Automatic Speaker Recognition with Limited Data

Ruirui Li
UCLA
rrli@cs.ucla.edu

Jyun-Yu Jiang
UCLA
jyunyu@cs.ucla.edu

Jiahao Liu
Tongji University
jiahaoliu1891@gmail.com

Chu-Cheng Hsieh
Amazon
chucheng@ucla.edu

Wei Wang
UCLA
weiwang@cs.ucla.edu

ABSTRACT

Automatic speaker recognition (ASR) is a stepping-stone technology towards semantic multimedia understanding and benefits versatile downstream applications. In recent years, neural network-based ASR methods have demonstrated remarkable power to achieve excellent recognition performance with sufficient training data. However, it is impractical to collect sufficient training data for every user, especially for fresh users. Therefore, a large portion of users usually has a very limited number of training instances. As a consequence, the lack of training data prevents ASR systems from accurately learning users acoustic biometrics, jeopardizes the downstream applications, and eventually impairs user experience.

In this work, we propose an adversarial few-shot learning-based speaker identification framework (*AFEASI*) to develop robust speaker identification models with only a limited number of training instances. We first employ metric learning-based few-shot learning to learn speaker acoustic representations, where the limited instances are comprehensively utilized to improve the identification performance. In addition, adversarial learning is applied to further enhance the generalization and robustness for speaker identification with adversarial examples. Experiments conducted on a publicly available large-scale dataset demonstrate that *AFEASI* significantly outperforms eleven baseline methods. An in-depth analysis further indicates both effectiveness and robustness of the proposed method.

CCS CONCEPTS

• **Information systems → Speech / audio search**.

KEYWORDS

Speaker identification; few-shot learning; adversarial training

ACM Reference Format:
Ruirui Li, Jyun-Yu Jiang, Jiahao Liu, Chu-Cheng Hsieh, and Wei Wang. 2020. Automatic Speaker Recognition with Limited Data. In *The Thirteenth ACM International Conference on Web Search and Data Mining (WSDM ’20), February 3–7, 2020, Houston, TX, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3336191.3371802

WSDM ’20, February 3–7, 2020, Houston, TX, USA
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-6822-3/20/02…$15.00
https://doi.org/10.1145/3336191.3371802

1 INTRODUCTION

Within the last couple of years, voice has become one of the most ever-growing media through which people interact with their devices. For instance, over 47 million people in the United States own a smart home device while 23% of the Britons have a voice-controllable digital assistant at home in 2018 [7, 14]. To ignite the interactions between smart devices and their owners, automatic speaker recognition (ASR) plays an important role to determine the speaker identity based on a short piece of audio. Moreover, the capability of ASR comes with a wide range of applications, such as biometric authentication [23], forensics [10], and personalized services in electronics [13]. In particular, the text-independent ASR with only acoustic information is the most general and non-trial task, which can be used in everyday situations. In text-independent ASR, an arbitrary utterance from one of the known speakers in training set will be given and the system needs to identify which speaker the utterance belongs to.

Deep learning-based ASR methods are gaining popularity due to strong model capacities and superior performance [4, 9, 17, 35]. Most incremental improvements in existing deep learning methods rely on the use of deeper and more complex models with massive training data. More specifically, there are two inherent limitations for existing approaches. First, increasing model complexity is not always desirable in practice because of the greater costs of computation and storage. It thus becomes expensive to get such methods deployed in smart devices to provide offline services. Second, acquiring sufficient labeled training data for all speakers is impractical [21] while the lack of training supports can lead to worse generalization and high vulnerability to tiny perturbations for existing deep learning-based ASR methods [3, 38]. Hence, developing effective techniques for ASR with limited training data remains a daunting task.

To achieve remarkable performance with limited training data, meta-learning is one of the most promising approaches to comprehensively utilize the limited training instances. More specifically, meta-learning systematically observes how machine learning approaches perform on a wide range of similar learning tasks, and then learns to learn new tasks more efficiently [31]. In particular, few-shot learning is a contemporary meta-learning approach that introduces an auxiliary meta-learning phase to generalize and share transferable knowledge across tasks. To learn from extremely limited data, one type of few-shot learning, based on metric learning, looks to light-parametric models, which learn a distance metric among training instances rather than myriad model parameters [33].

More precisely, the essential knowledge can be learned and memorized by reasoning the distance metric between instances in a support module and a query module. Instances in the support module are labeled instances, thereby serving as references. Based on the reference instances, the query instances are then able to conduct reasoning. Finally, metric-learning-based few-shot learning models can be optimized by iterative comparisons between support and query instances such that instances from the same speaker are embedded as close to each other as possible in the hidden space and as far as possible from instances of the other speakers.

To comprehensively exploit the training instances, an alternative way is to generate augmented data based on the training set. Different from conventional methods that separately augment data apart from the training process, we construct augmented data automatically by leveraging adversarial training. In particular, we construct dynamic perturbations at the embedding level to form adversarial examples. These adversarial examples are formed by applying small but intentional perturbations to inputs from the dataset. Specifically, these adversarial examples can be treated as ultimate data augmentation as specific perturbations are created to best fool the model. Accordingly, the model trained in an adversarial manner can not only learn from the original static training data but also improve based on the dynamically constructed perturbed data. As a result, adversarial training significantly improves the robustness of ASR models and achieves out-of-instance generalization while the robustness is crucial for the security-sensitive ASR task. In a nutshell, data augmentation through adversarial training provides another effective solution to thoroughly utilize the training instances and train models resistant of nuisance perturbations to achieve high generalizations in both training and test.

In this paper, we study the problem of speaker identification with a shortage of training data. In essence, we address the data deficiency issue by applying few-shot learning and adversarial training. To be more specific, the main contributions of this work are as follows:

- Different from conventional neural network-based methods, which rely on the availability of a sufficient amount of training data to achieve high identification performance, we model it as a few-shot learning problem to conquer the data deficiency.
- To further improve the generalization of the model, we employ adversarial training. Adversarial examples serve as dynamic augmented data, the optimization of which results in a more generalized and robust speaker recognition system.
- We present a comprehensive empirical evaluation of our approach on a real-world dataset. The experimental results show that our approach, *AFEASI*, significantly outperforms 11 conventional baseline methods in speaker recognition.

2 PROBLEM STATEMENT

In this section, we formally define the objective of this work and summarize the notations in this paper.

Given a short piece of audio x and its mel frequency cepstral coefficients (MFCCs) $\boldsymbol{m}_x$ as features, the goal of this paper is to recognize the speaker identity y among a set of known speakers. In particular, in this work we focus on text-independent automatic speaker identification by leaning from limited pieces of training audios. To better explain the proposed method, Table 1 lists the main notations in this paper.

Table 1: Summary of symbols and their descriptions.

Symbol	Description
x	a piece of audio
y	the speaker identity behind the audio x
$\boldsymbol{m}_x$	the mel frequency cepstral coefficients of audio x
$\boldsymbol{E}_x$	the embedding of audio x
$\boldsymbol{E}_R$	the representative embedding of a set of audios
$\boldsymbol{W}$ & $\boldsymbol{b}$	network weight and bias
K	number of speakers in the support module
N	number of instances per speaker in the support module
α	importance weight in the attention mechanism
q	a query instance
R_k	the aggregated representation of speaker k
$d(q, R_k)$	the euclidean distance between query q and speaker k
S	a set of representatives
L	loss function
η	learning rate
ϵ	perturbation bound
λ	regularizer weight
Θ	model parameters
Δ	parameter perturbations
N_g	random Gaussian noises
τ	weight to control the noises injected
x_{au}	synthetic audio by injecting noises

3 METHODOLOGY

In this section, we discuss how to identify speakers by learning from limited training data. To achieve this goal, we strive to thoroughly utilize the limited instances during training by leveraging few-shot learning and adversarial training.

3.1 Framework Overview

In this paper, a metric-learning-based few-shot learning pipeline is applied to perform N-shot learning for previously rare speakers. More precisely, the model is capable of recognizing a previously rare speaker after having examined only N examples, where N is a small number.

Figure 1 shows the framework of *AFEASI* that performs speaker identification by conducting N-shot, K-way classification tasks with a support set of K different speakers and N training audio instances for each speaker in the support set. In addition, a set of query audio instances is given for prediction. Note that although Figure 1 shows only one query instance for illustration simplicity, *AFEASI* can cope with multiple query audio instances. For each audio instance x, *AFEASI* first extracts the mel frequency cepstral coefficients (MFCCs) [37][1] as acoustic features $\boldsymbol{m}_x$, thereby deriving a fixed-length vector as the audio embedding $\boldsymbol{E}_x$ with an embedding layer. Based on the embeddings of audio instances, an aggregated embedding is constructed as the representative for each speaker in the support module. *AFEASI* then optimizes the distances between the embeddings of the query instances and the representatives of the corresponding speakers so that the representatives can be applied to recognize the speaker identity. The process of the optimization can be summarized as finding a distance metric into a

[1] Note that the details of MFCC construction are discussed in section 4.1.

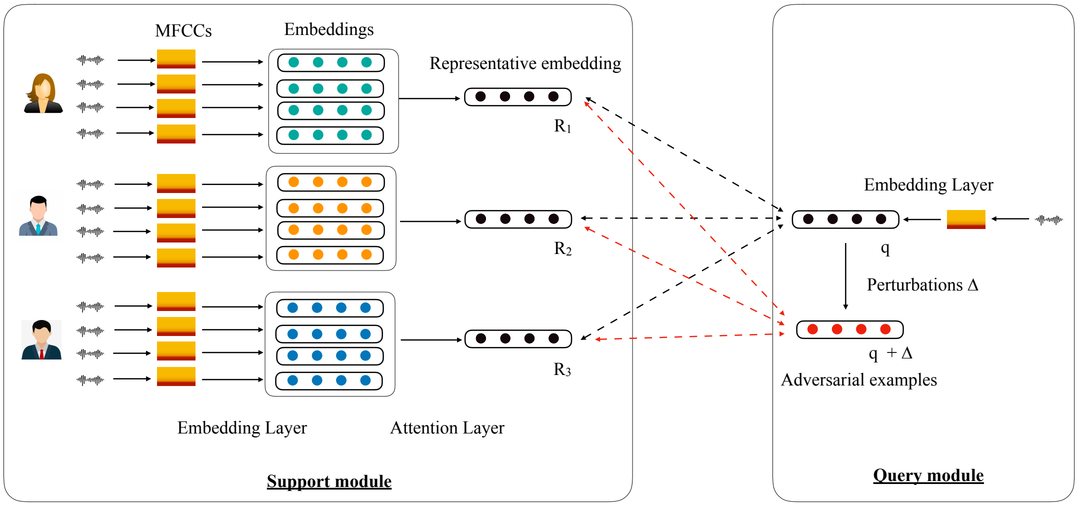


Figure 1: The overall framework of AFEASI.

space in which instances of the same speaker are embedded as close to each other as possible and as far as possible from instances of the other speakers. To further comprehensively utilize the training data, we introduce dynamic adversarial perturbations on the query instances to enhance the generalization of *AFEASI* through improving its robustness against unseen instances. To better visualize this part, adversarial learning is highlighted in red in the framework.

3.2 Embedding Representation Learning

In this section, we discuss how to construct an embedding given a piece of audio x_i.

We first convert the audio signal into frequency domains by constructing the mel frequency cepstral coefficients (MFCCs) [37] as acoustic features, which is denoted as $\boldsymbol{m}_{x_i}$. A 2D-convolutional layer is first utilized to extract informative features from the raw MFCC. Then the resulting feature maps are fed into an activation layer to introduce non-linearity. We further employ residual shortcut connection [11] to derive the representations for the audio MFCC. Equation 1 summarizes the key operations as follows:

$$C_1 = \text{Relu}(\text{Relu}(\text{Conv}_1(\boldsymbol{m}_{x_i})) + \boldsymbol{m}_{x_i}), \tag{1}$$

where Relu$(\cdot)$ and Conv$_1(\cdot)$ are the activation layer and the 2D-convolutional layer, respectively. To comprehensively distill the local features, we repeat the above residual-based covolutional operations for H times as:

$$C_h = \text{Relu}(\text{Relu}(\text{Conv}_h(C_{h-1})) + C_{h-1}), h > 1, \tag{2}$$

where C_h is the feature maps at the h-th convolutional layer. Finally, the embedding $\boldsymbol{E}_{x_i}$ can be constructed by flattening the feature maps C_H at the H-th convolutional layer, thereby serving as the representation of the input audio x_i.

3.3 Representative Embedding Construction

As shown in the support module of the framework, for each speaker, we aim to derive a representative embedding, which summarizes the acoustic biometric of the speaker. We develop an aggregation attention layer to learn the importance weights across each audio embedding of a particular speaker. Formally, the aggregation attention layer can be represented as follows:

$$\alpha_i = \text{softmax}(\boldsymbol{c} \cdot \tanh(\boldsymbol{W} \cdot \boldsymbol{E}_{x_i} + \boldsymbol{b})), \tag{3}$$

$$\boldsymbol{E}_R = \sum_i \alpha_i \boldsymbol{E}_{x_i}, \tag{4}$$

where $\boldsymbol{W}$ and $\boldsymbol{b}$ are the parameters for computing the attention weights α_i. Each audio embedding $\boldsymbol{E}_{x_i}$ is first fed into a one-layer neural network. Its output, together with the context vector $\boldsymbol{c}$, are further utilized to generate the importance weight α_i for each audio embedding $\boldsymbol{E}_{x_i}$ through a softmax function. The aggregated embedding $\boldsymbol{E}_R$ is calculated as a weighted sum of the audio embeddings based on the learned importance weights.

3.4 Few-Shot Learning

In this section, we discuss how to model the speaker identification task as a few-shot learning problem. A metric learning-based few-shot learning framework is employed in this work, which is composed of two modules, i.e., a support module and a query module. As shown in Figure 1, we first randomly sample a set of speakers from the training set as the start to construct the support module. For each speaker in the support module, we further randomly sample k pieces of his audio instances and derive the corresponding MFCCs. These MFCCs are further fed into an embedding layer so we can use a fixed length vector to represent each audio instance. To comprehensively represent the acoustic feature of a speaker, we utilize the attention mechanism to aggregate his acoustic embeddings. In the query module, we randomly select a piece of audio from a speaker, which is one of the speakers in the support module. We feed it into the embedding layer to derive the audio embedding. We then compare the distances between the query embedding and all the representative embeddings in the support module. The distances then are utilized to measure the relegation distribution over all speakers int support module. Model is optimized by such iterative comparisons and reasoning between the support and query modules.

In the comparisons and reasoning, we seek to separate audio embeddings in such a way that embeddings from different speakers are far from each other and embeddings from the same speaker are as close as possible in the hidden space. We achieve this by leveraging metric learning. In particular, the predicted probability of query q belonging to speaker k is given by:

$$p(y_k|q) = \frac{\exp(-d(q, R_k))}{\sum_{k'} \exp(-d(q, R_{k'}))}, \tag{5}$$

where $d(q, R_k)$ is the euclidean distance between the embedding $\boldsymbol{E}_q$ of query q and the representative embedding $\boldsymbol{E}_{R_k}$ of speaker k.

The loss function is then defined as the cross entropy between the predictions and the ground truth.

$$L(\Theta) = -\sum_k g(y_k|q) \log p(y_k|q, S, \Theta), \tag{6}$$

where $g(y_k|q)$ is probability that q goes to speaker k, which can be derived from the ground truth, and S denotes a set of audio representatives in the support module.

3.5 Adversarial Training

The goal of employing adversarial training is to allow the identification system not only get optimized by the instances in the training data, but also be robust to unseen adversarial perturbations. To enhance the robustness, we enforce the model to perform consistently well even when the adversarial perturbations are presented. To achieve this goal, we further optimize the model to minimize the objective function with the perturbed parameters. Formally, we define the objective function with adversarial examples incorporated as:

$$\begin{aligned} L_{adv}(S, q|\Theta) &= L(S, q|\Theta) + \lambda L(S, q + \Delta_{adv}|\Theta), \\ \text{where } \Delta_{adv} &= \arg \max_{\Delta, \|\Delta\| \le \epsilon} L(S, q + \Delta|\Theta), \end{aligned} \tag{7}$$

where Δ denotes the perturbations on the query instances, $\epsilon \ge 0$ controls the magnitude of the perturbations, and Θ denotes the model parameters. In this formulation, the adversarial term $L(S, q + \Delta_{adv}|\Theta)$ can be treated as a model regularizer, which stabilizes the identification performance. We use λ 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 expressed as playing a minimax game:

$$\Theta_{opt}, \Delta_{opt} = \arg \min_{\Theta} \max_{\Delta, \|\Delta\| \le \epsilon} L(S, q|\Theta) + \lambda L(S, q + \Delta|\Theta), \tag{8}$$

where the learning algorithm for model parameters Θ is the minimizing player, and the procedure to derive perturbations Δ acts as the maximizing player, which aims to identify the worst-case perturbations against the current model. The two players alternately play the game until convergence.

Constructing Adversarial Perturbations. Given a support set S and a query q, the problem of constructing adversarial perturbations Δ_{adv} is formulated as maximizing

$$\ell_{adv}(S, q|\Delta) = \sum_i g(y_i|q) \log p(y_i|q + \Delta, S, \hat{\Theta}), \tag{9}$$

where $\hat{\Theta}$ denotes a set of current model parameters. As it is difficult to get the exact optimal solutions of Δ_{adv}, we employ the fast gradient method proposed in [8], a common choice in adversarial training [12, 18, 19, 22], to estimate Δ_{adv}. The idea is to approximate the objective function around Δ as a linear function. To maximize the approximated linear function, we need to move towards the gradient direction of the objective function with respect to Δ. With the max-norm constraint $\|\Delta\| \le \epsilon$, we approximate Δ_{adv} as:

$$\Delta_{adv} = \epsilon \frac{\tau}{\|\tau\|}, \text{ where } \tau = \frac{\partial \ell_{adv}(S, q|\Delta)}{\partial \Delta}. \tag{10}$$

Learning Model Parameters. We now consider how to learn model parameters Θ. The local objective function to minimize for a query q given a support set S is as follows:

$$\begin{aligned} \ell_{adv}(S, q|\Theta) &= \sum_i g(y_i|q) \log p(y_i|q, S, \Theta) \\ &+ \lambda \sum_i g(y_i|q) \log p(y_i|q + \Delta_{adv}, S, \Theta), \end{aligned} \tag{11}$$

where Δ_{adv} is obtained from Equation 10. We can obtain the SGD update rule for Θ:

$$\Theta = \Theta - \eta \frac{\partial \ell_{adv}(S, q|\Theta)}{\partial \Theta}, \tag{12}$$

where η denotes the learning rate.

Algorithm 1: Parameter optimizations

```
Input: Training instances $D$, max iteration $iter_{\max}$;
Output: Model parameters $\Theta$
Initialization: initialize $\Theta$ with Normal distribution $N(0,0.01)$,
  $iter = 0$, $\Theta_{opt} = \Theta$, $L_{opt} = L_{vali}$;
repeat
    foreach support and query $S$, $q$ do
        // Constructing adversarial perturbations;
        $\Delta_{adv} \leftarrow$ Equation 10;
        // Updating model parameters;
        $\Theta \leftarrow$ Equation 12;
    if $L_{vali} < L_{opt}$ then
        $L_{opt} = L_{vali}$;
        $\Theta_{opt} = \Theta$;
    $iter + +$;
until $iter > iter_{max}$;
Return $\Theta_{opt}$;
```

Algorithm 1 summarizes the training process. In each training step, we randomly construct support set S and a query q. We then construct adversarial perturbations and optimize model parameters in a sequential order. The training involves multiple training steps and stops until reaching a certain number of training epochs. The parameters achieving the best performance on the validation dataset are utilized for evaluations.

4 EXPERIMENTS

In this section, we conduct extensive experiments on a real-world dataset to evaluate the performance of *AFEASI*.

4.1 Datasets and Experimental Settings

The experiments are conducted on the LibriSpeech dataset[2]. The audio data is derived from reading audio books from the LibriVox project. Table 2 shows the statistics of the dataset.

[2] LibriSpeech: http://www.openslr.org/12

We follow [1] to extract acoustic features from the raw audios. We convert all audio to streams at a 22 kHz sampling rate for consistency. The spectrograms are then generated by sliding window protocol by a hamming window. The width of the hamming window is 25 ms with a step size 10 ms. To remove the duplicated spectrograms coefficients, we further conduct discrete cosine transform. As a convention, 20 coefficients are kept at each time step as the acoustic features for the following speaker identification. The mel frequency cepstral coefficients (MFCCs) are constructed from the raw audio input without any pre-processing such as silence removal etc.

For each speaker, we randomly shuffle his/her audios. 80% of the audios are used to construct the training data. 10% are used for validation and the remaining 10% are used as the test data for evaluation. Table 3 shows the main parameters and their default values to tune in the experiments.

4.2 Comparative Baselines

To evaluate the performance of *AFEASI*, the following eleven methods are adopted as baselines, including seven conventional neural network-based methods, one few-shot learning-based method, one waveform-based method, and two variants of *AFEASI*.

Conventional neural network-based methods:

- **1D-CNN**. Multiple layers of 1D-CNN are utilized to construct audio embeddings from MFCCs, where the convolution is conducted along the time dimension. Global average pooling [20] is employed for aggregation before feeding into an output layer, where neutrons are equal to the total speakers for identification.
- **2D-CNN**. Different from 1D-CNN, 2D-CNN [16] is utilized to extract acoustic features from MFCCs.
- **LSTM** applies recurrent neural networks to investigate the acoustic frequency dependencies along all time steps. In the experiments, a bidirectional LSTM [28] is utilized to model such frequency dynamics along the time dimension and build audio embeddings.
- **Attentive-LSTM (A-LSTM)** differs from the LSTM method by introducing an attention layer [2] on top of the bidirectional LSTM to extract important acoustic signals at different time steps.
- **Attentive-CRNN (A-CRNN)** first utilizes a layer of 1D-CNN to extract local features at each time step and further builds an attentive LSTM model on top of such features to construct audio embeddings.
- **Self-attention (SA)** also seeks to extract audio embeddings by studying the frequency dynamics along the time dimension. More precisely, the self-attention technique [32] is utilized, where the same MFCC is considered as the input, query, and value matrices. Finally, the average of the fused vectors via self-attention operations serves as the audio embedding.
- **Attentive self-attention (A-SA)** first utilizes the self-attention technique [32] to fuse the acoustic vectors at different time steps. A weighted sum of the fused vectors over all time steps serves as the audio embedding.

Few-shot learning-based method:

- **Prototypical network (PN)** [1] adopts 2D-CNN as the building block to construct audio embeddings and applies prototypical loss [29] to learn from limited training data.

Raw waveform-based method:

- **Sincnet (SC)** [24] identifies speakers by directly training on the raw waveform of audios.

AFEASI variants:

- ***AFEASI$_s$*** differs from *AFEASI* in the choice of perturbation injections and only injects noises into audio instances in the support module.
- ***AFEASI$_b$*** injects noises into audio instances in both support and query modules.

Among the eleven baseline methods, the 1D-CNN, 2D-CNN, LSTM, A-LSTM, A-CRNN, SA, and A-SA differ in how to construct the audio embedding representation from the MFCCs. The CNN based methods employ convolutional operations to extract the local informative and discriminative features from MFCCs. The LSTM, A-LSTM, SA, and A-SA methods depend on investigating the dependencies of MFCC intensities at different time steps to construct audio embedding representations. The A-RCNN method utilizes both CNN and RNN to extract acoustic features and form audio embeddings. For these seven methods mentioned above, the constructed audio embeddings are further fed into the prediction blocks to yield speaker recognition. We include these seven methods as baselines to investigate which one of them is the most effective in extracting discriminative acoustic features from MFCCs in the context of ASR. PN utilizes 2D-CNN as the building block to construct audio embedding representations. It differs from the first seven baselines in how to conduct predictions. It utilizes metric learning to boost ASR performance. Sincnet differs from all baselines in the sense that it learns from the raw waveform of audios rather than from MFCCs. *AFEASI$_s$* and *AFEASI$_s$* are variants of *AFEASI*. They differ in where adversarial noises are injected. All parameters in these baselines are best tuned utilizing grid search.

4.3 Identification Performance

In this section, we evaluate the performances of *AFEASI* against different baseline methods. We adopt accuracy as the evaluation metric. Given a set of test audio instances, the accuracy *acc* is:

$$acc = \frac{\text{correctly identified test instances}}{\text{total test instances}}. \tag{13}$$

In this section, we investigate which technique is more effective on extracting informative acoustic biometric features from MFCCs. In particular, we compare 3 types of different methods, i.e., CNN, LSTM, and self-attention-based methods. Moreover, we also investigate the effectiveness of attention mechanisms on acoustic feature constructions. To further investigate how effective to directly identify speakers based on raw waveform of audios, we further include SC into the comparisons. To comprehensively compare these techniques on speaker identification, we vary the length of the audio instances from 1 second to 9 seconds with 2 seconds as the step size. Table 4 shows the corresponding performances on LibriSpeech. While the top seven rows show the performances of methods based on MFCCs, last row shows the performance of SC, which is waveform-oriented.

For MFCC-oriented methods, we have five observations. First, the longer the instance, the higher accuracy each method can achieve.

Table 2: The statistics of the experimental dataset.

Datasets	#(Female Speakers)	#(Male Speakers)	#(Total Speakers)	Total Hours	Per-speaker Minutes
LibriSpeech	125	126	251	~100 hours	~25 minutes

Table 3: Main parameters of AFEASI in the experiments after fine-tuning.

Parameters	Value	Parameters	Value
Learning rate η	0.01	Number of epochs	20
Regularizer weight λ	1	Perturbation bound ϵ	0.01
Way number K	150	Shot Number N	10

Table 4: Accuracy on test set over different audio embedding construction methods

Method	1s	3s	5s	7s	9s
1D-CNN	0.9021	0.9702	0.9853	0.9927	0.9931
2D-CNN	0.9038	0.9686	0.9780	0.9823	0.9879
LSTM	0.8650	0.9551	0.9607	0.9823	0.9888
A-LSTM	0.8848	0.9698	0.9819	0.9905	0.9922
A-CRNN	0.9198	0.9594	0.9720	0.9767	0.9810
SA	0.7537	0.8736	0.9107	0.9383	0.9405
A-SA	0.7886	0.9159	0.9435	0.9594	0.9642
SC	0.8147	0.8773	0.8806	0.8913	0.8991

It applies to all seven methods with different embedding construction strategies. It makes sense because the longer each audio instance, the richer acoustic information we have collected from each instance. Training, supported by rich acoustic information, contributes to high accuracy. The second observation is that the SA method achieves the worst accuracy performance on all different duration settings. The SA method depends on feature fusions to learn inter-dependent feature representations. However, a piece of audio, especially a short one, could contain a notable portion of silence which do not contain any distinguishing information. Feature fusions with such uninformative and misleading features lead to defective accuracy performance. The third observation is that all methods, except SA, work well when instances are 3 seconds long or longer than that. It demonstrates that audios with at least 3 seconds might be informative enough to construct a speaker's acoustic biometric. Moreover, by comparing LSTM with A-LSTM and A-CRNN, we also notice that it benefits the accuracy performance by adding an attention layer, especially when the audio instance is relatively short. When each instance is only 1 second long, the accuracy of the LSTM method is only 0.8650. By distinguishing important information at different time steps, A-LSTM and A-CRNN improve the accuracy to 0.8848 and 0.9198, respectively. Analogously, we find similar performance improvement when comparing SA with A-SA. The last observation is that the accuracy performance of LSTM, A-LSTM, SA, and A-SA are more sensitive to short audios than CNN-based methods. For example, when each instance is only 1 second long, the accuracy of these methods are only 0.8650, 0.8848, 0.7537, and 0.7886, respectively. It can be explained by the silence in the audio instances. When the audio instance is very short, each instance contains limited informative acoustic features. Therefore, short audio instances are more vulnerable to noises such as silence. In such scenarios, the frequency dynamics over time captured by LSTM and self-attention-based methods are less reliable and robust than the local features captured by CNN-based methods. The raw waveform-based method, SC, does not work very well generally compared with MFCC-based methods. SC skips the construction of MFCC, which involves fast Fourier transform and other hard-to-learn procedures, to learn speaker identification. It is still a daunting task since raw waveform-based methods are deemed to require a huge amount of training data in order to achieve success.

4.4 Performance with Limited Training Data

In this section, we investigate the performance of all methods when facing a shortage of data for training. In order to make instant identification response, we fix the length of each audio instance to 3 seconds. We vary the total number of training instances per speaker from 20 to 60. If the total training instances per speaker is only 20 and each instance is 3 seconds long, there are only 60 seconds audios used for training for each speaker. When we relax the number of training instances per speaker to 40 and 60, 120 seconds and 180 seconds long cumulative audios will be used for training per speaker, respectively. Figure 2 shows the accuracy performance for all methods on different settings.

We observe that the fewer training instances we have for a speaker, the lower accuracy we achieve for all methods. For example, when we have 180 seconds long training instances for a speaker, the accuracy of 1D-CNN can reach as high as about 0.9493. However, when the training instances are reduced to 120 and 60 seconds per speaker, the accuracy is only 0.9398 and 0.8645, respectively. We observe a similar performance drop for 2D-CNN, LSTM, A-LSTM, SA, A-SA, and SC. These observations demonstrate that the strong discriminative power of deep learning models significantly depends on the availability of a sufficient amount of training data. When only limited instances are provided for training, the performance might be far from expectations. Without applying few-shot learning mechanisms, 1D-CNN achieves the best accuracy performance on the three settings. This results from its simple network structure design, which is light-parameter dependent. The prototypical network, a metric-learning-based few-shot leaning method, achieves slightly higher accuracy. Its accuracy reaches about 0.93 as 60 seconds training instances are present for training per speaker. The high accuracy performance demonstrates the advantage of adopting few-shot learning, which fully utilizes the limited instances during training. Among all methods, *AFEASI* and its variants achieve the highest accuracy on all three settings, especially when the training instances are limited. Its accuracy is as high as about 0.95 for the setting of 60 seconds per speaker. This demonstrates the effectiveness of adopting attentive few-shot learning and adversarial learning when training from limited data.

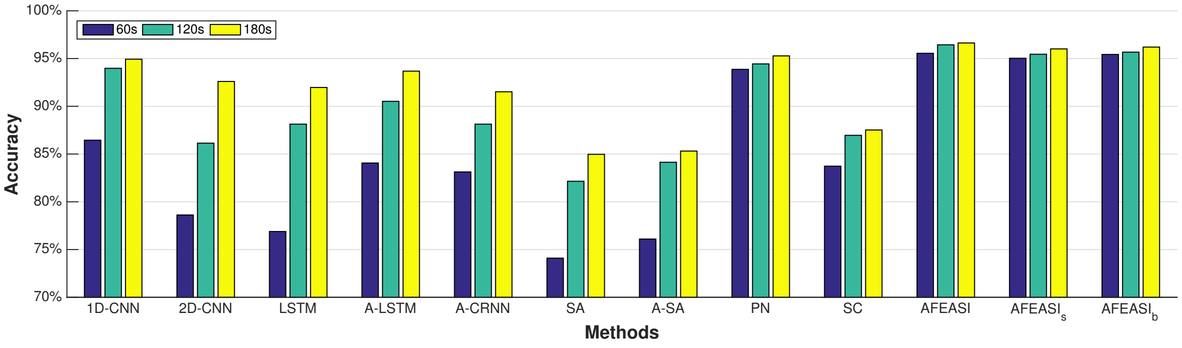


Figure 2: The accuracy of each method with different total training data per speaker on LibriSpeech.

Table 5: Accuracy on test set by injecting Gaussian noise

Methods	$AFEASI_{gaussian}$					$AFEASI$	$AFEASI_{-}$
τ	1e-6	1e-5	1e-4	1e-3	1e-2	N/A	N/A
60s	0.9432	0.9498	0.9481	0.9376	0.9317	0.9555	0.9411
120s	0.9522	0.9556	0.9532	0.9491	0.9487	0.9644	0.9512
180s	0.9587	0.9630	0.9620	0.9553	0.9531	0.9663	0.9563

4.5 Perturbation Injection Choice

In this section, we investigate and compare the effectiveness of different choices for injecting adversarial perturbations. *AFEASI* only injects perturbations into query instances. $AFEASI_s$ only injects perturbations into support instances, while $AFEASI_b$ injects dynamic perturbations into both query and support instances.

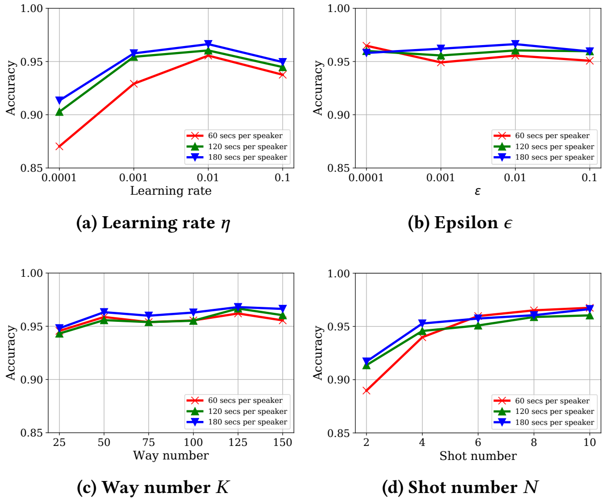
The last three groups of Figure 2 show the accuracy performance of *AFEASI* and its two variants on different settings. We observe that injecting adversarial perturbations into query instances only is effective enough to enhance identifications. For example, as training instances are as limited as 60 seconds per speaker. The accuracy of *AFEASI*, $AFEASI_b$, and $AFEASI_s$ are all as high as and close to 0.955. While injecting perturbations into only query instances allows us to quickly generate adversarial examples during training, as compared to generating noises in support instances or both query and support instances. Therefore, in this work we choose to inject perturbations into only query instances in consideration of computational efficiency.

4.6 Effectiveness of Adversarial Training

In this section, we compare the effectiveness of data augmentation between conducting adversarial training and applying conventional audio augmentation methods.

To conduct conventional data augmentation, we inject random Gaussian noises to raw audios with different parameters τ that controls the intensity of injected noises. Formally, the augmented audio piece x_{au} can be represented as $x_{au} = x + \tau N_g$, where x and N_g are the original audio and the Gaussian noise, respectively. Finally, $AFEASI_{gaussian}$ denotes the conventional approach by replace the adversarial training with the participation of augmented data injected by Gaussian noises. In addition, we use $AFEASI_{-}$ to indicate the method simply removing adversarial training from *AFEASI* and evaluate the impact of adversarial training.

Table 5 shows the performance comparisons among $AFEASI_{-}$, $AFEASI_{gaussian}$, and *AFEASI*. We observe that data augmentations by injecting Gaussian noises help address the data shortage issue. For example, the accuracy of $AFEASI_{-}$ is 0.9411 for the 60-seconds setting. When setting τ to $1e-5$ and $1e-4$, the accuracy of $AFEASI_{gaussian}$ improves to 0.9498 and 0.9841, respectively. However, the effectiveness of such data synthesis is sensitive to the setting of the weighting factor τ. For example, when τ is set to $1e-3$ and $1-e2$, heavier noises are injected into the raw audios. The injected noises obscure the raw informative signals and lead to worse accuracy performance. This could make it challenging to select a good τ in practice since it only helps with a narrow range of effective settings. In addition, we notice that *AFEASI* still achieves the highest accuracy performance over all three settings. This demonstrates that intentional adversarial noises are more helpful in improving identification performance.



(a) Learning rate η

(b) Epsilon ϵ

(c) Way number K

(d) Shot number N

Figure 3: Parameter sensitivity studies on LibriSpeech

4.7 Sensitivity Study

In this section, we examine how different choices of parameters influence the performance of *AFEASI*. Except for the parameter being tested, we set other parameters at the default values (see in Table 3). Figure 3 shows the evaluation results as a function of one selected parameter when fixing others.

Figure 3a shows the accuracy performances of *AFEASI* when we change the learning rate. It may get stuck to local optimal and lead to sub-optimal performance when the learning rate is either too small or too large. In this work, we set it as 0.01 with the consideration of the performance. Figure 3b shows the effect of varying ϵ, which controls the magnitude of the perturbations. *AFEASI* in general is not sensitive to the setting of ϵ and it achieves high accuracy performance with a wide range of ϵ from 0.0001 to 0.1. Figure 3c shows the performance of *AFEASI* when choosing different number of speakers in a training episode. We observe that *AFEASI* is not strictly sensitive to this parameter and it always achieves accuracy performance as high as around 0.95 for all settings with the parameter is larger than 25. Figure 3d shows the performance change via choosing different number of instances per speaker as references in the support module. We observe that the more instances we select to generate the speaker's acoustic biometric embedding as references, the higher accuracy we can achieve during the test in general. We also notice that the increase of accuracy performance saturates as the number of shots increases more than 6. This is because: at the beginning, a larger value of the shot number N brings a stronger representation power to express speaker's acoustic characteristic, but the further increase of shot number might only provide limited and repeated information.

4.8 Household Deployment

One notable application of ASR is to enable personalized services at different households. In such scenarios, audio-enabled devices, such as Echo Dot and Google Home, only need to serve several peoples in a household. Therefore, we may not have to include all the speakers as identification candidates. This could not only reduce the computation cost during inference and respond more quickly, but also significantly improve the identification accuracy. All these benefits depend on the flexibility of the identification model to accommodate only a portion of users as speaker candidates. All conventional deep learning methods mentioned in the baseline section fail to achieve this, since the output layer of these models is fixed with the number of neurons equal to the number of total speakers. *AFEASI* solves the issue by learning distance metric among different speaker candidates. The speaker with the smallest distance to the query instance among all candidates yields the prediction. In this way, *AFEASI* can enable fast and efficient speaker identification by considering only a small set of candidate speakers.

5 RELATED WORK

In this section, we discuss related works on automatic speaker recognition and few-shot learning.

5.1 Automatic Speaker Recognition

Most state-of-the-art solutions are based on the i-vector representation of speech segments [5], which contributed to significant improvements over the Gaussian Mixture Model-Universal Background Models (GMM-UBMs) [26]. Deep learning has shown remarkable success in speaker identification tasks recently. Deep speaker [17] takes filter bank coefficients as inputs, utilizes residual networks to extract audio embeddings, and employs triplet loss as the loss function to optimize the neural network. VGGVox [4] takes spectrograms as inputs. CNN based residual network is designed to extract audio embeddings. Contrastive loss is employed to optimize the training pairs in the network with pre-training using softmax classification. However, the number of training pairs can grow quadratically with the size of the dataset and elaborate pair selection heuristics are needed to make the training on large datasets feasible. Another Resnet-based model uses additive margin softmax [34] classification loss to improve the recognition accuracy in [35] and [9]. SincNet [24] utilizes convoluational neural networks to learn speaker recognition from raw audios. [1] is the most relevant work, which leverages prototypical network to conduct speaker recognition from limited training data.

The proposed method, *AFEASI*, differs from the above work by focusing on speaker identification by learning from limited instances. *AFEASI* leverages metric-learning few-shot learning to achieve competitive performance with limited training instances. In addition, adversarial training is adopted to improve the generalization and robustness of identification model.

5.2 Few-shot Learning

Recent deep learning-based few-shot learning approaches fall into three main categories: (1) metric-based approaches [15, 29, 30, 33], which try to learn a generalized distance metric. (2) model-based approaches [27], which use recurrent network with internal or external memory. (3) optimization-based approaches [25], which optimize model parameters explicitly for fast learning. Our model is most related to metric-based approaches.

A siamese neural network is utilized to conduct one-shot image classification in [15]. The siamese neural network is composed of two twin networks and their outputs are jointly trained on top of a similarity function to learn the relationship between pairs of data points. The Matching Network [33] makes classification predictions by comparing the input samples with a small labeled support set. The relation network [30] is similar to the siamese network, but differs by choosing a CNN to capture the relationship rather than a simple $L1$ distance. The prototypical network [29] defines a prototype vector to represent each class. The prototype vector is calculated as the mean vector of the support data samples in each class, without any differential weighting mechanisms.

Previous few-shot learning research mainly focuses on vision learning [6], text classification tasks [39], or entity predictions on knowledge graphs [36]. To the best of our knowledge, this work is the first research utilizing adversarial few-shot learning on ASR.

6 CONCLUSION

In this work, we study the problem of speaker identification with limited training data. To cope with the data deficiency issue, we utilize few-shot learning, which allows us to comprehensively utilize the limited training instances. In addition, to further enhance the generalization and robustness of the speaker identification model, we perform adversarial learning. Comprehensive experiments on a real-world dataset demonstrate a significant performance improvement of *AFEASI* with comparisons to eleven baseline methods.

ACKNOWLEDGMENTS

This project was supported by NSF DGE-1829071.

REFERENCES

[1] Prashant Anand, Ajeet Kumar Singh, Siddharth Srivastava, and Brejesh Lall. 2019. Few Shot Speaker Recognition using Deep Neural Networks. *CoRR* abs/1904.08775 (2019).

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

[3] Nicholas Carlini and David A. Wagner. 2018. 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*. 1–7.

[4] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. 2018. VoxCeleb2: Deep Speaker Recognition. *CoRR* abs/1806.05622 (2018). arXiv:1806.05622 http://arxiv.org/abs/1806.05622

[5] Najim Dehak, Patrick Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. 2011. Front-End Factor Analysis for Speaker Verification. *IEEE Trans. Audio, Speech & Language Processing* 19, 4 (2011), 788–798. https://doi.org/10.1109/TASL.2010.2064307

[6] Yan Duan, Marcin Andrychowicz, Bradly C. Stadie, Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. 2017. One-Shot Imitation Learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*. 1087–1098.

[7] Russell Feldman. 2018. Almost a quarter of Britons now own one or more smart home devices. https://yougov.co.uk/topics/technology/articles-reports/2018/08/10/almost-quarter-britons-now-own-one-or-more-smart-h.

[8] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *ICLR*.

[9] Mahdi Hajibabaei and Dengxin Dai. 2018. Unified Hypersphere Embedding for Speaker Recognition. *CoRR* abs/1807.08312 (2018). http://arxiv.org/abs/1807.08312

[10] John H. L. Hansen and Taufiq Hasan. 2015. Speaker Recognition by Machines and Humans: A tutorial review. *IEEE Signal Process. Mag.* 32, 6 (2015), 74–99.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. 770–778.

[12] Xiangnan He, Zhankui He, Xiaoyu Du, and Tat-Seng Chua. 2018. Adversarial Personalized Ranking for Recommendation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*. 355–364.

[13] Jae-Bok Kim and Jeong-Sik Park. 2016. Multistage data selection-based unsupervised speaker adaptation for personalized speech emotion recognition. *Eng. Appl. of AI* 52 (2016), 126–134.

[14] Bert Kinsella. 2018. Smart Speaker Owners Use Voice Assistants Nearly 3 Times Per Day. https://voicebot.ai/2018/04/02/smart-speaker-owners-use-voice-assistants-nearly-3-times-per-day/.

[15] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. 2015. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, Vol. 2.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*. 1106–1114.

[17] Chao Li, Xiaokong Ma, Bing Jiang, Xiangang Li, Xuewei Zhang, Xiao Liu, Ying Cao, Ajay Kannan, and Zhenyao Zhu. 2017. Deep Speaker: an End-to-End Neural Speaker Embedding System. *CoRR* abs/1705.02304 (2017). arXiv:1705.02304

[18] Ruirui Li, Liangda Li, Xian Wu, Yunhong Zhou, and Wei Wang. 2019. Click Feedback-Aware Query Recommendation Using Adversarial Examples. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*. 2978–2984.

[19] Ruirui Li, Xian Wu, and Wei Wang. 2020. Adversarial Learning to Compare: Self-Attentive Prospective Customer Recommendation in Location based Social Networks. In *Proceedings of WSDM, Houston, Texas, USA, February 3-7*.

[20] Min Lin, Qiang Chen, and Shuicheng Yan. 2014. Network In Network. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

[21] Zheli Liu, Zhendong Wu, Tong Li, Jin Li, and Chao Shen. 2018. GMM and CNN Hybrid Method for Short Utterance Speaker Recognition. *IEEE Trans. Industrial Informatics* 14, 7 (2018), 3244–3252.

[22] Sungrae Park, Jun-Keon Park, Su-Jin Shin, and Il-Chul Moon. 2018. Adversarial Dropout for Supervised and Semi-Supervised Learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*. 3917–3924.

[23] Vishal M. Patel, Rama Chellappa, Deepak Chandra, and Brandon Barbello. 2016. Continuous User Authentication on Mobile Devices: Recent progress and remaining challenges. *IEEE Signal Process. Mag.* 33, 4 (2016), 49–61.

[24] Mirco Ravanelli and Yoshua Bengio. 2018. Speaker Recognition from Raw Waveform with SincNet. In *2018 IEEE Spoken Language Technology Workshop, SLT 2018, Athens, Greece, December 18-21, 2018*. 1021–1028.

[25] Sachin Ravi and Hugo Larochelle. 2016. Optimization as a model for few-shot learning. (2016).

[26] Douglas A. Reynolds, Thomas F. Quatieri, and Robert B. Dunn. 2000. Speaker Verification Using Adapted Gaussian Mixture Models. *Digital Signal Processing* 10, 1-3 (2000), 19–41. https://doi.org/10.1006/dspr.1999.0361

[27] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy P. Lillicrap. 2016. Meta-Learning with Memory-Augmented Neural Networks. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*. 1842–1850.

[28] Mike Schuster and Kuldip K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Trans. Signal Processing* 45, 11 (1997), 2673–2681.

[29] Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical Networks for Few-shot Learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*. 4080–4090.

[30] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. 2018. Learning to Compare: Relation Network for Few-Shot Learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. 1199–1208.

[31] J. Vanschoren. 2019. *Meta-learning*. Springer, Germany, 39–68.

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*. 6000–6010.

[33] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. 2016. Matching Networks for One Shot Learning. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*. 3630–3638.

[34] Feng Wang, Weiyang Liu, Hanjun Dai, Haijun Liu, and Jian Cheng. 2018. Additive Margin Softmax for Face Verification. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*.

[35] Weidi Xie, Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. 2019. Utterance-level Aggregation for Speaker Recognition in the Wild. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*. 5791–5795.

[36] Wenhan Xiong, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2018. One-Shot Relational Learning for Knowledge Graphs. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*. 1980–1990.

[37] Min Xu, Ling-Yu Duan, Jianfei Cai, Liang-Tien Chia, Changsheng Xu, and Qi Tian. 2004. HMM-Based Audio Keyword Generation. In *Advances in Multimedia Information Processing - PCM 2004, 5th Pacific Rim Conference on Multimedia, Tokyo, Japan, November 30 - December 3, 2004, Proceedings, Part III*. 566–574.

[38] Hiromu Yakura and Jun Sakuma. 2018. Robust Audio Adversarial Example for a Physical Attack. *CoRR* abs/1810.11793 (2018). http://arxiv.org/abs/1810.11793

[39] Mo Yu, Xiaoxiao Guo, Jinfeng Yi, Shiyu Chang, Saloni Potdar, Yu Cheng, Gerald Tesauro, Haoyu Wang, and Bowen Zhou. 2018. Diverse Few-Shot Text Classification with Multiple Metrics. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*. 1206–1215.