Neural PLDA Modeling for End-to-End Speaker Verification

Shreyas Ramoji, Prashant Krishnan, Sriram Ganapathy

Learning and Extraction of Acoustic Patterns (LEAP) Lab,
Department of Electrical Engineering, Indian Institute of Science, Bengaluru, India
{shreyasr, prashantkv1, sriramg}@iisc.ac.in

Abstract

While deep learning models have made significant advances in supervised classification problems, the application of these models for out-of-set verification tasks like speaker recognition has been limited to deriving feature embeddings. The state-of-art systems in speaker verification use a generative model based on probabilistic linear discriminant analysis (PLDA) for computing the verification score. Recently, we had proposed a neural network approach for backend modeling in speaker verification called the neural PLDA (NPLDA) where the likelihood ratio score of the generative PLDA model is posed as a discriminative similarity function and the learnable parameters of the score function are optimized using a verification cost. In this paper, we extend this work to achieve joint optimization of the embedding neural network (x-vector network) with the NPLDA network in an end-to-end (E2E) fashion. This proposed end-to-end model is optimized directly from the acoustic features with a verification cost function and during testing the model directly outputs the likelihood ratio score. With various experiments using the NIST speaker recognition evaluation (SRE) 2018 and 2019 datasets, we show that the proposed E2E model improves significantly over the state-of-art PLDA based speaker verification system.

Index Terms: NPLDA, End-to-End Systems, Speaker Verification

1. Introduction

Automatic speaker verification (ASV) has several applications such as voice biometrics for commercial applications, speaker detection in surveillance, speaker diarization, etc. A speaker is enrolled by a sample utterance(s), and the task of ASV is to detect whether the target speaker is present in a given test utterance or not. Several challenges have been organized over the years for benchmarking and advancing speaker verification technology such as the NIST speaker recognition Evaluation (SRE) challenge 2019 [1], the VoxCeleb speaker recognition challenge (VoxSRC) [2] and the VOiCES challenge [3]. The major challenges in speaker verification include the language mis-match in testing, short duration audio and the presence of noise/reverberation in the speech data.

The state-of-art systems in speaker verification use a model to extract embeddings of fixed dimension from utterances of variable duration. The earlier approaches based on unsupervised Gaussian mixture model (GMM) i-vector extractor [4] have been recently replaced with neural embedding extractors [5, 6] which are trained on large amounts of supervised speaker classification tasks. These fixed dimensional embeddings are pre-processed with a length normalization [7] technique followed by probabilistic linear discriminant analysis (PLDA) based back-end modeling approach [8].

In our previous work, we had explored a discriminative neural PLDA (NPLDA) approach [9] to backend modeling where a discriminative similarity function was used. The learnable parameters of the NPLDA model were optimized using an approximation of the minimum detection cost function (DCF). This model also showed good improvements in our SRE evaluations and the VOiCES from a distance challenge [10, 11]. In this paper, we extend this work to propose a joint modeling framework that optimizes both the front-end x-vector embedding model and the backend NPLDA model in a single end-to-end (E2E) neural framework. The proposed model is initialized with the pre-trained x-vector time delay neural network (TDNN) and the NPLDA E2E is fully trained on pairs of speech utterances starting directly from the mel frequency cepstral coefficient (MFCC) features. The advantage of this method is that both the embedding extractor as well as the final score computation is optimized on pairs of utterances and with the speaker verification metric. With experiments on the NIST SRE 2018 and 2019 datasets, we show that the proposed NLPDA E2E model improves significantly over the baseline system using x-vectors and generative PLDA modeling.

2. Related Prior Work

The common approaches for scoring in speaker verification systems include support vector machines (SVMs) [12], and the probabilistic linear discriminant analysis (PLDA) [8]. Some efforts on pairwise generative and discriminative modeling are discussed in [13, 14, 15]. The discriminative version of PLDA with logistic regression and support vector machine (SVM) kernels has also been explored in [16]. In this work, the authors use the functional form of the generative model and pool all the parameters needed to be trained into a single long vector. These parameters are then discriminatively trained using the SVM loss function with pairs of input vectors. The discriminative PLDA (DPLDA) is however prone to over-fitting on the training speakers and leads to degradation on unseen speakers in SRE evaluations [17]. The regularization of embedding extractor network using a Gaussian backend scoring has been investigated in [18]. Other recent developments in this direction includes efforts in using the approximate DCF metric for text dependent speaker verification [19].

Recently, some end-to-end approaches for speaker verification have been examined. For example, in [20], the PLDA scoring which is done with the i-vector extraction has been jointly derived using a deep neural network architecture and the entire model is trained using a binary cross entropy training criterion. In [21], a generalized end to end loss by minimizing the centroid means of within speaker distances while maximiz-

This work was funded by the Ministry of Human Resources Development (MHRD) of India and the Department of Science and Technology (DST) EMR/2016/007934 grant.

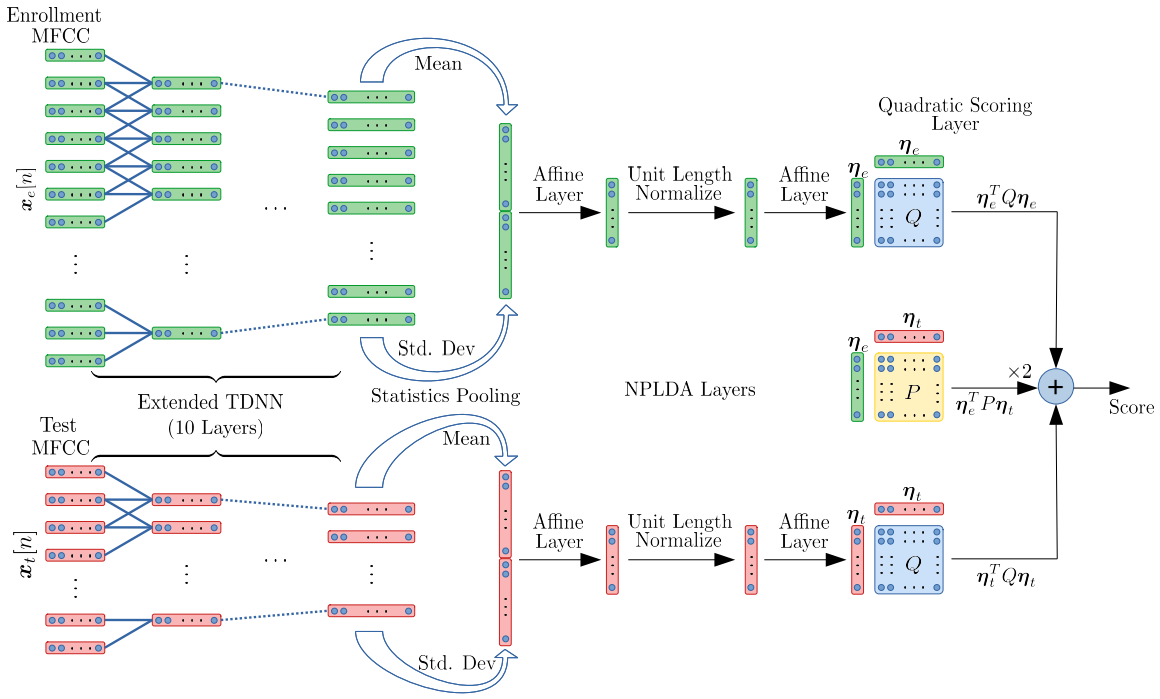

Figure 1: *End-to-End x-vector NPLDA architecture for Speaker Verification.*

ing across speaker distances. In another E2E effort, the use of triplet loss has been explored [22]. However, in spite of these efforts, most state of the art systems use a generative PLDA backend model with x-vector embeddings.

3. Background

3.1. Generative Gaussian PLDA (GPLDA)

The PLDA model on the processed x-vector embedding for a given recording is,

$$\boldsymbol{\eta}_r = \Phi\boldsymbol{\omega} + \boldsymbol{\epsilon}_r \tag{1}$$

where $\boldsymbol{\eta}_r$ is the x-vector for the given recording processed with centering, LDA transformation and a diagonalizing transformation which simultaneously diagonalizes the within and between speaker covariances, $\boldsymbol{\omega}$ is the latent speaker factor with a Gaussian prior of $\mathcal{N}(\mathbf{0}, \mathbf{I})$, Φ characterizes the speaker sub-space matrix and $\boldsymbol{\epsilon}_r$ is the residual assumed to have distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$. For scoring, a pair of embeddings, one from the enrollment recording $\boldsymbol{\eta}_e$ and one from the test recording $\boldsymbol{\eta}_t$ are used with the pre-trained PLDA model to compute the log-likelihood ratio score as,

$$s(\boldsymbol{\eta}_e, \boldsymbol{\eta}_t) = \boldsymbol{\eta}_e^\intercal \boldsymbol{Q} \boldsymbol{\eta}_e + \boldsymbol{\eta}_t^\intercal \boldsymbol{Q} \boldsymbol{\eta}_t + \boldsymbol{\eta}_e^\intercal \boldsymbol{P} \boldsymbol{\eta}_t + \text{const} \tag{2}$$

where,

$$\mathbf{Q} = \boldsymbol{\Sigma}_{tot}^{-1} - (\boldsymbol{\Sigma}_{tot} - \boldsymbol{\Sigma}_{ac}\boldsymbol{\Sigma}_{tot}^{-1}\boldsymbol{\Sigma}_{ac})^{-1} \tag{3}$$

$$\mathbf{P} = \boldsymbol{\Sigma}_{tot}^{-1}\boldsymbol{\Sigma}_{ac}(\boldsymbol{\Sigma}_{tot} - \boldsymbol{\Sigma}_{ac}\boldsymbol{\Sigma}_{tot}^{-1}\boldsymbol{\Sigma}_{ac})^{-1} \tag{4}$$

with $\boldsymbol{\Sigma}_{tot} = \Phi\Phi^T + \boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}_{ac} = \Phi\Phi^T$.

3.2. NPLDA

In the discriminative NPLDA approach [11], we construct the pre-processing steps of LDA as first affine layer, unit-length normalization as a non-linear activation and PLDA centering and diagonalization as another affine transformation. The final PLDA pair-wise scoring given in Eq. 2 is implemented as a quadratic layer in Fig. 1. Thus, the NPLDA implements the pre-processing of the x-vectors and the PLDA scoring as a neural backend.

3.2.1. Cost Function

To train the NPLDA for the task of speaker verification, we sample pairs of x-vectors representing target (from same speaker) and non-target hypothesis (from different speakers). The normalized detection cost function (DCF) [23] for a detection threshold θ is defined as:

$$C_{Norm}(\beta, \theta) = P_{Miss}(\theta) + \beta P_{FA}(\theta) \tag{5}$$

where β is an application based weight. defined as

$$\beta = \frac{C_{FA}(1 - P_{target})}{C_{Miss}P_{target}} \tag{6}$$

where C_{Miss} and C_{FA} are the costs assigned to miss and false alarms, and P_{target} is the prior probability of a target trial. P_{Miss} and P_{FA} are the probability of miss and false alarms respectively, and are computed by applying a detection threshold of θ to the log-likelihood ratios. A differentiable approximation of the normalized detection cost was proposed in [11, 19].

$$P_{Miss}^{(\text{soft})}(\theta) = \frac{\sum_{i=1}^{N} t_i \left[1 - \sigma(\alpha(s_i - \theta))\right]}{\sum_{i=1}^{N} t_i} \tag{7}$$

$$P_{FA}^{(\text{soft})}(\theta) = \frac{\sum_{i=1}^{N} (1 - t_i)\sigma(\alpha(s_i - \theta))}{\sum_{i=1}^{N} (1 - t_i)} \tag{8}$$

Here, i is the trial index, s_i is the system score and t_i denotes the ground truth label for trial i, and σ denotes the sigmoid function. By choosing a large enough value for the warping factor

α, the approximation can be made arbitrarily close to the actual detection cost function for a wide range of thresholds. The minimum detection cost (minDCF) is achieved at a threshold where the DCF is minimized.

$$\text{minDCF} = \min_{\theta} C_{Norm}(\beta, \theta) \tag{9}$$

The threshold θ is included in the set of learnable parameters of the neural network. This way, the network learns to minimize the minDCF through backpropagation.

4. End-to-end modeling

The model we explore is a concatenated version of two parameter tied x-vector extractor (TDNN network [24]) with the NPLDA model (Fig. 1). [1] The end-to-end model processes the mel frequency cepstral coefficients (MFCCs) of a pair of utterances to output a score. The MFCC features are passed through nine time delay neural network (TDNN) layers followed by a statistic pooing layer. The statistics pooling layer is followed by a fully connected layer with unit length normalization nonlinearity. This is followed by a linear layer and a quadratic layer as a function of the enrollment and test embeddings to output a score.

The parameters of the TDNN and the affine layers of the enrollment and test side of the E2E model are tied, which makes the model symmetric.

4.1. GPU memory considerations

We can estimate the memory required for a single iteration (batch update) of training as the sum of memory required to store the network parameters, gradients, forward and backward components of each batch. In this end-to-end network, each training batch of N trials can have upto $2N$ unique utterances assuming there are no repetitions. For simplicity, let us assume each of the utterances corresponds to T frames. We denote k_i to be the dimension of the input to the i^{th} TDNN layer, with a TDNN context of c_i frames. The total memory required can then be estimated as $2NT \sum_i k_i c_i \times 16$ bytes.. The GPU memory is limited by the total number of frames that go into the TDNN, which is denoted by the factor $2NT$. A large batchsize of 2048, as was used in [10], is infeasible for the end-to-end model (results in GPU memory load of 240GB). Hence, we resorted to a sampling strategy to reduce the GPU memory constraints.

4.2. Sampling of Trials

In this current work, in order to avoid memory explosion in the x-vector extraction stage of the E2E model, we propose to use a small number of utterances (64) in a batch with about 20 sec. of audio in each utterance. These 64 utterances are drawn from m speakers where m ranges from $3 - 8$. These 64 utterances are split randomly into two halves for each speaker to form enrollment and test side of trials. The MFCC features of the enrollment and test utterances are transformed to utterance embeddings η_e and η_t (as shown in Fig. 1). Each pair of enrollment, and test utterances is given a label as to whether the trial belongs to the target class (same speaker) or non-target class (different speakers). In this way, while the number of utterances is small, the number of trials used in the batch is 1024. Using the label information and the cost function defined in Eq. 5, the gradients are back-propagated to update the entire E2E model parameters.

This algorithm is applied separately to the male and female partitions of each training dataset to ensure the trials are gender and domain matched. All the 64 utterances used in a batch come from the same gender and same dataset (to avoid cross gender, cross language trials). The algorithm is repeated multiple times with different number of speakers (m), for the male and female partitions of every dataset. Finally, all the training batches are pooled together and randomized.

In contrast, the trial sampling algorithm used in our previous work on NPLDA [11, 10] was much simpler. For each gender of each dataset, we sample an enrollment utterance from a randomly sampled speaker, and sample another utterance from either the same speaker or a different speaker to get a target or a non-target trial. This was done without any repetition of utterances, to ensure that each utterance appears once per sampling epoch. This procedure was repeated numerous times for multiple datasets and for both genders to obtain the required number of trials. All the trials were then pooled together, shuffled and split into batches of 1024 or 2048 trials.

It is worth noting that the batch statistics of the two sampling methods are significantly different. A batch of trials in the previous sampling method (Algo. 1) can contain trials from multiple datasets and gender, whereas in the modified sampling method, which we will refer as Algo. 2, all the trials in a batch are from a particular gender of a particular dataset.

5. Experiments and Results

The work is an extension of our work in [10]. The x-vector model is trained using the extended time-delay neural network (E-TDNN) architecture described in [24]. This uses 10 layers of TDNNs followed by a statistics pooling layer. Once the network is trained, x-vectors of 512 dimensions are extracted from the affine component of layer 12 in the E-TDNN architecture. By combining the Voxceleb 1&2 dataset [2] with Switchboard, Mixer 6, SRE04-10, SRE16 evaluation set and SRE18 evaluation sets, we obtained with 2.2M recordings from 13539 speakers. The datasets were augmented with the 5-fold augmentation strategy similar to the previous models. In order to reduce the weighting given to the VoxCeleb speakers (out-of-domain compared to conversational telephone speech (CTS)), we also subsampled the VoxCeleb augmented portion to include only 1.2M utterances. The x-vector model is trained using 30 dimensional MFCC features using a 30-channel mel-scale filterbank spanning the frequency range 200 Hz - 3500 Hz,, mean-normalized over a sliding window of up to 3 seconds and with 13539 dimensional targets using the Kaldi toolkit. More information about the model can be found in [10].

The various backend PLDA models are trained on the SRE18 evaluation dataset. The evaluation datasets used include the SRE18 development and the SRE19 evaluation datasets. We perform several experiments under various conditions. The primary baseline to benchmark our systems is the Gaussian PLDA backend implementation in the Kaldi toolkit (GPLDA). The Kaldi implementation models the average embedding x-vector of each training speaker. The x-vectors are centered, dimensionality reduced using LDA to 170 dimensions, followed by unit length normalization.

In the traditional x-vector system, the statistic pooling layer computes the mean and standard deviation of the final TDNN layer. These two statistics then are concatenated into a fixed dimensional embedding. We also perform experiments where

[1] The implementation of this model can be found in `https://github.com/iiscleap/E2E-NPLDA`

Model	Duration of utterance	SRE18 Dev		SRE19 Eval	
		EER (%)	C_{Min}	EER (%)	C_{Min}
GPLDA (G1)	Full	6.43	0.417	6.18	0.512
GPLDA (G2)	20 secs	5.96	0.436	5.80	0.518
NPLDA (N1)	Full	5.33	0.389	5.10	0.443
NPLDA (N2)	20 secs	5.57	0.359	5.32	0.432

Table 1: *Performance comparison of training utterance durations (Full utterance vs 20 second segmenting) on GPLDA and NPLDA[10] models*

Model	Sampling	SRE18 Dev		SRE19 Eval	
		EER (%)	C_{Min}	EER (%)	C_{Min}
NPLDA (N2)	Algo. 1	5.57	0.359	5.32	0.432
NPLDA (N3)	Algo. 2	5.23	0.338	5.73	0.439

Table 2: *Performance comparison with different sampling techniques using NPLDA[10] method using previous sampling method (Algo. 1) and proposed new sampling method (Algo. 2)*

we use variance instead of the standard deviation and compare the performance.

In the following sections, we study the influence of reduced training duration, and provide a performance comparison of the sampling method (Algo.1 vs Algo.2). We then compare the performance of the generative Gaussian PLDA (GPLDA), Neural PLDA (NPLDA), and the proposed end-to-end approach (E2E). The PLDA backend training dataset used is the SRE18 Evaluation dataset. We report our results on the SRE18 Development set and the SRE19 Evaluation dataset using two cost metrics - equal error rate (EER) and minimum DCF (C_{Min}), which are the primary cost metrics for SRE19 evaluations.

5.1. Influence of training utterance duration

As discussed in Section 4.2, due to GPU memory considerations and ease of implementation, we create a modified dataset by splitting longer utterances into 20 second chunks (2000 frames) after voice activity detection (VAD) and mean normalization. We compare the performances of the models on the modified dataset and the original one. The results are reported in Table 1. We observe that the performance of the systems are quite comparable. This allows us to proceed using these conditions in the implementation of the End-to-End model. All subsequent reported models use 20 second chunks for PLDA training.

5.2. Comparison of sampling algorithms with NPLDA

The way the training trials are generated is crucial to how the model trains and its performance. The performance comparison of the two sampling techniques with PLDA models trained on SRE18 Evaluation dataset can be seen in Table 2. Although the nature of batch wise trials has changed significantly in terms of number of speakers in each batch and gender matched batches in the proposed new sampling method (Algo. 2), we see that its performance is comparable to our previous sampling method (Algo. 1).

Model	Pooling function	Init.	SRE18 Dev		SRE19 Eval	
			EER (%)	C_{Min}	EER (%)	C_{Min}
GPLDA (G2)	StdDev	-	5.96	0.436	5.80	0.518
GPLDA (G3)	Var	-	7.23	0.459	6.33	0.560
NPLDA (N2)	StdDev	G2	5.57	0.359	5.32	0.432
NPLDA (N4)	Var	G3	6.05	0.377	5.91	0.465
E2E (E1)	StdDev	N2	**5.36**	0.337	**5.31**	**0.405**
E2E (E2)	Var	N4	5.60	**0.307**	5.43	0.446

Table 3: *Performance comparison between GPLDA, NPLDA and E2E models using standard deviation and variance as the secondary pooling functions. The model that was used to initialize the network is denoted in the 3rd column*

5.3. End-to-End (E2E)

Using the proposed sampling method, we generate batches of 1024 trials using 64 utterances per batch. Both the NPLDA and E2E models were trained with this batch size. We use the Adam optimizer for the backpropagation learning. The performance of these models are reported in Table 3. The NPLDA model is initialized with the GPLDA model. The initialization details of the models along with the pooling functions are reported in the table. We compare performances using two different statistics (StdDev or Var). We observe significant improvements in NPLDA over the GPLDA system and subsequently in E2E system over the NPLDA. Comparing E2E and GPLDA when we use standard deviation as the pooling function, we observe relative improvements of about 23% and 22% in SRE18 development and SRE19 evaluation sets, respectively in terms of the C_{Min} metric. The relative improvements between E2E and GPLDA when we use Var as the pooling function are about 33% and 20% for SRE18 development and SRE19 evaluation sets, respectively for the C_{Min} metric. Though, the cost function in the neural network aims to minimize the detection cost function (DCF), we also see improvements in the EER metric using the proposed approach. These results show that the joint E2E training with a singal neural pipeline and optimization results in improved speaker recognition performance.

6. Summary and Conclusions

This paper explores a step in the direction of a neural End-to-End (E2E) approach in speaker verification tasks. It is an extension of our work on a discriminative neural PLDA (NPLDA) backend. The proposed model is a single elegant end-to-end approach that optimizes directly from acoustic features like MFCCs with a verification cost function to output a likelihood ratio score. We discuss the influence of the factors that were key in implementing the E2E model. This involved modifying the duration of the training utterance and developing a new sampling technique to generate training trials. The model shows considerable improvements over the generative Gaussian PLDA and the Neural PLDA in the NIST SRE 2018 and 2019 datasets.

Future work in this direction could include investigating better sampling algorithms such as the use of curriculum learning [25], different loss functions, improved architecture for the embedding extractor using attention and other sequence models such as LSTMs, use of more data, etc.

7. References

[1] O. Sadjadi, "NIST 2019 Speaker Recognition Evaluation: CTS Challenge - Evaluation Plan," https://www.nist.gov/system/files/documents/2019/07/22/2019_nist_speaker_recognition_challenge_v8.pdf.

[2] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: A Large-Scale Speaker Identification Dataset," in *Proc. Interspeech*, 2017, pp. 2616–2620. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2017-950

[3] M. K. Nandwana, J. van Hout, C. Richey, M. McLaren, M. A. Barrios, and A. Lawson, "The VOiCES from a Distance Challenge 2019," in *Proc. Interspeech 2019*, 2019, pp. 2438–2442. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-1837

[4] 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, 2010.

[5] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *Spoken Language Technology Workshop (SLT), 2016 IEEE*. IEEE, 2016, pp. 165–170.

[6] 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)*. IEEE, 2018, pp. 5329–5333.

[7] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of I-Vector Length Normalization in Speaker Recognition Systems," in *Proc. Interspeech*, 2011, pp. 249–252.

[8] P. Kenny, "Bayesian Speaker Verification with Heavy-Tailed Priors," in *Odyssey*, 2010, pp. 14–21.

[9] S. Ramoji, V. Krishnan, P. Singh, S. Ganapathy *et al.*, "Pairwise Discriminative Neural PLDA for Speaker Verification," *arXiv preprint arXiv:2001.07034*, 2020.

[10] S. Ramoji, P. Krishnan, B. Mysore, P. Singh, and S. Ganapathy, "Leap system for sre19 challenge–improvements and error analysis," *arXiv preprint arXiv:2002.02735*, 2020.

[11] S. Ramoji, P. Krishnan, and S. Ganapathy, "Nplda: A deep neural plda model for speaker verification," *arXiv preprint arXiv:2002.03562*, 2020.

[12] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE signal processing letters*, vol. 13, no. 5, pp. 308–311, 2006.

[13] S. Cumani, N. Brümmer, L. Burget, P. Laface, O. Plchot, and V. Vasilakakis, "Pairwise Discriminative Speaker Verification in the i-Vector Space," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 6, pp. 1217–1227, 2013.

[14] S. Cumani and P. Laface, "Large-Scale Training of Pairwise Support Vector Machines for Speaker Recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 22, no. 11, pp. 1590–1600, 2014.

[15] ——, "Generative pairwise models for speaker recognition," in *Odyssey*, 2014, pp. 273–279.

[16] L. Burget, O. Plchot, S. Cumani, O. Glembek, P. Matějka, and N. Brümmer, "Discriminatively trained probabilistic linear discriminant analysis for speaker verification," in *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2011, pp. 4832–4835.

[17] J. Villalba, N. Chen, D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, J. Borgstrom, L. P. García-Perera, F. Richardson, R. Dehak *et al.*, "State-of-the-art speaker recognition with neural network embeddings in nist sre18 and speakers in the wild evaluations," *Computer Speech & Language*, vol. 60, p. 101026, 2020.

[18] L. Ferrer and M. McLaren, "Optimizing a Speaker Embedding Extractor Through Backend-Driven Regularization," in *Proc. Interspeech 2019*, 2019, pp. 4350–4354. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-1820

[19] V. Mingote, A. Miguel, D. Ribas, A. Ortega, and E. Lleida, "Optimization of False Acceptance/Rejection Rates and Decision Threshold for End-to-End Text-Dependent Speaker Verification Systems," in *Proc. Interspeech 2019*, 2019, pp. 2903–2907. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2550

[20] J. Rohdin, A. Silnova, M. Diez, O. Plchot, P. Matějka, and L. Burget, "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)*. IEEE, 2018, pp. 4874–4878.

[21] L. Wan, Q. Wang, 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)*. IEEE, 2018, pp. 4879–4883.

[22] C. Zhang and K. Koishida, "End-to-end text-independent speaker verification with triplet loss on short utterances," in *Proc. Interspeech 2017*, 2017, pp. 1487–1491. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2017-1608

[23] D. A. Van Leeuwen and N. Brümmer, "An introduction to application-independent evaluation of speaker recognition systems," in *Speaker classification I*. Springer, 2007, pp. 330–353.

[24] D. Snyder, D. Garcia-Romero, G. Sell, A. McCree, D. Povey, and S. Khudanpur, "Speaker recognition for multi-speaker conversations using x-vectors," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5796–5800.

[25] S. Ranjan and J. H. Hansen, "Curriculum learning based approaches for noise robust speaker recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 1, pp. 197–210, 2017.