Periodica Polytechnica Electrical Engineering and Computer Science, 65(4), pp. 310–328, 2021

Deep Learning Methods in Speaker Recognition: A Review

Dávid Sztahó[1]*, György Szaszák[1], András Beke[1]

[1] Department of Telecommunications and Media Informatics, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, H-1117 Budapest, Magyar tudósok körútja 2., Hungary
* Corresponding author, e-mail: sztaho@tmit.bme.hu

Received: 12 August 2020, Accepted: 11 November 2020, Published online: 16 September 2021

Abstract

This paper reviews the applied Deep Learning (DL) practices in the field of Speaker Recognition (SR), both in verification and identification. Speaker Recognition has been a widely used topic of speech technology. Many research works have been carried out and little progress has been achieved in the past 5–6 years. However, as Deep Learning techniques do advance in most machine learning fields, the former state-of-the-art methods are getting replaced by them in Speaker Recognition too. It seems that Deep Learning becomes the now state-of-the-art solution for both Speaker Verification (SV) and identification. The standard x-vectors, additional to i-vectors, are used as baseline in most of the novel works. The increasing amount of gathered data opens up the territory to Deep Learning, where they are the most effective.

Keywords

Speaker Recognition (SR), Speaker Verification (SV), Speaker Identification (SI), Deep Learning (DL), x-vector, i-vector, Deep Neural Networks (DNN)

1 Introduction

Speaker Identification (SI) and Verification (SV) have a still growing literature, due to their importance in speech technology. It is a popular research topic with various applications, such as security, forensics, biometric authentication, speech recognition and speaker diarization [1]. Due to the high number of studies in the field, a lot of methods have come up, so state-of-the-art in the field is quite mature, but also versatile, hence hard to overview.

Nowadays, as the popularity of Deep Learning (DL) is constantly rising due to easy accessible software and affordable hardware solutions, it began to infiltrate every topic, where machine learning is applicable. So, it is only natural that experts and scientists began to use Deep Learning in Speaker Recognition (SR). The aim of this study is to review the Deep Learning methods that are applied in Speaker Identification and verification tasks from the earliest to the latest solutions.

First, it is necessary to clarify the definition of Speaker Identification and verification, since these tasks are generally referred to when performing Speaker Recognition [1]. Speaker Identification is the task to identify an unknown speaker from a set of already known speakers: find the speaker who sounds closest to the test sample. When all speakers within a given set are known, it is called closed-set (or in-set) scenario. Alternatively, if the set of known speakers may not contain the potential test subject, it is called open-set (or out-of-set) Speaker Identification.

In Speaker Verification (SV), the task is to verify if a speaker, who claims to be of an identity, really is of the identity. In other words, we have to verify if the subject is really who he or she says to be. This means comparing two speech samples/utterances and deciding if they are spoken by the same speakers. This is usually done - in general Speaker Verification practice - by comparing the test sample to a sample of the given speaker and a Universal Background Model (UBM) [2].

Both Speaker Identification and Verification have their portfolio of use-cases and they also share their methodological and algorithmic inventory. Therefore, in this review, we examine the DNN methods for both tasks, always indicating the given task in every mentioned literature.

We focus only on DL methods in the field, as current state-of-the-art builds almost exclusively on top of neural architectures. For a Speaker Recognition tutorial, we recommend the work of Hansen and Hasan [1]. Due to the extensive nature of the field of Deep Learning, it is beyond the scope of this paper to give a detailed introduction about it. The methods are discussed with the assumption

Cite this article as: Sztahó, D., Szaszák, G., Beke, A. "Deep Learning Methods in Speaker Recognition: A Review", Periodica Polytechnica Electrical Engineering and Computer Science, 65(4), pp. 310–328, 2021. https://doi.org/10.3311/PPee.17024

that the reader has usable knowledge about the basic concepts associated with the field.

The paper is structured as follows: in Section 2, databases are described that are used in the studies. In Section 3, we give a short history about the outdated (but still used) GMM-UBM and i-vector systems. In Section 4, the commonly used evaluation metrics are described. After that, the DNN based solutions are detailed divided into different approaches according to how Deep Learning is used (such as feature extraction, classification). The Appendices contain large tables and a list of abbreviations used throughout the paper.

2 Databases

Like in many speech technology (and other machine learning) related topics, the used database is crucial. Developed methods can be evaluated and compared only if the same testing circumstances (from a machine learning point of view) are used. It is hard to say that an approach performs better, if it is evaluated on a different set or corpus. Therefore, the selection of the training and evaluation datasets require taking different considerations into account. There are numerous databases that are created and used in the field of Speaker Recognition, identification and verification. Table 1 in Appendix A shows all the corpuses that are used in the literature that is reviewed throughout the present paper. Presently available corpuses are listed along with their different properties that are found publicly. There are some datasets that are free, some are freely available for non-commercial purposes only.

Corpuses that are created mainly for Automatic Speech Recognition (ASR) can also be used to train (and evaluate) SR methods, however, most researches use datasets that focus especially the field of Speaker Identification and Verification. The main difference is the number of speakers contained in the database. Databases made for speech recognition typically contain less speakers. Speech recognition needs much more speech data in order to train phoneme models, but it often comes with lower speaker number. In contrast, Speaker Recognition needs as many speakers as possible, with less needed recorded material from each speaker. Also, recruiting many speakers is a more challenging job that requires more effort and is time consuming. The most often used corpuses for speech recognition (such as TIMIT [3], WSJ [4], RSR2015 [5], CHiME 2013 [6], VCTK [7]) have a few hundred speakers, whereas Librispeech [8], VoxCeleb [9], NIST SRE [10] datasets contain thousands of speakers. Of course, these large corpuses likely contain audio samples with various background noises, signal-to-noise ratios, recording setups and equipment. Therefore, they are suitable for machine learning aspects, but may not for linguistic analysis, in which case a more homogenous and a clean recording quality is necessary.

The largest corpus especially made for SR tasks is VoxCeleb2 [11]. It is a recent extension to its previous version (VoxCeleb). It contains samples from more than 6000 speakers that are downloaded from Youtube. Thus, its sound quality varies largely. In contrast, LibriSpeech is mostly a clean, good quality corpus. It is created from audiobooks, therefore maybe less suitable from real-world usage point of view, but appropriate for evaluating SR methods and features. NIST SRE datasets are also a huge collection of speaker samples, but recorded through telephone line quality. Thus, suitable for evaluations in yet another usage environment. A noisy and band limited quality makes the SR task harder, therefore it is more suitable to make a comparison between SR methods.

Although most corpuses contain English speech material, there are also other languages available. Some even contain multilingual content (see Table 1 in Appendix A). Another important aspect is if the given corpus has any additional segmentation or transcription included. If so, a more subtle analysis can be carried out (for example, using only given phonemes or partitioning the corpus into chunks with different sizes).

3 Classic state-of-the-art methods: GMM-UBM and i-vector

3.1 GMM-UBM

One of the first automatic Speaker Identification methods was based on Gaussian Mixture Models (GMM) [1, 2]. GMM is a combination of Gaussian probability density functions (PDFs) that are commonly used to model multivariate data. It does not only cluster data in an unsupervised way, but also gives its PDF. Applying GMM to speaker modelling provides the speaker specific PDF, from which probability score can be obtained. Thus, testing a sample with an unknown label, based on the probability scores of the speaker GMMs, a decision can be made.

A GMM is a mixture of Gaussian PDFs parameterized by a number of mean vectors, covariance matrices and weights:

$$f\left(\boldsymbol{x}_n \middle| \lambda\right) = \sum_{g=1}^{M} \pi_g N\left(\boldsymbol{x}_n \middle| \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g\right),$$

where π_g, $\boldsymbol{\mu}_g$ and $\boldsymbol{\Sigma}_g$ indicate the weight, mean vector and covariance matrix of the g^{th} mixture component. For a sequence of acoustic features $\left(\boldsymbol{X}=\left\{\boldsymbol{x}_n \middle| n\in 1\ldots T\right\}\right)$, the probability of observing these features is computed as:

$$p(\boldsymbol{X}|\lambda)=\prod_{n=1}^{T} p\left(\boldsymbol{x}_n|\lambda\right).$$

For Speaker Verification scheme, a slightly different approach was developed [1]. Beside the claimed speaker's model, an alternate model is necessary, which represents an "opposing" model. This alternate model is called the Universal Background Model (GMM-UBM). The GMM-UBM represents all others than the target speaker and it is trained on a large number of speaker samples. It was first used in [12]. Later, UBM was used as an initial model to the speaker models: rather than training GMMs on speaker data directly, the specific speaker models were created by adapting a prior UBM [13]. In the GMM-UBM scheme, H_0 and H_1 in Eq. (1) (see in Subsection 3.4 about LR test) are represented by speaker dependent GMM and the GMM-UBM, respectively.

3.2 GMM supervectors

Because speech samples could have different durations, much effort was put into developing methods that can obtain a fixed number of features from samples with variable lengths. One of the methods that performed the best in Speaker Recognition is forming GMM supervectors [14]. Supervectors are created by concatenating the parameters of the GMM (the mean vectors). This fixed length "supervector" is than fed to an applicable machine learning technique. Before Deep Neural Networks began to take much attention, Support Vector Machines (SVM) [15] were found to be the best performing technique.

3.3 The i-vector

Also, before the Deep Neural Networks era, the state-of-the-art Speaker Recognition method was the i-vector approach [16–18]. In this model, Factor Analysis (FA) was used to compute a speaker- and session-dependent GMM supervector:

$$\boldsymbol{m}_{s,h}=\boldsymbol{m}_0+T\boldsymbol{w}_{s,h},$$

where $\boldsymbol{m}_0$ is the GMM-UBM supervector, T is the speaker and channel factor, called total variability space and $\boldsymbol{w}_{s,h}\sim N(0,1)$ are hidden variables, called total factors. The total factors are not observable, but can be estimated using FA. These total factors can be used as features to a classifier afterwards, and came to be known as *i-vectors* (short for identity vector). The i-vector approach can be considered as a dimensionality reduction technique of the GMM supervector.

3.4 LR test

In Speaker Verification, the decision if a test sample belongs to a certain speaker is generally given by the Likelihood Ratio test (LR test) [1]. There are two hypotheses for an observation O:

- H_0: O is from speaker s
- H_1: O is not from speaker s.

In most of the approaches these cases are represented by a certain model parameterized by λ_s and λ_1, respectively. For a given set of observations $\boldsymbol{X}=\left\{\boldsymbol{x}_n \middle| n\in 1\ldots T\right\}$, the LR test is applied by evaluating the following ratio:

$$p\left(\boldsymbol{X}|\lambda_s\right)\geq\tau \text{ accept } H_0$$
$$p\left(\boldsymbol{X}|\lambda_1\right)>\tau \text{ reject } H_0,$$

where τ is the threshold of the decision. Commonly, the LR test is computed by using logarithmic probabilities (log-LR):

$$\Lambda(\boldsymbol{X})=\log p\left(\boldsymbol{X}|\lambda_s\right)-\log p\left(\boldsymbol{X}|\lambda_1\right). \tag{1}$$

4 Speaker Verification (SV) measurements

In Speaker Recognition (especially in verification) there are two kinds of similarity measures that are commonly used to compute the probabilities if a test observation is from the target speaker or not. Almost all novel DL approaches use these measures (in Speaker Verification schemes): cosine distance of vectors and PLDA (Probabilistic Linear Discriminant Analysis).

4.1 Cosine Distance Score (CDS)

The cosine distance is simply computing the normalized dot product of target and test i-vectors ($\boldsymbol{w}_{\text{target}}$ and $\boldsymbol{w}_{\text{test}}$), which provides a match score:

$$\text{CDS}\left(\boldsymbol{w}_{\text{target}},\boldsymbol{w}_{\text{test}}\right)=\frac{\boldsymbol{w}_{\text{target}}\times\boldsymbol{w}_{\text{test}}}{\left\|\boldsymbol{w}_{\text{target}}\right\|\times\left\|\boldsymbol{w}_{\text{test}}\right\|}.$$

4.2 PLDA

LDA (Linear Discriminant Analysis) [19] is used to find orthogonal axes for minimizing within-class variation and maximizing between-class variation. PLDA, as an extension of LDA [20, 21], is a probabilistic approach to the same method.

Generally, PLDA was applied to compare i-vectors. Of course, PLDA is capable to be applied to any vectors. Therefore, it can be used in new DL approaches, where i-vectors are replaced with their Deep Learning alternatives. Here, we give a brief description using the traditional i-vector approach.

Given a set of d dimensional length-normalized i-vectors $\boldsymbol{X} = \{\boldsymbol{x}_{ij}; i = 1, \dots, N; j = 1, \dots, H_i\}$ obtained from N training speakers (each has H_i i-vectors), i-vectors can be written in the following form:

$$\boldsymbol{x}_{ij} = \boldsymbol{\mu} + W\boldsymbol{z}_i + \boldsymbol{\epsilon}_{ij}$$

$$\boldsymbol{x}_{ij}, \boldsymbol{\mu} \in R^D, W \in R^{D\times M}, \boldsymbol{z}_i \in R^M, \boldsymbol{\epsilon}_{ij} \in R^D,$$

where $Z = \{\boldsymbol{z}_i; i = 1, \dots, N\}$ are latent variables, $\omega = \{\mu, W, \Sigma\}$ are model parameters, W is a $D \times M$ matrix (called factor loading matrix), $\boldsymbol{\mu}$ is the global mean of $\boldsymbol{X}$, z_i's are called the speaker factors and $\boldsymbol{\epsilon}_{ij}$'s are Gaussian distributed noise with zero mean and Σ covariance.

Given a test i-vector $\boldsymbol{x}_t$ and a target-speaker i-vector $\boldsymbol{x}_s$, the LR score can be computed:

$$\begin{aligned} S_{\text{LR}}\{x_t, x_s\} &= \frac{P(\boldsymbol{x}_s, \boldsymbol{x}_t | \text{same speaker})}{P(\boldsymbol{x}_s, \boldsymbol{x}_t | \text{different speaker})} \\ &= \frac{\int p(\boldsymbol{x}_s, \boldsymbol{x}_t, z | \omega)\mathrm{d}z}{\int p(\boldsymbol{x}_s, z|\omega)\mathrm{d}z_s \int p(\boldsymbol{x}_t, z|\omega)\mathrm{d}z_t} \\ &= \frac{\int p(\boldsymbol{x}_s, \boldsymbol{x}_t | z, \omega) p(z)\mathrm{d}z}{\int p(\boldsymbol{x}_s | z_s, \omega) p(z_s)\mathrm{d}z_s \int p(\boldsymbol{x}_t | z_t, \omega) p(z_t)\mathrm{d}z_t} \\ &= \frac{N([\boldsymbol{x}_s^T \boldsymbol{x}_t^T] | [\boldsymbol{\mu}^T \boldsymbol{\mu}^T], \hat{W}\hat{W}^T + \hat{\Sigma})}{N([\boldsymbol{x}_s^T \boldsymbol{x}_t^T] | [\boldsymbol{\mu}^T \boldsymbol{\mu}^T], diag\{WW^T + \Sigma, WW^T + \Sigma\})}, \end{aligned} \tag{2}$$

where $\hat{W} = [W^T W^T]^T$ and $\hat{\Sigma} = diag\{\Sigma, \Sigma\}$. Using Eq. (2) and the standard formula for the inverse of block matrices [22], the log-likelihood LR score is given by [21]:

$$S_{\text{LR}}(\boldsymbol{x}_s, \boldsymbol{x}_t) = const + \boldsymbol{x}_s^T Q \boldsymbol{x}_s + \boldsymbol{x}_t^T Q \boldsymbol{x}^T + 2\boldsymbol{x}_s^T P \boldsymbol{x}_t,$$

where

$$P = \Lambda^{-1}\Gamma(\Lambda - \Gamma\Lambda^{-1}\Gamma)^{-1}; \Lambda = WW^T + \Sigma$$

$$Q = \Lambda^{-1} - (\Lambda - \Gamma\Lambda^{-1}\Gamma)^{-1}; \Gamma = WW^T.$$

5 Deep Learning (DL) in Speaker Recognition (SR)

Generally, Deep Learning in Speaker Recognition has two major directions. One approach is to replace the i-vector calculation mechanism with a Deep Learning method as feature extraction. These works train a network on speaker samples using acoustic features (such as MFCCs or spectra) as inputs and speaker IDs as target variable and commonly use the output of an internal hidden layer as i-vector alternative and apply cosine distance or PLDA as decision making. The other main strategy is to use Deep Learning for classification and decision making, like replacing the cosine distance and PLDA with a discriminating deep network.

The performance of automatic Speaker Recognition systems is commonly evaluated by Equal Error Rate (EER) and Decision Cost Function (DCF). Equal Error Rate (EER) is a biometric security system algorithm used to predetermine the threshold values for its false acceptance rate and its false rejection rate [1, 23]. When the rates are equal, the common value is referred to as the Equal Error Rate. The value indicates that the proportion of false acceptances is equal to the proportion of false rejections. The lower the Equal Error Rate value, the higher the accuracy of the biometric system. Alternatively, the Decision Cost Function takes the prior probabilities of the target speaker occurrences, the proportion of target and non-target speakers into consideration. The detection cost function is a simultaneous measure of discrimination and calibration. Often, the minimum value of the DCF curve is called minDCF.

In Subsections 5.1–5.3, we give a detailed overview of the related works. The summary of the filtered essential citations is shown in Table 2 in Appendix B. An overview on how the reviewed methods relate to each other is depicted in Fig. 1. Abbreviations and details can be found in the text.

5.1 Deep Learning (DL) for feature extraction

The paper of Chen and Salman [24] is a relatively early work in deep feature extraction, in which bottleneck features (speaker models) are created using a Deep Neural Network with multiple subsets. Each subset is a deep autoencoder originally proposed in [25]. A hybrid learning strategy is proposed: the weights of the middle layer are shared across multiple inputs (adjacent frames) by a cost function:

$$L(\boldsymbol{x}_1, \boldsymbol{x}_2; \theta) = [L_R(\boldsymbol{x}_1; \theta) + L_R(\boldsymbol{x}_2; \theta)] + L_E(\boldsymbol{x}_1, \boldsymbol{x}_2; \theta),$$

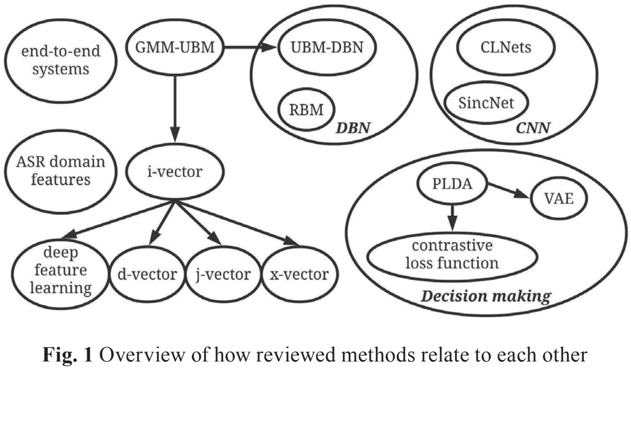

Fig. 1 Overview of how reviewed methods relate to each other

where $L_R(\boldsymbol{x}_i; \theta)$ is the loss of the network for input i, and $L_E(\boldsymbol{x}_1, \boldsymbol{x}_2; \theta)$ is a loss function optimized for learning the same speaker representation (model) at the layer, from which the speaker model features are extracted. For the experiments TIMIT, NTIMIT, KING, NKING, CHN and RUS dataset are used. According to the results, the proposed method outperformed the GMM-UBM baseline system in the case of all datasets.

5.1.1 The d-vector

There are numerous works that are aimed at extracting hidden layers of a DNN as features (substituting i-vectors). In [26] averaged activations of the last hidden layer of a network with multiple fully connected layers are selected as features, called as "d-vector" (Fig. 2). These vectors are later used in the same manner as i-vectors. Speaker Verification is done by cosine distance comparison. First, the network is trained in supervised manner, using 13-dimensional Perceptual Linear Predictive (PLP) features with Δ and ΔΔ values appended as frame-level feature vectors. After the training, the output layer is removed and the activations from the last hidden layers are used as features. The experiments were performed on a small footprint text-dependent corpus: 646 speakers speaking the same phrase: "ok google" multiple times. It was found that the general i-vector system mainly outperforms the newly proposed d-vector. The EERs (score normalized with t-norm) of the best performing setups were 1.21 % and 2.00 % for i-vector and d-vector, respectively.

5.1.2 The j-vector

The d-vector method was extended in [27] by a multi-task learning approach. The authors state that the intuition is that directly recognizing speaker seems to be hard but in reality, different speakers have their own style on each syllable or word. Therefore, using not only the speaker ids, but texts also as targets in a multi-learning setup, may increase the Speaker Verification performance. The used network is shown in Fig. 3. The applied cost function is the sum of the original loss functions:

$$C\left(\left[y_1, y_2\right], \left[y_1', y_2'\right]\right) = C_1\left(y_1, y_1'\right) + C_2\left(y_2, y_2'\right),$$

where C_1 and C_2 are two cross-entropy criteria for speakers and texts, y_1, y_2 indicate the true labels for speakers and texts individually and y_1', y_2' are the outputs of the two targets. As in the case of the original d-vector, after the supervised training phase, the output layer is removed and the output of the last hidden layer is used as a feature vector, defined as j-vector (joint vector). The experiments were done on the RSR2015 database [5]. The results show that the j-vector outperformed the d-vector approach. The EERs are 21.05 % and 9.85 % for *d-* and *j-vector*, respectively.

5.1.3 The x-vector

Another hidden layer extracted feature vector is called *x-vector* [28, 29]. It is based on DNN embeddings, based on a multiple layered DNN architecture (with fully connected layers) with different temporal context at each layer (which they call "frames"). Due to the wider temporal

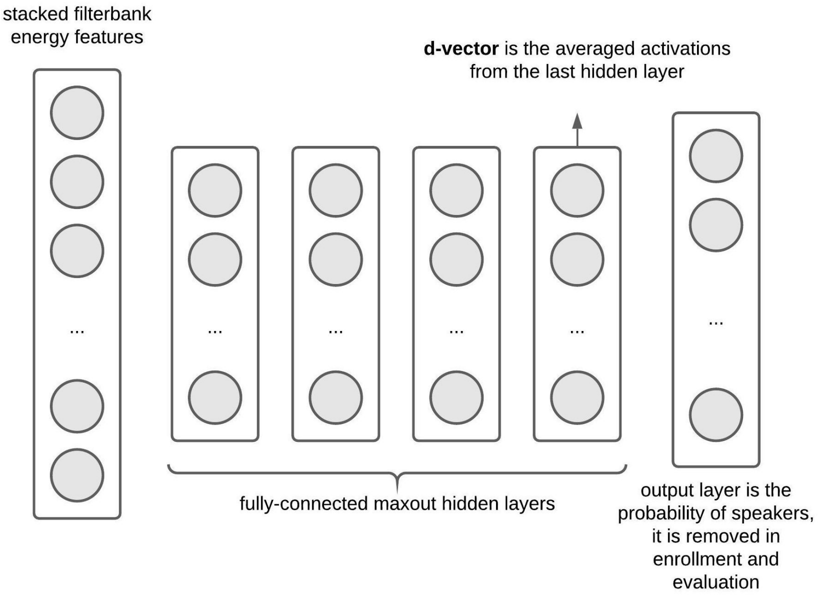
stacked filterbank energy features
d-vector is the averaged activations from the last hidden layer
...
fully-connected maxout hidden layers
output layer is the probability of speakers, it is removed in enrollment and evaluation

Fig. 2 DNN model in [26]

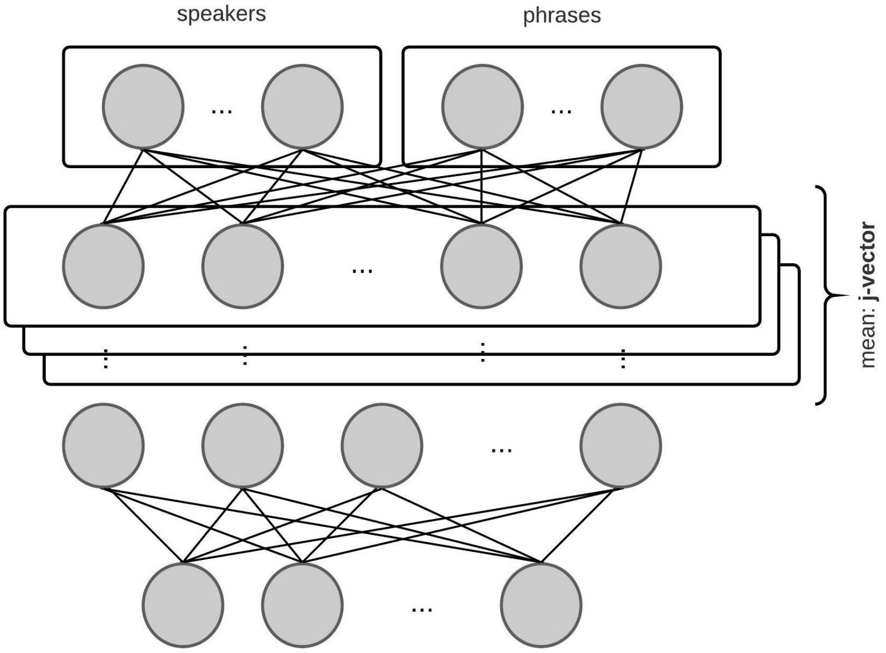

Fig. 3 Multi-task DNN in [27]

context, the architecture is called Time-delay Deep NN (TDNN). The TDNN embedding architecture can be seen in Fig. 4. The first five layers operate on speech frames, with small temporal context centered at the current frame *t*. For example, the frame indexed as "3" sees a total of 15 frames, due to the temporal context of the earlier layers. After training with speaker ids as target vectors, the output of layer segment6 ("x-vector") is used as input to a PLDA classifier. The input acoustic features are 24 dimensional filterbanks with 25 ms frame size, mean-normalized over a sliding window of up to 3 seconds. The used databases for evaluation include SWBD, NIST SRE 2016 and VoxCeleb. Data augmentation (increasing the amount of samples by adding babble noise, background music and reverb) was applied to various experimental setups. The main results show that x-vector outperforms the general i-vector based system (EERs are 9.23 % and 8.00 % for i-vector and x-vector, respectively). Using data augmentation, the difference is larger (EERs are 8.95 % and 5.86 % for i-vector and x-vector, respectively). The paper of Jiang et al. [30] extends the x-vector framework by so called dilated dense blocks, gate blocks and transition blocks. These blocks use convolutional layers to cover local features of different spans. On VoxCeleb, the extension results in 0.86 % EER decrease in absolute value (from 3.17 % to 2.31 %). Speaker representations can also be used to change the identity of the speaker. In [29] x-vectors are used for speaker anonymization. The extracted vector values are modified in order to change the speaker characterization and the speech is then re-synthetized, generating anonymized speech.

For short speech utterances, Kanagasundaram and colleagues [31] changed the dimension of the sixth and seventh layer ("segment6" and "segment7") to 150 in order to adapt to the shorter duration. It was found that the lower dimension of segment 6 and 7 helped in Speaker Verification in the case of 5-second-long utterances, but achieved higher EER on the original long utterances on the NIST SRE 2010 dataset. On the other hand, Garcia-Romero et al. [32] tried to optimize the x-vector system for long utterances (with 2–4 seconds duration) by a DNN refinement approach that updates a subset of the DNN parameters with full recordings and modifies the DNN architecture to produce embeddings optimized for cosine distance scoring. The results show that the method produces lower minDCF (minimum Decision Cost Function), but slightly higher EER than the baseline x-vector approach.

The x-vector was also applied in a multi-task learning scenario [33]. Beside the primary task (learning speaker identities), a second task was introduced: learning higher-order statistics of the input vector. By doing so, the system achieved slightly lower EER than the standard x-vector on the NIST SRE16 dataset: 7.79 % and 8.03 % for multi-task and baseline, respectively.

The x-vectors, in general, are incapable of leveraging unlabeled utterances, due to the classification loss over

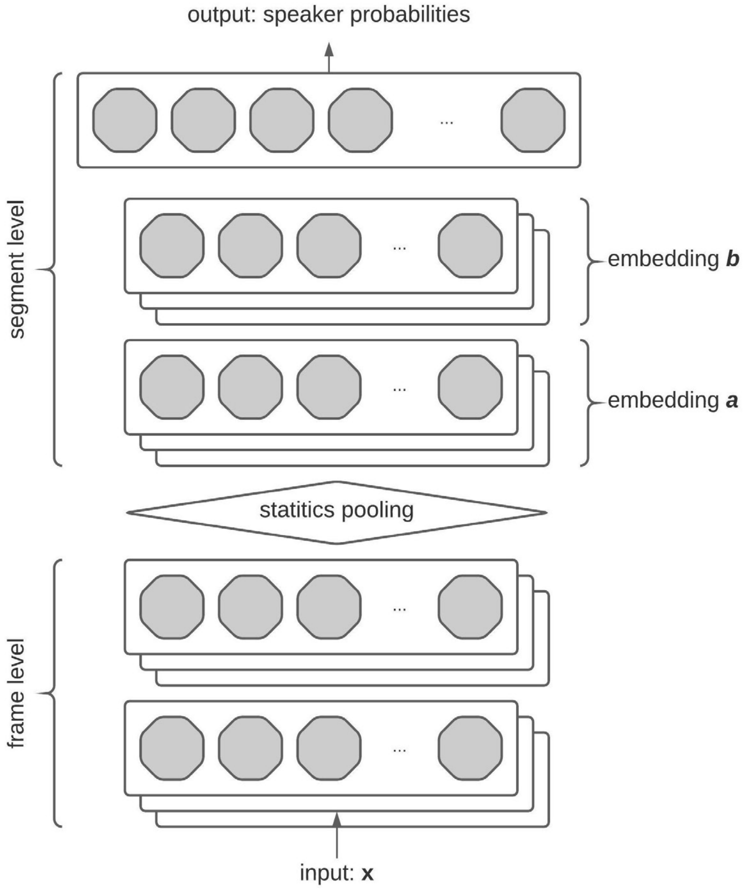

Fig. 4 The x-vector DNN embedding architecture in [28]. The features of the of the individual layers (top) and the structure of the network (bottom).

training speakers. The work of Stafylakis et al. [34] offers an alternative strategy based on x-vectors to train speaker embedding extractors via reconstructing the frames of a target speech segment, given the inferred embedding of another speech segment of the same utterance. They use a decoder network, to which the embedding vector is attached and by which the network serves as an autoencoder. The proposed decoder loss combined with the standard x-vector architecture and loss (i.e., crossentropy over training speakers) yielded improvement both on SITW and VoxCeleb datasets: ~0.4 % improvement in absolute EER compared to the standard x-vector system.

5.1.4 End-to-end systems

In order to do Speaker Verification, the embeddings are extracted and used in a standard backend, e.g., PLDA. Ideally the NNs should however be trained directly for the Speaker Verification task [35–38].

Instead of using cosine distance or PLDA classification, [35] apply an end-to-end solution for Speaker Verification with deep networks to obtain speaker representation vectors, estimation of a speaker model based on up to *N* enrollment utterances and also for verification (cosine similarity/logistic regression). The architecture is shown in Fig. 5. Both DNN (the same as the network used in d-vector extraction) and LSTMs are applied for speaker representation computation. The network is optimized using the end-to-end loss:

$$l_{e2e} = -\log p(\text{target})$$

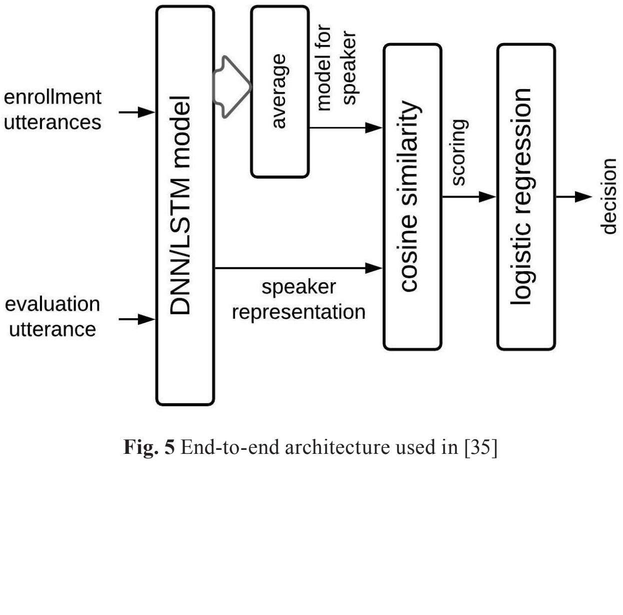

Fig. 5 End-to-end architecture used in [35]

with the binary variable $\text{target} \in \{\text{accept}, \text{reject}\}$, $p(\text{accept}) = \left(1+e^{-wS(\boldsymbol{X},\boldsymbol{spk})-b}\right)^{-1}$ and $p(\text{reject}) = 1 - p(\text{accept})$. The value $-b/w$ corresponds to the verification threshold. $S(\boldsymbol{X}, \boldsymbol{spk})$ is the cosine similarity between the speaker representation and the speaker model. The methods were tested on the "ok, google" dataset with more than 73 M utterances and 80 000 speakers. The results show that the end-to-end architecture performs similar to the d-vector approach if the same feature extractor (DNN) is used. However, LSTM lowered the EERs compared to the DNN solution: EERs are 2.04 % and 1.36 % for DNN and LSTM, respectively.

Another end-to-end system is proposed in [38], where the training was done by triplet loss aided by cosine similarity. A speaker embedding network is fed with raw speech waveform, which produces embedding vectors. This network is pre-trained with LibriSpeech by 1.5–2.0 sec uttarance chunks. Then the CHiME 2013 database [6] was used for Speaker Verification evaluation using specific 2 to 4 keywords only. The keywords were determined by an ASR, which was used in the training of the speaker embedding system in an adversarial way, forcing the embedding vectors to be speaker independent. Results show partial success: the triplet loss and ASR adversarial training did not improve the EER in the 2 keywords case, but it did if 3 or 4 keywords were examined.

5.1.5 Deep Belief Networks (DBN)

Deep Belief Networks (DBN) are another type of Deep Learning networks that are used in Speaker Recognition [39, 40]. Deep Belief Networks are generative models with numerous layers of latent variables, which are typically binary. Neurons in the same layers are not connected and connection between adjacent layers are undirected. Training of DBNs are hard due to the intractability of inferring the posterior distribution from the hidden (latent) layers. Stacked Restricted Boltzmann Machines (RBMs) can be applied as a DBN architecture (Fig. 6). For more details, see [41]. The objective of DBN is to learn abstract hierarchical representations of unlabeled input data. In [40], spectrograms (25 ms window size, 10 ms timestep) have been fed as input speech data after applying PCA transformation to reduce dimensionality. Activations of first and second layers of the RBM were used as features (both separately and together) appended to common MFCC features. After feature extraction, GMM-UBM was used to perform Speaker Recognition. The authors used the ELSDSR dataset with 22 speakers. Based on the results, the features extracted from the RBM helped the recognition: 90 % and 95 % final accuracies were obtained by using separate MFCC and mixed MFCC+RBM features, respectively.

Ali et al. [39] also use the same acoustic feature extraction method, but they add a Bag of Words method in order to convert the data with different lengths into vectors of the same dimensionality (using a k-means clustering technique). SVM is applied as a classifier. The experiments were done on the Urdu dataset [42] with ten speakers. Here, also hybrid (MFCC+DBN) features performed the best: 88.6 % and 92.60 % accuracies were obtained for MFCC and MFCC+DBN features, respectively.

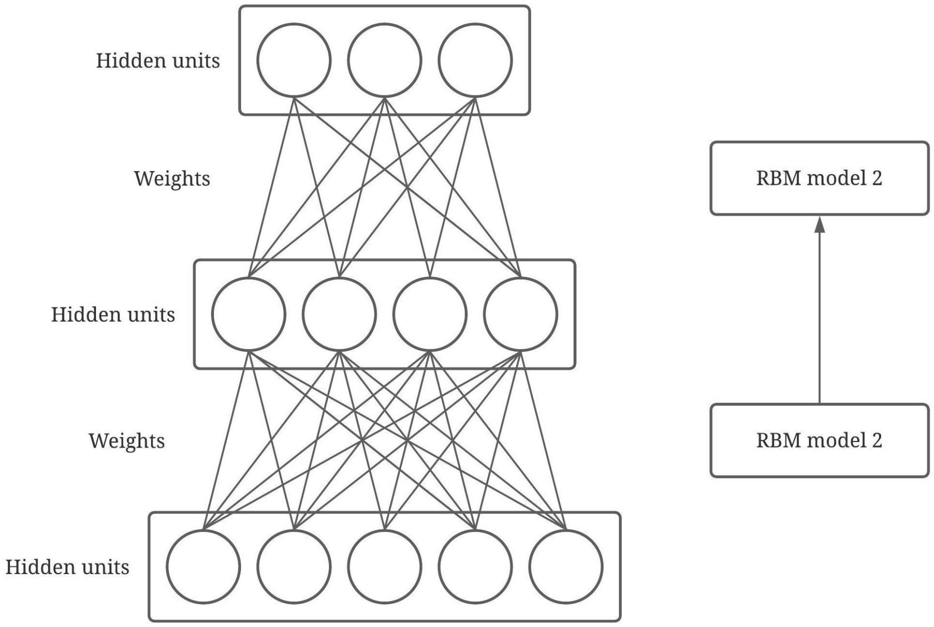

Fig. 6 Structure of the DBN used for extraction of short-term spectral features, with two hidden layers, can be visualized as a stack of 2 RBMs [40]

In [43], a widespread evaluation of multiple DNN methods for deep feature extraction are given using deep Restricted Boltzmann Machines (RBMs), speech-discriminant Deep Neural Network, speaker-discriminant Neural Network and multi-task joint-learned Deep Neural Networks. RBMs are used in the same way as in the previous Subsubsection 5.1.4 [39, 40]. A speech discriminant DNN was applied with text labels as training data and triphone states as target. This scenario can be useful in a text-dependent Speaker Verification task. The outputs of the last hidden layer are used as features. In the case of speaker discriminant DNN, the outputs of the speech discriminant network are changed to speaker IDs. This way, a more speaker specific feature set can be obtained and it is a more natural choice for Speaker Verification. In the multi-task setup, both previously mentioned (speaker IDs and triphones) outputs are used as targets. A standard i-vector system trained with PLP features was used as baseline (GMM-UBM with cosine similarity). The newly proposed deep features were tested separately and by combining them in various ways on the RSR2015 dataset [5]. Compared to the baseline result (1.5 % EER), the speaker discriminant and multi-task DNNs achieved the best performances (1.06 % and 0.80 % EER respectively). The best combination of deep features (concatenating RBM and multi-task features) gave 0.73 % EER. Also, with PLDA performed after deep feature extraction, 0.20 % EER was achieved for speaker discriminant features.

5.1.6 CLNets

In [44] a deep Corrective Learning Network (CLNet) is proposed to analyze independent samples by a recurrent formalism. Each new instance makes a corrective prediction to update the predictions made from prior data. This means that instead of averaging the results for segments of a speaker, an incremental strategy is used. CLNets are applied using convolution layers for Speaker Verification. NIST SRE 2004-2010 corpora are used for the experiments. By using cosine similarity, ~2.5 % lower EER was obtained compared to the standard i-vector system (7.3 %, 5.18 % and 4.87 % EERs for i-vector, standard CNN and CLNets, respectively). However, using PLDA, i-vector performed better.

5.1.7 Text dependency

Still, i-vector systems outperform the DNN ones in a text independent scenario [45]. So, taking the standard i-vector PLDA system as basis, [36] proposed an end-to-end DNN that learns sufficient statistics of GMM-UBM and provides i-vectors. In the first part of the network, GMM posteriors are learned by a multiple layered architecture, then the standard i-vectors are used as targets with cosine distance as loss function.

5.2 Deep Learning (DL) for classification

Rather than applying deep feature extraction to exchange the common i-vectors for a more robust and better performing speaker representation, DNNs can also be used to replace the backend systems for scoring and comparison (like PLDA and cosine distance). Such works are sparser in literature than those related to feature extraction.

5.2.1 Variational Autoencoder (VAE)

Variational Autoencoder (VAE) [46, 47] is a generative model for signal (and speech) modelling. It is used in voice conversion [48–50], speech recognition and also for Speaker Recognition [51, 52]. Instead of using just deterministic layers, a VAE consists of stochastic neurons also. The LR scoring is made by:

$$\mathrm{LR}\left(\boldsymbol{x}_1,\boldsymbol{x}_2\right)=\frac{P\left(\boldsymbol{x}_1,\boldsymbol{x}_2\middle|H_{tar}\right)}{P\left(\boldsymbol{x}_1,\boldsymbol{x}_2\middle|H_{imp}\right)}=\frac{P\left(\boldsymbol{x}_1,\boldsymbol{x}_2\middle|\theta\right)}{P\left(\boldsymbol{x}_1\middle|\theta\right)P\left(\boldsymbol{x}_2\middle|\theta\right)},$$

where H_{tar}, H_{imp} are the hypotheses about the facts that $\boldsymbol{x}_1, \boldsymbol{x}_2$ are related to the same or different speakers respectively and θ is the parameters of the speaker model. The results showed that VAEs don't seem to be superior to PLDA scoring.

5.2.2 Multi-domain features

Text dependent data were also used for classification in a Speaker Recognition task to help learning speaker IDs. Tang et al. [53] used the output of an ASR to improve the performance of Speaker Recognition. Fig. 7 shows the proposed multi-task learning scheme. The output of

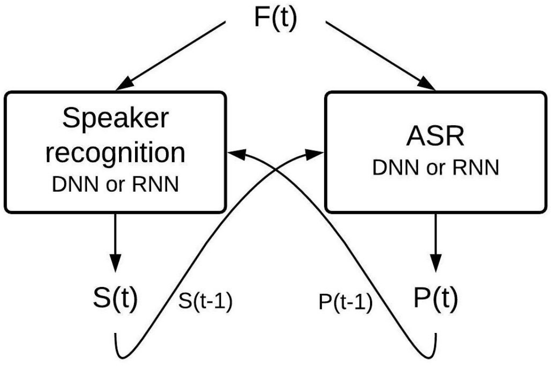

Fig. 7 Multi-task recurrent learning in [53] for ASR and SRE. *F*(*t*) denotes primary features, *P*(*t*) denotes phone identities, *S*(*t*) denotes speaker identities.

the ASR (phone-posteriors) is fed into the SRE system, and vice versa. The input of each task is formed from the extracted frame-level spectra (filterbanks and MFCCs for ASR and SRE, respectively). The experiments were done on the WSJ dataset. Based on the results, the proposed method achieved equal or slightly better EERs, than the i-vector baseline (0.57 % and 0.55 % for i-vector and multi-task method, respectively).

5.2.3 Replacing UBM with DNN

DNNs can be used to replace the UBM also. Universal Deep Belief Networks (UDBN) [54] are used as backend, in which a two-class hybrid DBN-DNN is trained for each target speaker to increase the discrimination between target i-vectors and the i-vectors of the other speakers (non-targets/ impostors). First, an unsupervised universal DBN is trained, which is then adapted to the target speakers by a special balanced training process. In the test phase, an unknown i-vector is matched to the adapted target i-vectors. Based on evaluation done on NIST SRE 2006 and 2014 datasets, the proposed algorithm did not achieve better performance than the i-vector PLDA baseline method. However, fusing the DNN approach with the PLDA (i-vector) method, revealed better performance than the i-vector alone.

5.2.4 Using contrastive loss for vector comparison

Since Speaker Identification is treated as a simple classification task, softmax layers can be applied to create a DNN backend system. However, in Speaker Verification, the comparison of two (speaker modelling) vectors is necessary. In a DNN, a way to achieved this is using contrastive loss [55] as loss function on deep features. Convolutional networks (namely VGG [56, 57]) [9] and ResNets [11, 58] can be trained this way to perform Speaker Verification tasks. On VoxCeleb and VoxCeleb2 datasets, lower EERs were obtained than in the case of standard i-vector PLDA systems: 8.8 %, 7.8 % and 3.95 % EERs for i-vector, CNN and ResNet, respectively. However, in [11] ResNet and the baseline system were not trained on the same dataset (RestNet: VoxCeleb2, i-vector: VexCeleb1), therefore this increase could come from the effect of the larger audio material.

5.2.5 SincNet

Convolutional Neural Networks (CNNs) are also used in Speaker Recognition, using spectrograms [9, 59, 60] or raw speech waveform as input [61, 62]. SincNet [61] is a special CNN architecture that gets raw waveforms as inputs. Before applying standard CNN/DNN layers it learns high and low cut-off frequencies of band-pass filters by a convolutional layer (Fig. 8). In Speaker Identification task, compared to MFCC-fed DNN, the SincNet achieved better performance on TIMIT and LibriSpeech: 0.99 %, 2.02 % Classification Error Rate (CER) for TIMIT and LibriSpeech with DNN, and 0.85 % and 0.96 % CER for SincNet, respectively. SincNet was also compared to CNN with filterbank energies as inputs. The conclusion was that on smaller dataset (such as TIMIT), the filter learning was not as effective as on a large dataset (LibriSpeech). On TIMIT, the results were comparable. On LibriSpeech, however, SincNet outperformed the CNN architecture (1.55 % and 0.96 % CER for CNN and SincNet, respectively). It was found that SincNet also outperformed the other DNN solutions (and the standard i-vector PLDA system) in a Speaker Verification setup. Both d-vector (used with cosine distance) and speaker class posteriors were applied.

SincNet was extended in [63] for an unsupervised speaker embedding learning by using mutual information as objective function for embedding vector comparison. An additional decrease in EER was examined: from 7.2 to 5.8 % on the VoxCeleb corpus.

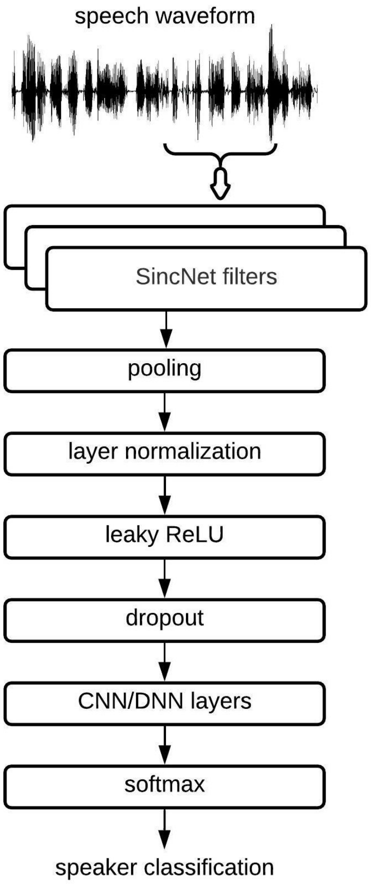

Fig. 8 Architecture of SincNet in [61]

5.2.6 Unlabeled data

When doing Speaker Recognition, labeled data is not always present. There are some approaches that take advantage of large scale unlabeled training data. Curriculum learning is one of them [64–66]. It starts by learning a DNN model using a labeled corpus and continuously introduces unlabeled, out-of-domain text independent speaker samples. Both LSTM [64] and TDNN [66] based systems are proposed that outperform baseline methods.

5.3 Other usage of DNN in Speaker Recognition (SR)

In [67] DNN is used in a non-common way to aid Speaker Recognition. The extraction of sufficient statistics for the general i-vector model is driven by a Deep Neural Network trained for Automatic Speech Recognition. This DNN is used to produce frame alignments, specifically providing posteriors of semitones. First, DNN is trained for segmenting the speech into senones, using a pre-trained general HMM-GMM ASR system. The i-vector training is done on the semitone-level segmented speech. The final flow diagram of the proposed method is shown in Fig. 9. The experiments were done using the two extended NIST SRE'12 conditions: clean and slightly noisy telephone speech. The pre-trained HMM-GMM system used a 39 dimensional MFCC vector, including 13 MFCC and their first and second order derivatives. The input of the DNN in the HMM-DNN was composed of 15 frames, using 40 log Mel-filterbank for each. The results of the proposed method was compared to a standard i-vector system (GMM-UBM and i-vector). The HMM-DNN method achieved a slightly lower EER: 1.39 % and 1.81 % for DNN and UBM, respectively for clean speech; 1.92 % and 2.55 % for DNN and UBM, respectively for noisy speech.

Yet another topic of Speaker Recognition is forensic sciences and applications. In forensics, all the above mentioned methods and technologies must be applied through the LR framework, in order to get evidence based and jurisdiction compliance decision making. For this kind of examinations, usually specific datasets are needed that are consistent with the given evidences and use-case scenarios. For a very good review and more details on forensics based on speech, see [68–71].

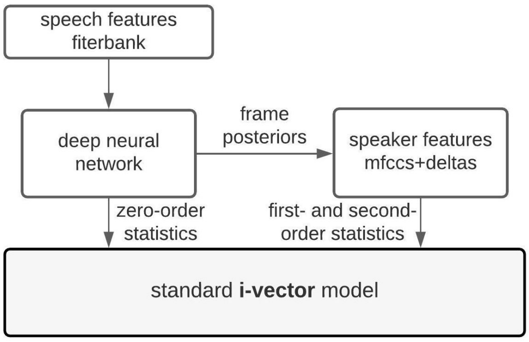

Fig. 9 Flow diagram of the DNN/i-vector hybrid framework in [67]

6 Conclusions

In this paper we summarized the applied Deep Learning practices in the field of Speaker Recognition, both for verification and identification. The early DL solutions to replace feature extraction (such as i-vectors) provided comparable but not higher performance than the previous state-of-the-art i-vector PLDA systems. Although newer DL architectures led to increasing classification accuracies, it is well-known in the literature that i-vectors provide competitive performance, when more training material is used for each speaker and when longer test sentences are employed [72–74]. However, the latest works offer superior results. In some cases, the reported results show significantly lower EERs, but mostly the achieved performances are only a little better than the previous ones. Nonetheless, it seems that DL becomes the now state-of-the-art solution for both Speaker Verification and identification. The standard x-vectors, additional to i-vectors, are used as baseline in most of the novel works. The increasing amount of gathered data opens up the territory to DL, where they are the most effective. Additionally, newer and newer DL architectures are developed, that can lead to a breakthrough in Speaker Recognition too. Based on the literature, it is hard to derive a final conclusion about the "best" method for sSpeaker Recognition. The *x-vector* became the de facto standard, used in practical applications and as baseline method to beat.

Acknowledgement

The work was funded by project no. FK128615, which has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the FK_18 funding scheme.

Nomenclature

SI	Speaker Identification
SV	Speaker Verification
DL	Deep Learning
SR	Speaker Recognition
DNN	Deep Neural Networks
GMM	Gaussian Mixture Model
UBM	Universal Background Model

ASR	Automatic Speech Recognition
PDF	Probability Distribution Function
LR	Likelihood Ratio
FA	Factor Analysis
CDS	Cosine Distance Score
PLDA	Probabilistic Linear Discriminant Analysis
LDA	Linear Discriminant Analysis
DCF	Decision Cost Function
EER	Equal Error Rate
PLP	Perceptual Linear Predictive
TDNN	Time-delay Deep Neural Network
NN	Neural Network
LSTM	Long Short-Term Memory
DBN	Deep Belief Networks
RBM	Restricted Bolzmann Machines
MFCC	Mel Frequency Cepstral Coefficient
SVM	Support Vector Machines
CLNet	Corrective Learning Network
VAE	Variational Autoencoder
UDBN	Universal Deep Belief Networks
CNN	Convolutional Neural Network
CER	Classification Error Rate

References

[1] Hansen, J. H. L., Hasan, T. "Speaker Recognition by Machines and Humans: A tutorial review", IEEE Signal Processing Magazine, 32(6), pp. 74–99, 2015.
https://doi.org/10.1109/MSP.2015.2462851

[2] Reynolds, D. A., Rose, R. C. "Robust text-independent speaker identification using Gaussian mixture speaker models", IEEE Transactions on Speech and Audio Processing, 3(1), pp. 72–83, 1995.
https://doi.org/10.1109/89.365379

[3] Garofolo, J. S., Lamel, L. F., Fisher, W. M., Fiscus, J. G., Pallett, D. S., Dahlgren, N. L. "DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus CD-ROM", U.S. Department of Commerce, Technology Administration, National Institute of Standards and Technology, Computer Systems Laboratory, Advanced Systems Division, Gaithersburg, MD, USA, Rep. NISTIR 4930, 1993.
https://doi.org/10.6028/nist.ir.4930

[4] Marcus, M., Santorini, B., Marcinkiewicz, M. A. "Building a Large Annotated Corpus of English: The Penn Teebank", Computational Linguistics, 19(2), pp. 313–330, 1993.

[5] Larcher, A., Lee, K. A., Ma, B., Li, H. "The RSR2015: Database for Text-Dependent Speaker Verification using Multiple Pass-Phrases", In: Interspeech-2012, Portland, OR, USA, 2012, pp. 1580–1583.

[6] Vincent, E., Barker, J., Watanabe, S., Le Roux, J., Nesta, F., Matassoni, M. "The second 'chime' speech separation and recognition challenge: Datasets, tasks and baselines", In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Vancouver, BC, Canada, 2013, pp. 126–130.
https://doi.org/10.1109/ICASSP.2013.6637622

[7] Veaux, C., Yamagishi, J., MacDonald, K. "CSTR VCTK Corpus: English Multi-speaker Corpus for CSTR Voice Cloning Toolkit", [sound], University of Edinburgh, The Centre for Speech Technology Research (CSTR), Edinburgh, UK, 2017.
https://doi.org/10.7488/ds/1994

[8] Panayotov, V., Chen, G., Povey, D., Khudanpur, S. "Librispeech: An ASR corpus based on public domain audio books", In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brisbane, Australia, 2015, pp. 5206–5210.
https://doi.org/10.1109/ICASSP.2015.7178964

[9] Nagrani, A., Chung, J. S., Zisserman, A. "VoxCeleb: A Large-Scale Speaker Identification Dataset", In: Interspeech 2017, Stockholm, Sweden, 2017, pp. 2616–2620.
https://doi.org/10.21437/Interspeech.2017-950

[10] Greenberg, C., Martin, A., Graff, D., Brandschain, L., Walker, K. "2010 NIST Speaker Recognition Evaluation Test Set", [sound, hard drive], LDC Catalog No.: LDC2017S06, Linguistic Data Consortium, Philadelphia, PA, USA, 2017.
https://doi.org/10.35111/fjsq-a117

[11] Chung, J. S., Nagrani, A., Zisserman, A. "VoxCeleb2: Deep Speaker Recognition", In: Interspeech 2018, Hyderabad, India, 2018, pp. 1086–1090.
https://doi.org/10.21437/Interspeech.2018-1929

[12] Reynolds, D. A., Quatieri, T. F., Dunn, R. B. "Speaker Verification Using Adapted Gaussian Mixture Models", Digital Signal Processing, 10(1–3), pp. 19–41, 2000.
https://doi.org/10.1006/dspr.1999.0361

[13] Gauvain, J. L., Lee, C. H "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains", IEEE Transactions on Speech and Audio Processing, 2(2), pp. 291–298, 1994.
https://doi.org/10.1109/89.279278

[14] Campbell, W. M., Sturim, D. E., Reynolds, D. A. "Support vector machines using GMM supervectors for speaker verification", IEEE Signal Processing Letters, 13(5), pp. 308–311, 2006.
https://doi.org/10.1109/LSP.2006.870086

[15] Cortes, C., Vapnik, V. "Support-vector networks", Machine Learning, 20(3), pp. 273–297, 1995.
https://doi.org/10.1007/BF00994018

[16] Dehak, N., Dehak, R., Kenny, P., Brümmer, N., Ouellet, P., Dumouchel, P. "Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification", In: Interspeech 2019, Brighton, UK, 2009, pp. 1559–1562.

[17] Dehak, N., Kenny, P. J., Dehak, R., Dumouchel, P., Ouellet, P. "Front-End Factor Analysis for Speaker Verification", IEEE Transactions on Audio, Speech, and Language Processing, 19(4), pp. 788–798, 2011.
https://doi.org/10.1109/TASL.2010.2064307

[18] Dehak, N., Kenny, P., Dehak, R., Glembek, O., Dumouchel, P., Burget, L., Hubeika, V., Castaldo, F. "Support vector machines and Joint Factor Analysis for speaker verification", In: 2009 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'09), Taipei, Taiwan, 2009, pp. 4237–4240.
https://doi.org/10.1109/ICASSP.2009.4960564

[19] Bishop, C. M. "Pattern recognition and machine learning", Springer, Berlin, Germany, 2006.

[20] Tipping, M. E., Bishop, C. E. "Probabilistic principal component analysis", Neural Computing Research Group, Aston University, Birmingham, UK, Rep. NCRG/97/010, 1997.

[21] Ioffe, S. "Probabilistic Linear Discriminant Analysis", In: Leonardis, A., Bischof, H., Pinz, A. (eds.) Computer Vision – ECCV 2006, Springer, Berlin, Germany, 2006, pp. 531–542. https://doi.org/10.1007/11744085_41

[22] Petersen, K. B., Pedersen, M. S. "9.1 Block matrices", In: The Matrix Cookbook, [pdf] Technical University of Denmark, Kgs. Lingby, Denmark, p. 46, 2012. Available at: https://www.math.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf [Acessed: 01 June 2020]

[23] van Leeuwen, D. A., Brümmer, N. "An Introduction to Application-Independent Evaluation of Speaker Recognition Systems", In: Müller, C. (ed.) Speaker Classification I, Springer, Berlin, Germany, 2007, pp. 330–353. https://doi.org/10.1007/978-3-540-74200-5_19

[24] Chen, K., Salman, A. "Learning Speaker-Specific Characteristics With a Deep Neural Architecture", IEEE Transactions on Neural Networks, 22(11), pp. 1744–1756, 2011. https://doi.org/10.1109/TNN.2011.2167240

[25] Hinton, G. E., Salakhutdinov, R. R. "Reducing the Dimensionality of Data with Neural Networks", Science, 313(5786), pp. 504–507, 2006. https://doi.org/10.1126/science.1127647

[26] Variani, E., Lei, X., McDermott, E., Moreno, I. L., Gonzalez-Dominguez, J. "Deep neural networks for small footprint text-dependent speaker verification", In: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 2014, pp. 4052–4056. https://doi.org/10.1109/ICASSP.2014.6854363

[27] Chen, N., Qian, Y., Yu, K. "Multi-task learning for text-dependent speaker verification", In: Interspeech 2015, Dresden, Germany, 2015, pp. 185–189.

[28] Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., Khudanpur, S. "X-Vectors: Robust DNN Embeddings for Speaker Recognition", In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 2018, pp. 5329–5333. https://doi.org/10.1109/ICASSP.2018.8461375

[29] Fang, F., Wang, X., Yamagishi, J., Echizen, I., Todisco, M., Evans, N., Bonastre, J.-F. "Speaker Anonymization Using X-vector and Neural Waveform Models", In: 10th ISCA Workshop on Speech Synthesis (SSW 10), Vienna, Austria, 2019, pp. 155–160. https://doi.org/10.21437/SSW.2019-28

[30] Jiang, Y., Song, Y., McLoughlin, I., Gao, Z., Dai L.-R. "An Effective Deep Embedding Learning Architecture for Speaker Verification", In: Interspeech 2019, Graz, Austria, 2019, pp. 4040–4044. https://doi.org/10.21437/Interspeech.2019-1606

[31] Kanagasundaram, A., Sridharan, S., Sriram, G., Prachi, S., Fookes, C. A. "Study of X-vector Based Speaker Recognition on Short Utterances", In: Interspeech 2019, Graz, Austria, 2019, pp. 2943–2947. https://doi.org/10.21437/Interspeech.2019-1891

[32] Garcia-Romero, D., Snyder, D., Sell, G., McCree, A., Povey, D., Khudanpur, S. "X-vector DNN Refinement with Full-length Recordings for Speaker Recognition", In: Interspeech 2019, Graz, Austria, 2019, pp. 1493–1496. https://doi.org/10.21437/Interspeech.2019-2205

[33] You, L., Guo, W., Dai, L., Du, J. "Multi-Task Learning with High-Order Statistics for X-vector Based Text-Independent Speaker Verification", In: Interspeech 2019, Graz, Austria, 2019, pp. 1158–1162. https://doi.org/10.21437/Interspeech.2019-2264

[34] Stafylakis, T., Rohdin, J., Plchot, O., Mizera, P., Burget, L. "Self-supervised Speaker Embeddings", In: Interspeech 2019, Graz, Austria, 2019, pp. 2863–2867. https://doi.org/10.21437/Interspeech.2019-2842

[35] Heigold, G., Moreno, I., Bengio, S., Shazeer, N. "End-to-end text-dependent speaker verification", In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 2016, pp. 5115–5119. https://doi.org/10.1109/ICASSP.2016.7472652

[36] Rohdin, J., Silnova, A., Diez, M., Plchot, O., Matějka, P., Burget, L. "End-to-End DNN Based Speaker Recognition Inspired by I-Vector and PLDA", In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 2018, pp. 4874–4878. https://doi.org/10.1109/ICASSP.2018.8461958

[37] Gao, Z., Song, Y., McLoughlin, I., Li, P., Jiang, Y., Dai, L. R. "Improving Aggregation and Loss Function for Better Embedding Learning in End-to-End Speaker Verification System", In: Interspeech 2019, Graz, Austria, 2019, pp. 361–365. https://doi.org/10.21437/Interspeech.2019-1489

[38] Yun, S., Cho, J., Eum, J., Chang, W., Hwang, K. "An End-to-End Text-independent Speaker Verification Framework with a Keyword Adversarial Network", In: Interspeech 2019, Graz, Austria, 2019, pp. 2923–2927. https://doi.org/10.21437/Interspeech.2019-2208

[39] Ali, H., Tran, S. N., Benetos, E., d'Avila Garcez, A. S. "Speaker recognition with hybrid features from a deep belief network", Neural Computing and Applications, 29(6), pp. 13–19, 2018. https://doi.org/10.1007/s00521-016-2501-7

[40] Banerjee, A., Dubey, A., Menon, A., Nanda, S., Nandi, G. C. "Speaker Recognition using Deep Belief Networks", [eess.AS], arXiv:1805.08865v1, Cornell University, Ithaca, NY, USA, 2018. [online] Available at: https://arxiv.org/abs/1805.08865 [Accessed: 01 June 2020]

[41] Hinton, G. E., Osindero, S., Teh, Y.-W. "A Fast Learning Algorithm for Deep Belief Nets", Neural Computation, 18(7), pp. 1527–1554, 2006. https://doi.org/10.1162/neco.2006.18.7.1527

[42] Appen Pty Ltd. "ARL Urdu Speech Database, Training Data", [sound, web download], LDC Catalog No.: LDC2007S03, Linguistic Data Consortium, Philadelphia, PA, USA, 2007. https://doi.org/10.35111/6z57-s580

[43] Liu, Y., Qian, Y., Chen, N., Fu, T., Zhang, Y., Yu, K. "Deep feature for text-dependent speaker verification", Speech Communication, 73, pp. 1–13, 2015. https://doi.org/10.1016/j.specom.2015.07.003

[44] Wen, Y., Zhou, T., Singh, R., Raj, B. "A Corrective Learning Approach for Text-Independent Speaker Verification", In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 2018, pp. 4894–4898.
https://doi.org/10.1109/ICASSP.2018.8461340

[45] Snyder, D., Ghahremani, P., Povey, D., Garcia-Romero, D., Carmiel, Y., Khudanpur, S. "Deep neural network-based speaker embeddings for end-to-end speaker verification", In: 2016 IEEE Spoken Language Technology Workshop (SLT), San Diego, CA, USA, 2016, pp. 165–170.
https://doi.org/10.1109/SLT.2016.7846260

[46] Kingma, D. P., Welling, M. "Auto-Encoding Variational Bayes", [stat.ML], arXiv:1312.6114v10, Cornell University, Ithaca, NY, USA, 2013. [online] Available at: https://arxiv.org/abs/1312.6114 [Accessed: 01 June 2020]

[47] Rezende, D. J., Mohamed, S., Wierstra, D. "Stochastic Backpropagation and Approximate Inference in Deep Generative Models", Proceedings of the 31st International Conference on Machine Learning, 32(2), pp. 1278–1286, 2014.

[48] Hsu, C. C., Hwang, H. T., Wu, Y. C., Tsao, Y., Wang, H. M. "Voice Conversion from Unaligned Corpora Using Variational Autoencoding Wasserstein Generative Adversarial Networks", In: Interspeech 2017, Stockholm, Sweden, 2017, pp. 3364–3368. 2017.
https://doi.org/10.21437/Interspeech.2017-63

[49] Hsu, W., Zhang, Y., Glass, J. "Unsupervised domain adaptation for robust speech recognition via variational autoencoder-based data augmentation", In: 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), Okinawa, Japan, 2017, pp. 16–23.
https://doi.org/10.1109/ASRU.2017.8268911

[50] Mohammadi, S. H., Kim, T. "Investigation of Using Disentangled and Interpretable Representations for One-shot Cross-lingual Voice Conversion", In: Interspeech 2018, Hyderabad, India, 2018, pp. 2833–2837.
https://doi.org/10.21437/Interspeech.2018-2525

[51] Villalba, J., Brümmer, N., Dehak, N. "Tied Variational Autoencoder Backends for i-Vector Speaker Recognition", In: Interspeech 2017, Stockholm, Sweden, 2017, pp. 1004–1008.
https://doi.org/10.21437/Interspeech.2017-1018

[52] Pekhovsky, T., Korenevsky, M. "Investigation of Using VAE for i-Vector Speaker Verification", [cs.SD], arXiv:1705.09185, Cornell University, Ithaca, NY, USA, 2017. [online] Available at: https://arxiv.org/abs/1705.09185 [Accessed: 01 June 2020]

[53] Tang, Z., Li, L., Wang, D. "Multi-task recurrent model for speech and speaker recognition", In: 2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA), Jeju, Korea (South), 2016, pp. 1–4.
https://doi.org/10.1109/APSIPA.2016.7820893

[54] Ghahabi, O., Hernando, J. "Deep Learning Backend for Single and Multisession i-Vector Speaker Recognition", IEEE/ACM Transactions on Audio, Speech, and Language Processing, 25(4), pp. 807–817, 2017.
https://doi.org/10.1109/TASLP.2017.2661705

[55] Chopra, S., Hadsell, R., LeCun, Y. "Learning a similarity metric discriminatively, with application to face verification", In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 2005, pp. 539–546.
https://doi.org/10.1109/CVPR.2005.202

[56] Simonyan, K., Zisserman, A. "Very Deep Convolutional Networks for Large-Scale Image Recognition", [cs.CV], arXiv:1409.1556v6, Cornell University, Ithaca, NY, USA, 2014. [online] Available at: https://arxiv.org/abs/1409.1556 [Accessed: 01 June 2020]

[57] Yadav, S., Rai, A. "Learning Discriminative Features for Speaker Identification and Verification", In: Interspeech 2018, Hyderabad, India, 2018, pp. 2237–2241.
https://doi.org/10.21437/Interspeech.2018-1015

[58] He, K., Zhang, X., Ren, S., Sun, J. "Deep Residual Learning for Image Recognition", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 770–778.
https://doi.org/10.1109/CVPR.2016.90

[59] Ji, R., Cai, X., Bo, X. "An End-to-End Text-Independent Speaker Identification System on Short Utterances", In: Interspeech 2018, Hyderabad, India, 2018, pp. 3628–3632.
https://doi.org/10.21437/Interspeech.2018-1058

[60] Hajavi, A., Etemad, A. "A Deep Neural Network for Short-Segment Speaker Recognition", In: Interspeech 2019, Graz, Austria, 2019, pp. 2878–2882.
https://doi.org/10.21437/Interspeech.2019-2240

[61] Ravanelli, M., Bengio, Y. "Speaker Recognition from Raw Waveform with SincNet", In: 2018 IEEE Spoken Language Technology Workshop (SLT), Athens, Greece, 2018, pp. 1021–1028.
https://doi.org/10.1109/SLT.2018.8639585

[62] Salvati, D., Drioli, C., Foresti, G. L. "End-to-End Speaker Identification in Noisy and Reverberant Environments Using Raw Waveform Convolutional Neural Networks", In: Interspeech 2019, Graz, Austria, 2019, pp. 4335–4339.
https://doi.org/10.21437/Interspeech.2019-2403

[63] Ravanelli, M., Bengio, Y. "Learning Speaker Representations with Mutual Information", In: Interspeech 2019, Graz, Austria, 2019, pp. 1153–1157.
https://doi.org/10.21437/Interspeech.2019-2380

[64] Marchi, E., Shum, S., Hwang, K., Kajarekar, S., Sigtia, S., Richards, H., Haynes, R., Kim, Y., Bridle J. "Generalised Discriminative Transform via Curriculum Learning for Speaker Recognition", In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 2018, pp. 5324–5328.
https://doi.org/10.1109/ICASSP.2018.8461296

[65] Ranjan, S., Hansen, J. H. L. "Curriculum Learning Based Approaches for Noise Robust Speaker Recognition", IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP), 26(1), pp. 197–210, 2018.
https://doi.org/10.1109/TASLP.2017.2765832

[66] Zheng, S., Liu, G., Suo, H., Lei, Y. "Autoencoder-based Semi-Supervised Curriculum Learning for Out-of-domain Speaker Verification", In: Interspeech 2019, Graz, Austria, 2019, pp. 4360–4364.
https://doi.org/10.21437/Interspeech.2019-1440

[67] Lei, Y., Scheffer, N., Ferrer, L., McLaren, M. "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), Florence, Italy, 2014, pp. 1695–1699.
https://doi.org/10.1109/ICASSP.2014.6853887

[68] Broeders, A. P. A. "Forensic Speech and Audio Analysis Forensic Linguistics: A Review 2001 to 2004", In: 14th INTERPOL Forensic Science Symposium, Lyon, France, 2004, pp. 171–188.

[69] Morrison, G. S., Rose, P., Zhang, C. "Protocol for the collection of databases of recordings for forensic-voice-comparison research and practice", Australian Journal of Forensic Sciences, 44(2), pp. 155–167, 2012.
https://doi.org/10.1080/00450618.2011.630412

[70] Nolan, F. "Voice Quality and Forensic Speaker Identification", Govor, 24(2), pp. 111–128, 2007.

[71] Morrison, G. S., Enzinger, E., Ramos, D., González-Rodríguez, J., Lozano-Díez, A. "Statistical Models in Forensic Voice Comparison", In: Banks, D. L., Kafadar, K., Kaye, D. H., Tackett, M. (eds.) Handbook of Forensic Statistics, Boca Raton, FL, USA, 2020, pp. 451–497.

[72] Sarkar, A. K., Matrouf, D., Bousquet, P. M., Bonastre, J. F. "Study of the Effect of I-vector Modeling on Short and Mismatch Utterance Duration for Speaker Verification", In: Interspeech 2012, Portland, OR, USA, 2012, pp. 2662–2665.
https://doi.org/10.21437/Interspeech.2012-347

[73] Travadi, R., Van Segbroeck, M., Narayanan, S. "Modified-prior i-Vector Estimation for Language Identification of Short Duration Utterances", In: Interspeech 2014, Singapore, 2014, pp. 3037–3041.
https://doi.org/10.21437/Interspeech.2014-609

[74] Kanagasundaram, A., Vogt, R., Dean, D., Sridharan, S., Mason, M. "i-vector Based Speaker Recognition on Short Utterances", In: Interspeech 2011, Florence, Italy, 2011, pp. 2341–2344.
https://doi.org/10.21437/Interspeech.2011-58

Appendix A

Table 1 Summary of datasets used for speaker recognition. Cells were left empty if no available information is present for the given detail.

Name	Language	#speakers	Mean of utterance duration	Total duration	Number of utterances	Type of recorded material	Annotation	Segmentation	Recording conditions	Dataset splitting	Multimodal	Date of creation
TIMIT, NTIMIT	English	630				phonetically compact and diverse read sentences		timestamp for words and phonetic segments	silent	train + test	no	1986
Librispeech	English	1239		~510 hours		audiobooks	transcription		silent	train + validation + test	no	2015
VoxCeleb	English	1251	8.2 s		153516 utterances	Youtube videos	identity, nationality		mixed	train + verification + identification	yes	2017
VoxCeleb2	English	6112		2442	1128246 utterances	Youtube videos	identity, nationality		mixed	train + development + test	yes	2018
NIST SRE 2010	English			2255		telephone conversation + interview			telephone + microphone	train + development	no	2010
NIST SRE 2008	Yue Chinese, Wu Chinese, Vietnamese, Uzbek, Urdu, Thai, Tagalog, Tamil, Russian, Panjabi, Min Nan Chinese, Lao, Korean, Japanese, Italian, Hindi, Persian, Mandarin Chinese, Bengali, Egyptian Arabic, Moroccan Arabic, Dari, Iranian Persian, English, Chinese, Arabic	400?		942 hours		telephone conversation + interview					no	2008
NIST SRE 2006	Yue Chinese, Urdu, Thai, Russian, Korean, Hindi, English, Mandarin Chinese, Bengali, Standard Arabic, Chinese, Arabic	3918				telephone conversation			telephone		no	2006

Name	Language	#speakers	Mean of utterance duration	Total duration	Number of utterances	Type of recorded material	Annotation	Segmentation	Recording conditions	Dataset splitting	Multimodal	Date of creation
NIST SRE 2000	Yue Chinese, Wu Chinese, Vietnamese, Uzbek, Urdu, Thai, Tagalog, Tamil, Russian, Panjabi, Min Nan Chinese, Lao, Korean, Japanese, Italian, Hindi, Persian, Mandarin Chinese, Bengali, Egyptian Arabic, Moroccan Arabic, Dari, Iranian Persian, English, Chinese, Arabic	~4000		more than 2300 hours		telephone conversation + interview			telephone + microphone	train + development	no	2000-2010
ELSDSR	English	22		4.56 hours	198 utterances				silent	train + test	no	2004
Urdu dataset	Urdu (Pakistan)	10			2500 isolated words	isolated words	transcript (list of words)		silent			
RSR2015	English	300		151 hours		sentences, short commands, random digits			phone + tablet	-	no	2015
WSJ	English	245			77800 utterances	predefined + free sentences	transcription		silent	train + development	no	1994
SWBD	English	543			2400 conversations	phone conversation		transcript + alignment	mixed		no	1993
KING, NKING	English	51		76.5 minutes	102 conversations	phone conversation	transcription		noisy (max 20 dB)	-		1995
VCTK	English	109			43600 sentences	read sentences	transcription				no	2017

Appendix B

Table 2 Summary of essential cited works in order of publication date. Cells were left empty if no available information is present for the given detail.

reference	publication date	main focus	datasets used for testing	applied method(s)	error rates	main advances	disadvantages
[24]	2011	speaker-specific characterization learning and Speaker Verification	TIMIT, NTIMIT, KING, NKING, CHN, RUS	feature extraction (learning) using multi-layered deep autoencoders	only DET curves are given	superior performance compared to GMM-UBM	outdated as to x-vector
[26]	2014	Speaker Verification	ok, google's dataset	speaker specific vector extraction from the last layer of a DNN (with fully connected layers): d-vector	1.21 % EER for i-vector and 2.00 % EER for d-vector	DNN application	usage of cosine distance scoring and the worse performance than i-vector
[54]	2014	Speaker Verification	SRE 2006, 2014	i-vector framework with Deep Learning based decision making	1.5 % lower EER with DNN	DNN based decision making instead of PLDA	performance improvement not convincing
[67]	2014	Speaker Verification	NIST SRE 2012	i-vector using DNN as i-vector extractor	1.99 % and 1.39 % EERs for baseline i-vector and DNN extracted i-vector approach	method is compatible with existing standard i-vector systems	relative performance to x-vector is not known
[27]	2015	Speaker Verification	RSR2015	speaker specific vector extraction from the last layer of a DNN with multi-task learning: j-vector	1.47 %, 1.62 % and 0.54 % EERs for i-vector, d-vector and j-vector	DNN and PLDA application, better perfomance than i-vector and j-vector	applied only to short utterances (average duration: 3.2 s)
[43]	2015	Speaker Verification	RSR2015	feature extraction from multiple Deep Learning methods	1.15 % and 0.22 % EERs for d-vector and proposed method	a complete comparison of multiple Deep Learning method based feature extraction	limitation: only text-dependent methods
[35]	2016	Speaker Verification	ok, google's dataset	end-to-end solution (with DNN and LSTM) that also includes PLDA decision making	4.66 % and 1.35 % EERs for i-vector and end-to-end LSTM-based DNN	application of LSTM layer and the inclusive of decision making	
[53]	2016	speaker identification	WSJ	multi-task learning scenario: automatic speech and speaker recognizer fusion	0.57 % and 0.62 % EERs for i-vector and applied methods	usage of speech recognizer in a multi-task scenario	worse performance than i-vector baseline
[51]	2017	Speaker Verification	NIST SRE 2010	variation autoencoder for i-vector learning	worse than baseline i-vector	VAE approach	worse performance than i-vector
[52]	2017	Speaker Verification	NIST SRE 1998-2008, NIST SRE 2010	variation autoencoder	varying EERs compared to i-vector baseline	VEA approach with large dataset	mixed results compared to i-vector
[28]	2018	Speaker Verification	SWBD, NIST SRE	speaker specific vector extraction from an internal layer of a TDNN: x-vector	6.09 % and 4.16 % EERs for i-vector and x-vector	comprehensive study using sample augmentation and robust speaker specific vector extraction	
[39]	2018	speaker identification	Urdu Dataset	deep belief network (DBN)	88.6 % and 92.6 % accuracy for MFCC and DBN	a basic work for DBNs	small dataset
[40]	2018	speaker identification	ELSDSR	deep belief network (DBN)	90 % and 95 % accuracy for MFCC and DBN features	a larger dataset than [39]	only 22 speakers

reference	publication date	main focus	datasets used for testing	applied method(s)	error rates	main advances	disadvantages
[44]	2018	Speaker Verification	NIST SRE 2004-2008, 2010	CLNets	2.71 %, 2.57 % and 1.79 % EERs for i-vector, CLNets and fusion	application of CLNets	
[50]	2018	voice conversion	TMIT, custom Chinese dataset	factorized hierarchical variational autoencoder		interesting work on voice conversion, performance measured by Speaker Verification scoring	not strictly topic-relevant
[61]	2018	speaker identification and verification	TIMIT, Librispeech	convolutional networks with raw speech input (SINCNET)	0.1–1 % lower EERs both in speaker identification and verification compared to DNN baseline; lower Speaker Verification EER on Librispeech: 0.32 %	application of raw speech input with convolutional networks	no comparison with x-vector or d-vector system