Deep Feature Engineering for Noise Robust Spoofing Detection

Yanmin Qian, *Member, IEEE*, Nanxin Chen, *Student Member, IEEE*, Heinrich Dinkel, *Student Member, IEEE*, and Zhizheng Wu, *Member, IEEE*

***Abstract*—Spoofing detection for automatic speaker verification (ASV) aims to discriminate between genuine and spoofed speech. This topic has received increased attentions recently due to safety concerns with deploying an ASV system. While the performance of spoofing detection has improved significantly in clean condition in recent studies, the performance degrades dramatically in noisy conditions. To address this issue, in this paper, we propose to extract *robust and discriminative deep features* by using *deep learning* techniques for spoofing detection. In particular, we employ deep feed-forward, recurrent, and convolutional neural networks to extract discriminative features. We also introduce *multicondition training*, *noise-aware training*, and *annealed dropout training* to make neural networks more robust against noise and to avoid overfitting to specific spoofing attacks and noise types. The proposed neural networks and training techniques are combined into a single framework for spoofing detection. Experimental evaluation is carried out on a noisy version of the standard ASVspoof 2015 corpus, including both additive noisy and reverberant scenarios. Experimental results confirm that the proposed system dramatically decreases averaged equal error rates from 19.1% and 22.6% to 3.2% and 5.1% for seen and unseen noisy conditions, respectively.**

***Index Terms*—Deep learning, deep features, noise robust, speaker verification, spoofing detection.**

I. Introduction

AUTOMATIC Speaker Verification (ASV) is the task of automatically accepting or rejecting a claimed identity based on provided speech samples. ASV is a convenient way to accomplish person authentication, for tasks such as unlocking smartphones. ASV technology has advanced significantly with the use of channel compensation techniques e.g., i-vector based approaches [1], [2], and deep learning techniques [3]–[5]. However, one of the major obstacles for successful deployment is the vulnerability of ASV systems to spoofing attacks. As reviewed in [6], impersonation, replay, speech synthesis, and voice conversion are spoofing attack techniques that may compromise the safety of state-of-the-art ASV systems.

A. Related Work

In order to protect ASV systems from spoofing attacks, spoofing countermeasures, which will reject an input speech signal if it is believed to be a spoofing attack, can be integrated into ASV systems. Spoofing detection can be treated as a binary classification problem: from a speech utterance u, the task is to decide whether u belongs to the genuine speech class hypothesis H_{gen}, or to the spoofed speech class hypothesis H_{spoof}. The decision is based upon the likelihood ratio score ω:

$$\omega(u) = \frac{P(u|H_{\text{gen}})}{P(u|H_{\text{spoof}})} \tag{1}$$

Spoofing detection performance can be improved using both feature and model approaches. There are a significant number of studies on novel features for spoofing detection. For instance, spectral ratio and modulation indexes [7], [8] and channel noise features [9] were have been demonstrated to be effective to detect replay attacks. Phase spectra such as cosine-normalized phase and modified group delay phase features [10], [11], F0 statistics [12], [13], higher order Mel-cepstral coefficients [14], and intra-frame differences [15] have been reported to protect ASV systems from speech synthesis and voice conversion attacks. Deep neural networks (DNN) based features also show promising performance [16], [17]. During the ASVspoof 2015 challenge [18], the best system utilized features called CFCCIF which combines cochlear filter cepstral coefficients (CFCC) and variation of instantaneous frequency (IF) [19]. More recently, constant-Q cepstral coefficient (CQCC) based features [20] have shown promising performance.

There are also studies on back-end classifiers for spoofing detection. Examples include Gaussian mixture models [19], [20], support vector machine (SVM) [21], [22], deep neural networks (DNN) [23], and linear discriminant analysis (LDA) [16], [17]. In a GMM based classifier, two separate GMMs are trained, where each models the feature distribution of one class in (1) individually. The log-likelihood ratio between the genuine and spoofing classes can be used as scores for the detection problem.

Manuscript received April 24, 2017; revised July 4, 2017; accepted July 21, 2017. Date of publication July 26, 2017; date of current version August 23, 2017. This work was supported in part by the Shanghai Sailing Program 16YF1405300, in part by the China NSFC Projects 61573241 and 61603252, and in part by the Interdisciplinary Program 14JCZ03 of Shanghai Jiao Tong University in China. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Kong Aik Lee. *(Corresponding author: Yanmin Qian.)*

Y. Qian and H. Dinkel are with the Computer Science and Engineering Department, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: yanminqian@sjtu.edu.cn; richman@sjtu.edu.cn).

N. Chen is with the Center for Language and Speech Processing, Johns Hopkins University, Baltimore, MD 21218 USA (e-mail: bobchennan@gmail.com).

Z. Wu was with the Centre for Speech Technology Research, University of Edinburgh, Edinburgh EH8 9YL, U.K. He is now with Apple Inc., Cupertino, CA 95014 USA (e-mail: wuzhizheng@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASLP.2017.2732162

B. Contribution of This Work

Noise robustness is always a major concern in almost all speech-related applications [24]–[27], and the ASV spoofing detection problem is no exception. Although previous studies have been effective in detecting spoofed speech in clean condition, e.g., on the ASVspoof 2015 database [18], their performance usually degrades significantly on noisy and reverberant speech data [28], [29]. In practical scenarios, it is inevitable that converted or synthesized speech is corrupted by additive noises, channel distortion, or reverberation. Recent work in [28], [29] confirmed that such noises can considerably decrease the performance of spoofing countermeasures. When a spoofing countermeasure is implemented as a stand-alone module and then followed with an ASV system, it is very interesting and important to improve the robustness of spoofing detection module against these distortions. In contrast, there is little research on noisy spoofing detection. Previous work has primarily investigated system performance by applying clean systems to the noisy scenarios directly [28], [29], and large degradations are observed.

In this paper, we focus on spoofing detection under noisy conditions and propose novel techniques for noise robust spoofing detection. We want to reveal several aspects by conducting this work, including: 1) Doing the comprehensive investigation on the impact from the noisy conditions for spoofing detection task, and different types of noisy scenarios are shown and compared; 2) Developing the related techniques which could improve the noise-robust performance and even address this problem; 3) Based on our previous work using deep feature, which got promising results on clean data [16], [17], we further extend and explore this technique to see how we can also make it robust and effective in the noisy environments.

The major contributions are summarized as follows:

1) *Deep, Recurrent and Convolutional neural networks for deep features:* There is a mounting evidence from various machine learning [30] and speech [17], [31]–[34] tasks that DNNs can potentially learn more discriminative and/or representative features from raw data, recurrent neural networks (RNNs) are expected to learn discriminative features to capture the temporal artifacts in the spoofed speech, while convolutional neural networks (CNNs) are designed to learn local discriminative features from speech signals. Moreover deep features learned from DNNs, RNNs and CNNs are complementary and can be combined for the better performance.
2) *Multi-condition and noise-aware training:* To make the spoofing detector noise robust, we introduce multi-condition training [35] and noise-aware training [36]. In the multi-condition training, the neural networks have access to distorted versions of the training data, which could reflect possible distortions at detection stage. In the noise-aware training, we augment each input observation to the neural network with an estimated noise code that presents in the signal. This is performed at both training and detection stages. In this way, the neural networks are aware of noise conditions in the signal and reduce the mismatch between training and detection stages.
3) *Annealed dropout training:* Annealed dropout is a kind of regularization, and can be treated as model averaging to avoid the over-fitting problem, thus improve the generalization capacity for the unseen scenarios. As seen in the ASVspoof 2015 challenge, over-fitting to known attacks may get poor performance for unknown attacks. We expect the annealed dropout training to avoid this issue, and enhance the generalization on both unknown attacks and unseen noises.

To the best of our knowledge, this is the first time that recurrent and convolutional neural networks have been employed for noise-robust spoofing detection. The proposed multi-condition training, noise-aware training and annealed dropout training techniques are applied to neural networks at different levels, and combined to boost performance. Systematic experiments and analysis for the noisy environments are performed.

II. Deep Learning for Discriminative Feature Extraction

Based on our previous preliminary attempt using deep models for spoofing detection [16], the related framework is further developed for spoofing detection, and in particular for the more challenging detection in noisy scenarios. A detailed description will be given in this and the next sections.

As stated previously, the traditional features, e.g., spectral features and phase features, are still not so satisfactory in the clean condition, and tend to perform poorly when encountering noises. Some special designed features may have advantages on certain types of attacks but may not work well on another type. Accordingly, we want to use a data-driven based method to extract the feature representations, which can learn the key knowledge from the data directly and gain embedded robustness within the features.

Deep models are good choices for this task. Their nonlinear modeling ability makes the deep model not only a powerful back-end classifier [28], [37] but also advantageous in feature engineering [4], [38], [39]. Utilizing its feature engineering ability, similar to the work in speech recognition [40] and speaker verification [4], [5], deep models are used to extract a feature representation. Traditional spectral features, such as FBANK (Filter Bank), MFCC (Mel Frequency Cepstral Coefficient) or PLP (Perceptual Linear Prediction), are fed into the deep models, and the outputs derived from one specific hidden layer can be obtained. These newly derived features from the deep models are named *deep features*. Deep-model based feature engineering has shown good performance in some tasks under different environments with distortion [41], [42]. Our hypothesis is that these deep features are also more robust and effective than the conventional spectral-based features due to the ability of deep models for spoofing detection.

Spoofing detection can be regarded as a sequence labeling problem which predicts the class based on the whole input utterance. Considering that different deep models could be used, the extraction process of the utterance-level identity representation can be grouped into Feed-forward NN based frame-level feature extraction and Recurrent NN based sequence-level feature extraction. In the feed-forward NNs, such as DNN/CNN, a

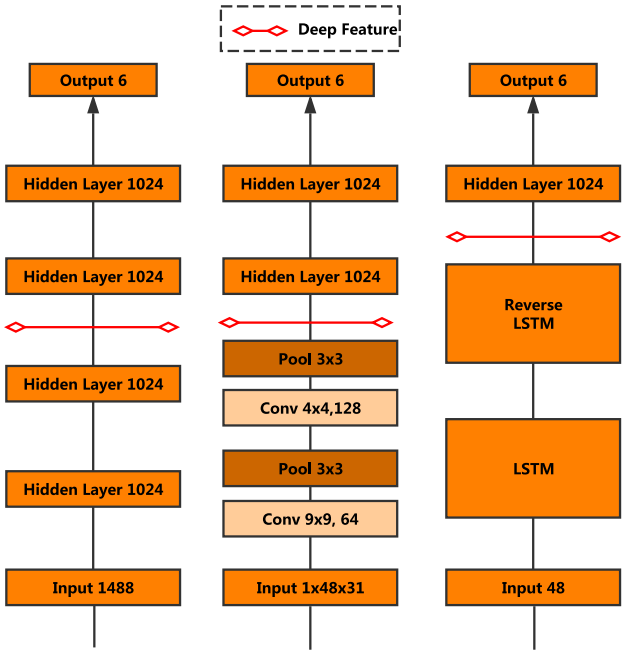

Fig. 1. Different deep models for deep feature extraction. Left: DNN; Middle: CNN; Right: BLSTM-RNN.

context window is taken as the input and the network predicts the probability of each class on each frame. To obtain the utterance-level identity representation, the outputs from the hidden layers are averaged ranging over the entire utterance duration to get the final identity representative vector. In contrast, a recurrent NN treats all frames within a utterance dependently and processes the utterance in a sequential mode. The whole utterance can be taken into the consideration as the input, and the single-frame representation at the last time step is obtained when finishing to process the whole sequence.

In this work, three types of deep models, including DNN, CNN and BLSTM-RNN (Bidirectional Long Short Term Memory RNN), are developed and compared for the deep feature extraction in noisy spoofing detection. All these deep models are spoofing-discriminant, meaning they are trained to discriminate the known spoofing types in the corpus. The ability of the spoofing-discrimination will be enhanced and retained in these models. For example in the ASVspoof2015 corpus [18], there are five known spoofing algorithms in the training data, thus the output layer of the deep models is designed with six classes, including the five known spoofing attacks plus the genuine (human) speech class.

A. Deep Feedforward Neural Network (DNN)

As shown in the left part of Fig. 1, the model takes a context window as the input (31×48 in our model, where 31 is the context window length and 48 is the feature dimension for each frame), consisting of several fully connected hidden layers. After the model optimization, the outputs of the hidden layers can be used as deep features. A similar approach is used in speech recognition [40] and speaker recognition [5]. DNN-based deep features were first adopted in our previous preliminary work for spoofing detection [16]. Based on this previous work, deep features from the middle layer performs better than others, so the second layer is used for DNN-based deep features in this work. It is noted that we directly used the middle layer based on the conclusion obtained on the clean condition, but it may not be the optimum for the noisy condition. The previous structure is utilized as a starting point for this research, and we will leave the architecture investigation in the future work.

B. Convolutional Neural Network (CNN)

Several prior investigations have shown that the use of CNNs has shown to yield better performance than standard fully connected DNNs for some speech-based tasks [43]–[45], and more recently some work in speech recognition shows that CNNs are particularly robust to noise [46], [47]. Accordingly, in this work, CNN is first investigated for spoofing detection, in an attempt to replace the basic DNNs for deep feature extraction.

A typical CNN contains two major parts: a convolutional module followed by several fully connected layers. The convolutional module uses two fundamental types of layers: the convolutional layer followed by a pooling layer. A convolutional layer performs convolution operations to generate output values from local regions (often called receptive fields due to the use with images) of feature maps of the previous layer, and all nodes/neurons in one feature map share the same filter. The pooling layers perform down-sampling on the feature maps of the previous layer and generate new feature maps with a reduced resolution. Usually max-pooling (outputting the maximum value of the pooling size region) is used in most CNN related work in speech processing [43], which is also utilized in our work.

As shown in the middle part of Fig. 1, the CNN is built with 2 convolutional layers following 2 fully-connected layers, with detailed configuration for the CNN as indicated in the figure. After the model training, the outputs of the whole CNN block, i.e., the outputs of the 2nd convolutional layer, are used to extract deep features.

C. Recurrent Neural Network (RNN)

The DNN and CNN are both feed-forward deep models. They are trained with frames independently, and do the averaging on the deep features within the whole utterance to obtain the final utterance-level spoofing identity representation, which assumes the equal contribution from all the frames. However this assumption may not be very accurate since not all frames within an utterance have equal importance and the frames are sequentially dependent on each other.

Accordingly, another advanced deep model, an RNN, is applied, which can take frame dependency into consideration. RNNs have several advantages compared to feed-forward NNs, such as the ability on sequential modeling and dependence modeling, the ability to memorize preceding information internally. Different from the DNN and CNN based deep feature extraction, when using RNN for deep feature extraction, the outputs of the hidden layer at the last time step are directly used as the final utterance-level identity representation. In this work,

long-short term memory cells are used as the RNN model, which is named LSTM-RNN. Moreover, the bi-directional LSTM-RNN (BLSTM-RNN) is constructed to get more advanced model, which can process the whole utterance in two opposite directions. It has been verified that combining forward and backward processing in the BLSTM-RNN can extract more useful information than the uni-directional RNN, therefore can achieve better results for several tasks [48], [49].

Compared to the normal BLSTM-RNN structure,[1] we used a slightly modified version which is shown as the right part of Fig. 1: 1) the input is only fed to the forward LSTM component, and the backward-direction LSTM is connected to the outputs of the preceding forward LSTM layer; 2) after the BLSTM-RNN block, one fully-connected hidden layer is followed before the final softmax output layer. Regarding deep feature extraction, the outputs of the whole BLSTM-RNN block at the last time step, i.e., the outputs of the 2nd backward LSTM layer, are used as the utterance-level spoofing identity representation.

D. Back-End Classifier

After deep feature extraction, every utterance can be represented as one identity vector, no matter whether it is from the DNN, CNN or RNN. This can be regarded as a spoofing identity representation, the same mode as our previous work [16], [17]. A back-end classifier is then applied on these spoofing identity vectors to do the final detection decision. Note that this NN-based identity vector seems similar to the i-vector [50], actually it performs more task-dependently than i-vector. We need to change the classification targets in deep model training if we change from spoofing to other tasks.

In this paper, we have adopted a Linear Discriminant Analysis (LDA) algorithm, which shows strong performance for a variety of tasks [51], [52]. The core idea of LDA is to define new special axes that minimize the intra-class variance caused by channel effects, and to maximize the variance between classes. Moreover the shared parameters give it good generalization capability even with limited number of training samples. It assumes that each class density can be modeled as a multivariate Gaussian:

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

where $\boldsymbol{\Sigma_k}$ and $\boldsymbol{\mu_k}$ is the covariance and mean for class k, p is the dimension of the identity vectors. LDA model assumes every class shares the same covariance, thus $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}, \forall k$. This aims to maximize between-class variance $\boldsymbol{\Sigma}_b$, which equals to maximize the class separation, and eigenvectors $\mathbf{w}$ of $\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_\mathrm{b}$ maximizes the ratio S of between-class variance to the within-class variance:

$$S = \frac{\mathbf{w}^T \boldsymbol{\Sigma}_b \mathbf{w}}{\mathbf{w}^T \boldsymbol{\Sigma} \mathbf{w}} \quad (3)$$

In our implementation we used 6 classes for training, representing genuine speech and the five known spoofing algorithms S1-S5 in training set. During decision function testing, (2) on the genuine speech class is used directly to measure the confidence of genuine speech.

[1] In normal BLSTM-RNN, the input is fed into both the forward and backward LSTM layers, and the outputs of these two layers are first concatenated before fed into the next layer. Moreover usually no fully-connected layer is included in the model.

III. Enhancement for Noise Robustness

As stated above, developing a noise robust spoofing detection system is rapidly becoming more important for real applications. Based on the basic deep feature framework for spoofing detection described in Section II, several advanced approaches are incorporated into the architecture to enhance the noise robustness of the deep features.

In this section, we denote the observed noisy features as $\mathbf{y}$, the corresponding original unknown clean features as $\mathbf{x}$, and the corrupting noise as $\mathbf{n}$.

A. Multi-Condition Training

Training a deep model on multi-condition data enables the network to learn higher level features that are more invariant to the effects of noise with respect to classification accuracy. In view of feature engineering, the lower layers in the deep models are implicitly seeking discriminative features that are invariant across the present acoustic conditions in the training data. Thus in deep model training with multi-condition data, the input vector $\mathbf{v}_t$ is simply an extended context window of the noisy observations.

$$\mathbf{v}_t = [\mathbf{y}_{t-\tau}, ..., \mathbf{y}_{t-1}, \mathbf{y}_t, \mathbf{y}_{t+1}, ..., \mathbf{y}_{t+\tau}] \quad (4)$$

where $\mathbf{y}_t$ represents the feature vector (FBANK) of the current noisy speech frame t, the context window size is $2 * \tau + 1$.

Although multi-condition training is commonly used in GMMs, the benefits from this training in DNNs are different. In the case of discriminative training, e.g., Cross-entropy based training in DNNs, the DNN can potentially extract some useful information from the noise corrupted features through the layers of nonlinear processing, in contrast to GMMs which ignore this information. Multi-condition training has been previously investigated in speech recognition [36], [53], and here we try to use it for spoofing detection.

B. Noise-Aware Training

Model adaptation is one main approach to enhance noise robustness and overcome the mismatch within training and testing [54]–[56]. Shown as the classic Vector Taylor Series (VTS) adaptation in GMMs framework for robust speech recognition [57], the relationship between the $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{n}$ in the log spectral domain is typically approximated as:

$$\mathbf{y}_t \approx \mathbf{x}_t + \log(1 + exp(\mathbf{n}_t - \mathbf{x}_t)) \quad (5)$$

One of the biggest challenges of noise robustness in speech is dealing with this nonlinear relationship. However, due to the multiple layers of nonlinear processing in DNNs, the deep models may have the capacity to learn this complex relationship from data directly. To enable this, the noise-aware training is implemented in deep models for spoofing detection. Although a similar idea is used for speech recognition [36], to the best of

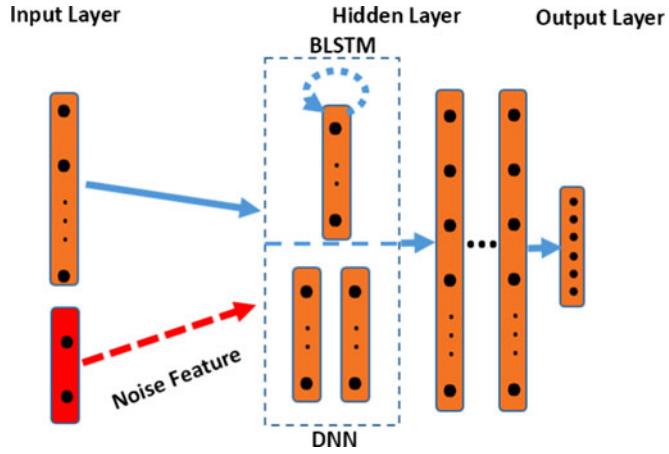

Fig. 2. Noise-aware training for DNN or BLSTM.

the authors' knowledge this is the first time this technology has been applied to spoofing detection.

As described in Section II, the noise information of each utterance is not specifically utilized in the basic deep models described above. To enable noise awareness, the deep model is trained with noisy speech features augmented with an estimate of the noise. In this way, additional online noise information can be used to better optimize the model parameters. Also the estimated noise can be regarded as a special noise code for one kind of adaptation. In the noise-aware training mode, the input vector of the proposed framework will be appended with the noise estimation:

$$\mathbf{v}_t = [\mathbf{y}_{t-\tau}, ..., \mathbf{y}_{t-1}, \mathbf{y}_t, \mathbf{y}_{t+1}, ..., \mathbf{y}_{t+\tau}, \mathbf{n}_t] \quad (6)$$

where $\mathbf{y}_t$ represents the feature vector (FBANK) of the current noisy speech frame t, the context window size is $2 \times \tau + 1$, and $\mathbf{n_t}$ is the appended noise code. The noise code for each utterance was computed by averaging the first T frames and fixed for the entire utterance. It is noted that we assume the first T frames contain only non-speech, e.g., T is 10 or 20, which is always true for the ASVspoof corpus. For actual implementations, we also assume that the spoofed or genuine speech will arrive at the detector after a short non-speech duration, which allows us to estimate the noise representation. In applications where such condition is not met, a more complicated method is required to extract noise codes.

In addition, in contrast to the recent work using noise-aware training with DNNs only [36], we further extended it to the CNN and BLSTM-RNN structures. Shown in Figs. 2 and 3, these can be grouped into two structures for noise-aware training in deep models: 1) In DNN and RNN training, shown in Fig. 2, the noise representation is used as an auxiliary feature and appended with the original spectral feature to form a new input feature vector. This new feature vector is fed into the DNN/RNN, and the following training stages are the same as the standard model training; 2) Considering that topographical features [58], [59], such as FBANK, are more appropriate than non-topographical feature, such as i-vector, for CNN usage, a different noise-aware training structure is designed for CNN. Shown in Fig. 3, rather than concatenating the auxiliary features with the FBANK to

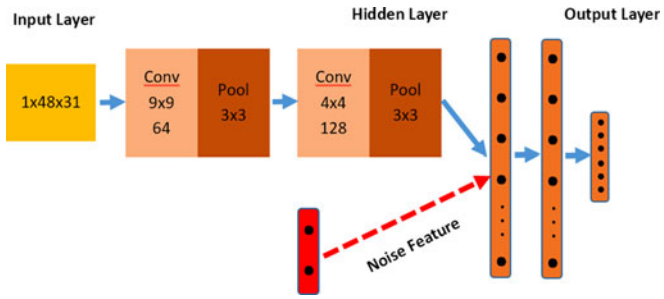

Fig. 3. Noise-aware training for CNN.

be fed into the neural networks as Fig. 2, here the noise representation is concatenated with the outputs of the FBANK-based CNN block, and then this concatenated vector is fed to several shared fully-connected hidden layers. This design combines the advantages from the topographical feature based CNN and the assistance in the noise code.

C. Annealed Dropout Training

One of the biggest problems in noisy robust spoofing detection is to address the mismatches between the training and testing stages, especially caused in the complexity reality by various SNRs and noise types. To better alleviate the mismatch issue, a strategy called "dropout" [60] can also be adopted to further improve the robustness and generalization of deep models. The basic idea of dropout is to randomly omit a certain percentage (e.g., $prob$) of the neurons in each hidden layer during each presentation of the samples during training. This can be treated as model averaging to avoid the over-fitting problem and improve the DNN's generalization capacity especially for the unseen scenarios. Accordingly the dropout training is implemented in all our deep models, including DNN/CNN/BLSTM-RNN, to enhance the robustness of the spoofing detection system.

In addition to the traditional dropout implementation, we further used the annealed version for our spoofing detection, similar to the work investigated in speech recognition [61]. The annealed dropout training is a more powerful regularization approach and can mitigate against the convergence to poor local minima much better [61]. In annealed dropout, the dropout probability of the nodes in the network is decreased as training progresses. In our implementation, the annealed function reduces the dropout rate from an initial rate $prob[0]$ to zero over N steps with constant rate. The dropout probability $prob[t]$ at epoch t is given as:

$$prob[t] = \max(0, 1 - t/N)\, prob[0] \quad (7)$$

Except for this annealed function, the dropout training procedure itself is straightforward and can be performed as normal. After the drouput training, the deep models are used to extract the related deep features as usual.

IV. Experiments

To fully explore the effectiveness of the proposed deep learning framework with enhanced noise robustness for spoofing detection under noisy conditions, experiments, comparison and

TABLE I
THE STATISTICS OF ASVSPOOF2015 CHALLENGE IN THE TRAINING, DEVELOPMENT AND EVALUATION SETS [18]

Subset	#Speakers		#Utterances	
	Male	Female	Genuine	Spoofed
Training	10	15	3750	12625
Development	15	20	3497	49875
Evaluation	20	26	9404	184000

analysis are designed and evaluated on both the original clean ASVspoof 2015 Challenge corpus [18], which is the standard database for this task, and the corresponding noisy version corpus [28].

A. Experimental Setup and Baseline System

1) Noisy Database: The ASVspoof 2015 Challenge dataset [18] is designed to be 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 the genuine speech are shown in Table I. More details about the data corpus can be found in the challenge introduction paper [18].

The spoofed speech in ASVspoof 2015 dataset consists 10 types of spoofed attacks (named as S1-S10 in ASVspoof 2015 challenge) implemented by three speech synthesis and seven voice conversion spoofing algorithms. As described in [18], these spoofing techniques, termed from S1-S10, can be grouped into:

1) *Voice conversion (VC):* S1, S2, S5, S6, S7, S8, S9
2) *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 as **known attacks**, and **S6-S10** are referred 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 [18]. This design enables us to evaluate the effect of the methods on both known and unknown spoofing attacks. More details and protocols about the ASVspoof database can be found in [18].

To evaluate the noise robustness of the proposed approach, a noisy version of the ASVspoof 2015 corpus is utilized in this work. It is an artificially generated noisy corpus, which is initially designed in [28]: the original clean data of ASVspoof 2015 is corrupted by the different noise types with various SNR levels. More details about the data generation can be referred to [28]. We know that there are still obvious differences between the real noisy data and artificial noisy data, but since data collection under real noisy scenarios is challenging, so the artificial noisy corpus is a good choice as a research starting point. Based on that initial study [28], in this work we do further extensions by separating the noises into seen and unseen noise types.

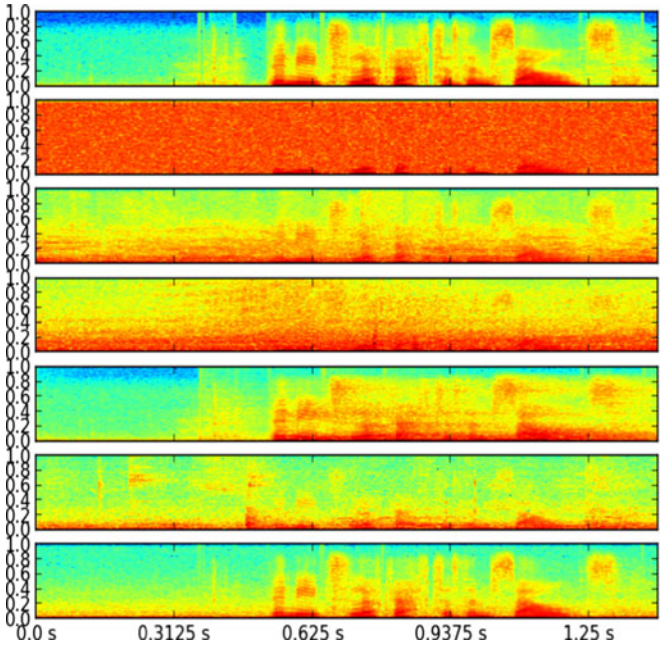

Fig. 4. The spectral comparison of the synchronized clean and noisy data from ASVspoof2015 corpus. From the top to the bottom: *clean*, *white_snr_0*, *babble_snr_0*, *street_snr_0*, *reverberation_T60_0.9*, *cafe_snr_0*, *volvo_snr_0*.

There are a total of 6 different noise types in this work, including five additive types, i.e., white noise, babble noise, volvo (car) noise, street noise, cafe noise, with one convolutional noise, i.e., reverberation. Each additive noise type has three SNR levels, i.e., 20 dB, 10 dB and 0 dB, and the reverberation data is generated with 3 different $T60$ values[2] to simulate the different room environments, i.e., 0.3, 0.6 and 0.9. Thus there are a total of 18 different conditions (noise types & noise strength) in this noisy version of ASVspoof 2015 corpus. The related spectrogram for one same sample waveform (File name E10000034 in the corpus) from the clean data and the corresponding different noise type corrupted data are illustrated in Fig. 4. It is observed that the spectral contamination from all noise types is obvious and serious, which results in a low SNR. This corrupted speech shows the new huge challenge on the noisy spoofing detection task.

In contrast to the experimental design in the preliminary work [28], this noisy corpus is further divided into seen and unseen noise types: the white, babble, street and reverberation are seen noisy scenarios which exist in both training, development and evaluation data, while the cafe and volvo (car) are grouped into the unseen noise types which only exist in the evaluation set. Another consideration is that the white and volvo noises are stationary noises, and babble, street and cafe are non-stationary noises. This division enables us to do the stationary and non-stationary noises analysis in both seen and unseen noisy conditions. Compared to the previous work on noisy spoofing detection [28],

[2] $T60$ reflects the time it takes the energy of an impulse response to decay 60 dB, and can be easily measured from a room impulse response by plotting its energy decay curve. $T60$ is commonly used as a characteristic of reverberation, which is related to the corresponding room [56].

TABLE II
THE FBANK BASELINE (MULTI-CONDITION TRAINING) PERFORMANCE EERS (%) ON ALL ATTACKS AND SCENARIOS

	Known						Unknown						
Condition	S1	S2	S3	S4	S5	S1-S5	S6	S7	S8	S9	S10	S6-S10	Average
clean	1.6	8.8	0.1	0.1	5.1	3.2	6.3	3.2	0.1	2.7	30.7	8.6	5.9
white_snr_20	16.3	28.8	9.4	9.1	21.5	17.0	23.9	14.7	0.4	19.0	37.4	19.1	18.0
white_snr_10	18.8	33.8	20.1	19.7	26.1	23.7	30.7	18.7	8.9	24.1	40.2	24.5	24.1
white_snr_0	24.6	38.2	28.6	27.9	34.7	30.8	38.4	26.1	20.2	32.4	40.9	31.6	31.2
babble_snr_20	14.0	29.9	6.5	6.5	15.7	14.5	17.6	11.4	0.9	14.5	42.6	17.4	16.0
babble_snr_10	19.1	34.0	9.8	9.9	17.7	18.1	20.9	14.7	6.3	17.8	42.9	20.5	19.3
babble_snr_0	30.0	37.9	25.5	25.1	29.2	29.6	31.4	26.8	22.8	26.7	48.0	31.1	30.3
street_snr_20	14.6	30.8	6.0	6.0	15.1	14.5	17.0	11.6	1.9	13.9	45.3	17.9	16.2
street_snr_10	21.6	34.8	9.8	9.8	17.3	18.7	20.1	15.6	6.9	16.3	46.2	21.0	19.8
street_snr_0	30.0	38.5	24.6	24.5	27.9	29.1	31.1	23.8	12.5	26.0	48.1	28.3	28.7
reverberation_0.3	5.0	18.4	1.8	1.9	12.0	7.8	11.9	12.5	2.2	9.7	22.3	11.7	9.8
reverberation_0.6	8.9	24.8	2.7	3.0	15.9	11.1	16.6	19.4	4.0	14.5	27.2	16.3	13.7
reverberation_0.9	9.8	26.7	2.9	3.0	16.7	11.8	17.8	20.4	4.6	15.1	28.9	17.4	14.6
Average EER across seen noisy scenarios	16.5	29.7	11.4	11.3	19.6	17.7	21.8	16.8	7.0	17.9	38.5	20.4	19.1
cafe_snr_20	18.2	33.5	9.6	9.8	18.4	17.9	21.1	14.2	2.9	17.3	47.0	20.5	19.2
cafe_snr_10	24.7	36.4	14.1	14.3	19.9	21.9	23.3	18.7	8.6	19.4	46.8	23.4	22.6
cafe_snr_0	40.5	45.2	39.2	39.6	39.3	40.8	41.7	37.9	25.3	38.3	49.2	38.5	39.6
volvo_snr_20	9.4	23.8	3.4	3.7	11.0	10.2	12.6	9.5	11.3	7.2	49.7	18.1	14.2
volvo_snr_10	14.3	31.3	5.6	5.6	14.9	14.3	16.1	10.9	9.9	11.4	49.9	19.7	17.0
volvo_snr_0	20.6	35.5	13.5	13.2	25.2	21.6	27.8	17.1	7.7	22.9	43.4	23.8	22.7
Average EER across unseen noisy scenarios	21.3	34.3	14.2	14.4	21.4	21.1	23.8	18.1	10.9	19.4	47.7	24.0	22.6

[29], this seen and unseen scenarios design enables us to evaluate the generalization of the proposed approaches and it is very useful for the real applications in noisy environments.

2) Performance Metric: According to the ASVspoof2015 Challenge evaluation plan [18], the Equal Error Rate (EER) was first determined independently for each spoofing algorithm, and then the averaged EER for all evaluated attacks was used. In this work, the same metric (averaged EER) is also utilized, which is implemented using the Bosaris toolkit [62]. As in the previous work [28], different noisy conditions are evaluated individually to obtain the EER for each scenario.

3) Baseline Systems: In this work, two kinds of features are extracted as the front-end for baselines: one is the traditional filter bank (FBANK), and the other is the recently proposed constant Q cepstral coefficients (CQCCs) [20]. For the FBANK system, 24-dim static FBANK with Δ was utilized. For the CQCC baseline, 20-dim feature CQCC was used, i.e., the 19-th order acceleration with C_0, which gets the best performance in the standard ASVspoof 2015 corpus recently [20].

With respect to the back-end classifier, recent work [20], [29] shows that the GMM-based classifier performs well on the spoofing task [20], and even consistently outperforms the more sophisticated i-vector method on both noisy and clean conditions [29] (mainly due to the short utterances in the corpus). Accordingly, the GMM classifier is also used as the back-end in our baseline systems. All seen noisy data in the training set are pooled for training, including the white, babble, street and reverberation noise types. This can be regarded as multi-condition training, and the FBANK or CQCC features are extracted. Expectation Maximization algorithm and Maximal Likelihood is used to estimate the parameters of two GMMs which represents genuine speech and spoofed utterance respectively. In the evaluation, the diversity within the scores given by these two models plays the criterion role to do the spoofing detection, and the decision is made based on (1).

The EERs of the baseline systems using two types of features are ilustrated in Tables II and III respectively for all attacks and scenarios, and the original clean data is also evaluated, as shown in the first line of the tables. It can be observed that although the noisy data has been used in the training, all types of noise still cause a very large degradation compared to the clean data evaluation. The recently proposed CQCC features indeed perform much better than the traditional FBANK features on clean data, especially on S10. However the CQCC features have no advantages over the FBANK when applied in the noisy scenarios, even with larger degradation for most noisy conditions.[3] Both features get poor performance and high EERs. Comparing different noisy scenarios, the influence of reverberation is less severe where most system performs relatively stable under different reverberation time conditions, similar to that reported in [28].

Another observation is that the performance gap between known and unknown attacks is reduced under noisy scenarios compared to that under the clean scenario. In ASVspoof2015 Challenge, all the data is clean, so the spoof pattern can be clear for each attack and there is no obvious mismatch within the patterns between training and testing. The known attacks, the patterns for which have been learned in training, can be detected more easily than the unknown attacks, so the performance gap

[3]Note that it is not very valuable when comparing the lines between Average EERs across seen / unseen noisy scenarios in Table III. Because CQCC feature performs very poorly for most cases in noisy scenarios, and only seems to get small degradation in reverberation and volvo conditions.

TABLE III
THE CQCC BASELINE (MULTI-CONDITION TRAINING) PERFORMANCE EERS (%) ON ALL ATTACKS AND SCENARIOS

	Known						Unknown						
Condition	S1	S2	S3	S4	S5	S1-S5	S6	S7	S8	S9	S10	S6-S10	Average
clean	0.0	0.3	0.0	0.0	0.2	0.1	0.2	0.1	3.5	0.1	0.6	0.9	0.5
white_snr_20	39.4	49.6	47.7	47.2	50.0	46.8	50.0	45.2	41.1	49.5	37.2	44.6	45.7
white_snr_10	46.9	49.9	48.9	48.6	50.0	48.9	50.0	48.4	47.0	49.9	45.1	48.1	48.5
white_snr_0	49.5	49.8	48.7	48.7	49.8	49.3	49.7	49.4	48.2	49.6	47.6	48.9	49.1
babble_snr_20	7.4	24.2	20.6	20.3	18.6	18.2	21.2	13.0	26.0	15.9	15.4	18.3	18.3
babble_snr_10	24.6	41.0	33.5	33.2	37.5	33.9	40.0	30.8	34.4	35.7	27.0	33.6	33.8
babble_snr_0	41.1	48.3	42.7	42.7	48.3	44.6	48.8	43.6	42.4	47.7	37.4	44.0	44.3
street_snr_20	13.3	29.4	22.9	22.7	24.9	22.7	27.7	18.2	25.4	21.4	18.9	22.3	22.5
street_snr_10	30.6	43.1	36.4	35.9	41.3	37.5	43.1	34.3	35.2	38.4	30.7	36.3	36.9
street_snr_0	41.6	48.5	45.8	45.6	49.0	46.1	49.3	44.0	45.0	47.6	41.3	45.4	45.8
reverberation_0.3	1.4	8.7	13.4	13.0	5.6	8.4	5.4	7.1	23.0	6.8	4.3	9.3	8.9
reverberation_0.6	0.7	6.1	21.3	21.1	3.7	10.6	3.5	5.4	22.6	4.6	3.1	7.8	9.2
reverberation_0.9	0.5	4.5	15.0	15.1	2.9	7.6	2.7	4.4	21.9	3.6	2.2	6.9	7.3
Average EER across seen noisy scenarios	24.8	33.6	33.1	32.9	31.8	31.2	32.6	28.6	34.3	30.9	25.8	30.5	30.8
cafe_snr_20	23.8	38.0	28.6	28.6	34.5	30.7	36.7	27.2	29.6	30.5	26.2	30.1	30.4
cafe_snr_10	37.7	46.6	40.4	39.7	46.1	42.1	47.2	40.0	39.3	43.5	36.4	41.3	41.7
cafe_snr_0	44.8	49.4	46.8	46.5	49.9	47.5	49.8	46.1	46.6	49.2	43.9	47.1	47.3
volvo_snr_20	0.1	1.1	1.2	1.2	0.7	0.9	0.8	0.4	10.1	0.6	1.6	2.7	1.8
volvo_snr_10	0.5	4.5	7.0	6.7	2.7	4.3	3.4	1.6	16.8	2.1	3.9	5.6	4.9
volvo_snr_0	3.5	17.1	16.1	16.0	12.3	13.0	14.7	7.9	22.1	9.9	10.6	13.0	13.0
Average EER across unseen noisy scenarios	18.4	26.1	23.3	23.1	24.4	23.1	25.4	20.5	27.4	22.6	20.5	23.3	23.2

is relatively large. In contrast, in the noisy environments, the spectral contamination makes the spoof pattern vague and corrupted. Although the known spoof types have been visited in training, the mismatch and difference of the patterns between training and testing are still large due to the corrupted spectrum. In those cases, the known attacks are also not easily to be detected, thus the performance gap between known and unknown attacks is small.

Although having applied the state-of-art technologies, the spoofing detection accuracy in noisy scenarios is very low on both known and unknown attacks, so the noisy spoofing detection is very challenging. Accordingly we need to develop more advanced approaches to improve the noise robustness for spoofing detection systems. Considering that FBANK seems better than CQCC features in the noisy scenarios in baselines, so we only utilized FBANK feature in our proposed deep learning based approach as described in the following sections.

B. Evaluation of Deep Features in Noisy Scenarios

1) Neural Network Architecture Configuration: To evaluate the proposed deep feature framework, different neural networks are trained following Section II, and then the related enhanced technologies are applied, compared and analyzed following the approaches described in Section III. Considering that the main focus of this work is noise robust problem, so the architecture of deep models are based on our previous work and experience (the structures may not be the optimum in the new noisy scenario, but it is still a good starting point): the structures of DNN and BLSTM followed our previous work [17], and they performed relatively well. For the CNN, the most common structure, which is popularly applied in speech recognition [43], is utilized in this work directly. All the details on these models are listed:

DNNs: Four hidden layers with 1024 sigmoid nodes in each layer are used in all the DNN-based deep feature extractions, and 48-dim FBANK_D features (static with Δ) with a 31 frame context are concatenated as the DNN input.

CNNs: Shown as the middle part of Fig. 1, there are 2 convolutional layers in CNNs, with 64 feature maps in the first layer and 128 feature maps in the second layer. This uses 9×9 filter with 3×3 pooling in the 1st convolutional layer, and 4×4 filter with 3×3 pooling in the 2nd convolutional layer. Following the CNN block, two fully-connected hidden layers with 1024 sigmoid nodes in each layer are added. 48×31 input FBANK features, the same as DNNs, are used for CNNs.

BLSTM-RNNs: The structure is shown as the right part of Fig. 1, and a 48-dim single FBANK frame is used as the input. It has 2 LSTM layers with 1024 memory cells in each, following one fully-connected layer (1024 nodes) on the top. The LSTM layer has two components: one is the forward recurrent component and the other is the reversed backward recurrent component.

The output layer, for all the DNN, CNN and BLSTM-RNN, depends on the specific targets for different attack types, i.e., the five known spoofing attacks plus one human speech class that are used for the deep model training. The normal SGD (Stochastic Gradient Descent) based back-propagation is used to train DNNs and CNNs, and the truncated version of BPTT (Back Propagation Through Time) was used for BLSTM-RNN training. All the networks are trained using Adam [63] and early stopping strategy is adopted which stops the training process when there is no improvement between two iterations.

After the deep model training, the utterance-level spoofing identity vectors are obtained using the deep features from the corresponding models for each utterance, as described in Section II.

TABLE IV
THE EERS (%) COMPARISON OF DNN-BASED DEEP FEATURES, WHICH UTILIZED DIFFERENT TRAINING STRATEGIES, INCLUDING CLEAN-CONDITION TRAINING, MULTI-CONDITION TRAINING AND ANNEALED DROPOUT TRAINING

	Clean-condition Training			Multi-condition Training			+ Annealed dropout Training		
Condition	Known	Unknown	Average	Known	Unknown	Average	Known	Unknown	Average
clean	0.1	5.1	2.6	1.1	4.5	2.8	0.9	3.5	2.2
white_snr_20	39.7	34.7	37.2	2.5	5.3	3.9	1.6	4.5	3.0
white_snr_10	39.7	31.1	35.4	4.2	6.2	5.2	3.4	5.3	4.4
white_snr_0	47.9	47.9	47.9	9.2	11.0	10.1	8.5	11.0	9.7
babble_snr_20	48.0	45.4	46.7	3.9	5.5	4.7	3.4	5.2	4.3
babble_snr_10	43.3	35.8	39.6	7.2	8.6	7.9	6.5	8.0	7.2
babble_snr_0	43.2	34.6	38.9	14.7	17.2	15.9	13.3	15.8	14.6
street_snr_20	46.9	43.7	45.3	4.9	6.5	5.7	4.2	6.2	5.2
street_snr_10	43.8	38.0	40.9	7.5	8.8	8.2	6.7	8.8	7.7
street_snr_0	44.9	38.6	41.7	12.8	14.8	13.8	11.6	14.8	13.2
reverberation_0.3	49.7	49.4	49.5	2.1	3.5	2.8	1.4	3.2	2.3
reverberation_0.6	48.5	47.9	48.2	2.6	4.0	3.3	2.1	3.9	3.0
reverberation_0.9	47.7	46.6	47.2	3.5	4.8	4.2	2.6	4.3	3.4
Average EER across seen noisy scenarios	41.8	38.4	40.1	5.9	7.8	6.8	5.1	7.3	6.2

2) Evaluation on the Clean-Condition Training: First of all, to investigate the impact from the noisy environments, testing using the clean model is performed. In this case, the DNN is trained just using the original clean ASVspoof2015 corpus, and then the related DNN-based deep feature is extracted using this clean model for the spoofing detection. The results, including all noisy types and the original clean data, are shown in the left part of the Table IV. It is observed that the deep feature from the clean-condition training gets a good performance in the matched clean data. However due to the mismatch between training (only clean data) and testing (noisy data), the degradation is very serious, and there are large drops in all noisy conditions. Accordingly, although deep features are generated by deep learning models, which have more advanced ability than the traditional shallow models [16], the noise robustness and mismatch within training and testing are still very challenging problems.

3) Evaluation on the Multi-Condition Training: After this the deep models, used for deep feature extraction, are trained with multi-condition training. The seen noise data are used for training, including white, babble, street and reverberation, and the results are illustrated in the middle of Table IV. Compared to the clean-condition training, this shows that although the EERs are slightly worse on the clean data, the system performance is dramatically improved on all noisy conditions. The EERs in most of the conditions are decreased from $\sim 40.0\%$ to below 10.0%, some of which are even approaching the EERs on the clean data. The deep features learned from the multi-condition training are more invariant to the different effects across various noisy environments, so they obtain more robust performance. Moreover, compared to the baselines results in Tables II and III, which also used multi-condition training, although the multi-condition training is not so helpful in the baselines, in contrast it can bring a much larger gain within the proposed deep feature framework. This also demonstrates the advantage of the deep feature framework for spoofing detection.

4) Evaluation on the Annealed Dropout Training: Based on the multi-condition training, the annealed dropout training described in Section III-C is incorporated. The DNN-based deep feature system is illustrated as the right part of Table IV. Another obvious gain is obtained by the annealed dropout training for spoofing detection on all conditions, including clean and noisy data, and this demonstrates that the proposed annealed dropout training can avoid over-fitting, and make the extracted deep features more robust for the spoofing detection in noisy environments. Comparing the different noisy types, reverberation and white noise are relatively easy to compress with the proposed approach, and the other two non-stationary noises, i.e., babble and street noises, are more difficult and have higher EERs.

5) Evaluation on the Different Deep Features: In addition to the DNN-based deep features, the other deep model based deep features are extracted and compared. Based on the above results from DNN-based deep feature systems, the multi-condition training and annealed dropout training are also applied when training the CNN and BLSTM-RNN (These are implemented in all following experiments unless otherwise noted). After model training, the related deep features are extracted and spoofing detection systems are built as usual. Fig. 5 shows the results and comparison for these three types of deep features, and the average EERs across seen noisy scenarios are illustrated. It is observed that no matter which deep model is used for the deep feature extraction, all kinds of deep features achieve tremendous improvements on all conditions when compared to the baseline. Doing the comparison within these three deep features, CNN-based deep features perform the best on most conditions, which demonstrates the advantage of the CNN in noisy scenarios.

6) Evaluation on the Noise-Aware Training: To make deep features more robust for spoofing detection, noise adaptation using noise-aware training is implemented. According to the description in Section III-B, different deep models will use different appropriate structures to make the models noise-aware.

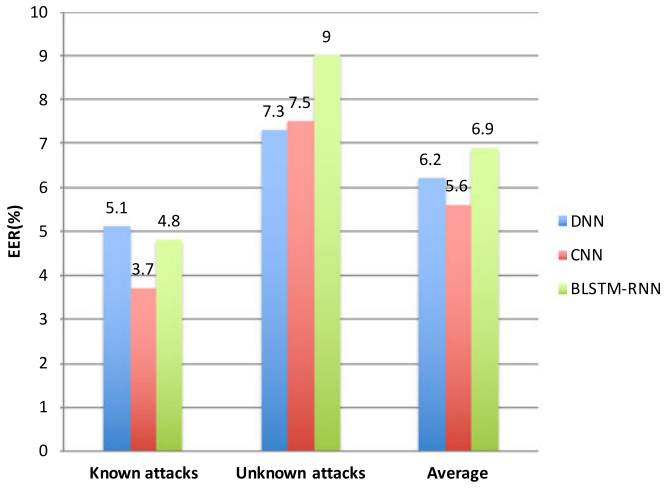

Fig. 5. The comparison of average EERs (%) across seen noisy scenarios for different deep model based deep features, including DNN, CNN and BLSTM-RNN.

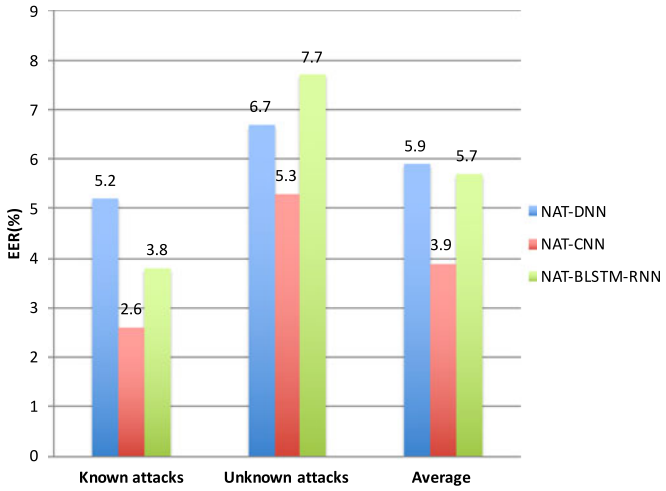

Fig. 6. The comparison of average EERs (%) across seen noisy scenarios for noise-aware training with different deep model based deep features, including NAT-DNN, NAT-CNN and NAT-BLSTM-RNN. NAT indicates **N**oise-**A**ware **T**raining.

The results of the noise-aware training for the three models are also illustrated in Fig. 6. Compared to the results without using adaptation in Fig. 5, noise-aware training achieves another significant improvement for the noisy spoofing detection, and this kind adaptation works well within all deep models. The best single system using NAT-CNN based deep features gets the averaged EER 3.9% across all seen scenarios (from the 19.1% in baseline), and we also find that half of the conditions are even around 1.0%$\sim$ 2.0%.

7) Combination Within Different Deep Features: Based on the results and comparison presented above, all deep features tested show a large improvement. Some deep features perform better on some attacks or environmental types and other deep features may be better on other conditions, so these three types of deep features are combined to get an overall improved system.

First three scores are obtained from each individual best system, i.e., NAT-DNN, NAT-CNN and NAT-BLSTM-RNN. Then using the pre-computed mean and standard variance which are estimated on the training set, these scores are normalized to zero mean and unit variance respectively. Finally the weighted average of these three normalized scores is calculated for the detection decision. The results of the combination system are shown in Table V.

The results show that the combination of these three kinds deep features blends the advantages from DNN, CNN and BLSTM-RNN, with the score-fusion system achieving a 3.2% averaged EER across all seen scenarios, which is better than the best single deep feature based system (3.9% in NAT-CNN in Fig. 6). This conclusion verifies that the different deep model based deep features complement each other, and the appropriate combination approach can obtain a better performance.

C. Evaluation in Unseen Noisy Scenarios

Although it is possible to collect a variety of noise types in training and optimize the model with the multi-condition training, there are still many new unseen noisy scenarios possible in real applications. Accordingly, to validate the effectiveness and generalization of the proposed approach, the evaluation on unseen noisy scenarios is performed, and detailed results and comparison are shown in Tables VI and VII. As described in Section IV-A, the cafe noise and volvo (car) noise are selected as the unseen noise types, and different training strategies are compared in Table VI and different deep features are summarized in Table VII.

Taken together, this shows that almost all the conclusions, observed in the seen noise types, also apply to the unseen noise scenarios: 1) All the training strategies, including multi-condition training, annealed dropout training and noise-aware training get significant gains on all conditions, and these training methods can be further combined to obtain additional improvement; 2) Comparing between DNN/CNN/BLSTM-RNN based deep features, the advantage from CNN is obvious and this approach still performs the best in the unseen noisy scenarios. Combining these three deep features gives all these advantages, and an more improved system is obtained. 3) Compared to the baseline system on the unseen data shown in the bottom part of Table II, the new proposed deep feature framework with advanced training strategies can improve the EERs dramatically. The best system achieves an averaged EER 5.1% across unseen noisy environments, which demonstrates the excellent generalization and robustness of the new proposed approach for noisy spoofing detection.

D. Experimental Summary

For easy comparison, all the experimental results on both seen and unseen noises are summarized in Table VIII, including the FBANK baseline and our proposed approach with respect to each technique. In addition to the known and unknown attacks, the performance on the S10 type is also illustrated separately because it is the most difficult and interesting spoofing type.

Table VIII shows that the largest improvement is from the multi-condition training: if the noise types are seen in training, the deep model can learn the knowledge more effectively

TABLE V
THE EERS (%) COMPARISON OF THE DIFFERENT FEATURES COMBINATION WITHIN THREE DEEP FEATURE TYPES, I.E., NAT-DNN, NAT-CNN AND NAT-BLSTM-RNN

	Known						Unknown						
Condition	S1	S2	S3	S4	S5	S1-S5	S6	S7	S8	S9	S10	S6-S10	Average
clean	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6.5	1.3	0.7
white_snr_20	0.0	1.9	0.0	0.0	0.1	0.4	0.3	0.1	0.0	0.1	13.5	2.8	1.6
white_snr_10	0.0	5.7	0.0	0.0	0.3	1.2	1.4	0.7	0.0	0.2	13.9	3.2	2.2
white_snr_0	0.2	17.0	0.2	0.2	1.6	3.8	6.3	5.5	0.4	2.4	21.4	7.2	5.5
babble_snr_20	0.0	5.4	0.0	0.0	0.1	1.1	0.3	0.1	0.0	0.1	13.1	2.7	1.9
babble_snr_10	0.0	16.5	0.0	0.0	0.4	3.4	1.3	0.7	0.0	0.3	17.7	4.0	3.7
babble_snr_0	1.3	28.7	0.7	0.8	4.7	7.3	8.9	7.2	1.0	3.4	29.3	10.0	8.6
street_snr_20	0.0	9.5	0.0	0.0	0.2	2.0	0.5	0.3	0.1	0.2	15.6	3.3	2.6
street_snr_10	0.1	18.2	0.0	0.0	0.4	3.7	1.5	1.0	0.2	0.4	19.0	4.4	4.1
street_snr_0	0.7	26.7	0.4	0.4	3.5	6.3	7.4	5.2	1.0	2.0	29.6	9.0	7.7
reverberation_0.3	0.0	0.7	0.0	0.0	0.1	0.2	0.2	0.1	0.1	0.1	5.8	1.3	0.7
reverberation_0.6	0.0	1.4	0.0	0.0	0.2	0.3	0.4	0.2	0.1	0.2	5.3	1.2	0.8
reverberation_0.9	0.0	2.2	0.0	0.0	0.3	0.5	0.7	0.4	0.2	0.4	5.2	1.4	0.9
Average EER across seen noisy scenarios	0.2	10.3	0.1	0.1	0.9	2.3	2.3	1.7	0.2	0.7	15.1	4.0	3.2

TABLE VI
UNSEEN NOISES EVALUATION

	Clean-condition Training			Multi-condition Training			+ Annealed dropout Training			+ Noise-aware Training		
Condition	Knw	Unknw	Avg	Knw	Unknw	Avg	Knw	Unknw	Avg	Knw	Unknw	Avg
clean	0.1	5.1	2.6	1.1	4.5	2.8	0.9	3.5	2.2	0.5	1.9	1.2
cafe_snr_20	47.0	43.9	45.5	5.6	7.6	6.6	4.8	7.6	6.2	4.9	6.3	5.6
cafe_snr_10	43.3	36.6	39.9	10.1	12.4	11.2	8.7	11.8	10.2	9.1	11.0	10.1
cafe_snr_0	47.3	43.3	45.3	21.8	25.5	23.6	20.5	24.8	22.6	20.9	24.6	22.7
volvo_snr_20	47.2	47.4	47.3	4.4	6.3	5.3	3.6	5.8	4.7	3.3	5.4	4.4
volvo_snr_10	44.8	44.1	44.4	6.8	8.0	7.4	5.4	6.9	6.1	5.8	8.0	6.9
volvo_snr_0	44.0	40.6	42.3	7.8	9.4	8.6	7.5	11.9	9.7	7.2	10.8	9.0
Average EER across unseen noisy scenarios	39.1	37.3	38.2	8.2	10.5	9.4	7.3	10.3	8.8	7.4	9.7	8.6

The EERs (%) comparison of DNN-based deep features, utilizing different training strategies, including clean-condition training, multi-condition training, annealed dropout training and noise-aware training.

TABLE VII
UNSEEN NOISES EVALUATION

	NAT DNN			NAT CNN			NAT BLSTM-RNN			nat-DNN+nat-CNN+nat-RNN		
Condition	Knw	Unknw	Avg	Knw	Unknw	Avg	Knw	Unknw	Avg	Knw	Unknw	Avg
clean	0.5	1.9	1.2	0.1	2.8	1.4	0.2	4.3	2.2	0.0	1.3	0.7
cafe_snr_20	4.9	6.3	5.6	3.0	5.4	4.2	6.6	10.7	8.6	3.0	4.6	3.8
cafe_snr_10	9.1	11.0	10.1	5.7	8.2	7.0	9.7	15.1	12.4	5.5	7.4	6.4
cafe_snr_0	20.9	24.6	22.7	13.8	20.2	17.0	18.2	24.4	21.3	12.9	18.4	15.6
volvo_snr_20	3.3	5.4	4.4	1.1	3.9	2.5	2.3	8.1	5.2	0.8	2.8	1.8
volvo_snr_10	5.8	8.0	6.9	2.6	5.1	3.8	5.0	10.0	7.5	2.3	4.0	3.2
volvo_snr_0	7.2	10.8	9.0	3.6	5.2	4.4	6.8	10.2	8.5	3.7	4.7	4.2
Average EER across unseen noisy scenarios	7.4	9.7	8.6	4.3	7.3	5.8	7.0	11.8	9.4	4.0	6.2	5.1

The EERs (%) comparison of different deep feature systems and combination systems.

than the shallow model, which demonstrates the powerful ability of deep model once more. In addition to multi-condition training, annealed dropout training and noise-aware training add incremental gains on all conditions and types. Doing combination within different deep features further improves the system significantly.

V. DISCUSSION AND FUTURE WORK

Noise robustness is a very important component of most speech applications. The work presented here suggests that our proposed deep feature engineering framework can significantly improve the performance of spoofing detection system under noisy scenarios.

TABLE VIII
EXPERIMENTAL SUMMARY ON BOTH SEEN AND UNSEEN NOISES EVALUATION

	FBANK baseline system				Clean-condition Training				Multi-condition Training				+ Annealed dropout Training			
Condition	Knw	Unk	S10	Avg	Knw	Unk	S10	Avg	Knw	Unk	S10	Avg	Knw	Unk	S10	Avg
Average EER across Seen noises	17.7	20.4	38.5	19.1	41.8	38.4	47.4	40.1	5.9	7.8	20.6	6.8	5.1	7.3	19.5	6.2
Average EER across Unseen noises	21.1	24.0	47.7	22.6	39.1	37.3	46.5	38.2	8.2	10.5	26.1	9.4	7.3	10.3	26.3	8.8
	+ Noise-aware Training DNN				NAT CNN				NAT BLSTM-RNN				nat-DNN+nat-CNN+nat-RNN			
Condition	Knw	Unk	S10	Avg	Knw	Unk	S10	Avg	Knw	Unk	S10	Avg	Knw	Unk	S10	Avg
Average EER across Seen noises	5.2	6.7	15.4	5.9	2.6	5.3	20.8	3.9	3.8	7.7	24.3	5.7	2.3	4.0	15.1	**3.2**
Average EER across Unseen noises	7.4	9.7	22.8	8.6	4.3	7.3	25.4	5.8	7.0	11.8	30.7	9.4	4.0	6.2	20.5	**5.1**

The EERs (%) comparison among the baseline and the proposed approach with each technique.

However, to fully interpret some of the results and conclusions, there may be two main concerns: 1) Adding noise will make spoofing detection more difficult, but how does this impact the final speaker verification performance? Is it possible that noisy spoofed speech could make speaker verification easier? 2) The noisy speech in reality is more complicated, due to complexities such as the Lombard effect and masking phenomenon, and the difference between real noisy speech and the artificial noisy speech is significant. So it is not certain that the conclusions obtained here will be guaranteed to improve results for the real noisy scenarios.

Specific observations that support the idea that these approaches may be useful include the following:

1) Currently, to protect the speaker verification from the spoofing attacks, many ASV systems are constructed with a spoofing detection module. The most common approach is to equip the ASV system with a stand-alone spoofing detection module at the front-end. In that architecture, the accuracy of the spoofing detection module in any scenario is important for the entire ASV system.
2) When implementing the spoofing detection system as a stand-alone module, there are two purposes for this module: detecting the spoofed speech & accepting the real speech. The previous works have demonstrated that the noisy scenario will make the original difference between the spoofed speech and genuine speech unclear [29], so these two purposes can not be achieved with a high performance. Although the noisy spoofed speech may make the speaker verification easier, we also need to care on the accepted rate of the genuine speech. If the accepted rate of the genuine speech in the noisy spoofing detection dropped dramatically, the performance of the back-end ASV system will still become worse. Therefore, no matter whether "noisy spoofed speech" could make speaker verification easier or not, the demands of the research on noisy spoofing detection system is still necessary and meaningful.
3) There is difference between the real noisy speech and the artificial noisy speech, however it is very time consuming and difficult to collect data in the real scenarios with many noise types. This is the same problem existing in speech recognition community for noisy scenarios, and the ideal real noisy speech under various noise conditions is difficult to be collected. Accordingly a well designed and prepared artificial noisy corpus is a good choice as a research starting point. That is the same reason for the corpus usage in the work [28], [29], which also used the artificial data set for noisy spoofing detection.

This work is a starting point of using deep learning techniques for noisy spoofing detection. There are still many aspects to be explored in the future: 1) Although the spoofing detector is studied as a stand-alone system in this work, the spoofing detection and speaker verification are not independent. It would be valuable to study the joint impact of these two systems together under noisy and reverberant scenarios, and see what will happen when the ASV system processes the noisy spoofed speech. 2) The deep model structures used here are based on our previous work and experience, but they may not be the optimum for the noisy spoofing detection. So the more architecture investigation is still useful to further improve the system performance. 3) Other features also have promising performance on spoofing detection, such as CFCCIF [19] and CQCC [20] features. The exploration of integrating these features into the proposed deep feature framework to further improve the system is interesting and demanded. 4) We need to collect the real noisy data, so that the investigation on real noisy scenario evaluation can be performed. All these aspects will be done in our future work.

VI. CONCLUSION

This paper has proposed a deep learning framework for spoofing detection in reverberant and noisy scenarios. By using the powerful feature engineering capability of deep models, discriminative and robust features are learned from speech data directly for detecting spoofing speech. Three types of deep models are developed, including the DNN, CNN and BLSTM-RNN. The feature generated by these models obtain significant improvement over baseline features, with the CNN generated features performing the best, especially in noisy environments. Moreover, several advanced training strategies, including

multi-condition training, noise-aware training, and annealed dropout training, are integrated into the deep feature engineering framework for robustness against noises and avoiding over-fitting to spoofing attacks and noises in the training data. Experimental results show that these training strategies produce further improvement when combined with deep models. In addition, features generated by the three types of deep models are complementary and can be combined to achieve a more improved performance.

The proposed approach is evaluated on a distorted version of the ASVspoof 2015 corpus, including both additive noisy and reverberant scenarios. Compared with the baseline system, the best performance obtained with the proposed approach decreases the averaged EERs on eval data from 19.1% & 22.6% to 3.2% & 5.1% for seen and unseen distorted conditions, respectively. With the proposed features and training strategies, the performance gap between clean and distorted noisy test data is significantly reduced.

ACKNOWLEDGMENT

The authors would like to thank Mr. Xiaohai Tian from Nanyang Technological University, Singapore for sharing the noisy version of ASVspoof 2015 database, and Prof. Michael T. Johnson in University of Kentucky and Dr. Xiong Xiao in Nanyang Technological University for assistance with writing improvement. Experiments have been carried out on the PI supercomputer at Shanghai Jiao Tong University.

REFERENCES

[1] M. Senoussaoui, P. Kenny, N. Dehak, and P. Dumouchel, "An i-vector extractor suitable for speaker recognition with both microphone and telephone speech," in *Proc. Odyssey*, 2010, pp. 28–33.

[2] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice modeling with sparse training data," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 3, pp. 345–354, May 2005.

[3] N. Chen, Y. Qian, and K. Yu, "Multi-task learning for text-dependent speaker verifciation," in *Proc. InterSpeech*, 2015, pp. 185–189.

[4] Y. Liu, Y. Qian, N. Chen, T. Fu, Y. Zhang, and K. Yu, "Deep feature for text-dependent speaker verification," *Speech Commun.*, vol. 73, pp. 1–13, 2015.

[5] E. Variani, X. Lei, E. McDermott, I. Lopez Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2014, pp. 4052–4056.

[6] Z. Wu, N. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, and H. Li, "Spoofing and countermeasures for speaker verification: A survey," *Speech Commun.*, vol. 66, pp. 130–153, 2015.

[7] J. Villalba and E. Lleida, "Preventing replay attacks on speaker verification systems," in *Proc. 2011 IEEE Int. Carnahan Conf. Security Technol.*, 2011, pp. 1–8.

[8] J. Villalba and E. Lleida, "Detecting replay attacks from far-field recordings on speaker verification systems," in *European Workshop on Biometrics and Identity Management*. New York, NY, USA: Springer, 2011, pp. 274–285.

[9] Z.-F. Wang, G. Wei, and Q.-H. He, "Channel pattern noise based playback attack detection algorithm for speaker recognition," in *Proc. 2011 Int. Conf. Mach. Learn. Cybern.*, 2011, vol. 4, pp. 1708–1713.

[10] Z. Wu, C. E. Siong, and H. Li, "Detecting converted speech and natural speech for anti-spoofing attack in speaker recognition," in *Proc. InterSpeech*, 2012, pp. 1700–1703.

[11] Z. Wu, T. Kinnunen, E. S. Chng, H. Li, and E. Ambikairajah, "A study on spoofing attack in state-of-the-art speaker verification: The telephone speech case," in *Proc. 2012 Asia–Pac. Signal Inf. Process. Assoc. Annu. Summit Conf.*, 2012, pp. 1–5.

[12] A. Ogihara, U. Hitoshi, and A. Shiozaki, "Discrimination method of synthetic speech using pitch frequency against synthetic speech falsification," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. 88, no. 1, pp. 280–286, 2005.

[13] P. L. De Leon, B. Stewart, and J. Yamagishi, "Synthetic speech discrimination using pitch pattern statistics derived from image analysis," in *Proc. InterSpeech*, 2012, pp. 370–373.

[14] L.-W. Chen, W. Guo, and L.-R. Dai, "Speaker verification against synthetic speech," in *Proc. 2010 7th Int. Symp. Chin. Spoken Lang. Process.*, 2010, pp. 309–312.

[15] T. Satoh, T. Masuko, T. Kobayashi, and K. Tokuda, "A robust speaker verification system against imposture using an HMM-based speech synthesis system," in *Proc. InterSpeech*, 2001, pp. 759–762.

[16] N. Chen, Y. Qian, H. Dinkel, B. Chen, and K. Yu, "Robust deep feature for spoofing detection—The SJTU system for ASVspoof 2015 challenge," in *Proc. InterSpeech*, 2015, pp. 2097–2101.

[17] Y. Qian, N. Chen, and K. Yu, "Deep features for automatic spoofing detection," *Speech Commun.*, vol. 85, pp. 43–52, 2016.

[18] Z. Wu *et al.*, "ASVspoof 2015: The first automatic speaker verification spoofing and countermeasures challenge," in *Proc. InterSpeech*, 2015, pp. 2037–2041.

[19] T. B. Patel and H. A. Patil, "Combining evidences from mel cepstral, cochlear filter cepstral and instantaneous frequency features for detection of natural vs. spoofed speech," in *Proc. InterSpeech*, 2015, pp. 2062–2066.

[20] M. Todisco, H. Delgado, and N. Evans, "A new feature for automatic speaker verification anti-spoofing: Constant Q cepstral coefficients," in *Proc. Odyssey*, 2016, pp. 249–252.

[21] F. Alegre, R. Vipperla, and N. Evans, "Spoofing countermeasures for the protection of automatic speaker recognition systems against attacks with artificial signals," in *Proc. InterSpeech*, 2012.

[22] Y. Liu, Y. Tian, L. He, J. Liu, and M. T. Johnson, "Simultaneous utilization of spectral magnitude and phase information to extract supervectors for speaker verification anti-spoofing," in *Proc. InterSpeech*, 2015, pp. 2082–2086.

[23] M. J. Alam, P. Kenny, V. Gupta, and T. Stafylakis, "Spoofing detection on the ASVspoof2015 challenge corpus employing deep neural networks," in *Proc. Odyssey*, 2016, pp. 270–276.

[24] A. Varga and H. J. Steeneken, "Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech Commun.*, vol. 12, no. 3, pp. 247–251, 1993.

[25] H.-G. Hirsch and D. Pearce, "The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *Proc. Automat. Speech Recognit., Challenges New Millenium ISCA Tuts. Res. Workshop*, 2000, pp. 29–32.

[26] J. Li, L. Deng, Y. Gong, and R. Haeb-Umbach, "An overview of noise-robust automatic speech recognition," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 4, pp. 745–777, Apr. 2014.

[27] J. Ming, T. J. Hazen, J. R. Glass, and D. A. Reynolds, "Robust speaker recognition in noisy conditions," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 5, pp. 1711–1723, Jul. 2007.

[28] X. Tian, Z. Wu, X. Xiao, E. S. Chng, and H. Li, "An investigation of spoofing speech detection under additive noise and reverberant conditions," in *Proc. InterSpeech*, 2016, pp. 1715–1719.

[29] C. Hanilçi, T. Kinnunen, M. Sahidullah, and A. Sizov, "Spoofing detection goes noisy: An analysis of synthetic speech detection in the presence of additive noise," *Speech Commun.*, vol. 85, pp. 83–97, 2016.

[30] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[31] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Proc. InterSpeech*, 2011, pp. 437–440.

[32] G. Hinton *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[33] J. Villalba, A. Miguel, A. Ortega, and E. Lleida, "Spoofing detection with DNN and one-class SVM for the ASVspoof 2015 challenge," in *Proc. 16th Annu. Conf. Int. Speech Commun. Assoc.*, 2015, pp. 2067–2071.

[34] C. Zhang, C. Yu, and J. H. Hansen, "An investigation of deep-learning frameworks for speaker verification antispoofing," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 4, pp. 684–694, Jun. 2017.

[35] D. Garcia-Romero, X. Zhou, and C. Y. Espy-Wilson, "Multicondition training of Gaussian PLDA models in i-vector space for noise and re-

verberation robust speaker recognition," in *Proc. 2012 IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2012, pp. 4257–4260.
[36] M. L. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 7398–7402.
[37] C. Zhang *et al.*, "Joint information from nonlinear and linear features for spoofing detection: An i-vector/DNN based approach," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2016, pp. 5035–5039.
[38] S. Thomas, S. Ganapathy, and H. Hermansky, "Multilingual MLP features for low-resource LVCSR systems," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2012, pp. 4269–4272.
[39] Y. Miao, F. Metze, and S. Rawat, "Deep maxout networks for low-resource speech recognition," in *Proc. 2013 IEEE Workshop Automat. Speech Recognit. Understanding*, 2013, pp. 398–403.
[40] F. Grézl, M. Karafiát, S. Kontár, and J. Cernocky, "Probabilistic and bottleneck features for LVCSR of meetings," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2007, vol. 4, pp. IV-757–IV-760.
[41] Y.-h. Chen, I. Lopez-Moreno, T. N. Sainath, M. Visontai, R. Alvarez, and C. Parada, "Locally-connected and convolutional neural networks for small footprint speaker recognition," in *Proc. 16th Annu. Conf. Int. Speech Commun. Assoc.*, 2015, pp. 1136–1140.
[42] D. Yu, M. L. Seltzer, J. Li, J.-T. Huang, and F. Seide, "Feature learning in deep neural networks-studies on speech recognition tasks," arXiv:1301.3605, 2013.
[43] T. N. Sainath *et al.*, "Deep convolutional neural networks for large-scale speech tasks," *Neural Netw.*, vol. 64, pp. 39–48, 2015.
[44] M. Bi, Y. Qian, and K. Yu, "Very deep convolutional neural networks for LVCSR," in *Proc. InterSpeech*, 2015, pp. 3259–3263.
[45] T. Sercu, C. Puhrsch, B. Kingsbury, and Y. LeCun, "Very deep multilingual convolutional neural networks for LVCSR," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2016, pp. 4955–4959.
[46] Y. Qian, M. Bi, T. Tan, and K. Yu, "Very deep convolutional neural networks for noise robust speech recognition," *IEEE/ACM Trans. Audio, Speech Lang. Process.*, vol. 24, no. 12, pp. 2263–2276, Dec. 2016.
[47] Y. Qian and P. Woodland, "Very deep convolutional neural networks for robust speech recognition," in *Proc. Spoken Lang. Technol. Workshop*, 2016, pp. 481–488.
[48] A. Graves, S. Fernández, and J. Schmidhuber, "Bidirectional LSTM networks for improved phoneme classification and recognition," in *Artificial Neural Networks: Formal Models and Their Applications—ICANN 2005*. New York, NY, USA: Springer, 2005, pp. 799–804.
[49] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Netw.*, vol. 18, no. 5, pp. 602–610, 2005.
[50] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 4, pp. 788–798, May 2011.
[51] Q. Jin and A. Waibel, "Application of LDA to speaker recognition," in *Proc. InterSpeech*, 2000, pp. 250–253.
[52] M. McLaren and D. Van Leeuwen, "Source-normalised-and-weighted LDA for robust speaker recognition using i-vectors," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2011, pp. 5456–5459.
[53] R. Lippmann, E. Martin, and D. Paul, "Multi-style training for robust isolated-word speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1987, vol. 12, pp. 705–708.
[54] T. Tan *et al.*, "Speaker-aware training of LSTM-RNNs for acoustic modelling," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2016, pp. 5280–5284.
[55] Y. Qian, T. Tan, D. Yu, and Y. Zhang, "Integrated adaptation with multi-factor joint-learning for far-field speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2016, pp. 5770–5775.
[56] Y. Qian, T. Tan, and D. Yu, "Neural network based multi-factor aware joint training for robust speech recognition," *IEEE/ACM Trans. Audio, Speech Lang. Process.*, vol. 24, no. 12, pp. 2231–2240, Dec. 2016.
[57] S. Bu, Y. Qian, and K. Yu, "A novel dynamic parameters calculation approach for model compensation," in *Proc. InterSpeech*, 2014, pp. 2744–2748.
[58] Wikipedia, "Topography." [Online]. Available: https://en.wikipedia.org/wiki/Topography
[59] H. Soltau, G. Saon, and T. N. Sainath, "Joint training of convolutional and non-convolutional neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2014, pp. 5609–5613.
[60] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," arXiv:1207.0580, 2012.
[61] S. Rennie, V. Goel, and S. Thomas, "Annealed dropout training of deep networks," in *Proc. Spoken Lang. Technol. Workshop*, 2014, pp. 159–164.
[62] N. Brümmer and E. de Villiers, "The BOSARIS toolkit: Theory, algorithms and code for surviving the new DCF," arXiv:1304.2865, 2013.
[63] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv:1412.6980, 2014.

Yanmin Qian (S'09–M'13) received the B.S. degree from the Department of Electronic and Information Engineering, Huazhong University of Science and Technology, Wuhan, China, in 2007, and the Ph.D. degree from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2012. In 2013, he joined the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China, where he is currently an Associate Professor. From 2015 to 2016, he was an Associate Researcher in the Speech Group, Department of Engineering, Cambridge University, Cambridge, U.K. His research interests include the acoustic and language modeling in speech recognition, speaker and language recognition, key word spotting, and multimedia signal processing.

Nanxin Chen (S'14) received the B.Sc. degree in computer science and technology from Shanghai Jiao Tong University, Shanghai, China, in 2015, under the supervision of Prof. Kai Yu and Prof. Yanmin Qian. He is currently working toward the Ph.D. degree in the Center for Language and Speech Processing, Johns Hopkins University, Baltimore, MD, USA, advised by Najim Dehak. He is involved in deep learning, Bayesian inference, and other state-of-the-art machine learning techniques at present. His research interests include speaker verification, speech profiling, and pattern recognition. In 2015, he got the Excellent Students of China Computer Federation award.

Heinrich Dinkel (S'17) received the B.Sc. degree from the Computer Science Department, Furtwangen Applied Science University, Furtwangen, Germany, in 2014, and the M.Sc. degree, in 2017, from the Computer Science Department, Shanghai Jiao Tong University, Shanghai, China, where he is currently working toward the Ph.D. degree in the field of speaker separation. His research interests include speaker recognition, verification, and separation, as well as spoofing detection.

Zhizheng Wu received the Ph.D. degree from Nanyang Technological University, Singapore. Since May 2016, he has been a Research Scientist in the Apple Inc., Cupertino, CA, USA, prior to which he was a Research Fellow at University of Edinburgh from 2014 to 2016. During his studies, he joined Microsoft Research Asia (2007–2009) and the University of Eastern Finland (2012) as a Visiting Scientist and received the best paper award at the Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC) 2012. He co-organized the first Automatic Speaker Verification Spoofing and Countermeasures Challenge at Interspeech 2015 and the first Voice Conversion Challenge as a special session at Interspeech 2016. He delivered a tutorial on Spoofing and Anti-Spoofing: A Shared View of Speaker Verification, Speech Synthesis, and Voice Conversion at APSIPA ASC 2015. He is the Principal Architect of the open-source speech synthesis system, Merlin.