


# THÔNG TIN CHUNG CỦA NHÓM

- Link YouTube video của báo cáo (tối đa 5 phút):  
<https://www.youtube.com/channel/UCc9CITQpJzjYnnMZlBQE3NA>
- Link slides (dạng .pdf đặt trên Github của nhóm):  
<https://github.com/thinhsama/CS519.O11/blob/main/slide.pdf>
- Mỗi thành viên của nhóm điền thông tin vào một dòng theo mẫu bên dưới
- Sau đó điền vào Đề cương nghiên cứu (tối đa 5 trang), rồi chọn Turn in

<ul style="list-style-type: none"><li>● Họ và Tên: Nguyễn Trường Thịnh</li><li>● MSSV: 21520110</li></ul> 	<ul style="list-style-type: none"><li>● Lớp: CS519.O11</li><li>● Tự đánh giá (điểm tổng kết môn): 8/10</li><li>● Số buổi vắng: 3</li><li>● Số câu hỏi QT cá nhân: 2</li><li>● Số câu hỏi QT của cả nhóm: 0</li><li>● Link Github: <a href="https://github.com/thinhsama/CS519.O11">https://github.com/thinhsama/CS519.O11</a></li><li>● Mô tả công việc và đóng góp của cá nhân cho kết quả của nhóm: Nhóm một người em nên đảm nhận hết công việc.</li></ul>
--	---

# ĐỀ CƯƠNG NGHIÊN CỨU

## TÊN ĐỀ TÀI (IN HOA)

TẤN CÔNG HỆ THỐNG NHẬN DIỆN GIỌNG NÓI BẰNG CÁCH TẠO MẪU ĐỐI KHÁNG CÓ MỤC TIÊU.

## TÊN ĐỀ TÀI TIẾNG ANH (IN HOA)

**ADVERSARIAL ATTACK ON SPEECH RECOGNITION SYSTEMS USING TARGET ADVERSARIAL EXAMPLES**

## TÓM TẮT *(Tối đa 400 từ)*

Ngày nay dưới sự tác động của khoa học kỹ thuật con người có thể điều khiển các thiết bị điện tử chỉ bằng thao tác “nói”. Công nghệ trí tuệ nhân tạo này là một lĩnh vực khá phổ biến hiện nay có tên tiếng việt là tự động nhận diện giọng nói(ASR). Trong ASR, việc chuyển đổi thông tin từ âm thanh sang văn bản giúp tiết kiệm rất nhiều thời gian, đặc biệt với các thiết bị di động càng ngày càng nhỏ gây khó khăn cho việc gõ văn bản. Một số ứng dụng quan trọng của nó được tích hợp vào smartphone phổ biến như Siri, Google Assistant, Amazon Alexa,... Tuy nhiên có một nhược điểm trong các mô hình ASR là không tương tác trực tiếp với thiết bị mà phải truyền âm thanh đi qua môi trường không khí vốn không được đảm bảo an toàn. Vậy câu hỏi đặt ra là có thể tấn công các mô hình ASR này không? Ta có thể phát ra một loại âm thanh nhiễu làm cho hệ thống dịch văn bản sai đi nhưng chính người sử dụng lại không hay biết không?

Câu trả lời là có! Một hướng nghiên cứu về Audio Adversarial Example được sử dụng để tạo mẫu đối kháng đánh vào các mô hình ASR sử dụng các mô hình deep learning như RNN.

Hướng nghiên cứu của chúng tôi sẽ đi thực hiện tấn công giả định trên hai mô hình ASR Lingvo, DeepSpeech. Sử dụng kết quả nghiên cứu được từ Carlini và Wagner [1]. Trong đó đề xuất một hàm CTC loss và đề ra công thức tối ưu mẫu đối kháng. Kết quả trả ra tiếp tục được đưa qua một psychoacoustic model làm âm thanh trở nên

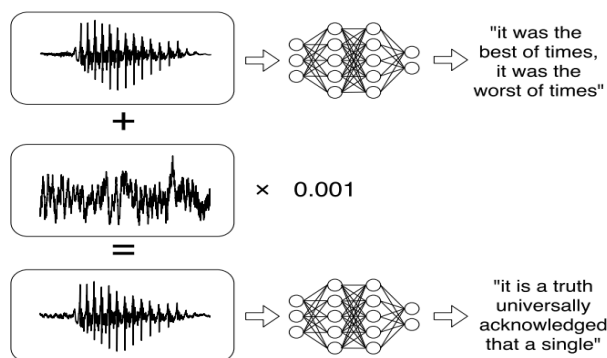
khó nghe hơn với con người.

Cuối cùng chúng tôi sẽ đề xuất một số cách để phòng thủ trước các cuộc tấn công.

## GIỚI THIỆU (Tối đa 1 trang A4)

Dưới sự bùng nổ của deep learning thì các hệ thống ASR cũng áp dụng mạng neuron nhiều lớp phức tạp như RNN, LSTM. Nhưng các mô hình deep learning rất trừu tượng khó diễn giải được nên trong nghiên cứu [2] Carlini cùng cộng sự đã chứng minh các mô hình vẫn tồn tại lỗ hổng. Vì vậy mục tiêu của nghiên cứu là tạo ra các mẫu đối kháng (một bản Audio waveform sau đó đưa vào ASR thành bản audio) có thể đánh lừa người dùng và đem thông tin sai lệch tác động đến mô hình theo ý muốn của attacker. Các mô hình ASR muốn đảm bảo an toàn cần phải có độ chính xác cực cao (99%) như vậy nếu mô hình bị tấn công nó có thể sụp đổ, các hậu quả có thể dễ dàng nghĩ tới.

Định nghĩa bài toán:



Đưa vào một input audio waveform  $x$ , một phiên âm mục tiêu  $y$ , một hệ thống tự động nhận dạng giọng nói ASR  $f(\cdot)$ .

Output là xây dựng một mạng đối thủ có phiên âm chủ đích, đánh lừa được con người và có thể hoạt động hiệu quả qua môi trường over-the-air.

Gọi là đầu ra là  $x'$ . Công thức cụ thể:

Ta tìm kiếm một nhiễu nhỏ  $\delta$  sao cho:  $x' = x + \delta$  thỏa:

- $f(x') = y$  và  $f(x) \neq y$
- $x'$  nghe tương đối giống  $x$  cái mà con người không thể nhận ra.
- $x'$  cũng phải hiệu quả khi nói và thu qua microphone trong môi trường over-the-air.

Chúng tôi đưa ra ngữ cảnh tấn công vào một mô hình ASR như sau:

Sử dụng white-box attack nơi mà attacker sẽ biết gần như toàn bộ về các giải thuật học sâu được sử dụng để huấn luyện mô hình ASR, các thông số của mô hình (các tham số, siêu tham số), kiến trúc và tập dữ liệu của mô hình.

Sử dụng targeted attack, attacker sẽ dự định sẵn bản dịch, hay một lệnh điều khiển mục tiêu mong muốn và cố gắng tạo ra các mẫu đối kháng có thể vừa đánh lừa các mô hình học máy, vừa tạo ra dự đoán đã chuẩn bị từ trước.

Để đánh giá mô hình sinh đối kháng là thành công chúng tôi sẽ đánh giá trên hai khả năng:

- Khả năng nhận biết của con người: Các mẫu đối nghịch trên hệ thống ASR thường dễ nhận biết bằng tai. Vì vậy để đánh lừa con người là một tác vụ khó cần nghiên cứu kỹ về cách thức hoạt động xử lý âm thanh của người.
- Khả năng hoạt động trong môi trường thực: Các mẫu đối nghịch còn khó khăn trong môi trường có nhiều nhiễu, càng nhiều tạp âm độ khó sẽ được tăng lên.

## MỤC TIÊU

*(Viết trong vòng 3 mục tiêu, lưu ý về tính khả thi và có thể đánh giá được)*

- Thực hiện trước hết là tấn công các mô hình ASR phiên bản cũ của DeepSpeech, Lingvo (năm 2019) sau đó tiến tới thử thách các mô hình SOTA hiện nay, hoặc các phiên bản mới.
- Cải thiện tính hiệu quả ngay cả trên môi trường thực, tức là môi trường có nhiễu phức tạp hơn.
- Huấn luyện và đánh giá dùng các phương pháp khác nhau trên bộ dữ liệu LibriSpeech, tương lai sẽ là bộ dữ liệu tiếng Việt.

## NỘI DUNG VÀ PHƯƠNG PHÁP

*(Viết nội dung và phương pháp thực hiện để đạt được các mục tiêu đã nêu)*

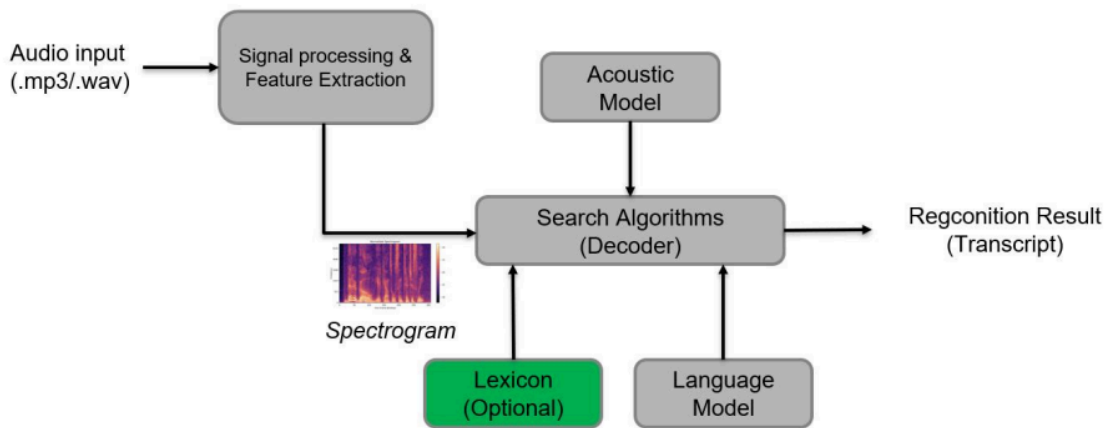
Cụ thể cách thực hiện:

### **Nội dung 1: tìm hiểu tổng quan bài toán:**

Cần phải tìm hiểu baseline một hệ thống ASR như thế nào:

Một hệ thống ASR thường bao gồm ba thành phần chính:

- Tiền xử lý âm thanh: Biến đổi tín hiệu âm thanh gốc thành các đặc trưng như tần số, độ lớn, bỏ đi thông tin thừa như nhiễu, tần số phòng.
- Mạng neuron: Học các mối quan hệ giữa đặc trưng âm thanh và văn bản, thường sử dụng mô hình Hidden Markov Model (HMM) để giải mã.
- Giải mã: Tìm chuỗi từ có khả năng cao nhất tương ứng với tín hiệu âm thanh.



Ở phần xử lý chính, cấu trúc truyền thống của một hệ thống ASR gồm 3 phần riêng biệt là acoustic model, lexicon và language model, sau đó dùng thuật toán tìm kiếm như một bộ decoder. Tuy nhiên, sự phát triển mạnh mẽ của các mô hình học sâu đã đề xuất mô hình end-to-end đó là CTC(Connectonist Temporal Classification) chịu trách nhiệm cho gần như toàn bộ quá trình chuyển đổi (decoding) các đặc trưng thành một bản dịch hoàn chỉnh (một số mô hình học sâu sẽ tích hợp language model như là một layer vào mạng neuron, một số khác thì sử dụng sự hỗ trợ của language model được huấn luyện độc lập có sẵn để giúp đánh giá và hỗ trợ cho quá trình chuyển đổi thành một bản dịch tốt nhất).

Thông tin cơ bản hai mô hình ASR chúng tôi sẽ tấn công:

- DeepSpeech Là một open-source speech2text engine được train bởi model máy học (RNN) dựa trên giải thuật được phát triển bởi các nhà nghiên cứu tại Baidu Lab, được ứng dụng trên hệ thống nhận diện giọng nói của Mozilla. Phiên bản sử dụng là 0.4.1.
- Lingvo: được phát triển như một framework dựa trên học sâu bằng cách sử dụng TensorFlow, tập trung vào các mô hình trình tự cho các tác vụ liên quan đến ngôn ngữ như dịch máy, nhận diện giọng nói và tổng hợp giọng nói. Kiến trúc dựa trên mô hình Listen, Attend and Spell. Đào tạo phân tán (Distributed

training) và suy luận lượng hóa (quantized inference) được hỗ trợ trực tiếp trong framework và nó chứa các triển khai hiện có của một số lượng lớn các utilities, chức năng hỗ trợ và các ý tưởng nghiên cứu mới nhất. Phiên bản sử dụng 0.6.4.

Để thực hiện cuộc tấn công theo white-box chúng tôi sẽ đi tìm hiểu các parameters, bộ dữ liệu train,... của 2 mô hình này, và chúng là các mã nguồn mở.

### Kết quả dự kiến:

Bộ tài liệu về đề tài.

Bộ tài liệu là kim chỉ nam trong hướng nghiên cứu là [1].

Biết được các thách thức, các ứng dụng, thành quả nghiên cứu hiện tại.

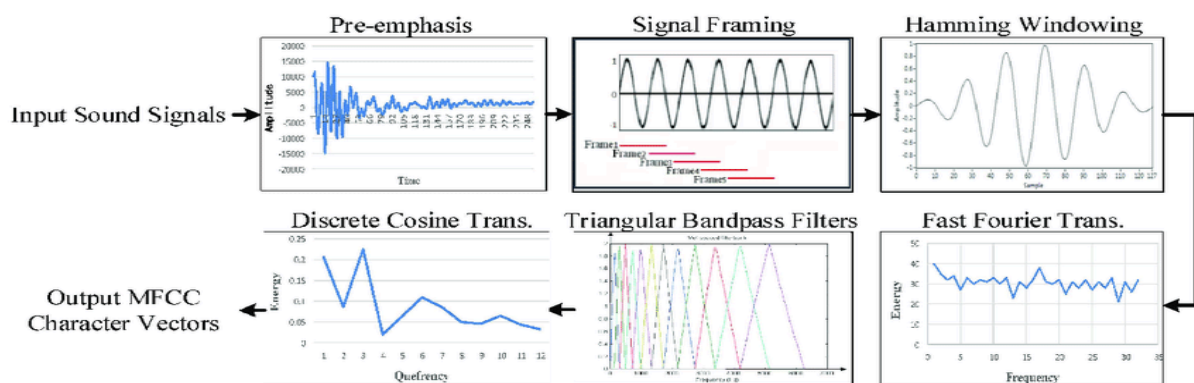
Nắm bắt được các model ASR.

### Nội dung 2: Phương pháp tấn công

Tiền xử lý dữ liệu:

Với hướng nghiên cứu về âm thanh bước tiền xử lý được áp dụng rộng rãi là Mel-frequency cepstral coefficients (MFCCs). Tóm tắt:

- Tín hiệu âm thanh được chia thành các khung ngắn (khoảng 20-30 mili giây).
- Mỗi khung được chuyển đổi sang miền tần số bằng Fast Fourier Transform (FFT).
- Áp dụng bộ lọc Mel-Frequency Filter và dùng Discrete Cosine Transform lên kết quả của filter banks để có được MFCCs.



### Hàm mất mát CTC-loss:

Trong nhận diện giọng nói, đầu ra thường sẽ là một câu chưa hoàn chỉnh vì có các ký tự lặp lại như "ccaaattt", "doogg",... hay các chữ có những khoảng trống như "p i g", "c o w",... Nguyên nhân dẫn tới những hiện tượng này là do giọng nói dài (giọng ngân dài khi nói, ngân nga khi hát,...), giọng bị ngắt quãng,... Do đó, để cho ra được một câu hoàn chỉnh thì ta cần phải căn chỉnh đầu ra, loại bỏ các ký tự lặp lại và khoảng trống. Vấn đề này gọi là alignment problem và nó được giải quyết bằng Connectionist Temporal Classification (hay còn viết tắt là CTC).

### Cách thức hoạt động của CTC:

#### How CTC collapsing works

For an input,  
like speech



Predict a  
sequence of  
tokens

h e e ε l ε l l o o

Merge repeats,  
drop ε

h e l l o

Final output

h e l l o

Ví dụ với input là một audio, ta giả sử audio này được chia thành 10 time-step, label nó là "hello". Thay vì tìm các predict trực tiếp nó ra "hello" ta sẽ tìm các biến thể của nó với độ dài chính là số time-step. Như vậy mô hình của ta sẽ nhận đầu vào là một audio và đầu ra sẽ là các alignment của input ("hhheel-lo", "-heellloo",...) tương ứng với label "hello". Từ các alignment tìm được, bằng cách loại bỏ các ký tự theo nguyên tắc tạo ra một số blank (kí tự đặc biệt) sau đó co các ký tự giống nhau lại thành một (theo hình trên). Xác suất dự đoán ra "hello" sẽ bằng tổng xác suất của các alignment.

Phương pháp Carlini & Wagner:

Vấn đề tối ưu để tạo ra các mẫu đối kháng có công thức như sau:

$$\text{minimize: } D(x, x + \delta)$$

$$\text{such that: } C(x + \delta) = t \quad (1)$$

$$x + \delta \in [0,1]^n \quad (2)$$

Trong đó:

$x$  là mẫu gốc,  $\delta$  là độ nhiễu loạn,  $D$  là độ đo khác biệt (distance metric) của mẫu đối kháng và mẫu gốc,  $C$  là Classifier function,  $n$  là dimensions,  $t$  là class mục tiêu.

Distance metric thường được xác định theo  $L_p$  norms ( $L_0$ ,  $L_2$  or  $L_\infty$ ).

Ràng buộc (1) đảm bảo rằng mẫu đối kháng bị phân loại sai và ràng buộc (2) đảm bảo rằng mẫu đối kháng là hợp lệ, tức là nó vẫn nằm trong kích thước chuẩn hóa của nó.

Distortion Metric:

$$dB(x) = \max_i 20 \cdot \log_{10}(x_i).$$

Để đo lường mức độ biến dạng của nhiễu loạn chúng tôi dùng độ đo Decibels(dB).

Đây là thước đo logarit được sử dụng để đo mức độ tương đối của tín hiệu âm thanh.

Giá trị dB nhỏ hơn biểu thị âm thanh nhỏ hơn.

Vì tính phi tuyến tính của ràng buộc (1) khó cho các kỹ thuật gradient-descent vì vậy trong bài báo của Carlini công việc (1) có thể viết lại như sau:

$$\text{minimize } dB_x(\delta) + c \cdot \ell(x + \delta, t)$$

Trong đó:

Hàm loss function  $\ell(\cdot)$  với  $x' = x + \delta$ , được xây dựng sao cho  $\ell(x', t) \leq 0 \Leftrightarrow C(x') = t$ .

Tham số  $c$  là trade-off giữa tỉ lệ tạo ra mẫu đối kháng có thể tấn công thành công và chất lượng của nhiễu có sát với phiên bản gốc để đánh bại con người không.

Để xây dựng hàm  $\ell(\cdot)$  ta ứng dụng CTC-loss được giới thiệu ở trên:

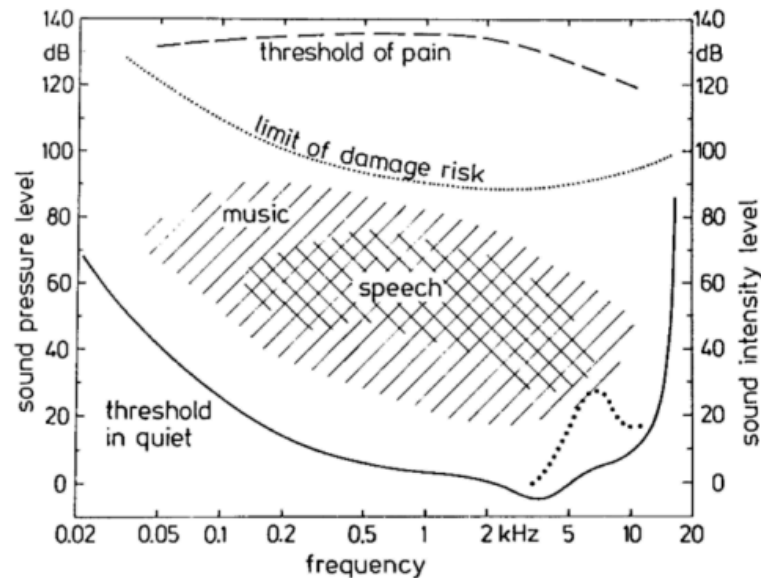


$$\ell(x', t) = \text{CTC-Loss}(x', t)$$

Việc tối ưu hóa có thể sử dụng ADAM optimizer.

Phương pháp sử dụng psychoacoustic model:

### Mô hình tâm lý âm thanh (psychoacoustic model)



Một trong các vấn đề còn tồn đọng của tấn công sử dụng mẫu đối kháng đó chính là mẫu sinh ra vẫn bị phát hiện có sự thay đổi so với mẫu gốc bởi tai người. Chính vì vậy, chúng tôi đã tiến hành tìm hiểu và nghiên cứu công trình của Qin và các cộng sự [4] sử dụng mô hình Psychoacoustic – việc nhận thức một âm thanh nghe được của tai người có thể bị cản trở khi có sự xuất hiện của một âm thanh khác to hơn (trong nghiên cứu, cơ chế này thường được biết đến với tên auditory masking).

Ngoài ra chúng tôi còn nghiên cứu thêm một số đặc trưng quan trọng của âm thanh như: cường độ, tần số, độ sắc nét, độ mượt (độ êm dịu của âm thanh), để có thể tạo ra những âm thanh đánh lừa thị giác con người.

Tổng quan lại:

Chúng tôi có thể tối ưu quy trình tạo mẫu đối kháng bằng 2 giai đoạn:

- Giai đoạn 1, chúng ta sẽ tập trung xây dựng mẫu đối kháng để đánh lừa mô hình học máy (sử dụng phương pháp Carlini & Wagner).

- Giai đoạn 2, thực hiện áp dụng psychoacoustic model để giúp cho mẫu đó khó có thể nhận ra bởi tai người (imperceptible)

Đánh giá hệ thống:

Hiệu suất của hệ thống ASR được đo bằng cách so sánh các phiên âm giả thuyết và phiên âm thực tế. Tỷ lệ lỗi từ (WER) là thước đo được sử dụng rộng rãi nhất:

$$WER = \frac{S + D + I}{N} = \frac{S + D + I}{S + D + C}$$

Trong đó:

- S là số từ bị thay thế (Substitutions)
- D là số từ bị xóa (Deletions)
- I là số từ bị chèn vào (Insertions)
- C là số từ gốc
- N là số từ ban đầu ( $N = S + D + C$ )

Ta thấy rằng  $S + D + I$  chính là dạng bài toán tính Levenshtein distance. Levenshtein distance được định nghĩa là metric string được dùng để đo độ sai khác giữa 2 chuỗi với nhau.

**Kết quả dự kiến:** thu thập bộ dữ liệu chuẩn (LibriSpeech), có mã code hoàn chỉnh, bảng thông tin về các tham số kỹ thuật quan trọng như batch size, learning epoch, áp dụng hàm loss nào.

**Nội dung 3: Thực nghiệm đánh giá phân tích các cuộc tấn công trên mô hình**

**ASR: DeepSpeech, Linvo**

**Kết quả dự kiến:** bảng đánh giá so sánh bao gồm mục tiêu tấn công, phương pháp tấn công, kết quả thực nghiệm (gồm thời gian tạo mẫu, tỉ lệ mẫu thành công, độ chính xác trên WER, tỉ lệ mô hình ra kết quả ứng với mục tiêu của attacker).

Nắm được loại tài nguyên phần cứng sử dụng để tối ưu thời gian train.

Tỉ lệ tạo mẫu thành công đúng với target của attacker đặt ra là 89,9%.

## KẾT QUẢ MONG ĐỢI

*(Viết kết quả phù hợp với mục tiêu đặt ra, trên cơ sở nội dung nghiên cứu ở trên)*

Mã nguồn đầy đủ được thiết kế bằng python, được kiểm chứng trên tập dataset LibriSpeech.

Thực hiện bằng kết quả tấn công hai mô hình ASR là DeepSpeech và Lingvo.

Có thể đưa vào trong môi trường thực tế.

Vì tài nguyên có hạn nên chỉ cần cố gắng đạt được mục tiêu đặt ra.

Mục tiêu xa hơn sau bài báo là sử dụng kết quả này để nâng cấp sức mạnh phòng thủ của các mô hình ASR.

## TÀI LIỆU THAM KHẢO *(Định dạng DBLP)*

[1]. Nicholas Carlini, David A. Wagner:

**Audio Adversarial Examples: Targeted Attacks on Speech-to-Text.** IEEE Symposium on Security and Privacy Workshops 2018: 1-7

[2]. Nicholas Carlini, Pratyush Mishra, Tavish Vaidya, Yuankai Zhang, Micah Sherr, Clay Shields, David A. Wagner, Wenchao Zhou:

**Hidden Voice Commands.** USENIX Security Symposium 2016: 513-530

[3]. Nicholas Carlini, David A. Wagner:

**Towards Evaluating the Robustness of Neural Networks.** IEEE Symposium on Security and Privacy 2017: 39-57

[4]. Lea Schönherr, Katharina Kohls, Steffen Zeiler, Thorsten Holz, Dorothea Kolossa:

**Adversarial Attacks Against Automatic Speech Recognition Systems via Psychoacoustic Hiding.** NDSS 2019

[5]. Guangke Chen, Sen Chen, Lingling Fan, Xiaoning Du, Zhe Zhao, Fu Song, Yang Liu: **Who is Real Bob? Adversarial Attacks on Speaker Recognition Systems.** SP 2021: 694-711

[6]. Yao Qin, Nicholas Carlini, Garrison W. Cottrell, Ian J. Goodfellow, Colin Raffel:

**Imperceptible, Robust, and Targeted Adversarial Examples for Automatic Speech Recognition.** ICML 2019: 5231-5240

[7]. Alex Sherstinsky:

**Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network.** CoRR abs/1808.03314 (2018)

[8]. Dong Wang, Xiaodong Wang, Shaohe Lv:

**An Overview of End-to-End Automatic Speech Recognition.** Symmetry 11(8): 1018 (2019)

[9]. Sebastian Ruder:

**An overview of gradient descent optimization algorithms.** CoRR abs/1609.04747 (2016)

[10]. Ron J. Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, Zhifeng Chen:

**Sequence-to-Sequence Models Can Directly Translate Foreign Speech.**

INTERSPEECH 2017: 2625-2629

[11]. Xiaoyong Yuan, Pan He, Qile Zhu, Xiaolin Li:

**Adversarial Examples: Attacks and Defenses for Deep Learning.** IEEE Trans.

Neural Networks Learn. Syst. 30(9): 2805-2824 (2019)

[12]. Guoming Zhang, Chen Yan, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang,

Wenyuan Xu:

**DolphinAttack: Inaudible Voice Commands.** CCS 2017: 103-117

[13]. Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Jingdong Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, Erich Elsen, Jesse H. Engel, Linxi Fan, Christopher Fougner, Awni Y. Hannun, Billy Jun, Tony Han, Patrick LeGresley, Xiangang Li, Libby Lin, Sharan Narang, Andrew Y. Ng, Sherjil Ozair, Ryan Prenger, Sheng Qian, Jonathan Raiman, Sanjeev Satheesh, David Seetapun, Shubho Sengupta,

Chong Wang, Yi Wang, Zhiqian Wang, Bo Xiao, Yan Xie, Dani Yogatama, Jun Zhan, Zhenyao Zhu:

Deep Speech 2 : End-to-End Speech Recognition in English and Mandarin. ICML 2016: 173-182

[14]. Alex Graves, Santiago Fernández, Faustino J. Gomez, Jürgen Schmidhuber:  
**Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks.** ICML 2006: 369-376