task.

### 3.2.2. Bi-directional long short term memory network

The uni-directional LSTM–RNN only takes forward-direction frames into sequential modeling, which means that the latter sequence can not contribute to the preceding frames modeling. The model accuracy should be enhanced if we can use the whole sequence with both forward and backward directions. Accordingly, another LSTM layer which processes the sequence from the end to the start is added. This model is called a bi-directional long short term memory network (BLSTM) (Schuster and Paliwal, 1997). In BLSTM–RNN, two LSTM layers are used to process in two opposite directions (Graves et al., 2005; Graves and Schmidhuber, 2005).

In our implementation, the normal BLSTM layers are slightly modified, as shown in Fig. 7:

- One LSTM component processes the input sequence in the forward direction
- Another backward-direction LSTM is connected to the outputs of the forward LSTM layer, but in the opposite direction
- Only the outputs of the backward-direction LSTM layer are connected to the softmax layer

It is noted that with this bi-directional LSTM design, the outputs of the backward LSTM layer at the last time step, i.e. the first frame $o_0$ of the sequence in the natural order, are used as the final sequence feature representation, and it can take both the advantages from forward and backward directions.

## 4. Back-end classifiers with deep features

After the deep feature extraction, every utterance can be represented with one identity vector, either from the DNN or the RNN. This is the spoofing identity representation, similar to the normal i-vector in speaker verification (Dehak, 2009), and these spoofing identity vectors can be used with different back-end classifiers. In this paper, several back-end classifiers are investigated and compared.

### 4.1. Linear discriminant analysis

Linear discriminant analysis (LDA) (McLaren and Van Leeuwen, 2011; Scholkopft and Mullert, 1999) provides good generalization capability even with a limited number of training samples. One advantage of this model is that LDA attempts to define new special axes that minimize the intra-class variance caused by channel effects, and to maximize the variance between classes. Due to this it has been used on many tasks related to speaker recognition (Jin and Waibel, 2000; McLaren and Van Leeuwen, 2011). LDA assumes that each class density can be modelled as a multivariate gaussian:

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)=\frac{1}{(2\pi)^{\frac{p}{2}}|\boldsymbol{\Sigma}_k|^{\frac{1}{2}}}\mathrm{e}^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^\top\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)} \tag{5}$$

where $\boldsymbol{\Sigma}_k$ and $\boldsymbol{\mu}_k$ are the covariance and mean for class $k$, $p$ is the dimension of the vector. The LDA model assumes $\boldsymbol{\Sigma}_k=\boldsymbol{\Sigma}, \forall k$, and aims to maximize between-class variance $\boldsymbol{\Sigma}_b$, which equals to maximize the class separation, and the transformation matrix $\boldsymbol{w}$ is estimated by the training data:

$$S=\frac{\boldsymbol{w}^T\boldsymbol{\Sigma}_b\boldsymbol{w}}{\boldsymbol{w}^T\boldsymbol{\Sigma}\boldsymbol{w}} \tag{6}$$

In this implementation, we used six classes during the training (five known spoofing algorithms S1-S5 in training set plus genuine speech), and the “genuine” class is used to score each utterance.

### 4.2. Gaussian density function

Sometimes the estimation of transformation matrix $\boldsymbol{w}$ in LDA does not lead to improvement. Without $\boldsymbol{w}$, the discriminant function for class $k$ can be written as

$$\mathrm{df}_k(\mathbf{x})=-\frac{1}{2}\times(\mathbf{x}-\boldsymbol{\mu}_k)^\top\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_k) \tag{7}$$

For classification, the equation can be expanded and the first term $\mathbf{x}^\top\boldsymbol{\Sigma}\mathbf{x}$ has the same value across all classes. The remaining terms can be written as the linear expression (Gaussian Density Function (Chen et al., 2015b; Liu et al., 2015a), indicated as GDF in the following descriptions), also refered as Gaussian Classifier (Zhang et al., 2016)):

$$\mathrm{df}_k^{'}(\mathbf{x})=(\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k)\mathbf{x}+(-\frac{1}{2}\boldsymbol{\mu}_k{}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k) \tag{8}$$

and the probability can be estimated by Bayes theorem

$$Pr_k(\mathbf{x})=\frac{\exp\left(\mathrm{df}_k^{'}(\mathbf{x})\right)}{\sum_i\exp\left(\mathrm{df}_i^{'}(\mathbf{x})\right)} \tag{9}$$

**Table 1**
The statistics of ASVspoof2015 challenge in the training, development and evaluation sets (Wu et al., 2015b).

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

Since GDF can be regarded as a simplified version of LDA, the same configuration as LDA is adopted.

### 4.3. Support vector machine

A support vector machine separates data points in a high dimensional space defined by a kernel function. In this manner, we first obtain a binary function that describes the probability density function where the normal data lives, corresponding to the genuine speech without spoofing. This function returns +1 in the small region corresponding to the training data and −1 elsewhere. This structure is referred to as a binary SVM. This type of SVM was previously used in spoofing detection but with i-vector (Novoselov et al., 2015; Weng et al., 2015), or supervector features (Liu et al., 2015b). Here it is used with the deep features based spoofing identity vectors, where +1 indicates genuine speech and −1 indicates spoofed speech.

Complex classifiers may overfit for the training spoofs. To create a spoof-independent system, we decided to try a derivative model that can be only trained on non-spoof data. This is a type of one-class SVMs (Schölkopf et al., 2001, 1999), usually used to find abnormal data, which was first tried in spoofing detection but with phase-based features (Villalba et al., 2015). This kind of SVM is also applied here with deep features, and only genuine speech was used to train the one-class SVM model.

## 5. Experiments

To fully explore the effectiveness of the proposed deep features and various classifiers for automatic spoofing detection, experiments and comparison are designed, and evaluations are implemented on the ASVspoof2015 Challenge corpus.

### 5.1. ASVspoof 2015 challenge

The ASVspoof 2015 challenge dataset (Wu et al., 2015b) is designed to give a standard data corpus for research on spoofing detection: it contains genuine and spoofed speech, and covers several commonly used attacks. There is no overlap speakers within training, dev and eval sets. The statistics of genuine speech are shown in Table 1. More details about the data corpus can be found in the challenge introduction paper (Wu et al., 2015b).

The spoofed speech is generated by 10 different voice conversion and speech synthesis approaches. As described in (Wu et al., 2015b), these spoofing techniques, termed from S1–S10, can be grouped into:

- **Voice conversion (VC)**: S1, S2, S5, S6, S7, S8, S9
- **Speech synthesis (SS)**: S3, S4, S10

The spoofed speech in training and development sets are only generated using 5 of the algorithms: S1–S5, which are referred to as **known attacks**, and S6–S10 are referred to as unknown attacks, which only exist in the evaluation. All methods, except S4 and S10, are trained with 20 utterances of the target speaker. The speech synthesis systems of S4 and S10 are trained with 40 utterances per speaker (Wu et al., 2015b). This design enables us to evaluate the effect of the methods on both known and unknown spoofing attacks.

**Table 2**
Baseline performance EER (%) on known conditions.

| Systems | S1 | S2 | S3 | S4 | S5 | Known |
|---|---|---|---|---|---|---|
| GMM | 4.4 | 7.2 | 1.4 | 1.5 | 6.1 | 4.1 |
| i-vector | 4.7 | 4.7 | 2.2 | 2.8 | 6.6 | 4.2 |
| LDA | **2.6** | **2.2** | **1.0** | **1.0** | **4.6** | **2.3** |

**Table 3**
Baseline performance EER (%) on unknown and all conditions.

| Systems | S6 | S7 | S8 | S9 | S10 | Unknown | All |
|---|---|---|---|---|---|---|---|
| GMM | 4.9 | 5.0 | 1.7 | 8.2 | 39.3 | 11.8 | 8.0 |
| i-vector | 6.1 | 6.5 | 8.1 | 3.6 | 45.5 | 14.0 | 9.0 |
| LDA | **4.2** | **3.3** | **0.4** | **2.3** | **38.6** | **9.8** | **6.0** |

### 5.2. Experimental setup and baseline systems

In all experiments, the 24-dimensional static filter bank (FBANK) feature with $\Delta$ was utilized. Three baseline systems are constructed in this work:

- Gaussian mixture model (GMM) system: As described in Section 2.1, a GMM is used to model the input features. The 512 mixture GMM is initialized using the whole training dataset. The Maximum a posteriori (MAP) algorithm is used to adapt the initial model to two GMMs which represent the genuine and spoofed speech. The scores given by these two models are used to do spoofing detection.
- i-vector with PLDA system: As described in Section 2.2, the i-vector system shows promising results for speaker verification. In this paper we use i-vector for spoofing detection. The i-vector model is trained on training set, and each algorithm plus genuine speech is considered as a potential class. There are six classes in the training set. 400-dimension i-vectors are extracted and then standard PLDA is applied (Kenny et al., 2013; Matejka et al., 2011; Prince and Elder, 2007) to calculate the similarities between the genuine speech class and each test utterance. The latent subspace dimension of PLDA is 256 in our experiments.
- Linear discriminant analysis (LDA) system: In addition to the basic methods described in Section 2, an LDA based baseline is also built. The mean of the 48-dim FBANK features (static filter banks with delta feature) over each utterance is obtained, and then the standard variance of each dimension is calculated. This 96-dim vector, which includes 48-dim mean and 48-dim standard variance, is used as the identity vector for each utterance. The linear discriminant analysis algorithm is used to classify between genuine and spoofed speech.

According to the ASVspoof2015 Challenge evaluation plan (Wu et al., 2015b), the Equal Error Rate (EER) was first determined independently for each spoofing algorithm, and then the averaged EER for all evaluation dataset was used for the final ranking results. In this work, the same metric (averaged EER) is also utilized for most of the experimental results (except the further analysis in Section 5.7). The EERs of the baseline systems using traditional spectral features are illustrated in Tables 2 and 3. It can be seen that the LDA system performed the best on all types of spoofing attacks. The traditional i-vector system is not as good as the other two in the spoofing detection task, so we need to develop more advanced methods for spoofing detection.

**Table 4**
CV accuracy on spoofing classification of different neural networks. The number in the brackets for the multi-task joint-learn DNN is the accuracy for the second speaker classification task.

| Network | Accuracy |
|---|---|
| Stacked Autoencoder | – |
| Spoofing-discriminant DNN | 84.9% |
| Multi-task joint-learn DNN | 85.4% (84.2%) |
| LSTM | 97.0% |
| BLSTM | 97.2% |

### 5.3. Neural network architecture configuration

To evaluate the proposed types of deep feature extractors described in Section 3, different neural networks are trained, including the Feed Forward DNNs and Recurrent Neural Networks.

**Feed Forward DNNs**: Four hidden layers with 1024 nodes in each layer are used in the DNN-based deep feature extractions, and a context window of 31 frames 48-dim FBANK_D features is concatenated to be used as the DNN input. The output layer depends on the specific targets for different types: no special labels are needed for the stacked autoencoder network since it is designed to re-construct the inputs. The five known spoofing attack labels plus one human speech label are used for the spoofing-discriminative DNN, and both spoofing attack labels and speaker labels are utilized for the multi-task joint-learned DNN. The normal SGD based back-propagation is applied to train all the models.

**Recurrent NNs**: Both the LSTM–RNN and BLSTM–RNN have 2 hidden layers which are followed by the soft-max layer, and each LSTM (BLSTM) layer has 1024 memory cells. For the LSTM network, only one LSTM component is used, and for the BLSTM network, the recurrent layers contain two LSTM components, with one forward and the other backward. Input to the LSTM–RNN (BLSTM–RNN) is a single acoustic frame, and the truncated version of Back-Propagation Through Time (BPTT) was used for training LSTM–RNN (BLSTM–RNN).

All these neural networks, including DNNs and RNNs are trained using Keras (Chollet, 2015). After the NN training, the accuracy of the related spoofing classifier outputs on the cross validation (CV) is listed in Table 4.

As illustrated in Table 4, the multi-task joint-learn DNN achieved slightly better accuracy than the one-task based spoofing-discriminant DNN, and recurrent neural networks obtained much more better classification performance compared to the feed-forward DNNs[1].

After the deep models training, the utterance-level spoofing identity vectors are obtained within the related framework for each utterance, as described in the Section 3.

### 5.4. Evaluation of DNN based deep features

#### 5.4.1. Evaluation of different deep features

As described in Section 3.1, performance given by different DNN-based deep features are compared, with results shown in Fig. 8. The deep features from the last hidden layer are used as features and LDA is used as the back-end classifier.

It can be observed that SAE network do not improve the performance on both known and unknown cases when compared to the LDA baseline in the spoofing detection task. Both the supervised trained DNNs, including the spoofing discriminant DNN and multi-task joint-learn DNN, achieved improved performance, thus demonstrating the effectiveness of the DNN-based deep features. One possible explanation on the worse performance of SAE may be that the SAE is an unsupervised model, and no discriminative criterion is used in model training. In contrast, the other two DNNs are discriminative models and use cross-entropy for the model optimization. Considering that the spoofing detection is a type of classification tasks, the discriminative model based approach may be more appropriate.

Doing comparison within the DNNs shows that the multi-task joint-learned DNN has a better frame-level accuracy, but does not contribute to the final spoofing detection. Although the multi-task learning framework has been shown to be effective for speech recognition (Seltzer and Droppo, 2013) and speaker recognition (Chen et al., 2015b), more appropriate designs need to be further developed for spoofing detection. Accordingly it is not evaluated in the following experiments, and more exploration on multi-task joint-learning and its implementation within RNN structure will be investigated in future work.

**Table 5**
Performance EER (%) of deep features from different hidden layers of spoofing-discriminant DNN.

| Hidden layer | 1st | 2nd | 3rd | 4th |
|---|---|---|---|---|
| Known | 0.2 | **0.1** | 0.1 | 0.1 |
| Unknown | 7.8 | **5.1** | 6.5 | 6.5 |
| All | 4.0 | **2.6** | 3.3 | 3.3 |

**Table 6**
Performance EER (%) comparison of different back-end classifiers using the same DNN-based deep feature.

| Classifier | All | Known | Unknown |
|---|---|---|---|
| Baseline | 6.0 | 2.3 | 9.8 |
| LDA | **2.6** | **0.1** | **5.1** |
| GDF | 22.7 | 16.6 | 28.7 |
| Binary SVM | 3.9 | 0.2 | 7.6 |
| One-class SVM | 5.9 | 1.1 | 10.7 |

#### 5.4.2. Evaluation of deep features from different layers

Secondly the deep features from the different positions of DNNs are investigated. According to the results shown in Fig. 8, the best spoofing-discriminant DNN is utilized, and LDA is used as the back-end classifier. The same procedure was applied on the four hidden layer positions, with results shown in Table 5. It is noted that a smaller index indicates a layer closer to the input layer, and in contrast the larger one is the layer closed to the output layer.

It can be observed that the hidden layer position is important for the deep feature generation in spoofing detection. As one can see, the performance varies among different hidden layers. It appears that the top and bottom hidden layers may not be as appropriate for the deep feature generation, as the middle layers.

#### 5.4.3. Evaluation of different classifiers

Different back-end classifiers are compared. Based on the results in Table 5, the deep features from the 2nd hidden layer of the spoofing-discriminant DNN are used. Four classifiers, as described in Section 4, are applied to the deep features, and the comparison and results are shown in Table 6.

Doing the comparison within these back-end classifiers using new proposed deep features, LDA achieves the best performance, and it is much better than the others on all conditions. It seems that transforming vectors into a discriminative space by LDA is more effective in this task, while SVM may need more appropriate kernel methods. Applying LDA as the back-end classifier is thus the best spoofing detection system using DNN-based deep features.

[1] It is noted that there is no accuracy value for stacked autoencoder since the SAE is designed to re-construct the input features.

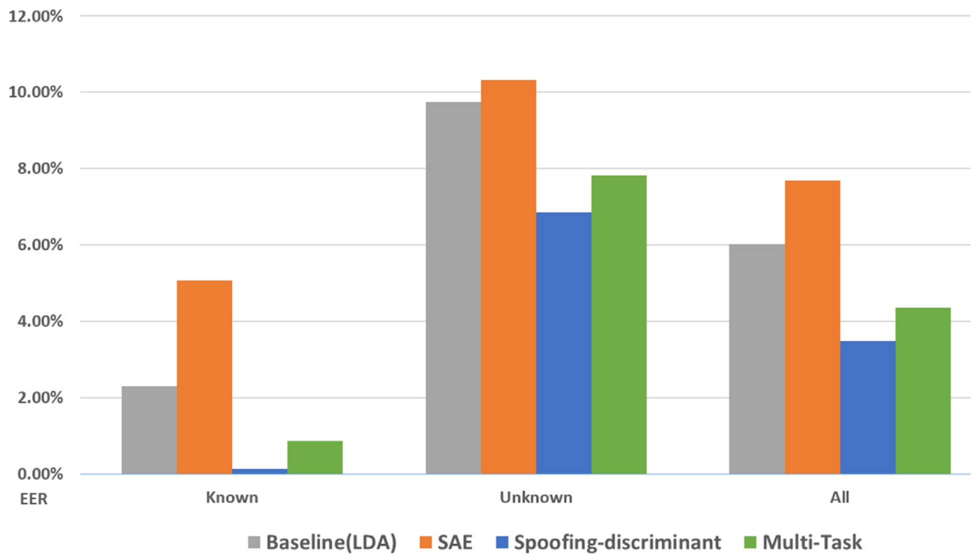

**Fig. 8.** Performance EER comparison of DNN-based deep features from the last hidden layer, with LDA back-end classifier.

**Table 7**
Performance EER (%) comparison of two types RNN-based deep features on five known conditions.

| Network types | S1 | S2 | S3 | S4 | S5 | Known |
|---|---|---|---|---|---|---|
| LSTM | 0.0 | 0.7 | 0.0 | 0.0 | 0.4 | 0.2 |
| BLSTM | **0.0** | **0.2** | **0.0** | **0.0** | **0.1** | **0.1** |

**Table 8**
Performance EER (%) comparison of two types RNN-based deep features on five unknown conditions.

| Network types | S6 | S7 | S8 | S9 | S10 | Unknown |
|---|---|---|---|---|---|---|
| LSTM | 0.7 | 0.4 | 0.1 | 0.3 | **14.1** | 3.1 |
| BLSTM | **0.3** | **0.2** | **0.0** | **0.2** | 15.3 | **3.1** |

**Table 9**
Performance EER (%) comparison of different back-end classifiers using the same RNN-based deep feature.

| Classifier | All | Known | Unknown |
|---|---|---|---|
| LDA | 1.6% | **0.1%** | 3.2% |
| GDF | 2.2% | 0.2% | 4.3% |
| Binary SVM | **1.4**% | 0.2% | **2.5**% |
| One-class SVM | 1.5% | 0.2% | 2.6% |

Compared to the baseline using traditional spectral features, the new proposed deep features are more robust and can improve the detection accuracy significantly.

### 5.5. Evaluation of RNN based deep features

Following the methods described in Section 3.2, the RNN-based deep features are investigated. The deep features are extracted from the last hidden layer, and LDA is implemented on these utterance representations. It is noted that for the RNN-based deep features, the implemented RNN models only have 2 hidden layers. Our initial experiments showed that there was similar performance within these two layers, so only the last hidden layer is used in the experiments for the RNN-based deep features.

The performance comparison of deep features from LSTM–RNN or BLSTM–RNN is illustrated in Tables 7 and 8 on all conditions. Results show that both the uni- and bi-directional LSTM based deep features achieve very good performance. Compared to the DNN-based deep features, the RNN-based deep feature systems have much better results on the unknown attack conditions (most of the contribution is from the S10 detection), showing the superiority of RNN-based deep features. Doing the comparison within two RNN models, the BLSTM-based deep feature, which takes the both advantages from forward and backward directions processing, obtains better performance on most of the spoofing attack types and is slightly better than the LSTM-based deep feature.

Using the BLSTM–RNN based deep features, different classifiers are applied with results shown in Table 9. Results show that the RNN-based deep features are relatively more robust and stable when applied with different back-end classifiers. Compared to the results shown in the Table 8 on the DNN-based deep feature, which fluctuates dramatically using various classifiers, the differences are relatively small from one to another with the RNN-based deep feature, and this also demonstrates the robustness of the RNN-based deep features for spoofing detection. With the BLSTM–RNN based deep features, the binary SVM back-end has the best results, with EER of 1.4%.

### 5.6. System combination on two types deep features

Based on the results and comparison presented above, all deep features have obtained an improved performance compared to the traditional spectral features or traditional i-vector approach. We observed that the deep features from the DNN or the RNN framework have differential properties, therefore these two types of deep features are combined to get a better detection system.

To accomplish this, the spoofing-discriminant DNN based system and BLSTM–RNN based system are combined. First the two scores are obtained from the individual best system: one from the DNN-based deep features with LDA classifier and the other from the BLSTM-based deep features with binary SVM classifier. Using the pre-computed mean and standard variance estimated on the training set, these two scores are normalized to zero mean and unit variance individually. Finally the equally weighted average score on these two normalized scores is calculated for the detection decision. The results of the combination system are shown in Tables 10 and 11.

**Table 10**
Performance EER (%) of the deep features combination on five known conditions. Baseline refers the LDA baseline shown as Table 2.

| Systems | S1 | S2 | S3 | S4 | S5 | Known |
|---|---|---|---|---|---|---|
| Baseline | 2.6 | 2.2 | 1.0 | 1.0 | 4.6 | 2.3 |
| Best DNN | 0.0 | 0.1 | 0.0 | 0.0 | 0.2 | 0.1 |
| Best RNN | 0.0 | 0.9 | 0.0 | 0.0 | 0.3 | 0.2 |
| Combination | **0.0** | **0.0** | **0.0** | **0.0** | **0.0** | **0.0** |

**Table 11**
Performance EER (%) of the deep features combination on five unknown conditions and all condition. Baseline refers the LDA baseline shown as Table 3.

| Systems | S6 | S7 | S8 | S9 | S10 | Unknown | All |
|---|---|---|---|---|---|---|---|
| Baseline | 4.2 | 3.3 | 0.4 | 2.3 | 38.6 | 9.8 | 6.0 |
| Best DNN | 0.2 | 0.0 | 0.0 | 0.0 | 25.5 | 5.1 | 2.6 |
| Best RNN | 0.8 | 0.5 | 0.0 | 0.7 | 10.7 | 2.5 | 1.4 |
| Combination | **0.1** | **0.0** | **0.0** | **0.0** | **10.7** | **2.2** | **1.1** |

**Table 12**
Performance EER (%) comparison of some spoofing detection systems using two different EER calculation methods (Averaged EER vs. Pooled EER).

| Systems | All (Averaged) | All (Pooled) |
|---|---|---|
| Baseline | 6.0 | 7.4 |
| Best DNN | 2.6 | 6.1 |
| Best RNN | 1.4 | 2.2 |

The results show that both of the single deep feature based system gets a large improvement compared to the baseline. The combination of these two kinds deep features can take advantages from both the frame-level DNN and the sequence-level RNN, and the score-fusion system achieves the best results on all ten spoofing attack types. Moreover, compared to the recent works which utilized some handcrafted delicate features for spoofing detection, such as phase feature (Alam et al., 2015; Liu et al., 2015b, 2015b; Novoselov et al., 2015; Sanchez et al., 2015; Villalba et al., 2015; Wang et al., 2015; Weng et al., 2015; Xiao et al., 2015), cochlea feature (Patel and Patil, 2015) and prosody feature (Weng et al., 2015), the proposed deep feature based system can obtain almost 0.0% EER on all S1∼ S9, and get 1.1% on all averaged conditions. This performance is much better than our first attempt using deep features for spoofing detection in ASVspoof2015 Chalenge (which was the third place system (Chen et al., 2015a)), and shows a very promising performance using deep features for the ASVspoof2015 Challenge.

### *5.7. Other analysis and discussion*

As stated previously, the averaged EER was used in the formal evaluation plan for the ASVspoof2015 Challenge, i.e. different thresholds are chosen for each spoofing type individually, and the averaged EER is then calculated for all conditions. In contrast another EER calculation approach is the pooled EER, i.e. pooling the scores of all spoofing types and computing the EER on all detection types, and this is more realistic without using the prior knowledge of the spoofing type.

We have also run the experiments using the pooled EER, with results illustrated in Table 12. It is observed that the new proposed deep features approaches are much better than the baseline on both EER calculation methods. The pooled EER is worse than the averaged EER in all detection systems. The pooled EER, which uses one threshold for all detection types, is more realistic for real applications. The worse EER also indicates that building a robust spoofing detection system in the real scenario is still a difficult task, still needing future work.

## 6. Conclusion and future work

This paper presents detailed work on using various types of deep features for automatic spoofing detection. Two deep feature engineering frameworks are proposed, including DNN-based frame-level feature extraction and RNN-based sequence-level feature extraction. Within the DNN-based features, three model structures are developed, including stacked autoencoders, spoofing-discriminant deep neural networks and multi-task joint-learned deep neural networks. The spoofing-discriminant DNN is more useful for the spoofing detection task. In the RNN system, uni- and bi-directional LSTM–RNN are implemented. These RNN-based deep features show better stability and robustness when combined with different back-end classifiers in spoofing detection, and get improved system performance compared to the DNN-based deep features. In addition, the DNN-based and RNN-based deep features are finally combined to achieve a better system performance. The final proposed deep feature based automatic spoofing detection system almost obtains nearly 0.0% EER on all S1∼ S9 attacks, with 1.1% on all averaged conditions on ASVspoof2015 Challenge.

Based on these experimental results, we can see that using deep models based feature extraction is very promising in spoofing detection. In the future, other ways to extract deep features or other deep structures will be developed. In addition, among these ten spoofing attacks, S10 is relatively more difficult to detect than others because unit selected based synthesized speech sounds more natural than other spoofed speech. To address this, special features, such as cochlea based features (Patel and Patil, 2015) and phase based features (Alam et al., 2015), will be added into the deep learning framework to achieve a more robust system.

## Acknowledgment

The authors would like to thank for the help from Prof. M. T. Johnson in University of Kentucky on the writing improvement. 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 Interdisciplinary Program (14JCZ03) of Shanghai Jiao Tong University in China.

## References

Alam, M.J., Kenny, P., Bhattacharya, G., Stafylakis, T., 2015. Development of crim system for the automatic speaker verification spoofing and countermeasures challenge 2015. In: Proceedings of InterSpeech.

Alegre, F., Vipperla, R., Evans, N., 2012. Spoofing countermeasures for the protection of automatic speaker recognition systems against attacks with artificial signals. In: Proceedings of InterSpeech.

Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H., et al., 2007. Greedy layer-wise training of deep networks. Adv Neural Inf Process Syst 19, 153.

Burget, L., Matejka, P., Schwarz, P., Glembek, O., Cernockỳ, J.H., 2007. Analysis of feature extraction and channel compensation in a gmm speaker recognition system. Audio Speech Lang. Process., IEEE Trans. 15 (7), 1979–1986.

Chen, D., Mak, B., Leung, C., Sivadas, S., 2014. Joint acoustic modeling of triphones and trigraphemes by multi-task learning deep neural networks for low-resource speech recognition. In: Proceedings of ICASSP, pp. 5592–5596.

Chen, L.-W., Guo, W., Dai, L.-R., 2010. Speaker verification against synthetic speech. In: Chinese Spoken Language Processing (ISCSLP), 2010 7th International Symposium on. IEEE, pp. 309–312.

Chen, N., Qian, Y., Dinkel, H., Chen, B., Yu, K., 2015. Robust deep feature for spoofing detection - the sjtu system for asvspoof 2015 challenge. In: Proceedings of InterSpeech, pp. 2097–2101.

Chen, N., Qian, Y., Yu, K., 2015. Multi-task learning for text-dependent speaker verification. In: Proceedings of InterSpeech, pp. 185–189.

Chollet, F., 2015. keras. https://github.com/fchollet/keras.

De Leon, P.L., Stewart, B., Yamagishi, J., 2012. Synthetic speech discrimination using pitch pattern statistics derived from image analysis.. In: Proceedings of InterSpeech, pp. 370–373.

Dehak, N., 2009. Discriminative and Generative Approaches for Long-and Short-term Speaker Characteristics Modeling: Application to Speaker Verification. Ecole de Technologie Superieure (Canada).

Dehak, N., Kenny, P., Dehak, R., Dumouchel, P., Ouellet, P., 2011. Front-end factor analysis for speaker verification. Audio Speech Lang. Process. IEEE Trans. 19 (4), 788–798.

Feng, X., Zhang, Y., Glass, J., 2014. Speech feature denoising and dereverberation via deep autoencoders for noisy reverberant speech recognition. In: In proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP). IEEE, pp. 1759–1763.

Gehring, J., Miao, Y., Metze, F., Waibel, A., 2013. Extracting deep bottleneck features using stacked auto-encoders. In: In proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP). IEEE, pp. 3377–3381.

Gers, F.A., Schmidhuber, J., 2001. Lstm recurrent networks learn simple context-free and context-sensitive languages. Neural Networks IEEE Trans. 12 (6), 1333–1340.

Graves, A., Fernández, S., Schmidhuber, J., 2005. Bidirectional Lstm Networks for Improved Phoneme Classification and Recognition. In: Artificial Neural Networks: Formal Models and Their Applications–ICANN 2005. Springer, pp. 799–804.

Graves, A., Mohamed, A.-r., Hinton, G., 2013. Speech recognition with deep recurrent neural networks. In: In Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP). IEEE, pp. 6645–6649.

Graves, A., Schmidhuber, J., 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. Neural Netw. 18 (5), 602–610.

Greenberg, C.S., Stanford, V.M., Martin, A.F., Yadagiri, M., Doddington, G.R., Godfrey, J.J., Hernandez-Cordero, J., 2013. The 2012 nist speaker recognition evaluation.. In: Proceedings of InterSpeech, pp. 1971–1975.

Grézl, F., Karafiát, M., Kontár, S., Cernocky, J., 2007. Probabilistic and bottle-neck features for lvcsr of meetings. In: In proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP), 4. IEEE, pp. IV–757.

Hochreiter, S., Bengio, Y., Frasconi, P., Schmidhuber, J., 2001. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput 9 (8), 1735–1780.

Hubeika, V., Burget, L., Matejka, P., Schwarz, P., 2008. Discriminative training and channel compensation for acoustic language recognition.. In: Proceedings of InterSpeech, pp. 301–304.

Jaitly, N., Vanhoucke, V., Hinton, G., 2014. Autoregressive product of multi-frame predictions can improve the accuracy of hybrid models. In: Proceedings of Interspeech, pp. 1905–1909.

Jin, Q., Waibel, A., 2000. Application of lda to speaker recognition.. In: Proceedings of InterSpeech, pp. 250–253.

Kenny, P., Boulianne, G., Ouellet, P., Dumouchel, P., 2007. Joint factor analysis versus eigenchannels in speaker recognition. Audio, Speech, Lang.e Process., IEEE Trans. on 15 (4), 1435–1447.

Kenny, P., Stafylakis, T., Ouellet, P., Alam, M.J., Dumouchel, P., 2013. Plda for speaker verification with utterances of arbitrary duration. In: In proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP). IEEE, pp. 7649–7653.

Khoury, E., Kinnunen, T., Sizov, A., Wu, Z., Marcel, S., 2014. Introducing i-vectors for joint anti-spoofing and speaker verification. Proeedings of InterSpeech.

Liu, Y., Qian, Y., Chen, N., Fu, T., Zhang, Y., Yu, K., 2015. Deep feature for text-dependent speaker verification. Speech Commun. 73, 1–13.

Liu, Y., Tian, Y., He, L., Liu, J., Johnson, M. T., 2015b. Simultaneous utilization of spectral magnitude and phase information to extract supervectors for speaker verification anti-spoofing.

Matejka, P., Glembek, O., Castaldo, F., Alam, M.J., Plchot, O., Kenny, P., Burget, L., Cernocky, J., 2011. Full-covariance ubm and heavy-tailed plda in i-vector speaker verification. In: In proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP). IEEE, pp. 4828–4831.

McLaren, M., Van Leeuwen, D., 2011. Source-normalised-and-weighted lda for robust speaker recognition using i-vectors. In: In proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP). IEEE, pp. 5456–5459.

Novoselov, S., Kozlov, A., Lavrentyeva, G., Simonchik, K., Shchemelinin, V., 2015. Stc anti-spoofing systems for the asvspoof 2015 challenge. In: Proceedings of InterSpeech.

Ogihara, A., Hitoshi, U., Shiozaki, A., 2005. Discrimination method of synthetic speech using pitch frequency against synthetic speech falsification. IEICE Trans.s on Fundam. lectron.,Communi.Comput.Sci. 88 (1), 280–286.

Patel, T.B., Patil, H.A., 2015. Combining evidences from mel cepstral, cochlear filter cepstral and instantaneous frequency features for detection of natural vs. spoofed speech. In: Proc. InterSpeech.

Prince, S.J., Elder, J.H., 2007. Probabilistic linear discriminant analysis for inferences about identity. In: 2007 IEEE 11th International Conference on Computer Vision. IEEE, pp. 1–8.

Sanchez, J., Saratxaga, I., Hernaez, I., Navas, E., Erro, D., 2015. The aholab rps ssd spoofing challenge 2015 submission. In: Proceedings of InterSpeech.

Satoh, T., Masuko, T., Kobayashi, T., Tokuda, K., 2001. A robust speaker verification system against imposture using an hmm-based speech synthesis system.. In: Proc. InterSpeech, pp. 759–762.

Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C., 2001. Estimating the support of a high-dimensional distribution. Neural Comput. 13 (7), 1443–1471.

Schölkopf, B., Williamson, R.C., Smola, A.J., Shawe-Taylor, J., Platt, J.C., et al., 1999. Support vector method for novelty detection.. In: NIPS, 12. Citeseer, pp. 582–588.

Scholkopft, B., Mullert, K.-R., 1999. Fisher discriminant analysis with kernels. Neural Netw. Signal Process IX 1 (1), 1.

Schuster, M., Paliwal, K.K., 1997. Bidirectional recurrent neural networks. IEEE Trans. Sig. Process. 45 (11), 2673–2681.

Seltzer, M.L., Droppo, J., 2013. Multi-task learning in deep neural networks for improved phoneme recognition. In: In Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP). IEEE, pp. 6965–6969.

Shang, W., Stevenson, M., 2010. Score normalization in playback attack detection. In: In Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP). IEEE, pp. 1678–1681.

Solomonoff, A., Campbell, W.M., Boardman, I., 2005. Advances in channel compensation for svm speaker recognition.. In: In Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP), pp. 629–632.

Srivastava, N., Mansimov, E., Salakhutdinov, R., 2015. Unsupervised learning of video representations using lstms. CoRR, abs/1502.04681 2.

Sundermeyer, M., Schlüter, R., Ney, H., 2012. Lstm neural networks for language modeling.. In: Proceedings of InterSpeech, pp. 194–197.

Sutskever, I., Vinyals, O., Le, Q.V., 2014. Sequence to sequence learning with neural networks. In: Advances in neural information processing systems, pp. 3104–3112.

Tian, X., Wu, Z., Xiao, X., Chng, E.S., Li, H., 2016. Spoofing detection from a feature representation perspective. In: In Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP).

Vair, C., Colibro, D., Castaldo, F., Dalmasso, E., Laface, P., 2006. Channel factors compensation in model and feature domain for speaker recognition. In: Speaker and Language Recognition Workshop, 2006. IEEE Odyssey 2006: The. IEEE, pp. 1–6.

Variani, E., Lei, X., McDermott, E., Lopez Moreno, I., Gonzalez-Dominguez, J., 2014. Deep neural networks for small footprint text-dependent speaker verification. In: In proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP). IEEE, pp. 4052–4056.

Villalba, J., Lleida, E., 2011. Detecting Replay Attacks from Far-field Recordings on Speaker Verification Systems. In: Biometrics and ID Management. Springer, pp. 274–285.

Villalba, J., Lleida, E., 2011. Preventing replay attacks on speaker verification systems. In: Security Technology (ICCST), 2011 IEEE International Carnahan Conference on. IEEE, pp. 1–8.

Villalba, J., Miguel, A., Ortega, A., Lleida, E., 2015. Spoofing detection with dnn and one-class svm for the asvspoof 2015 challenge. In: Proceedings of InterSpeech.

Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.-A., 2008. Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th international conference on Machine learning. ACM, pp. 1096–1103.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.-A., 2010. Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. The J. of Machine Learning Research 11, 3371–3408.

Wang, L., Yoshida, Y., Kawakami, Y., Nakagawa, S., 2015. Relative phase information for detecting human speech and spoofed speech. In: Proceedings of InterSpeech.

Wang, Z.-F., Wei, G., He, Q.-H., 2011. Channel pattern noise based playback attack detection algorithm for speaker recognition. In: Machine Learning and Cybernetics (ICMLC), 2011 International Conference on, 4. IEEE, pp. 1708–1713.

Weng, S., Chen, S., Yu, L., Wu, X., Cai, W., Liu, Z., Li, M., 2015. The sysu system for the interspeech 2015 automatic speaker verification spoofing and countermeasures challenge. In: Proceedings of InterSpeech.

Wöllmer, M., Eyben, F., Schuller, B., Rigoll, G., 2011. A multi-stream asr framework for blstm modeling of conversational speech. In: In Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP). IEEE, pp. 4860–4863.

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

Wu, Z., King, S., 2016. Improving trajectory modelling for dnn-based speech synthesis by using stacked bottleneck features and minimum trajectory error training. IEEE/ACM Trans. Audio Speech Lang. Process 24 (7), 1255–1265.

Wu, Z., Kinnunen, T., Chng, E.S., Li, H., Ambikairajah, E., 2012. A study on spoofing attack in state-of-the-art speaker verification: the telephone speech case. In: Signal &amp; Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific. IEEE, pp. 1–5.

Wu, Z., Kinnunen, T., Evans, N., Yamagishi, J., Hanilçi, C., Sahidullah, M., Sizov, A., 2015b. Asvspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge.

Wu, Z., Siong, C.E., Li, H., 2012. Detecting converted speech and natural speech for anti-spoofing attack in speaker recognition. In: Proceedings of InterSpeech, pp. 1700–1703.

Xiao, X., Tian, X., Du, S., Xu, H., Chng, E.S., Li, H., 2015. Spoofing speech detection using high dimensional magnitude and phase features: The ntu approach for asvspoof 2015 challenge. In: Proceedings of InterSpeech.

Zhang, C., Ranjan, S., Nandwana, M. K., Zhang, Q., Misra, A., Liu, G., Kelly, F., Hansen, J. H., 2016. Joint information from nonlinear and linear features for spoofing detection: an i-vector/dnn based approach.