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

# Adversarial Regularization for End-to-end Robust Speaker Verification

*Qing Wang*[1,2], *Pengcheng Guo*[1], *Sining Sun*[1], *Lei Xie*[1,*], *John H.L. Hansen*[2]

[1]School of Computer Science, Northwestern Polytechnical University, Xi'an, China
[2]CRSS: Center for Robust Speech Systems, UT-Dallas, TX, USA

{qingwang, pcguo, snsun, lxie}@nwpu-aslp.org, john.hansen@utdallas.edu

## Abstract

Deep learning has been successfully used in speaker verification (SV), especially in end-to-end SV systems which have attracted more interest recently. It has been shown in image as well as speech applications that deep neural networks are vulnerable to adversarial examples. In this study, we explore two methods to generate adversarial examples for advanced SV: (i) fast gradient-sign method (FGSM), and (ii) local distributional smoothness (LDS) method. To explore this issue, we use adversarial examples to attack an end-to-end SV system. Experiments will show that the neural network can be easily disturbed by adversarial examples. Next, we propose to train an end-to-end robust SV model using the two proposed adversarial examples for model regularization. Experimental results with the TIMIT dataset indicate that the EER is improved relatively by (i) +18.89% and (ii) +5.54% for the original test set using the regularized model. In addition, the regularized model improves EER of the adversarial example test set by a relative (i) +30.11% and (ii) +22.12%, which therefore suggests more consistent performance against adversarial example attacks.

**Index Terms**: end-to-end robust SV, adversarial example, adversarial regularization, fast gradient-sign method (FGSM), local distributional smoothness (LDS)

## 1. Introduction

Speaker verification is the task of determining whether an input speech sample belongs to an assumed identity or not, and is a popular topic in biometric authentication. It has drawn more attention of the safety and robustness in SV, and evidence shows SV can be susceptible to many kinds of spoofing attacks [1, 2, 3], such as impersonation, replay, speech synthesis, and voice conversion. These spoofing attacks present high risk to a SV system, so anti-spoofing [4] has become a crucial focus recently. Apart from these spoofing attacks, there can be other kinds of attacks. It has been shown by many studies that deep neural networks are vulnerable to minor (even imperceptible) perturbations added to their inputs. Such minor perturbations which can disturb the neural network are called adversarial perturbations [5].

Adversarial examples were first proposed by Szegedy et al. [6] for a computer vision task. The input sample added with adversarial perturbation, which results in incorrect output from the network, is called an adversarial example. Szegedy et al. discovered that a correctly classified example could be mis-classified by a neural network when an adversarial perturbation, imperceptible to human beings, is added to the original example. In [5, 7], the authors focused on improving robustness while resisting the adversarial perturbations. The authors used adversarial examples on discrete sequences to attack a whole-binary malware detector in [8].

Adversarial examples have also been applied previously in speech processing. In [9], the authors constructed targeted audio adversarial examples on automatic speech recognition and were able to turn any audio waveform into any target transcription. Adversarial examples were also used for fooling a SV system in [10], through adding a peculiar noise to the original speaker examples which is almost indistinguishable by humans. Additionally, after they presented white-box and black-box attacks to the end-to-end SV system, the accuracy of the system decreased significantly. Adversarial examples cannot only be used for attacking, but also can be used for improving robustness of speech recognition systems. For example, in [11, 12], adversarial examples were used for data augmentation to improve the robustness of the system in adverse environments in speech recognition and keyword spotting tasks, respectively.

End-to-end SV has been an attractive topic recently. Many deep learning methods have been successfully applied in speaker identification tasks [13, 14, 15, 16, 17, 18, 19, 20]. Therefore, many end-to-end SV system will face the risk of adversarial examples. As shown in [10], SV system can be easily attacked by adversarial examples. To solve this problem, in this study, we propose to use adversarial regularization based on adversarial examples to improve the robustness of end-to-end SV. We adopt the structure of the generalized end-to-end SV system proposed in [19] as our text-independent SV baseline. We first use the fast gradient-sign method (FGSM) [5] to generate a set of adversarial examples as the test set to attack the baseline SV system. The performance degrades significantly, which indicates the SV network is vulnerable to adversarial examples. Next, we propose training the end-to-end SV models using adversarial regularization. The essence of adversarial regularization is to seek the worst point around the current data point, and then using this worst point to optimize the system. Therefore, the adversarial regularization can improve the robustness of the model to adversarial perturbations and in turn make the output distribution smoother. We use two methods to generate the adversarial examples: FGSM and local distributional smoothness (LDS) [21], for model regularization. The goal is to disturb well-trained models in order to make them more robust to small variations in input. Experiments will demonstrate that the regularized model can make the the end-to-end SV system more robust and smoother, and can also weaken the impact of adversarial perturbations.

There are many challenges in increasing the robustness of SV systems due to adversarial attacks. While other corpora such as the NIST SRE [22, 23] are attractive for speaker recognition research advancement, the diversity of mismatch including language, hand-set, microphone, and combinations of these make addressing the problem of adversarial attacks more difficult. As such, in this study, we focus on a probe investigation that re-

Qing Wang is currently a visiting student with CRSS-UTDallas.
*Lei Xie is the corresponding author.



http://dx.doi.org/10.21437/Interspeech.2019-2983

moves all forms of mismatch, so that the only research question will be the adversarial attack challenge. To this end, we employ the TIMIT corpus [24] to establish a solution to first address adversarial attacks without other mismatch issues. Having established this, it would be possible to generalize this to more open SV data sets. Having established this foundational solution, we suggest that the following experiments can be performed with non-mismatch SRE data, as well as other SRE sets.

## 2. End-to-end speaker verification

We use the generalized end-to-end loss (GE2E) for speaker verification proposed in [19] as our baseline architecture (in Fig. 1).

In the GE2E method, $N \times M$ utterances are retrieved to build a batch. These utterances are drawn from $N$ different speakers, where each speaker has $M$ utterances. The feature vector $x_{ji}$ represents the input feature of $j$-th speaker's $i$-th utterance. All features in the batch are then submitted into an LSTM network [25] densely connected layer with linear activations. The mapping function is $f(x_{ji};\theta)$, where $\theta$ is the parameter set of the network. The embedding vector of the $j$-th speaker's $i$-th utterance is expressed as follows:

$$e_{ji} = \frac{f(x_{ji};\theta)}{\|f(x_{ji};\theta)\|_2}. \tag{1}$$

The centroid of the embedding vectors from the $k$-th speaker is then defined as $c_k$.

$$c_k = \frac{1}{M}\sum_{m=1}^{M} e_{km}. \tag{2}$$

The similarity matrix $S_{ji,k}$ is the scaled cosine similarities between each embedding vector $e_{ji}$ and all centroids $c_k$:

$$S_{ji,k} = w \cdot cos(e_{ji}, c_k) + b, \tag{3}$$

where $w$ and $b$ are scaled learnable parameters.

In [19], Wang et al. remove $e_{ji}$ when calculating the centriod of the true speaker to ensure the training is stable and to avoid trivial solutions. Equation 2 is still used to compute the centriod when $k \neq j$. When $k = j$, Equation 4 is used instead for the centriod computation.

$$c_j^{(-i)} = \frac{1}{M-1}\sum_{m=1, m\neq i}^{M} e_{jm}. \tag{4}$$

As a result, the $S_{ji,k}$ should be computed as:

$$S_{ji,k} = \begin{cases} w \cdot cos(e_{ji}, c_j^{(-i)}) + b & k = j \\ w \cdot cos(e_{ji}, c_k) + b & k \neq j. \end{cases} \tag{5}$$

During training, the embedding of each utterance is assumed to be close to its own centroid, and far from other speakers' centroids. Therefore, the GE2E loss $L_G$ can be defined as:

$$L_G(x;\theta) = -\sum_{ji} S_{ji,j} + \sum_{ji} \log \sum_{k=1}^{N} \exp S_{ji,k}. \tag{6}$$

## 3. Adversarial regularization

In this section, we introduce the notion of adversarial example and two methods of generating adversarial examples: the fast gradient-sign method (FGSM) [5] and local distributional smoothness (LDS) [21]. Moreover, we will detail how we use adversarial examples for model regularization in end-to-end SV.

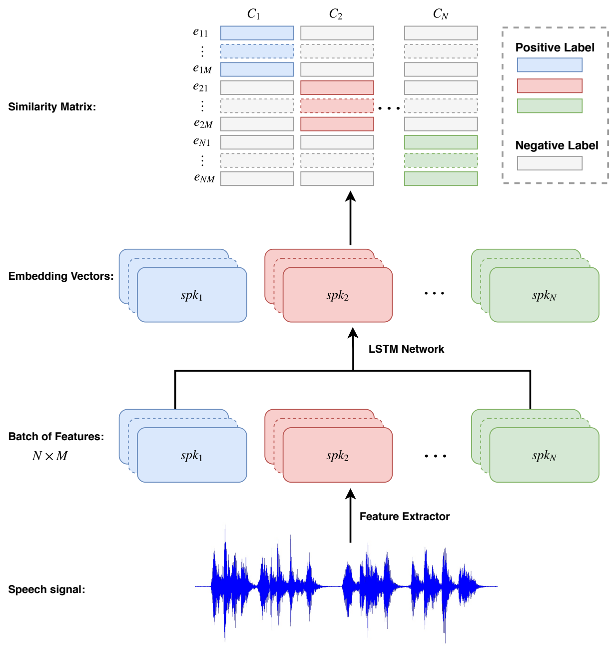


Figure 1: *Baseline end-to-end speaker verification architecture as introduced in [19]*

### 3.1. Adversarial example

When a very small perturbation, even if it is imperceptible to human listeners, is added to the original sample, the new sample is mis-classified by the neural network. This type of perturbation is the adversarial perturbation, and this new sample is the adversarial example. So, the definition of adversarial examples is as follows: $f(x;\theta)$ represents the objective function of a neural network, where $x$ is the input with its corresponding label $y$, and $\theta$ is the parameter set of the network. An adversarial example $\hat{x}$ can be constructed as:

$$\hat{x} = x + \delta\,,\ \| \delta \| \ll \| x \|, \tag{7}$$

$$y \neq f(\hat{x};\theta), \tag{8}$$

here, $\delta$ is called the adversarial perturbation, which is far less than the input $x$ and is imperceptible to humans.

### 3.2. Generation of adversarial example

#### 3.2.1. FGSM

The FGSM was proposed in [5] to generate adversarial examples $\hat{x}$, which is considered as a supervised generation because it needs the true label. We assume that the model works with input samples $x \in X$, where $X$ is the input space, and certain labels (outputs) $y$ from the label space $Y$. So, we assume we have a set of training samples:

$$D = \{x, y \mid x \in X, y \in Y\}. \tag{9}$$

We are given the input sample $(x, y)$ and model parameter $\theta$, the model is trained to minimize the loss function $L(x, y;\theta)$. The stochastic gradient descent (SGD) method [26] is used for optimization during training. According to the definition of adversarial examples, we want to generate a new input sample $\hat{x}(\approx x)$ which can increase the value of the loss function

$L(x, y; \theta)$. In other words, it points the direction where gradient climbs up as fast as possible. In FGSM, the adversarial perturbation $\delta_{F-adv}$ is generated as:

$$\delta_{F-adv} = \epsilon \ \text{sign}(\bigtriangledown_x L(x, y; \theta)), \tag{10}$$

here, $\epsilon$ is a small constant. Note that FGSM only use the sign of the gradient instead of its value, which aims to obtain an optimal max-norm constrained perturbation as described in Equation 8.

#### 3.2.2. *LDS - a virtual adversarial training*

The second method is virtual adversarial training (VAT) based on local distributional smoothness (LDS) [21], which does not require the ground truth label for adversarial example generation. LDS is defined as the negative of the sensitivity of model distribution $p(x, \theta)$ with respect to the perturbation of $x$. The KL-divergence is used to measure the sensitivity of model distributions before and after perturbation. Detail definitions are as follows:

$$\triangle_{KL}(\delta, x, \theta) = KL[p(x, \theta) \parallel p(x + \delta, \theta)], \tag{11}$$

$$\delta_{L-adv} = \arg\max_{\delta} \{\triangle_{KL}(\delta, x, \theta); \|\delta\|_2 \leq \epsilon\}, \tag{12}$$

$$\text{LDS}(x, \theta) = - \triangle_{KL} (\delta_{L-adv}, x, \theta), \tag{13}$$

here, $\epsilon$ is a small positive constant and $\delta_{L-adv}$ is called the virtual adversarial perturbation for input sample $x$. In order to improve the smoothness of model, we should find an accurate perturbation $\delta_{V-adv}$ which wrecks the model distribution in a direst way. The value of $\delta_{V-adv}$ can be effectively estimated through an iterative algorithm as described in [7]. Specifically, we first initial a $\delta_i$ randomly and weight it with a parameter to compute the KL divergence as denoted in Equation 11, where $i$ refers to the step of the iteration and $i \leq Iter$. Then, we regard the derivative of KL-divergence in regards to $\delta_i$ as the new perturbation in the $i + 1$ th steps. It should be noted that we also normalize the derivative using $L_1$-Norm. After several times, we assign $\delta_{Iter}$ to $\delta_{L-adv}$. Usually $Iter = 1$ is able to get a good result.

### 3.3. Adversarial regularization for end-to-end speaker verification

If the generated adversarial examples can easily fool the model, it means the model is not robust enough to resist the adversarial perturbations and the output distribution of the model is unsmooth in regards to the inputs. Therefore, we train the model using adversarial regularization, which uses adversarial examples for model regularization to improve the robustness. Adversarial regularization seeks to find a worst spot around the current data point, and then optimize using this worst data point just found, which can make the overall model robust to adversarial perturbation as well as the output distribution smoother. In this study, we use both kinds of adversarial examples for the model regularization.

After we generate an adversarial example $\hat{x}$ with FGSM, the neural network can be trained using a regularized objective function as follows:

$$\hat{x} = x + \delta_{F-adv}, \tag{14}$$

$$L_{AT}(x, y; \theta) = L(x, y; \theta) + \alpha L(\hat{x}, y; \theta), \tag{15}$$

where $\alpha > 0$. The original loss function is amended from the loss on adversarial examples.

In the LDS method, the goal is to improve the smoothness of the model in the neighborhood of all observed inputs. Therefore, the regularized objective function becomes:

$$\hat{x} = x + \delta_{L-adv}, \tag{16}$$

$$L_{LDS}(x; \theta) = L(x; \theta) - \alpha \text{LDS}(x, \theta), \tag{17}$$

here, $\alpha > 0$. In our solution, LDS here is used as a regularization term to promote the smoothness of the model distribution. Algorithm 1 outlines the GE2E model, integrated with adversarial regularization. The parameter $P_a$ refers to the probability of performing adversarial training, which is similar as schedule sampling.

**Algorithm 1** Training GE2E SV model using FGSM or LDS

Initialize model parameters $\boldsymbol{\theta}$, let epoch = 0
Given hyper parameters
- $\epsilon$ in Equation 10 and 12
- $\alpha$ in Equation 15 and 17
- $Iter$, $\xi$ and $P_a$
- $N \geq 1$, starting epoch number for FGSM and LDS

Training set $D = \{x, y \mid x \in X, y \in Y\}$
**while** not converge **do**
  Get a mini-batch training data $M = \{x, y\}$
  Forward the network using data $M$
  **if** epoch $>$ N && use FGSM && random(0,1) $< P_a$ **then**
    Generate $\hat{M}$ using Equation 14 from $M$
    Train $\theta$ using Equation 15 with $M \cup \hat{M}$
  **else if** epoch $>$ N && use LDS && random(0,1) $< P_a$ **then**
    Generate $\hat{M}$ using Equation 16 from $M$
    Train $\theta$ using Equation 17 with $M \cup \hat{M}$
  **else**
    Train $\theta$ using GE2E loss with $M$
  **end if**
  epoch = epoch + 1
**end while**
**return** $\theta$

# 4. Experiments

### 4.1. Dataset

We use the TIMIT corpus [24] as the evaluation data set. We recognize that other data sets such as NIST SRE are possible, but we used TIMIT to provide a proof-of-concept, since it is phonetically balanced, with full transcriptions and balanced geographical speaker distribution. This will provide a good base assessment specifically for our adversarial regularization study. The dataset contains studio quality recordings of 630 speakers (192 female, 438 male), sampled at 16 kHz, covering the eight major regional dialects of American English. Each speaker reads ten phonetically balanced sentences. We randomly select 540 speakers for the training set, 30 speakers for the validation set, and 60 speakers for the test set. In this study, we mainly focused on an investigation without taking all kinds of mismatch into consideration, so that the only research question will be the adversarial attack challenge. Therefore, we employ the TIMIT corpus to establish a solution to first address adversarial attacks without other mismatch issues.

### 4.2. Experimental setup

#### 4.2.1. Baseline

We adopt the GE2E SV system in [19] as baseline system. The feature extraction and network configuration are the same as the text-independent SV application in [19]. A 3-layer LSTM with 768 hidden nodes, connected with a projection layer with 256 hidden nodes is trained for SV. When we train the GE2E model, each batch contains $N = 4$ speakers with $M = 5$ utterances.

#### 4.2.2. Adversarial regularization

We use FGSM to generate the adversarial examples based on the baseline system for the original test set. The configuration is: $p_a = 0.5$, $\epsilon = 0.15$ and $\alpha = 0.3$. The network configuration of the regularized model is the same as our baseline system. The hyper parameter in FGSM is: $p_a = 0.5$, $\epsilon = 0.15$ and $\alpha = 0.3$. In our LDS based adversarial regularization experiments, we set the hyper parameter to be: $p_a = 1.0$, $\epsilon = 0.15$, $\alpha = 1.0$, $\xi = 10$ and $Iters = 1.0$.

### 4.3. Experimental results and analysis

#### 4.3.1. Using adversarial example to attack the SV system

In Table 1, System 1 is the GE2E speaker verification baseline system. The EER of the baseline model is 4.87%. We generate adversarial examples using FGSM for the test set based on our baseline model. This is used as the adversarial examples testing set, denoted as FGSM-AE test set. For System 2, we use FGSM-AE as the test set to test with the baseline model. The EER of System 2 is 11.89%, which is a 144.15% relative loss compared with System 1. This shows that the SV model can be easily affected by the adversarial examples.

In this experiment, we can achieve low EER when using a deep learning method for end-to-end SV tasks. However, the performance degrades significantly when we use adversarial examples to test the model. This experiment indicates that end-to-end SV is vulnerable when attacked by adversarial examples; the model is not robust and therefore un-smooth.

Table 1: *End-to-end SV results against adversarial example.*

| System | Model | Testing | EER(%) | Rel.(%) |
|---|---|---|---|---|
| 1 | *Baseline* | *Original* | 4.87 | – |
| 2 | *Baseline* | *FGSM-AE* | 11.89 | -144.15 |

#### 4.3.2. Adversarial regularization for SV system

Table 2 shows performance of the original test set for the adversarial regularization models. In System 3, the adversarial examples used for model regularization are generated based on FGSM, which we call the the FGSM-REG model. The EER of the original test set on the FGSM-REG model is 3.95%, with a +18.89% relative improvement compared to System 1. The adversarial examples for model regularization in System 4 are generated based on the LDS method. This model is called the LDS-REG model. Here, the EER of the original test set on the LDS-REG model is 4.60%, with a +5.54% relative error reduction achieved from this solution.

The experiments indicate that adversarial regularization can make the SV model distribution smoother and improve overall robustness of the model.

Table 2: *Adversarial regularization based end-to-end SV results on TIMIT dataset.*

| System | Model | Testing | EER(%) | Rel.(%) |
|---|---|---|---|---|
| 1 | *Baseline* | Original | 4.87 | – |
| 3 | *FGSM-REG* | Original | 3.95 | +18.89 |
| 4 | *LDS-REG* | Original | 4.60 | +5.54 |

#### 4.3.3. Adversarial regularization against adversarial example attack

Table 3 shows results of the FGSM-AE test set on adversarial regularization models. In System 5, we test the FGSM-REG model using the FGSM-AE test set data, where the EER is 8.31%, with a +30.11% relative error rate reduction compared with System 2. The LDS-REG model is tested with adversarial examples in System 6, and its EER improves from 11.89% to 9.26%, achieving a +22.12% improvement over System 2.

We can observe that when we use adversarial examples to attack the SV system, both adversarial regularization models can improve their robustness and can achieve lower EERs than the original baseline model. The above results show the effectiveness of adversarial regularization for the GE2E SV model. Adversarial regularization can not only improve robustness and smoothness of the model, but can also achieve consistent performance against adversarial example attack.

Table 3: *Adversarial regularization based end-to-end SV results against adversarial example.*

| System | Model | Testing | EER(%) | Rel.(%) |
|---|---|---|---|---|
| 2 | *Baseline* | *FGSM-AE* | 11.89 | – |
| 5 | *FGSM-REG* | *FGSM-AE* | 8.31 | +30.11 |
| 6 | *LDS-REG* | *FGSM-AE* | 9.26 | +22.12 |

## 5. Conclusion

In this study, we have proposed to improve the robustness of end-to-end speaker verification systems via adversarial regularization. In particular, we added adversarial examples generated by two methods: FGSM or LDS, as a regularization term into the objective function. With the proposed methods, we obtained +18.89% and +5.54% relative EER improvement on the original test set using FGSM and LDS respectively. Moreover, the EER of the adversarial example test set was shown to improve relatively +30.11% and +22.12% compared with the original model. These results suggest the effectiveness of adversarial regularization in advancing SV systems to be more robust, and prevent against system attacks from adversarial examples.

In our future work, we will conduct experiments on more challenging data sets, such as SRE. In addition, we will use the generated adversarial examples for data augmentation when we train the model to avoid the adversarial example attack.

## 6. Acknowledgement

This research work is supported by the National Natural Science Foundation of China (No.61571363) and the China Scholarship Council (Grant No. 201806290113).

# 7. References


[1] Z. Wu, T. Kinnunen, N. Evans, J. Yamagishi, C. Hanilçi, M. Sahidullah, and A. Sizov, "Asvspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[2] Z. Wu, N. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, and H. Li, "Spoofing and countermeasures for speaker verification: A survey," *Speech Communication*, vol. 66, pp. 130–153, 2015.

[3] Z. Wu, P. L. De Leon, C. Demiroglu, A. Khodabakhsh, S. King, Z.-H. Ling, D. Saito, B. Stewart, T. Toda, M. Wester *et al.*, "Anti-spoofing for text-independent speaker verification: An initial database, comparison of countermeasures, and human performance," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 24, no. 4, pp. 768–783, 2016.

[4] Z. Wu, E. S. Chng, and H. Li, "Detecting converted speech and natural speech for anti-spoofing attack in speaker recognition," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[5] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[6] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[7] T. Miyato, S.-i. Maeda, S. Ishii, and M. Koyama, "Virtual adversarial training: a regularization method for supervised and semi-supervised learning," *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[8] F. Kreuk, A. Barak, S. Aviv-Reuven, M. Baruch, B. Pinkas, and J. Keshet, "Adversarial examples on discrete sequences for beating whole-binary malware detection," *arXiv preprint arXiv:1802.04528*, 2018.

[9] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2018, pp. 1–7.

[10] F. Kreuk, Y. Adi, M. Cisse, and J. Keshet, "Fooling end-to-end speaker verification with adversarial examples," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 1962–1966.

[11] S. Sun, C.-F. Yeh, M. Ostendorf, M.-Y. Hwang, and L. Xie, "Training augmentation with adversarial examples for robust speech recognition," *arXiv preprint arXiv:1806.02782*, 2018.

[12] X. Wang, S. Sun, C. Shan, J. Hou, L. Xie, S. Li, and X. Lei, "Adversarial examples for improving end-to-end attention-based small-footprint keyword spotting," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019.

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

[14] 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.

[15] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5115–5119.

[16] C. Zhang and K. Koishida, "End-to-end text-independent speaker verification with triplet loss on short utterances." in *Interspeech*, 2017, pp. 1487–1491.

[17] 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.

[18] 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.

[19] 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.

[20] Q. Wang, W. Rao, S. Sun, L. Xie, E. S. Chng, and H. Li, "Unsupervised domain adaptation via domain adversarial training for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4889–4893.

[21] T. Miyato, S.-i. Maeda, M. Koyama, K. Nakae, and S. Ishii, "Distributional smoothing with virtual adversarial training," *arXiv preprint arXiv:1507.00677*, 2015.

[22] S. O. Sadjadi, T. Kheyrkhah, A. Tong, C. S. Greenberg, D. A. Reynolds, E. Singer, L. P. Mason, and J. Hernandez-Cordero, "The 2016 nist speaker recognition evaluation." in *Interspeech*, 2017, pp. 1353–1357.

[23] S. O. Sadjadi, C. S. Greenberg, D. A. Reynolds, E. Singer, L. P. Mason, and J. Hernandez-Cordero, "The 2018 nist speaker recognition evaluation." in *Interspeech (submitted)*, 2019.

[24] W. M. Fisher, "The darpa speech recognition research database: specifications and status," in *Proc. DARPA Workshop on Speech Recognition, Feb. 1986*, 1986, pp. 93–99.

[25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[26] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*. Springer, 2010, pp. 177–186.