Speaker Identification for Household Scenarios with Self-attention and Adversarial Training

Ruirui Li[†] *Jyun-Yu Jiang*[§] *Xian Wu*[¶] *Chu-Cheng Hsieh*[†] *Andreas Stolcke*[†]

[†] Amazon [§]University of California Los Angeles [¶]University of Notre Dame

[†]{ruirul, stolcke}@amazon.com, [§]jyunyu@cs.ucla.edu, [¶]xwu9@nd.edu, [†]chucheng@ucla.edu

Abstract

Speaker identification based on voice input is a fundamental capability in speech processing enabling versatile downstream applications, such as personalization and authentication. With the advent of deep learning, most state-of-the-art methods apply machine learning techniques and derive acoustic embeddings from utterances with convolutional neural networks (CNNs) and recurrent neural networks (RNNs). This paper addresses two inherent limitations of current approaches. First, voice characteristics over long time spans might not be fully captured by CNNs and RNNs, as they are designed to focus on local feature extraction and adjacent dependencies modeling, respectively. Second, complex deep learning models can be fragile with regard to subtle but intentional changes in model inputs, also known as adversarial perturbations. To distill informative global acoustic embedding representations from utterances and be robust to adversarial perturbations, we propose a Self-Attentive Adversarial Speaker-Identification method (*SAASI*). In experiments on the VCTK dataset, *SAASI* significantly outperforms four state-of-the-art baselines in identifying both known and new speakers.

Keywords— Self-attention, adversarial training, speaker identification in households

1. Introduction

Smart speakers like Amazon Echo and Google Home allow convenient voice-enabled access to a wide variety of services and experiences, and have gained widespread use. As these devices are typically used by multiple speakers in a household, speaker identification is key to enable many important functionalities such as authentication and user-based customization. In this paper, we develop two novel techniques for speaker identification geared toward household-like scenarios with a small number of competing speaker identities.

Deep learning-based speaker identification methods have shown superior performance compared to older approaches based on i-vectors [1] and GMM-UBMs [2]. For example, Deep Speaker [3] and VGGVox [4] adopt CNN-based residual networks to learn voice acoustic representations based on utterance spectrograms, while SincNet [5] applies CNNs to perform speaker identification from raw waveforms. Generalized end-to-end speaker (GE2E) identification [6] utilizes RNNs to model utterances and develops a similarity-based loss function so that the similarity between utterance representations from the same/different speaker is maximized/minimized, respectively. GE2E with shared-parameter non-linear attention (SNL) [7] further extends GE2E to obtain more informative acoustic features by weighting contributions of RNN outputs differently. However, these neural methods still potentially face problems capturing dependencies and characteristics expressed over long time spans within an utterance. CNNs by design are biased toward modeling features over nearby frames and frequencies, and RNNs are hard to train for retention of relevant information over long time intervals.

Adversarial training, which minimizes the maximal risk for label-preserving input perturbations, was shown to be effective to enhance both security and generalization of deep learning models [8–13]. Although previous studies [14, 15] apply domain adversarial training, they only focus on adapting a well trained speaker model to a new domain or language, instead of boosting the model robustness. Li et al. [13] investigate the vulnerability of Gaussian Mixture Model i-vector based speaker verification systems to adversarial attacks. Meng et al. [16] strive to enhance the robustness of speaker identification through multi-task learning. Suthokumar et al. [17] utilize adversarial multi-task learning with a focus on distinguishing genuine and replayed speech. In this paper, we will use adversarial training as a tool to enhance the generalization of trained models, rather than as a defense against attacks.

In this work, we leverage the self-attention mechanism [18, 19] to enhance long-span modeling of speaker characteristics. More precisely, the self-attention mechanism allows us to fully utilize dependencies over all frames in an utterance, resulting in informative global acoustic embedding representations. To enhance generalization and robustness, we incorporate small but deliberate perturbations to the input spectrograms of training utterances. The model is then trained in an adversarial manner, which not only learns from the original training data but also improves based on the dynamically constructed out-of-distribution samples. As a result, adversarial training should improve the robustness of speaker models to attacks; however, here we evaluate the resulting effect on identification performance in terms of generalization to unseen test cases.

In a nutshell, our proposed solution combines global acoustic feature extraction and adversarial perturbation in training for more effective speaker identification. Section 2 gives the algorithmic details of our approach (representation learning in Section 2.1 and adversarial training in Section 2.3). In Section 3 we describe experiments, showing that *SAASI* substantially outperforms all baseline methods, even when the utterances are as short as 1.5 seconds. Section 4 summarizes our findings.

2. Problem Statement & Methodology

We formulate the speaker identification task as follows. Given a closed set of users, with a few short registered voice utterances for each user as enrollment, and another short test utterance from a test user, the goal is to recognize the speaker identity behind the test utterance. In this work, we focus on text-independent speaker identification and presume that each utterance is very short [6, 20–23], for example, one to two seconds.

2.1. Self-Attentive Utterance Representation Learning

In this section, we discuss how we extract the acoustic features from an utterance and represent it in a fixed-length vector. Each utterance u is first represented by a sequence of frame-level feature vectors, each giving the frequency distribution over a short time window. We use the spectrogram $\boldsymbol{SP}_u$ of an utterance u as the input, and wish to learn a representation of speaker-relevant acoustic features of u, in two steps. First, we aim at mining correlations across frames in an utterance. Second, we aggregate the frame embeddings, including their correlational information, into a fixed-length embedding vector that expresses the speaker-relevant information in the utterance.

In the first step, to uncover correlations across frames, each utterance u is represented by an array of frame embeddings resulting from attention to itself and the other frames in u. In other words, we attempt to "explain" each frame in u by the frame itself and any other related frames. The self-attention transform is defined as:

$$\text{Self-Att}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}(\frac{\boldsymbol{Q} \cdot \boldsymbol{K}^T}{\sqrt{d_Q}})\boldsymbol{V}, \tag{1}$$

where $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ represent the query, key, and value matrices in the self-attention mechanism, respectively. The scale factor $\sqrt{d_Q}$ is used to avoid overly large values of the inner product, where d_Q is the feature dimension of $\boldsymbol{Q}$.

In our case, the self-attention operation takes the utterance spectrograms $\boldsymbol{SP}_u \in \mathbb{R}^{c \times d}$, where c is number of frames and d gives the dimension of a frame, as the inputs and feeds them into the self-attention layer to learn the transformed frame representations. To incorporate the frame location information, we follow [18] and add sinusoidal positional embedding $\boldsymbol{E}_p$ into $\boldsymbol{SP}_u$ before fusion. Formally,

$$E_p(t, pos) = \begin{cases} \sin \frac{pos}{10000^{t/d}}, \text{if } t \text{ is even}, \\ \cos \frac{pos}{10000^{t/d}}, \text{if } t \text{ is odd}, \end{cases} \tag{2}$$

where pos is the position of a frame, d is the dimension of a frame, and $E_p(t, pos)$ gives the t-th element in the positional embedding of a frame, which is at position pos.

$$\boldsymbol{SP}_u^{'} = \boldsymbol{SP}_u + \boldsymbol{E}_p. \tag{3}$$

$$\tilde{\boldsymbol{E}}_u = \text{Self-Att}(\boldsymbol{SP}_u^{'} \cdot \boldsymbol{W}^Q, \boldsymbol{SP}_u^{'} \cdot \boldsymbol{W}^K, \boldsymbol{SP}_u^{'} \cdot \boldsymbol{W}^V), \tag{4}$$

where $\boldsymbol{W}^Q$, $\boldsymbol{W}^K$, and $\boldsymbol{W}^V$ are query, key, and value projection matrices, respectively.

Thus, the self-attention result $\tilde{\boldsymbol{E}}_u$ learns the transformed embeddings of frames by comparing the pairwise closeness between frames. Each transformed frame embedding is a weighted sum of frame embedding of itself and other related frames, where each weight gauges the similarity between one frame and another one in u. In this way, $\tilde{\boldsymbol{E}}_u$ encodes the correlated frame information, with each one frame explained by itself and others. In particular, $\tilde{\boldsymbol{E}}_u$ is good at modeling distant frame relationships, as all frames are treated equally regardless of temporal distance in the self-attention mechanism.

To increase the non-linearity of the self-attention mechanism, we further feed the transformed frame embeddings $\tilde{\boldsymbol{E}}_u$ into a feed-forward neural network:

$$\tilde{\boldsymbol{E}}_u^f = \boldsymbol{W}_2^f \cdot \text{ReLU}(\boldsymbol{W}_1^f \cdot \tilde{\boldsymbol{E}}_u + \boldsymbol{b}_1^f) + \boldsymbol{b}_2^f, \tag{5}$$

where $\boldsymbol{W}_1^f$, $\boldsymbol{W}_2^f$, and $\boldsymbol{b}_2^f$, $\boldsymbol{b}_2^f$ are the weight matrices and biases in the feed-forward layer. To comprehensively correlate the frame-level information across an utterance, we perform the self-attention operation twice via residual shortcut connections [24].

To derive a summarized global acoustic representation of an utterance, we average $\tilde{\boldsymbol{E}}_u^f$ over the time dimension into one embedding vector, denoted as $\bar{\boldsymbol{E}}_u^f$. In addition, the summarized embedding vector is further L2-normalized. Formally, an utterance u is represented by a fixed-length vector $\boldsymbol{E}_u$:

$$\boldsymbol{E}_u = \frac{\bar{\boldsymbol{E}}_u^f}{||\bar{\boldsymbol{E}}_u^f||_2}. \tag{6}$$

2.2. End-To-End Training

We follow [6, 7] and train the speaker identification model in an end-to-end manner. We construct a batch by $N \times M$ utterances, where N is the number of speakers and M is the number of utterances from each speaker. We use u_{ji} to represent the i-th utterance from speaker j. Moreover, we use $\boldsymbol{E}_{ji}$ to represent the embedding vector of the j-th speaker's i-th utterance. The acoustic biometry of speaker j is further represented by the embedding centroid $\boldsymbol{C}_j$ of his/her M utterances. Formally,

$$\boldsymbol{C}_j = \frac{1}{M} \sum_{m=1}^{M} \boldsymbol{E}_{jm} \tag{7}$$

The similarity matrix $\boldsymbol{S}_{ji,k}$ is defined as the scaled cosine similarities between each embedding vector $\boldsymbol{E}_{ji}$ to all centroids $\boldsymbol{C}_k$:

$$\boldsymbol{S}_{ji,k} = \boldsymbol{W}^s \cdot cos(\boldsymbol{E}_{ji}, \boldsymbol{C}_k) + \boldsymbol{b}^s, \tag{8}$$

where $\boldsymbol{W}^s$ and $\boldsymbol{b}^s$ are learnable parameters.

During training, the embedding of each utterance is expected to be similar to the centroid of all of that speaker's embeddings, while at the same time, far from other speakers' centroids. The loss on each embedding vector $\boldsymbol{E}_{ji}$ is defined as:

$$L(\boldsymbol{E}_{ji}|\Theta) = -\boldsymbol{S}_{ji,j}^{\Theta} + \log \sum_{k=1}^{N} \exp(\boldsymbol{S}_{ji,k}^{\Theta}), \tag{9}$$

where Θ represents the model parameters. The loss function allows us to push each embedding vector close to its centroid while also pulling it away from all other centroids. The final end-to-end loss is the sum of all losses over all utterances involved in the similarity matrix.

$$L(\boldsymbol{S}|\Theta) = \sum_{j,i} L(\boldsymbol{E}_{ji}|\Theta) \tag{10}$$

2.3. Adversarial Training

Adversarial attacks refer to techniques that fool models through malicious perturbations of inputs. To enhance robustness (primarily for the benefit of generalization, but also to defend against adversarial attacks), we force the model during training to perform well consistently even when adversarial perturbations are presented. To achieve this goal, we additionally optimize the model to minimize the objective function with the perturbed utterances. Formally, we define the objective function with adversarial examples incorporated as:

$$\begin{aligned} L_{adv}(\boldsymbol{S}|\Theta) &= L(\boldsymbol{S}|\Theta) + \lambda L(\boldsymbol{S}_{\Delta_{adv}}|\Theta), \\ \text{where } \Delta_{adv} &= \arg \max_{\Delta, \|\Delta\| \leq \epsilon} L(\boldsymbol{S}_{\Delta}|\hat{\Theta}), \end{aligned} \tag{11}$$

where Δ denotes the perturbations on input utterances, $\boldsymbol{S}_{\Delta_{adv}}$ is the similarity matrix after applying Δ_{adv} perturbations to the

samples, $\epsilon \geq 0$ ensures that the perturbations are perceptible but not too large, and $\hat{\Theta}$ denotes the current model parameters. In this formulation, the adversarial term $L(\boldsymbol{S}_{\Delta_{adv}}|\Theta)$ can be treated as a model regularizer, which stabilizes the identification performance. λ is introduced to control the strength of the adversarial regularizer, where the intermediate variable Δ maximizes the objective function to be minimized by Θ. The training process can be summarized as playing a minimax game:

$$\Theta_{opt}, \Delta_{opt} = \arg\min_{\Theta} \max_{\Delta, \|\Delta\| \leq \epsilon} L(\boldsymbol{S}|\Theta) + \lambda L(\boldsymbol{S}_{\Delta}|\Theta), \quad (12)$$

where the optimizer for the model parameters Θ acts as the minimizing player while the procedure to derive dynamic perturbations Δ acts as the maximizing player. The maximizing player strives to construct the worst-case perturbations against the current model. The two players alternately play the minmax game until convergence.

Constructing adversarial perturbations. Given a training utterance u_{ji}, the adversarial perturbations Δ_{adv} to be constructed are expected to best fool the current model. Therefore, the problem of constructing Δ_{adv} is formulated as maximizing

$$\ell_{adv}(\boldsymbol{E}_{ji}|\hat{\Theta}) = -\boldsymbol{S}^{\hat{\Theta}}_{\Delta\, ji,j} + \log \sum_{k=1}^{N} \exp(\boldsymbol{S}^{\hat{\Theta}}_{\Delta\, ji,k}), \quad (13)$$

where $\hat{\Theta}$ denotes a set of current model parameters. As it is difficult to derive the exact optimal solutions of Δ_{adv}, we apply the fast gradient method proposed in [9] to estimate Δ_{adv}, where we approximate the objective function around Δ as a linear function. To maximize the approximated linear function, we move along the gradient direction of the objective function with respect the Δ. With the max-norm constraint $\|\Delta\| \leq \epsilon$, we approximate Δ_{adv} as:

$$\Delta_{adv} = \epsilon \frac{\tau}{\|\tau\|}, \text{ where } \tau = \frac{\partial \ell_{adv}(\boldsymbol{E}_{ji}|\hat{\Theta})}{\partial \boldsymbol{S}\boldsymbol{P}_{ji}}. \quad (14)$$

Learning model parameters. We now discuss how to learn model parameters Θ. The local adversarial objective function to minimize for a training instance is as follows:

$$\begin{aligned} \ell_{adv}(\boldsymbol{E}_{ji}|\Theta) &= -\boldsymbol{S}^{\Theta}_{ji,j} + \log \sum_{k=1}^{N} \exp(\boldsymbol{S}^{\Theta}_{ji,k}) \\ &-\lambda\{\boldsymbol{S}^{\Theta}_{\Delta_{adv}\, ji,j} - \log \sum_{k=1}^{N} \exp(\boldsymbol{S}^{\Theta}_{\Delta_{adv}\, ji,k})\}, \end{aligned} \quad (15)$$

where Δ_{adv} is obtained from Equation 14.

The final adversarial end-to-end loss is the sum of all adversarial losses over all utterances:

$$L_{adv}(\boldsymbol{S}|\Theta) = \sum_{j,i} \ell_{adv}(\boldsymbol{E}_{ji}|\Theta) \quad (16)$$

We obtain the stochastic gradient update rule for Θ as

$$\Theta = \Theta - \eta \frac{\partial L_{adv}(\boldsymbol{S}|\Theta)}{\partial \Theta}, \quad (17)$$

where η denotes the learning rate.

Algorithm 1 summarizes the training process. In each training step, we first randomly generate a mini-batch of utterances from N speakers, with each speaker M utterances. We then follow Equation 10 to calculate the loss based on utterances from this mini-batch and optimize the model. After that, we construct a corresponding mini-batch of modified utterances with adversarial perturbations, feed them into the model, and update model parameters so that the resulting model learns to resist such adversarial perturbations. The training involves multiple training steps and stops after completing a certain number of training iterations. The parameters achieving the best equal error rate (EER) on a validation dataset are utilized for evaluations.

Algorithm 1: Adversarial parameter optimizations

Input: Training utterances U, max iteration $iter_{\max}$;
Output: Model parameters Θ

Initialization: initialize Θ with Normal distribution $N(0,0.01)$, $iter = 0$, $\Theta_{opt} = \Theta$, $EER_{opt} = EER_{vali}$;
repeat
  foreach *batch of training utterances* **do**
    // Updating model parameters;
    $\Theta \leftarrow$ Equation 10;
    // Constructing adversarial perturbations;
    $\Delta_{adv} \leftarrow$ Equation 14;
    // Updating model parameters with adversarial training;
    $\Theta \leftarrow$ Equation 17;
  if $EER_{vali} < EER_{opt}$ **then**
    $EER_{opt} = EER_{vali}$;
    $\Theta_{opt} = \Theta$;
  $iter$ + +;
until $iter > iter_{max}$;
Return Θ_{opt};

3. Experiments

We conduct experiments on the VCTK dataset to evaluate the performance of *SAASI* against four state-of-the-art algorithms.

3.1. Dataset and Experimental Settings

The experiments are conducted on the publicly available VCTK dataset[1]. Table 1 shows the statistics of the dataset. The model is both trained and evaluated on the VCTK dataset. From the full dataset, 80% of speakers are treated as known users and the remaining 20% of speakers are treated as new users. Utterances from the known users are used for training and unseen utterances from both known and new users are used for evaluation. We follow the previous work [23] to extract acoustic features from the raw audio. The 40-dimensional spectrograms are extracted from each utterance after an energy-based voice activity detection. Table 2 shows the main parameters and their default values to tune in the experiments.

Table 1: *The statistics of the experimental dataset*

	Gender		Age		
	Female	Male	[10, 20)	[20, 30)	[30, 40)
# of speakers	61	47	14	91	3

	Major Accent					
	English	American	Scottish	Irish	Canadian	South African
# of speakers	33	22	25	9	8	4

[1] VCTK: http://homepages.inf.ed.ac.uk/jyamagis

Table 2: *Main parameters of SAASI in the experiments*

Parameters	Value	Parameters	Value
Learning rate η	0.01	Max number of iterations	5000
Regularizer weight λ	1	Perturbation bound ϵ	0.1
# of speakers N in a batch	4	Utterances per speaker M	5

3.2. Baseline Methods

To evaluate the performance of *SAASI*, the following four methods are adopted as baselines.

- **GE2E** [6] uses an LSTM-RNN to construct utterance embeddings and optimizes the end-to-end speaker identification system by maximizing the similarity among utterances coming from the same speaker.
- **SNL** [7] extends GE2E by adding a shared-parameter non-linear attention layer on top of LSTM to extract more informative acoustic features in utterances to conduct speaker identification.
- **GE2E**$_{adv}$ extends GE2E by conducting training in an adversarial manner similarly as described in Section 2.3.
- **SNL**$_{adv}$ conducts adversarial training on SNL.

3.3. Evaluation Performance

We evaluate the performance of *SAASI* against the different baseline methods on the VCTK dataset. As most smart speakers serve customers in household scenarios, we simulate a plausible speaker verification task for such a scenario, and define a corresponding version of equal error rate (EER), called household-level EER (H-EER). To form a household, we randomly shuffle the test speakers and then sample 1000 households composed of 4 speakers each, with replacement. Speaker profiles are derived from 5 enrollment utterances each, and a disjoint set of 20 utterances (5 per speaker) is selected for within-household testing. Utterance are about 1.5 seconds long. For each speaker we thus obtain 5 target trials and 15 non-target trials. Based on the speaker verification scores for these, we compute the EER for each household, and finally the H-ERR is computed as the macro-average over all households.

Table 3: *H-EER performance on known users*

Utt Length	Embed Size	GE2E	GE2E$_{adv}$	SNL	SNL$_{adv}$	*SAASI*
1.5s	64	6.95%	5.76%	4.22%	4.13%	3.67%
1.5s	128	6.49%	5.66%	4.03%	3.85%	3.39%

Table 4: *H-EER performance on new users*

Utt Length	Embed Size	GE2E	GE2E$_{adv}$	SNL	SNL$_{adv}$	*SAASI*
1.5s	64	13.84%	13.58%	10.86%	9.31%	6.56%
1.5s	128	13.11%	12.73%	10.30%	9.11%	6.39%

We want to evaluate performance separately for both known users (those used in model training) and new users (those not seen in training), i.e., we sample households as described above, but drawing all *speakers* either from the "known" set or the "new" set). However, *test utterances* were always drawn from a pool unseen during model training. Tables 3 and 4 show the performance of different methods on known users and new users, respectively.

We can make three observations from these results. First, GE2E$_{adv}$ and SNL$_{adv}$ outperform GE2E and SNL, respectively, in all conditions. This demonstrates the effectiveness of training with adversarial examples in speaker identification, and shows the usefulness of this training objective beyond defense against attacks.

Second, SNL and SNL$_{adv}$ achieve lower H-EER than GE2E and GE2E$_{adv}$, respectively, in all conditions. The improvement stems from the shared-parameter non-linear attention mechanism in SNL and SNL$_{adv}$, as it summarizes the RNN outputs differently with consideration of their contributions to identification performance. In this way, more informative global acoustic features are extracted from utterances.

Third, we observe that *SAASI* consistently achieves the best H-EER compared with the four baselines, in all conditions. The gain over the next best model, SNL$_{adv}$, is particularly pronounced on new users, and less for known users. We can attribute the superior performance of SAASI to the self-attention mechanism, in conjunction with adversarial training. The conventional attention mechanism on RNNs would tend to give a higher weight to frames that are closer to a given frame position and therefore make it hard to find correlations between frames that are far apart in the utterance. The self-attention mechanism applied in this work can easily and comprehensively correlate the acoustic information between all frames in an utterance, yielding a more powerful modeling mechanism for utterance-level embeddings. The adversarial training helps generalize the model and makes it more robust against sample noise, and helps presumably especially for new speakers.

4. Conclusion

In this work, we investigate deep-learning based methods for a household-based speaker identification task that serves as a proxy for speaker recognition as used on smart speaker devices. We present *SAASI*, a framework that utilizes self-attention to learn global acoustic features from spoken utterances. Moreover, the model is trained in an adversarial end-to-end manner so that the identification system is trained to be robust to perturbations in the input representations. We define a household-level equal error rate to measure the speaker identification for the household scenario. We evaluate our method against the recently proposed GE2E and SNL methods, as well as adversarially trained version of these. Experiments on the short (about 1.5-second long) utterances from the public VCTK dataset show that *SAASI* outperforms the four baseline methods, both for seen and, especially, for unseen speakers. In the future work, we would like to train *SAASI* on large datasets to involve more speakers and evaluate the performance of *SAASI* under different SNR and far-field scenarios.

5. Acknowledgement

We would like to thank Minhua Wu for her informative feedback and Charles Chang for the high-level support of this research.

6. References

[1] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-End Factor Analysis for Speaker Verification," *IEEE Trans. Audio, Speech & Language Processing*, pp. 788–798, 2011.

[2] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker Verification Using Adapted Gaussian Mixture Models," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[3] 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," *CoRR*, vol. abs/1705.02304, 2017.

[4] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep Speaker Recognition," *CoRR*, vol. abs/1806.05622, 2018. [Online]. Available: http://arxiv.org/abs/1806.05622

[5] M. Ravanelli and Y. Bengio, "Speaker Recognition from Raw Waveform with SincNet," in *2018 IEEE Spoken Language Technology Workshop, SLT, Athens, Greece, December 18-21*, 2018, pp. 1021–1028.

[6] L. Wan, Q. Wang, A. Papir, and I. Moreno, "Generalized End-to-end Loss For Speaker Verification," in *Proceedings of ICASSP, Calgary, AB, Canada, April 15-20*, 2018, pp. 4879–4883.

[7] F. A. R. R. Chowdhury, Q. Wang, I. Lopez-Moreno, and L. Wan, "Attention-Based Models for Text-Dependent Speaker Verification," in *Proceedings of ICASSP, Calgary, AB, Canada, April 15-20, 2018*, pp. 5359–5363.

[8] C. Zhu, Y. Cheng, Z. Gan, S. Sun, T. Goldstein, and J. Liu, "Freelb: Enhanced adversarial training for language understanding," in *Proceedings of ICLR, 26-30 April 2020, Addis Ababa, Ethiopia*, pp. 770–778.

[9] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," in *ICLR*, 2015.

[10] N. Carlini and D. A. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *2018 IEEE Security and Privacy Workshops, SP Workshops 2018, San Francisco, CA, USA, May 24, 2018*, 2018, pp. 1–7.

[11] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," in *Proceedings of ICML, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, J. G. Dy and A. Krause, Eds., pp. 284–293.

[12] H. Wu, S. Liu, H. Meng, and H. Lee, "Defense against adversarial attacks on spoofing countermeasures of ASV," in *Proceedings of ICASSP, Barcelona, Spain, May 4-8, 2020*.

[13] X. Li, J. Zhong, X. Wu, J. Yu, X. Liu, and H. Meng, "Adversarial attacks on GMM i-vector based speaker verification systems," in *Proceedings of ICASSP, Barcelona, Spain, May 4-8, 2020*.

[14] 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 *Proceedings of ICASSP, Calgary, AB, Canada, April 15-20, 2018*, pp. 4889–4893.

[15] G. Bhattacharya, M. J. Alam, and P. Kenny, "Adapting end-to-end neural speaker verification to new languages and recording conditions with adversarial training," in *Proceedings of ICASSP, Brighton, United Kingdom, May 12-17, 2019*, pp. 6041–6045.

[16] Z. Meng, Y. Zhao, J. Li, and Y. Gong, "Adversarial speaker verification," in *Proceedings of ICASSP, Brighton, United Kingdom, May 12-17, 2019*, pp. 6216–6220.

[17] G. Suthokumar, V. Sethu, K. Sriskandaraja, and E. Ambikairajah, "Adversarial multi-task learning for speaker normalization in replay detection," in *Proceedings of ICASSP, Barcelona, Spain, May 4-8, 2020*.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Proceedings of NIPS, 4-9 December 2017, Long Beach, CA, USA*, pp. 5998–6008.

[19] S. Karita, X. Wang, and et al, "A comparative study on transformer vs RNN in speech applications," in *IEEE Automatic Speech Recognition and Understanding Workshop, ASRU, Singapore, December 14-18, 2019*, pp. 449–456.

[20] J. Wang, K. Wang, M. Law, F. Rudzicz, and M. Brudno, "Centroid-based Deep Metric Learning For Speaker Recognition," in *Proceedings of ICASSP, Brighton, United Kingdom, May 12-17*, 2019, pp. 3652–3656.

[21] R. Li, J.-Y. Jiang, X. Wu, H. Mao, C.-C. Hsieh, and W. Wang, "Bridging Mixture Density Networks with Meta-learning for Automatic Speaker Identification," in *Proceedings of ICASSP, Barcelona, Spain, May 04-08, 2020*, pp. 5359–5363.

[22] R. Li, J. Jiang, J. Liu, C. Hsieh, and W. Wang, "Automatic Speaker Recognition with Limited Data," in *Proceedings of WSDM, Houston, Texas, USA, February 3-7*, 2020.

[23] P. Anand, A. K. Singh, S. Srivastava, and B. Lall, "Few Shot Speaker Recognition using Deep Neural Networks," *CoRR*, vol. abs/1904.08775, 2019.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of CVPR, Las Vegas, NV, USA, June 27-30, 2016*, pp. 770–778.