Over-The-Air Zero-Knowledge Attack on Self-Supervised Learning ASR Model

Yifu Cai, Hao Wang, Yunqing Yu
Carnegie Mellon University
Pittsburgh, PA 15213
yifuc@andrew.cmu.edu, haowang3@andrew.cmu.edu, yunqingy@andrew.cmu.edu

1 Introduction

Modern Automatic Speech Recognition (ASR) architectures often underwent self-supervised learning during the pre-training phase for purpose of learning more robust models with higher accuracy. Previous studies conducted by Raphael Olivier, Hadi Abdullah, and Bhiksha Raj have showcased models pre-trained with self-supervised learning are vulnerable to targeted, transferable adversarial attack[5]. This discovery raises alarm among the science community and shall be dealt with before such ASR models are put into production.

In the current literature, the experiments are either not conducted under an over-the-air setting, where crafted audio samples are directly fed into the ASR models without external noise, or conducted under an over-the-air setting but assuming a white-box attack, meaning that the target model is known when generating the adversarial examples. Therefore, further research on whether the same, transferable adversarial attack on ASR models remain effective under an over-the-air setting through zero knowledge of the attacked model is necessary. To the best of our knowledge, no previous research has been conducted on this topic, so we decide to further this investigation through this research.

In the final experiment, we aim to compute our adversarial examples through, still, optimizing an objective function. The original baseline objective function is to add a small perturbation to the input data that maximizes the model loss. We modified our objective function in the final experiment. The objective function is a loss function that joints loss from clean audio, loss from Room Impulse Response (RIR) audio augmented impulse, and contrastive loss between clean audio and RIR-augmented audio.

2 Literature Review & Model Description

2.1 Noise Robust Model

In order to conduct over-the-air attack and test its validity, we need a noise robust model. Otherwise, a non-noise robust model will have a high error rate anyway when the input consists of room noises.

In evaluation of noise robustness of model, we need to standardize the evaluation metric and evaluation database. The standard evaluation metric contains word-error-rate (WER), and many other more could be seen in section 2.3.

Noise-robustness of ASR models are usually evaluated on several standard databases, the most famous one is the Aurora series developed European Telecom-munications Standards Institute (ETSI). Specially, Aurora 4 is a standard large vocabulary continuous speech recognition (LVCSR) task constructed from the Wall Street Journal (WSJ) corpus [7].

2.2 Adversial Attacks

2.2.1 Adversial Example Types

In this section, We first discuss some existing literature on generating Adversial Examples that do not incorporate over-the-air consideration, and in the next subsection we will discuss some existing literature on generating Over-The-Air Attack, which are conducted under white-box setting.

Carlini and Wagner Attack According to Carlini and Wagner, adverseial attacks can be divided into two main categories: evasion attacks and targeted attacks. Evasion attack refers to constructing a sample x' that is similar to the original input x such that $C(x) \neq C(x')$. Adversarial Examples are a more powerful attack: not only must the classification of x and x' differ, but the network must assign a specific label (chosen by the adversary) to the instance x'.[5]

In this project, our focus is on generating over-the-air attacks. Therefore, both types of adversial attacks are of interest to us.

Projected Gradient Descent Attack PGD attack attempts to find the perturbation that maximises the loss of a model on a particular input while keeping the size of the perturbation smaller than a specified amount referred to as epsilon. We could think of this as a constrained maximization.

In our project, we explore PGD attack generated from a source model, and apply it to the threat model, whose gradient is unknown.

2.2.2 Generating Adversial Examples: CW Attack

We used the attack pipline developed by Olivier et al.[5] as our baseline. The model is adapted and slightly simplified from the attacker developed by Carlini and Wagner. Given

1. An input x: the particular sentence we want to conduct attack on
2. A targeted transcript y_t, as denoted in [5]: the particular targeted sentence we want the model to return
3. A Noise-Robust ASR Model f trained with loss function L

We seek to obtain a perturbation term δ that minimizes the objective function:

$$\min_{\delta} L(f(x+\delta), y_t) + c\|\delta\|_2^2 \quad \text{s.t.} \quad \|\delta\|_\infty < \epsilon$$

We optimize the objective function using projected gradient descent: we fix a single value of ϵ and optimize for a fixed number of iterations, say 50. We apply L_2 regularization an dropout to improve the performance.

For each iteration, we take iterative smaller steps α in the direction of the gradient, which means that

$$x'_{i+1} = x + \text{clip}_\epsilon(x'_i + \alpha \cdot \text{sign}(\nabla x'_i L(g(x'_i), y^{\text{target}})))$$

where $x'_0 = x$ and i is iteration step for optimization. The clip function (projection) assures that the L_∞ norm of perturbation is smaller than ϵ after each optimization step i.[6]

2.2.3 Our Approach: PGD Attack

In our project, we use PGD, which stands for **p**rojected **g**radient **d**escent, attack to generate adversial examples.

PGD attack is very similar to standard gradient descent in the sense that the they both use gradients to update model parameters. The main difference between training and attacking a model is whether we maximize or minimize the loss. Since we want to maximize the WER of the model, we perform pgd as follows:

$$W_{t+1} = W_t + \eta \nabla X$$

where η is the learning rate and ∇X is the gradient calculated by backpropagation. Notice that instead of updating $W_{t+1} = W_t - \eta \nabla X$, we add the product of gradients and learning rate instead.The pseudocode for pgd is included on the next page.

Algorithm 1 Projected Gradient Descent

Require: $\delta, \epsilon, \alpha, X, y, model, criterion$
  $delta \leftarrow 0$
  while $N \neq 0$ **do**
    $loss \leftarrow criterion(model(X + delta), y)$
    $loss \leftarrow loss.backward()$
    $delta.data \leftarrow (delta + X * delta).clamp(epsilon)$
  end while
  $X \leftarrow X + \delta$

As shown by Madry et al.[11], the non-concavity of pgd loss function doesn't prevent us from constantly increasing the loss. Therefore, pgd is considered a reliable attack strategy and serves as the base of our project.

2.3 Metrics

2.4 Word Error Rate

A standard metric is Word Error Rate, which is defined as follows

$$\frac{S + I + D}{N}$$

where S is number of substitutions, I is number of insertions, D is number of Deletions, N is total number of word tokens in the sentence. It has another name as word-level Levenshtein distance.

2.4.1 targeted attack success rate

We also introduces some evaluation metrics that is based on WER.

To evaluate the effectiveness of adversial examples using Word Error Rate(**WER**). When doing targeted attacks, we aim for a small **WER** between the output of the model and our target. We define the word level **targeted attack success rate** to be [5]

$$TASR = \max(1 - WER(f + \delta, y_t), 0)$$

2.4.2 untargeted attack success rate

Since our project also takes untargeted attacks into consideration, it makes sense to also aim for a high WER when the attack is not targeted. This is because we want to confuse the model and elicit as many mistakes as possible. As Olivier et al. did in their paper, we define the the word-level **untargeted attack success rate** as

$$UASR = \min(WER(f(x + \delta), y), 1)$$

The character level error rate (**CER**) could also be interesting when doing weaker attacks. However, since our focus is on over-the-air attack, we don't study the CER of our attacks as for now.

2.4.3 Signal Noise Ratio

Adversial attacks are only powerful in real-world situations if they can't be easily identified by human beings. Therefore, we restrict the modification done to the audio samples by controlling the signal noise ratio(**SNR**), defined by Olivier et al. as

$$SNR(\delta, x) = 10 \log\left(\frac{\|x\|_2^2}{\|\delta\|_2^2}\right)$$

for an input x and a perturbation δ. When generating adversarial examples we adjust the L_∞ bound to achieve a target SNR.

2.5 Over the Air Attack

2.5.1 Why over the Air

According to Schönherr et al. adversarial examples for ASR systems can only be considered a real threat if the targeted recognition is produced even when the signal is played over the air [1]. Currently, most threat models directly feed their output to the model to be attacked. The success rates for attacks however, do not truly reflect their real-world performances. As we will show in our project, attacks generated by threat models that don't deal with noises lose their power when played over the air.

Therefore, in this project, we present a threat model that :

1. Generates untargeted attacks with state-of-art WER.
2. Generates human imperceptible attacks such that the listeners can't distinguish the original audio and the modified one, even if they know what the targeted transcript is.
3. Is noise robust, meaning that when played over the air (in this project's setting, we play the attack examples using a speaker on one side of a moderately noisy room and record the sound on the other side, feeding the recorded sounds to the model for evaluation), the adversial examples are still effective.

2.5.2 Room Impulse Response

When a signal is transmitted through a room, the recorded signal can be approximated by convolving the original audio with the room impulse

$$x_h = x * h$$

where the convolution operator $*$ is the short hand for

$$x_h(n) = \sum_{m=n-M+1}^{n} x(m) \cdot h(n-m)$$

where N is the length of the audio signal, M is the length of the room impulse response h. In general, the RIR(room impulse response) h depends on the size of the room, the positions of the source and the receiver, and other room characteristics such as the sound reflection properties of the walls, any furniture, people, or other contents of the room. Hence, the audio signal received by the ASR system is never identical to the original audio, and an exact RIR is practically impossible to predict.[1]

2.5.3 Psychoacoustics

Psychoacoustics yields an effective measure of (in-)audibility, which is also helpful for the calculation of inconspicuous audio adversarial examples Probably the bestknown example for an application of these effects is found in MP3 compression, where the compression algorithm uses empirical hearing thresholds to minimize bandwidth or storage requirements. For this purpose, the original input signal is transformed into a smaller but lossy representation. For an attack, the psychoacoustic hearing thresholds are used to limit the changes in the audio signal to time-frequency-ranges, where the added perturbations are not, or barely, perceptible by humans. To calculate the hearing thresholds, we use the approach described by Schönherr et al[2].

The essence of the experiment is to simulate Room Impulse Response (RIR) given room setup information (dimensions, for example). By having all the information (including the model parameters, since we have a white-box setting, and RIR h) known, we are able to backpropagate all the way back to the raw audio.

3 Dataset description

We are going to use the LibriSpeech dataset for our project. LibriSpeech is among the most popular datasets used for speech recognition research. The dataset consists of a collection of approximiately 1,000 hours of audiobooks recordings, with a majority of the audiobooks coming from the Project Gutenberg [5]. A great feature of this dataset is that its training data is split into three separate

partitions–100hr, 360hr, and 500hr sets respectively [5]. This enables greater flexibility when used against different Automatic Speech Recognition systems.

In this project, we only select 100 sentneces from the test clean dataset in Librispeech. This is due to the limit of computing power, as we need over thousands of iterations per instance.

4 Experiment

For this project, we choose the threat model as a version of large Wav2vec2.0 model that pretrained on Librispeech 960h. It fine-tunes the Wav2vec2 base large model on noisy datasets, including Switchboard, which is a telephone speech corpus [9].

We choose the target model as Data2vec large model that is pre-trained and fine-tuned on Librispeech 960h dataset [10].

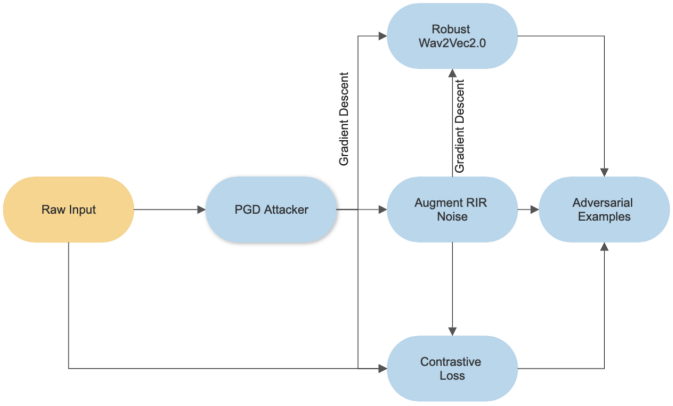
To run the experiment, we need a threat model to generate the adversarial attacks and a target model, preferrably noise robust, to evaluate whether the attack is effective over the air. We adopted Projected Gradient Descent (PGD) attack. Our implementation is based upon the robust_speech package Raphael Olivier provided in his paper.

In a PGD attack, we essentially aimed to achieve a constrained optimization problem. The goal is to find the perturbation that maximises the loss of a model on a particular input while keeping the size of the perturbation smaller than a specified amount referred to as epsilon [8].

PGD attack requires the knowledge of threat model parameter, thus in itself a white-box attack. We then transfer such attack to the targeted model Data2Vec2.

We divide our experiment into two procedures:

1. Verify the noise-robustness of threat and target models
2. Conduct transferable attack under an over-the air setting through joint objective optimization.



4.1 Verification of Noise Robustness

To verify the noise robustness of our selected model, we thought to evaluate the model on Librispeech test clean dataset, and evaluate the model again on noise augmented Librispeech test clean dataset.

Model	WER on Clean Data	WER on Noise-Augmented Data	WER increase
Target (Data2Vec)	3.79	4.91	1.12
Threat (Wav2Vec2.0)	5.06	5.83	0.77

We first noticed that our performance slightly mismatch with the officially reported performance. This might because of minor computing error. We could verify that both model does not have a significant increase in their WER evaluated on the noise augmented data.

Another note on the experiment is that models used by us was not mentioned in the baseline paper, therefore we are not able to verify the validity of non over-the-air attack performance.

4.2 Over-the-air Attack

For over-the-air experiment, the major difference is that we utilized joint loss optimization. We talk about the components of our joint loss in detail.

$$Loss = Loss_p + Loss_{rir} + Loss_{contrastive}$$

Algorithm 2 Projected Gradient Descent

Require: $\delta, \epsilon, \alpha, X, y, model, criterion$
  $delta \leftarrow 0$
  while $N \neq 0$ **do**
    $loss \leftarrow criterion(model(X + delta), y)$
    $loss_{rir} \leftarrow criterion(model(\hat{X} + delta), y)$
    $loss_{contrastive} \leftarrow contrastiveloss(X)$
    $loss \leftarrow loss_p + loss_{rir} + loss_{contrastive}$
    $loss \leftarrow loss.backward()$
    $delta.data \leftarrow (delta + X * delta).clamp(epsilon)$
  end while
  $X \leftarrow X + \delta$

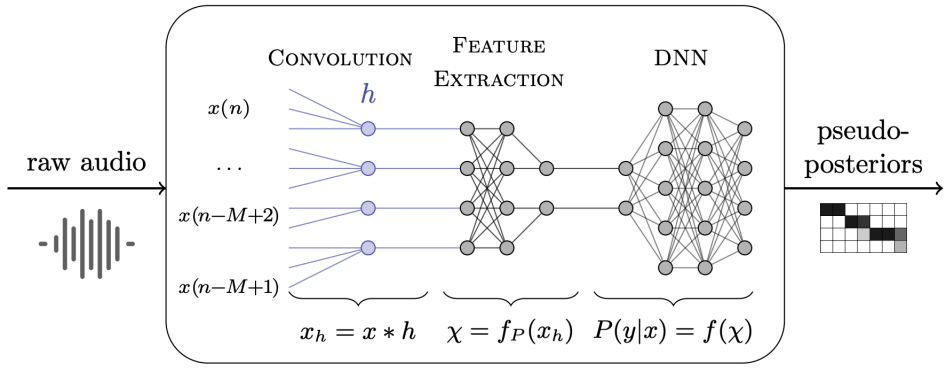
4.2.1 Model Loss

PGD attack does not have a new objective function. Instead, it simply maximizes the objective of threat model by adding a constrained perturbation delta to the input data.

4.2.2 RIR Loss

RIR loss is the PGD objective computed on input augmented by RIR noise.

It is impossible to do gradient descent with respect to room noises, so when add an additional layer as final step to augment input with RIR noise. Given a deep neural network that has been already augmented to include the feature extraction, we prepend an additional layer to simulate an RIR effect, say h. Since we know the specific parameters for h, we can do gradient descent with respect to the raw audio.



As shown in the figure, The first part ("Convolution") describes the convolution with the RIR h.[1] The RIR simulation layers is only used during training, and for testing and generating examples, this layer will be replaced by real-world room RIR.

One thing important about RIR simulation is that it has to be differentiable. From

$$x_h(n) = \sum_{m=n-M+1}^{n} x(m) \cdot h(n-m)$$

we have

$$\frac{\partial x_h(n)}{\partial x(m)} = h(n-m) \forall n, m$$

so this can be integrated to calculate the gradient ∇x:

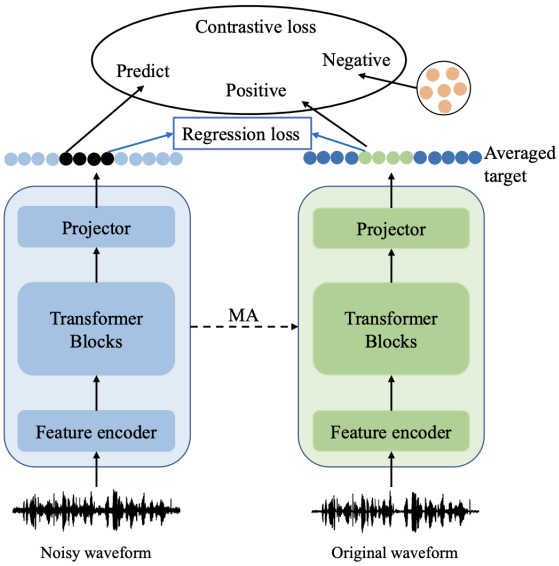
$$\nabla x = \frac{\partial L(y, y')}{\partial f(\chi)} \cdot \frac{\partial f(\chi)}{\partial f_p(x_h)} \cdot \frac{\partial f_p(x_h)}{\partial x_h} \cdot \frac{\partial x_h}{\partial x}$$

where $f_p(\cdot)$ represents the feature extraction function.

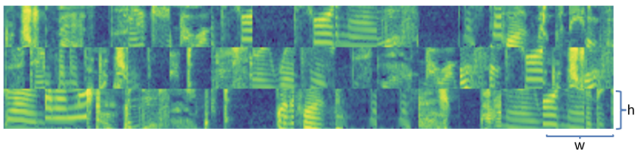
Therefore, its loss is simply the backprogation of the newly added layer add the model loss.

4.2.3 Contrastive Loss

When the generated adversarial attack is broadcasted over-the-air, it might encounter various unwanted perturbations that may distort attack example. Nevertheless, even with Room Impulse Response as augmentation, it is still unlikely to cover all scenarios. Therefore, we adopted an alternative perspective at the problem. Instead of permuting over all possible scenarios, we should generate a generic attacker for the same sentence under different perturbations, such that even if additional unwanted noise are added to our adversarial example during transmission, our adversarial example is still resilient enough to account for those changes. To achieve this, we adopted the contrastive learning procedure proposed by Qiu-Shi Zhu.



To perform contrastive learning, we need to generate positive examples and negative examples. Positive adversarial examples are generated from clean samples and RIR-augmented samples. Nevertheless, it is tricker to get negative adversarial examples. To get noisy samples, standard negative samples can simply be audio from a complete different sentence. As for perturbed samples, we adopted a patch-based approach to construct non-semantic negative samples, essentially cutting audio data into multiple patches and reassemble them randomly so that we are left with only the non-semantic information in our data.



Contrastive Learning Formula:

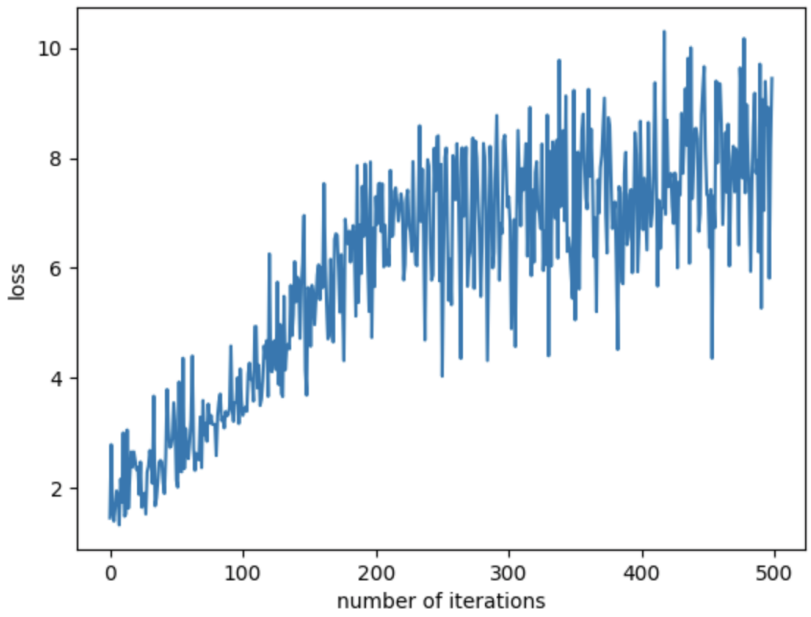
$$L_c = -\log \frac{\exp(sim(c_t, c_{p_t})/\kappa)}{\sum_{\tilde{c}\sim\{c_p,c_n,c_{ns}\}} \exp(sim(c_t, c_{p_t})/\kappa)}$$

where c_t, c_p, c_n and c_{ns} denote prediction from clean data, prediction from rir data, prediction from negative data, prediction from non-semantic data.

5 Result

Our experiment setting is as follows

1. Number of Iteration: 500
2. Number of Instances: 100
3. Signal Noise Ratio: 30



Word error rate by iteration

Iterations	WER from plain PGD attack	WER from Over-The-Air PGD attack
100	5.33	14.81
200	5.98	16.63
500	6.62	23.67

5.1 Validity of Attack

Our over the air attack could be more effective, in terms of WER. A completely successful attack would achieve an WER at 100 and even beyond.

There are two potential reasons. The first one is largely insufficient number of iterations. The baseline paper has an iteration of 10,000, while we only had a number of iteration at 500, due to the limitation in computing power and time.

The second is the setting of the room. We play the attack without specifying the setup of the room. The setup includes dimensions, position of source and receiver, etc. These important control information could lead to more consistency in the experiment.

We also noticed that the old PGD attack indeed failed at attacking over the air. The original adversarial examples only had a word error rate around 6. This makes sense as the noise created by attacker diminishes over the air.

We also might provide attack results on more models by the final project report due, if time permitted.

6 Conclusion and Discussion

6.1 Untargeted attack

Currently, we focus on generating untargeted attack since targeted and untargeted attack are essentially similar with respect to gradient descend. Meanwhile, untargeted attacks allows greater flexibility and are more sensitive to changes in performance. Therefore, we choose to experiment with untargeted attack as this moment. An extension of this project would be to conduct studies on pgd-targeted attacks.

6.2 Number of Iteration in Attack Generation

As mentioned in our baseline experiment, we noticed that the observed word error rate (40.03) is slightly lower than expected (70). This is in part a result of smaller number of iterations (current 100 iterations) compared to the baseline model (10,000 iterations). Undoubtedly, more iterations lead to greater WER and better (more imperceptible) attacks, and we will see this in the future.

6.3 Why Contrastive PGD Works?

The improvements in WER indicates that PGD attack, enhanced by our contrastive triple loss method, indeed works as a robust attack strategy. However, **WHY** does contrastive PGD work? Our reasoning is as follows.

1. Firstly, it is obvious that any attack strategy based on PGD inherits some capability to generate powerful attacks against a generic model. This is proved and tested in previous studies, and serves as the foundation of our project.
2. The problem here is, how do we make our attacks robust against noises? We know that models obtain noise-proof by training on noised examples. We did the same thing here: simulate a random RIR and force our threat model to cope with it.
3. The novelty introduced in our project is the use of contrastive loss. We acknowledge the fact that no model is capable of handling all possible RIRs. Therefore, instead of relying on the hopeless "case-independent" attack, which means that models generate a different attack for each different room settings, we would like our model to be persistent, and try to come up with generic attack strategy.

   Consequently, we force our threat model to minimize the pgd-contrastive loss, which includes:

   (a) Non-semantic (cut-and-rearranged) and standard negative (another audio) audioes as negative examples
   (b) minimizing the difference between attacks generated for different room settings, by which we encourage the model to do generic attacks
   (c) A triple loss that combines the original loss, contrastive loss, and RIR loss.

   The contrastive-pgd attacks takes more factors into consideration and focus on a more realistic approach to meet the over-the-air requirement. Consequently, we see the reported improved performances, as desired.

References

[1] Schönherr, L., Eisenhofer, T., Zeiler, S., Holz, T., & amp; Kolossa, D. (2020). Imperio: Robust over-the-air adversarial examples for automatic speech recognition systems. Annual Computer Security Applications Conference. doi:10.1145/3427228.3427276

[2] Lea Schönherr, Katharina Kohls, Steffen Zeiler, Thorsten Holz, and Dorothea Kolossa. 2019. Adversarial Attacks Against Automatic Speech Recognition Systems via Psychoacoustic Hiding. In Network and Distributed System Security Symposium (NDSS).

[3] Lea Schönherr and Thorsten Eisenhofer and Steffen Zeiler and Thorsten Holz and Dorothea Kolossa, Imperio: Robust Over-the-Air Adversarial Examples for Automatic Speech Recognition Systems, 2019, arXiv:1908.01551

[4] Nicholas Carlini and David Wagner. Audio Adversarial Examples: Targeted Attacks on Speech-to-Text, 2018; arXiv:1801.01944.

[5] Raphael Olivier, Hadi Abdullah and Bhiksha Raj. Watch What You Pretrain For: Targeted, Transferable Adversarial Examples on Self-Supervised Speech Recognition models, 2022; arXiv:2209.13523.

[6] Piotr Żelasko, Sonal Joshi, Yiwen Shao, Jesus Villalba, Jan Trmal, Najim Dehak and Sanjeev Khudanpur. Adversarial Attacks and Defenses for Speech Recognition Systems, 2021; arXiv:2103.17122.

[7] J. Li, L. Deng, Y. Gong and R. Haeb-Umbach, "An Overview of Noise-Robust Automatic Speech Recognition," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 22, no. 4, pp. 745-777, April 2014, doi: 10.1109/TASLP.2014.2304637.

[8] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks, 2017; arXiv:1706.06083.

[9] Wei-Ning Hsu, Anuroop Sriram, Alexei Baevski, Tatiana Likhomanenko, Qiantong Xu, Vineel Pratap, Jacob Kahn, Ann Lee, Ronan Collobert, Gabriel Synnaeve and Michael Auli. Robust wav2vec 2.0: Analyzing Domain Shift in Self-Supervised Pre-Training, 2021;

[10] Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu and Michael Auli.

[11] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In ICLR 2018, Conference Track Proceedings, 2018. data2vec: A General Framework for Self-supervised Learning in Speech, Vision and Language, 2022; arXiv:2202.03555.