INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Investigation of NICT submission for short-duration speaker verification challenge 2020

Peng Shen, Xugang Lu, Hisashi Kawai

National Institute of Information and Communications Technology, Japan

peng.shen@nict.go.jp

Abstract

In this paper, we describe the NICT speaker verification system for the text-independent task of the short-duration speaker verification (SdSV) challenge 2020. We firstly present the details of the training data and feature preparation. Then, x-vector-based front-ends by considering different network configurations, back-ends of probabilistic linear discriminant analysis (PLDA), simplified PLDA, cosine similarity, and neural network-based PLDA are investigated and explored. Finally, we apply a greedy fusion and calibration approach to select and combine the subsystems. To improve the performance of the speaker verification system on short-duration evaluation data, we introduce our investigations on how to reduce the duration mismatch between training and test datasets. Experimental results showed that our primary fusion yielded minDCF of 0.074 and EER of 1.50 on the evaluation subset, which was the 2nd best result in the text-independent speaker verification task.
Index Terms: speaker verification, short duration, SdSV challenge

1. Introduction

Speaker verification refers to the task of automatically determining whether a test segment was spoken by a target speaker or not. [1, 2]. Speaker verification techniques have been studied for several decades and begin to play an important role in many e-commerce applications as well as in logical and physical access control [3], forensics [4], and law enforcement. However, there are still many challenges to solve to create truly robust systems and to integrate them into applications that can perform effectively in the real world. One of the challenges of current speaker verification systems is short duration problem. Short duration-based speaker verification is an important task since users of the system prefer short utterances for enrollment and authentication. Previous works of speaker verification on short-duration evaluation conditions have been reported [5, 6, 7].

In this paper, we describe the NICT submission to the Short-duration Speaker Verification (SdSV) Challenge 2020. The main goal of the SdSV challenge 2020 is to evaluate new technologies for text-dependent and text-independent speaker verification in a short duration scenario. The proposed challenge evaluates SdSV with varying degree of phonetic overlap between the enrollment and test utterances (cross-lingual). It is the first challenge with a broad focus on systematic benchmark and analysis on varying degrees of phonetic variability on short-duration speaker recognition. The SdSV challenge 2020 includes two tasks, where task 1 is defined as speaker verification in text-dependent mode: given a test segment of speech and the target speaker's enrollment data, automatically determine whether a specific phrase and the test segment was spoken by the target speaker. Task 2 is speaker verification in text-independent mode: given a test segment of speech and the target speaker enrollment data, automatically determine whether the test segment was spoken by the target speaker. Our work focus on the text-independent task, i.e., task 2.

For a short-duration-based speaker verification system, one of the challenges is the duration mismatch of the training, enrollment, and evaluation data. By analyzing the duration distribution of the evaluation data, we firstly investigated the influence of the data duration mismatch on front-ends and back-ends. Then, we built x-vector-based speaker embedding systems as front-ends. The idea of speaker embedding is to find representation for speaker idiosyncrasy base on the specific hidden layer of neural networks. By using speaker embedding, variable-length utterances are converted to fixed-dimensional embedding vectors. The embedding vectors are used to decide on true speakers or imposters. Previous works showed that time-delay neural networks (TDNN) [8] and extended TDNN (E-TDNN) [9] are effective networks for speaker embedding extraction. In this work, we also investigated factorized TDNN (F-TDNN) [10], TDNN followed with long short-term memory (TDNN-LSTM) recurrent neural networks, convolutional neural network TDNN (CNN-TDNN), and residual networks (ResNet). The cross-entropy with softmax is arguably one of the most commonly used loss functions to train the speaker embedding neural network. In our work, besides softmax function, we investigated angular softmax (ASoftmax) [11], additive angular margin softmax (ArcSoftmax), and additive margin softmax (AMSoftmax) functions [12]. After speaker embeddings were extracted, several probabilistic linear discriminant analysis (PLDA)-based back-ends and cosine similarity were used for scoring. Then, adaptive symmetric score normalization (AS-Norm) [13] was used to produce well-calibrated speaker verification scores. Finally, a greedy fusion method was used to obtain the final score for submission.

This paper is organized as follows. In section 2, we introduce the datasets, data preparation, and feature extraction for the text-independent task of the SdSV challenge 2020. Network configurations and objective functions used for speaker embedding front-ends are introduced in Section 3. We describe our back-ends in Section 4. In section 5, the greedy fusion and calibration are introduced. We summarize the investigation results in section 6. Conclusions are drawn in section 7.

2. Datasets and feature extraction

2.1. Training data

The SdSV challenge 2020 is a fixed training condition task where the system should only be trained using a designated set. The fixed training set consists of VoxCeleb 1 and 2, LibriSpeech, and DeepMine datasets. We summarize the training

The work is partially supported by JSPS KAKENHI No. 19K12035

http://dx.doi.org/10.21437/Interspeech.2020-2351

Dataset	#utt	#speaker	Languages
VoxCeleb 1 & 2	1,282K	7363	Multiple
LibriSpeech	292K	5831	English
DeepMine	85K	588	Persian
Enrollment data	111K	15555	Persian
Test data	69K	-	Persian and English

Table 1: *Datasets of task 2.*

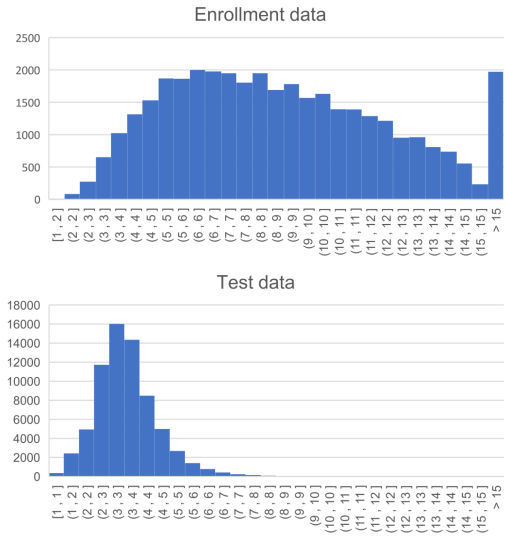

Figure 1: *Duration distribution of enrollment and test data.*

datasets in Table 1. DeepMine data for task 2 are the in-domain training data contains text-independent Persian utterances from 588 speakers. Non-speech data is allowed for data augmentation purposes. Other public or private speech data and task 1 in-domain data for task 2 training are forbidden.

2.2. Enrollment and test data

We list the enrollment and test data in Table 3. The enrollment data in task 2 consists of one to several variable-length utterances. The net speech duration for each model is roughly 3 to 120 seconds (after applying an energy-based VAD). Each trial in the evaluation contains a test utterance and a target model. The duration of the test utterances varies between 1 to 8 seconds. Figure 1 illustrates the duration distribution of the enrollment and test data. The duration average and standard deviation for the enrollment data are 8.7 and 3.6, and they are 3.2 and 1.1 for the test data. The whole set of trials is divided into two subsets: a progress subset (30%), and an evaluation subset (70%). The progress subset is used to monitor progress on the leaderboard. The evaluation subset is used to generate the official results at the end of the challenge.

2.3. Data preparation and feature extraction

We followed the training data preparation of the baseline x-vector system supplied by the SdSV challenge organizer. We firstly combined the VoxCeleb, LibriSpeech, and DeepMine in-domain data as the x-vector extractor training data. Then, data augmentation (additive noise, music, babble, and reverberation)

Layer No.	Layer Type	Context	Size
1	TDNN-ReLU	t-2:t+2	512
2	Dense-ReLU	t	512
3	TDNN-ReLU	t-2, t, t+2	512
4	Dense-ReLU	t	512
5	TDNN-ReLU	t-3, t, t+3	512
6	Dense-ReLU	t	512
7	TDNN-ReLU	t-4, t, t+4	512
8	Dense-ReLU	t	512
9	Dense-ReLU	t	1500
10	Pooling (mean+stddev)	Full-seq	2x1500
11	Dense(Embedding)-ReLU		512
12	Dense-ReLU		512
13	Dense-Softmax		#spks.

Table 2: *Extended TDNN x-vector architecture.*

as described in [14] was used on the whole training data. Because of this task focused on short-duration test data, to reduce the duration mismatch of the training data and test data, we picked up examples with 2 seconds (200 frames) for network training.

Three types of acoustic features were applied, i.e., the Mel-frequency cepstral coefficient (MFCC), perceptual linear predictive cepstrum (PLP), a log Mel-filter bank (FBANK). MFCC features were computed using 30 Mel-filter banks. The PLP analysis computed 20-order PLP-cepstra. FBANK features were estimated using 40 and 60 Mel-filter banks. The feature extraction was progressed with a frame window of 25 ms and a frame shift of 10 ms. The frames of silence and low signal-to-noise ration were removed with an energy-based voice activity detection (VAD) after doing feature extraction.

3. Speaker embedding front-ends

The model for extracting speaker embedding representations consists of three modules: a frame-level feature extractor, a statistics pooling layer, and utterance-level representation layers. The model is trained to classify the speakers in the training dataset. Various neural networks can be used to calculate the frame-level representation, e.g., TDNN [8] or CNN [15, 16]. In this work, by fixing the statistics pooling layer and utterance-level representation layers, we investigated the frame-level feature extractor with several neural networks for extracting the speaker embedding.

3.1. TDNN and E-TDNN

TDNN is the most commonly used for x-vector extraction [8]. An extended TDNN architecture (E-TDNN) has been shown its effectiveness for extracting x-vectors [17]. Our TDNN network includes three time-delay layers and two fully connected layers. There are 512 channels except the last one, which has 1500 channels. The kernel sizes are 5, 3, and 3; and dilation factors are 1, 2, and 3 for time-delay layers, respectively.

Compared with TDNN, E-TDNN consists of one more time-delay layer and three fully-connected layers. The new fully connected layers are inserted into every two time-delay layers. The kernel sizes are 5, 3, 3, and 3; and dilation factors are 1, 2, 3, and 4, respectively. Therefore, the temporal context of E-TDNN is wider than that of TDNN. And E-TDNN has more parameters. Table 2 illustrates the configuration of the E-TDNN network.

3.2. ResNet

In our ResNet configuration, we replaced the TDNN network with a ResNet34 network [18]. A channel average pooling was applied to the output of the final layer of the ResNet. The dimension of the average pooling was 512. Then, the statistic pooling and utterance-level representation were processed.

3.3. Other networks

We also borrowed some effective networks from speech recognition tasks. For example, the factorized TDNN (F-TDNN) [10] showed its effectiveness on many speech recognition tasks than the conventional TDNN network. We evaluated F-TDNN [1], TDNN-LSTM [2] and TDNN-LSTM with attention [3], and CNN-TDNN [4] networks [19]. In the speech recognition task, these networks had an L2 regularization setting to overcome overfitting, in this task, we removed the regularization setting.

3.4. Objective function

In conventional speaker embedding training, softmax-based categorical cross-entropy is commonly used as the objective loss function. In this work, besides softmax, we also evaluated angular softmax (ASoftmax) [11], additive angular margin softmax (ArcSoftmax) and additive margin softmax (AMSoftmax)-based functions [12].

4. Back-ends

With the extracted embedding vectors, we firstly applied in-domain global mean subtraction on training, enrollment, and test data. Then, linear discriminant analysis (LDA) was used to select the most speaker relevant feature and reduce the dimension of the original x-vector. To further reduce the variabilities between training data and testing data, in-domain whitening was applied before classifier. The in-domain whitening calculated the mean and covariance of the in-domain data and applied them to whiten the training and test data similar to [20]. Finally, length normalization was applied to the speaker discriminant vectors.

The first classifier was the Gaussian PLDA [21] with a full covariance residual noise term and a full-rank eigenvoice subspace. A simplified PLDA with 150 eigenvoices was also investigated. Finally, we further investigated cosine similarity and a neural PLDA (NPLDA) [22]. The parameters of a PLDA system were used to initialize the NPLDA model, then the parameters were trained in a backpropagation setting. We used the DeepMine training data and their augmented data as the in-domain data. The LDA dimension was selected as 150 for cosine similarity and 200 for other classifiers.

After scoring, all trial results were subject to score normalization. We utilized adaptive symmetric score normalization (AS-Norm) [13] in our systems. And, we evaluated two variants of AS-Norm with different adaptive cohort selection.

5. Fusion and calibration

We implemented a greedy fusion algorithm to obtain the final submission. Algorithm 1 illustrates our greedy fusion method. Firstly, all the subsystems are evaluated to obtain minDCF and EER values. Then, the top N best subsystems are selected as the candidate list. After that, we prepare new lists by adding a new subsystem to the candidate list. The linear logistic regression with the Bosaris toolkit [23] is used to fuse and evaluate the new lists. Then, the candidate list is updated by selecting the top N best lists. The final submission is obtained when there is no further improvement. In this work, N was set to 3.

[1] egs/swbd/s5c/local/chain/tuning/run_tdnn_7r.sh

[2] egs/swbd/s5c/local/chain/tuning/run_tdnn_lstm_1n.sh

[3] egs/tedlium/s5_r2/local/chain/tuning/run_tdnn_lstm_attention_bs_1b.sh

[4] egs/swbd/s5c/local/chain/tuning/run_cnn_tdnn_1a.sh

Datasets	MinDCF	EER
Vox	0.249	5.91
Libri	0.347	8.19
DM	0.483	12.00
Vox & Libri	0.227	5.32
Vox & DM	0.236	5.51
Libri & DM	0.319	7.39
Vox & Libri & DM	0.214	4.98

Table 3: *Investigation of training datasets on evaluation subset. (Vox means VoxCeleb 1 and 2; Libri means LibriSpeech; DM means DeepMine in-domain training data.)*

Algorithm 1 : Greedy fusion algorithm.

Step 1: Subsystem evaluation
Calculate minDCF and EER for all the subsystems.
Select the top N best subsystems with minDCF+EER.
Step 2: Fusion and calibration
for number of fusion iterations **do**
    Prepare lists by adding one subsystem to each list.
    Parallelized implement fusion on the new lists.
    Select the top N best lists with minDCF+EER.
    Decide to CONTINUE or END based on improvement.
end for

6. Performance metric and results

6.1. Performance metric

The SdSV challenge uses the normalized minimum Detection Cost Function (minDCF) and EER as performance metrics. This detection cost function for minDCF is defined as a weighted sum of miss and false alarm error probabilities:

$$C_{Det} = C_{Miss} \times P_{Miss|Target} \times P_{Target} + C_{FalseAlarm} \times P_{FalseAlarm|NonTarget} \times (1 - P_{Target}) \quad (1)$$

where $C_{Miss} = 10$, $C_{FalseAlarm} = 1$, and $P_{Target} = 0.01$. Based on the parameters, the normalized DCF (DCF_{norm}) will be DCF divide by 0.1 as the best cost that could be obtained without processing the input data.

6.2. Investigation on the training dataset and features

The whole training data consists of three datasets, they are VoxCeleb 1 and 2, LibriSpeech, and DeepMine in-domain data. We investigated these three datasets by using them separately or their combinations to train speaker embedding networks. This investigation was done with the E-TDNN network and PLDA, where PLDA was built with DeepMine in-domain data. Table 3 shows the results of this investigation. From the results, we can see that using VoxCeleb data to train the embedding network obtained the best result among these three datasets. The performance of the model trained with VoxCeleb even outperformed the model trained with in-domain DeepMine data. Using the combination of all three datasets obtained the best performance.

We investigated three types of features for training speaker embedding extraction networks. Table 4 shows the compari-

Feature	MinDCF	EER
MFCC	0.227	5.39
PLP	0.244	5.87
FBANK40	0.215	5.07
FBANK60	0.208	4.89

Table 4: *Investigation of MFCC, PLP and FBANK features on progress subset. (E-TDNN and PLDA were used.)*

	MFCC		FBANK40	
Network	MinDCF	EER	MinDCF	EER
TDNN	0.223	5.32	0.211	5.02
E-TDNN	0.219	5.19	0.215	5.08
F-TDNN	0.290	7.26	0.260	6.63
TDNN-LSTM	0.232	5.56	-	-
TDNN-LSTM-Att	0.248	5.60	0.235	4.97
CNN-TDNN	0.278	6.82	0.267	6.39
ResNet	0.263	6.04	0.193	4.38

Table 5: *Front-ends investigation: Network configurations for speaker embedding extractor on progress subset.*

son of MFCC, PLP, and FBANK with 40 and 60 dimensions. The E-TDNN network and PLDA back-end were used in this investigation. From the results, we can see that network trained with 60-dimensional filter bank features outperformed that using MFCC, PLP, and 40-dimensional filter bank.

6.3. Investigation on speaker embedding network

Table 5 illustrates the comparison of different network configurations of speaker embedding extractor. The results for both MFCC and FBANK features are listed. The TDNN-LSTM with FBANK40 was not trained successfully. With the MFCC feature, the E-TDNN network obtained the best results on both minDCF and EER. However, the ResNet network outperformed others with minDCF 0.193 and EER of 4.38. For all the networks, using the FBANK features obtained better performance. Table 6 shows the results of investigation on objective loss functions. Compared with other functions, AMSoftmax showed its effectiveness on both MFCC and FBANK features.

6.4. Investigation on back-ends classifier

The investigation results on back-ends are listed in Table 7. From the results, we can see that the cosine similarity-based classifier even outperformed the PLDA classifier. The new NPLDA performed the best within all the back-end classifiers. In this table, we also list the results with score normalization. AS-Norm 1 means using the DeepMine in-domain data as the cohort, and AS-Norm 2 means the test data was used.

As shown in Figure 1, the duration distribution of enrollment and test datasets has a mismatch problem. To reduce this mismatch, we reorganized the enrollment data by picking up shorter utterances from the original data. "NPLDA (short)" is the system with the reorganized enrollment data. From the results, we can see that "NPLDA (short)" obtained better results than NPLDA that shows reducing duration mismatch between enrollment and test is an effective method.

	MFCC		FBANK40	
Loss func.	MinDCF	EER	MinDCF	EER
Softmax	0.227	5.39	0.215	5.07
ASoftmax	0.215	5.06	0.203	4.72
ArcSoftmax	0.210	4.92	0.212	4.94
AMSoftmax	0.203	4.84	0.194	4.58

Table 6: *Front-ends investigation: objective loss functions of E-TDNN network on progress subset.*

	Softmax		AMSoftmax	
Back-ends	MinDCF	EER	MinDCF	EER
PLDA	0.215	5.07	0.194	4.58
Cosine	0.184	4.04	0.167	3.71
SPLDA	0.220	4.94	0.185	4.04
NPLDA	0.162	3.60	0.139	3.10
NPLDA (AS-Norm1)	0.159	3.33	0.136	2.85
NPLDA (AS-Norm2)	0.163	3.45	0.139	2.95
NPLDA (short)	0.152	3.39	0.137	3.07

Table 7: *Back-ends investigation: Results of different back-ends with x-vector obtained from E-TDNN network. (FBANK40 feature was used; Results were reported on progress subset.)*

Network	Feature	Loss func.	Back-ends	MinDCF	EER
E-TDNN	FBANK40	Softmax	PLDA	0.215	5.07
E-TDNN	FBANK40	AMSoftmax	PLDA	0.194	4.58
E-TDNN	FBANK40	AMSoftmax	NPLDA	0.139	3.10
E-TDNN	FBANK40	AMsoftmax	Fusion	0.124	2.67
E-TDNN	FBANK40	Fusion	NPLDA	0.131	2.74
E-TDNN	Fusion	AMSoftmax	NPLDA	0.124	2.64
E-TDNN	FBANK40	Fusion	Fusion	0.113	2.41
E-TDNN	Fusion	Fusion	Fusion	0.111	2.33
Fusion	Fusion	Fusion	Fusion	0.075	1.51
Primary submission (evaluation subset)				0.074	1.50

Table 8: *Fusion investigation on progress subset and results.*

6.5. Fusion investigation and results

Figure 8 shows the investigation of fusion on features, loss functions, and back-ends. Limited by the time consumption of training, some models have not been trained, such as some ResNet- and FBANK60-based models, therefore, this investigation is based on the E-TDNN with FBANK40. From the results, we can see that different features and back-ends obtained almost the same contribution when fusion was applied. A combination of different features, loss functions, and back-ends could further improve the performance. Our primary submission was obtained using the proposed greedy fusion method.

7. Conclusions

In this study, we analyzed the NICT submission for the SdSV challenge 2020. Our systems are based on x-vector-based speaker embedding with several neural network configurations. We investigated the influence of MFCC, PLP, and FBANK features; and PLDA, SPLDA, cosine similarity, and NPLDA back-ends. The final submission was obtained with a greedy fusion algorithm. Our investigations showed that ResNet-based front-end with AMSoftmax, NPLDA back-ends with AS-Norm are effective techniques for building a single system. Our primary submission yielded minDCF of 0.074 and EER of 1.50 on the evaluation subset, which was the 2nd best result in the text-independent task.

8. References

[1] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: from features to supervectors," *Speech Communication*, vol. 52, no. 1, pp. 12-40, 2010.

[2] J. H. L. Hansen and T. Hasan, "Speaker recognition by machines and humans: a tutorial review," *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 74-99, 2015.

[3] K. A. Lee, B. Ma, and H. Li, "Speaker verification makes its debut in smartphone," *IEEE Signal Processing Society Speech and language Technical Committee Newsletter*, 2013.

[4] J. F. Bonastre, J. Kahn, S. Rosatto, and M. Ajili, "Forensic speaker recognition: mirages and reality," *Individual Differences in Speech Production and Perception*, 2015.

[5] M. McLaren, R. Vogt, B. Baker, and S. Sridharan, "Experiments in SVM-based speaker verification using short utterances," in Proc. of *Odyssey*, 2010.

[6] A. Kanagasundaram, R. Vogt, D. Dean, S. Sridharan, and M. Mason, "i-vector based speaker recognition on short utterances," in Proc. of *Interspeech*, 2011.

[7] A. Kanagasundaram, R. J. Vogt, D. B. Dean, and S. Sridharan, "PLDA based speaker recognition on short utterances," in Proc. of *Odyssey*, 2012.

[8] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in Proc. of *Interspeech*, 2017.

[9] D. Snyder, D. Garcia-Romero, G. Sell, A. McCree, D. Povey, and S. Khudanpur, "Speaker Recognition for Multi-Speaker Conversations Using X-Vectors," in Proc. of *ICASSP*, 2019.

[10] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohamadi, and S. Khudanpur, "Semi-Orthogonal Low-Rank Matrix Factorization for Deep Neural Networks," in Proc. of *Interspeech*, 2018.

[11] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in Proc. of *CVPR*, 2017.

[12] Y. Liu, L. He, and J. Liu, "Large margin softmax loss for speaker verification," in Proc. of *Interspeech*, 2019.

[13] P. Matjka, O. Novotn, O. Plchot, L. Burget, M. D. Snchez, and J. ernock, "Analysis of Score Normalization in Multilingual Speaker Recognition," in Proc. of *Interspeech*, 2017.

[14] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in Proc. of *ICASSP*, 2018.

[15] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, "Deep speaker: an end-to end neural speaker embedding system," arXiv preprint arXiv:1705.02304, 2017.

[16] A. Nagrani, J. Chung, and A. Zisserman, "VoxCeleb: A large-scale speaker identification dataset," in Proc. of *Interspeech*, 2017.

[17] D. Snyder, et al., "The JHU Speaker Recognition System for the VOiCES 2019 Challenge," in Proc. of *Interspeech*, 2019.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. of *CVPR*, 2016.

[19] D. Povey, et al., "The Kaldi speech recognition Toolkit," in Proc. of *ASRU*, 2011.

[20] B. Sun, J. Feng and K. Saenko, "Return offrustratingly easy domain adaptation," in Proc of *AAAI*, 2016.

[21] S. Ioffe, "Probabilistic linear discriminant analysis," in Proc. of *the 9th European Conference on Computer Vision*, 2006.

[22] S. Ramoji, P. Krishnan, and S. Ganapathy, "NPLDA: A Deep Neural PLDA Model for Speaker Verification," in Proc. of *Odyssey* 2020.

[23] N. Brummer and E. De Villiers, "The BOSARIS Toolkit: Theory, Algorithms and Code for Surviving the New DCF," in *NIST SRE11 Speaker Recognition Workshop*, Atlanta, Georgia, USA, 2011.