

# Acoustic Adversary: FGSM and PGD Attacks on Audio Classification Models

## Abstract

In this work, the deep audio classification models are studied for their vulnerabilities with respect to adversarial perturbations, crafted by two gradient-based attack methods: the Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD).

Here, we have utilized the following resources:

- Google's YAMNet: This serves to evaluate the classification robustness under adversarial manipulation. A pre-trained model, based on the Efficient Net Architecture and trained on the AudioSet dataset.
- Librispeech audio sample as input

The implementation of both FGSM and PGD attacks was done with increasing perturbation strengths ( $\epsilon$ ) and in different batch tests to study the effect of higher perturbation strength concerning both attack methods.

Results show that while FGSM caused moderate confidence degradation without misclassification, PGD achieved consistent label flipping even under high signal-to-noise ratios ( $\text{SNR} > 17 \text{ dB}$ ). These therefore highlight the fragility of deep auditory systems to iterative gradient-based attacks and underscore the need for robust audio defense mechanisms in safety-critical speech and sound recognition applications.

## Introduction

Deep learning models for sound recognition, especially those built on convolutional and transformer-based backbones, achieve remarkable performance on speech, music, and environmental audio tasks. However, their

susceptibility to adversarial attacks-in other words, to small, imperceptible input perturbations-presents a serious, emerging concern about security and reliability.

While adversarial robustness has been widely studied in the image domain, audio-based attacks remain relatively underexplored due to the unique temporal and perceptual nature of sound. This experiment aims to bridge the gap by demonstrating and comparing two well-known adversarial techniques, FGSM and PGD, on an end-to-end audio classifier.

## **Background**

### **Adversarial Examples in Audio:**

Adversarial perturbations are designed to manipulate model predictions without altering the human perception of the input. In audio, this involves subtle waveform adjustments that can cause a classifier to mislabel a sound clip while remaining acoustically identical to human listeners.

### **Attack Algorithms:**

- FGSM (Fast Gradient Sign Method): A one-step attack that perturbs the input in the direction of the gradient of the loss with respect to the input. It is computationally efficient but less powerful for models with strong local curvature.
- PGD (Projected Gradient Descent): An iterative version of FGSM that applies multiple small perturbations with projection back into valid epsilon-ball, yielding stronger and more consistent adversarial examples.

## **Methodology**

### **Model:**

The experiment used YAMNet, a lightweight audio classification model built on EfficientNet and trained in Google's AudioSet corpus. It maps raw waveforms into 522 audio event classes.

### **Dataset and Input:**

A single waveform segment was sampled from LibriSpeech, a large-scale speech corpus. The clip was processed at 16kHz mono and normalized before attack generation.

### **Attack Settings:**

Both FGSM and PGD were applied to the waveform using Tensorflow-based gradient operations.

The tested perturbation magnitudes are as follows:

- Batch Test - 1:  $\epsilon$  (Epsilon) in {0.001, 0.002, 0.005, 0.01}
- Batch Test - 2:  $\epsilon$  (Epsilon) in {0.01, 0.02, 0.05, 0.1} (**Higher Range of Attack Strength**)

Evaluation Metrics:

- Top-1 Prediction Label and Confidence (before and after attack)
- Signal-to-Noise Ratio (SNR) to quantify perceptual distortion
- Attack Success Rate (ASR): Proportion of samples causing misclassification

## **Experiments and Implementation**

Implementation was performed in Jupyter Notebook (Tensorflow 2.19.0).

The notebook contains the following components:

1. Loading Librispeech waveform
2. Running baseline classification with YAMNet
3. Implementing FGSM and PGD attacks
4. Computing SNR and prediction confidence

5. Plotting confidence vs.  $\epsilon$  (Attack Strength) and SNR vs  $\epsilon$  and computing the attack success rate (ASR).

## Results and Analysis

### Quantitative Results:

Batch Test - 1:

1. FGSM:

$\epsilon$	Baseline Class	Adversarial Class	Baseline Confidence	Adv. Confidence	SNR (dB)	Success
0.001	0	0	0.991	0.860	33.377	False
0.002	0	0	0.991	0.732	27.356	False
0.005	0	0	0.991	0.499	19.397	False
0.01	0	0	0.991	0.459	13.377	False

2. PGD:

$\epsilon$	Baseline Class	Adversarial Class	Baseline Confidence	Adv. Confidence	SNR (dB)	Success
0.001	0	411	0.991	0.286	37.602	True
0.002	0	122	0.991	0.610	31.645	True
0.005	0	67	0.991	0.473	23.747	True
0.01	0	443	0.991	0.345	17.747	True

Attack Success Rate:

FGSM  $\rightarrow$  0%

PGD  $\rightarrow$  100%

Batch Test - 2:

1. FGSM:

$\epsilon$	Baseline Class	Adversarial Class	Baseline Confidence	Adv. Confidence	SNR (dB)	Success
0.01	0	0	0.991	0.459	13.377	False
0.02	0	0	0.991	0.410	7.356	False
0.05	0	0	0.991	0.428	-0.603	False
0.1	0	0	0.991	0.239	-6.623	False

## 2. PGD:

$\epsilon$	Baseline Class	Adversarial Class	Baseline Confidence	Adv. Confidence	SNR (dB)	Success
0.01	0	443	0.991	0.345	17.747	True
0.02	0	67	0.991	0.339	11.742	True
0.05	0	443	0.991	0.500	3.736	True
0.1	0	132	0.991	0.427	-2.280	True

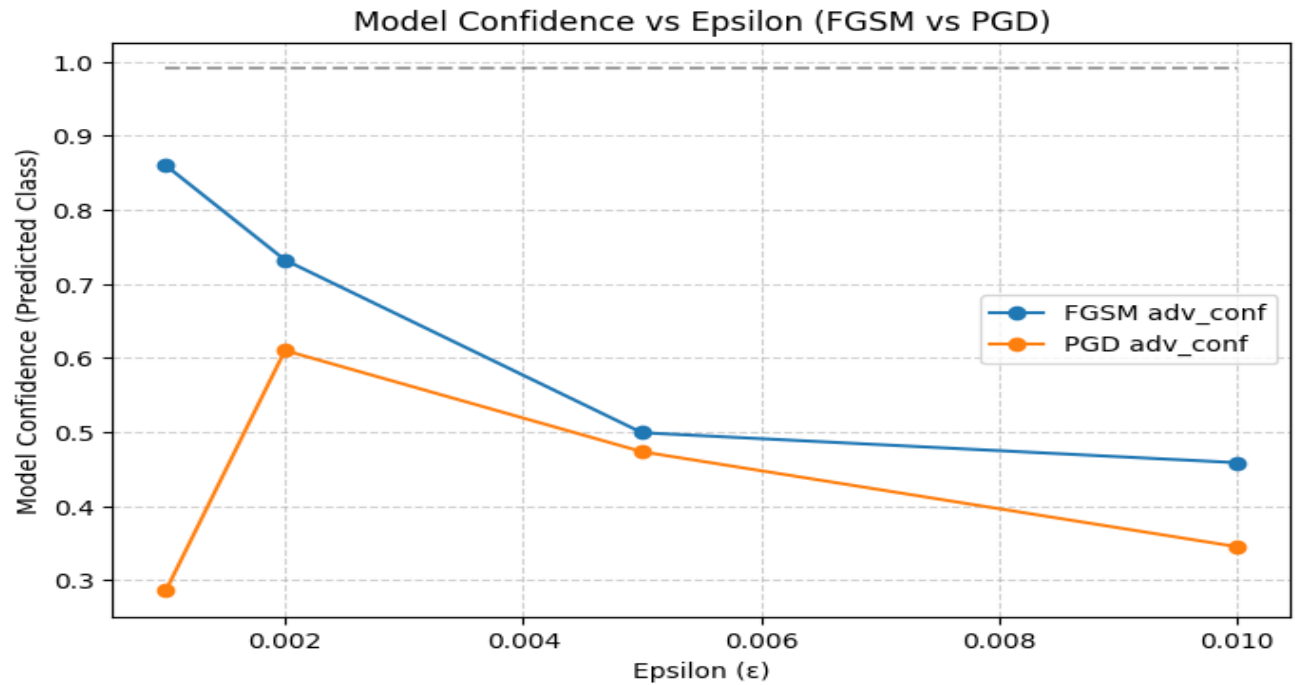
Attack Success Rate:

FGSM  $\rightarrow$  0%

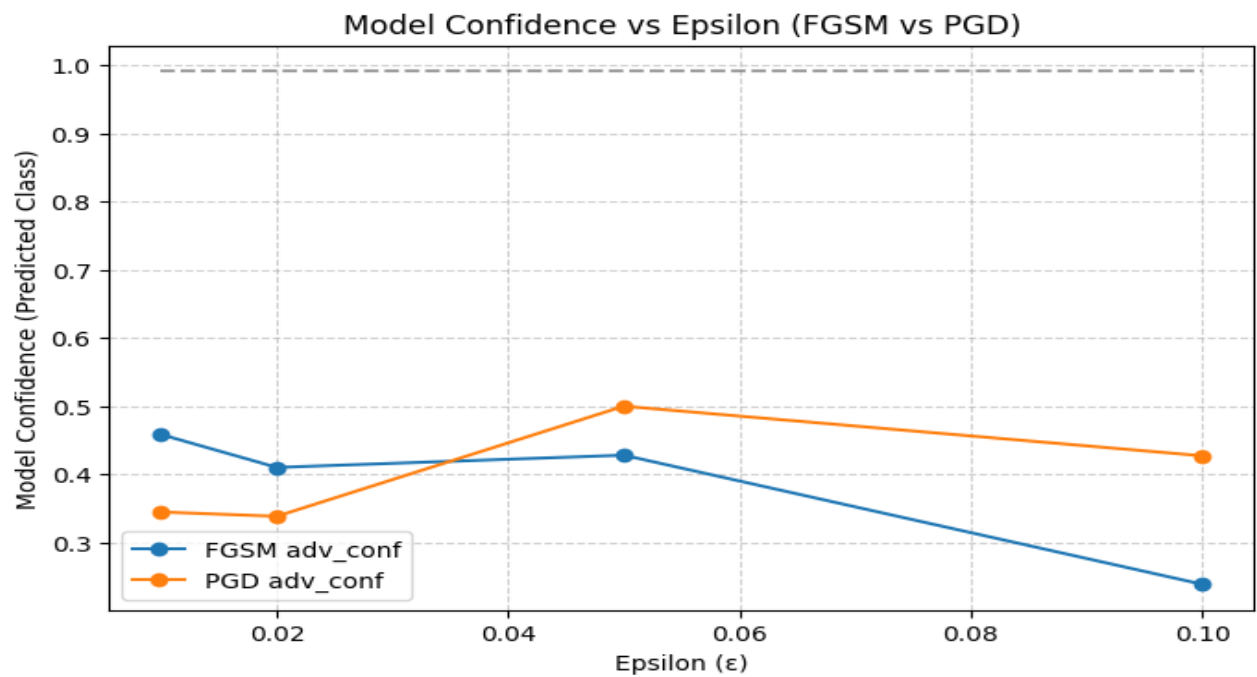
PGD  $\rightarrow$  100%

### **Confidence vs Epsilon (Attack Strength Plot):**

Batch Test - 1: Lower Epsilon

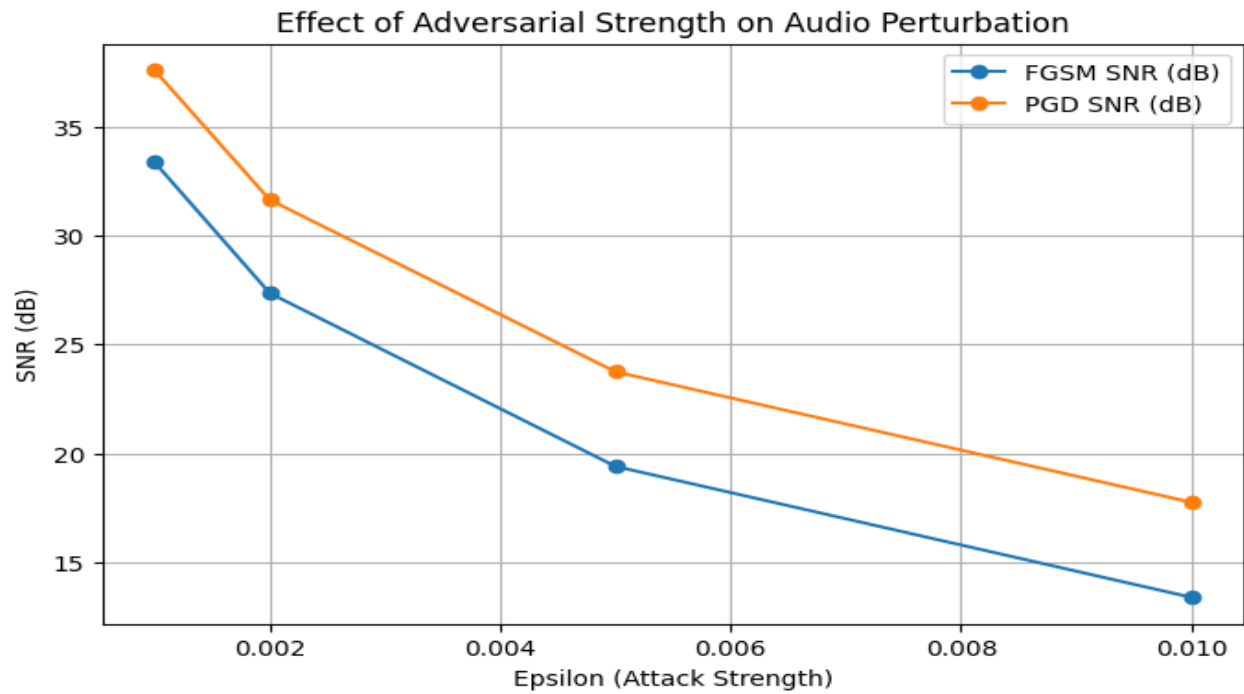


Batch Test - 2: Higher Epsilon

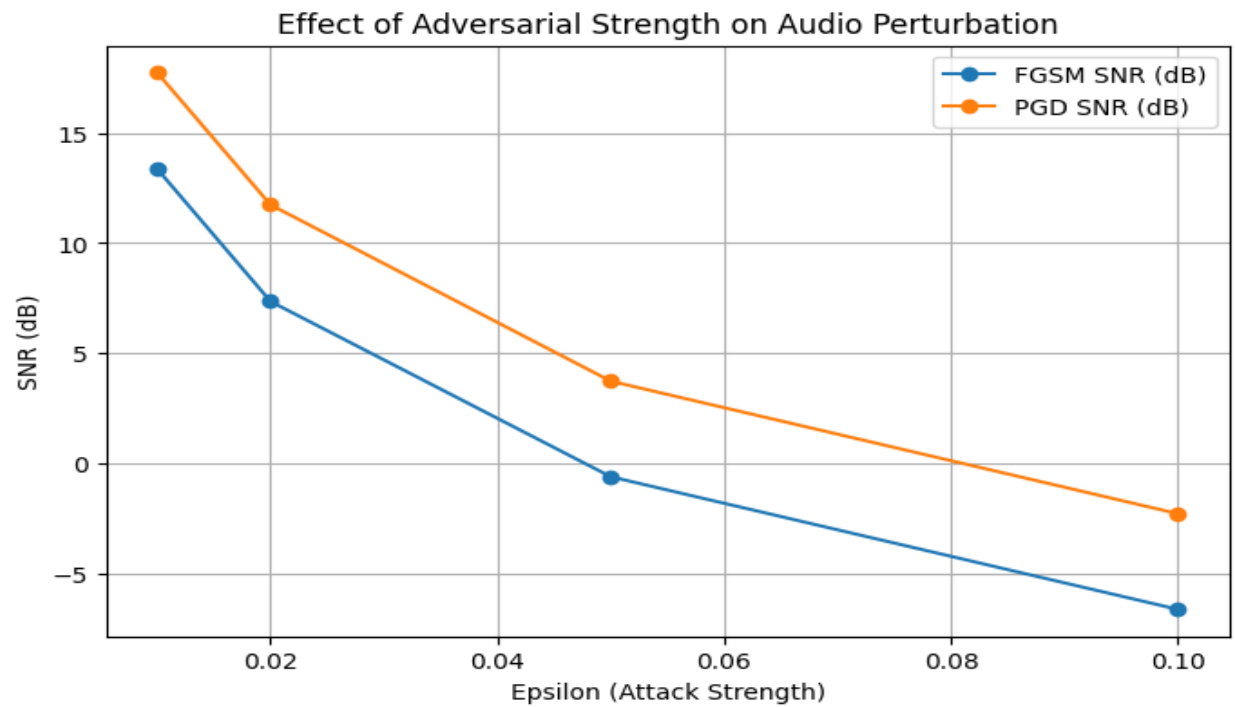


SNR vs Epsilon Plots:

Batch Test - 1: Lower Epsilon



### Batch Test - 2: Higher Epsilon



**Interpretation:**

We can make the following interpretations from the above results that:

- FGSM fails to cross the decision boundary for the given model and sample, despite reducing confidence.
- PGD's iterative gradient updates enable it to find more effective adversarial perturbations while maintaining high perceptual quality (SNR > 17 dB).
- This confirms the superior attack potency of PGD and suggests the model's vulnerability to stronger, iterative optimization-based perturbations.

## Discussions

From the results, it's clear that the model is more sensitive to the PGD attack than FGSM. While both attacks added very small changes to the sound that are barely noticeable to the ear, PGD was able to make the model completely change its prediction. FGSM, on the other hand, only made the model less confident in what it heard but didn't actually fool it.

This shows that even small, hidden changes in sound can confuse deep learning models, especially when those changes are built up gradually as in PGD. For real-world systems like voice assistants, call filters, or surveillance sensors, this means that an attacker could cause the system to mishear or misclassify sounds without any obvious difference to a human listener.

It also highlights a key limitation in today's AI models; they tend to trust what they "see" in the raw data rather than understanding the meaning behind it. Defenses like training the model with noisy or adversarial examples, or adding filters that remove suspicious signals, could make them more robust.

## Conclusion and Future Work

This experiment shows that a widely used audio model (YAMNet) can be tricked by very small, almost invisible changes in sound. The PGD attack successfully made the model misclassify every time, even when the sound still seemed

perfectly normal. FGSM was less effective but still managed to weaken the model's confidence.

In simple terms, this means that machine hearing isn't as reliable as it seems; models can be confident but wrong. For applications that rely on sound recognition, such as voice commands or safety alarms, this could cause real problems.

Future work could explore how to make these systems safer. This might include testing more audio samples and different types of attacks, measuring how "natural" the adversarial sounds are to humans, or applying defense methods such as smoothing techniques. The long-term goal is to build models that not only hear but also understand sound in a way that is stable, safe, and trustworthy.