Boosting Contextual Information for Deep Neural Network Based Voice Activity Detection

Xiao-Lei Zhang, *Member, IEEE*, and DeLiang Wang, *Fellow, IEEE*

***Abstract*—Voice activity detection (VAD) is an important topic in audio signal processing. Contextual information is important for improving the performance of VAD at low signal-to-noise ratios. Here we explore contextual information by machine learning methods at three levels. At the top level, we employ an ensemble learning framework, named multi-resolution stacking (MRS), which is a stack of ensemble classifiers. Each classifier in a building block inputs the concatenation of the predictions of its lower building blocks and the expansion of the raw acoustic feature by a given window (called a resolution). At the middle level, we describe a base classifier in MRS, named boosted deep neural network (bDNN). bDNN first generates multiple base predictions from different contexts of a single frame by only one DNN and then aggregates the base predictions for a better prediction of the frame, and it is different from computationally-expensive boosting methods that train ensembles of classifiers for multiple base predictions. At the bottom level, we employ the multi-resolution cochleagram feature, which incorporates the contextual information by concatenating the cochleagram features at multiple spectrotemporal resolutions. Experimental results show that the MRS-based VAD outperforms other VADs by a considerable margin. Moreover, when trained on a large amount of noise types and a wide range of signal-to-noise ratios, the MRS-based VAD demonstrates surprisingly good generalization performance on unseen test scenarios, approaching the performance with noise-dependent training.**

***Index Terms*—Cochleagram, deep neural network, ensemble learning, multi-resolution stacking, noise-independent training, voice activity detection.**

I. Introduction

VOICE activity detection (VAD) is an important preprocessor for many audio signal processing systems. For example, it improves the efficiency of a speech communication system [1] by detecting and transmitting only speech signals. It helps a speech enhancement algorithm [2] or a speech recognition system [3], [4] by filtering out silence and noise segments. One of the major challenges of VAD is to make it perform in low signal-to-noise ratio (SNR) environments. Early research focused on signal processing based acoustic features, including energy in the time domain, pitch detection, zero-crossing rate, and several spectral energy based features such as energy-entropy, spectral correlation, spectral divergence, higher-order statistics [5]. Recent development includes low-frequency ultrasound [6] and single frequency filtering [7]. Exploring features is important in improving VAD research from the aspect of acoustic mechanism. However, each acoustic feature reflects only some characteristics of human voice. Moreover, using the features independently is not very effective in extremely difficult scenarios. Hence, fusing the features together as the input of some data-driven methods may be an effective usage of the features for improving the overall performance of VAD.

Another important research branch of VAD is statistical signal processing. These techniques make model assumptions on the distributions of speech and background noise (usually in the spectral domain) respectively, and then design statistical algorithms to dynamically estimate the model parameters. Typical model assumptions include the Gaussian distribution [8], [9], Laplace distribution [10], Gamma distribution [11], or their combinations [11]. The most popular parameter estimation method is the minimum mean square error estimation [12]. In addition, long-term contextual information is shown to be useful in improving the performance [13]. Due to the simplicity of the model assumptions and online updating of the parameters, this kind of methods may generate reasonable results in various noise scenarios. In many cases, they work better than energy based methods. But statistical model based methods have limitations. First, model assumptions may not fully capture global data distributions, since the models usually have too few parameters and they estimate parameters on-the-fly from limited local observations. Second, with relatively few parameters, they may not be flexible enough in fusing multiple acoustic features. Moreover, most methods update parameters during the pure noise phase which may cause them fail when the noise changes rapidly during the voice phase.

The third popular branch of VAD research is machine learning methods, which train acoustic models from given noisy corpora and apply the models to real-world test environments. They have two main research objectives: one is to improve the *discriminative ability* of models when the noise scenarios of training and test corpora are matching; the other is to improve the *generalization ability* (i.e. detection accuracy) of models to test noise scenarios when the test noise scenarios are unseen from or mismatching with the training noise scenarios.

Most machine learning methods focus on how to improve the discriminative ability. We summarize them briefly as follows. In terms of whether their training corpora are manually labeled, they can be categorized to *unsupervised learning* which uses

Manuscript received May 21, 2015; revised September 04, 2015; accepted December 01, 2015. Date of publication December 04, 2015; date of current version January 04, 2016. The work was supported in part by the Air Force Office of Scientific Research under Grant FA9550-12-1-0130. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Yunxin Zhao.

The authors are with the Department of Computer Science and Engineering and Center for Cognitive and Brain Sciences, The Ohio State University, Columbus, OH 43210-1277 USA (e-mail: xiaolei.zhang9@gmail.com; dwang@cse.ohio-state.edu).

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.2015.2505415

2329-9290 © 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

Authorized licensed use limited to: NATIONAL TAIWAN NORMAL UNIVERSITY. Downloaded on February 15,2020 at 05:29:19 UTC from IEEE Xplore. Restrictions apply.

unlabeled training corpora, or *supervised learning* which uses labeled training corpora. Many unsupervised methods belong to dimensionality reduction, which first extract noise-robust low-dimensional features from highly-variant high-dimensional observations and then apply the features to classifiers. They include principle component analysis [14], non-negative matrix factorization [15], and spectral decomposition of graph Laplacian [16]. Some methods use clustering algorithms directly, such as k-means clusterings [17] and Gaussian mixture models [14]. Unsupervised methods are able to explore multiple features and train robust models from vast amount of recorded data, however, when the tasks are too difficult that most speech signal is drowned in back ground noise, such as babble noise with an SNR below 0 dB, unsupervised methods are helpless. Note that, statistical signal processing based VADs can also be regarded as unsupervised methods, which train models from a few local observations and accumulated historical information.

Supervised learning methods take VAD as a binary-class classification problem—speech or non-speech. The techniques can be roughly categorized to four classes: probabilistic models, kernel methods, neural networks, and ensemble methods. Probabilistic models include Gaussian mixture models [18] and conditional random fields [15]. Kernel methods mainly include various support vector machines (SVM), such as [19], [20]. These two kinds cannot handle large-scale corpora well, so that they are difficult to be used in practice since we need large-scale training corpora to cover rather complicated real-world noisy environments.

Recently, deep neural networks (DNN) and their extensions [21]–[26], which have a strong scalability to large-scale corpora, showed good performance in extremely difficult scenarios and are competitive in real-world applications. Specifically, in [21], Zhang and Wu proposed to apply standard deep belief networks to VAD and reported better performance than SVM, where the networks were pretrained as in [4]. In [25], Zhang and Wang further proposed to generate multiple different predictions from a single DNN by boosting contextual information and reported significant improvement over the standard DNN in difficult noise scenarios and at low SNR levels. In [22], [23], the authors applied deep recurrent neural networks to capture historical contextual information and reported significant improvement over Gaussian mixture models and statistical signal processing methods. However, the performance improvements of the aforementioned DNN methods were observed when the DNNs were trained *noise-dependently*, i.e. the noise scenarios of training and test are matching. When the DNN-based VADs were applied to unseen test scenarios, the performance dropped significantly as shown in [23], [27]. Recently, in [24], [26], the authors trained DNN and convolutional neural networks together from large-scale real-world data [28] and demonstrated impressive two-phase improvements. However, because each model in [24], [26] was binded to a given channel, we still do not know exactly how the models will generalize to different noise scenarios. Due to the restriction of the task setting, the results do not have a quantitative evaluation on how the models vary with SNR levels, which need a further investigation.

To summarize, DNN-based VADs with noise dependent training have demonstrated good performance and have shown strong potential in practice. In this paper, we further develop DNN-based VADs by exploring contextual information heavily in three novel levels. Motivated by recent progress of speech separation [29], [30], we also investigate quantitatively how DNN-based VADs can generalize to unseen test noise scenarios with the variation of SNR through noise-independent training. The main contributions of this paper are summarized as follows:

- **Multi-resolution stacking (MRS).** MRS is a stack of ensemble classifiers. Each classifier in a building block takes the concatenation of the soft output predictions of the lower building block and the expansion of the original acoustic feature in a window (called a resolution). The classifiers in the same building block have different resolutions, which is the novelty of this framework.
- **Boosted deep neural network (bDNN).** bDNN is proposed as the base classifier of MRS. It first generates multiple base predictions on a frame by boosting the contextual information of the frame, and then aggregates the base predictions for a stronger one. bDNN generates multiple predictions from a single DNN, which is its novelty compared to ensemble DNNs. Preliminary results [25] showed that it can significantly outperform DNN-based VAD without increasing computational complexity.
- **Multi-resolution cochleagram (MRCG) feature.** MRCG [31], which was first proposed for speech separation, is employed as a new acoustic feature for VAD. It concatenates multiple cochleagram features calculated at different spectral and temporal resolutions.
- **Noise-independent training.** We train the proposed method with a corpus that has a vast amount of noise scenarios with a wide variation of SNR levels, and test it in unseen and difficult noise scenarios. We find that the method can generalize well.

Empirical results on the AURORA2 [32] and AURORA4 corpora [33] show that the MRS-based VAD outperforms a DNN-based VAD [21] and 5 other comparison methods. Moreover, when the proposed method is trained noise-independently, its performance on unseen test noise scenarios at various SNR levels is surprisingly as good as the proposed method with noise-dependent training. See Supplementary Material[1] for more results and [34] for the long version. This paper differs from our preliminary work [25] in several major aspects, which include the use of MRS and noise-independent training in this paper (but not in [25]) and new parameter settings for bDNN and MRCG. Consequently, experimental results in this paper are different from those reported in [25].

The paper is organized as follows. In Section II, we introduce the MRS framework. In Section III, we present the bDNN model. In Section IV, we introduce the MRCG feature. In Section V, we present results with noise-dependent training. In Section VI, we present results with noise-independent training. Finally, we conclude in Section VII.

II. Multi-Resolution Stacking

We formulate VAD as a supervised classification problem. Specifically, a long speech signal is divided to multiple short-term overlapped frames, each of which ranges usually from 10

[1]Available at https://sites.google.com/site/zhangxiaolei321/vad

Authorized licensed use limited to: NATIONAL TAIWAN NORMAL UNIVERSITY. Downloaded on February 15,2020 at 05:29:19 UTC from IEEE Xplore. Restrictions apply.

to 25 milliseconds. Each frame in the time domain is transformed to an acoustic feature in the spectral domain, denoted as $\mathbf{x}_m$, where $m = 1, \ldots, M$ indexes time frame. To construct a training set of a classification problem, $\mathbf{x}_m$ is manually labeled as $y_m = 1$ or $y_m = 0$, indicating $\mathbf{x}_m$ is a speech or noise frame respectively. A classifier $f(\cdot)$ is trained on $\{(\mathbf{x}_m, y_m)\}_{m=1}^{M}$ and tested on another set $\{\mathbf{x}_n\}_{n=1}^{N}$.

It is known that contextual information is important in improving the performance of VAD. One common technique to incorporate contextual information is to train models with a fixed window length that performs the best among several choices of window lengths. We denote the technique of adding a window to incorporate neighboring frames the *resolution*. Here, we argue that (i) for a certain task, although only one resolution performs the best, other resolutions may still provide useful information that may further improve the performance; (ii) although we can manage to pick the best resolution for a certain task, it is still inconvenient to do so case by case. We propose a simple framework, named *multi-resolution stacking*, to solve the two problems together.

As described in Fig. 1, MRS is a stack of classifier ensembles. In the training stage of MRS, suppose we are to train S building blocks ($S = 3$ in Fig. 1). The sth building block has K_s classifiers, denoted as $\{f_{s,k}(\cdot)\}_{k=1}^{K_s}$, each of which has a predefined resolution $W_{s,k}$. The kth classifier $f_{s,k}(\cdot)$ takes vector $\mathbf{z}_{s,k,m}$ as the input:

$$\mathbf{x}'_{s,k,m} = \begin{cases} \mathbf{v}_{s,k,m} & \text{if } s = 1 \\ \left[\hat{y}_{s-1,1,m}, \ldots, \hat{y}_{s-1,K_{s-1},m}, \mathbf{v}_{s,k,m}{}^T\right]^T & \text{if } s > 1 \end{cases} \quad (1)$$

and takes y_m as the training target, where $\{\hat{y}_{s-1,k',m}\}_{k'=1}^{K_{s-1}}$ are the soft predictions of $\mathbf{x}_m$ produced by the $(s-1)$th building block and $\mathbf{v}_{s,k,m}$ is an expansion of $\mathbf{x}_m$ given the resolution $W_{s,k}$:

$$\mathbf{v}_{s,k,m} = \left[\mathbf{x}_{m-W_{s,k}}^T, \mathbf{x}_{m-W_{s,k}+1}^T \ldots, \mathbf{x}_m^T, \ldots, \mathbf{x}_{m+W_{s,k}-1}^T, \mathbf{x}_{m+W_{s,k}}^T\right]^T \quad (2)$$

After $f_{s,k}(\cdot)$ is trained, it produces a soft prediction $\hat{y}_{s,k,m}$ of $\mathbf{x}_m$ for the upper building block.

The resolution W will double the size of training data, therefore, MRS is hard to handle both a large W and a large training set. To reduce the memory requirement of computing power, we present a trick: one can pick a subset of frames within the window instead of all frames. In this paper, we replace parameter W by a new pair of parameters (W, u) which chooses the neighboring frames indexed by $\{-W, -W + u, -W + 2u, \ldots, -1 - u, -1, 0, 1, 1 + u, \ldots, W - 2u, W - u, W\}$ and derives the following feature:

$$\mathbf{v}_{s,k,m} \left[\mathbf{x}_{m-W_{s,k}}^T, \mathbf{x}_{m-W_{s,k}+u}^T \ldots, \mathbf{x}_{m-1-u}^T, \mathbf{x}_{m-1}^T, \mathbf{x}_m^T, \mathbf{x}_{m+1}^T, \mathbf{x}_{m+1+u}^T, \ldots, \mathbf{x}_{m+W_{s,k}-u}^T, \mathbf{x}_{m+W_{s,k}}^T\right]^T \quad (3)$$

where u is a user defined integer parameter. This trick not only makes all classifiers in a building block have the same memory requirement but also does not decrease the performance significantly in experience.

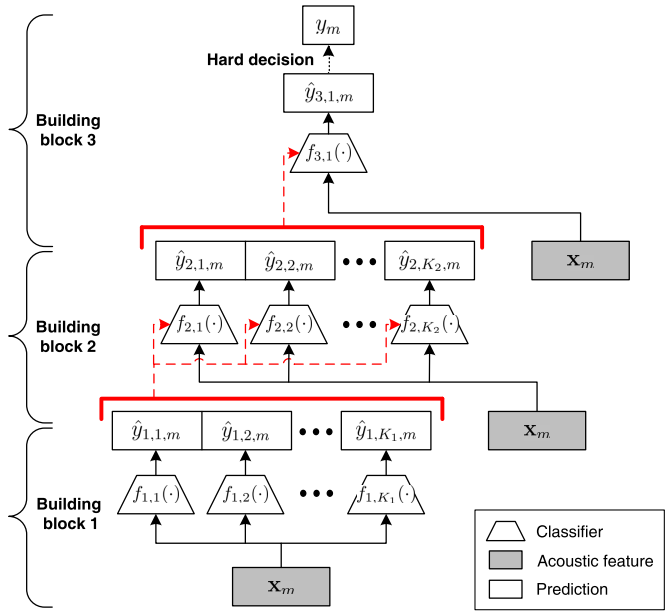

Fig. 1. Principle of multi-resolution stacking. The soft predictions of all base classifiers in a building block are combined in the red line as part of the input of the base classifiers in the upper building block. The input of a base classifier is the concatenation of the soft predictions from the lower building block and the acoustic feature that is extended by a window.

In the test stage of MRS, we obtain a serial soft predictions as we did in the training stage from the bottom building block to the top building block. After getting the output of the Sth building block which contains only one classifier as shown in Fig. 1, we do a hard decision on the output, e.g. $\hat{y}_{S,1,n}$, by:

$$\bar{y}_n = \begin{cases} 1 & \text{if } \hat{y}_{S,1,n} \geq \delta \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

and take $\bar{y}_n$ as the final prediction of the test frame $\mathbf{x}_n$, where $\delta \in [0, 1]$ is a decision threshold tuned on a development set.

A. Motivation

This paper uses "boosting" as a concept of ensemble learning [35], somewhat different from its use in AdaBoost [36] and its variants which recursively add new base classifiers that discriminate misclassified training data points made by previous base classifiers. The theory of weak learnability [37], which is a cornerstone of ensemble learning, suggests that an ensemble of weak learners can group to a strong learner, if (i) the weak learners are stronger than random guess and (ii) the weak learners are different from each other in terms of classification errors. Ensemble learning has four kinds of techniques: manipulating training data points, features, hyperparameters of classifiers, and output targets.

MRS integrates the base predictions of a lower building block into the training process of the upper building block by *manupulating output targets*. It is derived as follows. The simplest stacking technique is majority voting, which averages the base predictions in the upper building block. However, an ensemble method needs hundreds of base predictions to reach an improved final prediction, which is too costly for computationally-expensive base learners, such as DNN. To

Authorized licensed use limited to: NATIONAL TAIWAN NORMAL UNIVERSITY. Downloaded on February 15,2020 at 05:29:19 UTC from IEEE Xplore. Restrictions apply.

overcome this problem, we need a stronger upper building block than majority voting. DNN, which nonlinearly combines the base predictions, meets our requirement. However, a few base predictions do not provide enough information for DNN to reach a reasonable result. To overcome this problem, we take both the base predictions from the lower building block and original features as the input of DNN. The original features ensure that the amount of information will not decrease with the increase of the number of the building blocks. On the other hand, the base predictions provide additional information for performance improvement.

Note that the principles of AdaBoost, which corrects training errors recursively, is not suitable for VAD. It is known that, when a data set is noisy, AdaBoost easily overfits [38].

B. Related Work

In noise-robust speech signal processing, using the optimal resolution is a common technique, such as statistical signal processing based VADs [13] and recent machine learning based VADs. However, the models with suboptimal resolutions may also provide useful information. Fusing ensemble models is another common technique, such as fusing DNN and convolutional neural networks in [24], [26], but, they do not consider different resolutions and stacking, and do not take the raw feature as the input of the consensus model. Stacking ensemble classifiers has been used in speech separation [39] and recognition [40], but they did not consider different resolutions. To summarize, stacking ensembles of classifiers in different resolutions are the novelty of the framework.

III. Boosted DNN

In this section, we fill MRS by a strong base classifier—boosted DNN. We also introduce our DNN model in Section III-A and motivation of bDNN in Section III-B.

Deep neural network is a strong classifier that can approach to the minimum expectation risk—the ideal minimum risk given the infinite amount of training data—when the input data is large scale. It has been adopted in recent VAD studies. One common technique to further improve the prediction accuracy of DNN is ensemble learning, which trains multiple DNNs that yield different base predictions, such that when the base predictions are aggregated, the final prediction is boosted to be better than any of the base predictions. However, it is too expensive to train a set of DNNs if they do not receive significantly different knowledge from the input. To alleviate the computational load but benefit from ensemble learning, we proposed bDNN, which can generate multiple different base predictions on a single frame by training only one DNN.

In the training stage of bDNN, we expand each training frame $\mathbf{x}_m$ to $\mathbf{x}'_m$ by Eq. (1). Different from DNN training, we further expand y_m to $\mathbf{y}'_m$ by a squared window:

$$\mathbf{y}'_m = [y_{m-W}, y_{m-W+1}, \ldots, y_m, \ldots, y_{m+W-1}, y_{m+W}]^T \tag{5}$$

The square window is the simplest form to incorporate neighboring labels of y_m. Applying different windows, particularly those emphasizing the importance of y_m, could further improve the performance (see Section VII).

Given the new training target $\mathbf{y}'_m$, bDNN is a DNN model trained on a new corpus $\{(\mathbf{x}'_m, \mathbf{y}'_m)\}_{m=1}^M$. It has $(2W+1)d$ input units when bDNN is used in the bottom building block of MRS, and $(2W+1)d + K_{s-1}$ input units when bDNN is not in the bottom building block of MRS, where d is the dimension of $\mathbf{x}_m$. It has $2W+1$ output units. It optimizes the following objective by backpropagation training:

$$\min_{\boldsymbol{\alpha}} \sum_{m=1}^{M} \|\mathbf{y}'_m - f_{\boldsymbol{\alpha}}(\mathbf{x}'_m)\|^2 \tag{6}$$

where $f_{\boldsymbol{\alpha}}(\cdot)$ is the DNN mapping function, $\boldsymbol{\alpha}$ is the parameter of DNN, and $\|\cdot\|^2$ denotes the squared loss.

In the test stage of bDNN, we aim to predict the label of frame $\mathbf{x}_n$, which consists of three steps as shown in Fig. 2. The first step expands $\mathbf{x}_n$ to a large observation $\mathbf{x}'_n$ as done in the training phase, so as to get a new test corpus $\{\mathbf{x}'_n\}_{n=1}^N$ (Fig. 2(A)). The second step gets the $(2W+1)$-dimensional prediction of $\mathbf{x}'_n$ from the DNN, denoted as $\mathbf{y}'_n = [y_{n-W}^{(-W)}, y_{n-W+1}^{(-W+1)}, \ldots, y_n^{(0)}, \ldots, y_{n+W-1}^{(W-1)}, y_{n+W}^{(W)}]^T$ (Fig. 2(B)). The third step aggregates the results to reach the soft prediction of $\mathbf{x}_n$, denoted as $\hat{y}_n$ (Fig. 2(C)):

$$\hat{y}_n = \frac{\sum_{w=-W}^{W} y_n^{(w)}}{2W+1} \tag{7}$$

Finally, we make a hard decision by

$$\bar{y}_n = \begin{cases} 1 & \text{if} \quad \hat{y}_n \geq \eta \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where $\eta \in [0,1]$ is a decision threshold tuned on a development set.

Note that when we adopt the trick in Section II to alleviate the memory requirement, Eq. (5) should be modified accordingly as follows:

$$\mathbf{y}'_m = [y_{m-W}, y_{m-W+u}, \ldots, y_{m-1-u}, y_{m-1}, y_m, y_{m+1}, y_{m+1+u}, \ldots, y_{m+W-u}, y_{m+W}]^T \tag{9}$$

A. DNN Model

We adopt contemporary DNN training methods, and use the *area under the receiver operating characteristic curve* (AUC) as the performance metric for selecting the best DNN model in the training process, where an efficient calculation of AUC is provided in Supplementary Material.

The template of deep models is described as follows:

$$\mathbf{y} = h_o\left(h_{(L)}\left(\ldots h_{(l)}\left(\ldots h_{(2)}\left(h_{(1)}\left(\mathbf{x}^{(0)}\right)\right)\right)\right)\right) \tag{10}$$

where $l = 1, \ldots, L$ denotes the lth hidden layer from the bottom, $h_{(l)}(\cdot)$ is a group of nonlinear mapping functions (or units) at the lth layer, $h_o(\cdot)$ is the output layer, and $\mathbf{x}^{(0)}$ is the input feature vector. We use the rectified linear unit $y = \max(0, x)$ as the unit of the hidden layers, sigmoid function

Authorized licensed use limited to: NATIONAL TAIWAN NORMAL UNIVERSITY. Downloaded on February 15,2020 at 05:29:19 UTC from IEEE Xplore. Restrictions apply.

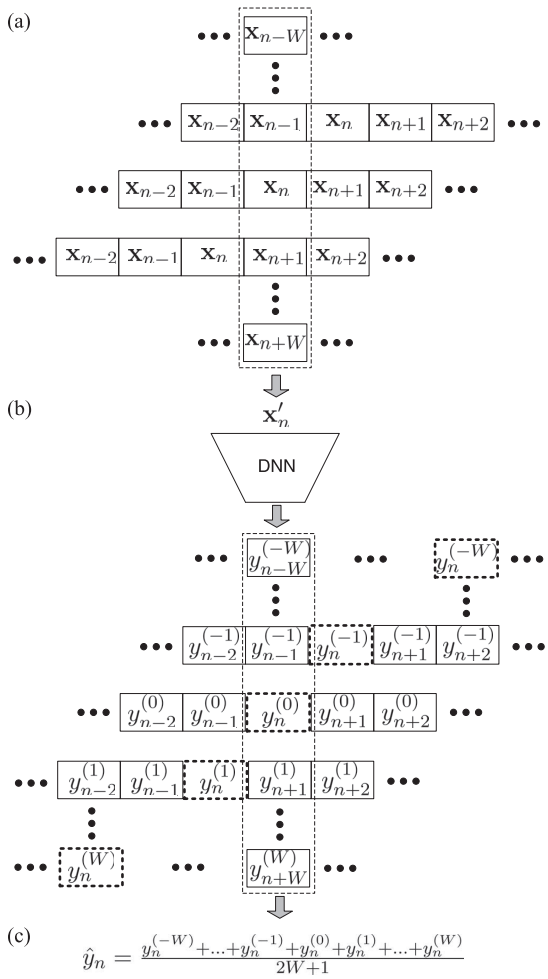

Fig. 2. Test phase of bDNN. (A) Expanding $\mathbf{x}_n$ to a new feature (included in the dashed rectangle, denoted as $\mathbf{x}'_n$) given the half-window size W. (B) Predicting labels of $\mathbf{x}'_n$ to produce a $(2W+1)$-dimensional vector (included in the dashed rectangle) by DNN. (C) Aggregating the prediction results by the given equation from the soft output units drawn in the bold dashed rectangles of Fig. 1(B).

$y = \frac{1}{1+e^{-x}}$ as the unit of the output layer. Rectified linear unit can be trained faster than the traditional sigmoid function and helps DNN learn local patterns better.

We use a dropout strategy [41] to regularize the DNN model. Dropout randomly deactivates the units in a given layer. Specifically, the hidden units of a layer were dropped randomly with a given probability, such as 20%. The dropped units output 0 regardless their input. The upper layer takes the randomly corrupted feature as its input, and randomly deactivates its output units in the same way. Due to such a regularization, bDNN is able to train a much larger model with a stronger generalization ability than the standard DNN model in [21].

In addition, we employ the adaptive stochastic gradient descent [42] and a momentum term [43] to train DNN. These training schemes accelerate traditional gradient descent training and facilitate parallel computing. We do not use pretraining in our DNN training. Recent results show that, when a data set is large enough, the performance of DNN without pretraining is also good enough.

B. Motivation

Originally, we planed to first train multiple DNNs $g_{-W}(\cdot), \ldots, g_W(\cdot)$ and then aggregate the predictions of the DNNs. Specifically, each DNN learns a mapping function from an expansion of the input $\mathbf{x}_m$ to its manual label y_m, and the expansions in different DNNs use different sliding windows that incorporate $\mathbf{x}_m$ as part of input. For example, the ith DNN $g_i(\cdot)$ takes $\mathbf{x}_m^{(i)} = [\mathbf{x}_{m-W+i}^T, \ldots, \mathbf{x}_m^T, \ldots, \mathbf{x}_{m+i}^T, \ldots, \mathbf{x}_{m+W+i}^T]^T$ as its input and outputs the base prediction $y_m^{(i)}$. The ensemble method gets the final prediction $\hat{y}_m$ by aggregating the base predictions $\hat{y}_m = \sum_{i=-W}^{W} y_m^{(i)}/(2W+1)$.

After observing the fact that $[\mathbf{x}_{m-W}^T, \ldots, \mathbf{x}_m^T, \ldots, \mathbf{x}_{m+W}^T]^T$ appears as the expanded feature of $\mathbf{x}_{m-i}$ for training $y_m^{(i)} = g_i(\cdot)$ where $i = -W, \ldots, W$, we propose to integrate the outputs $y_m^{(i)}$ together and train a new DNN model:

$$\begin{bmatrix} y_{m-W}^{(-W)} \\ \vdots \\ y_{m+W}^{(W)} \end{bmatrix} = g\left(\begin{bmatrix} \mathbf{x}_{m-W} \\ \vdots \\ \mathbf{x}_{m+W} \end{bmatrix}\right) \tag{11}$$

where $g(\cdot)$ is the DNN model of bDNN that has multiple output units. Then, we aggregate the base predictions for the final prediction as in Eq. (7). The main difference between bDNN and the aforementioned inefficient method is that the base predictions $y_m^{(i)}$ of bDNN share the same parameters of the hidden units of a single DNN model, while the base predictions of the inefficient method are generated independently from multiple DNN models. bDNN saves the computational load greatly with some loss of flexibility of model training.

C. Related Work

1) On the Relationship Between Boosted DNN and the Common Boosting Techniques: For bDNN, the output target of the mth frame, i.e. y_m, is assumed to be generated from $[\mathbf{x}_{m-2W}^T, \ldots, \mathbf{x}_m^T, \ldots, \mathbf{x}_{m+2W}^T]^T$. bDNN generates multiple base predictions of y_m by extracting part of the input feature. The method of manipulating the input feature only is different from bagging [44] and Adaboost [36] which manipulate the input data set; it is also different from random forests [38] which manipulates the input data set and features together. We also tried to generate base predictions from different subsets of data, but we found that the performance was not as good as the performance produced from the entire data set due to the performance decrease of each base classifier.

2) On the Difference Between the bDNN Based VAD and the DNN Based VAD in [21] : The training targets of bDNN and the method in [21] are different. bDNN reformulates VAD as a structural learning problem that learns an encoder that projects the concatenated frames in a window to a binary code, while the method in [21] takes VAD as a traditional binary-class classification problem that predicts the classes of frames in sequence. The structural learning of bDNN fully utilizes the contextual information of the output.

The DNN implementations of bDNN and the method in [21] are also different in respect of the network structure and training method. (i) The DNN model in [21] does not use a regularization

Authorized licensed use limited to: NATIONAL TAIWAN NORMAL UNIVERSITY. Downloaded on February 15,2020 at 05:29:19 UTC from IEEE Xplore. Restrictions apply.

method (e.g., dropout). (ii) It uses the sigmoid function as the hidden unit and softmax function $y_i = \frac{e^{x_i}}{\sum_j e^{x_j}}$ as the output unit, which is not very effective in learning the local distribution of data. (iii) It uses pretraining to prevent bad local minima, which is unnecessary when a training set is large enough.

IV. MRCG Feature

In this section, we introduce the MRCG feature which was first proposed in [31] for speech separation.[2] The key idea of MRCG is to incorporate both local information and global information through multi-resolution extraction. The local information is produced by extracting cochleagram features with a small frame length and a small smoothing window (i.e., high resolutions). The global information is produced by extracting cochleagram features with a large frame length or a large smoothing window (i.e., low resolutions). It has been shown that cochleagram features with a low resolution, such as frame length $=$ 200 ms, can detect patterns of noisy speech better than that with only a high resolution, and features with high resolutions complement those with low resolutions. Therefore, concatenating them together is better than using them separately.

As illustrated in Fig. 3(A), MRCG is a concatenation of 4 cochleagram features with different window sizes and different frame lengths. The first and fourth cochleagram features are generated from two U-channel gammatone filterbanks ($U = 8$ in this paper) with frame lengths set to 20 ms and 200 ms respectively. The second and third cochleagram features are calculated by smoothing each time-frequency unit of the first cochleagram feature with two squared windows that are centered on the unit and have the sizes of 11×11 and 23×23. Because the windows on the first and last few channels (or frames) of the two cochleagram features may overflow, we cut off the overflowed parts of the windows. Note that the multi-resolution strategy is a common technique not limited to the cochleagram feature [45], [46].

After calculating the $(4 \times U)$-dimensional MRCG feature, we further calculate its Deltas and double Deltas, and then combine all three into a $(12 \times U)$-dimensional feature (Fig. 3(B)). A Delta feature is calculated by

$$\Delta x_n = \frac{(x_{n+1} - x_{n-1}) + 2(x_{n+2} - x_{n-2})}{10} \qquad (12)$$

where x_k is the kth unit of MRCG in a given channel. The double-Delta feature is calculated by applying equation (12) to the Delta feature.

The calculation of the U-dimensional cochleagram feature in Fig. 3(A) is detailed in Fig. 3(C). We first filter input noisy speech by the 8-channel gammatone filterbank, then calculate the energy of each time-frequency unit by $\sum_{k=1}^{K} s_{c,k}^2$ given the frame length K, and finally rescale the energy by $\log_{10}(\cdot)$, where $s_{c,k}$ represents the kth sample of a given frame in the cth channel [47].

Note that when MRCG is used for bDNN training, it should be normalized to zero means and unit standard deviations in dimension *globally*, and the normalization factors should be used to normalize each test frame, where the word "globally" means that the normalization is conducted on the entire training corpus but not on each training utterance separately.

[2]Code is downloadable from http://web.cse.ohio-state.edu/pnl/software.html

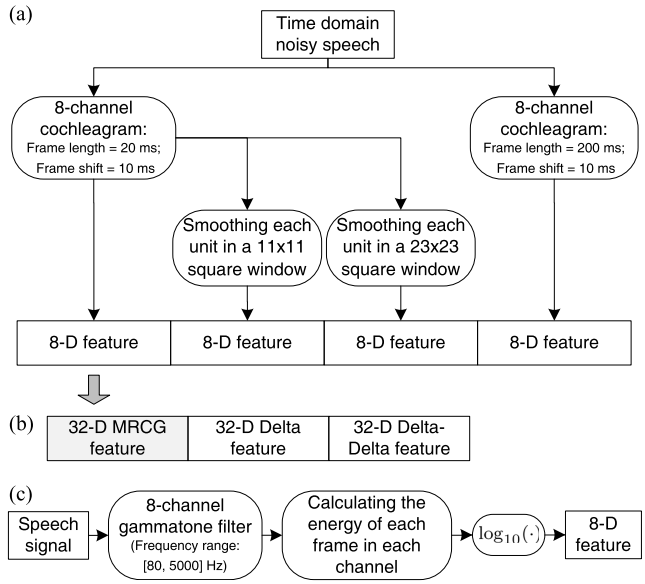

Fig. 3. The MRCG feature. (A) Diagram of the process of extracting a 32-dimensional MRCG feature. "$(2W + 1) \times (2W + 1)$ square window" means that the value of a given time-frequency unit is replaced by the average value of its neighboring units that fall into the window centered at the given unit and extending in the axes of time and frequency. (B) Expanding MRCG to a 96-dimensional feature that consists of the original MRCG feature, its Delta feature and Delta-Delta feature. (C) Calculation of the 8-dimensional cochleagram features in detail.

A. Related Work

For VAD research, the idea of incorporating both local and global information into an acoustic feature has been explored in [48]. There are three differences between MRCG and the features in [48], named long-term power level difference (LT-PLD) and short-term PLD (ST-PLD). First, MRCG uses cochleagram as the basic acoustic feature which is monaural, whereas LT-PLD and ST-PLD use PLD which is based on two microphones. Second, the smoothing of a time-frequency unit of MRCG spreads across the neighboring units along both the time and frequency axes, while the smoothing of a time-frequency unit of PLD spreads across the neighboring units along the time axis. Third, the smoothing range of a sub-feature of MRCG is controlled by a window, while the smoothing range of PLD is determined by a recursive averaging technique.

V. Evaluation Results of Noise-Dependent Models

The term *noise-dependent* (ND) means that the noise scenarios of the training and test sets of machine-learning-based models are the same in terms of noise types and SNR levels.

In this section, we first report the results of the proposed methods in Section V-B, then analyze how MRS, bDNN, and MRCG improve the performance over comparison methods in Section V-C and Section V-D, and finally analyze the advantage of MRCG over its components in Section V-E. We also report further results in Supplementary Material.

Authorized licensed use limited to: NATIONAL TAIWAN NORMAL UNIVERSITY. Downloaded on February 15,2020 at 05:29:19 UTC from IEEE Xplore. Restrictions apply.

A. Experimental Settings

1) Data Sets: We used the noisy speech corpora of AURORA2 [32] as well as the clean speech corpus of AURORA4 [33] mixed with the NOISEX-92 noise corpus [49] for evaluation. AURORA2 contains the pronunciations of digits. AURORA4 contains the utterances of continuous speech. The data sets were preprocessed as follows.

The ground-truth labels of each noisy speech corpus in either AURORA2 or AURORA4 were the results of Sohn VAD [8] applied to the corresponding clean speech corpus. We have released a clean test corpus of AURORA2 at https://sites.google.com/site/zhangxiaolei321/vad where we can see that the automatic labels are accurate. We will further study how this automatic labeling method affects the performance in Section VI-C, compared to manual labeling. The frame length and frame shift of the proposed method were described in the MRCG feature. The frame length and frame shift of all competitive methods were 25 ms and 10 ms respectively.

We used 7 noisy test sets of AURORA2 [32] at SNR levels of $\{-5, 0, 5, 10, 15, 20\}$ dB, which had 42 noisy environments in total. The sampling rate is 8 kHz. We split each test corpus to three subsets for training, developing, and test, each of which contains 300, 300, and 401 utterances respectively. All utterances in each set were concatenated to a long conversation for simulating real working environments of VAD, such as phone calls.

We used the clean speech corpus of AURORA4 [33] corrupted by the 'babble" and "factory" noise in the NOISEX-92 noise corpus at extremely low SNR levels (i.e. $\{-5, 0, 5, 10\}$ dB) for a more broaden and harsh comparison between the proposed method and the competitors and for an investigation of the effectiveness of the components of the proposed method. That is to say, we constructed 8 difficult noisy speech corpora. The sampling rate is 16 kHz. The preprocessing is as follows. The clean speech corpus consists of 7,138 training utterances and 330 test utterances. We randomly selected 300 and 30 utterances from the training utterances as our training set and development set respectively, and used all 330 test utterances for testing. All utterances in each set were concatenated to a long conversation. We will also study how the proposed method behaves on individual utterances in Section VI-D. Note that for each noisy corpora, the additive noises for training, development, and test were cut from different intervals of a given noise.

2) Evaluation Metrics: Receiver operating characteristic (ROC) curve is considered as an overall metric of the VAD performance rather than the detection accuracy, since the speech-to-nonspeech ratio is usually imbalanced, and also since one usually tunes the decision threshold of VAD for specific applications. Due to the length limitation of the paper, we cannot draw all ROC curves. Because AUC can measure ROC curve quantitatively, we took AUC as the main metric. We also gave the *speech hit rate minus false alarm rate* (HIT - FA) at the optimal operating points of the ROC curves in Supplementary Material. Because over 70% frames are speech, we did not use detection accuracy as a metric, so as to prevent reporting misleading results caused by class imbalance.

3) Comparison Methods and Parameter Settings: We compared bDNN- and MRS-based VADs with Zhang13 VAD [21], and an SVM-based VAD using MRCG as the feature. The parameters of Zhang13 VAD were the same as in [21].

For bDNN-based VAD, the parameters were as follows. The numbers of hidden units were set to 512 for the two hidden layers. The number of epoches was set to 50. The batch size was set to 512. The scaling factor for the adaptive stochastic gradient descent was set to 0.0015, and the learning rate decreased linearly from 0.08 to 0.001. The momentum of the first 5 epoches was set to 0.5, and the momentum of other epoches was set to 0.9. The dropout rate of the hidden units was set to 0.2. The half-window size W was set to 19, and the parameter u of the window was set to 9.

For MRS-based VAD, we trained two building blocks (i.e. parameter $S = 2$). For the bottom one, we trained 10 bDNNs with resolution parameter (W, u) set to [(3,1), (5,2), (9,4), (13,6), (15,7), (17,8), (19,9), (21,10), (23,11), (25,12)] respectively. The parameter setting of each bDNN was exactly the same as that of the aforementioned bDNN-based VAD. For the top building block, we trained 1 bDNN with (W, u) set to (19,9). The parameter setting of the bDNN at the top building block was as follows. The numbers of hidden units were set to 128 for both the first and second hidden layers. The number of epoches was set to 7.

B. Results with Noise Dependent Training

Table I lists the AUC result of all 4 VAD methods on the 42 noisy environments of AURORA2. Table II lists the result on the 8 noisy environments of AURORA4. From the tables, we observe that (i) the proposed method significantly outperforms Zhang13 VAD and SVM-based VAD, particularly when the background is very noisy; (ii) the experimental phenomena of the proposed method on different noisy scenarios of AURORA2 and AURORA4 are quite consistent, which means its superiority is not affected by whether the spoken words were isolated or continuous.

Because MRS contains 11 bDNN models, the training and test time of the MRS-based VAD is about 11 times that of the bDNN-based VAD. Due to space limitation, we omit a detailed complexity analysis.

C. Effects of Boosted DNN and MRS on the Performance

To investigate how bDNN and MRS improve the performance, we ran DNN, bDNN, and MRS on the AURORA4 corpus with MRCG as the input feature, where the model "DNN" used the same input as bDNN, i.e. $[\mathbf{x}_{m-W}^T, \ldots, \mathbf{x}_m^T, \ldots, \mathbf{x}_{m+W}^T]^T$, but used y_m as the target instead of $[y_{m-W}, \ldots, y_m, \ldots, y_{m+W}]^T$.

Fig. 4 shows the comparison result with respect to the window length. From the figure, we observe that (i) bDNN and MRS significantly outperform DNN, and their superiority becomes more and more apparent when the window is gradually enlarged; (ii) MRS is less sensitive to the window length than bDNN; (iii) DNN can also benefit from the contextual

Authorized licensed use limited to: NATIONAL TAIWAN NORMAL UNIVERSITY. Downloaded on February 15,2020 at 05:29:19 UTC from IEEE Xplore. Restrictions apply.

TABLE I
AUC (%) COMPARISON BETWEEN THE COMPARISON VADS AND PROPOSED BDNN- AND MRS-BASED VADS ON THE AURORA2 CORPUS. THE NUMBERS IN BOLD INDICATE THE BEST RESULTS

Noise	SNR	SVM	Zhang13	bDNN	MRS
Babble	−5 dB	70.14	72.21	81.55	**82.51**
	0 dB	79.91	83.28	89.03	**89.85**
	5 dB	88.14	89.99	92.72	**92.93**
	10 dB	91.86	94.07	94.18	**94.84**
	15 dB	93.58	95.33	95.21	**95.36**
	20 dB	94.65	95.73	95.76	**95.85**
Car	−5 dB	82.18	82.76	91.34	**92.40**
	0 dB	89.50	91.64	94.87	**95.56**
	5 dB	93.59	93.96	95.60	**96.30**
	10 dB	95.01	95.69	96.30	**96.98**
	15 dB	96.05	96.45	96.97	**97.40**
	20 dB	96.78	97.33	97.16	**97.67**
Restaurant	−5 dB	72.01	74.20	82.40	**84.03**
	0 dB	81.11	81.14	88.07	**89.81**
	5 dB	89.25	91.01	93.13	**94.20**
	10 dB	91.78	93.25	94.80	**95.45**
	15 dB	93.43	94.61	95.60	**96.29**
	20 dB	94.92	95.57	96.13	**96.73**
Street	−5 dB	72.77	74.32	85.57	**86.31**
	0 dB	81.29	81.38	89.22	**90.23**
	5 dB	88.03	90.01	92.53	**93.06**
	10 dB	91.19	92.39	93.95	**94.21**
	15 dB	92.61	94.69	94.23	**94.66**
	20 dB	94.15	95.31	94.89	**95.14**
Airport	−5 dB	73.59	76.30	82.77	**85.02**
	0 dB	81.97	83.38	90.87	**92.00**
	5 dB	88.90	89.59	94.29	**95.22**
	10 dB	92.61	94.35	95.82	**96.39**
	15 dB	95.14	96.11	96.52	**96.96**
	20 dB	95.68	96.88	97.15	**97.53**
Train	−5 dB	74.43	76.55	85.55	**86.82**
	0 dB	83.21	84.89	90.49	**91.68**
	5 dB	89.72	91.12	93.54	**94.60**
	10 dB	92.07	93.29	94.43	**95.21**
	15 dB	94.10	95.13	95.28	**96.18**
	20 dB	94.84	95.34	95.64	**96.55**
Subway	−5 dB	83.36	84.29	92.37	**93.28**
	0 dB	89.22	90.11	93.90	**94.23**
	5 dB	91.86	93.09	95.18	**95.63**
	10 dB	93.32	93.97	95.34	**95.95**
	15 dB	94.22	94.84	95.68	**96.17**
	20 dB	94.81	95.38	95.92	**96.43**

TABLE II
AUC (%) COMPARISON BETWEEN THE COMPARISON VADS AND PROPOSED BDNN-BASED AND MRS-BASED VADS ON THE AURORA4 CORPUS. THE NUMBERS IN BOLD INDICATE THE BEST RESULTS

Noise	SNR	SVM	Zhang13	bDNN	MRS
Babble	−5 dB	81.05	82.84	85.75	**86.60**
	0 dB	86.06	88.33	89.62	**90.15**
	5 dB	90.49	91.61	92.75	**93.02**
	10 dB	91.05	93.01	93.81	**93.93**
Factory	−5 dB	78.63	81.81	85.78	**85.81**
	0 dB	86.05	88.39	90.64	**90.76**
	5 dB	89.10	91.72	92.82	**92.98**
	10 dB	92.21	93.13	93.64	**93.69**

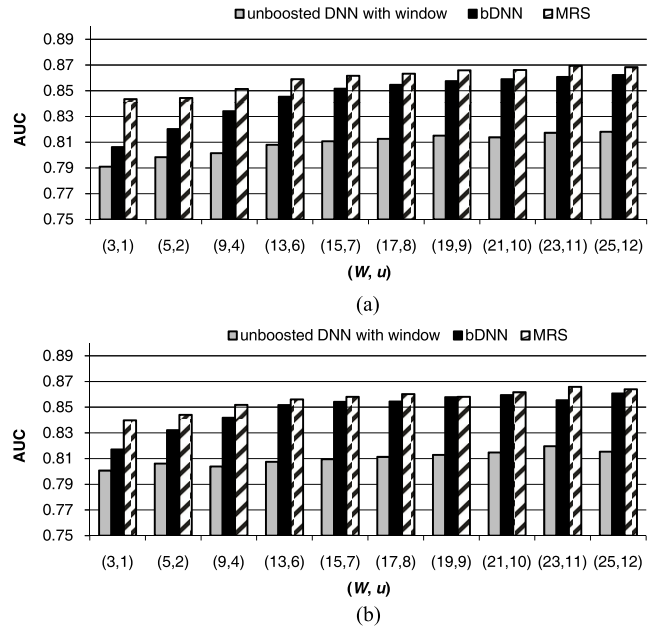

Fig. 4. AUC analysis of the advantage of the boosted algorithm in bDNN-based and MRS-based VADs over the unboosted counterpart that uses the same input $\mathbf{x}'_n$ as bDNN and MRS but uses the original output y_n as the training target instead of $\mathbf{y}'_n$. (A) Comparison in the babble noise environment with $\mathrm{SNR} = -5$ dB. (B) Comparison in the factory noise environment with $\mathrm{SNR} = -5$ dB. Note that (W, u) are two parameters of the window of bDNN.

TABLE III
AUC (%) COMPARISON BETWEEN MRCG AND COMB, WITH EITHER DNN, BDNN, OR MRS AS THE CLASSIFIER ON THE AURORA 4 CORPUS

Noise	SNR	DNN		bDNN		MRS	
		COMB	MRCG	COMB	MRCG	COMB	MRCG
Babble	−5 dB	81.53	81.54	84.62	85.75	86.11	86.60
	0 dB	85.48	86.48	88.84	89.62	89.76	90.15
	5 dB	89.08	90.05	92.11	92.75	92.82	93.02
	10 dB	90.56	91.64	93.10	93.81	93.65	93.93
Factory	−5 dB	80.16	79.70	83.51	85.78	85.75	85.81
	0 dB	84.59	86.51	88.95	90.64	90.35	90.76
	5 dB	87.79	89.76	91.91	92.82	92.70	92.98
	10 dB	89.16	90.95	92.79	93.64	93.75	93.69

information, but this performance gain is limited. Note that bDNN has the same computational complexity with DNN.

D. Effects of MRCG Feature on the Performance

To evaluate how MRCG affects the performance, we compared it with the combination (COMB) of 10 conventional acoustic features in Zhang13 VAD [21], on AURORA4 with either DNN, bDNN, or MRS as the classifier, where the model "DNN" is described in Section V-C.

Table III lists the comparison result between MRCG and COMB. From the table, we observe that the 96-dimensional MRCG is generally better than the 273-dimensional COMB feature. In our preliminary work [25], we have further enlarged the dimension of MRCG from 96 to 768. The comparison result in [25] shows that the 768-dimensional MRCG significantly outperforms COMB.

In [21], the authors have investigated the advantage of COMB over its sub-features including MFCC, DFT (with carefully selected bins suggested by the speech enhancement

Authorized licensed use limited to: NATIONAL TAIWAN NORMAL UNIVERSITY. Downloaded on February 15,2020 at 05:29:19 UTC from IEEE Xplore. Restrictions apply.

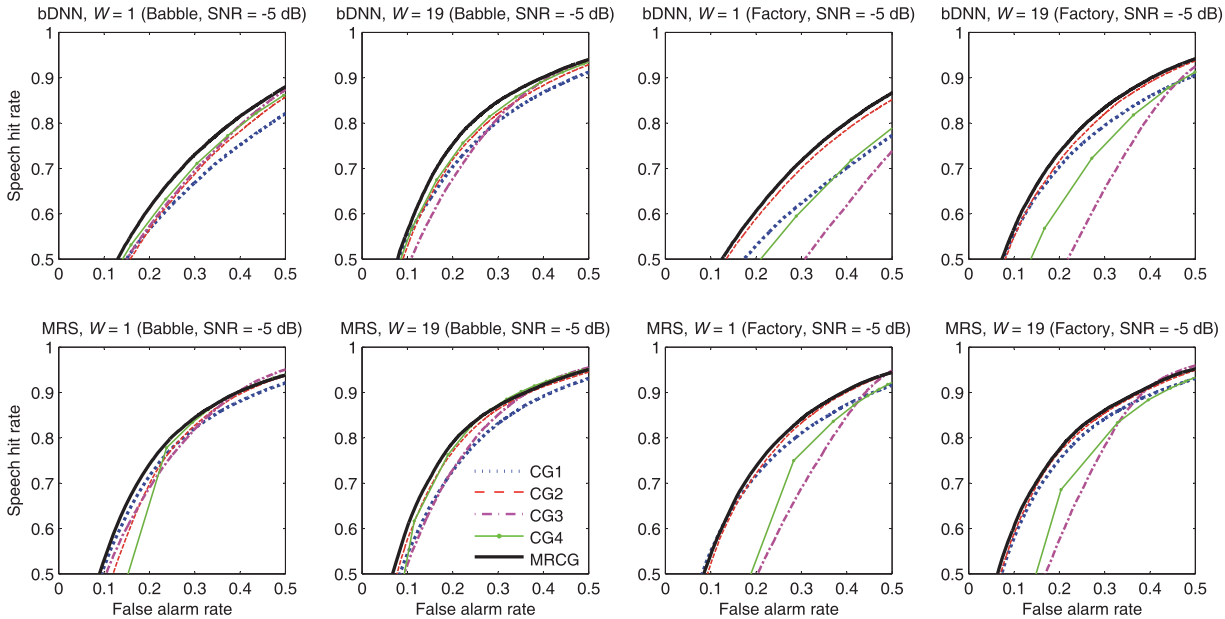

Fig. 5. ROC curve analysis of the MRCG feature versus its components at AURORA4.

standard IS-127), LPC and RASTA-PLP. The result shows that COMB is better than any of its sub-features. Given the advantage of MRCG over COMB, MRCG is better than the conventional features.

E. Advantage of MRCG Feature over Its Components

Fig. 5 shows the ROC curve comparison between the MRCG feature and its four components in the two difficult environments with parameters (W, u) set to (0,0) and (19,9), where $W = 0$ means that the input and output of bDNN do not use window. From the figure, we observe that (i) MRCG is at least as good as the best of its 4 components, which shows the effectiveness of the multi-resolution technique; (ii) CG2 yields a better ROC curve than the other 3 components. The same phenomena can also be observed when the dimension of MRCG is enlarged to 768 as shown in [25].

VI. Evaluation Results of Noise-Independent Models

The term *noise-independent* (NI) means that once trained, the machine learning based VADs can achieve reasonable performance in various noise scenarios, even though the noise scenarios are unseen from the training set. Training good NI models is one of the ultimate goals of machine learning based VADs in real-world applications and also one of the most difficult tasks that prohibit machine learning methods from practical use. In this section, we evaluate the performance of NI models in difficult and unseen test scenarios. We also report some results in Supplementary Material.

A. Experimental Settings

We randomly selected 300 clean utterances from AURORA2 and AURORA4 respectively as the clean corpora, which were also used as the clean corpora in Section V for synthesizing noisy speech corpora. We used a large-scale sound effect library[3] as our noise corpus, which contains over 20,000 sound effects. For constructing the noisy training corpus of AURORA2, we first randomly selected 4000 noise segments and concatenated them to a long noise wave which is about 35 hours long; then, we randomly picked clean utterances from the clean corpus of AURORA2 and added them one by one in time slot to the long noise wave with SNR levels varying in $\{-10, -9, -8, -7, -6, -4, -3, -2, -1, 1, 2, 3, 4, 6, 7, 8, 9, 11, 12\}$ dB, where repeated selection of the clean utterances was allowed. Note that when synthesizing each noisy speech segment in the long noisy speech wave, we fixed the clean utterance and rescaled the noise segment. For constructing the noisy test corpora of AURORA2, we used the same test noisy corpora as in Section V, which contain 28 noisy scenarios with SNR levels ranging in $\{-5, 0, 5, 10\}$ dB. We constructed the noisy training corpus of AURORA4 in the same way as that of AURORA2, and used the same noisy test corpora as in Section V for evaluating the NI models. From the above description, it is clear that the test noise scenarios are unseen in the training corpora.

We trained 1 DNN-, 1 bDNN-, and 1 MRS-based VAD on the noisy training corpus of AURORA2, and evaluated the 3 NI models on all 28 test corpora. We conducted an experiment on AURORA4 in the same way as that on AURORA2. The parameter settings of the DNN, bDNN, and MRS models were the same as their corresponding ND models in Section V, except that the batch size was set to 4096.

We compared with Sohn VAD [8], Ramirez05 VAD [13], and Ying VAD [9], which are noise-independent methods based on statistical signal processing. The parameters of the referenced methods were well tuned according to the authors.

B. Results with Noise-Independent Training

It was supposed that ND models, which were trained and tested in the same noise scenarios, might perform better than

[3]The library was requested from http://www.sound-ideas.com/sound-effects/series-6000-combo-sound-effects.html

Authorized licensed use limited to: NATIONAL TAIWAN NORMAL UNIVERSITY. Downloaded on February 15,2020 at 05:29:19 UTC from IEEE Xplore. Restrictions apply.

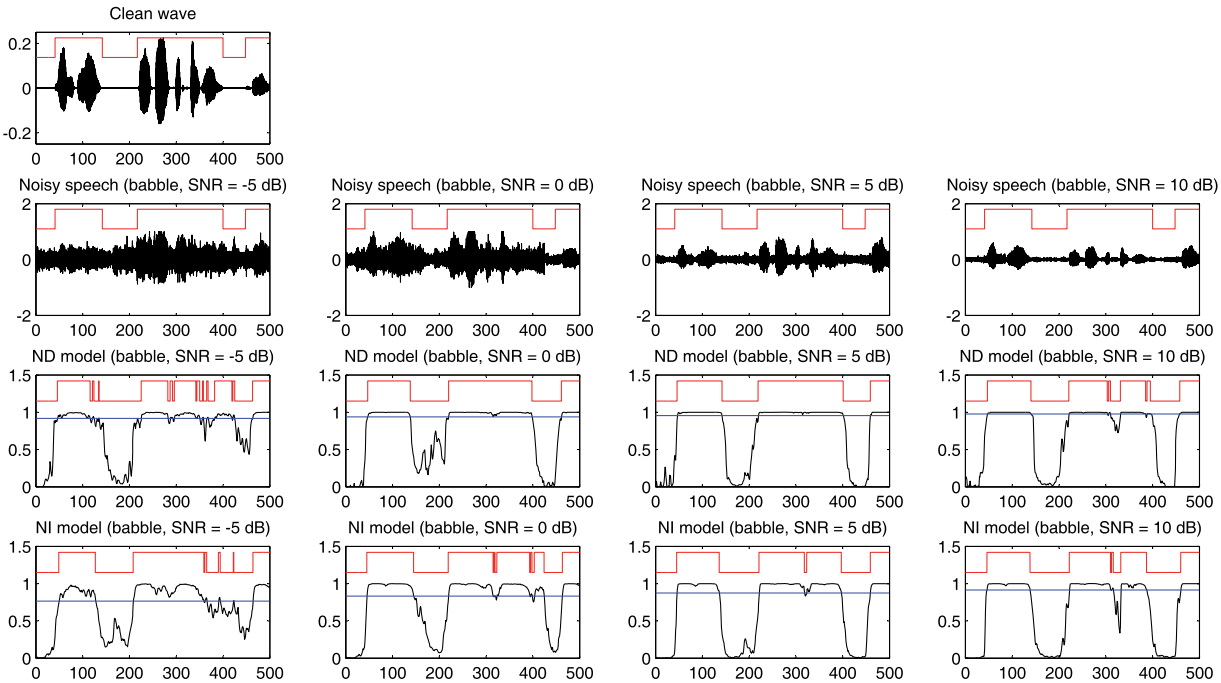

Fig. 6. Visualization of the output of noise-dependent (ND) MRS model and noise-independent (NI) MRS model in the babble noise environment at various SNR levels. Each test scenario of ND model is exactly the same as its training scenario. The test scenarios of NI model are unseen from its training corpus.

NI models. In this section, we investigated how much their performance differed. Table IV lists the comparison between the 3 referenced VADs and the NI and ND models of the DNN-, bDNN-, and MRS-based VADs on AURORA2. Table V lists the comparison on AURORA4. Fig. 6 shows a visualized comparison of the soft output of the NI model and 4 ND models on AURORA2. From the figure and tables, we observe that (i) the proposed VADs with the NI training are significantly better than Sohn VAD, Ramirez05 VAD, and Ying VAD which can be used in various noise scenarios without offline training; (ii) the performance of the NI models approaches to and even outperforms the performance of the ND models in most cases of AURORA2 when the SNR is equal or greater than 0 dB and in all cases of AURORA4; (iii) The NI models perform slightly worse than the ND models on AURORA2 when the SNR is extremely low, e.g. -5 dB; (iv) MRS-based VAD with the NI training outperforms bDNN-based VAD at extremely low SNR of AURORA2 and AURORA4, and performs similarly with the latter in other cases.

C. Comparison Between Automatic and Manual Labeling

The ground-truth labels of AURORA2 and AURORA4 were generated by applying Sohn VAD to clean utterances. This method seems sensible as the main objective is to evaluate the robustness of VAD to background noise, and it is commonly used for producing ground truths for pitch tracking in noisy speech (see e.g. [50]). To examine how this automatic labeling method affects the performance, compared to much less efficient manual labeling, we randomly selected 10 clean utterances from AURORA4 and labeled them manually (see Supplementary Material for detailed illustrations). The 10 utterances are 76.31 seconds long. The difference between the manual labels and automatic labels is 4.1%.

For each environment of AURORA4, we concatenated the noisy counterparts of the 10 clean utterances to a long conversation. We used the noise-independent models in Section VI-B to evaluate the conversation, and compared with the 5 VADs. The comparison results for the automatic labels and manual labels are summarized in Tables VI and VII respectively. From Table VI, we observe that the results on the 10 utterances are similar to those on the entire AURORA4 in Table V. Comparing Table VI and Table VII, we observe that the results given the automatic labels are broadly consistent with the results given the manual labels, which supports the validity of the results in Section VI-B and Section V.

To summarize, the labels generated by Sohn VAD on clean utterances can be reasonably used as ground-truth labels. Note that forced-alignment speech recognition [51] can be used to automatically generate reasonable VAD labels. Also, the Linguistic Data Consortium recently released the RATS Speech Activity Detection corpus that provides manually annotated labels for degraded speech signals.

D. Results on Short Utterances

All experiments so far were conducted on long conversations aiming to emulate phone calls. However, in many tasks such as the Google speech assistant, input utterances are very short. To study this case, we used the noise-independent models to evaluate 10 noisy speech utterances in Section VI-C individually, where each utterance is uttered by a single speaker and lasts about 5 to 10 seconds. All other settings were the same as in Section VI-C.

The comparison results given the automatic labels and manual labels are summarized in Tables VIII and IX respectively. Comparing Table VIII with Table VI, and Table IX with Table VII, we find that (i) the results on the individual utterances are consistent with those on the long conversations; (ii) the

Authorized licensed use limited to: NATIONAL TAIWAN NORMAL UNIVERSITY. Downloaded on February 15,2020 at 05:29:19 UTC from IEEE Xplore. Restrictions apply.

TABLE IV
AUC (%) COMPARISON BETWEEN THE NOISE-INDEPENDENT (NI) MODELS, NOISE-DEPENDENT (ND) MODELS, AND 3 REFERENCED VADS AT AURORA2. THE NUMBERS IN BOLD INDICATE THE BEST RESULTS AMONG SOHN VAD, RAMIREZ05 VAD, YING VAD, AND NI MODELS

Noise	**SNR**	Sohn	Ramirez05	Ying	DNN		bDNN		MRS	
					NI	ND	NI	ND	NI	ND
Babble	−5 dB	60.45	61.33	59.17	73.36	78.62	76.95	81.55	**77.92**	82.51
	0 dB	68.66	70.08	65.63	86.28	85.82	89.36	89.03	**90.16**	89.85
	5 dB	79.83	81.94	76.52	92.65	89.07	94.27	92.72	**94.32**	92.93
	10 dB	86.76	88.12	84.46	94.49	88.85	**95.88**	94.18	95.76	94.84
Car	−5 dB	59.03	60.62	61.75	86.36	89.75	88.71	91.34	**89.01**	92.40
	0 dB	69.05	72.00	69.27	92.72	93.74	**94.24**	94.87	94.06	95.56
	5 dB	79.83	82.22	78.53	94.66	94.53	**95.90**	95.60	95.86	96.30
	10 dB	87.22	88.64	84.73	95.33	95.35	**96.59**	96.30	96.48	96.98
Restaurant	−5 dB	55.62	55.04	57.67	71.16	78.15	73.33	82.40	**75.06**	84.03
	0 dB	66.00	64.52	62.98	83.49	86.57	86.27	88.07	**87.24**	89.81
	5 dB	72.24	73.65	71.54	91.49	92.23	93.41	93.13	**93.84**	94.20
	10 dB	79.85	80.85	79.29	94.10	93.96	95.31	94.80	**95.46**	95.45
Street	−5 dB	53.68	54.80	55.63	79.50	82.89	80.88	85.57	**81.34**	86.31
	0 dB	60.03	60.06	61.62	89.78	87.77	**91.36**	89.22	91.13	90.23
	5 dB	68.74	71.52	70.28	93.39	89.90	**94.56**	92.53	94.14	93.06
	10 dB	76.04	78.21	76.41	94.62	91.18	**95.84**	93.95	95.60	94.21
Airport	−5 dB	56.60	59.39	59.06	77.34	80.54	79.78	82.77	**81.04**	85.02
	0 dB	64.22	66.11	66.07	88.09	89.58	90.22	90.87	**90.38**	92.00
	5 dB	73.78	76.90	74.48	93.08	92.97	**94.63**	94.29	94.58	95.22
	10 dB	83.18	86.06	84.21	94.82	94.59	**96.24**	95.82	96.16	96.39
Train	−5 dB	55.31	57.68	61.35	80.23	84.18	82.98	85.55	**83.84**	86.82
	0 dB	60.04	63.19	67.85	89.20	88.89	**91.24**	90.49	91.14	91.68
	5 dB	73.00	77.26	77.58	93.32	92.20	94.88	93.54	**94.92**	94.60
	10 dB	83.76	84.51	82.18	94.05	92.80	**95.52**	94.43	95.39	95.21
Subway	−5 dB	55.42	55.00	57.74	72.64	91.35	74.47	92.37	**75.80**	93.28
	0 dB	62.66	61.63	62.75	85.10	93.43	86.78	93.90	**87.18**	94.23
	5 dB	70.49	76.50	68.35	91.57	94.39	92.76	95.18	**93.13**	95.63
	10 dB	79.02	81.18	76.73	93.61	95.07	94.61	95.34	**94.93**	95.95

TABLE V
AUC (%) COMPARISON BETWEEN THE NOISE-INDEPENDENT (NI) MODELS, NOISE-DEPENDENT (ND) MODELS, AND 3 REFERENCED VADS AT AURORA4. THE NUMBERS IN BOLD INDICATE THE BEST RESULTS AMONG SOHN VAD, RAMIREZ05 VAD, YING VAD, AND NI MODELS

Noise	**SNR**	Sohn	Ramirez05	Ying	DNN		bDNN		MRS	
					NI	ND	NI	ND	NI	ND
Babble	−5 dB	70.69	75.90	64.63	78.79	81.54	81.65	85.75	**84.09**	86.60
	0 dB	77.67	83.05	70.72	84.31	86.48	86.55	89.62	**88.32**	90.15
	5 dB	84.53	87.85	78.70	88.90	90.05	90.40	92.75	**91.19**	93.02
	10 dB	89.18	89.93	85.61	91.28	91.64	92.83	93.81	**93.19**	93.93
Factory	−5 dB	58.17	58.37	62.56	78.58	79.70	81.20	85.78	**83.40**	85.81
	0 dB	64.56	67.21	68.79	84.52	86.51	86.81	90.64	**88.34**	90.76
	5 dB	72.92	76.82	75.83	89.25	89.76	90.99	92.82	**91.46**	92.98
	10 dB	80.80	84.72	82.64	91.21	90.95	92.94	93.64	**93.16**	93.69

TABLE VI
AUC (%) COMPARISON ON 10 RANDOMLY SELECTED NOISY UTTERANCES OF AURORA4 THAT ARE AUTOMATICALLY LABELED. THE NUMBERS IN BOLD INDICATE THE BEST RESULTS

Noise	**SNR**	Sohn	Ramirez05	Ying	DNN	bDNN	MRS
Babble	−5 dB	69.00	77.26	60.32	79.22	81.70	**84.27**
	0 dB	80.43	85.63	72.31	83.50	86.38	**88.19**
	5 dB	84.89	90.42	77.51	89.60	91.61	**92.09**
	10 dB	92.19	93.88	85.70	93.59	95.49	**95.67**
Factory	−5 dB	60.67	62.75	59.35	79.84	82.05	**83.92**
	0 dB	64.50	69.17	68.04	83.93	86.50	**88.01**
	5 dB	74.27	82.66	78.22	91.61	93.20	**93.43**
	10 dB	84.05	90.48	85.03	94.34	95.91	**96.05**

TABLE VII
AUC (%) COMPARISON ON 10 RANDOMLY SELECTED NOISY UTTERANCES OF AURORA4 THAT ARE MANUALLY LABELED. THE NUMBERS IN BOLD INDICATE THE BEST RESULTS

Noise	**SNR**	Sohn	Ramirez05	Ying	DNN	bDNN	MRS
Babble	−5 dB	68.57	78.41	59.99	79.51	82.23	**85.12**
	0 dB	79.60	86.22	71.80	83.69	85.98	**88.08**
	5 dB	83.77	91.48	76.78	89.44	91.36	**91.88**
	10 dB	91.57	95.42	86.50	94.22	95.95	**96.20**
Factory	−5 dB	59.97	62.14	58.61	79.94	81.91	**84.11**
	0 dB	63.92	68.94	68.42	84.54	86.88	**88.61**
	5 dB	73.91	83.71	77.90	92.20	93.71	**93.96**
	10 dB	83.13	91.15	84.96	95.11	96.67	**96.80**

AUC scores of the DNN-, bDNN-, and MRS-based methods on the individual utterances are slightly better than those on the long conversations; (iii) the AUC scores of the three referenced statistical-signal-processing-based methods on the individual utterances are improved over those on the long conversations.

E. Results with Fixed Decision Threshold

All results so far are evaluated in terms of AUC and optimal HIT-FA with a tunable decision threshold. We further report the HIT and FA rates of NI models with a fixed decision threshold

Authorized licensed use limited to: NATIONAL TAIWAN NORMAL UNIVERSITY. Downloaded on February 15,2020 at 05:29:19 UTC from IEEE Xplore. Restrictions apply.

TABLE VIII
AUC (%) COMPARISON ON 10 RANDOMLY SELECTED NOISY UTTERANCES OF AURORA4 THAT ARE AUTOMATICALLY LABELED AND INDIVIDUALLY EVALUATED. THE NUMBERS IN BOLD INDICATE THE BEST RESULTS

Noise	SNR	Sohn	Ramirez05	Ying	DNN	bDNN	MRS
Babble	−5 dB	70.10	80.91	63.27	79.79	82.65	**85.40**
	0 dB	82.25	85.97	73.06	84.41	87.46	**89.28**
	5 dB	85.44	90.52	77.57	90.28	92.33	**93.03**
	10 dB	92.11	94.13	79.13	94.23	96.06	**96.30**
Factory	−5 dB	59.86	65.32	59.02	80.06	82.36	**84.24**
	0 dB	63.82	72.28	66.24	84.24	86.95	**88.39**
	5 dB	74.02	85.53	75.15	92.11	93.75	**93.94**
	10 dB	83.27	92.18	79.32	94.87	96.31	**96.57**

TABLE IX
AUC (%) COMPARISON ON 10 RANDOMLY SELECTED NOISY UTTERANCES OF AURORA4 THAT ARE MANUALLY LABELED AND INDIVIDUALLY EVALUATED. THE NUMBERS IN BOLD INDICATE THE BEST RESULTS

Noise	SNR	Sohn	Ramirez05	Ying	DNN	bDNN	MRS
Babble	−5 dB	69.37	81.07	62.23	79.55	82.65	**85.68**
	0 dB	81.15	86.55	71.90	84.54	87.02	**89.13**
	5 dB	84.30	91.48	77.01	90.21	92.26	**92.88**
	10 dB	91.48	95.53	79.35	94.66	96.35	**96.63**
Factory	−5 dB	59.21	64.65	58.36	80.13	82.16	**84.36**
	0 dB	63.06	72.06	66.06	85.23	87.55	**89.09**
	5 dB	73.39	86.39	74.71	92.57	94.16	**94.34**
	10 dB	82.46	92.76	79.00	95.43	96.82	**97.03**

$\delta = 0.8$ in all conditions in Tables VI and VII of Supplementary Material. From the tables, we observe that the results with this fixed threshold are close to the optimal results with a tunable threshold. Note that the threshold may be tuned for different applications to obtain better results.

VII. CONCLUDING REMARKS

In this paper, we have proposed a supervised VAD method, named MRS-based VAD, using a new base classifier—bDNN—and a newly introduced acoustic feature—MRCG. The proposed method explores contextual information heavily in three levels. At the top level, MRS is a stack of ensemble classifiers. The classifiers in a building block explore context in different resolutions and output different predictions which are further integrated in their upper building block. At the middle level, bDNN is a strong DNN classifier that first produces multiple base predictions on a single frame by boosting the contextual information encoded in a given resolution and then aggregates the base predictions for a stronger one. At the bottom level, MRCG consists of cochleagram features at multiple spectrotemporal resolutions. Experimental results on AURORA2 and AURORA4 have shown that when the noise scenarios of training and test are matching, the proposed method outperforms the referenced VADs by a considerable margin, particularly at low SNRs. Our further analysis shows that (i) both bDNN and MRS contribute to the improvement; (ii) the 96-dimensional MRCG feature is comparable to the 273-dimensional COMB feature. Moreover, when trained with a large number of noise scenarios and a wide range of SNR levels, the proposed method performs as good as the method with noise-dependent training, which is a promising sign for the practical use of machine-learning-based VADs in real-world environments.

The following topics are worth further investigation in the future. (i) The framework of MRS is not limited to ensemble learning. We may train all DNNs in MRS jointly. (ii) The performance of bDNN may be further improved by using other types of windows that incorporate neighboring labels into the training target. (iii) As manual annotation for VAD is an expensive task, how to produce accurate ground-truth VAD labels automatically is an important topic.

ACKNOWLEDGMENT

We thank the associate editor and anonymous reviewers for their helpful comments. We also thank Yuxuan Wang for providing his DNN code, Jitong Chen for providing the MRCG code and the large-scale sound effect library, Arun Narayanan for helping with AURORA4, and the Ohio Supercomputing Center for providing computing resources. Part of the work was conducted when the authors were visiting the Center of Intelligent Acoustics and Immersive Communications, Northwestern Polytechnical University, China.

REFERENCES

[1] A. Benyassine, E. Shlomot, H. Y. Su, D. Massaloux, C. Lamblin, and J. P. Petit, "ITU-T recommendation G. 729 Annex B: A silence compression scheme for use with G. 729 optimized for V. 70 digital simultaneous voice and data applications," *IEEE Commun. Mag.*, vol. 35, no. 9, pp. 64–73, Sep. 1997.
[2] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-27, no. 2, pp. 113–120, Apr. 1979.
[3] ETSI, "Speech processing, transmission and quality aspects (STQ); distributed speech recognition; advanced front-end feature extraction algorithm; compression algorithms," ETSI ES vol. 202, no. 050.
[4] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 30–42, Jan. 2012.
[5] E. Nemer, R. Goubran, and S. Mahmoud, "Robust voice activity detection using higher-order statistics in the LPC residual domain," *IEEE Trans., Speech, Audio Process.*, vol. 9, no. 3, pp. 217–231, Mar. 2001.
[6] I. V. McLoughlin, "The use of low-frequency ultrasound for voice activity detection," in *Proc. Interspeech*, 2014, pp. 1553–1557.
[7] G. Aneeja and B. Yegnanarayana, "Single frequency filtering approach for discriminating speech and nonspeech," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 23, no. 4, pp. 705–717, Apr. 2015.
[8] J. Sohn, N. S. Kim, and W. Sung, "A statistical model-based voice activity detection," *IEEE Signal Process. Lett.*, vol. 6, no. 1, pp. 1–3, Jan. 1999.
[9] D. Ying, Y. Yan, J. Dang, and F. Soong, "Voice activity detection based on an unsupervised learning framework," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 8, pp. 2624–2644, Nov. 2011.
[10] S. Gazor and W. Zhang, "A soft voice activity detector based on a Laplacian-Gaussian model," *IEEE Trans. Speech Audio Process.*, vol. 11, no. 5, pp. 498–505, Sep. 2003.
[11] J. H. Chang, N. S. Kim, and S. K. Mitra, "Voice activity detection based on multiple statistical models," *IEEE Trans. Signal Process.*, vol. 54, no. 6, pp. 1965–1976, Jun. 2006.
[12] Y. Ephraim and D. Malah, "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," *IEEE Trans. Audio, Speech, Signal Process.*, vol. 32, no. 6, pp. 1109–1121, Dec. 1984.
[13] J. Ramírez, J. C. Segura, C. Benítez, L. García, and A. Rubio, "Statistical voice activity detection using a multiple observation likelihood ratio test," *IEEE Signal Process. Lett.*, vol. 12, no. 10, pp. 689–692, Oct. 2005.
[14] S. O. Sadjadi and J. H. L. Hansen, "Unsupervised speech activity detection using voicing measures and perceptual spectral flux," *IEEE Signal Process. Lett.*, vol. 20, no. 3, pp. 197–200, Mar. 2013.

Authorized licensed use limited to: NATIONAL TAIWAN NORMAL UNIVERSITY. Downloaded on February 15,2020 at 05:29:19 UTC from IEEE Xplore. Restrictions apply.

[15] P. Teng and Y. Jia, "Voice activity detection via noise reducing using non-negative sparse coding," *IEEE Signal Process. Lett.*, vol. 20, no. 5, pp. 475–478, May 2013.
[16] S. Mousazadeh and I. Cohen, "Voice activity detection in presence of transient noise using spectral clustering," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 6, pp. 1261–1271, Jun. 2013.
[17] J. Gorriz, J. Ramirez, E. Lang, and C. Puntonet, "Hard c-means clustering for voice activity detection," *Speech Commun.*, vol. 48, no. 12, pp. 1638–1649, 2005.
[18] T. Ng, B. Zhang, L. Nguyen, S. Matsoukas, X. Zhou, N. Mesgarani, K. Veselý, and P. Matejka, "Developing a speech activity detection system for the DARPA RATS program," in *Proc. Interspeech*, 2012, pp. 1969–1972.
[19] D. Enqing, L. Guizhong, Z. Yatong, and Z. Xiaodi, "Applying support vector machines to voice activity detection," in *Proc. Int. Conf. Signal Process.*, 2002, vol. 2, pp. 1124–1127.
[20] J. W. Shin, J. H. Chang, and N. S. Kim, "Voice activity detection based on statistical models and machine learning approaches," *Comput. Speech Lang.*, vol. 24, no. 3, pp. 515–530, 2010.
[21] X.-L. Zhang and J. Wu, "Deep belief networks based voice activity detection," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 4, pp. 697–710, Apr. 2013.
[22] T. Hughes and K. Mierle, "Recurrent neural networks for voice activity detection," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 7378–7382.
[23] F. Eyben, F. Weninger, S. Squartini, and B. Schuller, "Real-life voice activity detection with lstm recurrent neural networks and an application to Hollywood movies," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process*, 2013, pp. 483–487.
[24] G. Saon, S. Thomas, H. Soltau, S. Ganapathy, and B. Kingsbury, "The IBM speech activity detection system for the DARPA RATS program," in *Proc. Interspeech*, 2013, pp. 3497–3501.
[25] X.-L. Zhang and D. L. Wang, "Boosted deep neural networks and multi-resolution cochleagram features for voice activity detection," in *Proc. Interspeech*, 2014, pp. 1534–1538.
[26] S. Thomas, G. Saon, M. Van Segbroeck, and S. S. Narayanan, "Improvements to the IBM speech activity detection system for the DARPA RATS program," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process*, 2015, pp. 4500–4504.
[27] X.-L. Zhang, "Unsupervised domain adaptation for deep neural network based voice activity detection," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, 2014, pp. 6864–6868.
[28] K. Walker and S. Strassel, "The RATS radio traffic collection system," in *Proc. ISCA Odyssey*, 2012, pp. 291–297.
[29] Y. Wang, J. Chen, and D. L. Wang, "Deep neural network based supervised speech segregation generalizes to novel noises through large-scale training," Dept. of Comput. Sci. and Eng., The Ohio State Univ., Columbus, OH, USA, Tech. Rep. OSU-CISRC-3/15-TR02, 2015.
[30] Y. Wang and D. L. Wang, "Towards scaling up classification-based speech separation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 7, pp. 1381–1390, Jul. 2013.
[31] J. Chen, Y. Wang, and D. L. Wang, "A feature study for classification-based speech separation at very low signal-to-noise ratio," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 12, pp. 1993–2002, Dec. 2014.
[32] D. Pearce and H. Hirsch *et al.*, "The AURORA experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *Proc. ICSLP'00*, 2000, vol. 4, pp. 29–32.
[33] D. Pearce and J. Picone, "Aurora working group: DSR front end LVCSR evaluation AU/384/02," Inst. for Signal & Inf. Process., Mississippi State Univ., Tech. Rep., 2002.
[34] X.-L. Zhang and D. L. Wang, Boosting contextual information for deep neural network based voice activity detection Dept. of Comput. Sci. Eng., The Ohio State Univ., Columbus, OH, USA, Tech. Rep. OSU-CISRC-5/15-TR06, 2015, Tech. Rep..
[35] T. G. Dietterich, "Ensemble methods in machine learning," in *Multiple Classifier Sys.*, G. Goos, J. Hartmanis, and J. van Leeuwen, Eds. New York, NY, USA: Springer, 2000, pp. 1–15.
[36] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *Proc. Int. Conf. Comput. Learn. Theory*, 1995, pp. 23–37.
[37] R. E. Schapire, "The strength of weak learnability," *Mach. Learn.*, vol. 5, no. 2, pp. 197–227, 1990.
[38] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[39] S. Nie, H. Zhang, X. Zhang, and W. Liu, "Deep stacking networks with time series for speech separation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2014, pp. 6667–6671.
[40] D. Yu, L. Deng, and F. Seide, "The deep tensor neural network with applications to large vocabulary speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 2, pp. 388–396, Feb. 2013.
[41] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[42] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, M. Z. Mao, M. Ranzato, A. W. Senior, and P. A. Tucker *et al.*, "Large scale distributed deep networks," in *Adv. Neural Inf. Process. Syst.*, 2012, pp. 1232–1240.
[43] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1–8.
[44] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.
[45] G. Hu and D. L. Wang, "Auditory segmentation based on onset and offset analysis," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 2, pp. 396–405, Feb. 2007.
[46] S. K. Nemala, K. Patil, and M. Elhilali, "A multistream feature framework based on bandpass modulation filtering for robust speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 2, pp. 416–426, Feb. 2013.
[47] , D. L. Wang and G. J. Brown, Eds., *Computational Auditory Scene Analysis: Principles, Algorithms and Applications*. New York, NY, USA: Wiley-IEEE Press, 2006.
[48] J.-H. Choi and J.-H. Chang, "Dual-microphone voice activity detection technique based on two-step power level difference ratio," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 6, pp. 1069–1081, Jun. 2014.
[49] The Rice University, "Noisex-92 database," [Online]. Available: http://spib.rice.edu/spib
[50] K. Han and D. Wang, "Neural network based pitch tracking in very noisy speech," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 12, pp. 2158–2168, Dec. 2014.
[51] Z.-H. Tan and B. Lindberg, "Low-complexity variable frame rate analysis for speech recognition and voice activity detection," *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 5, pp. 798–807, Oct. 2010.

Xiao-Lei Zhang (S'08–M'12) received the Ph.D. degree from the Information and Communication Engineering, Tsinghua University, Beijing, China, in 2012. He is currently a Postdoctoral Researcher with the Department of Computer Science and Engineering, The Ohio State University, Columbus, OH. He was a visitor of the Perception and Neurodynamics Lab at The Ohio State University, and a visitor of the Center of Intelligent Acoustics and Immersive Communications, Northwestern Polytechnical University, China, since 2013. His research interests are the topics in audio signal processing, machine learning, statistical signal processing, and artificial intelligence. He has published over 20 peer-reviewed articles in journals and conference proceedings including IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, IEEE SIGNAL PROCESSING LETTERS, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, ICASSP, and Interspeech. He was a recipient of the first-class Beijing Science and Technology Award and the first-class Scholarship of Tsinghua University. He is a member of IEEE SPS and ISCA.

DeLiang Wang (F'04) photograph and biography not provided at the time of publication.

Authorized licensed use limited to: NATIONAL TAIWAN NORMAL UNIVERSITY. Downloaded on February 15,2020 at 05:29:19 UTC from IEEE Xplore. Restrictions apply.