Hindawi
IET Biometrics
Volume 2024, Article ID 6694481, 10 pages
https://doi.org/10.1049/2024/6694481

Research Article

Learning Deep Embedding with Acoustic and Phoneme Features for Speaker Recognition in FM Broadcasting

Xiao Li,[1,2] **Xiao Chen**,[1] **Rui Fu**,[2,3] **Xiao Hu**,[2] **Mintong Chen**,[1] **and Kun Niu**[1]

[1] *School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2] *Academy of Broadcasting Science, National Radio and Television Administration, Beijing 100866, China*
[3] *School of Information and Communication Engineering, Communication University of China, Beijing 100024, China*

Correspondence should be addressed to Xiao Li; lixiao1911@bupt.edu.cn and Kun Niu; niukun@bupt.edu.cn

Received 12 July 2023; Revised 28 February 2024; Accepted 1 March 2024; Published 22 March 2024

Academic Editor: Ahilan Kanagasundaram

Text-independent speaker verification (TI-SV) is a crucial task in speaker recognition, as it involves verifying an individual's claimed identity from speech of arbitrary content without any human intervention. The target for TI-SV is to design a discriminative network to learn deep speaker embedding for speaker idiosyncrasy. In this paper, we propose a deep speaker embedding learning approach of a hybrid deep neural network (DNN) for TI-SV in FM broadcasting. Not only acoustic features are utilized, but also phoneme features are introduced as prior knowledge to collectively learn deep speaker embedding. The hybrid DNN consists of a convolutional neural network architecture for generating acoustic features and a multilayer perceptron architecture for extracting phoneme features sequentially, which represent significant pronunciation attributes. The extracted acoustic and phoneme features are concatenated to form deep embedding descriptors for speaker identity. The hybrid DNN demonstrates not only the complementarity between acoustic and phoneme features but also the temporality of phoneme features in a sequence. Our experiments show that the hybrid DNN outperforms existing methods and delivers a remarkable performance in FM broadcasting TI-SV.

1. Introduction

Speaker recognition is a process that involves identifying individuals from their speech segments without any human intervention. The field of speaker recognition can be categorized into two main tasks: speaker identification (SI) and speaker verification (SV). SI is focused on identifying the speaker's identity, while SV seeks to authenticate whether the speaker is the target individual. As the ubiquity of smart devices increases, SV has become a crucial technology in many applications, such as identity verification [1], criminal investigation [2], and financial services [3]. For example, in terms of identity verification, SV can be applied to personal smart devices such as mobile phones, vehicles, and laptops, ensuring the security of bank transactions and remote payments. SV can be divided into two categories [4]: text-dependent SV (TD-SV) and text-independent SV (TI-SV). TD-SV restrains speech content, e.g., Google devices adopt the fixed "ok, google" as a voice password, and TI-SV does not have such restriction. As speech content is not taken into account, the variation of speech in TI-SV is much larger than that in TD-SV, making TI-SV a challenging task. In general, the TI-SV can be categorized into two-stage and end-to-end implementations. The two-stage TI-SV systems consist of a front-end for extracting embedding descriptors and a back-end for calculating the similarity score between a pair of embedding descriptors, while the end-to-end TI-SV systems combine the two ends together and calculate the similarity score directly for two embedding descriptors. In this paper, we focus on the deep speaker embedding learning for the two-stage TI-SV.

Traditional TI-SV approaches utilize an unsupervised Gaussian mixture model-universal background model (GMM-UBM) framework [5] since 2000, and GMM-UBM-based *i-vector* [6] demonstrates effectiveness as well for TI-SV. Though the aforementioned approaches have proven to be effective, the main issue is the drawback of unsupervised approaches, where models are not necessarily supervised by speaker discriminative features. Several supervised GMM-UBM-based TI-SV approaches have been developed, e.g., the approach [7] of

GMM-UBM-based *i-vector* front-end with probabilistic linear discriminant analysis back-end is superior for TI-SV. The supervised approaches have been developed as discriminative models for supervising the generative frameworks and demonstrated promising results. However, these GMM-UBM-based *i-vector* approaches suffer from sensitivity to lexical variability for short speech utterances. Due to the remarkable success of deep neural network (DNN) [8–10] and the easy availability of large corpus [11–13], recent TI-SV studies have shifted from GMM-UBM-based *i-vector* systems towards DNN-based paradigms, in which long short-term memory network (LSTM), convolution neural network (CNN), as well as time-delay neural network (TDNN) are employed as front-end for learning deep speaker embedding from acoustic features of speech segments. The DNNs consider and process the input acoustic features as a grayscale image, which has three dimensions for frequency, time, and channel, respectively. As the acoustic features propagate on the front-end DNNs, speaker identity-related information is extracted layer by layer. The DNN-based front-end TI-SV paradigms can be categorized into frame-level and utterance-level methods for processing acoustic features. Frame-level methods adopt DNNs for extracting deep speaker embedding from acoustic features of a frame to represent the speaker, and *d-vector* [14] is a typical frame-level approach. Differently, utterance-level methods utilize DNNs for learning deep speaker embedding from acoustic features of an utterance to represent the speaker, in which temporal average pooling or one of other advanced pooling is introduced for feature aggregation. *x-vector* [15] and *r-vector* [16] are two famous utterance-level methods, which outperform both *i-vector* and *d-vector*, making utterance-level methods the mainstream in TI-SV.

In the past years, phoneme has been successfully applied in speech recognition (speech-to-text), and phoneme features are abstract classes describing the movements or positions of different phonetic units during speech production. The features are rarely investigated in deep speaker embedding learning for TI-SV. Intuitively, phoneme is regarded as a nuisance for TI-SV since the speaker embedding should be independent for speech content. However, phoneme features have been tried to be integrated into embedding learning for performance boost, e.g., it is shown that the performance of the *x-vector*-based system [17] composed of speaker classification and phoneme unit recognition is improved significantly. The counter-intuitive result is that the multitask learning mechanism can instead focus on the specific phonemes that contain rich speaker information, improving the discriminability of the speaker embedding descriptors for speaker identity. Related works [18, 19] also prove that in multitask models, it is effective to adopt phoneme unit recognition as an auxiliary task to learn deep speaker embedding descriptors for TI-SV. A reasonable explanation for this situation is that phoneme unit recognition in TI-SV is beneficial to capture the important speaker pronunciation attributes, which may contain discriminative phonetic units for encouraging interclass separability and intraclass compactness in a series of speakers. These above methods treat phoneme feature learning as an auxiliary task for multitask networks; that is, the extracted deep speaker embedding descriptors are still extracted only from acoustic information, which motivates us that it may be meaningful to introduce phoneme information as prior rather than posterior knowledge into deep speaker embedding learning.

In this paper, we propose a deep speaker embedding learning approach, which integrates both acoustic and phoneme information for speaker identity. Introducing phoneme is helpful for presenting speaker pronunciation attributes to improve TI-SV performance. First, a proposed CNN is utilized as UBM to process acoustic features. Second, a multilayer perceptron (MLP) structure is constructed as phoneme feature extraction (PFE) to extract phoneme features. Third, the extracted acoustic and phoneme features are aggregated into embedding descriptors for speaker idiosyncrasy. Open-source English corpus Voxceleb1 and Mandarin corpus Aishell1 are, respectively, employed to evaluate our hybrid network. The two corpora are well-known and large-scale, widely used in TI-SV studies. Self-collected Mandarin corpus *FMAudio_v1* is used to further evaluate the network performance in an FM broadcasting environment. Experiment results indicate the learned deep speaker embedding descriptors are more centralized to the corresponding speakers. Our contributions in this paper are summarized as follows:

(1) A deep speaker embedding learning approach using phoneme information as prior compensation is proposed, which can aggregate phoneme into acoustic features to produce discriminative deep speaker embedding at the utterance level.

(2) An audio dataset called *FMAudio_v1* is created and open-sourced, and its audio is collected from FM broadcasting. In this paper, we utilize the dataset to evaluate the performance of the proposed approach in noised signal.

(3) Multiple losses are explored as target functions. Experimental results on three corpora show that the proposed approach is benefit to boost TI-SV performance. The superiority of our hybrid DNN with two subnets over existing networks is verified.

2. Related Works

Traditional TI-SV works adopt hand-crafted features to represent speaker's time–frequency properties, such as GMM-UBM-based *i-vector* [6, 7], eigenvoice-motivated vectors [20], and Mel-frequency cepstral coefficients [21]. These hand-crafted features are shallow-model-based features that cannot deeply represent the differences in characteristics for speaker identity. Simultaneously, these features are designed for specific situations, so they lack generalization ability when using them in other conditions. To overcome the deficiencies of these hand-crafted features, some recent studies learn deep-model-based features from DNNs. Existing DNN-based TI-SV approaches are built on several architectures, such as LSTM [22, 23], CNN [11, 16, 24], as well as TDNN [15, 25], to extract utterance-level embedding descriptors from speech signals. For example, Snyder et al. [15] exploit a TDNN to extract utterance-level speaker embedding, and the embedding is known as *x-vector*, which is the state-of-the-art for TI-SV. Zeinali et al. [16] proposed a CNN-based TI-SV model to extract utterance-level speaker embedding named *r-vector*, which has been proven to have superior performance compared to *x-vector*. In 2021, VoxCeleb Speaker Recognition Challenge, systems with CNN architecture performed excellently, proving that CNN-based approaches have great potential for TI-SV. In order to aggregate frame-level features to utterance-level features, a series of pooling mechanisms, i.e., average [11, 26], statistics [15, 27], dictionary [28], as well as attention pooling [29, 30], are employed to highlight important frames or other components when aggregate. In the process, irrelevant information is gradually eliminated when acoustic features propagate on DNNs.

Recently, phoneme classification has been explored as an additional task for multitask training [17–19, 31–34] to learn deep speaker embedding, making phoneme information be a posteriori probability problem similar to acoustic information for speaker recognition. The target of phoneme classification is to extract a phonetic vector from a DNN architecture, which is shared for multitasking speaker and speech unit recognition tasks. In [31], an LSTM network is used for speaker recognition and speech unit classification simultaneously. In [32], a TDNN architecture is adopted as a shared frame-level network of the speaker embedding and speech unit learning. Related works [33, 34] also prove that the success of one task improves the performance of the other task, and multitask learning is less susceptible to the problem of overfitting compared to single-task learning. Thus, multitask learning is the mainstream to introduce phoneme information into deep speaker embedding extraction. On the other hand, existing studies in the field of speech-to-text [35, 36] have introduced speaker features as prior knowledge, which helps improve performance significantly. Similarly, the performance of TD-SV is consistently better than that of TI-SV, proving that certain prior knowledge can be a supplementary condition to optimize embedding learning. In [37], an MLP-based phoneme feature extractor with 120 output states is proposed, where three output states represent one phoneme units with a total of 39 phoneme units and one silence. In [38], a CNN-based phoneme feature extractor is proposed, and it consists of an encoder and context subnets for extracting contextual phoneme feature representations. These pre-trained single networks can be used for providing prior knowledge. Inspired by these studies, borrowing phoneme features as a prior knowledge rather than a posteriori probability problem may make a certain sense for deep speaker embedding learning.

Another trend toward learning discriminative deep speaker embedding is to reinforce the DNNs with powerful loss functions. Softmax loss [11, 27] or one of its variants, e.g., angular Softmax (A-Softmax) [28], additive margin Softmax (AM-Softmax) [27], and additive angular margin Softmax (AAM-Softmax) [26, 39], has been employed intensively for TI-SV. In some cases, a contrastive loss or one of the metric losses, triplet loss, is utilized for a further performance boost.

3. Learning Framework

3.1. Description. For most existing DNN-based TI-SV, deep speaker embedding extraction is single-task learning. The process can be regarded as predicting the posterior probability $P(t|x)$ of target function t based on input signals x. In this paper, we borrow phoneme features as supplementary condition s into TI-SV, so the posterior probability $P(t|x)$ calculation is converted into a marginal probability calculation, as formulated:

$$P(t|x) = \sum_{s} P(t|x, s) P(s|x), \tag{1}$$

where $P(s|x)$ denotes the posterior probability of predicted target s under given x, which can be treated as a probability distribution of input x on different phonetic units. Further, $P(s|x)$ is a prior knowledge irrelevant to target function t, and it can be learned in advance to reduce the impact of itself.

3.2. Proposed Hybrid DNN Architecture. In this paper, we propose a hybrid DNN for TI-SV, which is composed of a CNN-based UBM and an MLP-based PFE, to introduce phoneme features as prior knowledge into acoustic features so the acoustic-based and phoneme-aware deep speaker embedding descriptors can be learned. Figure 1 shows the proposed hybrid DNN architecture.

3.2.1. Universal Background Model. Our previously proposed CNN architecture [39] is modified to be UBM for acoustic feature extraction, and Figure 1 depicts the architecture. The CNN architecture has a residual structure similar to that of *r-vector* [16], but the benefits of its deeper depth and adaptive weight pooling (AWP) are witnessed in the evaluation [39]. The CNN architecture is modified from ResNet-50 to a full convolution mode and cuts down the number of channels in each convolution layer for reducing trainable parameters to 1.5 million, compared with 25 million of the original ResNet-50.

In Table 1, ReLU and BatchNorm layers are omitted. Input acoustic signals $\boldsymbol{x}$ with size $R^{1\times40\times T}$ are encoded into a group of deep acoustic feature maps $\boldsymbol{M}_x$ with size $R^{512\times5\times(T/8)}$ for filtering out irrelevant information, making the deep acoustic feature maps highly relevant for speaker identity, where T denotes sample time length. Subsequently, the deep acoustic feature maps are partitioned into three local parts on the frequency axis for representing high, middle, and low-frequency components, each with the size of $R^{512\times2\times(T/8)}$, $R^{512\times1\times(T/8)}$, and $R^{512\times2\times(T/8)}$, i.e., $\boldsymbol{M}_x^{l_i}$, where $i = 1, 2, 3$. We employ global average pooling (GAP) to map $\boldsymbol{M}_x$ to global acoustic embedding descriptor $\boldsymbol{F}_x^g$, map $\boldsymbol{M}_x^l$ to local acoustic embedding descriptor $\boldsymbol{F}_x^l$. After which $\boldsymbol{F}_x^l$ is weighted adaptively by AWP to highlight key local patterns and suppress inessential ones for discriminative deep speaker acoustic embedding descriptor. The AWP is defined as follows:

$$\boldsymbol{F}_x = \left[\boldsymbol{F}_x^g, \left(\boldsymbol{F}_x^l \odot \boldsymbol{W} + \boldsymbol{B}\right)\right], \tag{2}$$

where $\boldsymbol{W}$ and $\boldsymbol{B}$ are weight and bias vectors, which have the same dimension with $\boldsymbol{F}_x^l$, and $\odot$ denotes the Hadamard

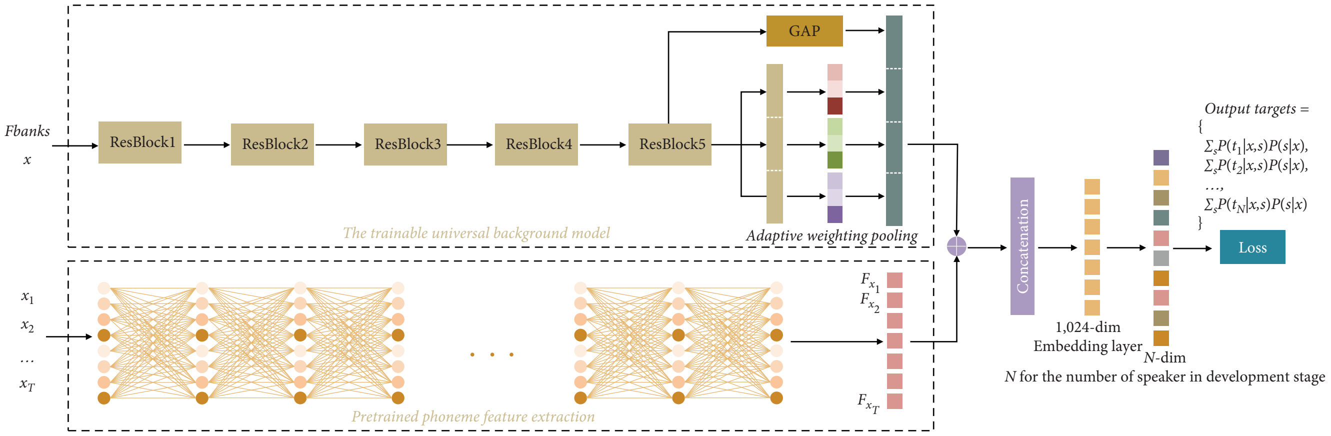
Fbanks
x
ResBlock1
ResBlock2
ResBlock3
ResBlock4
ResBlock5
GAP
The trainable universal background model
Adaptive weighting pooling
x_1
x_2
…
x_T
Pretrained phoneme feature extraction
F_{x_1}
F_{x_2}
F_{x_T}
Concatenation
1,024-dim
Embedding layer
N-dim
N for the number of speaker in development stage
Output targets =
{
$\Sigma_s P(t_1|x,s)P(s|x)$,
$\Sigma_s P(t_2|x,s)P(s|x)$,
…,
$\Sigma_s P(t_N|x,s)P(s|x)$
}
Loss

Figure 1: The architecture of the proposed hybrid network. It consists of two subnets: universal background model (UBM) and phoneme feature extraction (PFE). GAP refers to global average pooling.

Table 1: The universal background model based on CNN architecture.

Layer type	Operation	Data sizes
Input	—	$1 \times 40 \times T$
Convolution	conv2d, 7×7, 16	$16 \times 40 \times T$
Residual block	$\begin{bmatrix} \text{conv}, & 1 \times 1, 16, & \text{stride}(1,1) \\ \text{conv}, & 3 \times 3, 16, & \text{stride}(1,1) \\ \text{conv}, & 1 \times 1, 64, & \text{stride}(1,1) \end{bmatrix} \times 3$	$64 \times 40 \times T$
Residual block	$\begin{bmatrix} \text{conv}, & 1 \times 1, 32, & \text{stride}(2,2) \\ \text{conv}, & 3 \times 3, 32, & \text{stride}(2,2) \\ \text{conv}, & 1 \times 1, 128, & \text{stride}(2,2) \end{bmatrix} \times 4$	$128 \times 20 \times (T/2)$
Residual block	$\begin{bmatrix} \text{conv}, & 1 \times 1, 64, & \text{stride}(2,2) \\ \text{conv}, & 3 \times 3, 64, & \text{stride}(2,2) \\ \text{conv}, & 1 \times 1, 256, & \text{stride}(2,2) \end{bmatrix} \times 6$	$256 \times 10 \times (T/4)$
Residual block	$\begin{bmatrix} \text{conv}, & 1 \times 1, 64, & \text{stride}(2,2) \\ \text{conv}, & 3 \times 3, 64, & \text{stride}(2,2) \\ \text{conv}, & 1 \times 1, 256, & \text{stride}(2,2) \end{bmatrix} \times 3$	$512 \times 5 \times (T/8)$

T represents sample time length.

product of a pair of vectors. $\boldsymbol{F}_x$ is the concatenation of global and local acoustic embedding descriptors and serves as the deep acoustic embedding descriptor.

3.2.2. PFE. Inspired by the MLP-based network [37], we propose an MLP architecture with full FC layers as PFE, and the input signals are constructed by 40-dim acoustic features from a 0.1 s window, which can present one kind of phoneme unit features in 0.1-s. Figure 1 illustrates the architecture, which is pretrained so that extracted phoneme features can be used as prior knowledge to train the proposed hybrid DNN.

The MLP architecture is composed of 20 FC layers with ReLU activation function, and each hidden FC layer in the MLP has a dimensionality of 512, with the exception of the final classification layer, which has a dimensionality of the total number of phoneme classes. This last layer is connected to a Softmax loss function that helps to enforce discriminative learning of phonetic units. The phoneme features for each signal fragment are extracted from the penultimate FC layer of the proposed MLP. During the MLP training, the input signals are divided into fragments with phonetic unit labels based on alignment information and serialized on the time axis for MLP training. After the MLP training, the input signals with the size of $R^{1 \times 40 \times T}$ are sliced into a series of 0.1-s fragments, which are sent into the MLP, as formulated:

$$\boldsymbol{x}_t^{(n+1)} = f\left(\boldsymbol{W}^{(n)} \boldsymbol{x}_t^{(n)} + \boldsymbol{b}^{(n)}\right), \tag{3}$$

and

$$\boldsymbol{F}_{x_t} = \boldsymbol{x}_t^{(20)}, \tag{4}$$

where $\boldsymbol{F}_{x_t}$ is the tth phoneme embedding descriptor in the sequence, $\boldsymbol{x}_t^{(n)}$ denotes the output of the frame $\boldsymbol{x}_t$ at the nth

FC layer in which $\boldsymbol{x}_t^{(0)}$ is equivalent to $\boldsymbol{x}_t$, $f(\cdot)$ denotes ReLU function, and $\boldsymbol{W}^{(n)}$ and $\boldsymbol{b}^{(n)}$ are the trainable weight and bias parameters of the nth FC layer in which $0 \leq n \leq 19$. As the input signal fragments propagate in MLP, deep phoneme embedding descriptors are extracted as a supplementary condition for deep speaker embedding learning.

3.2.3. Phoneme Feature Introduction (PFI). To enable cooperative learning, the extracted phoneme embedding descriptors should be integrated with acoustic embedding descriptors, allowing for the introduction of phoneme features into the desired discriminative deep speaker embedding.

As shown in Figure 1, fixed-size input signals are adopted for acoustic feature learning; meanwhile, these signals are equivalent to a series of 0.1-s fragments for phoneme embedding descriptor sequence learning. In this paper, two PFI methods are proposed to combine acoustic embedding descriptor and phoneme embedding descriptor sequence. The first one (*PFI_1*) is illustrated in Figure 1, concatenating all phoneme embedding descriptors in the sequence and concatenating the concatenated phoneme embedding descriptor with the acoustic embedding descriptor, i.e., in Equation (5),

$$\boldsymbol{F}=\left[\boldsymbol{F}_x, \left(\boldsymbol{F}_{x_1}, \ldots, \boldsymbol{F}_{x_t}, \ldots, \boldsymbol{F}_{x_T}\right)\right]. \tag{5}$$

The second one (*PFI_2*) is to sum the phoneme embedding descriptors in the sequence using vector addition and concatenating the summed phoneme embedding descriptor with the acoustic embedding descriptor, i.e., in Equation (6),

$$\boldsymbol{F}=\left[\boldsymbol{F}_x, \left(\boldsymbol{F}_{x_1} + \ldots + \boldsymbol{F}_{x_t} + \ldots + \boldsymbol{F}_{x_T}\right)\right], \tag{6}$$

where $\boldsymbol{F}_{x_t}$ is tth phoneme embedding descriptor in the sequence, and $\boldsymbol{F}$ is the concatenation of acoustic and phoneme embedding descriptors, which is compressed into a 1,024-dim utterance-level embedding descriptor at the followed embedding layer.

3.2.4. Loss Functions. Softmax loss is widely used in classification functions, and it performs well when the classification scale is not very large. It is employed for pre-training the MLP-base PFE as a supplementary subnetwork. In addition, the loss can also be explored to direct the entire hybrid DNN for desired deep speaker embedding. Softmax loss is formulated as follows:

$$L_{\text{Softmax}} = -\frac{1}{N}\sum_{i=1}^{N}\log\left(\frac{e^{\boldsymbol{W}_{y_i}^T \boldsymbol{x}_i + \boldsymbol{b}_{y_i}}}{\sum_{j=1}^{n} e^{\boldsymbol{W}_j^T \boldsymbol{x}_i + \boldsymbol{b}_j}}\right), \tag{7}$$

where N is the batch size. $\boldsymbol{x}_i$ denotes ith input, and $\boldsymbol{y}_i$ denotes its label. $\boldsymbol{W}$ and $\boldsymbol{b}$ are trainable parameters.

On this basis, a related study [40] has shown that Softmax loss does not explicitly encourage intraclass compactness when the classification scale is large, which means the hybrid network trained by Softmax loss may not be generalization enough. In order to force deep speaker embedding from the same speaker to be centralized and the clusters of different speakers to be separated, AAM Softmax loss is also explored to direct the hybrid network for discriminative and robust deep speaker embedding. It is a variant of Softmax loss, which can impose a fixed margin between speakers. The AAM Softmax loss is given by the following:

$$L_{\text{AAM-Softmax}} = -\frac{1}{N}\sum_{i=1}^{N}\log\left(\frac{e^{s\left(\cos\left(\theta_{y_i,i}+m\right)\right)}}{e^{s\left(\cos\left(\theta_{y_i,i}+m\right)\right)} + \sum_{j=1,j\neq i}^{n} e^{s\cos\left(\theta_{j,i}\right)}}\right), \tag{8}$$

where s denotes the scaling factor, and m denotes the angular margin. The $\theta_{j,i}$ is the angle between ith input $\boldsymbol{x}_i$ and jth column vector $\boldsymbol{W}_j$ of trainable $\boldsymbol{W}$.

4. Experiments

4.1. Datasets. To evaluate the proposed approach, we conduct experiments on four public datasets, including TIMIT [41], VoxCeleb1 [11], Aishell1 [12], and FMAudio_v1. Experiments are based on the analysis of four situations according to the properties of datasets.

TIMIT [41] is a small-scale open-source English corpus, containing 6,300 utterances of 630 speakers. Every utterance is labeled with phoneme unit tags in detail. In this paper, 3,420 utterances, longer than 2.5 s, are selected from its *train* subset and employed to train the PFE subnet as a phoneme unit recognition task.

VoxCeleb1 [11] is a large-scale open-source English corpus, which contains 153,516 utterances of 1,251 speakers. The corpus is recorded at 16-bit streams, 16 kHz sampling rate, and single channel. It includes two subsets: *dev* and *test*. The *dev* contains 1,211 speakers and their 148,642 utterances; *test* contains 40 speakers and their 4,874 utterances. This study uses *dev* to train the target networks and *test* to evaluate them.

Aishell1 [12] is a large-scale open-source mandarin corpus, containing 141,600 utterances of 400 speakers. The corpus is recorded at 16-bit streams, single channel, and downsampled to 16 kHz. *Train* and *dev* of it contains 380 speakers and their 134,424 utterances totally; *test* of it contains 20 speakers and their 7,176 utterances. This study uses *train* and *dev* to train the target networks and *test* to evaluate them.

FMAudio_v1 is a small-scale self-collected Mandarin corpus, which contains nearly 200 utterances of 20 speakers. The corpus is collected from FM broadcasting and transformed to 16-bit streams, 16 kHz sampling rate, and single channel. It includes two subsets: *dev* and *test*. The *dev* contains 15 speakers and their 117 utterances; *test* contains five speakers and their 61 utterances. This study adopts *dev* to fine-tune the target networks and *test* to evaluate them.

4.2. Signal Preprocessing. Typically, acoustic feature is prominent in TI-SV. The feature is essentially a type of magnitude

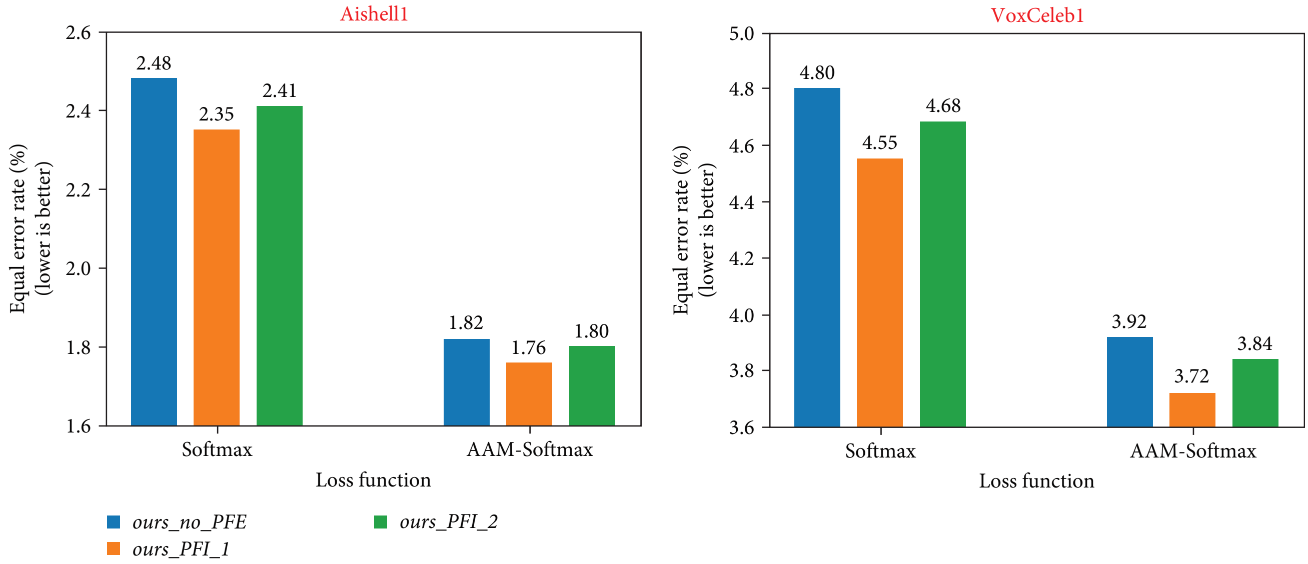

Figure 2: Results for the hybrid DNNs with different loss functions and structures.

information, which can be produced by eliminating phase using fast Fourier transform (FFT) and analytic signal analysis in the temporal frequency domain. The log mel-filter bank coefficients (Fbanks) are a common acoustic feature representation. We transform raw speech utterances to Fbanks as input signals of the proposed hybrid DNN. First, a 2-s fixed-length speech utterance is selected from each speech file and processed by the first-order high-pass filter for boosting higher frequencies. Second, the speech utterance is segmented using a 25 ms sliding window with a 10 ms shift between frames. Third, a hamming window is used to frames, enforcing the signal smooth by tapering the two ends of it. Finally, after computing a 512-point FFT and Mel filter bank containing 40 filters, each speech utterance is converted into corresponding Fbanks with the size of $R^{1\times40\times T}$.

4.3. Experimental Setup. In this study, all experiments are conducted on a Linux server computer whose main configurations are as follows: an Intel CPU i9-9900K with 3.60 GHz, a RAM of 64 GB, a NVIDIA GeForce RTX 2080Ti GPU, an environment of Python 3.5 as well as CUDA 11.1 The proposed networks are all implemented on the PyTorch 1.4 toolkit. The processing of our proposed TI-SV approach is divided into three stages, i.e., the development stage for network building, the enrollment stage for speaker-specific model construction, and the evaluation stage for SV.

In the development stage, a TI-SV network is trained to define the speakers manifold, which is optimized from a large collection of speech utterances. Our hybrid DNNs are trained using the stochastic gradient descent optimizer on the *dev* set of VoxCeleb1 and *train* and *dev* sets of Aishell1, respectively. The learning rate is initially set to 1e–1 and decreasing by 10 after the loss function value no longer drops more than twice. According to Li et al. [39], when the AAM-Softmax loss is used as the optimization criterion, the scaling factor s and the margin m are set to 32 and 0.1, respectively. PFE is pretrained with TIMIT, and the process is considered a phoneme unit classification task due to the accurate labels of the dataset. Fixed-size Fbanks with size $R^{1\times40\times T}$ preprocessed from raw speech segments are formed by 200 input frames with corresponding 40 frequency components, and the Fbanks are randomly divided into several training batches with a batch size of 64.

In the enrollment stage, a series of speaker-specific models are built for representing targeted speakers. Targeted speakers are completed disjoint from speakers in development. Part of speech segments in *test* set are sampled into frames and fed to the trained TI-SV networks as enrollment data. Then, the frame-level features are learned and aggregated into utterance-level embedding descriptors, in which the 1,024-dim utterance-level embedding descriptors can be extracted at the embedding layer. Multiple fixed-length utterances-level embedding descriptors of the same speaker are calculated into an average vector and stored as a speaker-specific model.

In the evaluation stage, every speech segment in *test* set except enrollment data is fed into the trained network to generate a 1,024-dim utterance-level embedding descriptor, which is compared to all speaker-specific models in a one-vs.-all way, and the final decision is made on the identity claim and similarity score. In this paper, cosine distance is utilized for calculating similarity scores between a pair of deep speaker embedding descriptors, and equal error rate (EER) is used as an evaluation indicator in the following experiments. Note that the lower the indicator EER, the better the TI-SV approach performance.

5. Results

5.1. Evaluation of the Hybrid DNNs with Different Loss Functions and Structures. Figure 2 shows the results of the hybrid networks with different loss functions and structures on two datasets. We conduct a set of experiments: (i) to

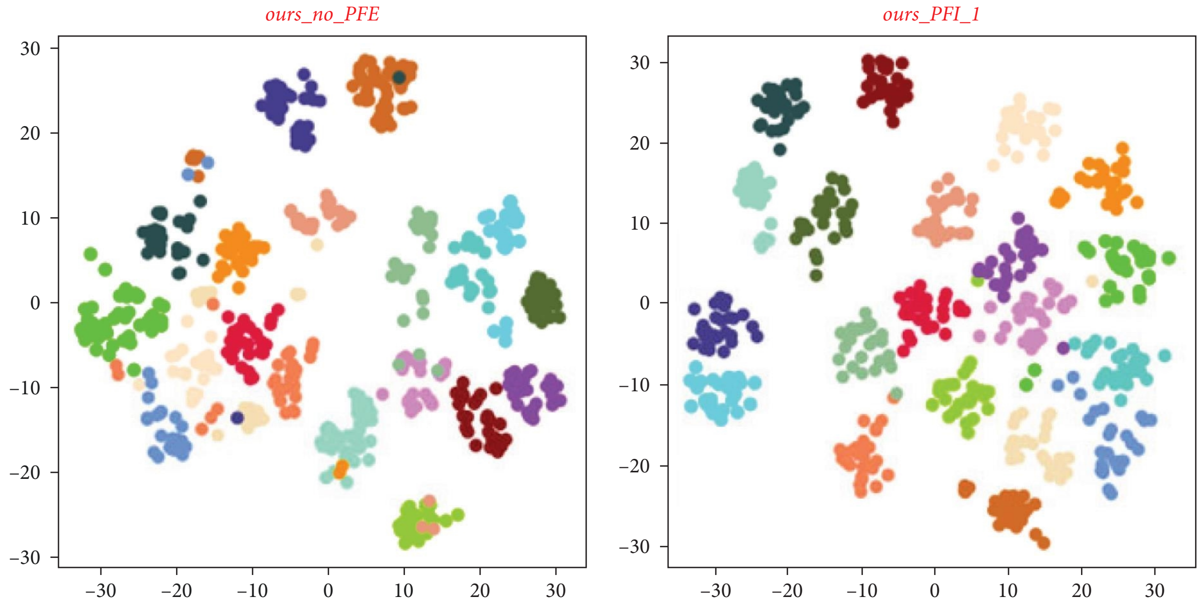

Figure 3: Visualization of deep speaker embedding descriptors learned from two networks. The first 20 speakers (#10270–#10289) and some of their speech utterances are used for visualization.

evaluate the hybrid network without PFE (*ours_no_PFE*) on the direction of Softmax loss and AAM Softmax loss, respectively; (ii) to evaluate the hybrid network, which adopts concatenating method for phoneme introduction (*ours_PFI_1*) on the direction of Softmax loss and AAM Softmax loss, respectively; (iii) to evaluate the hybrid network, which adopts summing method for phoneme introduction (*ours_PFI_2*) on the direction of Softmax loss and AAM Softmax loss, respectively.

In Figure 2, the cross-entropy loss Softmax, as well as the margin-based AAM-Softmax, are introduced. AAM-Softmax not only separates speaker classes but also maintains a fixed margin between speaker classes. The networks with AAM-Softmax loss surpass that with Softmax loss on two datasets, especially *ours_no_PFE* achieves 26% EER reduction on Aishell1 and 18% EER reduction on Voxceleb1, proving that Softmax loss does not explicitly encourage interclass separability and intraclass compactness, and AAM-Softmax is a benefit to speaker classification for TI-SV [40]; and the results also demonstrate that the margin of cross-entropy loss is the key to obtain discriminative deep speaker embedding descriptors. On two datasets, the performance of the networks with phoneme introduction is superior to the networks without phoneme introduction, and this phenomenon is more obvious on VoxCeleb1 than on Aishell1. This may be due to the PFE trained by the English corpus is language-related for accurately capturing pronunciation attributes of the target language rather than a language outside the target. Furthermore, *ours_PFI_1* invariably outperforms *ours_PFI_2*, in which 2.35% vs. 2.41%, 1.76% vs. 1.8% on Aishell1, and 4.55% vs. 4.68%, 3.72% vs. 3.84% on Voxceleb1.

To further analyze the results, we visualize the deep speaker embedding descriptors obtained by *ours_no_PFE* and *ours_PFI_1* on *test* set of Voxceleb1, respectively. Every speaker of *test* set is independent, so their speech sampling should exhibit clustering characteristics in the visualization. Figure 3 depicts the visualizations of the two types of deep speaker embedding descriptors, in which the first 20 speakers in *test* set with their speech segments are used for constructing visualization figures with moderate data volume. Each plot corresponds to the deep speaker embedding obtained over the fixed size of the speech utterance, projected into a 2-dim space by t-distributed stochastic neighbor embedding. Figure 3 shows the low variance and more sparse distribution of speaker clusters with phoneme feature integration embedding descriptors (*ours_PFI_1*) when compared to stand-alone acoustic-based embedding descriptors (*ours_no_PFE*). Therefore, the network *ours_PFI_1* outperforms the network *ours_no_PFE*, where *ours_PFI_1* encourages interclass separability and intraclass compactness by integrating acoustic and phoneme features, and *ours_no_PFE* does not possess the encouragement because it extracts deep speaker embedding descriptors from only acoustic features.

5.2. Comparison of the Hybrid DNN with Existing Benchmarks. Table 2 compares the performance of the hybrid DNNs using two PFIs to previous benchmarks. We evaluate these models all on *test* set of Voxceleb1. Note that no data augmentation is utilized in development. With introducing phoneme features to acoustic features, the hybrid DNNs using the proposed AWP outperform all other acoustic-based models by a significant margin. Particularly, the hybrid DNN using concatenating as PFI (*ours_PFI_1*) with AWP achieves a further performance boost with the EER of 3.72%, and it suggests that *PFI_1* can accurately preserve the temporality of phoneme unit features in a sequence, thereby significantly enhancing the performance of the proposed approach for TI-SV. Consequently, the use of phoneme features as prior knowledge, combined with acoustic features to learn highly discriminative deep speaker embedding descriptors, can lead to a substantial improvement in the performance of TI-SV systems.

Table 2: TI-SV results for the hybrid DNNs compared with existing benchmarks.

Feature	Method	Aggregation	Loss	EER (%)
AC	Nagrani et al. [11]	—	—	8.8
	Nagrani et al. [11]	TAP	Softmax	10.2
	Kim and Park [26]	TAP	AAM-Softmax	5.68
	Han et al. [27]	SAP	Softmax	5.75
	Han et al. [27]	SAP	AM-Softmax	4.15
	Cai et al. [28]	SAP	A-Softmax	4.40
	Cai et al. [28]	LDE	A-Softmax	4.48
	Wang et al. [29]	MHA	CosAMS	4.46
	Wang et al. [29]	MRMHA	CosAMS	4.10
	Wang et al. [29]	MRMHA	CosAMS	3.98
	Wang et al. [29]	MRMHA	CosAMS	3.96
AC&PH	*ours_PFI_1*	TAP	AAM-Softmax	4.24
	ours_PFI_2	TAP	AAM-Softmax	4.46
	ours_PFI_1	AWP	AAM-Softmax	**3.72**
	ours_PFI_2	AWP	AAM-Softmax	3.84

AC, acoustic; PH, phoneme; TAP, temporal average pooling; SAP, self-attention pooling; LDE, learnable dictionary encoding; MHA, multihead attention; MRMHA, multiresolution multihead attention; AWP, adaptive weight pooling. The use of "bold" is to emphasize the experimental result (3.72%).

Table 3: TI-SV results of the hybrid DNNs in FM broadcasting.

Method	Loss	Optimization	EER (%)
ours_no_PFE	AAM-Softmax	—	14.29
ours_PFI_1	AAM-Softmax	—	10.71
ours_no_PFE	AAM-Softmax	Fine-tuning	8.93
ours_PFI_1	AAM-Softmax	Fine-tuning	**7.14**

The use of "bold" is to emphasize the experimental result (7.14%).

5.3. Evaluation of the Hybrid DNNs Using FM Broadcasting Data. In our hybrid network architecture, the embedding layer is simply a dense layer, also known as a linear transformation layer with weights and biases. We take the pretrained hybrid network on the Aishell1 corpus as the initial network. Inspired by Zhu and Mak [42], we retrain the embedding layer while fixing the remaining structure for adapting the network to *FMAudio_v1* corpus. We increase the angular margin of the AAM-Softmax loss to 0.2, which enables better differentiation between speakers and decreases the learning rate exponentially from 1e–4 to 1e–6. Table 3 presents the results of our proposed hybrid networks for TI-SV on the small-scale, self-collected FM broadcasting corpus *FMAudio_v1*. No voice activity detection or automatic silence removal is applied in the experiment; due to the channel mismatch [43] between the microphone and FM broadcasting recordings, pretrained DNNs by Aishell1 result in suboptimal performance when directly applied to *FMAudio_v1*. To address this issue, we employ fine-tuning to adapt the pretrained DNNs from Aishell1 to *FMAudio_v1*. Fine-tuning improves the overall performance of our proposed hybrid networks significantly, as evidenced by the 20.04% reduction in EER achieved by *ours_PFI_1* (7.14%) compared to *ours_no_PFE* (8.93%). Whether fine-tuned or not, the performance of *ours_PFI_1* is relatively superior (10.71% vs. 14.29% and 7.14% vs. 8.93%), providing strong evidence that our use of PEE and introduction is effective in handling the inherent noise in FM broadcasting signals.

6. Conclusions

In this paper, we propose a deep speaker embedding learning method for TI-SV, viz., a hybrid DNN-introduced phoneme information. Our method involves the use of a hybrid DNN that combines a UBM and a PFE to extract acoustic and phoneme features, respectively. The PFE has been trained as a phoneme recognition model before it is connected to the hybrid DNN as a supplementary subnet. By means of inputting phoneme as prior knowledge, the deep speaker embedding learning is converted into a marginal probability calculation from a posterior probability calculation. The hybrid DNN not only serves the target of acoustic and phoneme feature aggregation but also makes use of the temporality of phoneme feature sequence to improve articulatory expressiveness. It is demonstrated that the proposed PFE and introduction method (PFI) are benefit to enforce the hybrid DNN to be discriminative for speaker identity, even in FM broadcasting. Experimental results verify the superiority of the hybrid DNN over existing benchmarks. Our approach provides new insights into the design of TI-SV systems and the potential benefits of combining acoustic and phoneme features. In future work, we plan to investigate a deep speaker embedding learning approach for SI tasks. By introducing phoneme information to reduce the impact of inherent noise in FM broadcasting signals, specific announcers can be accurately identified from numerous unknown speakers in real-time FM broadcasting.

Data Availability

Data are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the Joint funds for Regional Innovation and Development of the National Natural Science Foundation of China (grant no. U21A20449), the State's Key Project of Research and Development Plan (grant no. 2020YFB1807700), National Natural Science Foundation of China (grant no. 61971066), and BUPT Excellent Ph.D. Students Foundation (grant no. CX2022221).

References

[1] X. Li, X. Chen, D. Wang, Z. Guo, and K. Niu, "Deep speaker embedding with multi-part information aggregation infrequency-time domain for ASV," in *2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC)*, pp. 8–13, IEEE, Los Alamitos, CA, USA, June 2022.

[2] J. P. Campbell, W. Shen, W. M. Campbell, R. Schwartz, J.-F. Bonastre, and D. Matrouf, "Forensic speaker recognition," *IEEE Signal Processing Magazine*, vol. 26, no. 2, pp. 95–103, 2009.

[3] M. Aljasem, A. Irtaza, H. Malik et al., "Secure automatic speaker verification (SASV) system through sm-ALTP features and asymmetric bagging," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 3524–3537, 2021.

[4] L. Lu, L. Liu, M. J. Hussain, and Y. Liu, "I sense you by breath: speaker recognition via breath biometrics," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 2, pp. 306–319, 2020.

[5] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1–3, pp. 19–41, 2000.

[6] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[7] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *NTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association*, pp. 27–31, ISCA, Florence, Italy, 2011.

[8] M. Abadi, A. Agarwal, P. Barham et al., "Tensorflow: large-scale machine learning on heterogeneous distributed systems," 2016, https://arxiv.org/pdf/1603.04467.pdf.

[9] A. Paszke, S. Gross, S. Chintala et al., "Automatic differentiation in PyTorch," in *31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA, 2017.

[10] A. Vedaldi and K. Lenc, "MatConvNet: convolutional neural networks for MATLAB," in *MM '15: Proceedings of the 23rd ACM International Conference on Multimedia*, pp. 689–692, Association for Computing Machinery, Brisbane, Australia, October 2015.

[11] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: a large-scale speaker identification dataset," in *INTERSPEECH*, pp. 20–24, ISCA, Stockholm, Sweden, August 2017.

[12] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, "AISHELL-1: an open-source Mandarin speech corpus and a speech recognition baseline," in *2017 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA)*, pp. 1–5, IEEE, Seoul, Korea (South), November 2017.

[13] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: deep speaker recognition," in *INTERSPEECH*, pp. 1086–1090, ISCA, Graz, Austria, September 2018.

[14] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4052–4056, IEEE, Florence, Italy, May 2014.

[15] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-Vectors: robust DNN embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), ICASSP*, pp. 5329–5333, IEEE, Calgary, AB, Canada, April 2018.

[16] H. Zeinali, S. Wang, A. Silnova, P. Matejka, and O. Plchot, "But system description to voxceleb speaker recognition challenge," 2019.

[17] N. Tawara, A. Ogawa, T. Iwata, M. Delcroix, and T. Ogawa, "Frame-level phoneme-invariant speaker embedding for text-independent speaker recognition on extremely short utterances," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6799–6803, IEEE, Barcelona, Spain, May 2020.

[18] L. Yun, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1695–1699, IEEE, Florence, Italy, May 2014.

[19] S. Wang, J. Rohdin, L. Burget et al., "On the usage of phonetic information for text-independent speaker embedding extraction," in *INTERSPEECH*, pp. 1148–1152, ISCA, Graz, Austria, September 2019.

[20] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice modeling with sparse training data," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 3, pp. 345–354, 2005.

[21] A. Chowdhury and A. Ross, "Fusing MFCC and LPC features using 1D triplet CNN for speaker recognition in severely degraded audio signals," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1616–1629, 2020.

[22] W. Li, W. Quan, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4879–4883, IEEE Press, Calgary, AB, Canada, April 2018.

[23] Y. Tang, G. Ding, J. Huang, X. He, and B. Zhou, "Deep speaker embedding learning with multi-level pooling for text-independent speaker verification," in *ICASSP 2019—2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6116–6120, IEEE, Brighton, UK, May 2019.

[24] A. Torfi, J. Dawson, and N. M. Nasrabadi, "Text-independent speaker verification using 3d convolutional neural networks," in *2018 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, IEEE Computer Society, Los Alamitos, CA, USA, 2018.

[25] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *INTERSPEECH*, pp. 999–1003, ISCA, Stockholm, Sweden, August 2017.

[26] S.-H. Kim and Y.-H. Park, "Adaptive convolutional neural network for text-independent speaker recognition," in *INTERSPEECH*, pp. 66–70, ISCA, Brno, Czechia, 2021.

[27] S. Han, J. Byun, and J. W. Shin, "Time-domain speaker verification using temporal convolutional networks," in *ICASSP 2021—2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6688–6692, IEEE, Toronto, ON, Canada, June 2021.

[28] W. Cai, J. Chen, and M. Li, “Exploring the encoding layer and loss function in end-to-end speaker and language recognition system,” in *The Speaker and Language Recognition Workshop (Odyssey 2018)*, pp. 74–81, ISCA, Les Sables d’Olonne, France, June 2018.

[29] Z. Wang, K. Yao, X. Li, and S. Fang, “Multi-resolution multi-head attention in deep speaker embedding,” in *ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6464–6468, IEEE, Barcelona, Spain, May 2020.

[30] X. Ma, T. Liang, S. Zhang, S. Huang, and L. He, “Improved Light CNN with attention modules for ASV spoofing detection,” in *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, IEEE, Shenzhen, China, July 2021.

[31] X. Li and X. Wu, “Modeling speaker variability using long short-term memory networks for speech recognition,” in *INTERSPEECH*, pp. 1086–1090, ISCA, Dresden, Germany, September 2015.

[32] Y. Liu, L. He, J. Liu, and M. T. Johnson, “Speaker embedding extraction with phonetic information,” in *INTERSPEECH*, pp. 2247–2251, ISCA, Hyderabad, India, September 2018.

[33] G. Pironkov, S. Dupont, and T. Dutoit, “Speaker-aware long short-term memory multi-task learning for speech recognition,” in *2016 24th European Signal Processing Conference (EUSIPCO)*, pp. 1911–1915, IEEE, Budapest, Hungary, August 2016.

[34] Z. Tang, L. Li, D. Wang, and R. Vipperla, “Collaborative joint training with multitask recurrent model for speech and speaker recognition,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 3, pp. 493–504, 2017.

[35] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, “Speaker adaptation of neural network acoustic models using i-vectors,” in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 55–59, IEEE, Olomouc, Czech Republic, December 2014.

[36] P. Karanasou, Y. Wang, M. J. F. Gales, and P. C. Woodland, “Adaptation of deep neural network acoustic models using factorised i-vectors,” in *Fifteenth Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 2184–2180, ISCA, Singapore, September 2014.

[37] A. Silnova, P. Matejka, O. Glembek et al., “BUT/phonexia bottleneck feature extractor,” in *The Speaker and Language Recognition Workshop (Odyssey 2018)*, pp. 283–287, ISCA, Les Sables d’Olonne, France, June 2018.

[38] S. Schneider, A. Baevski, R. Collobert, and M. Auli, “wav2vec: unsupervised pre-training for speech recognition,” 2019.

[39] X. Li, X. Hu, X. Chen, H. Pan, and K. Niu, “Deep speaker embedding using hybrid network of multi-feature aggregation and multi-loss fusion for TI-SV,” in *2022 26th International Conference on Pattern Recognition (ICPR), ICPR*, pp. 506–512, IEEE, Montreal, QC, Canada, August 2022.

[40] X. Xiang, S. Wang, H. Huang, Y. Qian, and K. Yu, “Margin matters: towards more discriminative deep neural network embeddings for speaker recognition,” in *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 1652–1656, IEEE, Lanzhou, China, November 2019.

[41] L. F. Lamel, R. H. Kassel, and S. Seneff, “Speech database development: design and analysis of the acoustic-phonetic corpus,” in *Speech Input/Output Assessment and Speech Databases*, pp. 161–170, ISCA, The Netherlands, September 1989.

[42] Y. Zhu and B. Mak, “Orthogonal training for text-independent speaker verification,” in *ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6584–6588, IEEE, Barcelona, Spain, May 2020.

[43] H. Zhang, L. Wang, K. A. Lee, M. Liu, J. Dang, and H. Chen, “Meta-learning for cross-channel speaker verification,” in *ICASSP 2021—2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5839–5843, IEEE, Toronto, ON, Canada, June 2021.