# Quasi-Newton Adversarial Attacks on Speaker Verification Systems

Keita Goto, Nakamasa Inoue
Tokyo Institute of Technology
E-mail: goto.k.al@m.titech.ac.jp, inoue@c.titech.ac.jp

***Abstract*—This paper proposes a framework for generating adversarial utterances for speaker verification systems. Our main idea is to formulate an optimization problem to generate adversarial utterances that fool speaker verification models and solve it by a second-order optimization method. We first present our algorithm, which uses the first-order Gauss-Newton method, and then extend it to second-order Quasi-Newton methods. Our experiments on the VoxCeleb 1 dataset show that the proposed method can fool a speaker verification system with a smaller degree of perturbations than those of conventional methods. We also show that second-order optimization methods are effective for finding small perturbations.**

## I. Introduction

Speaker verification is a task to determine whether a test utterance is from the same speaker as an enrollment utterance. It has a wide range of applications in biometric authentication, audio surveillance, and robotics. Recent research has proposed highly precise verification methods based on statistical models, such as Gaussian mixture models for i-vectors [1] and time-delay neural networks for x-vectors [2]. Among these methods, i-vectors are known to be computationally efficient and are often used with lightweight devices, including smartphones [3].

To improve the security of devices and data, verification systems should be robust against perturbations on inputs. However, previous studies have found that small perturbations, so-called adversarial perturbations, can fool the systems. Adversarial perturbations were first found on image recognition systems in the field of computer vision. After that, finding the minimum degree of adversarial perturbations for each system has become an important research topic.

To minimize the degree of perturbations, most algorithms use first-order gradient descent methods. Examples of these algorithms include the fast gradient sign method (FGSM) [4], the Carlini and Wagner (C&W) method [5], and DeepFool [6]. They often generate very small adversarial perturbations. It is also known that larger networks tend to have smaller adversarial perturbations. This conversely means in general that it is difficult to fool systems using simple statistical models.

For speaker verification, Li et al. [7] found that even on i-vector + probabilistic linear discriminant analysis systems, adversarial perturbations exist. Although the reported degree of perturbations is not very small, this shows that there is a demand to explore adversarial perturbations on such simple statistical models more in depth. This motivates us to propose a new framework that aims to generate the minimum adversarial perturbations that fool speaker verification systems.

In this paper, we present a framework to generate adversarial perturbations on speaker verification systems. The proposed framework formulates an optimization problem to generate adversarial utterances and solves the problem by non-linear optimization methods. Our experiments on the VoxCeleb 1 dataset show that the proposed method is able to fool a speaker verification system with a smaller degree of perturbations compared with the conventional method. We also show that second-order optimization methods are effective for finding small-degree perturbations.

Our contributions are summarized as the following three points:

1) We propose a framework to generate adversarial utterances on speaker verification systems and define an optimization problem to obtain adversarial perturbations.
2) We propose two types of attacks, namely Gauss-Newton and Quasi-Newton attacks. These attacks apply different levels of approximation based on Taylor expansion to the optimization problem.
3) We conduct comparable experiments on the VoxCeleb 1 dataset and show that our method fools speaker verification models with a smaller degree of perturbation compared with the state-of-the-art method in [7].

## II. Related Work

### A. Adversarial Attacks

Adversarial attack is a method to generate samples intended to be misclassified by existing neural models. The purpose of adversarial attacks is to make a prediction fail by adding a slight perturbation to the input. In the white-box condition, if we know about the target model's parameters, adversarial perturbation is generated by calculating the gradient to increase classification losses.

For image recognition, Goodfellow et al. [4] proposed the FGSM as an adversarial attack method. The FGSM generates an adversarial perturbation in only one gradient calculation, but Moosavi-Dezfooli et al. [6] showed that an iterative method, named DeepFool, allows for smaller perturbations to be misidentified. Carlini and Wagner [5] presented another attack that introduces a unique loss function and can defeat models defended using distillation [8]. Pin-Yu et al. [9] proposed elastic-net attacks, which generate less-discriminating samples by elastic-net regularization. Yao et al. [10] proposed an efficient attack method that considers the trust region.

Adversarial attacks that target speech have been studied for speech recognition. Carlini and Wagner [11] attacked the speech-to-text model called DeepSpeech [12]. Qin et al. [13] extended the method of Carlini and Wagner for the real world with reverberation.

For speaker verification, Kreuk et al. [14] attacked a text-dependent model [15] constructed by long short-term memory [16] and cosine similarity using the FGSM. Li et al. [7] proved that the FGSM could fool typical models, such as Gaussian mixture model (GMM) i-vectors [1] and x-vectors [2], but further experiments using other methods have not been done.

In these former approaches, researchers assumed that the entire model is locally linear, but it actually has a more complex shape. We focused on modeling more accurately with approximations to achieve realistic computation times.

### *B. Speaker Verification*

Speaker verification, which is also called automatic speaker verification, aims to verify whether test and enrollment utterances are from the same speaker. We work under a text-independent condition, without any assumptions about the content of the speech. In general, a speaker verification system consists of a feature extractor for obtaining speaker identity features from an utterance and a similarity scoring function for these features.

For text-independent speaker verification, research in recent decades has proposed statistical methods using GMMs [17] for feature extraction. Dehak et al. showed that the accuracy can be improved by using i-vectors, which are vectors with dimensions reduced from those of GMM supervectors [18]. For measuring the speaker identity of these features, distance functions and probabilistic linear discriminant analysis [19] are employed as a similarity scoring function.

Recently, researchers have proposed highly precise methods based on deep neural networks, such as x-vectors. Convolutional neural networks are valid not only for images [20] but also for audio time-frequency features like spectrograms [21]. These neural models are highly accurate, but they require a lot of computational resources. Therefore, in recent years, statistical models continue to be studied, as in [22], because they are lightweight.

## III. Proposed Method

This section presents the proposed adversarial attacks, namely *Quasi-Newton attacks*, on speaker verification systems. Our main idea is to formulate an optimization problem to generate adversarial utterances, and solve it by introducing Quasi-Newton methods. In this section, we first describe the notation and settings, and then present the proposed method.

### *A. Notations and Settings*

Let $(\boldsymbol{x}_e, \boldsymbol{x}_t)$ be a paired enrollment utterance and test utterance to be verified. We assume that the speaker verification system determines whether these two utterances are from the same speaker by the sign of a discriminative function $f$ given by

$$f(\boldsymbol{x}_e, \boldsymbol{x}_t) = S(E(\boldsymbol{x}_e), E(\boldsymbol{x}_t)) - \theta, \tag{1}$$

where $E(\boldsymbol{x}) \in \mathbb{R}^d$ is the embedding of an utterance $\boldsymbol{x}$, $S$ is a similarity metric between embeddings, and $\theta$ is a threshold.

The proposed adversarial attacks aim to fool the speaker verification system by adding a slight perturbation $\boldsymbol{\delta}$ to the test utterance as

$$\tilde{\boldsymbol{x}}_t = \boldsymbol{x}_t + \boldsymbol{\delta}. \tag{2}$$

Note that this setting mimics a realistic attack as described in [7], where the enrollment utterance is first given to the system, and then an attacker tries to fool the system by feeding it an adversarial utterance.

Among the perturbations, our main interest is in finding the *minimum* perturbation $\boldsymbol{\delta}^*$ that fools the system. Specifically, $\boldsymbol{\delta}^*$ is defined by

$$\boldsymbol{\delta}^* = \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \|\boldsymbol{\delta}\|_2 \text{ subject to } \operatorname{sign}(f(\boldsymbol{x}_e, \tilde{\boldsymbol{x}}_t)) \neq y, \tag{3}$$

where $y \in \{+1 \text{ (target)}, -1 \text{ (non-target)}\}$ is the ground truth label, that is, $y$ takes the value of $+1$ if and only if the two utterances $\boldsymbol{x}_e$ and $\boldsymbol{x}_t$ are from the same speaker. Note that the constraint $\operatorname{sign}(f(\boldsymbol{x}_e, \tilde{\boldsymbol{x}}_t)) \neq y$ means the decision by the system is wrong. Our objective is to solve the problem of Equation (3).

### *B. Proposed Adversarial Attacks*

#### *B-1. Algorithm Overview*

In speaker verification systems, directly solving Equation (3) is difficult because function $f$ is often assumed to be non-linear. Thus, we introduce an alternative problem to be solved:

$$\tilde{\boldsymbol{x}}_t = \underset{\boldsymbol{x}}{\operatorname{argmin}}\, g(\boldsymbol{x}_e, \boldsymbol{x}), \tag{4}$$

where $g(\boldsymbol{x}_e, \boldsymbol{x}) = |f(\boldsymbol{x}_e, \boldsymbol{x})y + \epsilon|$ and $\epsilon \simeq 0$ ($\epsilon > 0$) is a small positive constant to overturn the decision. Note that, because the constraint in Equation (3) is equivalent to $f(\boldsymbol{x}_e, \tilde{\boldsymbol{x}}_t)y < 0$, minimizing $|f(\boldsymbol{x}_e, \tilde{\boldsymbol{x}}_t)y + \epsilon|$ with respect to $\tilde{\boldsymbol{x}}_t$ approximately solves the original problem.

Algorithm 1 summarizes the overall procedure to obtain an adversarial utterance based on Equation (4). In this algorithm, the variable $\boldsymbol{x}$ is initialized by $\boldsymbol{x}_t$ and is updated as

$$\boldsymbol{x} \leftarrow \boldsymbol{x} - J(\boldsymbol{x})\nabla g(\boldsymbol{x}_e, \boldsymbol{x}) \tag{5}$$

**Algorithm 1**

**Input:** verification system $f(\boldsymbol{x}_e, \cdot)$, test utterance $\boldsymbol{x}_t$, label $y$
**Output:** adversarial example $\tilde{\boldsymbol{x}}_t$

```
x ← x_t
while f(x_e, x)y > 0 do
    x ← x − J(x)∇g(x_e, x)
end while
Return x̃_t = x
```

at each iteration with the coefficient $J(\boldsymbol{x})$, where $\nabla$ is applied only to the augmented second of $g$.

In the following, we present the Gauss-Newton adversarial attack based on the first-order optimization, and extend it to Quasi-Newton adversarial attacks based on the second-order optimization. The difference between them is in the approximation order of the Taylor expansion applied to $g(\boldsymbol{x}_e, \boldsymbol{x})$ to define $J(\boldsymbol{x})$.

*B-2. Gauss-Newton Adversarial Attack*

The Gauss-Newton adversarial attack applies the first-order Taylor expansion to $g$ for test utterance $\boldsymbol{x}_t$ as follows:

$$g(\boldsymbol{x}_e, \boldsymbol{x}) \simeq g(\boldsymbol{x}_e, \boldsymbol{x}_t) + \nabla g(\boldsymbol{x}_e, \boldsymbol{x}_t)(\boldsymbol{x} - \boldsymbol{x}_t). \quad (6)$$

With this approximation, Equation (4) has an analytical solution:

$$\boldsymbol{x} = \boldsymbol{x}_t - \frac{g(\boldsymbol{x}_e, \boldsymbol{x}_t)}{\|\nabla g(\boldsymbol{x}_e, \boldsymbol{x}_t)\|_2^2} \nabla g(\boldsymbol{x}_e, \boldsymbol{x}_t). \quad (7)$$

From this solution, we define the coefficient $J(\boldsymbol{x})$ in Algorithm 1 as

$$J(\boldsymbol{x}) = \frac{g(\boldsymbol{x}_e, \boldsymbol{x})}{\|\nabla g(\boldsymbol{x}_e, \boldsymbol{x})\|_2^2}. \quad (8)$$

Note that this update rule can be viewed as a simplified Gauss-Newton method and also can be viewed as an extension of the DeepFool image-generation algorithm [6] for speaker verification.

*B-3. Quasi-Newton Adversarial Attacks*

To improve the accuracy of approximation, a straightforward method is to apply the second-order Taylor expansion:

$$\begin{aligned} g(\boldsymbol{x}_e, \boldsymbol{x}) \simeq\ & g(\boldsymbol{x}_e, \boldsymbol{x}_t) + \nabla g(\boldsymbol{x}_e, \boldsymbol{x}_t)(\boldsymbol{x} - \boldsymbol{x}_t) \\ & + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}_t)^T H(\boldsymbol{x}_t) g(\boldsymbol{x}_e, \boldsymbol{x}_t)(\boldsymbol{x} - \boldsymbol{x}_t) \end{aligned} \quad (9)$$

where $H(\boldsymbol{x}_t)$ is a Hessian matrix at $\boldsymbol{x}_t$. With this approximation, we have the following solution:

$$\boldsymbol{x} = \boldsymbol{x}_t - H(\boldsymbol{x}_t)^{-1} \nabla g(\boldsymbol{x}_e, \boldsymbol{x}_t). \quad (10)$$

This shows that the coefficient $J(x)$ at each iteration should be defined by $J(\boldsymbol{x}) = H(\boldsymbol{x})^{-1}$. However, exact computation of the inverse of the Hessian matrix is time consuming in practice because speaker verification models often have a large number of parameters.

To efficiently and stably compute $H(\boldsymbol{x})^{-1}$, we introduced modified Quasi-Newton methods. Specifically, we use the Davidon-Fletcher-Powell (DFP) and the Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) methods with two modifications: 1) introducing stabilization parameter $\lambda$ and 2) employing a small step size to prevent large perturbations. These modifications are effective for adversarial utterance generation because the gradient $\nabla g(\boldsymbol{x}_e, \boldsymbol{x})$ at each iteration often needs to be sparse and small to find small perturbations. The definition of $g$ is also modified as $g(\boldsymbol{x}_e, \boldsymbol{x}) = |f(\boldsymbol{x}_e, \boldsymbol{x})y + \epsilon|^2$ to accelerate convergence speed.

**Algorithm 2**

**Input:** verification system $f(\boldsymbol{x}_e, \cdot)$, test utterance $\boldsymbol{x}_t$, label $y$
**Output:** adversarial example $\tilde{\boldsymbol{x}}_t$

```
x ← x_t
A ← A_0
while f(x_e, x)y > 0 do
    x ← x − αA∇g(x_e, x)
    A ← Ψ(A)
end while
Return x̃_t = x
```

Algorithm 2 summarizes the overall procedure to obtain an adversarial utterance with the modified Quasi-Newton methods. Note that the variable $A$ is used to retain an approximation of $H(x)^{-1}$. It is first initialized by an identity matrix, $A_0 = I$, and is updated at each iteration by $\Psi(A)$. The update function $\Psi$ is defined by either of the following formulas.

*DFP formula*: The update rule of the DFP formula [23] was proposed in 1959 as a Quasi-Newton method. It is given by

$$\Phi(A, \boldsymbol{x}, \boldsymbol{x}') = A + \frac{\boldsymbol{u}\boldsymbol{u}^T}{\boldsymbol{v}^T\boldsymbol{u} + \lambda} - \frac{A\boldsymbol{v}\boldsymbol{v}^T A}{\boldsymbol{v}^T A \boldsymbol{v} + \lambda}, \quad (11)$$

where $\boldsymbol{u} = \boldsymbol{x}' - \boldsymbol{x}$ and $\boldsymbol{v} = \nabla f(\boldsymbol{x}_e, \boldsymbol{x}') - \nabla f(\boldsymbol{x}_e, \boldsymbol{x})$. Note that $\lambda > 0$ is introduced to stabilize computation, because the absolute value of the denominator in each term of this rule tends to be small in the optimization steps to find small adversarial perturbations.

*L-BFGS formula*: The update rule of the BFGS formula [24] was proposed in 1970. In practice, BFGS often converges faster than DFP. It is given by

$$\begin{aligned} \Phi(A, \boldsymbol{x}, \boldsymbol{x}') = & \left(I - \frac{\boldsymbol{u}\boldsymbol{v}^T}{\boldsymbol{u}^T\boldsymbol{v} + \lambda}\right) A \left(I - \frac{\boldsymbol{v}\boldsymbol{u}^T}{\boldsymbol{u}^T\boldsymbol{v} + \lambda}\right) \\ & + \frac{\boldsymbol{u}\boldsymbol{u}^T}{\boldsymbol{u}^T\boldsymbol{v} + \lambda}, \end{aligned} \quad (12)$$

where the definitions of $\boldsymbol{u}$ and $\boldsymbol{v}$ are the same as those in the DFP formula. To further improve computational efficiency, we applied L-BFGS, as proposed in [25].

In the standard Quasi-Newton methods, the step size $\alpha$ is determined by applying a line search. However, this often chooses a large $\alpha$ and makes perturbations large. Thus, in our algorithm, the step size is determined in the same way as in the Gauss-Newton attack as follows:

$$\alpha = \frac{g(\boldsymbol{x}_e, \boldsymbol{x})}{\nabla g(\boldsymbol{x}_e, \boldsymbol{x})^T (A + \lambda' I) \nabla g(\boldsymbol{x}_e, \boldsymbol{x})}. \quad (13)$$

Compared to Equation 8, the approximated Hessian inverse matrix $A$ with a stabilizing parameter $\lambda'$ is introduced.

## IV. Experiments

### A. Evaluation Settings

We use the VoxCeleb 1 dataset [26] for evaluation, which consists of 148,642 utterances for training and 37,720 trials (enrollment-test utterance pairs) for testing. To fairly compare our method with a state-of-the-art method, we use exactly the

TABLE I
EER (%) OF THE SYSTEM WITH DIFFERENT PERTURBATION DEGREES $\varepsilon$.

| Method | $\varepsilon = 0$ | 0.001 | 0.002 | 0.005 | 0.01 | 0.02 | 0.05 | 0.1 | 0.2 | 0.5 | 1.0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| MFCC-ivec Attack [7] | 7.30 | 7.45 | 7.59 | 8.02 | 8.88 | 10.68 | 17.85 | 33.25 | 62.75 | 95.20 | 97.62 |
| Gauss-Newton Attack | 7.30 | 23.20 | 43.32 | 82.39 | 98.83 | 99.97 | **100.00** | **100.00** | **100.00** | **100.00** | **100.00** |
| Quasi-Newton Attack (DFP) | 7.30 | 24.19 | 46.12 | 87.08 | **99.63** | **100.00** | **100.00** | **100.00** | **100.00** | **100.00** | **100.00** |
| Quasi-Newton Attack (L-BFGS) | **7.30** | **24.24** | **46.16** | **87.13** | **99.63** | **100.00** | **100.00** | **100.00** | **100.00** | **100.00** | **100.00** |

TABLE II
AVERAGE PERTURBATIONS AND WORST-CASE PERTURBATIONS (%) ON MFCC FEATURES. LOWER VALUES SHOW BETTER PERFORMANCE.

| Method | Average | | Worst | |
|---|---|---|---|---|
| | $\bar{\rho}_1$ | $\bar{\rho}_2$ | $\bar{\bar{\rho}}_1$ | $\bar{\bar{\rho}}_2$ |
| MFCC-ivec Attack [7] | 9.91 | 6.94 | 15.24 | 10.59 |
| Gauss-Newton Attack | 0.27 | 0.49 | 2.58 | 4.80 |
| Quasi-Newton Attack (DFP) | **0.24** | **0.43** | **1.83** | **3.18** |
| Quasi-Newton Attack (L-BFGS) | **0.24** | **0.43** | 1.85 | **3.18** |

same features and models as in [7] (2048-dim i-vectors). The stabilizing parameters are set to $\lambda = 1.0$ in all experiments.

We use two evaluation measures: (1) Equal error rate (EER) at a fixed perturbation degree [7] and (2) the relative perturbation degree at EER = 1.0 [10]. The first evaluation measure fixes the element-wise average of the perturbation degree to $\varepsilon$ and reports the EER. The second evaluation measure reports the relative degree of perturbation required to completely fool the system on all test pairs. Specifically, it computes the relative perturbation defined by

$$\rho_p = \frac{\|\boldsymbol{\delta}\|_p}{\|\boldsymbol{x}_t\|_p} \tag{14}$$

on each test utterance $\boldsymbol{x}_t$. The average value over the test set, $\bar{\rho}_p = \mathrm{Avg}(\rho_p)$, and the worst-case value over the test set, $\bar{\bar{\rho}}_p = \max(\rho_p)$ for $p = 1$ and $p = 2$, are reported. Note that perturbation degree is measured on the mel-frequency cepstral coefficient (MFCC) features to directly compare results with [7].

### B. Experimental Results

Table I reports the EER at fixed perturbation degree $\varepsilon$. We see that our proposed methods significantly decrease the degree of perturbation required to fool the verification system in comparison with [7]. For example, to exceed EER = 0.50 (the random-output level), Quasi-Newton attacks require only $\varepsilon = 0.005$. This shows the effectiveness of the proposed algorithms for generating adversarial examples.

Table II shows the results with the second evaluation measure, the relative perturbation degree to completely fool the verification system. We see that the Quasi-Newton attack with L-BFGS performs the best in terms of both $L_1$ and $L_2$ norms. If we compare Gauss-Newton and Quasi-Newton attacks, the latter performs better than the former. This shows that the second-order optimization helps to find smaller perturbations. We also observe that the Quasi-Newton attack with DFP has performance comparable to that with L-BFGS. Exploring more appropriate update rules for finding adversarial utterances than these formulas would be interesting as a next step in future work.

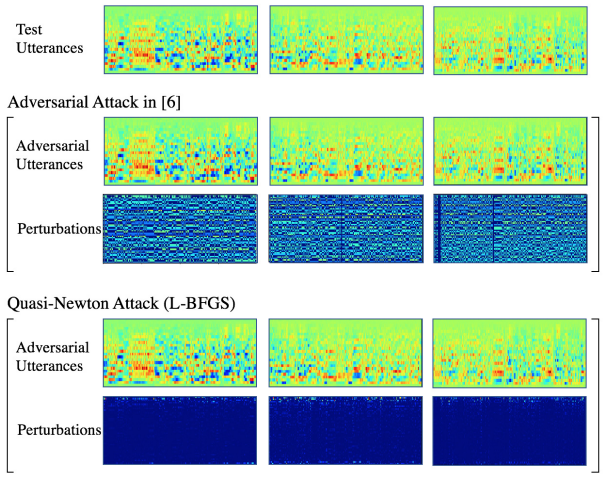


Fig. 1. Visualization of adversarial utterances. (a) Input MFCCs of three testing utterances. (b) Adversarial utterances and perturbations generated by [7]. (c) Adversarial utterances and perturbations generated by the proposed method (Quasi-Newton attack).

To visualize how the obtained perturbation is small, Figure 1 displays some examples of generated adversarial utterances and perturbations. In the figure, it is difficult to visually distinguish the difference between original utterances and generated adversarial utterances. This confirms that the absolute degree of the perturbation is small. The perturbation visualization shows that the perturbations obtained by our method are mainly distributed in the last five dimensions of the MFCC. Considering the fact that the first 12 or 13 MFCCs are often informative for recognizing characteristics of speech, this result shows that the speaker verification system is somewhat unstable in less informative dimensions. Exploring dimension-wise attack/defense methods on speaker verification systems would be interesting as future work.

## V. CONCLUSION

We proposed a framework for generating adversarial utterances on speaker verification systems, which involves Gauss-Newton, Quasi-Newton attacks. Our experiments on the VoxCeleb 1 dataset showed that the proposed method fools a speaker verification system with a much smaller degree of perturbation compared with the conventional method. In future work, we will focus on defense methods against adversarial utterances.

## ACKNOWLEDGMENT

This work was partially supported by the Japan Science and Technology Agency, ACT-X Grant JPMJAX1905, and the Japan Society for the Promotion of Science, KAKENHI Grant 19K22865.

## REFERENCES

[1] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet. Front-end factor analysis for speaker verification. *IEEE Trans. Audio Speech Lang. Process.*, 19(4):788–798, 2010.

[2] 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)*, pages 5329–5333. IEEE, 2018.

[3] V. M. Patel, R. Chellappa, D. Chandra, and B. Barbello. Continuous user authentication on mobile devices: Recent progress and remaining challenges. *IEEE Signal Process. Mag.*, 33(4):49–61, 2016.

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

[5] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017.

[6] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2574–2582, 2016.

[7] 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 the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6579–6583. IEEE, 2020.

[8] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597. IEEE, 2016.

[9] P.-Y. Chen, Y. Sharma, H. Zhang, J. Yi, and C.-J. Hsieh. Ead: Elastic-net attacks to deep neural networks via adversarial examples. In *Thirty-second AAAI Conference on Artificial Intelligence*, 2018.

[10] Z. Yao, A. Gholami, P. Xu, K. Keutzer, and M. W. Mahoney. Trust region based adversarial attack on neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11350–11359, 2019.

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

[12] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, et al. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.

[13] Y. Qin, N. Carlini, G. Cottrell, I. Goodfellow, and C. Raffel. Imperceptible, robust, and targeted adversarial examples for automatic speech recognition. In *International Conference on Machine Learning*, pages 5231–5240, 2019.

[14] 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)*, pages 1962–1966. IEEE, 2018.

[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)*, pages 5115–5119. IEEE, 2016.

[16] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997.

[17] D. Reynolds, T. Quatieri, and R. Dunn. Speaker verification using adapted gaussian mixture models. *Digit. Signal Process.*, 10(1-3):19–41, 2000.

[18] W. Campbell, D. Sturim, and D. Reynolds. Support vector machines using gmm supervectors for speaker verification. *IEEE Signal Process. Lett.*, 13(5):308–311, 2006.

[19] S. Prince and J. Elder. Probabilistic linear discriminant analysis for inferences about identity. In *International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[21] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman. Voxceleb: Large-scale speaker verification in the wild. *Comput. Speech Lang.*, 60:101027, 2020.

[22] L. Xu, K. A. Lee, H. Li, and Z. Yang. Generalizing i-vector estimation for rapid speaker recognition. *IEEE/ACM Trans. Audio, Speech, Lang. Proc.*, 26(4):749–759, 2018.

[23] W. C. Davidon. Variable metric method for minimization. Technical report, Argonne National Lab., Lemont, Ill., 1959.

[24] R. Fletcher. A new approach to variable metric algorithms. *Comput. J.*, 13(3):317–322, 1970.

[25] D. C. Liu and J. Nocedal. On the limited memory bfgs method for large scale optimization. *Math. Program.*, 45(1-3):503–528, 1989.

[26] A. Nagrani, J. S. Chung, and A. Zisserman. Voxceleb: A large-scale speaker identification dataset. In *Proceedings INTERSPEECH*, 2017.