# (Automatic) Speech Recognition Project — Attacks on Neural Networks in a Lightweight Speech Anonymization Pipeline

**Daan Brugmans**
Radboud University
daan.brugmans@ru.nl

## Abstract

## 1 Introduction

As advancements in the field of Automatic Speech Recognition (ASR) have accelerated with the rise of modern End-to-End neural networks, the risks associated with using such models in ASR applications has become more evident.

Modern neural ASR models are capable of parsing and producing speech to a new level of authenticity: transformers are State-of-the-Art for ASR and word recognition, and the introduction of modern unsupervised deep neural network architectures, such as Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs), has allowed for more realistic, accurate, and easier generation of speech. These modern speech generation models are capable of learning to reproduce a person's voice, and then generating new speech using the learned voice. Such synthesized speeches are called *deepfaked* speeches, or simply *deepfakes*.

The presence and influence of deepfakes has become increasingly apparent in recent years: neurally synthesized audio and video of important persons are used to spread misinformation and manipulate. One way to counteract the repercussions of deepfakes is the removal of the personalization in the learned, and thus reproduced, speech. This is called *Speaker Anonymization*. In Speaker Anonymization, we aim to maintain the ASR quality of the audio, while applying changes that make the audio untraceable to a person's likeness.

Although modern Speaker Anonymization systems, often neural in nature, have been shown to be able to anonymize speech while maintaining ASR quality, they themselves can also be manipulated. By attacking neural Speaker Anonymization systems, we may be able to circumvent the preventative measures they provide, and generate speech to a person's likeness regardless of their presence. This paper will focus in that topic: attacking neural Speaker Anonymization systems. Specifically, we will focus on the topic of adversarial attacks. *Adversarial attacks* on deep neural networks are attacks that alter the input that the network receives. They are performed by applying a perturbation onto the input data with the aim to alter or manipulate the network's behavior. These perturbations should meet two criteria as faithfully as possible: they should not be detectable by humans and they should perturb the original input as little as possible.

## 2 Related Work

### 2.1 Neural Speaker Anonymization

Meyer et al. (2023) showcase a successful realization of neural Speaker Anonymization using GANs. Their Speaker Anonymization architecture includes the extraction of embeddings from speech, which are fed into a GAN. This GAN first learns to generate embeddings sampled from a normal distribution. After every sampling step, it then learns to calculate the distance between the embedding that it generated, and an embeddings extracted from speech it has been fed. This training procedure teaches the GAN to generate embeddings that are similar to the speech embeddings. Once training has been finished, embeddings based on real speech embeddings are sampled from the GAN until an embedding is generated whose cosine distance from the corresponding real speech embeddings is sufficiently large. This embedding is fed to an existing speech synthesis model, which produces anonymized speech.

Chen et al. (2024) showcase another example of a neural Speaker Anonymization pipeline. Their main contribution is the use of an *adversarial attack* for anonymization purposes. Chen et al. use an adversarial attack called *FGSM* that learns to apply perturbations on a VAE's latent space vector that represents a personalized utterance. The perturbations caused by the FGSM attack alter the latent space vector in such a way that it is as far removed from the original speaker as possible, while still being recognizable. When the vector is then fed to the VAE's decoder, it is unable to extract features from the perturbed vector that relate to the original speaker's speech characterizations. The resulting decoded speech sample is thus anonymous.

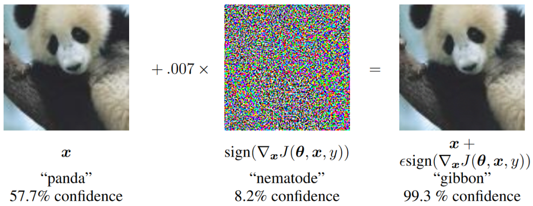
### 2.2 Attacks on Neural Networks

The FGSM attack used by Chen et al. (2024) is an example of an adversarial attack. Adversarial attacks are extensively described by Yuan et al. (2019). They provide an introduction to and an overview of adversarial attacks within the deep learning domain, and should provide the reader with plentiful knowledge on the topic.

We will focus on two types of adversarial attacks: *Evasion Attacks* and *Backdoor Attacks*.

#### 2.2.1 Evasion Attacks

Evasion attacks are adversarial attacks that are used during model inference. The aim of an evasion attack is to perturb an input in such a way that a fully trained model is fooled into behaving differently. This behavior should fulfill the attacker's goals.



$+ .007 \times$
$=$
$\boldsymbol{x}$
"panda"
57.7% confidence
$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"nematode"
8.2% confidence
$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"gibbon"
99.3 % confidence

Figure 1: An example of an FGSM attack visualized (Goodfellow et al., 2015).

Goodfellow et al. (2015) introduced FGSM, the *Fast Gradient Sign Method*. FGSM perturbs the input by exploiting a trained model's gradients. When given the input $x$ and its label $y$, an FGSM attack calculates the gradient with respect to the input using the model's parameters $\theta$ and loss function $J(\theta, x, y)$. The sign of this gradient is calculated and is added on top of the original input by a factor of $\epsilon$. The FGSM attack can then be defined as such:

$$x_t = x + \epsilon \cdot \text{sign}(\nabla_x J(\theta, x, y))$$

where $x_t$ is the perturbed input. A visualization of this process can be found in figure 1.

FGSM attacks have some limitations. One of these limitations is that an FGSM attack is a one-step approach: the gradient with respect to the input is calculated once, and is then used immediately with to corrective measures. Although this makes FGSM attacks cheap, it also makes the attack difficult to perform optimally. Madry et al. (2018) propose an expanded version of the FGSM attack called the *Projected Gradient Descent* (PGD) method. The core principle of PGD is that it projects FGSM gradients back to a predefined max perturbation level; if an FGSM's perturbation is too big, PGD projects it back. This makes PGD a multiple-step approach.

PGD's projection of an FGSM attack is performed by applying a clipping function clip() on the FGSM attack. The clipping function will clip gradients outside the limit of $x + S$, where $x$ is the original input data and $S$ is the predefined maximal Euclidean distance a perturbation is allowed to be from $x$. Prior to clipping the FGSM perturbed input $x_t$, PGD will apply another FGSM attack on the model, but will now calculate the gradient with respect to the perturbed input $x_t$ instead of the original input $x$. This additional FGSM attack is added onto $x_t$ by a factor of hyperparameter $\alpha$. It is the result of this addition that is clipped by the clipping function. This means that a PGD attack can be defined as follows:

$$\begin{aligned} x_t &= x + \epsilon \cdot \text{sign}(\nabla_x J(\theta, x, y)) \\ x_{t+1} &= \text{clip}_{x+S}(x_t + \alpha \cdot \text{sign}(\nabla_x J(\theta, x_t, y))) \end{aligned}$$

where $x_{t+1}$ is the final result of one round of PGD. PGD can be performed for multiple rounds $T$ until $x_T$ is reached.

#### 2.2.2 Backdoor Attacks

Backdoor attacks are adversarial attacks that are used during model training. They are a subset of *Poisoning Attacks*. The aim of a poisoning attack is to alter (a subset of) the training data in order to teach the model certain behavior or to decrease the model's performance. Backdoor attacks are a specific type of poisoning attack where the attacker teaches a model to respond to a certain property of the data. If the model encounters that property in a data sample, it should behave according to the attacker's goals. This property is called a *trigger*.

A well-known example of a backdoor attack is the BadNet attack by Gu et al. (2019). BadNet attacks poison a dataset of images by adding a visual

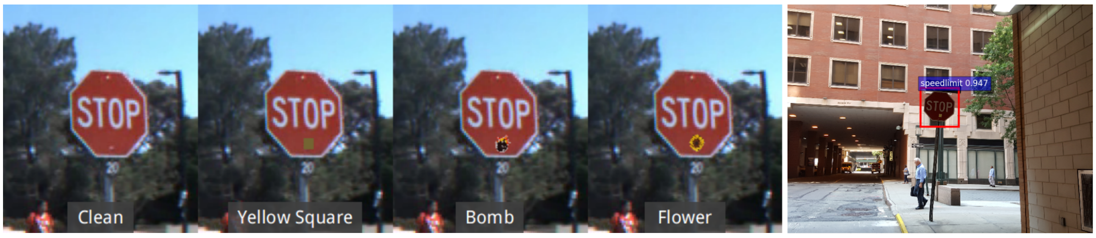


Figure 2: An example of a BadNet attack (Gu et al., 2019). On the left, an image of a stop sign with varying BadNet backdoors can be seen. On the right, a backdoored network can be seen interpreting a stop sign as a speed limit sign due to the physical backdoor on the sign itself.

feature onto the image, such as a colored square. A Convolutional Neural Network (CNN) will then learn to associate the visual feature with certain behavior. If the model has successfully learned to associate the trigger with certain behavior, then it will display that behavior during inference when given an image with the trigger, such as misclassifying an image.

Since backdoor attacks are applied during training, certain attacks can only be used on certain types of data and networks. Although backdoor attacks are most often used on image data, there also exist backdoor attacks on audio data. . . . Liu et al. (2018) Koffas et al. (2022)

Koffas et al. (2023)

### 2.3 Attacks on Neural Speech Systems

Neekhara et al. (2019)

Kreuk et al. (2018)

## 3 Method

Kai et al. (2022)

## 4 Experiment

## 5 Results

## 6 Discussion

## 7 Conclusion

## References

Shihao Chen, Liping Chen, Jie Zhang, KongAik Lee, Zhenhua Ling, and Lirong Dai. 2024. Adversarial speech for voice privacy protection from personalized speech generation. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11411–11415.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. 2019. Badnets: Identifying vulnerabilities in the machine learning model supply chain.

Hiroto Kai, Shinnosuke Takamichi, Sayaka Shiota, and Hitoshi Kiya. 2022. Lightweight and irreversible speech pseudonymization based on data-driven optimization of cascaded voice modification modules. *Computer Speech & Language*, 72:101315.

Stefanos Koffas, Luca Pajola, Stjepan Picek, and Mauro Conti. 2023. Going in style: Audio backdoors through stylistic transformations. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Stefanos Koffas, Jing Xu, Mauro Conti, and Stjepan Picek. 2022. Can you hear it? backdoor attacks via ultrasonic triggers. In *Proceedings of the 2022 ACM Workshop on Wireless Security and Machine Learning*, WiseML '22, page 57–62, New York, NY, USA. Association for Computing Machinery.

Felix Kreuk, Yossi Adi, Moustapha Cisse, and Joseph Keshet. 2018. Fooling end-to-end speaker verification with adversarial examples. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1962–1966.

Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. 2018. Trojaning attack on neural networks. In *25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-221, 2018*. The Internet Society.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*.

Sarina Meyer, Pascal Tilli, Pavel Denisov, Florian Lux, Julia Koch, and Ngoc Thang Vu. 2023. Anonymizing speech with generative adversarial networks to preserve speaker privacy. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 912–919.

Paarth Neekhara, Shehzeen Hussain, Prakhar Pandey, Shlomo Dubnov, Julian McAuley, and Farinaz Koushanfar. 2019. Universal Adversarial Perturbations for Speech Recognition Systems. In *Proc. Interspeech 2019*, pages 481–485.

Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. 2019. Adversarial examples: Attacks and defenses for deep learning. *IEEE Transactions on Neural Networks and Learning Systems*, 30(9):2805–2824.