INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Optimization of False Acceptance/Rejection Rates and Decision Threshold for End-to-End Text-Dependent Speaker Verification Systems

Victoria Mingote, Antonio Miguel, Dayana Ribas, Alfonso Ortega, Eduardo Lleida

ViVoLab, Aragón Institute for Engineering Research (I3A), University of Zaragoza, Spain
{vmingote,amiguel,dribas,ortega,lleida}@unizar.es

Abstract

Currently, most Speaker Verification (SV) systems based on neural networks use Cross-Entropy and/or Triplet loss functions. Despite these functions provide competitive results, they might not fully exploit the system performance, because they are not designed to optimize the verification task considering the performance measures, e.g. the Detection Cost Function (DCF) or the Equal Error Rate (EER). This paper proposes a first approach to this issue through the optimization of a loss function based on the DCF. This mechanism allows the end-to-end system to directly manage the threshold used to compute the ratio between the False Rejection Rate (FRR) and the False Acceptance Rate (FAR). This way connecting the system training directly to the operating point. Results in a text-dependent speaker verification framework, based on neural network supervectors over the RSR2015 dataset, outperform reference systems using Cross-Entropy and Triplet loss, as well as our previously proposal based on an approximation of the Area Under the Curve (*aAUC*).

Index Terms: Speaker Verification, loss functions, Cross-Entropy, aAUC, Triplet loss

1. Introduction

The main purpose of SV systems is to verify whether a claimed identity is true or false. State-of-the-art SV systems are trained to generally make two decisions, acceptance or rejection. There are two types of decision errors, the FAR referred to accept an impostor speaker (Type I error), and the FRR related to the incorrect rejection of a true speaker (Type II error) [1, 2]. These errors compared to a threshold determines the system performance. The threshold selection is what actually relates the system to the interesting operating point in terms of the application. Thus, when FAR = FRR we have the Equal Error Rate (EER), which is an operating point frequently used as a measure of the system discrimination capability, specially for commercial applications [3]. However, the EER may not be the best option for some applications, so there are alternative operating points that implement a trade-off between FAR and FRR. Once verification scores are computed, a following stage of the system considers the cost of the decision errors and readjust the operating point to the requirements of the application. This process is known as calibration [4], which represents the power of the system to choose the optimal decision threshold. However, the fact is that the verification system itself is trained without considering the real operating point.

State-of-the-art neural networks SV systems [5, 6, 7] use a front-end trained with a multi-class Cross-Entropy loss function and an average pooling mechanism which produces embeddings that represent the whole utterance. Then, the verification process is performed through a separated back-end, either through a Probabilistic Linear Discriminative Analysis (PLDA) [8, 9], a similarity metric [10], a Triplet Neural Network [11, 12], or more complex methods like the Angular loss [13]. Finally, verification results are rectified through score normalization (S-norm) [14], and a calibration [4] in order to choose an optimal threshold for detection. Although these systems already provide reasonably good results, the loss functions used, namely Cross-Entropy and Triplet loss, are general approaches which were not designed to optimize the verification task itself.

Previously [15] we proposed an alternative back-end which combines the triplet loss philosophy with the optimization of the AUC as loss function [16, 17, 18]. The AUC is a measure of the probability that all pairs of examples are ranked correctly. Thus, this provides a performance measure independent of the operating point. We demonstrated how the maximization of our approximation of the AUC (*aAUC*) improves the overall system performance in terms of EER. However, *aAUC* focuses on the discrimination performance, while we cannot be sure that the system is well calibrated at the end. Moreover, the triplet strategies of training employed are very slow and have a high computational cost.

In this paper, we propose a new loss function to replace the classical Cross-Entropy, which is usually employed to make a multi-class classification. This function is inspired by the DCF [19, 20] used for National Institute of Standards and Technology during the Speaker Recognition Evaluations (NIST-SRE). It measures the cost of detection error in terms of FA and FR, so we will call it *aDCF* loss function. We have approximated the DCF by a differentiable function, which allows the training algorithm to adapt the network parameters to minimize the cost and learn the optimal decision threshold. Preliminary results outperform reference systems based on alternative loss functions: Cross-Entropy, Triplet loss, and *aAUC*. The capability to manage the threshold regarding the operating point is a useful skill to provide this type of end-to-end systems with. Experimental results also show that this function provides a better FRR when the FAR is low, which in practice is a desirable quality for a commercial SV system.

From now on, Sections 2, 3 present a review of loss functions. Section 4 presents the proposal. Section 5 describes the SV system description, followed by the experimental setup in Section 6. Finally, Section 7 presents and discusses results and Section 8 concludes the paper.

2. Loss Functions

Most SV systems based on DNN are designed from the following two strategies:

1. An architecture trained with a multi-class classification philosophy using "traditional" loss functions such as the Cross-Entropy.
2. An architecture that produces speech embeddings followed by a trainable back-end with a triplet neural net-

http://dx.doi.org/10.21437/Interspeech.2019-2550

work structure and loss functions, such as Triplet loss function or *aAUC*.

Each of these functions are briefly described below.

2.1. Cross-Entropy Loss

One of the most common used loss functions in many SV systems based on DNNs is the Cross-Entropy loss [21, 22, 23]. Due to its simplicity and probabilistic interpretation, this function has been widely applied for the multi-class classification. The input to Cross-Entropy loss function is used as feature in many systems [6]. The Cross-Entropy loss is defined as,

$$L_{CE} = -\frac{1}{m}\sum_{i}^{m} log\frac{\exp\left(W_{y_i}^T \cdot x_i + b_{y_i}\right)}{\sum_{j}^{N}\exp\left(W_j^T \cdot x_i + b_j\right)}, \quad (1)$$

where x_i is the input sample with $i \in \{1, ..., m\}$ and m is the number of samples, y_i is the class label, W is the weight matrix, b indicates the bias value, W_{y_i} and W_j are the y_i and j column of W with $j \in \{1, ..., N\}$ and N is the total number of classes.

When this function is used to train the DNN, their parameters are adapted to push the features as far as possible from the decision boundary. Therefore, the inter-class separability is improved, but the features are not necessarily encouraged to enhance the intra-class compactness. This behaviour is problematic because the learned features are prone to be separable for the multi-class classification, rather than to be well calibrated for the task of speaker verification itself.

2.2. Ring Loss

It is a loss function oriented to bring compactness to the features extracted from the network. It applies a convex norm constraint over the primary loss, for example a Cross-Entropy loss, to normalize the features [24]. By using Ring Loss, the system is trained to learn the features norm constrained close to the unit circle. This way, the features will increase their discrimination power among different classes while the intra-class features variability is reduced. The Ring loss can be formulated as,

$$L_R = \frac{\lambda}{2m}\sum_{i}^{m}(||x_i||_2 - R), \quad (2)$$

where R is the target norm value, usually 1, λ is the loss weight, x_i is the input sample of the penultimate layer with $i \in \{1, ..., m\}$ and m is the number of samples.

2.3. Triplet Loss

Moreover, motivated by the need for reducing the intra-class variance while enlarging the inter-class separation, Triplet loss function was created [11]. A triplet neural network structure is employed for training this loss function. Herein, three examples are selected. First, an example from a specific identity which is the reference called anchor (e). Then, a positive example of the same identity of the anchor (e^+), and a negative example of a different identity of the anchor (e^-). Thus, three instances of the same neural network with shared parameters are trained, aiming to make larger the similarity metric anchor-positive example than the similarity metric anchor-negative example, and add a margin. The Triplet loss can be written as,

$$L_{TR} = \sum_{i}^{m^+}\sum_{j}^{m^-}||(s_\Theta(p_i^+)||_2 - ||s_\Theta(p_j^-)||_2 + \tau, \quad (3)$$

where $s_\Theta(p_i^+)$ is the similarity metric of each pair of anchor-positive embeddings where $p_i^+ = (e, e_i^+)$ with $i \in \{1, ..., m^+\}$ and m^+ is the total number of positive examples, $s_\Theta(p_j^-)$ indicates the metric of each pair of anchor-negative embeddings where $p_j^- = (e, e_j^-)$ with $i \in \{1, ..., m^-\}$ and m^- is the total number of negative examples, and τ is the minimum margin between those similarities.

3. aAUC Loss

In [15] we developed a SV system with the triplet philosophy that optimized directly the AUC as loss function. The AUC measures the probability that all the pairs of examples are ranked correctly. This way, it provides a measure of the whole system performance independently of the operating point. SV systems are generally trained to optimize the classification performance without considering the verification process and the relative measures during the training. Therefore, AUC is more intuitive for the detection task than the previously mentioned loss functions. Since AUC function is not differentiable, we proposed an effective approximation to enable the backpropagation of the gradients. Hence, given a set of network parameters Θ, our approximation of the AUC loss function can be written using a sigmoid function as,

$$aAUC(\Theta) = \frac{1}{m^+m^-}\sum_{i}^{m^+}\sum_{j}^{m^-}\sigma(\alpha(s_\Theta(p_i^+) - s_\Theta(p_j^-))), \quad (4)$$

where $s_\Theta(p_i^+)$ is the similarity metric of each pair of anchor-positive embeddings, $s_\Theta(p_j^-)$ indicates the metric of each pair of anchor-negative embeddings, and α is an adjustable parameter which is set using development data.

Beyond the advantage of introducing the FAR-FRR trade-off in the verification learning framework, *aAUC* cannot contribute to the calibration of the system. *aAUC* loss optimizes every operating point to achieve a performance improvement of the whole system, so a specific decision threshold for a practical application operating point is not possible to handle. Furthermore, the triplet approach has some drawbacks on the slow convergence and instability. To address this issue, the triplets should be carefully selected with sample mining strategies such as Hard Negative Mining [11]. This way the performance improves, but this solution involves a high computational cost which excessively slows down the training process.

4. Proposal: aDCF Loss Function

This paper proposes a loss function that considers the performance measure of the SV systems, keeping the philosophy of Cross-Entropy loss and maintaining the same speed of training. Inspired by the DCF [19, 20], the *aDCF* loss function allows to adapt the network parameters to minimize the cost and learning the optimal decision threshold for the specific application. Since the original loss function is not differentiable, we propose an effective approximation for both probabilities.

The *aDCF* loss function measures a weighted sum of the probability of false alarm or FAR (P_{fa}) and the probability of misses or FRR (P_{miss}). For *m* number of examples, the P_{fa} can be determined empirically by the average number of times the scores of non-target speakers N_{non} are greater than the detection threshold (Ω), so a false alarm is produced. While the P_{miss} is determined by the average number of times the scores of target speakers N_{tar} are smaller than the decision threshold Ω, so the system cannot detect and a miss is produced. Both

probabilities are also expressed as a function of the network parameters Θ. Therefore, the P_{fa} and the P_{miss} can be written as,

$$P_{fa}(\Theta,\Omega) = \frac{\sum_{y_i \in y_{non}} \mathbb{1}(s_\Theta(x_i,y_i) > \Omega)}{N_{non}}, \quad (5)$$

$$P_{miss}(\Theta,\Omega) = \frac{\sum_{y_i \in y_{tar}} \mathbb{1}(s_\Theta(x_i,y_i) < \Omega)}{N_{tar}}, \quad (6)$$

where $\mathbb{1}()$ function has a value equal to *'1'* whenever the score $s_\Theta(x_i,y_i)$ meets the condition with respect to Ω, and *'0'* otherwise. The score $s_\Theta(x_i,y_i)$ is obtained from the last linear layer of the neural network as,

$$s_\Theta(x_i,y_i) = W_{y_i}^T \cdot x_i + b_{y_i}, \quad (7)$$

where x_i is the input signal to the last linear layer, and $W_{y_i}^T$ and b_{y_i} are the layer parameters of the speaker class y_i. This expression can be interpreted as a cosine product plus a speaker dependent threshold correction.

However, the expressions (5), (6) for the probabilities are not differentiable, so we replace the $\mathbb{1}()$ function by a sigmoid function of the difference to make an approximation which enables the backpropagation of the gradients:

$$\hat{P}_{fa}(\Theta,\Omega) = \frac{\sum_{y_i \in y_{non}} \sigma(s_\Theta(x_i,y_i) - \Omega)}{N_{non}}, \quad (8)$$

$$\hat{P}_{miss}(\Theta,\Omega) = \frac{\sum_{y_i \in y_{tar}} \sigma(\Omega - s_\Theta(x_i,y_i))}{N_{tar}}, \quad (9)$$

where $\sigma()$ is the sigmoid function. Thus, with this expressions, we can now propose to minimize the following approximated loss function defined by,

$$aDCF(\Theta,\Omega) = \gamma \cdot \hat{P}_{fa}(\Theta,\Omega) + \beta \cdot \hat{P}_{miss}(\Theta,\Omega), \quad (10)$$

where γ and β are variable parameters to provide more cost relevance to one of the terms over the other. In this work, we aim to do a proof of concept, so γ will be defined as equal to β to maintain the balance between both terms. Note that Ω will be optimized as part of the system parameters. Furthermore, since the output of the last layer (7) is calculated using subsets of training samples, we assume that the values corresponding to N_{tar} and N_{non} are 1, and $N-1$ for each input sample, where N is the total number of speakers. Thus, the training process has a similar efficiency and convergence speed to Cross-Entropy.

5. Supervector Neural Network System

In the following section, we present the structure of the system used for experiments Fig.1. First, we describe the front-end based on neural networks combined with the differentiable alignment mechanism proposed in our previous work [25, 15]. Then, the back-end strategies are described. Finally, a score normalization is applied to the scores.

5.1. Front-end with Alignment Mechanism

The most recent SV systems employ a front-end based on a DNN with an average mechanism to extract embeddings. In the context of the text-dependent verification tasks [6, 26], the averaging dismisses the order of phonetic information in the utterance. In [25, 15], we addressed this problem by replacing the average pooling mechanism by a frame-to-state alignment method as a new layer into the DNN architecture. With this mechanism, the temporal structure of the uttered phrase

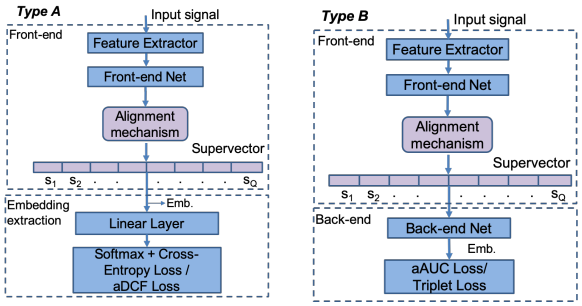

Figure 1: *Two architectures used to create the SV system.* Type A *is trained with two different loss functions, and in the case of Cross-Entropy loss, it is also used as pre-train for the other architecture.* Type B *is trained to optimize the back-end net for Triplet and aAUC loss.*

and the speaker information is kept and encoded in a supervector. In this paper, the Bayesian Dark Knowledge [27] approach is used to provide robustness to the neural network front-end. The alignment method employed is a Gaussian Mixture Model (GMM) combined with a Maximum A Posteriori (MAP) adaptation [28].

5.2. Back-end

For the verification process, two different architectures are used. Fig.1(a) shows the first architecture which uses the proposed front-end with the Cross-Entropy loss and the *aDCF* loss to train the system. Once the system is trained, a cosine similarity is applied over the embeddings to achieve the verification scores.

On the other side, Fig 1(b) depicts the second architecture which employs a trainable back-end using the Triplet loss function or the *aAUC* loss function. Due to the slow convergence, to initialize this architecture, we use a pre-trained model employing the architecture *type A* trained with the Cross-Entropy loss. In this case, one embedding is obtained for each example, and then the back-end is applied to obtain verification scores.

5.3. Score Normalization

A gender-dependent score normalization is applied to conclude the system. We used a symmetric normalization, denoted S-norm [14], with the whole cohort of files available as:

$$score\_norm = \frac{score - \mu_1}{\sigma_1} + \frac{score - \mu_2}{\sigma_2}, \quad (11)$$

where $score$ is the original score, μ_1 and σ_1 are the mean and standard deviation of the scores obtained from the evaluation of test vs. dev files, and μ_2 and σ_2 are the mean and standard deviation of the scores obtained from the evaluation enroll vs. dev files.

6. Experimental Setup

RSR2015-PartI text-dependent speaker verification corpus [29] was used for experiments. This dataset consists of speech samples from 157 male and 143 female speakers. For each speaker, there are 9 sessions pronouncing 30 different phrases. The corpus is divided into three speaker subset: background (bkg), development (dev), and evaluation (eval). Unlike our previous work [25, 15], in this paper, we only employ the bkg data (97 speakers, 47 female/50 male) for training, and reserve the dev

data to scores normalization. The evaluation part is used for enrollment and trial evaluation.

6.1. Experimental Description

We have trained a 64 component GMM per phrase without phrase transcription using the bkg partition. From these models, we extract the alignment information to use in the frame-to-state alignment mechanism of our DNN architecture. As input to the DNN, a set of features composed of 20 dimensional Mel-Frequency Cepstral Coefficients (MFCC) with their first and second derivates are employed. To manage a possible overfitting problem in our models due to the lack of data, we apply a data augmentation method called Random Erasing [30] on the input features.

As reference for experiments, we compare an average pooling (avg) after the front-end [6, 26] to the proposed front-end with alignment mechanism that creates the supervector (svec). On the other side, we contrast the architecture *type A* using Cross-Entropy (CE) loss, with/without the architecture *type B*. Finally, we compare with the proposed *aDCF* loss. These comparisons have been performed with and without the score normalization (snorm).

7. Results and Discussion

Table 1 presents EER and NIST 2008 (*DCF08* [31]) and 2010 minimum detection costs (*DCF10* [32]) for the mentioned architectures and loss functions. No matter what the loss function or normalization technique have been employed, the average pooling mechanism does not generate proper embeddings to represent the phrase and speaker information in this text-dependent SV task.

Table 1: *Experimental results on RSR2015-PartI [29] eval subset, showing EER% and NIST 2008 and 2010 min costs (*DCF08*,* DCF10*). These results were obtained to compare the different loss functions with and without normalization (snorm).*

Architecture				Results (Female+Male)		
Pooling	**Norm.**	**Loss**	**Type**	**EER%**	**DCF08**	**DCF10**
avg	–	*CE*	–	11.70	0.572	0.988
		aDCF	–	11.48	0.621	0.999
svec	–	*CE*	*A*	0.80	0.041	0.171
		Trloss	*B*	0.87	0.046	0.209
		aAUC	*B*	**0.68**	0.036	0.166
		aDCF	*A*	0.70	**0.033**	**0.130**
avg	*snorm*	*CE*	–	9.55	0.515	0.999
		aDCF	–	10.50	0.602	0.995
svec	*snorm*	*CE*	*A*	0.58	0.031	0.154
		Trloss	*B*	1.05	0.053	0.236
		aAUC	*B*	0.62	0.031	0.160
		aDCF	*A*	**0.56**	**0.029**	**0.123**

We can observe that the proposed loss function achieves the best minimum detection costs with both metrics *DCF08* and *DCF10*. These metrics focus the attention on different operating points in function of the parameters for controlling the relevance of associated costs to the P_{fa} and the P_{miss}. Especially relevant is the improvement in the *DCF10*, which comes from the fact that this metric provides more importance to low P_{fa} and the proposal achieves the best FAR (Fig.2). Comparing with the Cross-Entropy loss, which is the most comparable training strategy, we achieve a relative improvement of 24%, and 20.1% with normalization (S-norm). Notice that *aDCF* achieves the most stable results at the introduction of normalization, which

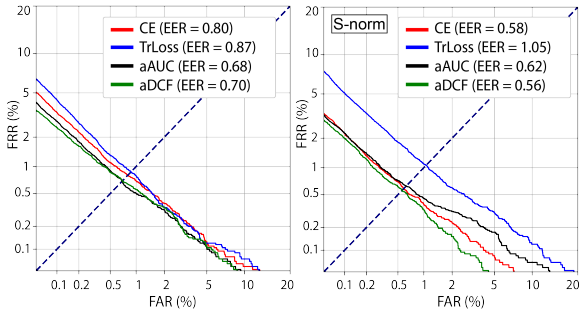

Figure 2: *DET curves of the systems with and without normalization using different loss functions.*

indicates that this function is better calibrated. When we apply S-norm, the Triplet loss function does not follow the tendency of the other loss functions, it gets worst. A possible cause of this issue could be that this function has been trained to maximize the distance among scores from different identities. Then, the scores without normalization are too sparsed, and when we normalize, they are forced to get back closed loosing discriminability.

In addition, we represent the Detection Error Trade-off (DET) curves in Fig.2. These curves depicted the relationship between FAR and FRR. The average pooling experiments are not represented since they do not provide relevant information. This representation demonstrates that if we do not apply a normalization technique, the performance of the *aAUC* and the *aDCF* systems are similar. Indeed, the EER is lower in the first one because it has been trained with the *aAUC* loss to achieve the best performance of the system in all operating point. Though, it does not take into account the operating points relation to the specific requirements of the system application. However, when applying a snorm technique, the tendency followed by the *aDCF* DET curve shows that is the best SV system for each operating point, even at the EER.

8. Conclusions

In this paper, we have presented a novel loss function based on the detection errors FA and FR as an alternative to replace the classical Cross-Entropy loss. This *aDCF* loss function allows the end-to-end system to handle the optimization of the decision threshold employed to compute the ratio between FAR and FRR. Even though this is a preliminary study, the proposal has been able to obtain competitive results. This encourages us to conclude that the *aDCF* loss function is a promising approximation of SV system performance considering the operating point. Undoubtedly, this is an interesting line of research. In the future, we plan to work on the definition of the *aDCF*, focusing on the weights of the P_{fa}/P_{miss}, and the decision threshold. Furthermore, our ultimate goal is to progress this proposal until we reach to an end-to-end system able to self-calibrate.

9. Acknowledgements

This work has been supported by the Spanish Ministry of Economy and Competitiveness and the European Social Fund through the project TIN2017-85854-C4-1-R, by the Government of Aragon (Reference Group T36_17R) and co-financed with Feder 2014-2020 "Building Europe from Aragon", and by Nuance Communications, Inc. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

10. References

[1] A. K. Jain, A. Ross, S. Prabhakar *et al.*, "An introduction to biometric recognition," *IEEE Transactions on circuits and systems for video technology*, vol. 14, no. 1, 2004.

[2] J. Gonzalez-Rodriguez, "Evaluating automatic speaker recognition systems: An overview of the nist speaker recognition evaluations (1996-2014)," *Loquens*, 2014.

[3] S. Z. Li, *Encyclopedia of Biometrics*, 1st ed. Springer Publishing Company, Incorporated, 2009.

[4] 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.

[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 *2016 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2016, pp. 165–170.

[6] E. Malykh, S. Novoselov, and O. Kudashev, "On residual CNN in text-dependent speaker verification task," in *International Conference on Speech and Computer*. Springer, 2017, pp. 593–601.

[7] 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.

[8] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of interspeaker variability in speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 5, pp. 980–988, 2008.

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

[10] H. V. Nguyen and L. Bai, "Cosine similarity metric learning for face verification," in *Asian conference on computer vision*. Springer, 2010, pp. 709–720.

[11] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[12] C. Zhang, K. Koishida, and J. H. Hansen, "Text-independent speaker verification based on triplet convolutional neural network embeddings," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 26, no. 9, pp. 1633–1644, 2018.

[13] Y. Li, F. Gao, Z. Ou, and J. Sun, "Angular Softmax Loss for End-to-end Speaker Verification," *arXiv preprint arXiv:1806.03464*, 2018.

[14] N. Brümmer and A. Strasheim, "Agnitio's speaker recognition system for evalita 2009," in *The 11th Conference of the Italian Association for Artificial Intelligence*. Citeseer, 2009.

[15] V. Mingote, A. Miguel, A. Ortega, and E. Lleida, "Optimization of the Area Under the ROC Curve using Neural Network Supervectors for Text-Dependent Speaker Verification," *arXiv preprint arXiv:1901.11332*, 2019. [Online]. Available: http://arxiv.org/abs/1901.11332

[16] L. P. Garcia-Perera, J. A. Nolazco-Flores, B. Raj, and R. Stern, "Optimization of the DET curve in speaker verification," in *Spoken Language Technology Workshop (SLT), 2012 IEEE*. IEEE, 2012, pp. 318–323.

[17] K.-A. Toh, J. Kim, and S. Lee, "Maximizing area under ROC curve for biometric scores fusion," *Pattern Recognition*, vol. 41, no. 11, pp. 3373–3392, 2008.

[18] A. Herschtal and B. Raskutti, "Optimising area under the ROC curve using gradient descent," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 49.

[19] A. Martin and M. Przybocki, "The NIST 1999 speaker recognition evaluationAn overview," *Digital signal processing*, vol. 10, no. 1-3, pp. 1–18, 2000.

[20] S. Bengio and J. Mariéthoz, "The expected performance curve: a new assessment measure for person authentication," IDIAP, Tech. Rep., 2003.

[21] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[22] A. Krizhevsky, I. Sulskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information and Processing Systems (NIPS)*, pp. 1–9, 2012.

[23] Y. Srivastava, V. Murali, and S. R. Dubey, "A Performance Comparison of Loss Functions for Deep Face Recognition," *arXiv preprint arXiv:1901.05903*, 2019.

[24] Y. Zheng, D. K. Pal, and M. Savvides, "Ring loss: Convex feature normalization for face recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5089–5097.

[25] V. Mingote, A. Miguel, A. Ortega, and E. Lleida, "Differentiable Supervector Extraction for Encoding Speaker and Phrase Information in Text Dependent Speaker Verification," in *Proc. IberSPEECH 2018*, 2018, pp. 1–5. [Online]. Available: http://dx.doi.org/10.21437/IberSPEECH.2018-1

[26] Y. Liu, Y. Qian, N. Chen, T. Fu, Y. Zhang, and K. Yu, "Deep feature for text-dependent speaker verification," *Speech Communication*, vol. 73, pp. 1–13, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.specom.2015.07.003

[27] A. K. Balan, V. Rathod, K. P. Murphy, and M. Welling, "Bayesian dark knowledge," in *Advances in Neural Information Processing Systems*, 2015, pp. 3438–3446.

[28] D. A. Reynolds, R. C. Rose *et al.*, "Robust text-independent speaker identification using Gaussian mixture speaker models," *IEEE transactions on speech and audio processing*, vol. 3, no. 1, pp. 72–83, 1995.

[29] A. Larcher, K. A. Lee, B. Ma, and H. Li, "Text-dependent speaker verification: Classifiers, databases and RSR2015," *Speech Communication*, vol. 60, pp. 56–77, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.specom.2014.03.001

[30] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," *arXiv preprint arXiv:1708.04896*, 2017.

[31] "The NIST Year 2008 Speaker Recognition Evaluation Plan," 2008. [Online]. Available: https://www.nist.gov/sites/default/files/documents/2017/09/26/sre08_evalplan_release4.pdf

[32] "The NIST Year 2010 Speaker Recognition Evaluation Plan," 2010. [Online]. Available: https://www.nist.gov/sites/default/files/documents/itl/iad/mig/NIST_SRE10_evalplan-r6.pdf