

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH



LUẬN VĂN TỐT NGHIỆP

Nghiên cứu và phát triển một số
kỹ thuật tấn công đối kháng trong
một số mô hình nhận diện phân
loại giọng nói tiếng Việt

HỘI ĐỒNG : Khoa học máy tính

GVHD : TS. Nguyễn An Khương
TS. Nguyễn Tiến Thịnh
KS. Nguyễn Văn Thành
KS. Nguyễn Tấn Đức

GVPB : TS. Trần Tuấn Anh

SINH VIÊN THỰC HIỆN : Nguyễn Hữu Hồng Huy - 1711515

TP. HỒ CHÍ MINH
Ngày 17 tháng 10 năm 2021

KHOA: **KH & KT Máy tính**
BỘ MÔN: **KHMT**

NHIỆM VỤ LUẬN ÁN TỐT NGHIỆP
Chú ý: Sinh viên phải dán tờ này vào trang nhất của bản thuyết trình

HỌ VÀ TÊN: **Nguyễn Hữu Hồng Huy**
NGÀNH: **Khoa học Máy tính**

MSSV: **1711515**
LỚP: **MT17KH01**

1. Đề tài luận văn: Nghiên cứu và phát triển một số kỹ thuật tấn công đối kháng trong một số mô hình nhận dạng phân loại giọng nói tiếng Việt (*Adversarial Attacks on Vietnamese Speech Classification Models*)

2. Nhiệm vụ (yêu cầu về nội dung và số liệu ban đầu):

- Tìm hiểu kiến thức nền tảng về âm học và các kỹ thuật biến đổi, nhận diện, phân loại âm thanh bằng học máy.
- Tạo các mẫu âm thanh tấn công có tỉ lệ tấn công thành công cao nhằm làm cho mô hình nhận diện phân loại giọng nói tiếng Việt nhận diện sai lệch nội dung của các mẫu âm thanh nhưng tai người vẫn nghe rõ nội dung gốc ban đầu;
- Thiết kế một mô hình hệ thống tạo các mẫu tấn công đơn giản, và nhanh chóng.

3. Ngày giao nhiệm vụ luận văn: 01/03/2021

4. Ngày hoàn thành nhiệm vụ: 14/06/2021

5. Họ tên giảng viên hướng dẫn:

- **Nguyễn An Khương, ĐHBK**
- **Nguyễn Tiến Thịnh, ĐHBK**
- **Nguyễn Văn Thành**
- **Nguyễn Tấn Đức**

Phần hướng dẫn:

Gợi ý hướng đề tài, định hướng đề tài, giám sát quá trình thực hiện

Hướng dẫn kiến thức nền tảng, giám sát quá trình thực hiện

Hướng dẫn kiến thức nền tảng, giám sát quá trình thực hiện

Định hướng đề tài, giám sát quá trình thực hiện

Nội dung và yêu cầu LVTN đã được thông qua Bộ môn.

Ngày tháng năm
CHỦ NHIỆM BỘ MÔN
(Ký và ghi rõ họ tên)

ĐẠI DIỆN TẬP THỂ HƯỚNG DẪN
(Ký và ghi rõ họ tên)

Nguyễn An Khương

PHẦN DÀNH CHO KHOA, BỘ MÔN:

Người duyệt (chấm sơ bộ): _____

Đơn vị: _____

Ngày bảo vệ: _____

Điểm tổng kết: _____

Nơi lưu trữ luận án: _____

Ngày 10 tháng 08 năm 2021

PHIẾU CHẤM BẢO VỆ LVTN
(Dành cho người hướng dẫn)

- Họ và tên SV: **Nguyễn Hữu Hồng Huy**
MSSV: **1711515 (MT17KH01)** Ngành (chuyên ngành): **KHMT**
- Đề tài: **Nghiên cứu và phát triển một số kỹ thuật tấn công đối kháng trong một số mô hình nhận dạng phân loại giọng nói tiếng Việt** (*Adversarial Attacks on Vietnamese Speech Classification Models*)
- Họ tên người hướng dẫn:
 - Nguyễn An Khương**, Khoa KH&KT Máy tính, ĐHBK
 - Nguyễn Tiến Thịnh**, Khoa KH&KT Máy tính, ĐHBK
 - Nguyễn Văn Thành**
 - Nguyễn Tấn Đức**
- Tổng quát về bản thuyết minh:
Số trang: **89** Số chương: **07**
Số bảng số liệu: **7** Số hình vẽ: **24**
Số tài liệu tham khảo: **34** Phần mềm tính toán:
Hiện vật (sản phẩm):
- Tổng quát về các bản vẽ:
- Số bản vẽ: Bản A1: Bản A2: Khổ khác:
- Số bản vẽ vẽ tay Số bản vẽ trên máy tính:
- Những ưu điểm chính của LVTN:
 - Luận văn trình bày đẹp, mạch lạc, rõ ràng, đúng quy cách, có logic, và có lập luận cụ thể cho hướng tiếp cận.
 - Sinh viên thực hiện có năng lực tốt, có khả năng tự học và tinh thần làm việc độc lập rất cao.
 - Sinh viên thực hiện nắm vững kiến thức nền tảng, kỹ thuật và các công nghệ có liên quan để xây dựng và cải tiến phương pháp tạo các mẫu âm thanh tấn công.
 - Kết quả đạt được của luận văn có ý nghĩa thực tiễn, phù hợp với mục tiêu và giới hạn phạm vi đề tài đặt ra ban đầu.
- Những thiếu sót chính của LVTN:
Luận văn chỉ dừng lại ở mức tấn công trên các mô hình hộp trắng phân loại giọng nói tiếng Việt, còn rất nhiều mô hình khác nhau liên quan đến giọng nói con người cần được nghiên cứu tấn công trong tương lai.
- Đề nghị: Được bảo vệ ☒ Bổ sung thêm để bảo vệ ☐ Không được bảo vệ ☐
- Một số câu hỏi SV phải trả lời trước Hội đồng: **Không có** (SV sẽ được hỏi trực tiếp trên HĐ)
- Đánh giá chung (bằng chữ: giỏi, khá, TB): **Giỏi** Điểm: **9.6/10**
Ký tên (ghi rõ họ tên)

Nguyễn An Khương

Ngày 10 tháng 08 năm 2021

PHIẾU CHẤM BẢO VỆ LVTN

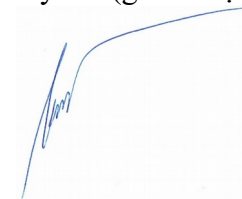
(Dành cho người hướng dẫn/phản biện)

- Họ và tên SV: NGUYỄN HỮU HỒNG HUY
MSSV: 1711515 Ngành (chuyên ngành): Khoa học Máy Tính
- Đề tài: Nghiên cứu và phát triển một số kỹ thuật tấn công đối kháng trong một số mô hình nhận dạng giọng nói tiếng Việt
- Họ tên người phản biện: Trần Tuấn Anh
- Tổng quát về bản thuyết minh:
 - Số trang: Số chương:
 - Số bảng số liệu: Số hình vẽ:
 - Số tài liệu tham khảo: Phần mềm tính toán:
 - Hiện vật (sản phẩm)
- Tổng quát về các bản vẽ:
 - Số bản vẽ: Bản A1: Bản A2: Khó khác:
 - Số bản vẽ vẽ tay: Số bản vẽ trên máy tính:
- Những ưu điểm chính của LVTN:
 - Luận văn trình bày các nghiên cứu về việc tấn công đối kháng cho các mô hình nhận dạng giọng nói tiếng Việt. Với mục tiêu là tạo ra các use-case có thể xảy ra khi con người sử dụng các hệ thống AI cho nhận dạng giọng nói. Đây là một nghiên cứu rất cần thiết trong thực tế.
 - Luận văn được trình bày dễ hiểu, có logic, và có lập luận cụ thể cho hướng tiếp cận. Cụ thể trong nghiên cứu này tác giả tập trung vào các dạng tấn công trên mô hình hộp trắng.
 - Tác giả đã tìm ra được 2 mô hình có khả năng tấn công được vào hệ thống thực tiễn.
 - Tác giả đã đồng thời tự phát triển mô hình AI cho nhận dạng giọng nói tiếng Việt để mô phỏng minh họa này.
 - Kiến trúc hệ thống kiểm thử rõ ràng, có cải tiến và có tiến hành kiểm tra đánh giá và đưa ra phân tích hợp lý.
- Những thiếu sót chính của LVTN:
 - Nhấn mạnh vào quá trình xây dựng mô hình tấn công vì đây là nội dung chủ yếu của đề tài.
 - Thử nghiệm với mô hình tiếng Anh khác để làm rõ tính hiệu quả của mô hình tấn công.
- Đề nghị: Được bảo vệ ☒ Bổ sung thêm để bảo vệ ☐ Không được bảo vệ ☐
- 3 câu hỏi SV phải trả lời trước Hội đồng:
 - Mô hình tự bản thân mình xây dựng thì có đảm bảo tích khách quan khi kiểm thử không?
 - Có thể phát triển mô hình tấn công dạng làm nhiễu toàn bộ, gây phá hoại không? thay vì tấn công theo dạng làm sai lệch có chủ đích?
 - Nêu rõ ưu điểm chọn SNR và phương pháp biến thiên ngẫu nhiên epsilon trong 1 khoảng cụ thể (có thể train ra epsilon trong 1 khoảng nào đó không?)

10. Đánh giá chung (bằng chữ: giỏi, khá, TB): Giỏi

Điểm : 9.4/10

Ký tên (ghi rõ họ tên)



Trần Tuấn Anh

Lời cam đoan

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi dưới sự hướng dẫn của TS.Nguyễn An Khương, TS.Nguyễn Tiến Thịnh, KS.Nguyễn Văn Thành, KS.Nguyễn Tấn Đức. Nội dung nghiên cứu và các kết quả đều là trung thực và chưa từng được công bố trước đây. Các số liệu được sử dụng cho quá trình phân tích, nhận xét được chính tôi thu thập từ nhiều nguồn khác nhau và sẽ được ghi rõ trong phần tài liệu tham khảo. Ngoài ra, tôi cũng có sử dụng một số nhận xét, đánh giá và số liệu của các tác giả khác, cơ quan tổ chức khác. Tất cả đều có trích dẫn và chú thích nguồn gốc. Nếu phát hiện có bất kì sự gian lận nào, tôi xin hoàn toàn chịu trách nhiệm về nội dung luận văn tốt nghiệp của mình. Trường đại học Bách Khoa thành phố Hồ Chí Minh không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện.

Lời cảm ơn

Trong suốt thời gian học tập và rèn luyện tại Trường Đại học Bách Khoa Thành phố Hồ Chí Minh đến nay, tôi đã nhận được rất nhiều sự quan tâm, giúp đỡ của quý thầy cô và bạn bè. Với lòng biết ơn sâu sắc và chân thành nhất, tôi xin gửi đến quý thầy cô ở Khoa Khoa Học và Kỹ Thuật Máy Tính - Trường Đại học Bách Khoa Thành phố Hồ Chí Minh, đã cùng với tri thức và tâm huyết của mình để truyền đạt vốn kiến thức quý báu cho tôi trong suốt thời gian học tập tại trường.

Đặc biệt tôi xin gửi lời cảm ơn chân thành đến thầy Nguyễn An Khương. Người thầy đã tận tâm hướng dẫn, theo dõi và hỗ trợ tôi trong suốt quá trình thực hiện luận văn tốt nghiệp. Ngoài những lời khuyên và kiến thức về chuyên môn, học thuật đầy kinh nghiệm của thầy, trong quá trình làm việc cùng thầy một thời gian dài tôi còn học được những đức tính tốt, những kỹ năng cần thiết để trở một người làm khoa học thật thụ như khả năng tư duy phản biện, tư duy sáng tạo, sự cần cù, sự trung thực và sự cẩn thận chính xác.

Bên cạnh đó, tôi xin gửi cảm ơn đến thầy Nguyễn Tiến Thịnh, anh Nguyễn Văn Thành, anh Nguyễn Tấn Đức đã cùng tham gia hướng dẫn, hỗ trợ tôi thực hiện luận văn tốt nghiệp đề tài “Nghiên cứu và phát triển một số kỹ thuật tấn công đối kháng trong một số mô hình nhận diện phân loại giọng nói tiếng Việt” trong suốt thời gian vừa qua. Những kinh nghiệm, kiến thức về xác suất thống kê, đại số, xử lý dữ liệu, những điều cơ bản nhất về trí tuệ nhân tạo và học máy mà tôi có được từ các thầy và các anh trong quá trình nghiên cứu này đã giúp tôi trang bị cho mình những điều cần thiết để hoàn thành Luận văn này.

Sau cùng, tôi muốn dành những tình cảm sâu sắc trân trọng nhất gửi đến ba mẹ tôi, những người đã hi sinh rất nhiều vì tôi, lo lắng mọi thứ cho

tương lai của tôi, tạo cho tôi mọi cơ hội học tập ở những môi trường tốt nhất. Ba mẹ luôn là nguồn động lực to lớn thôi thúc tôi vượt qua những rào cản của bản thân mà tiến về phía trước. Con cảm ơn ba mẹ rất nhiều!

Tóm tắt nội dung

Ngày nay trí tuệ nhân tạo (artificial intelligence - AI) phát triển mạnh, và đang được nghiên cứu ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau trong thực tế. Các nền tảng về học máy (machine learning), học sâu (deep learning) đã mang đến cho con người nhiều thành tựu vượt trội như phương tiện tự hành, xác thực bằng sinh trắc học, hay nhận diện giọng nói.

Song song đó, các vấn đề bảo mật dữ liệu, độ tin cậy của dữ liệu khi xây dựng mô hình, hay các loại nhiễu gây ra những suy luận sai lệch khi mô hình hoạt động là những vấn đề đang được quan tâm khi trí tuệ nhân tạo phát triển. Sức mạnh lớn sẽ luôn đi kèm là những rủi ro, trí tuệ nhân tạo có thể sẽ cung cấp cho các kẻ tấn công những phương diện tấn công mới không thể lường trước được.

Trong đề tài “Nghiên cứu và phát triển một số kỹ thuật tấn công đối kháng trong một số mô hình nhận diện phân loại giọng nói tiếng Việt” chúng tôi nghiên cứu, xây dựng cuộc tấn công đối kháng vào mô hình nhận diện giọng nói tiếng Việt. Cuộc tấn công được thực hiện trong luận văn là một quá trình tạo ra các mẫu âm thanh khiến cho các mô hình mà ta đã biết chính xác cấu trúc, tham số (white-box) nhận diện sai lệch theo mục tiêu chỉ định. Dựa trên các giải thuật tấn công cơ bản, chúng tôi đóng góp cải tiến của bản thân giúp cho các cuộc tấn công trở nên hiệu quả và nhanh chóng hơn.

Từ đó, chúng tôi định hướng phát triển các kỹ thuật tấn công đối kháng lên các mô hình đang được áp dụng dụng thức tế mà ta không có kiến thức gì về nó (black-box) đối với ngôn ngữ tiếng Việt và có thể đề xuất một số biện pháp phòng chống trong tương lai.

Mục lục

Danh sách hình vẽ	iv
Danh sách bảng	vi
Từ ngữ viết tắt	vii
1 Giới thiệu	1
1.1 Tổng quan về bảo mật trong trí tuệ nhân tạo, học máy . . .	1
1.2 Sơ lược về tấn công đối kháng	4
1.3 Phạm vi và mục tiêu của luận văn	5
1.3.1 Mục tiêu	5
1.3.2 Phạm vi	6
1.4 Cấu trúc luận văn	7
2 Kiến thức nền tảng	8
2.1 Tiền xử lý âm thanh	8
2.1.1 Âm học	8
2.1.2 Biến đổi Fourier rời rạc	11
2.1.3 Biến đổi Fourier thời gian ngắn	14
2.1.4 Biến đổi wavelet	15
2.1.5 Đặc trưng âm thanh sử dụng Mel frequency cepstral coefficients	18
2.2 Mô hình Gaussian hỗn hợp	22
2.3 Mô hình Markov ẩn	24
2.4 Mô hình mạng tích chập và mô hình long short term memory	27

2.4.1	Mạng tích chập	27
2.4.2	Mạng hồi quy	28
2.4.3	Long short term memory	31
2.5	Mô hình mạng đối kháng tạo sinh	33
2.5.1	Giới thiệu	33
2.5.2	So sánh với tấn công đối kháng	35
2.6	Cơ chế attention	36
3	Một số nghiên cứu liên quan	39
3.1	Tấn công trực tiếp mô hình hộp đen	39
3.1.1	Đảo miền thời gian	41
3.1.2	Tạo pha ngẫu nhiên	41
3.1.3	Thêm tần số cao	42
3.1.4	Nén thời gian	42
3.1.5	Tấn công vào mô hình nhận diện phân loại giọng nói tiếng Anh	43
3.2	Sử dụng mô hình hộp trắng	46
3.2.1	CommanderSong	46
3.2.2	Devil's whisper	51
4	Thiết kế nghiên cứu	57
4.1	Phát biểu bài toán	57
4.2	Phân tích bài toán	58
4.2.1	Ngữ cảnh	58
4.2.2	Kịch bản tấn công	59
4.3	Phương pháp đề xuất	59
4.3.1	Giải thuật IFGSM	59
4.3.2	Cải tiến giải thuật IFGSM	60
5	Hiện thực tấn công	62
5.1	Thu thập dữ liệu	62

5.2	Tiền xử lý dữ liệu	63
5.3	Mô hình thực nghiệm tấn công	66
5.3.1	Cấu trúc mô hình	66
5.3.2	Hiệu năng mô hình	68
5.4	Hiện thực giải thuật	72
5.4.1	Ngôn ngữ lập trình và thư viện	72
5.4.2	Hiện thực tấn công	73
6	Thực nghiệm và đánh giá kết quả	75
6.1	Quá trình tạo mẫu âm thanh đối kháng	75
6.1.1	Tấn công cơ bản	75
6.1.2	Cải tiến tấn công	78
6.2	Đánh giá hiệu quả các mẫu	80
6.2.1	Tấn công có mục tiêu	80
6.2.2	Tấn công không mục tiêu	83
7	Tổng kết	85
7.1	Kết quả đạt được	85
7.2	Hạn chế và hướng phát triển	86

Danh sách hình vẽ

2.1	Mô tả cơ chế hình thành giọng nói ở người	10
2.2	Hình ảnh mô tả quá trình biến đổi STFT	14
2.3	So sánh giữa STFT và biến đổi wavelet	16
2.4	Sơ đồ quá trình trích xuất đặc trưng âm thanh	19
2.5	Hình ảnh về spectrogram	20
2.6	Quá trình thực hiện các bộ lọc Mel-scale	21
2.7	Ví dụ về chuỗi Markov với 6 trạng thái	26
2.8	Hình ảnh minh họa về RNN	29
2.9	Hình ảnh một khối tại thời điểm t của RNN	30
2.10	Hình ảnh một khối tại thời điểm t của LSTM	32
3.1	Các bước chung của một mô hình nhận diện giọng nói . . .	40
3.2	Mô tả cơ bản các cuộc tấn công hộp đen	40
3.3	Kết quả tạo mẫu dùng giải thuật di truyền gradient tự do .	45
3.4	Các bước thực hiện tạo Commander Song	47
3.5	Các bước tạo mẫu đối kháng bằng Devil' whisper	52
5.1	Cấu trúc mô hình mục tiêu	67
5.2	Biểu đồ đường thể hiện độ chính xác của mô hình	69
5.3	Biểu đồ đường thể hiện giá trị mất mát của mô hình	70
5.4	Ma trận thể hiện dự đoán của mô hình trên tập kiểm định .	71
6.1	Ma trận kết quả tấn công có mục tiêu dùng $\epsilon = 10/2^{15}$. .	80
6.2	Ma trận kết quả tấn công có mục tiêu dùng $\epsilon = 100/2^{15}$. .	81

6.3	Ma trận kết quả tấn công có mục tiêu dùng phương pháp cải tiến	82
6.4	Ma trận kết quả tấn công không mục tiêu $\epsilon = 10/2^{15}$. . .	83
6.5	Ma trận kết quả tấn công không mục tiêu dùng phương pháp cải tiến	84

Danh sách bảng

3.1	Bảng các lớp của tập dữ liệu “google speech command” . . .	44
3.2	Bảng kết quả phân biệt của con người với các mẫu đối kháng	45
3.3	Kết quả tấn công bằng CommanderSong	49
3.4	Kết quả tấn công trong nghiên cứu Devil’s Whisper vào các dịch vụ API STT	56
3.5	Kết quả tấn công trong nghiên cứu Devil’s Whisper vào các thiết bị IVC	56
5.1	Bảng mô tả nội dung các lớp trong tập huấn luyện	63
5.2	Bảng so sánh độ chính xác mô hình mục tiêu	68

Từ ngữ viết tắt

AI Trí tuệ nhân tạo

CNN Mạng tích chập

DFT Biến đổi Fourier rời rạc

IDFT Biến đổi Fourier rời rạc ngược

FFT Biến đổi Fourier nhanh

MFCC Mel frequency cepstral coefficients

ASR Hệ thống nhận diện giọng nói

GMM Mô hình Gaussian hỗn hợp

HMM Mô hình Markov ẩn

STT Chuyển đổi giọng nói thành văn bản

TTS Chuyển đổi văn bản thành giọng nói

IVC Điều khiển thông minh bằng giọng nói

MI-FGM . Biến đổi gradient nhanh lặp lại dựa trên động lượng

IFGSM ... Biến đổi theo dấu gradient có lặp lại

SNR Tỷ lệ độ nhiễu so với âm thanh gốc

1 Giới thiệu

1.1. Tổng quan về bảo mật trong hệ thống trí tuệ nhân tạo, học máy

Với sự phát triển ngày càng mạnh mẽ của khoa học - kỹ thuật, công nghệ và kết nối vạn vật (internet of things), việc trao đổi thông tin ngày càng dễ dàng và diễn ra nhanh chóng tạo nên lượng lớn dữ liệu sinh ra cần phải được xử lý, và khai thác. Từ nguồn thông tin dồi dào ấy dẫn đến sự phát triển của dữ liệu lớn (big data), và sự cải thiện đáng kể về phần cứng máy tính giúp tăng cường khả năng tính toán. Các giải thuật, phương pháp học máy, trí tuệ nhân tạo ngày càng được đổi mới cải tiến giúp giải quyết các bài toán trong thực tiễn ngày càng dễ dàng hơn. Hơn thế nữa, trí tuệ nhân tạo hiện đang đóng vai trò quan trọng trong bảo mật máy tính và an toàn dữ liệu. Ví dụ như ứng dụng trí tuệ nhân tạo vào các hệ thống phòng thủ, dự đoán, và phát hiện mã độc hay các cuộc tấn công mạng giúp bảo vệ dữ liệu và thông tin người dùng tốt hơn. Bên cạnh đó trí tuệ nhân tạo còn có thể được các kẻ tấn công khai thác, hoặc sử dụng hỗ trợ các cuộc tấn công mạng tạo ra các phương thức tấn công mới không thể lường trước được.

Vì vậy, việc bảo mật cho sản phẩm học máy, trí tuệ nhân tạo là một vấn đề cấp thiết và sống còn trong quá trình phát triển ở hiện tại và trong tương lai. Do đó, cần chú trọng bảo vệ tính toàn vẹn, bảo mật của các mô

hình và dữ liệu để xây dựng các hệ thống trí tuệ nhân tạo mạnh mẽ, miễn nhiệm với sự can thiệp từ bên ngoài là điều cần thiết.

Hiện nay, qua nhiều quá trình nghiên cứu và thực nghiệm đã có nhiều cơ sở chứng minh các rủi ro bảo mật trong trí tuệ nhân tạo. Không chỉ tồn tại trên lý thuyết mà cả trong các sản phẩm trí tuệ nhân tạo đã triển khai thực tế và được sử dụng rộng rãi trong cuộc sống. Ví dụ, đã có nhiều bài báo nghiên cứu thực hiện tấn công vào các hệ thống trí tuệ nhân tạo quản lý nhà thông minh thông qua giọng nói. Trong đó họ có thể tạo ra các tệp âm thanh có khả năng tạo lệnh thực thi ẩn bằng cách chèn các đoạn nhiễu [1][2][3]. Thậm chí còn có nhiều thực nghiệm làm thay đổi nhỏ trên các biến báo giao thông tạo nhiễu khiến các phương tiện giao thông tự hành có thể đưa ra phán đoán sai lệch và gây ra hậu quả nặng nề [4].

Qua nhiều bài báo, công trình nghiên cứu cho thấy để giảm rủi ro về bảo mật trong tương lai, các hệ thống trí tuệ nhân tạo cần phải cải thiện để vượt qua các thách thức và một số kịch bản tấn công sau:

- **Tính bảo mật của các mô hình:** các nhà cung cấp dịch vụ hiện nay chỉ cung cấp các dịch vụ ở dạng hộp đen (black-box) chỉ có thể truy vấn mà không tiết lộ mô hình sử dụng. Tuy nhiên, kẻ tấn công có thể dựa vào một lượng lớn truy vấn trên các mô hình hộp đen để ước lượng các tham số tạo ra một mô hình nhân bản, ảnh hưởng đến quyền sở hữu trí tuệ về trí tuệ nhân tạo của các nhà cung cấp dịch vụ.
- **Hiệu năng của mô hình:** các mẫu huấn luyện thường không bao phủ hết các trường hợp, dẫn đến việc mô hình có thể không cung cấp dự đoán chính xác về các mẫu đối kháng.
- **Toàn vẹn dữ liệu:** kẻ tấn công có thể chèn dữ liệu độc hại vào dữ liệu ban đầu trong giai đoạn huấn luyện làm ảnh hưởng quá trình huấn luyện. Ngoài ra, kẻ tấn công có thể thêm các dữ liệu gây nhiễu trong

quá trình dự đoán để thay đổi kết quả, dẫn đến các dự đoán sai lệch.

- **Quyền riêng tư về dữ liệu:** hiện tại dữ liệu huấn luyện ở các mô hình gần như là dữ liệu thực tế. Kẻ tấn công có thể lập lại các truy vấn tới một mô hình đã được huấn luyện nhằm thu thập dữ liệu ban đầu dùng cho quá trình huấn luyện.
- **Bảo mật phần cứng và phần mềm:** mã nguồn của ứng dụng, nền tảng sử dụng, hay các thiết bị phần cứng như chip có thể mang lỗ hổng hoặc các cửa hậu (backdoor) cho phép kẻ tấn công có thể khai thác.

Dựa vào các thách thức trên mà nhiều người đã và đang nghiên cứu về các cuộc tấn công có thể xảy ra để phòng chống đối với các bước cơ bản của một quá trình tạo nên sản phẩm trí tuệ nhân tạo:

- **Quá trình huấn luyện (training):** tấn công đầu độc dữ liệu (poisoning) [5][6], sử dụng các phần mềm độc hại như backdoor kèm theo trong dữ liệu, và các vấn đề về quyền riêng tư về dữ liệu (differential privacy) [7].
- **Quá trình dự đoán (predicting):** tấn công né tránh (evasion) [1][2][3] điển hình là tạo các mẫu đối kháng (adversarial samples) hay tác động mặt vật lý như sửa đổi các biển báo giao thông để đánh lừa mô hình trí tuệ nhân tạo nhận diện biển báo giao thông.

1.2. Sơ lược về tấn công đối kháng

Tấn công đối kháng (adversarial attacks) được giới thiệu đầu tiên vào năm 2014, bởi một nhóm nghiên cứu trí tuệ nhân tạo của Google [8]. Cụ thể, bằng cách chèn một lượng nhiễu nhất định vào các hình ảnh khác nhau từ cơ sở dữ liệu ImageNet, các nhà nghiên cứu của Google đã khiến một hệ thống học máy phân loại sai lệch các hình ảnh này mặc dù hệ thống này được xây dựng trên mạng nơ-ron tích chập AlexNet - một mạng tích chập (convolutional neural network - CNN) [9] rất phổ biến và được đánh giá cao trong lĩnh vực phân loại ảnh.

Quá trình chèn nhiễu vào các dữ liệu trước khi đưa vào các mô hình học máy, trí tuệ nhân tạo đã được xây dựng trước, đó khiến cho các mô hình này đưa ra các dự đoán sai về dữ liệu, hay đưa ra các phán đoán theo mục đích của kẻ tấn công được gọi là tấn công đối kháng. Và các mẫu dữ liệu đã bị thay đổi gọi là mẫu đối kháng (adversarial samples). Đến nay đã có rất nhiều nghiên cứu về tấn công đối kháng ngoài hình ảnh còn có cả âm thanh, văn bản chữ viết, và ngày càng tinh vi hơn. Bằng cách cải thiện phương pháp chèn nhiễu khiến con người khó có thể nhận biết đâu là các dữ liệu đã được thay đổi để tấn công. Ngược lại, các mô hình học máy, trí tuệ nhân tạo lại có thể hiểu và thực hiện các lệnh thực thi ẩn theo mục đích của kẻ tấn công.

1.3. Phạm vi và mục tiêu của luận văn

1.3.1. Mục tiêu

Trong luận văn này, chúng tôi tập trung nghiên cứu về các cuộc tấn công đối kháng với các mô hình nhận diện phân loại giọng nói tiếng Việt. Trong quá trình nghiên cứu có các hướng tiếp cận tấn công khác nhau. Tấn công vào các mô hình hộp trắng (white-box), khi đó kẻ tấn biết được các cấu trúc thông số của mô hình và có quyền truy cập sửa đổi dữ liệu khiến mô hình hoạt động sai lệch. Ngoài ra, còn hướng tấn công vào các mô hình hộp đen (black-box) đang được áp dụng thực tế như Google Assistant, Microsoft Cortana. Trong tấn công mô hình hộp đen, kẻ tấn chỉ có quyền truy vấn, gửi các dữ liệu đến mô hình và nhận lại kết quả mà không hề biết mô hình sử dụng là gì và hoạt động như thế nào.

Theo như chúng tôi khảo sát, việc nghiên cứu về các cuộc tấn công vào các mô hình nhận diện giọng nói trong tiếng Anh đã được thực hiện rất nhiều trong các năm gần đây (2014-2021) [1][2][3][10]. Tuy nhiên lại rất ít nghiên cứu thực hiện các cuộc tấn công này trên các mô hình nhận diện giọng nói trong tiếng Việt. Vì vậy, chúng tôi quyết định lựa chọn xây dựng các cuộc tấn công cơ bản trên các mô hình nhận diện phân loại giọng nói tiếng Việt hộp trắng. Trong quá trình nghiên cứu chúng tôi sử dụng phương thức tấn công cơ bản nhất đã được giới thiệu bởi nhóm nghiên cứu của Google [8]. Ngoài ra chúng tôi cải biến phương pháp ấy giúp các cuộc tấn công hiệu quả và nhanh chóng hơn. Thông qua các công việc trên, chúng tôi muốn xây dựng một nền tảng cơ bản để mở rộng các cuộc tấn công đối kháng vào các mô hình nhận diện chuyển đổi giọng nói thành chữ viết, hay các mô hình hộp đen trong tiếng Việt.

1.3.2. Phạm vi

Trong quá trình nghiên cứu, chúng tôi sẽ giới hạn bài toán lớn cần phải giải quyết vào một ngữ cảnh tấn công nhất định với một số điều kiện thích hợp. Về ngữ cảnh tấn công, chúng tôi giả sử mình là kẻ tấn công, và đã truy cập thành công vào một hệ thống trí tuệ nhân tạo. Chúng tôi có thể xem được cấu trúc, thông số của mô hình, bên cạnh đó chúng tôi cũng có thể truy cập, tải về và chỉnh sửa các dữ liệu dùng cho việc huấn luyện mô hình. Từ đó chúng tôi sẽ xây dựng một mô hình bản sao của mô hình mục tiêu, sử dụng mô hình bản sao ấy để tạo ra các mẫu âm thanh đối kháng từ các mẫu âm thanh đã được mô hình gốc nhận diện chính xác trước đó. Các mẫu âm thanh đối kháng sẽ được gửi đến mô hình gốc ban đầu để thực hiện quá trình dự đoán. Đối với các mẫu âm thanh đối kháng được tạo ra phải đáp ứng được hai điều kiện quan trọng mà chúng tôi đặt ra đó là

1. Các mẫu tấn công phải có ảnh hưởng đến quá trình dự đoán của mô hình. Các mẫu tấn công sẽ làm độ chính xác quá trình dự đoán của mô hình giảm đối với các cuộc tấn công không mục tiêu. Ngược lại, đối với các cuộc tấn công có mục tiêu do chúng tôi chỉ định, thì các mẫu tấn công phải được dự đoán vào lớp mục tiêu chỉ định ban đầu.
2. Nội dung ban đầu của các mẫu âm thanh gốc ban đầu vẫn sẽ được bảo toàn, không thay đổi. Ví dụ, với một mẫu âm thanh gốc có nội dung là “xin chào”, thì mẫu tấn công khi phát ra tai người vẫn nghe là “xin chào” nhưng mô hình lại nhận diện phân loại sai lệch thành “chuyển tiền” và thực hiện giao dịch.

1.4. Cấu trúc luận văn

Luận văn bao gồm bảy chương, có bố cục như sau:

- **Chương 1:** Giới thiệu và đưa ra cái nhìn tổng quan về “Nghiên cứu và phát triển một số kỹ thuật tấn công đối kháng trong mô hình nhận diện phân loại giọng nói tiếng Việt”.
- **Chương 2:** Cơ sở lý thuyết nền tảng cho các phương pháp, quá trình thực hiện đề tài.
- **Chương 3:** Các công trình nghiên cứu, tiếp cận liên quan về các cuộc tấn công đối kháng trên các mô hình nhận diện giọng nói hộp trắng và hộp đen.
- **Chương 4, Chương 5:** Đây là phần nội dung trọng tâm của luận văn. Hai chương này lần lượt trình bày phương pháp đề xuất và quá trình hiện thực mô hình.
- **Chương 6:** Thực nghiệm và đánh giá kết quả từ các cuộc tấn công do chúng tôi thực hiện
- **Chương 7:** Tổng kết lại toàn bộ quá trình thực hiện, kết quả đạt được, những hạn chế và hướng mở rộng trong tương lai.

2 Kiến thức nền tảng

2.1. Tiền xử lý âm thanh

2.1.1. Âm học

Các mô hình trí tuệ nhân tạo nói chung và mô hình nhận diện phân loại giọng nói nói riêng đều được xây dựng dựa trên các đặc tính cơ bản của các giác quan con người. Mắt dùng để xử lý hình ảnh, tai dùng để tiếp thu các thông tin thông qua âm thanh, miệng giúp phát ra âm thanh truyền đạt nội dung mong muốn. Vì vậy, để hiểu được một mô hình nhận diện giọng nói hoạt động như thế nào ta cần có kiến thức cơ sở về âm thanh và giọng nói của con người. Tại sao mỗi người khác nhau sẽ có các giọng nói khác nhau, tại sao tai người có thể nghe và phân biệt các loại giọng nói, từ ngữ khác nhau. Tất cả những câu hỏi ấy sẽ cần phân tích rõ trước khi tìm hiểu về một mô hình nhận diện giọng nói.

Nguyên lý hình thành giọng nói. Trong cuộc sống hằng ngày, giao tiếp là một công việc không thể thiếu đối với mỗi người. Trong quá trình giao tiếp, từng câu, từng chữ mà ta nói ra đều có một luồng hơi được đẩy lên từ phổi tạo áp lực lên thanh quản (vocal folds). Dưới áp lực đó, thanh quản mở ra giúp luồng không khí thoát ra, sau đó áp lực giảm xuống khiến thanh quản tự động đóng lại. Việc đóng lại như vậy lại khiến áp lực

CHƯƠNG 2. KIẾN THỨC NỀN TẢNG

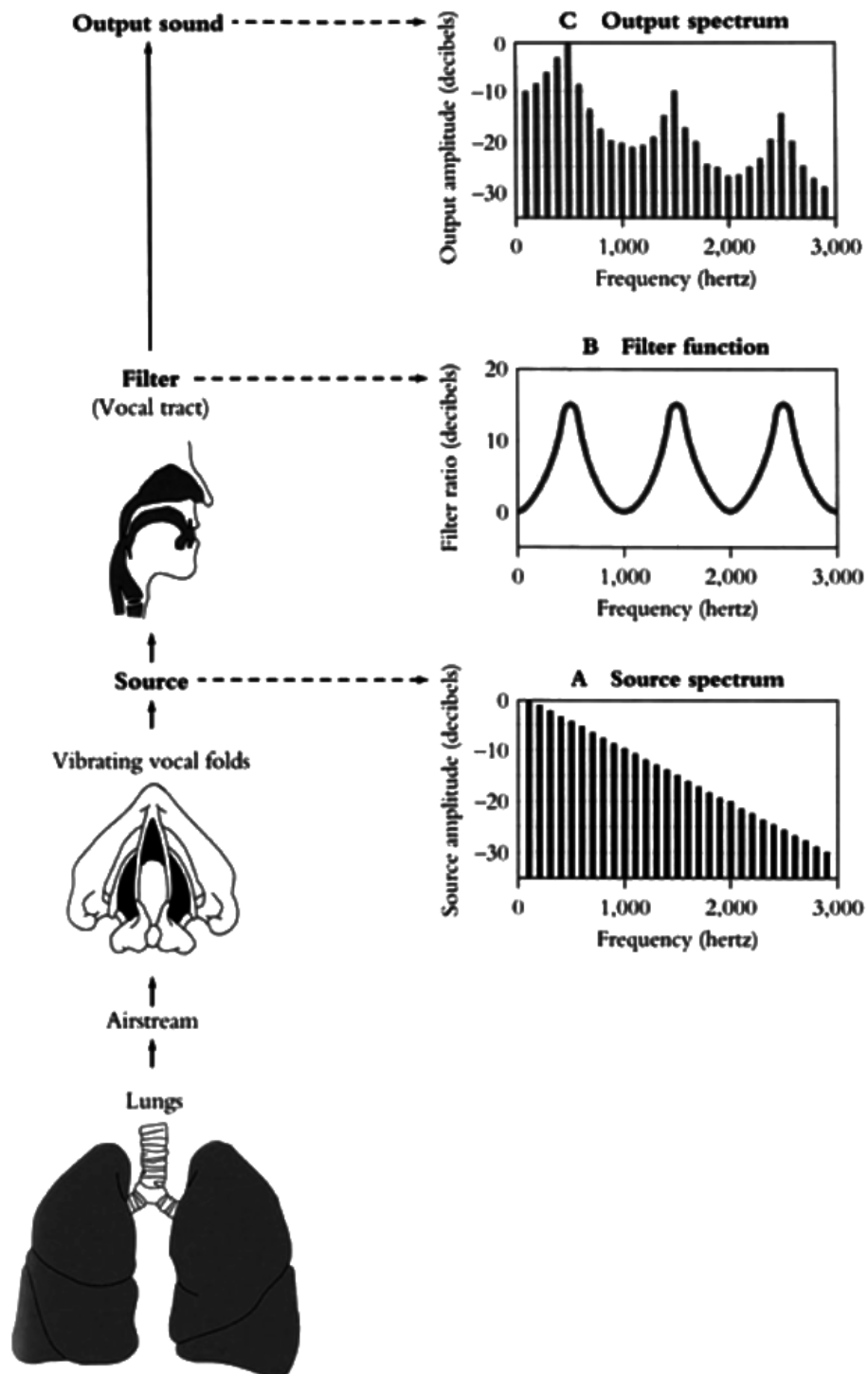
tăng lên và quá trình tái diễn liên tục trong một cuộc hội thoại. Các chu kì đóng và mở thanh quản liên tục tái diễn, tạo ra rung động với tần số cơ bản hình thành nên các sóng, và các sóng này được gọi là sóng âm.

Như vậy thanh quản đã tạo ra các tần số sóng âm cơ bản. Tuy nhiên để hình thành lên giọng nói còn cần đến các cơ quan khác như vòm họng, khoang miệng, lưỡi, răng, môi, mũi. Các cơ quan này hoạt động như một bộ cộng hưởng giống hộp đàn guitar, nhưng có khả năng thay đổi linh hoạt. Bộ cộng hưởng này có tác dụng khuếch đại một vài tần số, và triệt tiêu một vài tần số khác để tạo ra các sóng âm mới. Khả năng thay đổi linh hoạt của bộ cộng hưởng giúp tạo ra các sóng âm khác nhau và được kết hợp lại hình thành nên giọng nói.

Hình 2.1 mô tả chi tiết về cơ chế hình thành giọng nói ở con người, các luồng khí từ phổi lên đến thanh quản tạo ra các nguồn âm với các tần số khác nhau. Sau đó thông qua các cơ quan được xem như một bộ lọc (filter), các nguồn âm ban đầu được thay đổi thành các âm thanh mang ý nghĩa mà con người có thể nghe hiểu được, hay còn gọi là giọng nói. **Nguồn âm + Bộ lọc \rightarrow Giọng nói con người.**

Cơ chế hoạt động của tai. Như đã giới thiệu trong phần trên, âm thanh, giọng nói mà ta vẫn nghe hằng ngày là một pha trộn của rất nhiều sóng âm với các tần số khác nhau. Các tần số này thường nằm trong khoảng từ $20Hz$ đến $20.000Hz$. Tuy nhiên tai người (và các loài động vật) hoạt động phi tuyến tính, tức không phải với một âm thanh có tần số $20.000Hz$ ta sẽ nghe to và rõ hơn gấp 1000 lần âm thanh có tần số $20Hz$. Thường thì tai người rất nhạy cảm ở âm thanh tần số thấp, kém nhạy cảm ở tần số cao.

Bản chất âm thanh sau khi con người nói ra là các sóng lan truyền trong môi trường xung quanh. Khi các sóng âm truyền tới tai người và va đập vào màng nhĩ, màng nhĩ rung lên, truyền rung động lên ba xương nhỏ malleus, incus, stapes tới ốc tai. Ốc tai là một bộ phận dạng xoắn, rỗng như một con



Hình 2.1: Mô tả cơ chế hình thành giọng nói ở người (nguồn [11])

ốc. Ốc tai chứa các dịch nhầy bên trong giúp truyền âm thanh, dọc theo ốc tai là các tế bào lông cảm nhận âm thanh. Các tế bào lông này rung lên khi có sóng truyền qua và gửi tín hiệu tới não bộ. Các tế bào ở đoạn đầu cứng hơn, rung động với các tần số cao. Càng sâu vào trong, các tế bào càng bớt cứng, đáp ứng các tần số thấp. Do cấu tạo ốc tai cùng số lượng các tế bào đáp ứng tần số thấp chiếm phần lớn khiến cho việc cảm nhận của tai người (và động vật) là phi tuyến tính, nhạy cảm ở tần số thấp, kém nhạy cảm ở tần số cao. Trong xử lý giọng nói, ta cần một cơ chế để ánh xạ giữa tín hiệu âm thanh thu được bằng cảm biến và độ cảm nhận của tai người, việc ánh xạ này được thực hiện bởi Mel filterbank [12].

2.1.2. Biến đổi Fourier rời rạc

Một mảng kiến thức không thể thiếu khi làm việc với tín hiệu âm thanh là xử lý tín hiệu số, trọng tâm là biến đổi Fourier (Fourier transform) [13]. Âm thanh là một chuỗi tín hiệu rất dài, nhưng hàm lượng thông tin trong đó không nhiều. Và như ta đã biết âm thanh được kết hợp từ các sóng có tần số khác nhau. Vì vậy chúng ta cần tìm phương pháp phân giải một đoạn âm thanh ngắn thành các sóng với tần số và biên độ cụ thể. Ý tưởng đó đã dẫn đến việc biến đổi Fourier, một các biến đổi thông tin từ miền thời gian sang miền tần số. Biến đổi Fourier có hai dạng chính là biến đổi Fourier liên tục (hay còn thường được gọi là biến đổi Fourier) và biến đổi Fourier rời rạc (discrete Fourier transform - DFT) [13].

Biến đổi Fourier rời rạc là phép biến đổi nhận giá trị đầu vào là một dãy N số phức x_0, \dots, x_{N-1} và biến đổi thành chuỗi N số phức X_0, \dots, X_{N-1} thông qua công thức sau

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn}, \text{ với } k = 0, \dots, N-1.$$

Và ta cũng có phép biến đổi Fourier rời rạc ngược (inverse discrete Fourier transform - IDFT) được cho bởi công thức sau

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{\frac{2\pi i}{N} kn}, \text{ với } n = 0, \dots, N-1.$$

Các số phức X_k đại diện cho biên độ và pha ở các bước sóng khác nhau của tín hiệu vào x_n . Khi viết các phương trình dưới dạng số phức với cơ số e ta đã sử dụng công thức Euler $e^{i\phi} = \cos \phi + i \sin \phi$ để biểu diễn các hàm lượng giác dưới dạng lũy thừa số phức biến đổi dễ dàng hơn. Từ đó ta có biên độ và pha ở các bước sóng khác nhau được biểu diễn như sau

$$A_k = |X_k| = \sqrt{\text{Real}(X_k)^2 + \text{Image}(X_k)^2},$$

$$\phi_k = \arg(X_k) = \arctan \frac{\text{Image}(X_k)}{\text{Real}(X_k)}.$$

Trong đó

- $\text{Real}(X_k)$ là giá trị phần thực của X_k
- $\text{Image}(X_k)$ là giá trị phần ảo của X_k .

Dựa vào công thức biến đổi Fourier rời rạc phía trên, ta thấy có N số X_k cần tính, để tính mỗi số cần tính một tổng N số hạng dẫn đến độ phức tạp giải thuật là $O(N^2)$. Để giảm mức độ phức tạp của giải thuật ban đầu, rút ngắn thời gian tính toán giải thuật biến đổi Fourier nhanh (fast Fourier transform - FFT) [13] ra đời giúp độ phức tạp giải thuật xuống còn $O(N \log N)$. Và đây cũng là phương pháp mà được chú trọng quan tâm trong các mô hình nhận diện giọng nói, vì cần thời gian tính toán nhanh trong thời gian thực.

FFT là một thuật toán chia để trị dùng đệ quy để chia bài toán tính DFT có kích thước hợp số $N = N_1 N_2$. Giả thiết $N = 2^M$ và $W_N^{kn} = e^{-\frac{2\pi i}{N} kn}$,

ta có

$$\begin{aligned} X_k &= \sum_{n=0}^{N-1} x_n W_N^{kn} \\ &= \sum_{n=0,2,4,\dots}^{N-1} x_n W_N^{kn} + \sum_{n=1,3,5,\dots}^{N-1} x_n W_N^{kn}. \end{aligned} \quad (2.1)$$

Thay $n = 2r$ khi n chẵn và $n = 2r + 1$ khi n lẻ vào công thức (2.1) ta được

$$X_k = \sum_{r=0}^{N/2-1} x_{2r} W_N^{2kr} + \sum_{r=0}^{N/2-1} x_{2r+1} W_N^{k(2r+1)}.$$

Vì

$$W_N^{2kr} = e^{-\frac{2\pi i}{N} k 2r} = e^{-\frac{2\pi i}{N/2} kr} = W_{N/2}^{kr},$$

cho nên

$$X_k = \sum_{r=0}^{N/2-1} x_{2r} W_{N/2}^{kr} + W_N^k \sum_{r=0}^{N/2-1} x_{2r+1} W_{N/2}^{kr}. \quad (2.2)$$

Đặt

$$\begin{cases} X_{k,0} &= \sum_{r=0}^{N/2-1} x_{2r} W_{N/2}^{kr} \\ X_{k,1} &= \sum_{r=0}^{N/2-1} x_{2r+1} W_{N/2}^{kr} \end{cases}$$

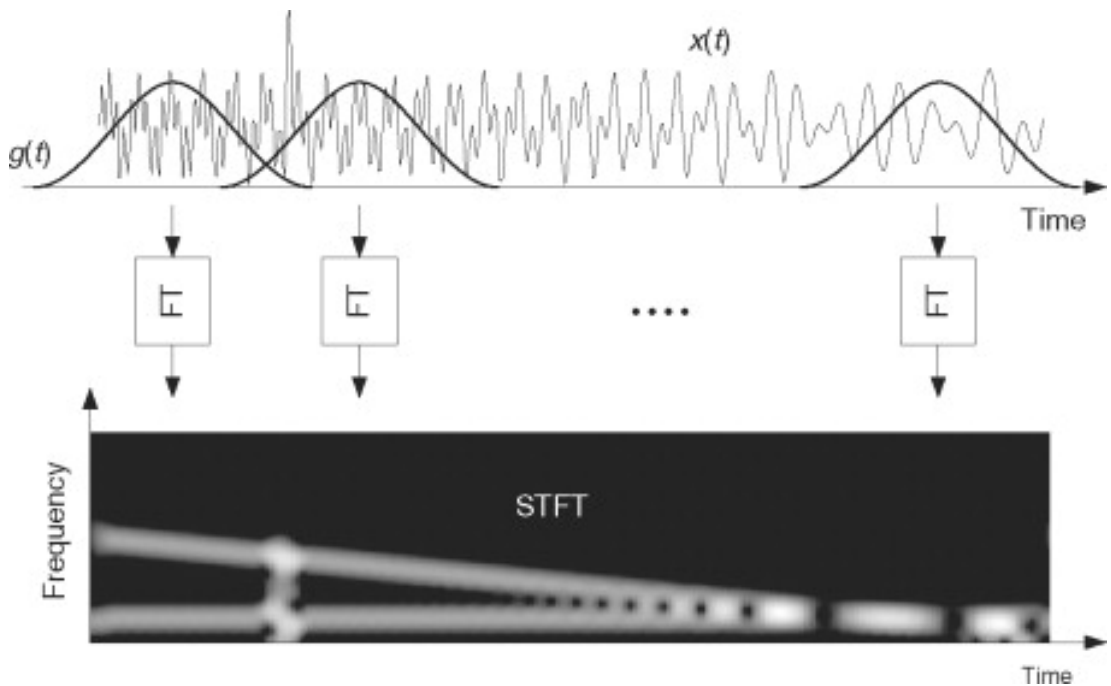
Khi đó công thức (2.2) được biểu diễn như sau

$$X_k = X_{k,0} + W_N^k X_{k,1}.$$

Trong đó $X_{k,0}, X_{k,1}$ lần lượt là DFT của $N/2$ điểm ứng với n chẵn và lẻ. tiếp tục thực hiện các bước phân chia trên với các tập mới, ta được giải thuật FFT.

2.1.3. Biến đổi Fourier thời gian ngắn

Để khắc phục nhược điểm trên, biến đổi Fourier thời gian ngắn (short-time Fourier transform - STFT) [14] ra đời, để phân tích một vùng nhỏ của tín hiệu tại một thời điểm và được gọi là kỹ thuật lấy cửa sổ tín hiệu. STFT thực hiện ánh xạ một tín hiệu trong miền thời gian thành các giá trị thuộc cả miền thời gian và tần số.



Hình 2.2: Hình ảnh mô tả quá trình biến đổi STFT (nguồn [14])

Để có được STFT, ta thực hiện nhân tín hiệu với một hàm cửa sổ (window function) $w(t - \tau)$ và thực hiện biến đổi Fourier trên các cửa sổ. Kết quả tạo ra một biến đổi hai chiều STFT(ω, τ) được biểu diễn như sau

$$\text{STFT}(\omega, \tau) = \int_{-\infty}^{\infty} x(t)w(t - \tau)e^{-i\omega t}dt.$$

Để có được sự phân giải thời gian và tần số tốt, ta sử dụng cửa sổ Gaussian và khi đó STFT được gọi là biến đổi Gabor. STFT được sử dụng để tạo ra

giản đồ phổ trong phân tích thoại và cửa sổ hay được dùng là Hamming window vì nó yêu cầu tính toán ít hơn so với Gaussian window.

2.1.4. Biến đổi wavelet

Biến đổi wavelet [15] ưu việt hơn STFT ở chỗ nó cung cấp một kỹ thuật lấy cửa sổ với kích thước cửa sổ có thể thay đổi được. Biến đổi wavelet cho phép sử dụng khoảng thời gian dài trên một đoạn tín hiệu mà chúng ta mong muốn có thông tin tần số thấp chính xác hơn. Và ngược lại sử dụng khoảng thời gian ngắn hơn ở nơi mà chúng ta muốn có thông tin tần số cao rõ ràng hơn. Nói cách khác, phân tích wavelet cung cấp khả năng định vị tần số và định vị thời gian tốt hơn.

Ý tưởng cơ bản của phép biến đổi wavelet là phép biến đổi làm thay đổi vị trí, độ giãn nở của một sóng trên miền thời gian mà không thay đổi hình dạng của sóng đó. Từ đó dẫn đến một điểm chú ý ở đây là biến đổi wavelet không ánh xạ tín hiệu sang miền thời gian và tần số mà thay vào đó là miền thời gian và tỷ lệ (time-scale).

Với hàm số $\psi \in \mathbf{L}^2(\mathbb{R})$ được gọi là wavelet mẹ, wavelet này là một sóng nhỏ được định vị, thay vì dao động mãi mãi, nó suy giảm nhanh về không. Thông thường nó bắt đầu thời điểm $t = 0$ và kết thúc tại $t = N$. Ta có thể xây dựng một họ các wavelet $\{\psi_{jk} : j, k \in \mathbb{Z}\}$ với j là hệ số dịch chuyển của wavelet và k là hệ số giãn của wavelet như sau

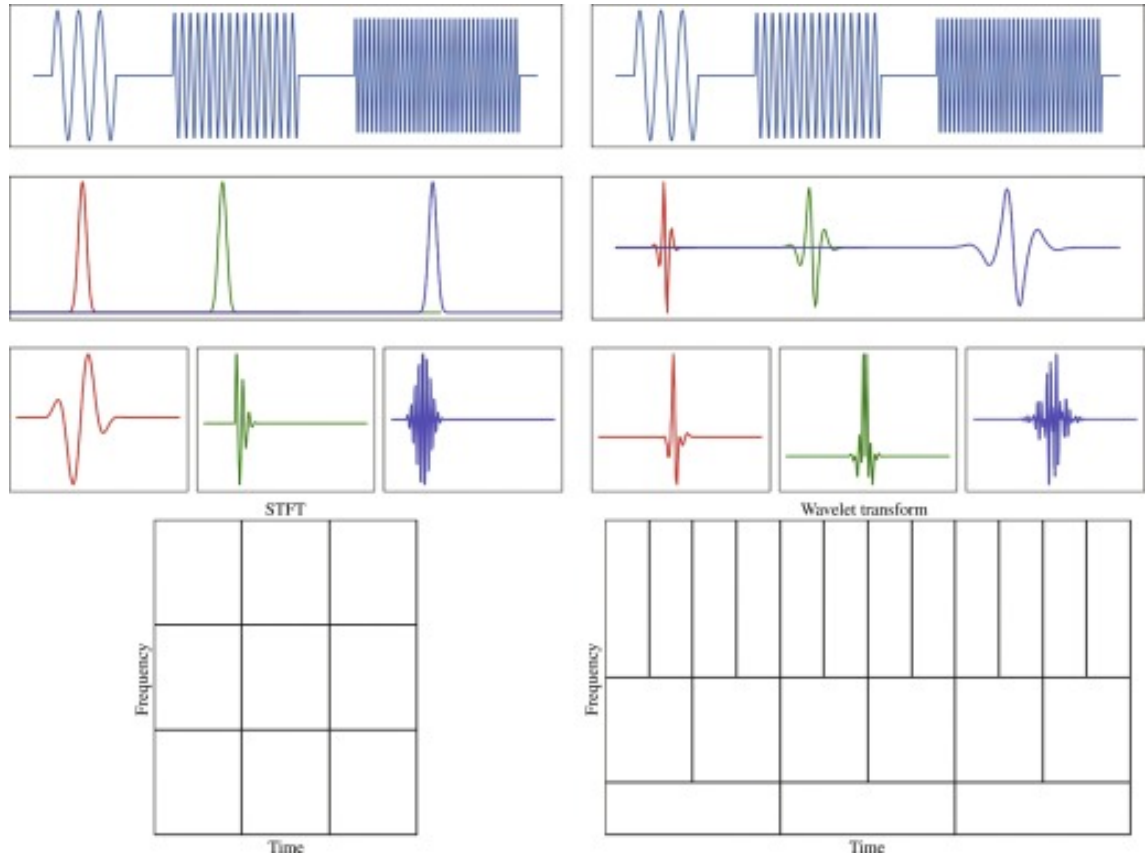
$$\psi_{jk}(t) = 2^{\frac{j}{2}} \psi(2^j t - k).$$

Wavelet được dịch chuyển $\psi_{0k}(t)$ bắt đầu tại $t = k$ và kết thúc tại $t = k + N$, đồ thị của chúng được dịch chuyển sang phải k lần. Wavelet tỷ lệ $\psi_{j0}(t)$ bắt đầu tại $t = 0$ và kết thúc tại $t = N \cdot 2^j$, đồ thị của chúng được nén lại 2^j lần.

CHƯƠNG 2. KIẾN THỨC NỀN TẢNG

Wavelet là những hàm cơ sở $\psi_{jk}(t)$ liên tục theo thời gian. Cơ sở là tập các hàm độc lập tuyến tính dùng tạo ra hàm $f(t)$ được biểu diễn như sau

$$f(t) = \sum_{j,k=-\infty}^{\infty} c_{jk} \psi_{jk}(t).$$



Hình 2.3: So sánh giữa STFT và biến đổi wavelet (nguồn [16])

Biến đổi wavelet liên tục. Với $f(t)$ là một hàm tín hiệu một chiều, biến đổi wavelet liên tục (continuous wavelet transform - CWT) của $f(t)$ sử dụng tích phân với hàm wavelet ψ được biểu diễn như sau

$$W_{\psi}(a, b) = \frac{1}{|a|^{1/2}} \int_{-\infty}^{\infty} x(t) \bar{\psi} \left(\frac{t-b}{a} \right) dt.$$

Trong đó

- W là hệ số biến đổi wavelet liên tục của hàm $f(t)$

- $\bar{\psi}$ là hàm liên hợp phức của wavelet ψ được gọi là hàm wavelet phân tích
- a là hệ số tỷ lệ ($a \in \mathbb{R}^*$), b là hệ số dịch chuyển ($b \in \mathbb{R}$) của hàm wavelet ψ .

Khi đó ta có những hệ số wavelet c_{jk} được tính như sau

$$c_{jk} = W_{\psi}(2^{-j}, k2^{-j}).$$

Biến đổi wavelet liên tục là tổng trên suốt khoảng thời gian của tín hiệu được nhân bởi phiên bản tỷ lệ và dịch chuyển của wavelet. Quá trình này tạo ra các hệ số wavelet là hàm của tỷ lệ và vị trí. Có năm bước để tạo ra CWT

1. Lấy một wavelet và so sánh với khởi đầu của một tín hiệu gốc
2. Tính toán giá trị c , đặc trưng cho tương quan gần của wavelet với đoạn tín hiệu này: c càng lớn, càng có sự tương tự. Chính xác hơn, nếu năng lượng của tín hiệu và wavelet là bằng nhau, c có thể hiểu là hệ số tương quan. Kết quả sẽ phụ thuộc vào wavelet mẹ
3. Dịch chuyển wavelet về phía bên phải và lặp lại bước 1 và 2 cho đến khi hết tín hiệu
4. Định tỷ lệ kéo dãn wavelet là lặp lại tự bước 1 đến bước 3
5. Lặp lại các bước từ 1 đến 4 cho mọi tỷ lệ.

Sau khi hoàn thành, ta sẽ có các hệ số ở các tỷ lệ khác nhau bởi các đoạn khác nhau của tín hiệu. Các hệ số tạo thành kết quả hồi quy của tín hiệu gốc thực hiện trên các wavelet. Để cảm nhận các hệ số đó, ta có thể tạo ra đồ thị với trục x thể hiện vị trí của tín hiệu theo thời gian, trục y đại diện

cho tỉ lệ, màu sắc ở mỗi điểm (x, y) đại diện cho độ lớn của hệ số c . Đồ thị này còn được gọi là Scalogram của biến đổi wavelet liên tục. Các hệ số của biến đổi wavelet liên tục vẽ ra chính xác hình ảnh thời gian và tỉ lệ của tín hiệu.

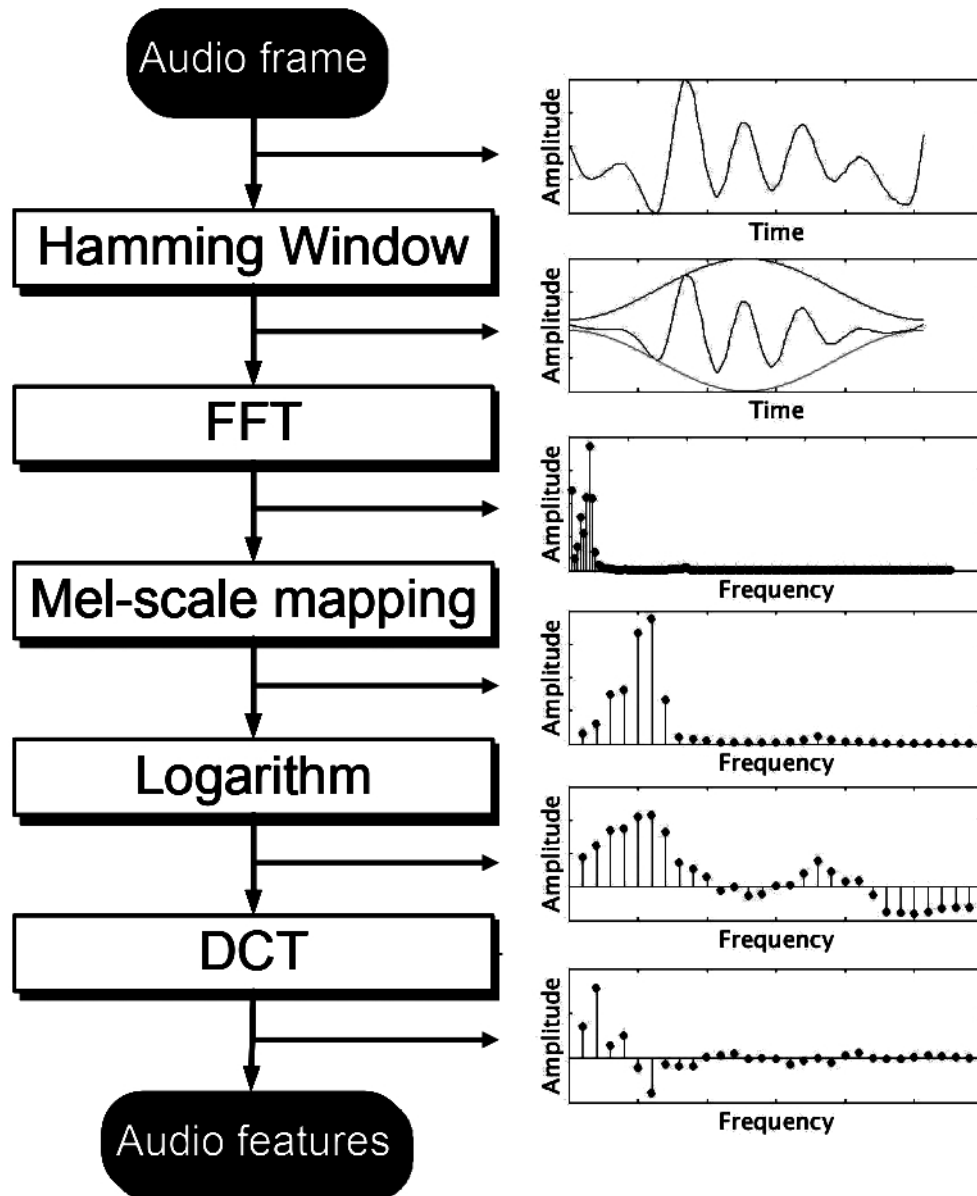
2.1.5. Đặc trưng âm thanh sử dụng Mel frequency cepstral coefficients

Hình 2.4 mô tả các bước cơ bản để trích xuất đặc trưng của âm thanh Mel frequency cepstral coefficients (MFCC), tạo ra các véc-tơ đặc trưng cho quá trình xử lý của các mô hình học máy.

Pre-emphasis. Do đặc điểm cấu tạo thanh quản và các bộ phận phát âm nên giọng nói của chúng ta có đặc điểm các âm ở tần số thấp có mức năng lượng cao, các âm ở tần số cao lại có mức năng lượng khá thấp. Trong khi đó, các tần số cao này vẫn chứa nhiều thông tin về âm vị. Vì vậy chúng ta cần một bước pre-emphasis để kích các tín hiệu ở tần số cao này lên.

Framing. Trong các nghiên cứu trên ngôn ngữ tiếng Anh, thay vì biến đổi Fourier trên cả đoạn âm thanh dài, ta trượt một cửa sổ dọc theo tín hiệu để lấy ra các frame rồi mới áp dụng FFT trên từng frame này. Tốc độ nói của con người trung bình khoảng 3, 4 từ mỗi giây, mỗi từ khoảng 3-4 âm, mỗi âm chia thành 3-4 phần, như vậy 1 giây âm thanh được chia thành 36-40 phần, ta chọn độ rộng mỗi frame khoảng 20-25ms là vừa đủ rộng để bao một phần âm thanh. Các frame được chồng lên nhau khoảng 10ms để có thể giữ lại sự thay đổi giữa các nội dung trong âm thanh.

Windowing. Tuy nhiên, việc cắt frame sẽ làm các giá trị ở hai biên của frame bị giảm đột ngột (về giá trị 0), sẽ dẫn tới hiện tượng khi FFT sang miền tần số sẽ có rất nhiều nhiễu ở tần số cao. Để khắc phục điều này, ta cần làm mượt bằng cách nhân chập frame với một vài loại cửa sổ. Có

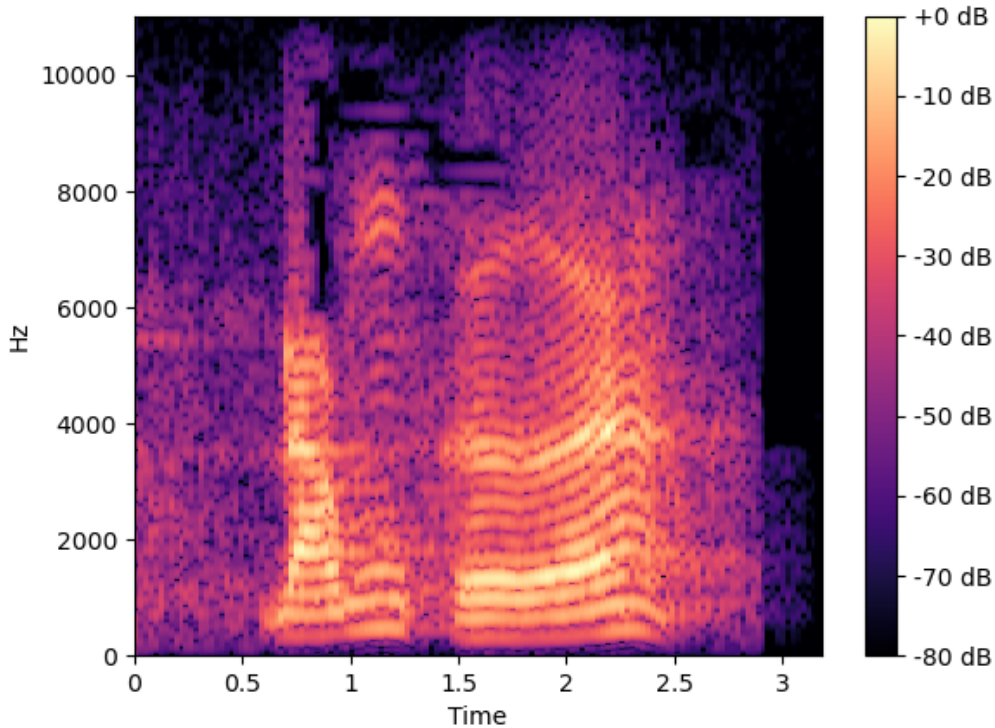


Hình 2.4: Sơ đồ quá trình trích xuất đặc trưng âm thanh (nguồn [17])

một vài loại cửa sổ phổ biến là Hamming window, Hanning window,... có tác dụng làm giá trị biên frame giảm xuống từ từ.

FFT. Mỗi frame ta thu được một danh sách các giá trị biên độ (magnitude) tương ứng với từng tần số từ 0 đến N . Áp dụng FFT trên tất cả các frame, ta đã thu được một spectrogram như Hình 2.5. Trục x là trục thời

gian (tương ứng với thứ tự các frame), trục y thể hiện dải tần số, giá trị biên độ tại từng tần số được thể hiện bằng màu sắc. Qua quan sát spectrogram này, ta nhận thấy các tại các tần số thấp thường có biên độ cao, tần số cao thường có biên độ thấp.

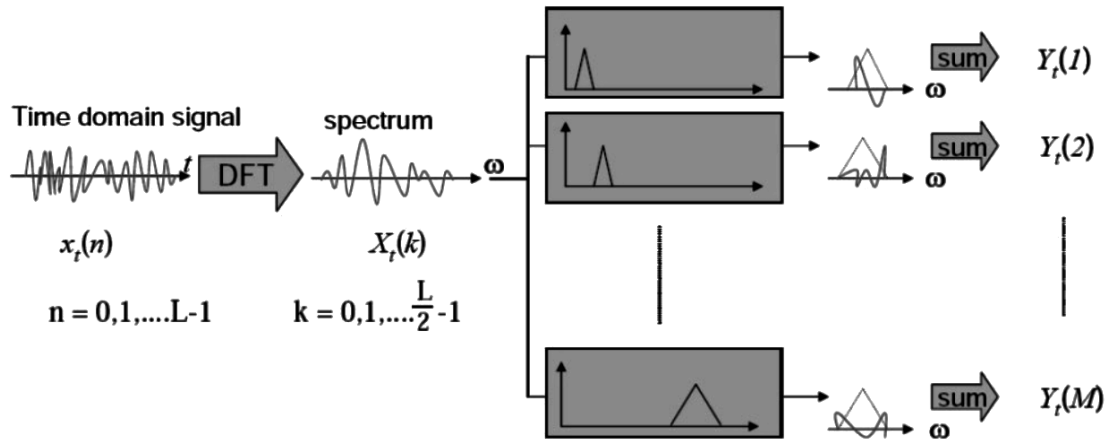


Hình 2.5: Hình ảnh về spectrogram (nguồn [18])

Mel filterbank. Như chúng tôi đã mô tả ở phần trước, cách cảm nhận của tai người là phi tuyến tính, không giống các thiết bị đo. Tai người cảm nhận tốt ở các tần số thấp, kém nhạy cảm với các tần số cao. Ta cần một cơ chế ánh xạ tương tự như vậy.

Trước hết, ta bình phương các giá trị trong spectrogram thu được phổ công suất (DFT power spectrum). Sau đó, ta áp dụng một tập các bộ lọc thông dải Mel-scale trên từng khoảng tần số (mỗi bộ lọc áp dụng trên một dải tần xác định). Giá trị đầu ra của từng bộ lọc là năng lượng dải tần số

mà bộ lọc đó bao phủ được. Ta thu được Mel-scale power spectrum. Ngoài ra, các bộ lọc dùng cho dải tần thấp thường hẹp hơn các bộ lọc dùng cho dải tần số cao.



Hình 2.6: Quá trình thực hiện các bộ lọc Mel-scale (nguồn [11])

Log. Mel filterbank trả về phổ công suất của âm thanh, hay còn gọi là phổ năng lượng. Thực tế rằng con người kém nhạy cảm trong sự thay đổi năng lượng ở các tần số cao, nhạy cảm hơn ở tần số thấp. Vì vậy ta sẽ tính log trên Mel-scale power spectrum. Điều này còn giúp giảm các biến thể âm thanh không đáng kể để nhận diện giọng nói.

IDFT. Phép biến đổi IDFT cũng tương đương với một phép biến đổi cosine rời rạc (discrete cosine transformation - DCT). DCT là một phép biến đổi trực giao. Về mặt toán học, phép biến đổi này tạo ra các tính năng không có quan hệ, có thể hiểu là các tính năng độc lập hoặc có độ tương quan kém với nhau. Trong các thuật toán học máy, tính năng không có quan hệ thường cho hiệu quả tốt hơn.

2.2. Mô hình Gaussian hỗn hợp

Trong lĩnh vực học máy, phân cụm (clustering) là một bài toán học không giám sát, trong đó chúng ta dự định tìm các cụm điểm trong tập dữ liệu ban đầu có chung một số đặc điểm, tính năng. Một trong các thuật toán phân cụm phổ biến hiện nay là k-means [19], sẽ phân cụm dữ liệu theo cách tiếp cận lặp đi lặp lại việc cập nhật các tham số của từng cụm. Cụ thể hơn, những gì k-means sẽ làm là tính toán giá trị trung bình (hoặc điểm trung tâm) của mỗi cụm, và sau đó tính toán khoảng cách của những điểm dữ liệu khác đến từng điểm trung tâm dữ liệu. Cuối cùng, chúng được gán là một phần của cụm được xác định bởi trung tâm gần nhất của chúng. Quá trình này được lặp lại cho đến khi một số tiêu chí hội tụ được đáp ứng, chẳng hạn như khi chúng ta không thấy có thay đổi nào trong việc phân loại các cụm.

Một đặc điểm quan trọng của k-means đó là một phương pháp phân cụm cứng (hard clustering), có nghĩa là nó sẽ liên kết mỗi điểm với một và chỉ một cụm. Hạn chế của cách tiếp cận này là không có giá trị đo hay đại lượng xác suất chính xác cho chúng ta biết mức độ liên kết của một điểm dữ liệu với một cụm cụ thể nào đó

Để tránh các hạn chế của của phương pháp phân cụm cứng như k-means, người ta sử dụng phương pháp phân cụm mềm (soft clustering), một trong các phương pháp phân cụm mềm phổ biến đó là mô hình Gaussian hỗn hợp (Gaussian mixture model - GMMs). Mô hình Gaussian hỗn hợp là một hàm số kết hợp nhiều hàm Gaussian với nhau, với N là số cụm của tập dữ liệu ban đầu mỗi hàm được xác định bởi một hệ số $k \in \{1, \dots, K\}$. Ứng với mỗi Gaussian k trong mô hình hỗn hợp, sẽ công thức tổng quát như sau [20]

$$P(\mathbf{x}|\mu, \Sigma) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu)^T \Sigma^{-1}(\mathbf{x} - \mu)\right).$$

Trong đó

- Giá trị kì vọng (giá trị trung bình) μ_k
- Phương sai σ_k đối phân loại một biến, hay ma trận hiệp phương sai Σ_k đối với phân loại đa biến
- D là số chiều của dữ liệu ban đầu
- Xác suất các điểm cho trước thuộc vào một cụm

$$\pi_k = \frac{\text{Số điểm thuộc cụm } k}{\text{Tổng số điểm dữ liệu ban đầu}},$$

với

$$\sum_{k=1}^K \pi_k = 1.$$

Đầu tiên, giả sử ta muốn biết xác suất của một điểm dữ liệu x_n , với $n \in \{1, \dots, N\}$ và N là tổng số điểm dữ liệu ban đầu có thuộc một Gaussian k hay không, nên ta có mệnh đề cần quan tâm là

$$p(z_{nk} = 1 | x_n).$$

Với z là biến tiềm ẩn (latent variable) chỉ có thể nhận hai giá trị là 1 ứng với việc x_n thuộc Gaussian k , và ngược lại 0 ứng với việc x_n không thuộc Gaussian k . Từ đây ta có được

$$\pi_k = p(z_k = 1).$$

Với $\mathbf{Z} = \{z_1, \dots, z_K\}$ là tập các biến tiềm ẩn có thể có của z , khi một điểm đã thuộc một cụm dữ liệu Gaussian thì không thể thuộc một cụm dữ liệu khác nên ta có giả thiết các giá trị của z xảy ra độc lập nhau, nên

$$p(\mathbf{Z}) = p(z_1 = 1)p(z_2 = 1) \dots p(z_K = 1) = \prod_{k=1}^K \pi_k.$$

Dễ dàng nhận thấy xác suất của một điểm x_n có thuộc Gaussian k hay không lại chính là hàm phân phối xác suất Gaussian

$$p(x_n | z_k = 1) = P(x_n | \mu_k, \Sigma_k),$$

$$p(x_n|\mathbf{Z}) = \prod_{k=1}^K P(x_n|\mu_k, \Sigma_k).$$

Sử dụng quy tắc Bayes, ta có

$$\begin{aligned} p(x_n) &= \sum_{k=1}^K p(x_n|z_k)p(z_k) \\ &= \sum_{k=1}^K \pi_k P(x_n|\mu_k, \Sigma_k). \end{aligned}$$

Đây chính là hàm mục tiêu cho mô hình Gaussian hỗn hợp, và nó phụ thuộc vào tất cả tham số μ_k, Σ_k, π_k mà ta đã đề cập phía trên. Để tối ưu các tham số này, ta phải xác định giá trị lớn nhất của mô hình (maximum likelihood) với hàm xác suất tổng hợp của các tất cả điểm dữ liệu x_n ban đầu

$$\begin{aligned} P(\mathbf{X}) &= \prod_{n=1}^N p(x_n) \\ &= \prod_{n=1}^N \sum_{k=1}^K \pi_k P(x_n|\mu_k, \Sigma_k). \end{aligned}$$

2.3. Mô hình Markov ẩn

Để hiểu về mô hình Markov ẩn (hidden Markov model - HMM), đầu tiên ta phải biết về chuỗi Markov (Markov chains). Với một tập các trạng thái khác nhau $S^t \in \{S_1, \dots, S_K\}$, thì chuỗi Markov được định nghĩa là một biểu đồ chuyển đổi giữa các trạng thái với nhau với một xác suất xảy ra, giả sử từ trạng thái S_i chuyển sang trạng thái S_j sẽ xảy ra với một xác suất p_{ij} .

Bên cạnh đó ta có ma trận xác suất chuyển tiếp (transition matrix) \mathbf{P} biểu diễn các xác suất xảy ra chuyển đổi từ trạng thái này sang trạng thái

khác của chuỗi Markov [20]

$$\mathbf{P} = \begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,k} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ p_{k,1} & p_{k,2} & \cdots & p_{k,k} \end{pmatrix}, \text{ với } \sum_{j=1}^N p_{i,j} = 1.$$

Với xác suất chuyển đổi trạng thái của chuỗi Markov, ta có được xác suất chiếm đóng tại mỗi trạng thái trong thời điểm t

$$p_j(t) = P[S^t = S_j],$$

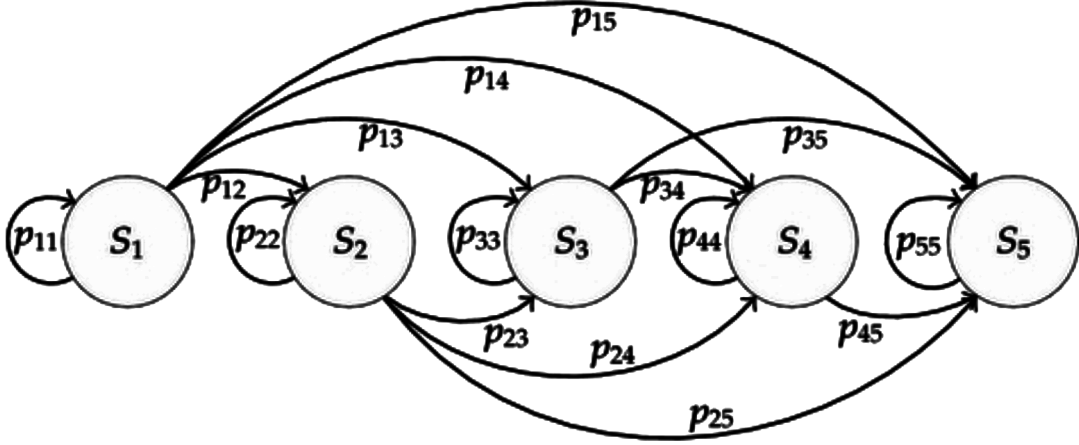
$$p_j(t+1) = \sum_{i=1}^N p_{i,j} p_i(t), \forall i.$$

Nếu phân phối xác suất chiếm đóng tại trạng thái nào đó của một chuỗi Markov hội tụ $p_j(t) \rightarrow \pi(S_j)$ với $t \rightarrow \infty$, ta gọi $\pi(S_j)$ là phân phối tĩnh của chuỗi Markov. Với phân phối tĩnh tồn tại, và xác suất chuyển tiếp $p_{i,j}$, thì chuỗi Markov phải thỏa điều kiện

$$\pi(S_i) = \sum_{j=1}^N p_{i,j} \pi(S_j), \forall i.$$

Mô hình Markov ẩn là một chuỗi Markov vô hình ta không biết trước các tham số của mô hình, nhưng có thể biết các giá trị đầu ra quan sát được, là một mô hình dùng để đặc tả một chuỗi thời gian trong đó giả sử các giá trị của chuỗi thời gian được sinh bởi k biến ngẫu nhiên khác nhau mà các biến ngẫu nhiên này phụ thuộc theo một chuỗi Markov.

Từ những gì ta đã biết về chuỗi Markov, một số đặc điểm chính về các thông số cơ bản của mô hình Markov ẩn gồm có:



Hình 2.7: Ví dụ về chuỗi Markov với 6 trạng thái (nguồn [21])

- Xác suất chuyển tiếp $\mathbf{P} = [p_{i,j}]$, với $i, j = 1, 2, \dots, N$ trong đó N là số trạng thái của chuỗi Markov

$$a_{i,j} = P(S^t = S_j | S^{t-1} = S_i), \text{ với } i, j = 1, 2, \dots, N.$$

- Xác suất chiếm đóng tại một vị ban đầu $\pi = [\pi_i]$ với $i = 1, 2, \dots, N$ và $\pi_i = P(S^1 = i)$.
- Với $O^t \in \{v_1, v_2, \dots, v_k\}$ là tập các kết quả quan sát được. Với sự phân phối xác suất gắn với mỗi trạng thái ta có xác suất của các quan sát đặc trưng

$$b_i(k) = P[O^t = v_k | S^t = S_i] \text{ với } i = 1, 2, \dots, N.$$

Phân phối xác suất phổ biến và tốt nhất được sử dụng trong xử lý giọng nói để mô tả đặc điểm phân bố xác suất của các kết quả quan sát liên tục trong mô hình Markov ẩn là phân phối Gaussian hỗn hợp đa biến cho các quan sát có giá trị véc-tơ ($\mathbf{O}^t \in \mathbb{R}^D$)

$$b_i(O^t) = \sum_{m=1}^M \frac{c_{i,m}}{(2\pi)^{D/2} |\Sigma_{i,m}|^{1/2}} \exp \left(-\frac{1}{2} (\mathbf{O}^t - \boldsymbol{\mu}_{i,m})^T \Sigma_{i,m}^{-1} (\mathbf{O}^t - \boldsymbol{\mu}_{i,m}) \right).$$

2.4. Mô hình mạng tích chập và mô hình long short term memory

2.4.1. Mạng tích chập

Như ta đã biết về mạng nhân tạo (artificial neural network - ANN) là những lớp ẩn liên kết hoàn toàn với nhau, hay còn gọi là các lớp liên kết hoàn toàn (fully connected layer - FC). Trong ANN, các nơ-ron đầu vào sẽ trả về kết quả riêng lẻ và kết quả ấy sẽ trở thành giá trị đầu vào của các nơ-ron tiếp theo trong các lớp sau. Từ đó, nhiều vấn đề được đặt ra cho vấn đề xử lý hình ảnh, giả sử với một hình ảnh màu có kích thước 64×64 khi biểu diễn dưới dạng một tensor $64 \times 64 \times 3$. Khi đó để biểu thị hoàn toàn nội dung của bức ảnh ta cần có 12288 nút ở lớp đầu vào ứng với số lượng điểm ảnh $64 \times 64 \times 3$. Việc đó dẫn đến vấn đề lớn, khi mạng càng nhiều lớp thì số lượng tham số trên mạng tăng một cách nhanh chóng ảnh hưởng đến độ chính xác của quá trình dự đoán. Bên cạnh đó mạng nhân tạo truyền thống không thể hiểu được các vùng điểm ảnh có thể ảnh hưởng đến nhau như thế nào. Từ đó mạng tích chập (convolution neural network - CNN) được ra đời.

Tích chập. Với một hình ảnh đầu vào $I(u, v)$, và một ma trận bộ lọc (kernel) H với kích thước $(2r + 1) \times (2r + 1)$. Khi đó phép tích chập tại một điểm ảnh (x, y) của hình ảnh I được định nghĩa là

$$I'(x, y) = \sum_{i=-r}^r \sum_{j=-r}^r I(x - i, y - j) H(i, j)$$

Phép tích chập sẽ thực hiện liên tục trên các điểm ảnh của ảnh I bằng cách dịch chuyển ma trận lọc H qua từng điểm ảnh và tính giá trị. Kết quả của phép tích chập trên hình ảnh là một ma trận mới mang các tính chất đặc trưng cục bộ của vùng điểm ảnh, và trừu tượng hơn hình ảnh ban đầu.

Đối với một hình ảnh, ta có thể sử dụng một ma trận lọc để trích xuất đặc trưng của cả bức ảnh đó, hay nói cách khác các điểm ảnh chia sẻ hệ số với nhau. Từ đó mô hình đã giải quyết được cả hai vấn đề lớn của mạng nơ-ron truyền thống, đó là giảm được số lượng hệ số và lấy được đặc trưng cục bộ của hình ảnh.

Mạng tích chập. CNN là một tập hợp các lớp tích chập chồng lên nhau và sử dụng các hàm kích hoạt phi tuyến như đơn vị chỉnh lưu tuyến tính (rectified linear unit - ReLU) và tanh để điều chỉnh trọng số trong các nút. Ma trận hình ảnh đầu vào ban đầu khi qua mỗi lớp và các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo.

Trong mô hình CNN có hai vấn đề cần quan tâm là tính bất biến (location invariance) và tính kết hợp (compositionality). Với cùng một đối tượng, nếu được biểu diễn theo các góc độ khác nhau khi sử dụng phép dịch chuyển (translation), phép xoay (rotation) hay phép co giãn (scaling) thì độ chính xác của thuật toán tích chập cũng bị ảnh hưởng đáng kể.

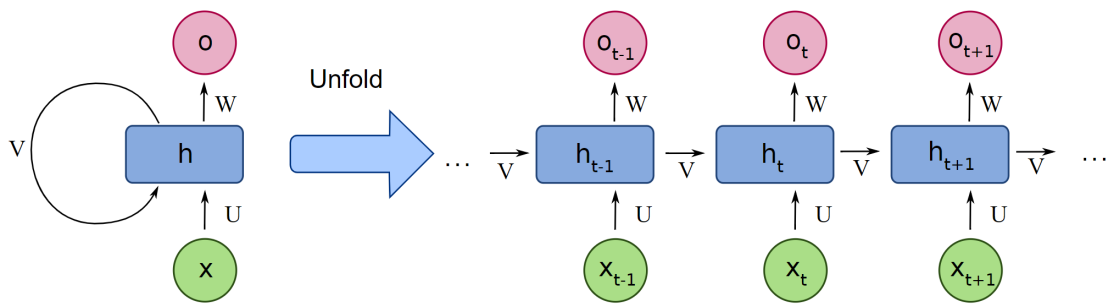
Vì vậy trong CNN có thêm các lớp tổng hợp (pooling layer) để mang lại tính bất biến đối với một đối tượng khi sử dụng các phép biến đổi trên. Các lớp tổng hợp thường đặt sau các lớp tích chập để đơn giản hóa thông tin như lấy các giá trị nổi bật hoặc giá trị trung bình của một vùng, từ đó có thể giảm bớt số lượng nơ-ron nhưng vẫn giữ được các tính chất quan trọng của đối tượng.

2.4.2. Mạng hồi quy

Trên thực tế rất nhiều dạng dữ liệu không phải chỉ cần một thời điểm cụ thể của nó ta có thể phân tích được đặc trưng và hiểu được thông tin nó mang lại, mà phải cần một chuỗi các dữ liệu liên tiếp của đối tượng ta mới có thể đánh giá được. Các loại dữ liệu mà ảnh hưởng với nhau liên tiếp,

hoặc mang theo tính chất thời gian được gọi là các dữ liệu dạng chuỗi. Một số dữ liệu dạng chuỗi như video các hình ảnh liên tục thay đổi theo thời gian, dữ liệu về tim mạch, hay chỉ đơn giản là dữ liệu từ một câu nói, một câu viết.

Để xử lý các loại dữ liệu đó, CNN không thể cho biết được việc ảnh hưởng bởi các dữ liệu liên tiếp nhau, nhưng một mô hình truyền thống mà ta đã biết thì có thể, đó là HMM. Nhưng HMM chỉ sử dụng xác suất liên tục để đưa ra kết quả cuối cùng, không thể sử dụng tốt các tính chất đặc trưng của đối tượng cần xử lý. Từ đó ý tưởng mạng hồi quy (recurrent neural network) ra đời.



Hình 2.8: Hình ảnh minh họa về RNN (nguồn [22])

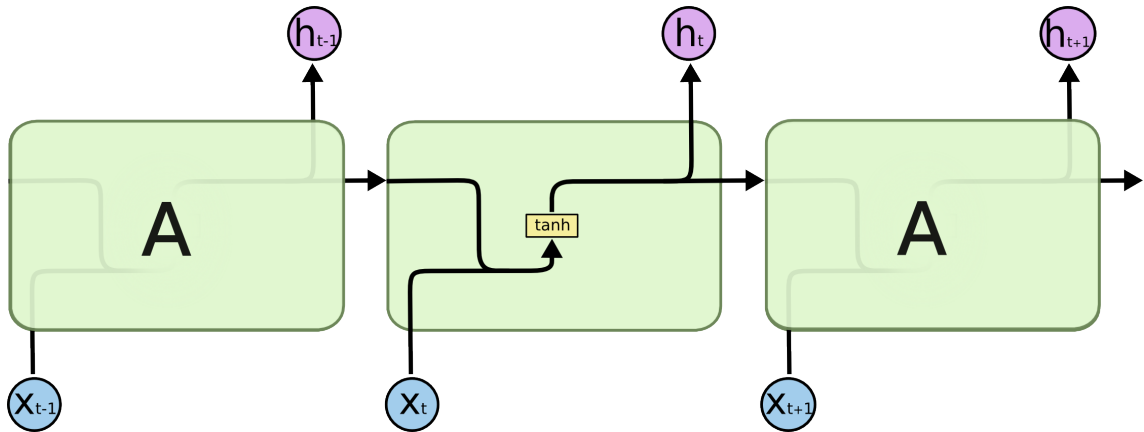
Hình 2.8 ta có mạng RNN gồm các thuộc tính quan trọng sau:

- \mathbf{x} là dữ liệu đầu vào dạng chuỗi, hoặc dữ liệu theo thời gian
- \mathbf{o} là giá trị đầu ra, o_t là giá trị đầu ra tại thời điểm t
- \mathbf{V} biểu diễn quá trình trao đổi thông tin giữa các thời gian liên tiếp nhau theo một trình tự
- \mathbf{h} là các khối chính của RNN, chứa các hệ số và các hàm kích hoạt của mạng. Sẽ sinh ra hai giá trị là giá trị đầu ra o_t ứng với khối h_t tại thời điểm t và một giá trị đầu ra được truyền tiếp tục thông qua đường

giao tiếp V để trở thành giá trị đầu vào cùng với x tại thời điểm kế tiếp.

Ứng với những bài toán khác nhau ta sẽ có những loại RNN khác nhau:

- **Bài toán một-một:** là các bài toán thường được giải quyết bởi các mạng nơ-ron truyền thống, và CNN với một giá trị đầu vào ta luôn nhận được một giá trị đầu ra.
- **Bài toán một-nhiều:** với một giá trị đầu vào ta có nhiều hơn một giá trị đầu ra. Ví dụ như bài toán đánh chú thích cho hình ảnh, chuyển đổi giọng nói thành văn bản.
- **Bài toán nhiều-một:** bài toán nổi bật là phân loại hành động từ một video.
- **Bài toán nhiều-nhiều:** với nhiều giá trị đầu vào ta cũng nhận được nhiều giá trị đầu ra, bài toán thường thấy là trong lĩnh vực xử lý ngôn ngữ tự nhiên về vấn đề dịch ngôn ngữ.



Hình 2.9: Hình ảnh một khối tại thời điểm t của RNN (nguồn [23])

Hình 2.9 mô tả cấu tạo bên trong của một khối RNN cơ bản là hàm tanh của giá trị tổng hợp giữa giá trị đầu vào x_t và kết quả của khối RNN trước đó h_{t-1} .

Với Hình 2.8 và 2.9 ta có

$$\begin{cases} h_0 &= 0 \\ h_t &= f(U * x_t + W * h_{t-1}), \text{ Với } t \geq 1 \end{cases}$$

và

$$f(x) = \tanh(x).$$

Khi đó

$$\begin{aligned} \frac{\partial h_t}{\partial h_{t-1}} &= W * (1 - \tanh^2(U * x_t + W * h_{t-1})) \\ &= W * (1 - h_t^2). \end{aligned}$$

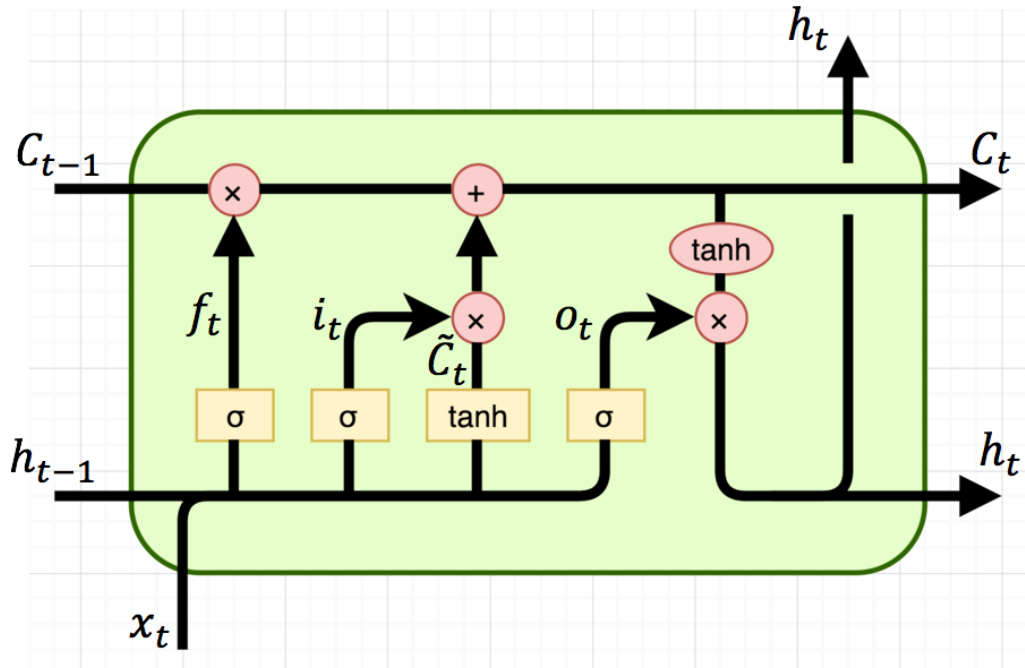
Với $\frac{\partial h_t}{\partial h_i} = \prod_{j=i}^{t-1} \frac{\partial h_{j+1}}{\partial h_j} = \prod_{j=i}^{t-1} [W_{j+1} * (1 - h_{j+1}^2)]$ và $1 - h_{j+1}^2 < 1$, ta thấy được khi $W < 1$ các trạng thái càng xa nhau càng khó ảnh hưởng đến nhau, dẫn đến việc giảm về 0 của gradient.

2.4.3. Long short term memory

Vấn đề lớn ở mô hình RNN là việc giảm về 0 của gradient khi các trạng thái cách xa các trạng thái trước đó dẫn đến các hệ số không được cập nhật. Theo lý thuyết RNN có thể mang thông tin từ các lớp trước đến các lớp sau, nhưng thực tế thông tin chỉ mang qua một số lượng trạng thái nhất định, các thông tin càng được truyền đi xa sẽ dẫn đến việc biến mất gradient làm cho việc học không hiệu quả. Ta có thể nói RNN chỉ có thể học thông tin từ các trạng thái gần nó (short term memory). Để giải quyết vấn đề này mô hình long short term memory (LSTM) ra đời.

Hình 2.10 mô tả trạng thái thứ t của LSTM với

- \mathbf{c} là các trạng thái tế bào (cell state), c_t là trạng thái tế bào tại thời điểm t



Hình 2.10: Hình ảnh một khối tại thời điểm t của LSTM (nguồn [23])

- \mathbf{h} là các trạng thái ẩn (hidden state), h_t là trạng thái ẩn tại thời điểm t .

Điểm mới ở LSTM so với RNN là \mathbf{c} , khối tính toán LSTM tại thời điểm t ngoài các giá trị đầu vào h_{t-1} , x_t như RNN thì có thêm giá trị c_{t-1} đầu vào. Và đầu ra của khối đó là c_t và h_t sẽ trở thành giá trị đầu vào cho khối tiếp theo. Nhờ có \mathbf{c} đóng vai trò như một băng truyền đối với mô hình RNN, các thông tin nào quan trọng và được sử dụng ở các khối phía sau sẽ được gửi đến và dùng khi cần thiết. LSTM có thể mang thông tin đi xa hơn và tránh được vấn đề tiêu biến gradient.

Với Hình 2.10 ta có các công là

$$\begin{aligned}f_t &= \sigma(U_f * x_t + W_f * h_{t-1}), \\i_t &= \sigma(U_i * x_t + W_i * h_{t-1}), \\o_t &= \sigma(U_o * x_t + W_o * h_{t-1}), \\\tilde{c}_t &= \tanh(U_c * x_t + W_c * h_{t-1}), \\c_t &= f_t * c_{t-1} + i_t * \tilde{c}_t.\end{aligned}$$

Khi đó

$$\frac{\partial c_t}{\partial c_{t-1}} = f_t.$$

Do các giá trị trên các khối tính toán cần ít khi phải quên đi nên $f_t \approx 1$, nên thông tin được truyền đi xa hơn giảm bớt được sự giảm về 0 của gradient.

2.5. Mô hình mạng đối kháng tạo sinh

2.5.1. Giới thiệu

Ta đã biết khi thực hiện huấn luyện các mô hình AI, học máy ta cần có một tập dữ liệu khá lớn để giúp cho các mô hình có thể dự đoán chính xác hơn. Các mô hình đa phần sẽ thực hiện trích xuất đặc trưng trên dữ liệu và đưa ra dự đoán cho các dữ liệu khác. Vì vậy cần có một mô hình có thể giải quyết vấn đề với lượng ít dữ liệu, hoặc tự bản thân mô hình có thể sinh ra dữ liệu mới. Đó là mô hình mạng đối kháng tạo sinh (generative adversarial network - GAN). Tại sao lại là đối kháng, vì GAN được cấu thành từ hai mạng là mạng sinh (generator) và mạng phân biệt (discriminator), hai mạng này sẽ luôn thực hiện các chức năng đối kháng nhau trong quá trình huấn luyện GAN.

Để hiểu đơn giản về GAN, ta có thể xem mạng sinh là những người làm hàng hóa giả, và mạng phân biệt là những người sử dụng hàng hóa. Người

làm hàng hóa giả sẽ cố gắng làm ra các hàng hóa mà người dùng không thể phân biệt được. Còn người dùng thì ngược lại phải phân biệt được đâu là hàng thật đâu là hàng giả. Quá trình huấn luyện mô hình GAN người dùng sẽ gồm có hai công việc, đầu tiên phải phân biệt được đâu là hàng thật đâu là hàng giả, và sau đó là thông báo đâu là hàng giả. Và người làm giả hàng hóa sẽ tiếp nhận thông tin và cải thiện hơn. Dần dần người làm hàng giả làm ra các hàng hóa càng giống thật hơn, người dùng cũng thành thạo trong việc phân biệt hàng thật và giả. Nhưng mục tiêu cuối cùng của GAN lại có lợi hơn cho người làm hàng giả, chúng ta mong đợi hàng hóa được làm giả sẽ đánh lừa được người dùng càng nhiều càng tốt.

Với ý tưởng trên mô hình sẽ cấu thành bởi hai mạng con là mạng sinh $G(z, \theta_g)$ được xem như một hàm sinh với các tham số θ_g . Mạng sinh sẽ học phân phối xác suất sinh p_g thông qua tập dữ liệu x , với giá trị đầu vào của mạng sinh là một nhiễu z có phân phối xác suất p_z hay ta có thể hiểu mạng sinh nhận giá trị nhiễu đầu vào là $p_z(z)$. Bên cạnh đó mạng thứ hai là mạng phân biệt $D(x, \theta_d)$ có kết quả đầu ra là một giá trị xác suất. Khi đó $D(x)$ thể hiện cho xác suất x có thuộc về tập dữ liệu gốc ban đầu hay từ kết quả của mô hình sinh với xác suất p_g . Khi ghép hai mô hình lại, chúng ta mong muốn mạng phân biệt D sẽ tối đa hóa xác suất nhận biết chính xác nhãn cho cả hai loại dữ liệu từ dữ liệu huấn luyện ban đầu và dữ liệu sinh ra từ mạng sinh G . Ngược lại mạng sinh G sẽ được huấn luyện để tối thiểu hàm $\log(1 - D(G(z)))$ [24].

$$\begin{aligned}\max_D V(D) &= \mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))], \\ \min_G V(G) &= \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))].\end{aligned}$$

Ta cũng có thể dễ hiểu hai mạng sinh G và mạng phân biệt D đang như hai đối thủ chơi một trò chơi đối kháng với hàm giá trị là $V(G, D)$, khi đó ta có

$$\min_G \max_D V(G, D) = \mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))].$$

2.5.2. So sánh với tấn công đối kháng

Với nhiều người mới bắt đầu tìm hiểu về tấn công đối kháng thường sẽ nhầm lẫn hai khái niệm này với nhau. Chúng ta rất dễ nhầm lẫn mạng đối kháng tạo sinh với quá trình tạo mẫu tấn công đối kháng là một. Vì vậy, chúng sẽ so sánh cơ bản về điểm giống và khác nhau của hai khái niệm này

Điểm giống nhau. Hai khái niệm có những điểm giống nhau như sau

- Điểm giống nhau cơ bản đó là “đối kháng”, giống nhau ở đây là trong quá trình thực hiện cả hai đều có một mô hình nhận biết sẽ có vai trò là mục tiêu mà ta cần phải đánh lừa. Ví dụ như trong mạng đối kháng tạo sinh thì có mô hình phân biệt, còn trong tấn công đối kháng có mô hình mục tiêu tấn công.
- Cả hai quá trình đều mong muốn tạo ra các mẫu dữ liệu có thể làm mô hình phân biệt, hoặc mô hình mục tiêu nhận diện sai lệch theo mục đích được đặt ra.
- Cả hai đều là các bài toán tối ưu hàm mất mát để đạt được giá trị mong muốn.

Điểm khác nhau. Với những điểm giống nhau cơ bản, nhưng thực tế cả hai khái niệm được áp dụng vào các trường hợp khác nhau như

- Mạng đối kháng tạo sinh: mục tiêu cuối cùng là tạo ra một mạng sinh dữ liệu giống với các tính chất của tập dữ liệu ban đầu. Từ đó, chúng ta có thể tăng cường dữ liệu huấn luyện cho mô hình.
- Tấn công đối kháng: mục tiêu là tạo ra các mẫu đối kháng khiến các mô hình có độ chính xác cao nhận diện sai lệch. Mặc dù vậy, các mẫu

tấn công đó vẫn thể hiện đúng nội dung gốc ban đầu đối với nhận thức của con người.

2.6. Cơ chế attention

Ta đã biết các vấn đề về trí tuệ nhân tạo, học máy chính là sự mô phỏng lại quá trình xử lý, tính toán, ra quyết định của não người, từ đó các mô hình học máy được ra đời và phát triển. Lấy một vài ví dụ thực tế như khi ta đọc một bài văn hay một bài báo mỗi câu, mỗi đoạn có thể mang một thông tin riêng của nó, hoặc nhiều câu chỉ là bổ sung ý cho một câu một đoạn khác. Nghe qua ta có thể thấy chỉ việc đọc một mẫu báo nhỏ thôi cũng đã rất nhiều thông tin, nhưng não chúng ta lại có thể chú ý vào các từ khóa cốt yếu để có thể tóm tắt lại nội dung chính của bài văn, bài báo ấy một cách nhanh chóng. Hoặc điển hình trong xử lý ảnh, mắt người có tầm nhìn rất rộng tuy nhiên khi lái xe chúng ta hầu như chỉ xử lý một phần nhỏ của hình ảnh mà mắt thu được và đưa ra quyết định một cách chính xác cho việc di chuyển. Cơ chế này giúp não bộ có thể xử lý nhanh chóng thông tin, với một mức năng lượng cần thiết thấp mà vẫn đem lại kết quả đáng tin cậy. Để mô phỏng lại cơ chế chú ý ấy, các nghiên cứu về cơ chế attention trong học máy ra đời.

Như ta đã biết người nhiều nghiên cứu thường dùng các mô hình RNN cho bài toán nhiều-nhiều (hay chuỗi sang chuỗi) mà điển hình nhất là bài toán máy dịch ngôn ngữ. Đối với bài toán này, mạng nhân tạo thường được cấu thành từ hai thành phần chính là bộ mã hóa (encoder) và bộ giải mã (decoder). Với đầu vào của bộ mã hóa là một câu ở ngôn ngữ gốc mà ta cần dịch, và đầu ra là một véc-tơ thông tin mang toàn bộ thông tin của câu nói cần dịch. Sau đó bộ mã hóa sẽ sử dụng véc-tơ thông tin đó để thực hiện quá trình giải mã qua từng mốc thời gian để đưa ra kết quả cuối cùng là một câu ở ngôn ngữ đích mong muốn. Việc thực hiện mã hóa các câu có độ dài khác

CHƯƠNG 2. KIẾN THỨC NỀN TẢNG

nhau thành một véc-tơ cố định khiến cho các câu dài không hoàn toàn tốt, mặc dù các mô hình cải biến của RNN như LSTM hay GRU đã giảm bớt vấn đề biến mất củaradient. Trong bài báo “Neural machine translation by jointly learning to align and translate” [25], tác giả đã đề xuất một cơ chế giúp mô hình có thể chú trọng vào những thành phần quan trọng hay còn gọi là cơ chế attention.

Trong bài toán máy dịch ngôn ngữ ta có một chuỗi văn bản gốc ban đầu x với độ dài n và một chuỗi văn bản dịch y với độ dài là m , ta có

$$x = [x_1, x_2, \dots, x_n],$$

$$y = [y_1, y_2, \dots, y_m].$$

Với mỗi giá trị đầu ra của bộ mã hóa y_i sẽ phụ thuộc vào giá trị y_{i-1} , trạng thái ẩn của RNN s_i và véc-tơ thông tin c , với g là một hàm phi tuyến ta có

$$p(y_i|y_1, \dots, y_{i-1}, x) = g(y_{i-1}, s_i, c_i).$$

Trạng thái ẩn s_i sẽ phụ thuộc vào trạng thái ẩn s_{i-1} từ y_{i-1} và một véc-tơ thông tin c_i thông qua RNN được biểu diễn như sau

$$s_i = f(s_{i-1}, y_{i-1}, c_i).$$

Véc-tơ thông tin được tạo bởi tổng trọng số của các giá trị đầu ra h_j tại thời điểm j , với α_{ij} (alignment score) là trọng số thể hiện mức độ chú ý của từng giá trị h_j . Khi đó c_i được tính như sau

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j.$$

Với một hàm số z nhận giá trị đầu vào là trạng thái ẩn trước đó của bộ giải mã s_{i-1} và giá trị đầu ra của bộ mã hóa h_j , khi đó giá trị đầu ra của

CHƯƠNG 2. KIẾN THỨC NỀN TẢNG

hàm z là một véc-tơ năng lượng chú ý e_{ij} , ta có thể biểu diễn α_{ij} như sau

$$e_{ij} = a(s_{i-1}, h_j),$$
$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^T \exp(e_{ik})}.$$

3 Một số nghiên cứu liên quan

Theo khảo sát cá nhân của chúng tôi, các tài liệu về tấn công đối kháng trên các mô hình nhận diện giọng nói tiếng Anh hiện tại rất nhiều, tuy nhiên lại không có một tài liệu nào áp dụng trên ngôn ngữ tiếng Việt. Vì vậy trong luận văn này, chúng tôi sẽ dựa trên các bài báo cáo nghiên cứu chủ yếu dựa trên ngôn ngữ tiếng Anh, từ đó tham khảo và hiện thực trên ngôn ngữ tiếng Việt.

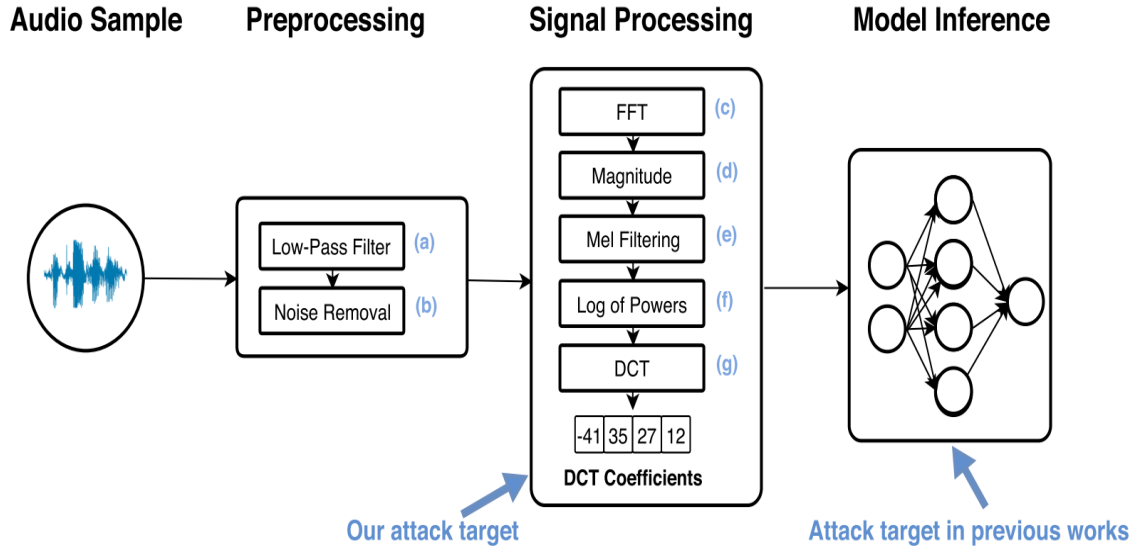
3.1. Tấn công trực tiếp mô hình hộp đen

Mô hình hộp đen là các mô hình điều khiển thông minh bằng giọng nói (intelligent voice control - IVC) đang được ứng dụng, và bán trên thị trường hiện nay [26]. Các mô hình này không thể biết được các thông số, hay cách hoạt động bên trong mô hình là như thế nào, ta chỉ có thể tương tác với giá trị đầu vào và kết quả đầu ra. Hình thức tấn công trên mô hình hộp đen có nhiều loại tấn công khác nhau, chủ yếu dựa trên các quá trình xử lý âm thanh ban đầu để rút trích các véc-tơ đặc trưng trước khi truyền vào các mô hình học máy. Các nhà nghiên cứu đã khai thác quá trình trích xuất đặc trưng MFCC thì có nhiều nguồn âm thanh khác nhau lại cho các véc-tơ tương tự nhau.

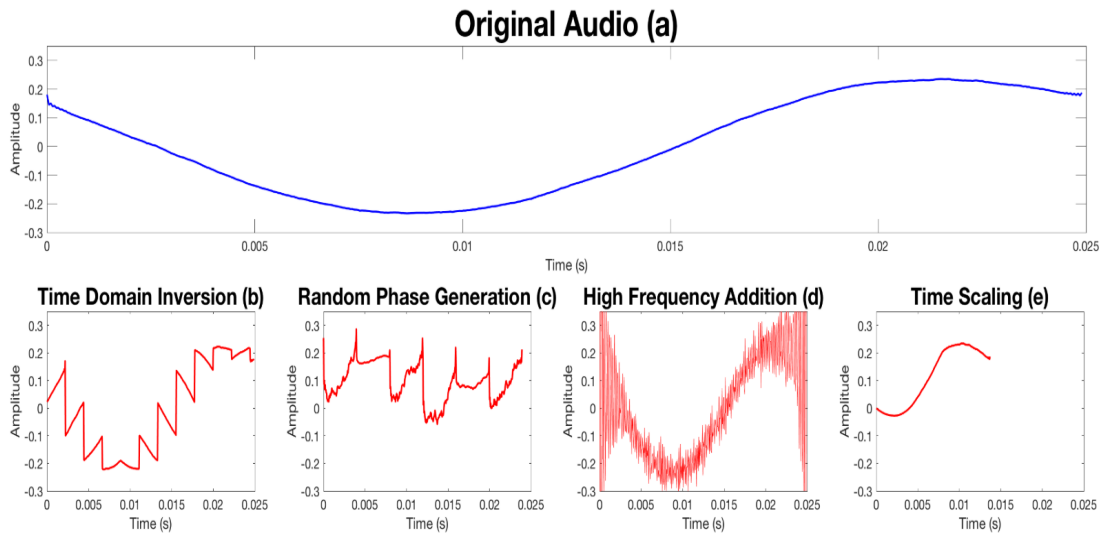
Trong bài nghiên cứu này, chúng tôi trình bày lại các phương pháp tiếp

CHƯƠNG 3. MỘT SỐ NGHIÊN CỨU LIÊN QUAN

cận nổi bật nhất dùng để tấn công vào các mô hình hộp đen đã và đang được nghiên cứu. Các cuộc tấn công sẽ chủ yếu ở quá trình tiền xử lý âm thanh, và quá trình xử lý âm thanh rút trích đặc trưng của tính hiệu.



Hình 3.1: Các bước chung của một mô hình nhận diện giọng nói (nguồn [26])



Hình 3.2: Mô tả cơ bản các cuộc tấn công hộp đen (nguồn [26])

3.1.1. Đảo miền thời gian

Ta đã biết quá trình trích xuất đặc trưng âm thanh, thực hiện FFT trên các frame nhỏ khác nhau để phân tích ra các biên độ và pha của từng tần số cấu tạo nên đoạn âm thanh trong frame. Và FFT là một hàm số ánh xạ nhiều một, với nhiều giá trị đầu vào khác nhau nhưng được tạo ra từ tổ hợp các tần số giống nhau, khi sử dụng FFT ta cũng sẽ thu được các giá trị giống nhau tương ứng. Vậy khi đó ta có thể thay đổi âm thanh bất kì với điều kiện đó, một cách đơn giản nhất đó là đảo các giá trị của mỗi frame theo miền thời gian. Như Hình 3.2(b) khi đảo ngược các giá trị ở mỗi frame khiến cho toàn bộ tín hiệu sẽ không còn mượt, nhiều vị trí thay đổi cường độ đột ngột gây ra nhiễu loạn khiến tai người khó có thể nghe và có thể diễn giải được [26][27].

3.1.2. Tạo pha ngẫu nhiên

Như ta đã biết, thông qua FFT kết quả trả về là một số phức dạng $a + bi$ để biểu diễn biên độ và pha của các tần số kết hợp nên âm thanh. Dựa vào Hình 3.2(c) ta thấy được bước tiếp theo của FFT là trích xuất biên độ của các tần số. Giả sử ta có một tần số $x = a_0 + b_0i$ công thức tính độ lớn biên độ

$$A = \sqrt{a_0^2 + b_0^2}.$$

Vì hàm lấy biên độ là hàm nhiều một, ta có thể dễ dàng tìm kiếm một cặp hệ số a_1, b_1 sau cho $A = \sqrt{a_0^2 + b_0^2} = \sqrt{a_1^2 + b_1^2}$. Khi đó ta có thể tìm kiếm một cách ngẫu nhiên a_n, b_n thỏa điều kiện biên độ như âm thanh gốc, và thu được các tín hiệu âm thanh mới có pha ngẫu nhiên làm cho tai người khi nghe khó có thể nhận diện được âm thanh đó là gì.

3.1.3. Thêm tần số cao

Như ta đã biết tai người rất kém nhạy cảm với các tần số cao, thường ta chỉ có thể nghe âm thanh với tần số $20Hz$ đến $20000Hz$. Nhưng các tần số càng cao lại có mức năng lượng rất thấp, nên lúc đó sẽ bị loại bỏ trong quá trình tiền xử lý âm thanh với các bộ lọc bỏ qua những loại tần số này, Hình 3.2(d). Từ ý tưởng đó, nhiều nghiên cứu đã áp dụng thêm tần số cao vào các âm thanh, để tấn công vào các hệ thống nhận diện giọng nói, nổi bật có thể nhắc đến là DolphinAttacks [28].

3.1.4. Nén thời gian

Các mô hình nhận diện giọng nói có nhận diện giọng nói của người dùng với nhiều tốc độ nói khác nhau. Nhưng con người khó có thể nghe thấy và nhận ra các từ được nói với tốc độ nhanh, so với các từ tốc độ chậm. Ở cách tấn công này, các nhà nghiên cứu đẩy nhanh các lệnh thoại đến một thời điểm mà chúng vẫn có thể phiên âm chính xác. Thực hiện nén âm thanh trong miền thời gian, loại bỏ các vùng không cần thiết và duy trì tần số của mẫu. Cho nên âm thanh có thời lượng ngắn nhưng vẫn giữ được phổ giống với bản gốc [26]. Loại tấn công này tuy có thể dễ dàng nhận ra bởi người dùng vì các lệnh thoại mang nội dung tấn công vẫn không bị biến đổi, nhưng khi kết hợp với các loại tấn công khác sẽ tạo hiệu quả đáng kể.

3.1.5. Tấn công vào mô hình nhận diện phân loại giọng nói tiếng Anh

Kịch bản tấn công. Trong báo cáo nghiên cứu của Alzantot và cộng sự [3] đã xây dựng một kịch bản tấn công vào các mô hình nhận diện phân loại giọng nói tiếng Anh cơ bản. Trong đó kẻ tấn công không quan tâm đến về kiến trúc và các tham số của mô hình mà chỉ quan tâm giá trị đầu ra của lớp phân loại cuối cùng của mô hình. Với quyền truy vấn kết quả đầu ra, ta có thể hiểu bài toán được biểu diễn với mô hình hình mục tiêu tấn công là $f(x)$. Như vậy $f : X \rightarrow [0, 1]^K$, khi đó f là một hàm số ta không biết trước tham số chỉ có thể có được kết quả đầu ra. Bên cạnh đó X là không gian của tất cả mẫu âm thanh đầu vào có thể có, và $[0, 1]^K$ đại diện cho xác suất dự đoán của các điểm đầu ra với K nhãn.

Cách tiến hành. Với bài toán được biểu diễn như trên, nhóm tác giả không thể sử dụng quá trình lan truyền ngược để cập nhật trực tiếp các giá trị cho các mẫu âm thanh ban đầu. Vì vậy nhóm tác giả đã sử dụng phương pháp tiếp cận dựa trên giải thuật di truyền gradient tự do (gradient free genetic) để tạo nên các mẫu đối kháng. Với mẫu âm thanh x đã được phân loại chính xác dựa trên nội dung và mục tiêu phân loại t , nhóm tác giả sử dụng giải thuật trên tạo ra mẫu âm thanh đối kháng x_{adv} vẫn mang nội dung ban đầu nhưng khiến mô hình nhận diện sai lệch. Khởi đầu nhóm tác giả thêm ngẫu nhiên các nhiễu ngẫu nhiên nhỏ để tạo ra một quần thể các mẫu đối kháng ứng cử viên. Nhằm giảm ảnh hưởng tiếng ồn lên nhận thức của con người, các nhiễu này sẽ được thêm vào các bit cuối tại mỗi điểm giá trị của các mẫu âm thanh ban đầu. Từ quần thể đã có, họ bắt đầu tính toán điểm số cho từng thành viên thông qua giá trị dự đoán của nhãn mục tiêu. Sau đó sử dụng các ứng viên trong quần thể sẽ được chọn lọc, giao phối chéo và gây đột biến để tạo ra các ứng viên đời sau. Quá trình trên sẽ được lặp lại với một số lần lặp nhất định, hoặc tạo ra một mẫu âm thanh

CHƯƠNG 3. MỘT SỐ NGHIÊN CỨU LIÊN QUAN

đối kháng có khả năng làm sai lệch mô hình với mục tiêu t đã chỉ định.

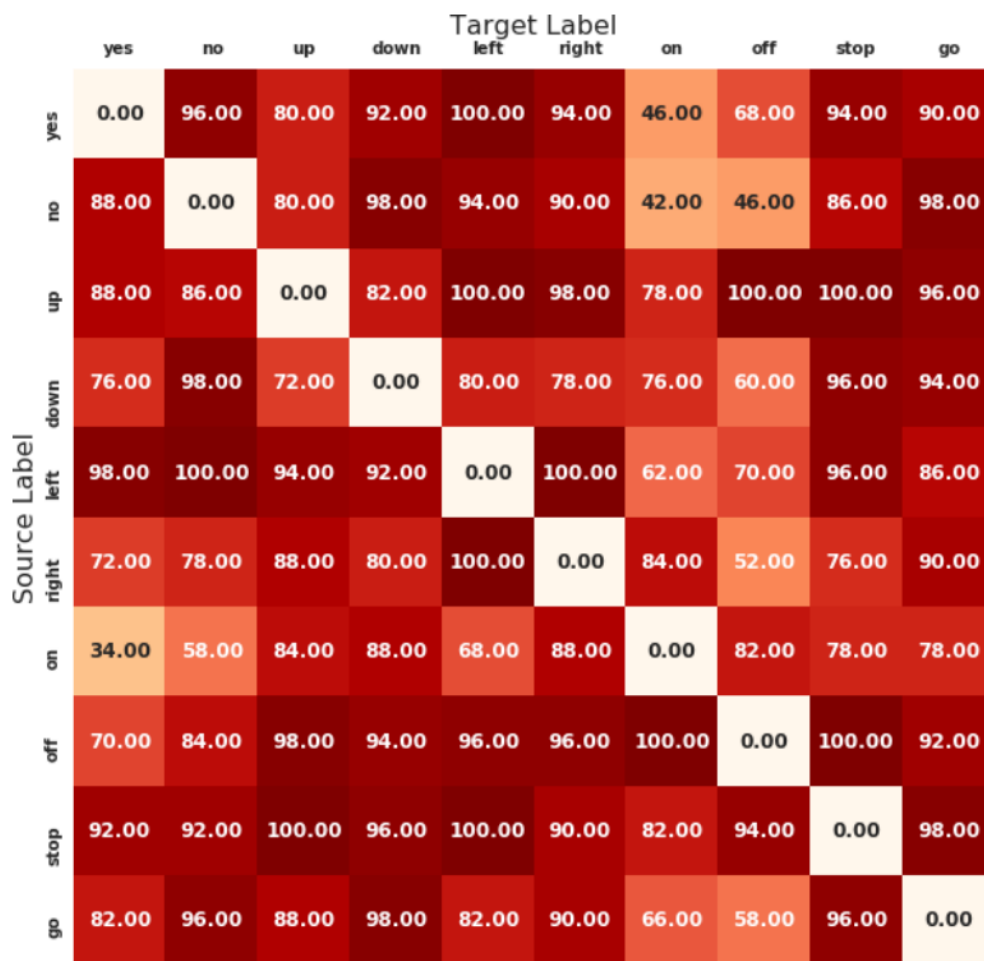
Dữ liệu. Bài toán trên được nhóm tác giả giả quyết với các mô hình nhận diện phân loại giọng nói tiếng Anh, cho nên bộ dữ liệu được sử dụng là “google speech command”. Bộ dữ liệu được sử dụng bao gồm 65000 mẫu âm thanh khác nhau, mỗi mẫu âm thanh có thời lượng là 1 giây và là từ đơn của 10 lớp như Bảng 3.1

Số thứ tự	Câu lệnh
1	yes
2	no
3	up
4	down
5	left
6	right
7	on
8	off
9	stop
10	go

Bảng 3.1: Bảng các lớp của tập dữ liệu “google speech command” (nguồn [3])

Kết quả. Bắt đầu quá trình tạo các mẫu đối kháng, nhóm tác giả lựa chọn ngẫu nhiên mỗi lớp 50 mẫu âm thanh khác nhau đã được phân loại chính xác từ tập dữ liệu ban đầu. Khi đó mỗi mẫu âm thanh ban đầu sẽ tạo ra 9 mẫu âm thanh đối kháng ứng với các lớp còn lại, hoàn thành quá trình tạo mẫu sẽ có tổng 4500 mẫu âm thanh đối kháng khác nhau. Đánh giá kết quả các mẫu âm thanh đối kháng được tạo ra, nhóm tác giả cho thấy được giải thuật tạo mẫu thành công lên đến 87%, Hình 3.3. Sau cùng, các mẫu âm thanh đối kháng sẽ được đánh giá so với âm thanh gốc thông qua nhận biết của con người được kết quả như Bảng 3.2

CHƯƠNG 3. MỘT SỐ NGHIÊN CỨU LIÊN QUAN



Hình 3.3: Kết quả tạo mẫu thành công sử dụng giải thuật di truyền gradient tự do (nguồn [3])

Nội dung gốc	Nội dung mục tiêu	Nội dung khác
89%	0.6%	9.4%

Bảng 3.2: Bảng kết quả phân biệt của con người với các mẫu đối kháng (nguồn [3])

3.2. Sử dụng mô hình hộp trắng

3.2.1. CommanderSong

Kịch bản tấn công. Bài báo CommanderSong [1] đưa ra một số thách thức kỹ thuật đối với một mẫu âm thanh đối kháng cần phải vượt qua như sau:

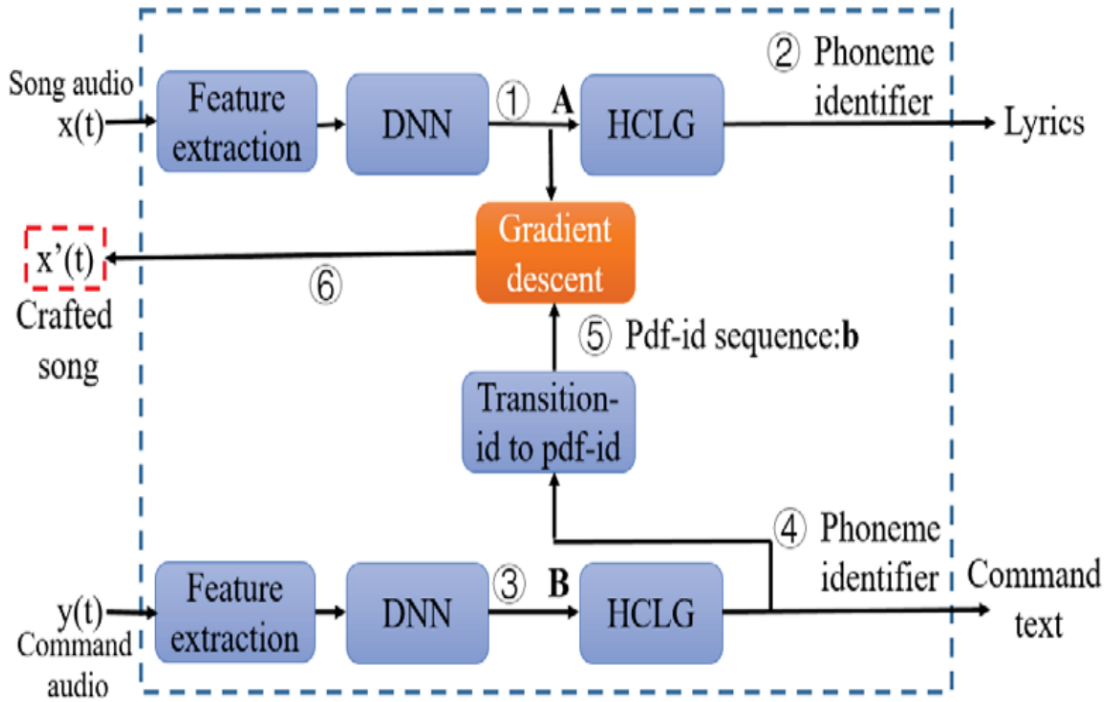
- **C1:** Mẫu âm thanh đối kháng sẽ có hiệu quả đối với các mô hình nhận diện âm thanh trong môi trường thế giới thực, âm thanh phức tạp, với sự hiện diện của các loại tiếng ồn từ loa và các loại tiếng ồn khác.
- **C2:** Mẫu âm thanh đối kháng phải mang tính ẩn dấu, người bình thường khó có thể nhận biết đó là một loại âm thanh mới mục đích tấn công các thiết bị thông minh.
- **C3:** Mẫu âm thanh đối kháng có tác dụng, phải dễ dàng phát tán, được thực thi trên các kênh thông tin khác nhau ảnh hưởng đến một số lượng lớn các thiết bị nhận diện giọng nói.

Dựa vào ba điều kiện trên, các nhà nghiên cứu đã lựa chọn các bài hát làm sóng mang mang theo các lệnh ẩn mà các hệ thống nhận diện giọng nói có thể nhận thấy và thực thi. Vì các bài hát khá phổ biến và rất dễ được tiếp cận bởi nhiều người. Hơn nữa, việc giải trí không bị hạn chế bằng cách sử dụng radio, đầu đĩa CD, hoặc máy tính để bàn. Một thiết bị di động, ví dụ điện thoại Android hoặc Apple iPhone, cho phép mọi người thưởng thức các bài hát ở mọi nơi.

Cách tiến hành. Để xây dựng cuộc tấn công, các nhà nghiên cứu sử dụng mô hình nhận diện giọng nói mã nguồn mở (hay còn gọi là mô hình hộp trắng) Kaldi ASpIRE Chain Model bao gồm mô hình âm thanh và mô

CHƯƠNG 3. MỘT SỐ NGHIÊN CỨU LIÊN QUAN

hình ngôn ngữ. Thông qua phương pháp giảm gradient (gradient descent), tạo mẫu âm thanh đối kháng bằng cách tổng hợp kết quả đầu ra của mô hình âm thanh với đầu vào là cả bài hát làm sóng mang và lệnh thoại đã cho.



Hình 3.4: Các bước thực hiện tạo Commander Song (nguồn [1])

Hình 3.3 mô tả quá trình tạo ra mẫu âm thanh đối kháng từ một mẫu bài hát và một lệnh thực thi. Với $x(t)$ là bài hát gốc ban đầu, $y(t)$ là âm thanh giọng nói thuần túy của một lệnh thực thi dùng để tấn công. Khi sử dụng Kaldi phân tích đầu ra của mô hình mạng học sâu (deep neural network - DNN) đối với bài hát gốc $x(t)$ (ở bước 1) ta có được ma trận A chứa xác suất xảy ra của mỗi pdf-id (probability distribution function identifier) tại mỗi frames. Giả sử có n frames và k pdf-id, ta có $a_{i,j} (1 \leq i \leq n, 1 \leq j \leq k)$. Tại mỗi frame, ta lấy pdf-id gần nhất bằng cách lấy xác suất lớn nhất ở mỗi

frame đó

$$m_i = \arg \max_j a_{i,j}.$$

Với $\mathbf{m} = (m_1, m_2, \dots, m_n)$ là chuỗi pdf-id gần nhất của bài hát gốc $x(t)$. Giả sử g là hàm số biểu diễn quá trình dự đoán của mô hình DNN với giá trị đầu vào là đoạn âm thanh gốc ban đầu

$$g(x(t)) = \mathbf{m}.$$

Song song với quá trình trên, ta định danh âm vị của âm thanh lệnh mong muốn $y(t)$ từ đó ta thu được chuỗi pdf-id chính xác đối với âm thanh lệnh là $\mathbf{b} = (b_1, b_2, \dots, b_n)$. Để bài hát gốc sẽ được mô hình Kaldi giải mã dưới dạng lệnh mong muốn, ta phải xác định một lượng tối thiểu $\delta(t)$ thêm vào $x(t)$ để \mathbf{m} gần giống hay giống với \mathbf{b} , từ đó ta có hàm khoảng cách giữa \mathbf{m} và \mathbf{b} cần được tối thiểu

$$\arg \min_{\delta(t)} \|g(x(t) + \delta(t)) - \mathbf{b}\|.$$

Khi đó $x'(t) = x(t) + \delta(t)$ là mẫu đối kháng được dùng để tấn công ngược lại vào mô hình nhận diện giọng nói Kaldi.

Dữ liệu. Trong nghiên cứu này, các nhà nghiên cứu sử dụng các công cụ chuyển đổi văn bản thành giọng nói (text-to-speech - TTS), và các ghi âm giọng nói của con người để có các âm thanh giọng nói của các lệnh mong muốn mà nền tảng Kaldi có thể nhận diện chính xác được. Các bài hát làm sóng mang là 26 bài hát từ internet, với các thể loại nhạc như nhạc phổ biến, nhạc nhẹ, nhạc rock, nhạc rap. Và 12 lệnh được cân nhắc sử dụng như “turn on GP”, “ask Capital One to make a credit card payment”,... như được biểu thị trong bảng.

Kết quả. Trong cuộc tấn công, các nhà nghiên cứu cung cấp trực tiếp các bài hát đối kháng mang các lệnh ẩn đã tạo được cho Kaldi bằng cách sử

CHƯƠNG 3. MỘT SỐ NGHIÊN CỨU LIÊN QUAN

Command	Success rate (%)	SNR (dB)	Efficiency (frames/hours)
Okay google restart phone now.	100	18.6	229/1.3
Okay google flashlight on.	100	14.7	219/1.3
Okay google read mail.	100	15.5	217/1.5
Okay google clear notification.	100	14	260/1.2
Okay google good night.	100	15.6	193/1.3
Okay google airplane mode on.	100	16.9	219/1.1
Okay google turn on wireless hot spot.	100	14.7	280/1.6
Okay google read last sms from boss.	100	15.1	323/1.4
Echo open the front door.	100	17.2	193/1.0
Echo turn off the light.	100	17.3	347/1.5
Okay google call one one zero one one nine one two zero.	100	14.8	387/1.7
Echo ask capital one to make a credit card payment.	100	15.8	379/1.9

Bảng 3.3: Kết quả tấn công bằng CommanderSong (nguồn [1])

dụng API chấp nhận đầu vào là một âm thanh thô. Tổng cộng có được hơn 200 bài hát đối địch ở định dạng “WAV” được tạo ra bằng phương pháp “pdf-id matching” và gửi chúng trực tiếp đến Kaldi để được công nhận. Nếu Kaldi xác định thành công lệnh được đưa vào bên trong, biểu thị cuộc tấn công là thành công.

Bảng 3.3 cho thấy kết quả tấn công vào API. Mỗi lệnh có thể được Kaldi

CHƯƠNG 3. MỘT SỐ NGHIÊN CỨU LIÊN QUAN

nhận ra một cách chính xác 100%. Tỷ lệ thành công được tính bằng tỷ số giữa số từ được giải mã thành công và số từ trong lệnh mong muốn. Có nghĩa là Kaldi đã giải mã chính xác từng từ trong lệnh mong muốn.

Bên cạnh đó, các nhà nghiên cứu đã tính toán thêm trung bình của tỷ lệ tín hiệu nhiễu (signal-noise ratio - SNR) [29] so với bài hát gốc như được biểu thị trong bảng. Tỷ lệ tín hiệu nhiễu là một thông số được sử dụng rộng rãi để định lượng mức độ năng lượng giữa tín hiệu âm thanh gốc và tín hiệu nhiễu do tạp âm gây ra. Trong nghiên cứu này tỷ lệ tín hiệu nhiễu được dùng để đo độ méo giữa mẫu âm thanh đối kháng với bài hát gốc. Với $SNR(dB) = 10 \log_{10}(P_{x(t)}/P_{\delta(t)})$, trong đó $x(t)$ là bài hát gốc, $\delta(t)$ là âm thanh nhiễu, và P là công suất của các mẫu âm thanh. Giá trị SNR càng lớn thì cho thấy độ nhiễu loạn càng nhỏ và ngược lại. Dựa trên kết quả trong Bảng 3.3, SNR nằm trong khoảng $14 - 18.6dB$, với nhiễu trong bài hát gốc là dưới 4% quá ít để có thể nhận thấy được dễ dàng.

3.2.2. Devil's whisper

Kịch bản tấn công. Devil's whisper [2] dựa vào các thách thức kỹ thuật như CommanderSong [1], và nhận thấy các cuộc tấn công vào hộp đen bằng phương pháp thay đổi âm thanh để được các véc-tơ MFCC tương tự tạo ra các tiếng ồn mà còn người có thể nghi ngờ đó là cuộc tấn công vào các sản phẩm thông minh nhận diện giọng nói. Vì vậy với mục tiêu tấn công vào hộp đen nhưng vẫn dùng các yếu tố là lấy các bài hát trong CommanderSong làm sóng mang, khiến con người khó có thể nhận biết được.

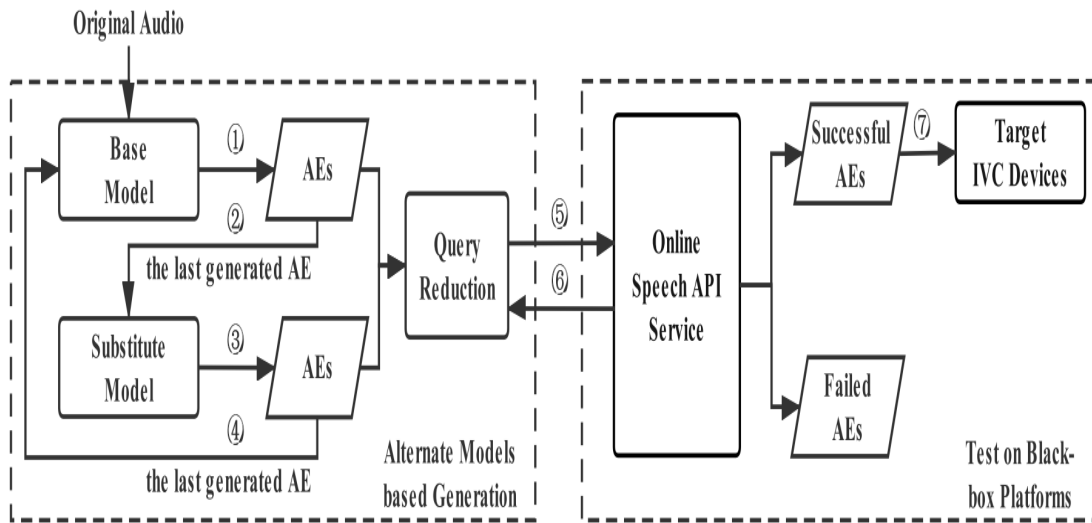
Vì mục tiêu là các thiết bị nhận diện giọng nói thương mại, đa phần là hộp đen các nhà nghiên cứu không có kiến thức nào về bên trong của hệ thống như các tham số hay các siêu tham số. Thay vào đó giả định rằng các thiết bị nhận diện giọng nói thương mại có những dịch vụ chuyển đổi giọng nói thành văn bản tương ứng, tức có thể lấy được kết quả giải mã thời gian thực từ các âm thanh đầu vào. Ví dụ ứng với Google Assistant ứng với dịch vụ Google Cloud Speech-to-Text, Microsoft Cortana ứng với dịch vụ Microsoft Bing Speech,...

Khi các âm thanh đối kháng được tạo ra, giả định các âm thanh này sẽ được phát bởi các loa (radio, TV, điện thoại thông minh, máy tính,...) được đặt không quá xa các thiết bị nhận diện giọng nói mà ta muốn tấn công. Các cuộc tấn công được xem là thành công nếu các thiết bị nhận diện giọng nói thực hiện lệnh đã ẩn trong âm thanh, ví dụ với lệnh “ok google navigate my home” đối với Google Assistant sẽ thực hiện điều hướng về nhà.

Cách tiến hành. Hình 3.4 cung cấp quá trình tiếp cận tấn công các mô hình hộp đen. Bằng cách tiếp cận trên khả năng chuyển nhượng của các mô hình, sử dụng mô hình hộp trắng (Base Model) tạo ra các mẫu âm thanh đối kháng tấn công vào các mô hình hộp đen (bước 1). Sau đó, thực hiện cách tiếp cận mới là tạo nên các mô hình thay thế (Substitute Model)

CHƯƠNG 3. MỘT SỐ NGHIÊN CỨU LIÊN QUAN

để tạo ra nhiều thể hệ mẫu âm thanh đối kháng (bước 2,3,4).



Hình 3.5: Các bước tạo mẫu đối kháng bằng Devil' whisper (nguồn [2])

Ở bước 1, ta không có kiến thức về mô hình hộp đen, vì vậy một phương pháp đơn giản là tạo ra các mẫu đối kháng trên mô hình hộp trắng (mà ở đây là mô hình Kaldi) và chuyển các mẫu đối kháng sang tấn công mô hình hộp đen mục tiêu. Sự thành công của khả năng tấn công chuyển nhượng là dựa trên sự tương đồng giữa cấu trúc của mô hình hộp đen và hộp trắng. Nghiên cứu gần đây chứng minh rằng khả năng chuyển giao có thể hoạt động trên các mô hình không đồng nhất thông qua việc cải tiến thuật toán tạo mẫu đối kháng [30].

Triển khai bước 1 dựa trên phương pháp chuyển nhượng, sử dụng mô hình Kaldi ASPIRE Chain làm mô hình hộp trắng chuyển nhượng (Base Model) và ý tưởng về thuật toán “pdf-id matching” được đề xuất trong CommanderSong và nâng cao khả năng chuyển giao của các mẫu đối kháng bằng cách áp dụng phương pháp biến đổi gradient nhanh lặp lại dựa trên động lượng (Momentum based Iterative Fast Gradient Method - MI-FGM). Phương pháp động lượng được giới thiệu trong Boosting adversarial attacks with momentum [31], có thể tích lũy một véc-tơ vận tốc theo hướng gradient

CHƯƠNG 3. MỘT SỐ NGHIÊN CỨU LIÊN QUAN

trong quá trình lặp lại. Trong mỗi lần lặp, gradient sẽ được lưu, sau đó được cộng dồn bằng cách sử dụng hệ số phân rã với các gradient đã lưu trước đó, giúp hướng gradient ổn định hơn và khả năng chuyển giao của các mẫu đối kháng cũng sẽ được tăng cường.

Với g_t là gradient tại lần lặp thứ t , g_0 là gradient ban đầu bằng 0, x_t^* là mẫu đối kháng được tạo ra tại lần lặp thứ t , x_0^* là mẫu âm thanh gốc ban đầu. Hàm $Clip_\epsilon$ là hàm cắt bớt các giá trị vượt quá giá trị ϵ cho trước. Khi đó MI-FGM với learning rate α tiến hành trên các hàm số

$$g_{t+1} = \mu g_t + \frac{\mathbf{J}(x_t^*, y)}{\nabla_x \mathbf{J}(x_t^*, y)},$$
$$x_{t+1}^* = x_t^* + Clip_\epsilon(\alpha g_t).$$

Song song với mô hình chuyển nhượng, các nhà nghiên cứu xây dựng thêm một mô hình thay thế gần giống với mô hình hộp đen mục tiêu. Vì mô hình thay thế này chỉ là một mô hình nhỏ không thể giống hoàn toàn với các mô hình hộp đen. Mô hình Kaldi có thể chuyển nhượng qua mô hình hộp đen với một mức độ nào đó, nên có thể xem đây là một mô hình cơ sở lớn sử dụng để tạo ra các mẫu đối kháng mang hầu hết các tính năng của lệnh mong muốn giúp nâng cao mô hình thay thế để tạo ra các mẫu đối kháng tốt hơn.

Trong hình 3.4 bước 2, các mẫu đối kháng được tạo từ mô hình cơ sở sẽ được gửi đến mô hình thay thế làm giá trị đầu vào và tiếp tục tạo ra các thể hệ mẫu đối kháng tiếp theo. Các mẫu đối kháng này sẽ mang thêm các tính năng riêng biệt gần giống với mô hình hộp đen thông qua mô hình thay thế. Song song với quá trình tạo các mẫu đối kháng ở cả hai mô hình cơ sở và mô hình thay thế, các mẫu đối kháng được tạo ra sẽ được gửi các lệnh truy vấn đến các dịch vụ nhận diện giọng nói ứng với mục tiêu cần tấn công.

Khi các mẫu đối kháng có thể truy vấn chính xác với các lệnh mục tiêu sẽ được lưu giữ lại, nếu không có một mẫu đối kháng nào có thể truy vấn

CHƯƠNG 3. MỘT SỐ NGHIÊN CỨU LIÊN QUAN

chính xác thì mẫu đối kháng cuối cùng trong quá trình tạo từ mô hình thay thế sẽ trở thành giá trị đầu vào cho mô hình cơ sở trong lần lập kế tiếp. Khi kết thúc số lần lập nhất định, ta sẽ chọn mẫu có hiệu quả cao nhất để tấn công các mô hình hộp đen mục tiêu ban đầu, nếu không có mẫu nào được tạo ra ta có thể kết luận ứng với câu lệnh và mô hình hộp đen mục tiêu không thể tạo ra mẫu đối kháng tương ứng với phương pháp này.

Dữ liệu. Trong nghiên cứu này, các nhà nghiên cứu đã chọn mô hình Mini Librispeech làm mô hình thay thế để ước lượng các mô hình mục tiêu. Để phong phú kho dữ liệu của mình, các nhà nghiên cứu đã sử dụng năm dịch vụ TTS để tổng hợp các âm thanh lệnh mong muốn như Google TTS, Alexa TTS, Bing TTS, IBM TTS, và một TTS không rõ tên cùng với 14 người nói gồm 6 nam và 8 nữ. Sau khi sử dụng các dịch vụ TTS ở trên để tạo các đoạn lệnh mong muốn, họ làm phong phú nó bằng cách thêm nhiều hoặc xoắn âm thanh. Thêm tiếng ồn trắng vào âm thanh gốc và đặt biên độ của tiếng ồn trắng thêm vào là α , thay đổi âm thanh gốc bằng cách thay đổi tốc độ giọng nói của chậm hơn hoặc nhanh hơn với tỷ lệ xoắn là β (β = thời gian âm thanh gốc/thời gian âm thanh đã xoắn). Sử dụng mô hình hộp đen mục tiêu để nhận ra âm thanh đã điều chỉnh và lọc nó dựa trên độ chính xác và mức độ hiệu quả của các kết quả được giải mã.

Vì mục tiêu của phương pháp tiếp cận của là tấn công các thiết bị IVC thương mại như Google Home, nên chỉ tập trung vào các lệnh đặc biệt thường được sử dụng trên các thiết bị này như “turn off the light”, “navigate to my home”,... Đối với mỗi mô hình mục tiêu, họ đã chọn 10 lệnh và thêm các từ đánh thức mặc định cho các hệ thống khác nhau.

Tương tự như CommanderSong, cuộc tấn công của chúng tôi sử dụng các bài hát làm sóng mang cho các lệnh ẩn, tạo các mẫu âm thanh đối kháng. Sử dụng tập dữ liệu được phát hành bởi dự án CommanderSong gồm 5 bài hát trong mỗi thể loại nhẹ nhàng, phổ biến, rock và rap. Trong số đó, họ

CHƯƠNG 3. MỘT SỐ NGHIÊN CỨU LIÊN QUAN

chọn các bài hát thuộc thể loại nhẹ nhàng và phổ biến, ít ồn ào hơn, hạn chế nhiều tích hợp nhiều khả năng lẫn át nhạc nền và được giải mã chính xác bởi các thiết bị IVC mục tiêu.

Kết quả. Các nhà nghiên cứu đánh giá hiệu quả của các mẫu âm thanh đối kháng được tạo ra bởi cách tiếp cận bằng mô hình chuyển nhượng (transferability based approach - TBA) và cả mẫu âm thanh đối kháng được tạo ra bởi cách tiếp cận bằng mô hình thay thế (alternate models generation approach - AGA) trên các API chuyển đổi giọng nói thành văn bản (speech-to-text - STT) và các thiết bị IVC thương mại.

Đối với các kiểu dịch vụ API STT trên đám mây của Google chỉ hiển thị kết quả của “phone_call model” và “command_and_search model”, vì trong quá trình thử nghiệm các nhà nghiên cứu đã nhận định rằng “phone_call model” tương tự như “video model” và “command_and_search model” tương tự “default model”. Hiệu quả của cách tiếp cận này được đánh giá bằng tỷ lệ các lệnh thành công (success rate of command - SROC), là tỷ lệ giữa số lệnh thành công và tổng số lệnh được đánh giá trên một mục tiêu.

Ở bảng, kết quả cho thấy các mẫu đối kháng do TBA tạo ra hoạt động tốt trên mô hình Google phone_call với tỷ lệ thành công là 100% nhưng lại không hiệu quả trên các mô hình khác. Đối với các mẫu đối kháng do AGA tạo ra có hoạt động tốt với tỷ lệ là 100% trên hầu hết các dịch vụ API ngoại trừ Amazon Transcribe. Các nhà nghiên cứu đã thực hiện nhiều thử nghiệm hơn trên Amazon Transcribe API và nhận thấy rằng dịch vụ API này thậm chí không thể nhận diện một số đoạn âm thanh TTS thuần túy cho các lệnh đích một cách chính xác.

Sau đó, chọn những mẫu đối kháng có thể tấn công hoàn toàn thành công dịch vụ API với kết quả cao (≥ 0.6) để tấn công các thiết bị IVC. Đặc biệt, vì các mẫu đối kháng hoạt động kém trên Amazon Transcribe API

CHƯƠNG 3. MỘT SỐ NGHIÊN CỨU LIÊN QUAN

Black -box	Google		Microsoft Bing	Amazon Transcribe	IBM STT
	Phone	Command			
TBA	10/10	0/10	2/10	1/10	3/10
AGA	10/10	10/10	10/10	4/10	10/10
SNR (dB)	11.97	9.39	13.36	11.21	10.06

Bảng 3.4: Kết quả tấn công trong nghiên cứu Devil's Whisper vào các dịch vụ API STT (nguồn [2])

Black -box	Google		Microsoft Cortana	Amazon Echo	IBM WAA
	Assistant	Home			
TBA	4/10	4/10	2/10	0/10	3/10
AGA	10/10	9/10	10/10	10/10	10/10
SNR (dB)	9.03	8.81	10.55	12.10	7.86

Bảng 3.5: Kết quả tấn công trong nghiên cứu Devil's Whisper vào các thiết bị IVC (nguồn [2])

không nhất thiết hoạt động kém trên Amazon Echo, nên vẫn kiểm tra trực tiếp các mẫu đối kháng trên Amazon Echo, ngay cả khi chúng không thành công trên Amazon Transcribe API.

Trong bảng, cho thấy cách tiếp cận được đề xuất rất hiệu quả trong việc tấn công các thiết bị IVC trong thế giới thực, luôn có thể xác định các mẫu đối kháng có thể tấn công thành công các thiết bị IVC tương ứng của các dịch vụ API đã bị tấn công. Tuy nhiên, Amazon Transcribe API và Amazon Echo là ngoại lệ.

4 Thiết kế nghiên cứu

Bài toán về tấn công đối kháng trên các mô hình học máy trên các mô hình nhận diện âm thanh cho việc phân loại hay chuyển đổi giọng nói thành văn bản đã được nghiên cứu trong nhiều năm gần đây cho tiếng Anh. Đã có nhiều bài báo thực hiện tấn công trên các mô hình hộp trắng, hay các mô hình hộp đen đã thương mại hóa và mang lại nhiều kết quả khả quan. Dựa trên các ý tưởng và các đề tài nghiên cứu đó, chúng tôi đặt ra bài toán tương tự trên những mô hình học máy hoạt động cho tiếng Việt. Trong luận văn này, chúng tôi giải quyết bài toán thực hiện tấn công đối kháng trên mô hình hộp trắng nhận diện phân loại một số câu lệnh thường dùng cho nhà thông minh.

4.1. Phát biểu bài toán

Bài toán được giải quyết trong luận văn này là một nền tảng cơ bản để có thể đánh giá, và phát triển các phương thức tấn công trên các mô hình hộp trắng nhận diện chuyển đổi giọng nói thành văn bản cho tiếng Việt. Để giải quyết bài toán và chứng minh tính khả thi của các cuộc tấn công đối kháng trên các mô hình đối với tiếng Việt chúng tôi cần có một mô hình có độ chính xác phân loại, và kháng nhiễu cao để thực hiện tấn công. Và các cuộc tấn công sẽ được thực hiện thông qua truyền trực tiếp các mẫu âm thanh vào các mô hình mục tiêu.

Để dễ hiểu hơn về bài toán chúng tôi đã mô hình hóa lại các bước thực hiện tấn công. Với một mô hình nhận diện phân loại giọng nói tiếng Việt $f(x)$ chúng tôi yêu cầu mô hình phải có độ chính xác cao trên 90% và khả năng nhận diện tốt với các nhiễu môi trường được thêm vào. Khi đó ta có với một mẫu dữ liệu mô hình đã nhận biết chính xác trước đó x với giá trị phân loại khi qua mô hình là $y = f(x)$. Chúng tôi sẽ thực hiện hai phương thức tấn công đối kháng là tấn công không mục tiêu, và tấn công có mục tiêu thông qua tính toán ước lượng δx để tạo mẫu dữ liệu mới $x' = x + \delta x$. Đối với cuộc tấn công không có mục tiêu chúng tôi đặt ra kết quả tốt sẽ là các mẫu dữ liệu mới sẽ làm cho mô hình nhận diện sai với giá trị ban đầu $f(x + \delta x) \neq y$. Và các đối với cuộc tấn công có mục tiêu chúng tôi đặt ra kết quả tốt sẽ là $f(x + \delta x) = y'$ với y' là mục tiêu lớp mà ta chỉ định. Ngoài ra chúng tôi còn đặt ra một đánh giá cho các mẫu tấn công ở cả hai phương thức đó là ước lượng được δx nhỏ để không làm ảnh hưởng lớn thay đổi mẫu âm thanh ban đầu. Hay nói cách khác các mẫu tấn công được tạo ra đối với tai người vẫn nghe rõ các câu lệnh của âm thanh gốc nhưng mô hình lại nhận diện và cho kết quả sai lệch.

4.2. Phân tích bài toán

4.2.1. Ngữ cảnh

Trong luận văn này, chúng tôi sẽ xây dựng một ngữ cảnh tấn công cơ bản và khá phổ biến trong các nghiên cứu tấn công đối kháng trên mô hình nhận diện giọng nói tiếng Anh, nhưng lại hoàn toàn mới đối tiếng Việt. Với vai trò nghiên cứu tấn công, chúng tôi đặt mình vào vị trí của những kẻ tấn công vào các mô hình học máy. Chúng tôi giả sử mình là những kẻ tấn công đã truy cập được vào được một hệ thống nhận diện phân loại giọng nói mục tiêu. Khi đó chúng tôi có thể xem được cấu trúc, thông số của mô hình, bên cạnh đó chúng tôi cũng có thể truy cập, tải về và chỉnh sửa các

dữ liệu dùng cho việc huấn luyện mô hình. Khi đó chúng tôi sẽ bắt đầu xây dựng một kịch bản để tạo ra cuộc tấn công né tránh (evasion attack) khiến các hệ thống học máy mục tiêu ban đầu nhận diện sai lệch theo mục tiêu chúng tôi hướng đến.

4.2.2. Kịch bản tấn công

Với các thông tin có đã được xác định từ mô hình học máy mục tiêu, chúng tôi xây dựng một mô hình bản sao tương đồng từ cấu trúc, thông số và tập dữ liệu huấn luyện. Sau đó, chúng tôi sử dụng mô hình bản sao để thực hiện các giải thuật tạo mẫu âm thanh đối kháng từ các mẫu âm thanh đã được nhận diện chính xác. Và cuối cùng sử dụng các mẫu âm thanh đối kháng vừa được tạo ra gửi đến mô hình mục tiêu ban đầu khiến mô hình nhận diện sai lệch so với mẫu âm thanh gốc.

4.3. Phương pháp đề xuất

Phương pháp đề xuất để giải quyết bài toán trên được dựa trên giải thuật biến đổi theo dấu của gradient có lặp lại (iterative fast gradient sign method - IFGSM) [8] với một số cải tiến về quá trình lựa chọn tham số và kết quả tạo mẫu.

4.3.1. Giải thuật IFGSM

Trong quá trình tạo mẫu đối kháng, bài toán mà chúng ta luôn phải giải quyết đó là ước lượng một lượng nhiễu thích hợp để thêm vào dữ liệu gốc ban đầu gây ra các phân loại sai lệch ở các mô hình học máy. Giải thuật IFGSM là một trong các thuật toán để giải quyết bài toán đó. Sử dụng ý tưởng cập nhật các giá trị tham số của mô hình trong quá trình huấn luyện

của thuật toán lan truyền ngược. Giải thuật IFGSM là một phương pháp cập nhật giá trị đầu vào x của mô hình để tối thiểu hóa hàm mất mát. Với hàm mất mát J , x_n là giá trị đầu vào của mô hình tại lần lặp thứ n , giải thuật IFGSM được lặp lại ba bước chính. Đầu tiên toán với giá trị mất mát $J(x, y_{mục\ tiêu})$ - là giá trị sai lệch giữa kết quả dự đoán của mô hình $y_{dự\ đoán}$ và kết quả mong muốn $y_{mục\ tiêu}$. Do đó, hàm mất mát J có thể được biểu diễn dưới dạng một hàm số với biến x . Sau đó, ta tính toán giá trị véc-tơ gradient của J theo x để xác định dấu của nhiễu cần thiết thêm vào. Cuối cùng, sử dụng véc-tơ dấu của gradient vừa tìm được nhân với một lượng ϵ được lựa chọn phù hợp để tạo ra các nhiễu cập nhật giá trị x ban đầu. Quá trình trên sẽ được lặp lại cho đến khi đạt đến số lần lặp nhất định hoặc tạo ra các mẫu đối kháng thỏa mãn điều kiện cho trước. Ta có thể biểu diễn giải thuật IFGSM theo công thức sau

$$x_n = x_{n-1} + \epsilon \cdot \text{sign}(\nabla_{x_{n-1}} J(x_{n-1}, y_{mục\ tiêu}))$$

Trong đó sign là hàm xác định dấu của gradient.

4.3.2. Cải tiến giải thuật IFGSM

Giải thuật IFGSM giải quyết bài toán tạo nhiễu khá nhanh và hiệu quả đối với hình ảnh, tuy nhiên đối với âm thanh các mẫu có sự khác nhau về âm lượng, các mẫu có các giá trị càng cao thì âm lượng càng lớn và ngược lại. Vì vậy trong giải thuật IFGSM việc lựa chọn một tham số ϵ phù hợp cho tất cả các mẫu âm thanh là một điều khó khăn. Với một số lần lặp nhất định, việc lựa chọn ϵ quá thấp sẽ làm cho nhiễu thêm vào có âm lượng khá nhỏ so với mẫu âm thanh gốc, khi đó không thể ảnh hưởng đến quá trình phân loại của mô hình. Mặt khác, nếu ta lựa chọn ϵ quá lớn âm lượng nhiễu sẽ lấn át hoàn toàn nội dung câu lệnh bên trong mẫu âm thanh ban đầu

khiến con người không còn nghe được nội câu lệnh ban đầu.

Mục tiêu luận văn này tạo ra các mẫu âm thanh đối kháng khiến cho mô hình nhận diện sai lệch, nhưng con người vẫn nhận ra nội dung gốc của các mẫu âm thanh. Vì vậy chúng tôi đề xuất cải tiến giải thuật IFGSM bằng cách sử dụng ϵ tùy biến ứng với từng mẫu âm thanh ban đầu, và kết hợp với việc giới hạn lượng nhiễu thêm vào để đảm bảo nội dung của các mẫu âm thanh. Cả hai giải thuật IFGSM và IFGSM cải tiến sẽ được chúng tôi trình bày rõ hơn trong việc ứng dụng vào quá trình tạo các mẫu âm thanh đối kháng ở Mục 6.

5 Hiện thực tấn công

5.1. Thu thập dữ liệu

Bộ dữ liệu một số câu lệnh tiếng Việt [32] với 15 câu loại câu lệnh khác nhau, của khoảng 250 người khác nhau được cân bằng trên các tiêu chí giới tính (nam và nữ), các nhóm độ tuổi (dưới 18, từ 18 đến 30, từ 30 đến 40 và từ 40 trở lên) và vùng miền (Bắc, Trung và Nam). Dữ liệu được thu thập thông qua một ứng dụng cá nhân do nhóm tác giả phát triển trên thiết bị điện thoại thông minh.

Bảng 5.1 thể hiện nội dung của 15 lớp khác nhau trong tập dữ liệu huấn luyện. Các câu lệnh trên là những câu lệnh phổ biến thường được dùng trong các mô hình nhà thông minh.

Thông qua tiếp xúc với dữ liệu cho thấy dữ liệu các mẫu âm thanh câu lệnh được ghi với tần số $8000Hz$. Mỗi giá trị trên miền thời gian của dữ liệu là một số thực 32 bit trong khoảng từ -1 đến 1. Do thực hiện ghi âm trên các thiết bị di động và trong các môi trường thực tế dẫn đến các mẫu âm thanh mang nhiều tạp âm do ảnh hưởng bởi phản ứng của thiết bị ghi âm và các âm thanh tiếng ồn môi trường xung quanh, các mẫu âm thanh có các độ dài và âm lượng khác nhau trên lệch rất lớn. Từ đó cần phải có cách biện pháp tiền xử lý âm thanh để đồng nhất các mẫu âm thanh huấn

Số thứ tự	Tiếng Việt	Tiếng Anh
1	Bật đèn	Turn on light
2	Tắt đèn	Turn off light
3	Bật điều hòa	Turn on air conditioner
4	Tắt điều hòa	Turn off air conditioner
5	Đô rê mon	Doraemon
6	Bật quạt	Turn on fan
7	Tắt quạt	Turn off fan
8	Bật tivi	Turn on TV
9	Tắt tivi	Turn off TV
10	Mở cửa	Open door
11	Đóng cửa	Close door
12	Khóa cửa	Lock door
13	Mở cổng	Open gate
14	Đóng cổng	Close gate
15	Khóa cổng	Lock gate

Bảng 5.1: Bảng mô tả nội dung các lớp trong tập huấn luyện (nguồn [32])

luyện mô hình, giúp mô hình mang lại độ hiệu quả cao.

5.2. Tiền xử lý dữ liệu

Qua quá trình đánh giá tập dữ liệu cho thấy bộ dữ liệu có nhiều nhiễu và số lượng còn hạn chế nên cần xử lý và tạo thêm để giúp cho mô hình huấn luyện có độ chính xác cao.

Đối với các nhiễu âm thanh do phần cứng của các thiết bị di động sẽ tạo ra các nhiễu liên tục tần số cao. Bên cạnh đó giọng nói của người được xác định trong khoảng mức tần số từ $300Hz$ đến $3400Hz$. Vậy khi đó ta cho

các mẫu âm thanh qua một bộ lọc băng tần với giới hạn cận dưới là $300Hz$ và cận trên là $3400Hz$. Khi đó ta được các mẫu âm thanh có cùng độ dài với các mẫu âm thanh ban đầu nhưng những khoảng nhiễu đã bị loại bỏ những tần số gây nhiễu không thuộc dải băng tần trên. Các mẫu âm thanh mới vẫn có các độ dài khác nhau do chứa các khoảng im lặng không cần thiết. Ta sẽ loại bỏ các khoảng im lặng đó để lấy được các mẫu âm thanh chỉ mang nội dung câu lệnh.

Sau quá trình lọc nhiễu và loại bỏ các khoảng im lặng chúng tôi thu được các mẫu âm thanh chỉ mang nội dung câu lệnh có độ dài tối đa khoảng 1.5 giây. Khi đó chúng tôi tăng cường tập dữ liệu ban đầu bằng thêm các loại nhiễu tần số cao như tiếng ồn trắng và 15 loại nhiễu môi trường thường xảy ra trong cuộc sống hằng ngày như tiếng chim hót, tiếng xe cộ, tiếng gió,... được chúng tôi thu thập qua internet. Tuy nhiên các nhiễu có giá trị âm lượng khác nhau có thể khó ảnh hưởng đến các mẫu âm thanh gốc trong tập dữ liệu. Nếu mẫu âm thanh gốc có âm lượng quá lớn so với các mẫu âm thanh nhiễu sẽ làm cho mẫu dữ liệu mới sau khi thêm nhiễu sẽ không khác mẫu âm thanh gốc nhiễu. Ngược lại nếu mẫu âm thanh gốc lại có âm lượng quá nhỏ so với nhiễu sẽ dẫn đến sau khi thêm nhiễu thì mẫu âm thanh mới mất đi nội dung câu lệnh mà chỉ có thể nghe được nhiễu. Cả hai trường hợp trên đều dẫn đến việc làm mất tính chính xác và cân bằng của tập dữ liệu huấn luyện cho mô hình dẫn đến kết quả mô hình không mang hiệu quả cao. Vì vậy chúng tôi sử dụng phương pháp biến đổi nhiễu bằng cách nhân với tỷ lệ biên độ được tính thông qua tỷ lệ độ nhiễu so với âm thanh gốc (signal-to-noise ratio - SNR) [29]. Với A là trung bình bình phương biên độ của một tính hiệu sóng x ta có cách tính tỷ lệ thích hợp để thêm nhiễu phù hợp theo mong muốn như sau

Với

$$SNR_{dB} = 10 \log_{10} \left(\frac{A_{\text{âm thanh gốc}}^2}{A_{\text{nhiều mong muốn}}^2} \right).$$

Ta có

$$A_{\text{nhiều mong muốn}} = \sqrt{\frac{A_{\text{âm thanh gốc}}^2}{10^{SNR_{dB}/10}}}.$$

Trong đó cách tính $A_{\text{âm thanh gốc}}$ được biểu diễn như sau

$$A_{\text{âm thanh gốc}} = \sqrt{\left(\frac{\sum_{t=0}^T x_t^2}{T}\right)}.$$

Ta có tỷ lệ điều chỉnh âm lượng âm thanh nhiều với $A_{\text{nhiều ban đầu}}$ được tính tương tự $A_{\text{âm thanh gốc}}$ là

$$\epsilon = \frac{A_{\text{nhiều mong muốn}}}{A_{\text{nhiều ban đầu}}}.$$

Khi đó

$$x_{\text{nhiều mong muốn}} = x_{\text{nhiều ban đầu}} * \epsilon.$$

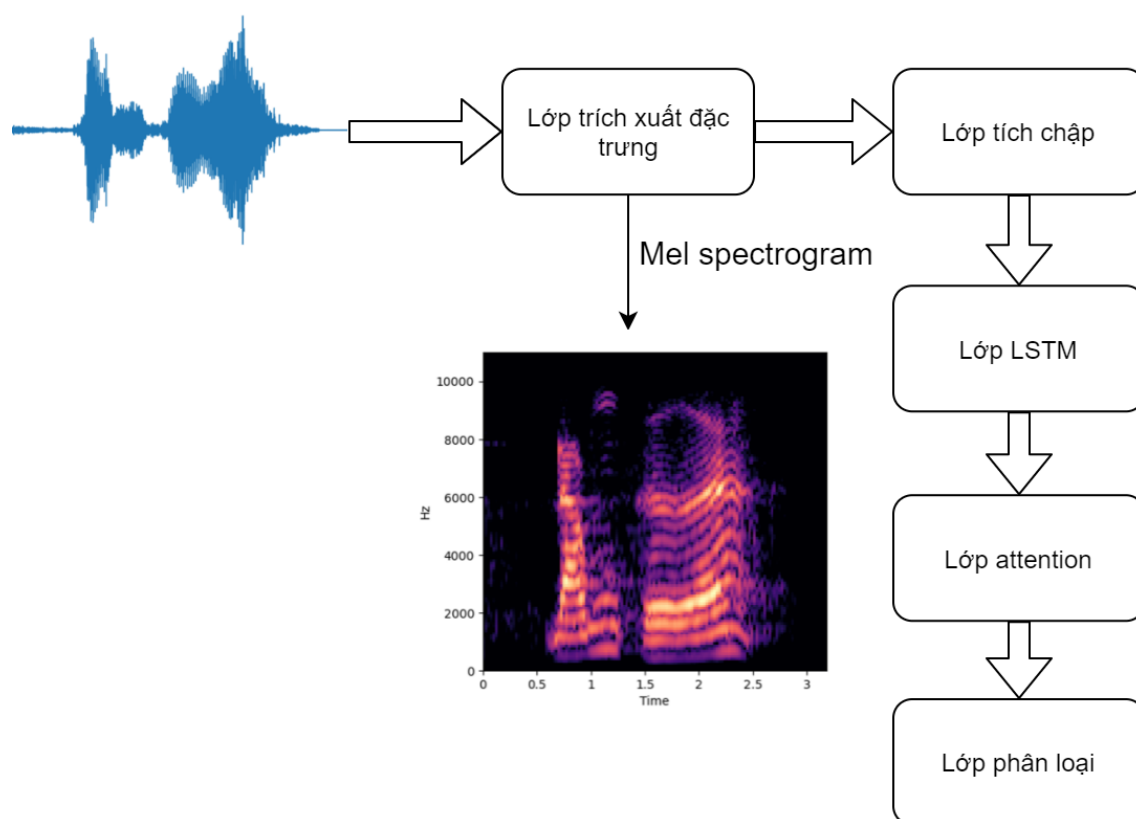
Sau quá trình tăng cường dữ liệu bằng cách thêm nhiễu, các mẫu âm thanh trong tập huấn luyện cần có sự chuẩn hóa về cùng một thời lượng giúp việc trích xuất các đặc trưng được sử dụng trong quá trình huấn luyện được dễ dàng hơn. Để đồng nhất về thời lượng của các mẫu dữ liệu và tăng thêm dữ liệu mới trong quá trình huấn luyện, chúng tôi sử dụng một lớp sinh dữ liệu. Lớp sinh dữ liệu sẽ được thiết lập tạo ra các tập dữ liệu mới khác nhau dựa vào bộ tập dữ liệu gồm các mẫu âm thanh gốc và các mẫu đã được chúng tôi thêm nhiễu. Về thời lượng chúng tôi đồng nhất các mẫu với thời gian 2 giây, ứng với tần số $8000Hz$ thì ta có mỗi mẫu dữ liệu mới sẽ là một véc-tơ 16000 chiều. Để tăng cường khả năng chú ý vào các trọng tâm nội dung câu lệnh ở các mẫu dữ liệu huấn luyện chúng tôi gán ngẫu nhiên cho véc-tơ 16000 giá trị trong khoảng từ -1 đến 1. Và sau đó chọn ngẫu nhiên một vùng dữ liệu phù hợp để thay thế thành các giá trị của mẫu dữ liệu huấn luyện. Do đó với mỗi lần lặp lại quá trình huấn luyện dữ liệu sẽ được tạo ra bởi các tập dữ liệu ban đầu nằm ở các vị trí khác nhau và bao quanh bởi các giá trị nhiễu. Từ đó giúp ta đánh giá đúng hơn về khả năng nhận diện, phân loại và kháng nhiễu của mô hình mục tiêu.

5.3. Mô hình thực nghiệm tấn công

5.3.1. Cấu trúc mô hình

Để chứng minh tính khả thi của các cuộc tấn công đối kháng trên mô hình nhận diện phân loại giọng nói chúng tôi lựa chọn mô hình có độ chính xác cao và số lượng hệ số thấp. Vì mô hình với các tiêu chí như vậy có khả năng cao sẽ được áp dụng vào các ứng dụng thực tế.

Qua quá trình tìm hiểu chúng tôi sử dụng cấu trúc mô hình học máy kết hợp cơ chế attention được giới thiệu bởi Douglas và đồng nghiệp [33]. Mô hình có cấu trúc gồm các lớp đầu vào là các véc-tơ 16000 chiều ứng với 2 giây dữ liệu âm thanh tần số $8000Hz$. Sau đó các giá trị đầu vào sẽ đi qua lớp trích xuất đặc trưng Mel-spectrogram của âm thanh với 2048 điểm biến đổi Fourier nhanh (1024 điểm biến đổi Fourier rời rạc đối với bài báo gốc), 80 giá trị tỷ lệ Mel, và khoảng dịch của mỗi cửa sổ là 128 điểm. Sau khi thu được Mel-spectrogram, các lớp tích chập tiếp theo sẽ nhận các Mel-spectrogram ấy và thực hiện các phép tích chập một chiều trên miền thời gian. Do đó các giá trị đầu ra của các lớp tích chập này sẽ tổng hợp các giá trị đặc trưng nhưng vẫn giữ nguyên mối quan hệ về mặt thời gian của Mel-spectrogram. Các giá trị đầu ra của các lớp tích chập sẽ là đầu vào cho các lớp LSTM hai chiều (bidirectional LSTM) để tổng hợp mối quan hệ liên tục theo thời gian của dữ liệu. Vì các mẫu dữ liệu chỉ mang một phần nhỏ nội dung cần thiết, những phần còn lại có thể là các giá trị gây nhiễu hoặc khoảng im lặng. Nên mô hình được tích hợp thêm cơ chế attention sử dụng một lớp dense để mô hình tự học các trọng số chú ý từ các giá trị đầu ra của lớp LSTM hai chiều. Nhờ các trọng số tự học đó mà mô hình có thể nhận diện nội dung trong nhiều một các chính xác với hiệu quả cao. Cuối cùng các đặc trưng được tổng hợp lại thông qua các lớp liên kết hoàn toàn và trả về giá trị phân loại bởi lớp phân loại softmax cuối cùng.



Hình 5.1: Cấu trúc mô hình mục tiêu (nguồn [33])

Hình 5.1 thể hiện cấu trúc mô hình chúng tôi lựa chọn để thực hiện tấn công đối kháng.

5.3.2. Hiệu năng mô hình

Đối với cấu trúc mô hình trên trong bài báo gốc, nhóm tác giả đã cho thấy mô hình có hiệu quả phân loại cao trong việc phân loại các câu lệnh từ tập dữ liệu “google speech commands”, mô hình có độ chính xác lên đến 96% và có số lượng tham số mô hình khá nhỏ.

Mô hình	Độ chính xác (%)	Tham số mô hình
res15	95.8	238K
res26	95.2	438K
res8	94.1	110K
ConvNet on raw WAV	89.4	700K
DS-CNN	95.4	498K
Attention RNN (nhóm tác giả)	96.9	202K

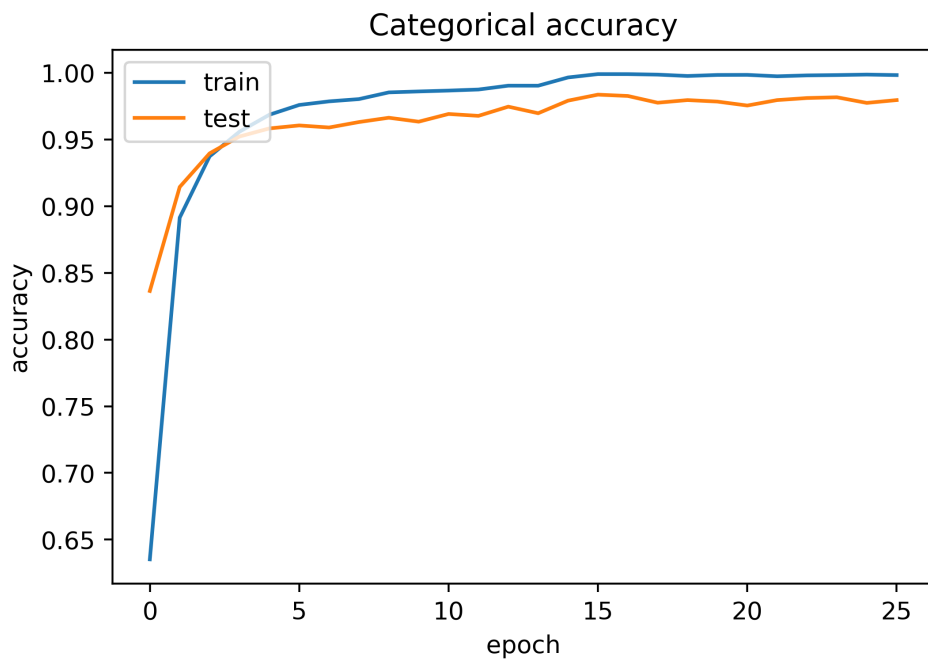
Bảng 5.2: Bảng so sánh kết quả của mô hình mục tiêu với một số mô hình khác trên tập dữ liệu “google speech commands” (nguồn [33])

Bảng 5.2 thể hiện độ chính xác cao của mô hình mục tiêu mà chúng tôi lựa chọn để thực hiện tấn công trên tập dữ liệu giọng nói tiếng Anh “google speech commands”.

Với cấu trúc mô hình và tập dữ liệu một số câu lệnh tiếng Việt, chúng tôi bắt đầu huấn luyện mô hình phân biệt các lớp câu lệnh trên. Sử dụng giải thuật “adam” [34] trong quá trình huấn luyện mô hình, với tỷ lệ học khởi tạo là 0.001 và giảm 60% sau mỗi 10 lần huấn luyện. Sử dụng kích thước batch là 64, và chạy trên Tesla P100 GPU của colab với mỗi lần huấn luyện khoảng 30 giây. Tập huấn luyện được chúng tôi thêm nhiều lớp nhiễu khác nhau như lớp các âm thanh tiếng chim hót, tiếng xe cộ và nhiều loại âm khác trong quá trình tăng cường dữ liệu, kết hợp với lớp nhiễu ngẫu nhiên được tạo ra trong quá trình chuẩn hóa dữ liệu. Vì vậy mô hình có

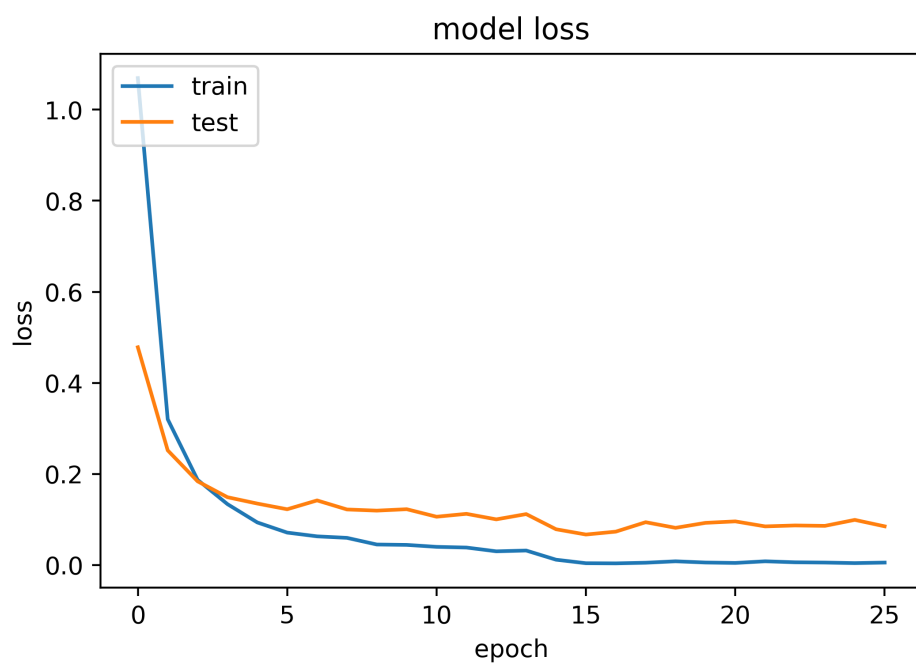
độ chính xác cao sẽ cho thấy khả năng kháng nhiễu của mô hình được đảm bảo. Không ngoài mong đợi mô hình huấn luyện có độ chính xác trong việc phân loại các câu lệnh tiếng Việt rất cao lên đến 98% và giá trị hàm mất mát là 0.1.

Với tập dữ liệu ban đầu chúng tôi chia thành 3 tập con gồm 70% cho tập huấn luyện, 10% tập kiểm thử để giúp kiểm tra trong quá trình huấn luyện mô hình có bị quá khớp hay không khớp hay không, và 20% còn lại cho tập kiểm định lại mô hình sau cùng.

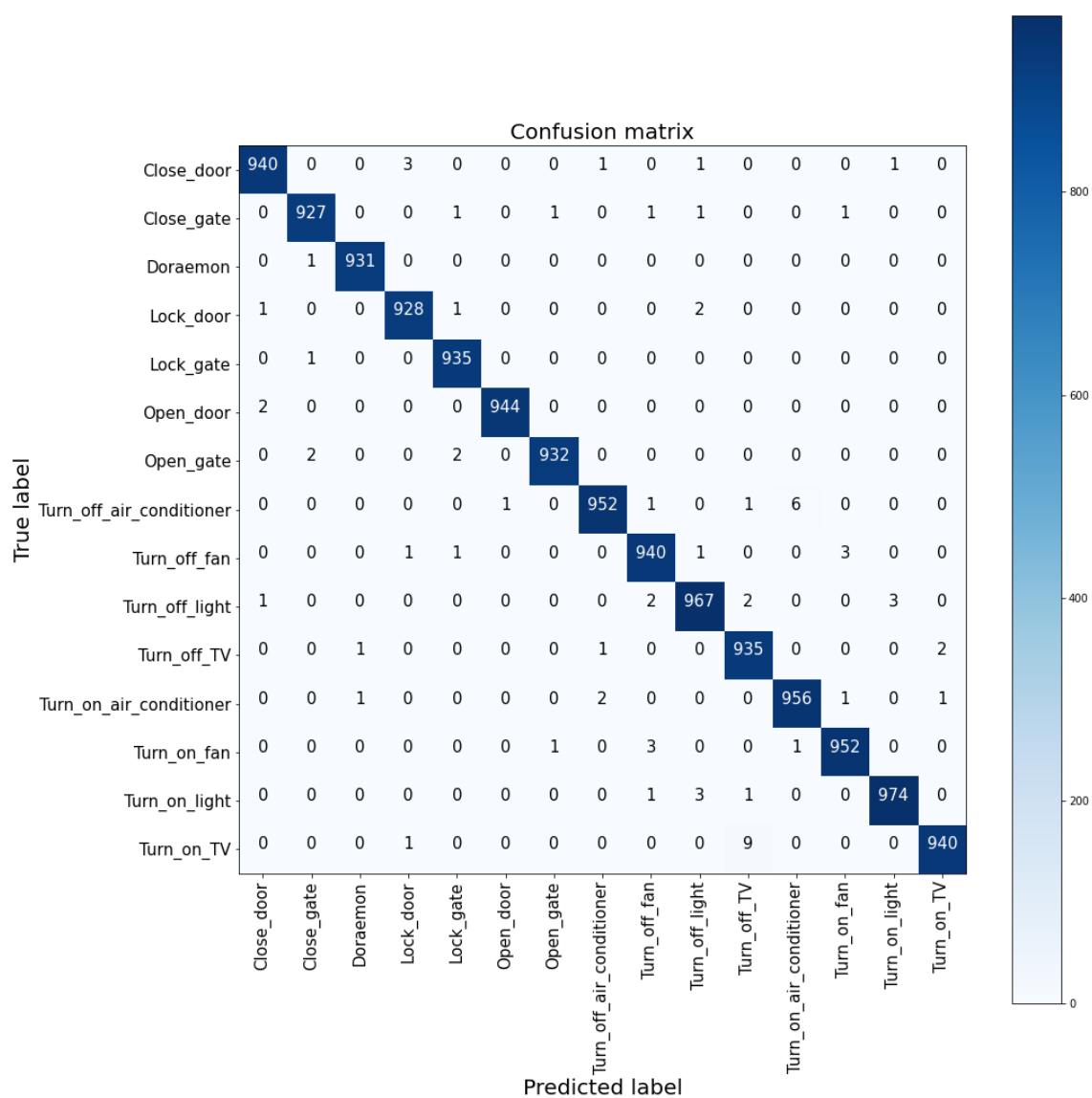


Hình 5.2: Biểu đồ đường thể hiện độ chính xác của mô hình

Các Hình 5.2, 5.3, 5.4 thể hiện độ chính xác cao trong việc phân loại các câu lệnh trong tập dữ liệu một số câu lệnh tiếng Việt. Vì vậy chúng tôi quyết định sử dụng mô hình đã được huấn luyện làm mục tiêu thực nghiệm các phương pháp tấn công của chúng tôi. Từ đó chứng minh dù một mô hình có độ chính xác, và khả năng kháng nhiễu cao nhưng vẫn khó kháng lại các dữ liệu gây nhiễu mới và làm sai lệch khả năng phân loại của mô hình.



Hình 5.3: Biểu đồ đường thể hiện giá trị mất mát của mô hình



Hình 5.4: Ma trận tương quan thể hiện dự đoán của mô hình trên tập kiểm định

5.4. Hiện thực giải thuật

5.4.1. Ngôn ngữ lập trình và thư viện

Chúng tôi sử dụng *Python* là ngôn ngữ lập trình chính cho toàn bộ hệ thống tấn công. Với khả năng tính toán mạnh mẽ và cộng đồng phát triển mạnh mẽ, vì vậy *Python* có thể được xem là một ngôn ngữ phổ biến trong nghiên cứu khoa học dữ liệu. Trong luận văn này, *Python* được sử dụng để thực hiện các bước tiền xử lý dữ liệu, tính toán, lưu trữ và hiện thực thuật toán tấn công IFGSM và IFGSM cải tiến. Tất cả mã nguồn và dữ liệu của chúng tôi đã được lưu trữ trên GitHub ¹.

Để hiện thực các giải thuật tấn công được đề xuất ở Chương 4, chúng tôi có sử dụng một vài thư viện để hỗ trợ quá trình hiện thực. Một số thư viện mà chúng tôi sử dụng trong luận văn là

- **Matplotlib** là một thư viện vẽ sơ đồ của *Python*, có thể biểu diễn trực quan dữ liệu dưới nhiều dạng hình vẽ khác nhau.
- **Numpy** là một thư viện hỗ trợ trong các bài toán đại số, hỗ trợ các ma trận lớn, đa chiều cùng với các hàm tính toán cấp cao. Trong luận văn này, chúng tôi sử dụng Numpy hỗ trợ quá trình tiền xử lý dữ liệu.
- **Tensorflow, Keras, Kapre** là ba thư viện hỗ trợ trong việc xây dựng các lớp nơ-ron trong các mô hình học máy. Trong luận văn này, chúng tôi sử dụng các thư viện này để xây dựng và huấn luyện lại một mô hình bản sao của mô hình mục tiêu và thực hiện các giải thuật tấn công.

¹<https://github.com/huynghuynitcs99/Adversarial-attack-Vietnamese-speech>

5.4.2. Hiện thực tấn công

Mô đun Tiền xử lý dữ liệu Nhận dữ liệu thô đầu vào, chuẩn hóa và sử dụng dữ liệu đã được chuẩn hóa để huấn luyện mô hình. Dữ liệu âm thanh ban đầu đều có dạng mảng với các độ dài khác nhau, khi đó việc sử dụng dữ liệu để huấn luyện sẽ rất khó khăn. Vì vậy, chúng tôi xây dựng một mô đun giúp các mẫu âm thanh sẽ được chuẩn hóa về độ dài đồng nhất, bên cạnh đó mô đun cũng sinh ra thêm dữ liệu ngẫu nhiên giúp việc huấn luyện đa dạng hơn. Với mỗi dữ liệu thô ban đầu, chúng tôi tạo một mảng có độ dài lớn cố định có thể chứa đoạn âm thanh có độ dài lớn nhất, và các giá trị của mảng sẽ được khởi tạo ngẫu nhiên các giá trị thực từ -1 đến 1. Sau đó chúng tôi lựa chọn một khoảng phù hợp ngẫu nhiên để thay thế thành các giá trị của mẫu âm thanh ban đầu. Dữ liệu sau khi qua mô đun các lần khác nhau sẽ tạo ra các dữ liệu mới khác nhau nhưng vẫn mang nội dung âm thanh thô ban đầu.

Mô đun Huấn luyện mô hình bản sao là một mô đun nhận dữ liệu từ mô đun tiền xử lý dữ liệu để bắt đầu huấn luyện mô hình đã được xây dựng. Mô đun sẽ gọi nhiều lần mô đun tiền xử lý dữ liệu để được các dữ liệu huấn luyện khác nhau sử dụng cho quá trình huấn luyện mô hình. Mô đun sẽ trả về một mô hình đã được huấn luyện.

Mô đun Tạo mẫu tấn công đối kháng là một mô đun nhận vào mô hình đã huấn luyện, mẫu dữ liệu âm thanh ban đầu, và một số hệ số liên quan như số lần lặp, hệ số ϵ , hệ số α . Mô đun này sẽ thực hiện giải thuật IFGSM và IFGSM cải tiến tùy vào các tham số truyền vào. Kết quả dữ liệu trả về của mô đun là một mẫu tấn công đối kháng có thể làm sai lệch phân loại của mô hình, hoặc một giá trị “Null” nếu không thể tạo ra một mẫu có thể làm sai lệch.

Mô đun **Hiện thị trực quan kết quả** dùng để hiển thị trực quan số lượng các mẫu tấn công đối kháng được tạo thành công. Sử dụng thư viện Matplotlib để hiển thị một ma trận tương quan giữa lớp chính xác và lớp được chúng tôi khiến mô hình nhận diện sai lệch.

6 Thực nghiệm và đánh giá kết quả

6.1. Quá trình tạo mẫu âm thanh đối kháng

6.1.1. Tấn công cơ bản

Sau khi có được mô hình mục tiêu, chúng tôi bắt đầu thực hiện các bước tấn công. Chúng tôi lựa chọn một hàm mất mát thích hợp để tính toán giá trị mất mát giữa giá trị đầu ra của mô hình và giá trị mục tiêu mong muốn. Trong cuộc tấn công này chúng tôi lựa chọn hàm sparse categorical crossentropy, hàm mất mát tương ứng với hàm mất mát được sử dụng trong quá trình huấn luyện mô hình. Với lớp phân loại cuối cùng

$$L(y_{\text{dự đoán}}, y_{\text{mục tiêu}}) = -y_{\text{mục tiêu}} \log(y_{\text{dự đoán}}).$$

Do lớp phân loại cuối cùng của mô hình là lớp phân loại xác suất softmax. Nên $y_{\text{mục tiêu}}$ là xác suất của lớp phân loại mà chúng tôi mong muốn, và sẽ luôn có giá trị là 1, còn $y_{\text{dự đoán}}$ sẽ là giá trị xác suất mô hình dự đoán tại lớp mục tiêu. Để thuận tiện chúng tôi có thể biểu diễn kết hợp mô hình và hàm mất mát thành một hàm mất mát

$$J(x, y_{\text{mục tiêu}}) = L(y_{\text{dự đoán}}, y_{\text{mục tiêu}}).$$

Khi đó chúng tôi dựa vào hàm $J(x, y_{\text{mục tiêu}})$ để tính toán gradient của

CHƯƠNG 6. THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

hàm đó đó theo giá trị x ban đầu. Với giá trị gradient tìm được, chúng tôi sử dụng một số phương pháp xác định lượng δx cần thêm vào x ban đầu để làm cho mô hình mục tiêu nhận diện sai theo mục đích của chúng tôi.

Chúng tôi lựa chọn phương pháp cơ bản để xác định lượng δx và cập nhật x là dựa vào dấu của gradient lặp lại (iterative fast gradient sign method - IFGSM) [8]. Khi đó δx được xác định bởi công thức sau

$$\begin{aligned}\delta x_{n-1} &= \epsilon \cdot \text{sign}(\nabla_x J(x_{n-1}, y_{\text{mục tiêu}})), \\ x_n &= x_{n-1} + \delta x_{n-1}.\end{aligned}$$

Trong đó sign là hàm xác định dấu của gradient hàm mất mát, x_n là mẫu tấn công được tạo ra tại lần lặp thứ n , $x_0 = x$, và ϵ là một thông số được lựa chọn phù hợp. Sau đây chúng tôi sẽ giới thiệu hai cách tấn công mô hình với việc lựa chọn ϵ ngược dấu nhau, đó là tấn công có mục tiêu và tấn công không mục tiêu.

Tấn công có mục tiêu. Tấn công có mục tiêu ở đây dễ hiểu là với một mẫu âm thanh x đã được mô hình phân loại chính xác vào một lớp cố định. Khi đó chúng tôi sẽ làm cho mô hình mục tiêu nhận diện phân loại mẫu âm thanh x_n được tạo nên từ x sai lệch vào lớp mà chúng tôi chỉ định.

Như chúng ta đã biết quá trình học của mô hình học máy là một quá trình cập nhật các tham số của mô hình thông qua quá trình lan truyền ngược. Trong quá trình ấy các tham số được cập nhật dựa trên gradient của hàm mất mát theo từng biến tham số của mô hình. Giả sử mô hình ban đầu có các tham số là θ , tỷ lệ học của mô hình là lr , và hàm mất mát sử dụng là L . Khi đó mô hình cần phải tối ưu hàm mất mát L theo các tham số θ về giá trị cực tiểu nên ta có cách cập nhật θ như sau

$$\theta' = \theta - lr \nabla_{\theta} L.$$

Đối với tấn công có mục tiêu, chúng tôi nhận thấy bài toán có sự tương

CHƯƠNG 6. THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

đồng với quá trình lan truyền ngược của mô hình trong quá trình huấn luyện. Nhưng điểm khác đó là thay vì chúng tôi cập nhật tham số mô hình, thì chúng tôi sẽ giữ nguyên chúng, thay vào đó chúng tôi sẽ hướng đến cập nhật giá trị đầu vào ban đầu x . Vì vậy, chúng tôi có thể lựa chọn một giá trị ϵ âm cập nhật giá trị x ngược chiều gradient hoặc có thể biểu diễn với một ϵ dương như công thức sau

$$\begin{aligned}x_n &= x_{n-1} - \delta x_{n-1} \\ &= x_{n-1} - \epsilon \cdot \text{sign}(\nabla_x J(x_{n-1}, y_{\text{mục tiêu}})).\end{aligned}$$

Tấn công không mục tiêu. Ngược lại với tấn công có mục tiêu, tấn công không mục tiêu chúng tôi chỉ đặt ra yêu cầu cuộc tấn công khiến cho mô hình nhận diện sai lệch với lớp dữ liệu đã được phân loại chính xác. Khi đó chúng tôi sẽ giải quyết bài toán ngược lại với tấn công có mục tiêu, là tối đa hóa giá trị hàm mất mát với $y_{\text{mục tiêu}}$ là lớp phân loại $y_{\text{dự đoán}}$ mô hình đã nhận diện chính xác ban đầu. Hay nói cách khác chúng tôi sẽ đi tìm giá trị x_n sao cho giá trị hàm mất mát J càng lớn càng tốt. Vì vậy x_n sẽ được tính ngược lại với tấn công có mục tiêu, với ϵ dương ta có công thức cho quá trình cập nhật giá trị x như sau

$$\begin{aligned}x_n &= x_{n-1} + \delta x_{n-1} \\ &= x_{n-1} + \epsilon \cdot \text{sign}(\nabla_x J(x_{n-1}, y_{\text{dự đoán}})).\end{aligned}$$

6.1.2. Cải tiến tấn công

Cải tiến ϵ . Như chúng tôi đã giới thiệu ở phần trên, việc cập nhật mẫu âm thanh x ở cả hai cách tấn công đều phụ thuộc vào việc lựa chọn phù hợp thông số ϵ . Không giống như hình ảnh, âm thanh là một hàm sóng liên tục theo thời gian, giá trị tại mỗi điểm của hàm sóng đó có thể được biểu diễn là một số thực từ -1 đến 1. Với hai hàm sóng có cùng hình dạng nhưng giá trị có tỷ lệ trên lệch nhau sẽ gây ra sự khác nhau về âm lượng. Âm lượng của một mẫu âm thanh là một yếu tố quan trọng trong quá trình nhận biết của mô hình, và cả quá trình thêm nhiễu để tấn công mô hình. Tại phần tăng cường dữ liệu để huấn luyện mô hình, chúng tôi đã giới thiệu việc ảnh hưởng của âm lượng các mẫu âm thanh gốc sẽ ảnh hưởng đến quá trình huấn luyện như thế nào. Việc thêm nhiễu vào các mẫu âm thanh gốc để tạo ra các mẫu âm thanh tấn công đối kháng cũng vậy. Mỗi mẫu âm thanh khác nhau sẽ có âm lượng khác nhau, sẽ rất khó để có thể xác định một lượng ϵ hợp lý cho tất cả mẫu âm thanh. Nếu lượng ϵ quá nhỏ việc cập nhật mẫu âm thanh x sẽ rất khó khăn, ngược lại ϵ quá lớn sẽ làm mất đi nội dung câu lệnh của mẫu âm thanh ban đầu.

Vì vậy, chúng tôi đề xuất sử dụng phương pháp tính ϵ ứng với từng mẫu âm thanh gốc thông qua SNR đã được chúng tôi giới thiệu ở Mục 5.2 như sau

$$A_{\text{nhiều mong muốn}} = \sqrt{\frac{A_{\text{âm thanh gốc}}^2}{10^{SNR_{dB}/10}}}.$$

Vì vậy

$$\epsilon = \frac{A_{\text{nhiều mong muốn}}}{A_{\text{nhiều ban đầu}}}.$$

Khi đó chúng tôi lựa chọn $A_{\text{nhiều ban đầu}} = 1$ sẽ đảm bảo được khi tạo các mẫu âm thanh tấn công, các nhiễu thêm vào sẽ có ảnh hưởng đến mô hình mục tiêu và làm sai lệch giá trị phân loại.

Cải tiến bảo toàn nội dung âm thanh. Phương pháp cải tiến trên đã đảm bảo tính ảnh hưởng của nhiễu đối với mô hình, nhưng còn một yêu cầu đặt ra đó là mẫu âm thanh mới vẫn phải đảm bảo về nội dung của mẫu âm thanh gốc ban đầu. Hay nói cách khác khi phát hai mẫu âm thanh tần công và âm thanh gốc thì tai người vẫn sẽ nhận diện là cùng nội dung câu lệnh và cùng người nói.

Để đảm bảo yêu cầu trên, nhiễu thêm vào sẽ không ảnh hưởng quá lớn đối với mẫu âm thanh gốc. Chúng tôi đề xuất sử dụng kết hợp với chiến thuật cắt tỉa giá trị nhiễu dựa vào một ngưỡng từ $-\alpha$ đến α nhất định theo công thức sau

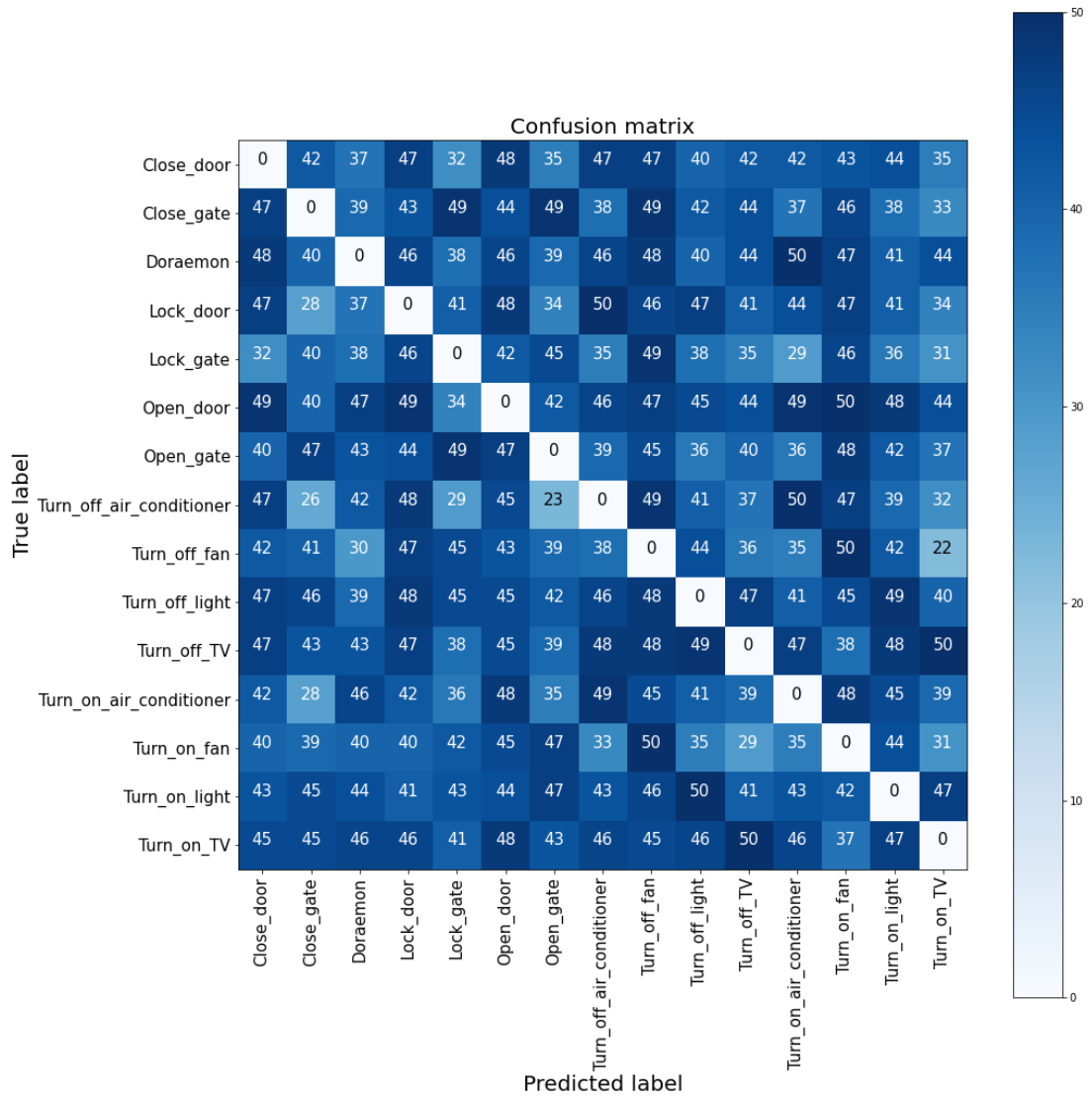
$$\Delta = \text{Clip}_\alpha(x_0 - x_n)$$

$$x_n = x_0 - \Delta$$

Trong đó Clip_α là hàm giới hạn lại các giá trị vượt qua ngưỡng $\pm\alpha$ về ngưỡng $\pm\alpha$. Khi đó, các mẫu âm thanh đối kháng được tạo ra sẽ mang lượng nhiễu được giới hạn so với âm thanh gốc ban đầu. Qua quá trình thực nghiệm, chúng tôi cũng áp dụng việc tính toán SNR để lựa chọn giá trị ngưỡng α phù hợp.

6.2. Đánh giá hiệu quả các mẫu

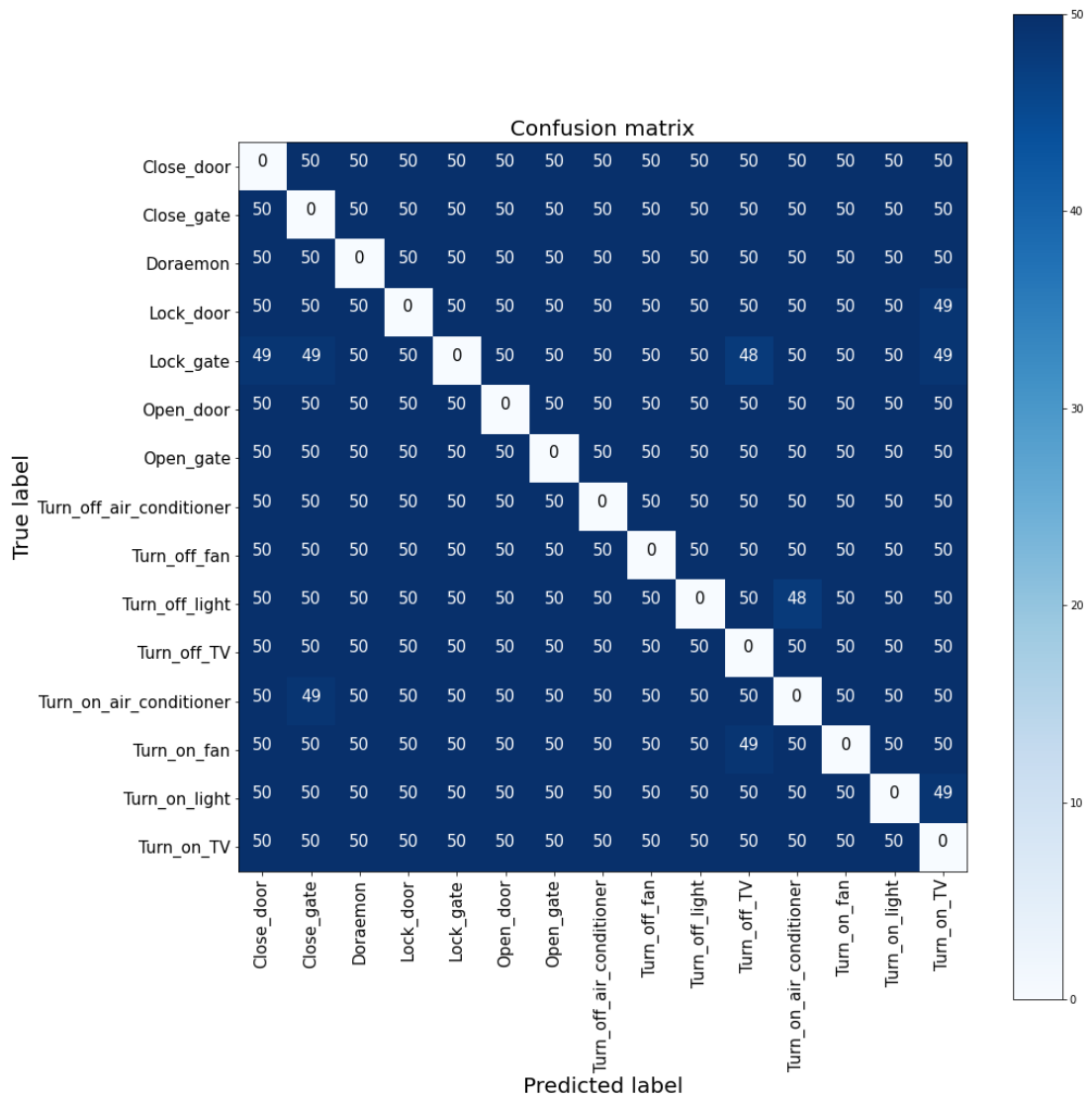
6.2.1. Tấn công có mục tiêu



Hình 6.1: Ma trận kết quả tấn công có mục tiêu dùng phương pháp cơ bản với $\epsilon = 10/2^{15}$

Với các Hình 6.1, 6.2, 6.3 chúng tôi thể hiện trên mỗi ô là số lượng các mẫu tấn công đối kháng được tạo thành công với mục tiêu mà chúng tôi chỉ

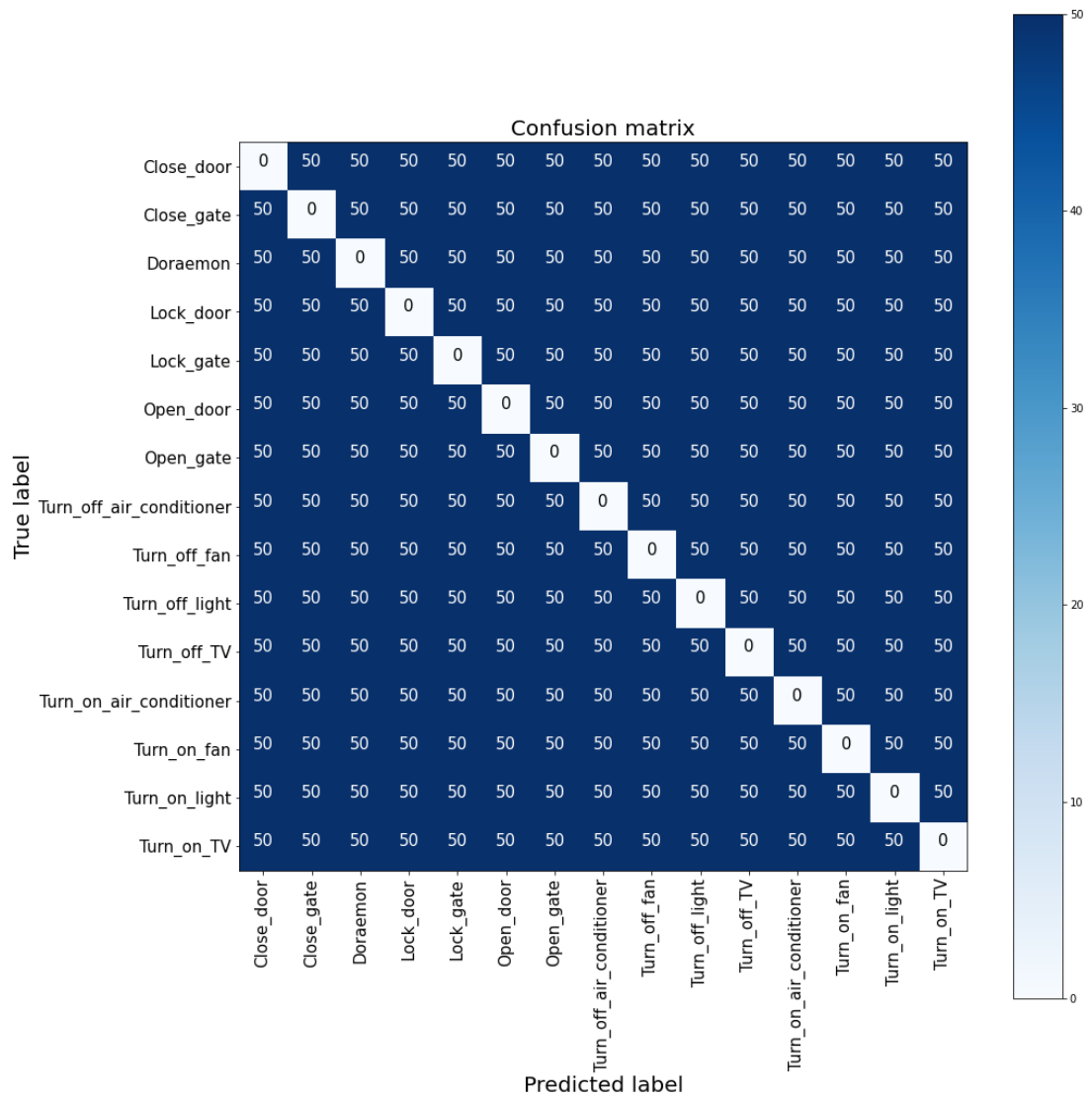
CHƯƠNG 6. THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ



Hình 6.2: Ma trận kết quả tấn công có mục tiêu dùng phương pháp cơ bản với $\epsilon = 100/2^{15}$

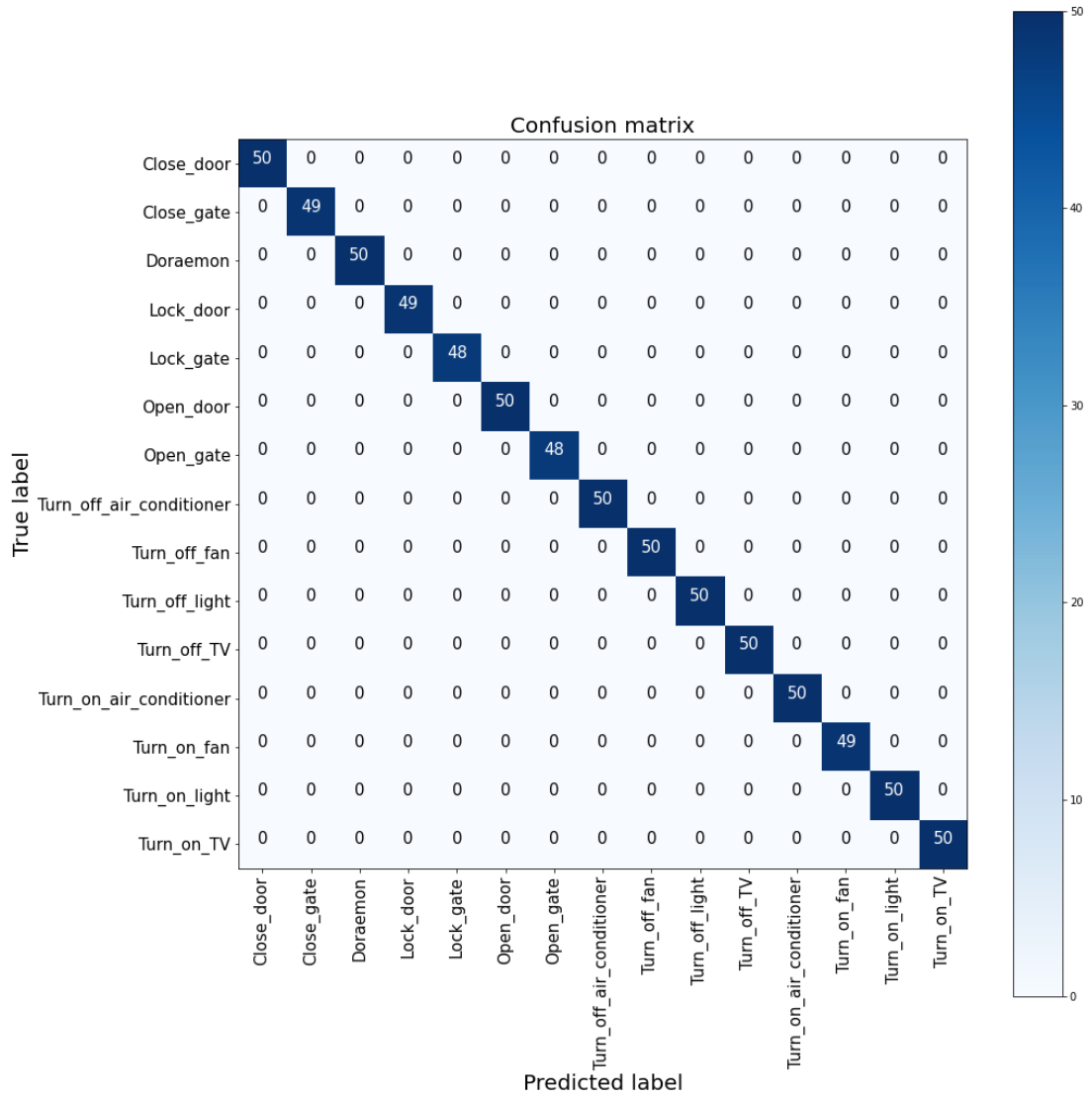
định. Qua đó, ta thấy khả năng tạo mẫu tấn công đối kháng với mô hình mục tiêu của phương pháp cải tiến có hiệu quả cao hơn rất nhiều so với phương pháp cơ bản ban đầu.

CHƯƠNG 6. THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ



Hình 6.3: Ma trận kết quả tấn công có mục tiêu dùng phương pháp cải tiến với $SNR_{dB} = 20dB$

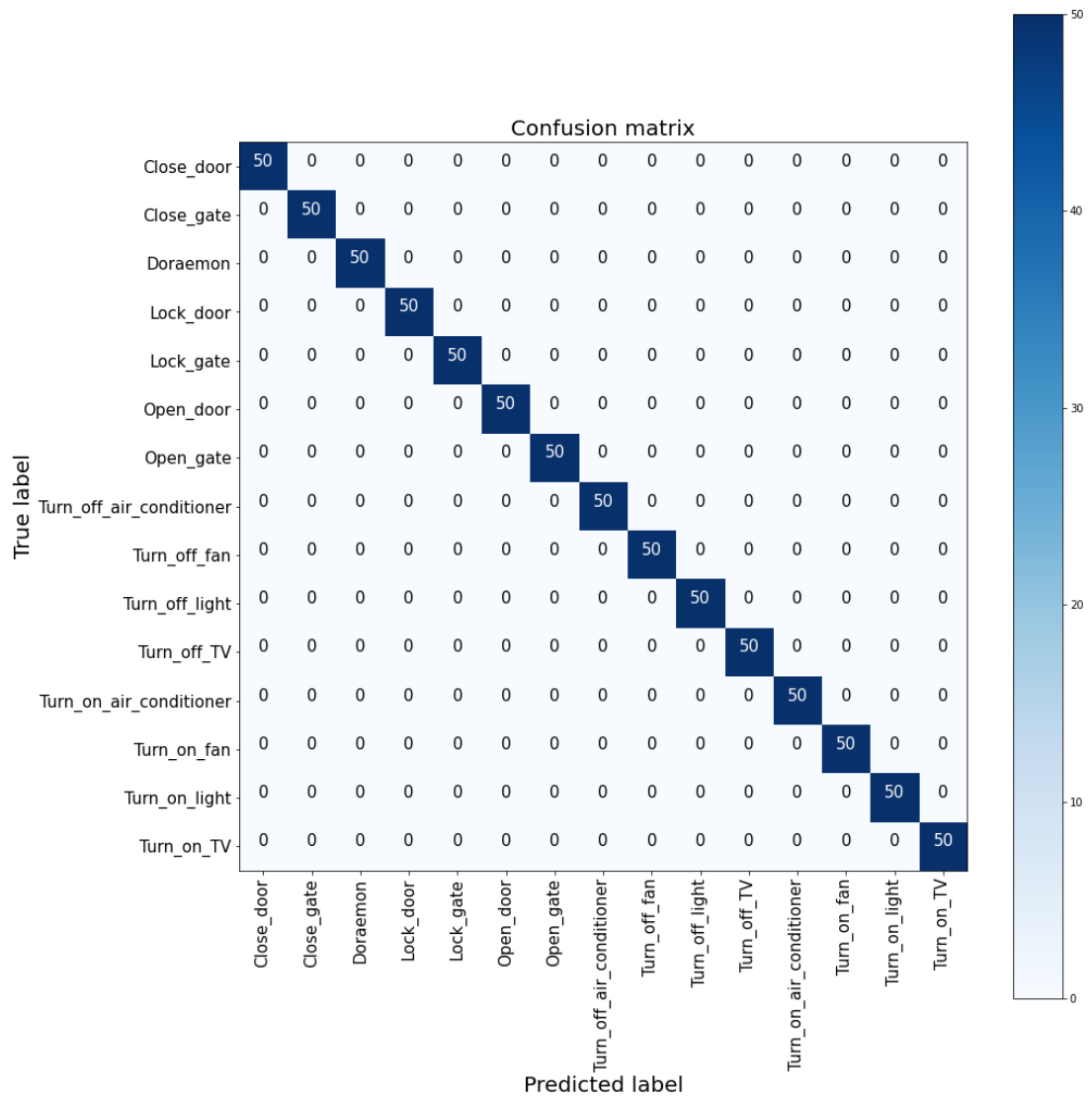
6.2.2. Tấn công không mục tiêu



Hình 6.4: Ma trận thể hiện kết quả tấn công không mục tiêu dùng phương pháp cơ bản với $\epsilon = 10/2^{15}$

Với các Hình 6.4, 6.5, chúng tôi thể hiện trên mỗi ô là số lượng các mẫu tấn công đối kháng được tạo thành công khiến cho mô hình nhận diện sai lệch khỏi lớp đó. Trong tấn công không mục tiêu, hiệu quả của giải thuật IFGSM và IFGSM cải tiến là gần tương đương nhau

CHƯƠNG 6. THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ



Hình 6.5: Ma trận thể hiện kết quả tấn công không mục tiêu dùng phương pháp cải tiến với $SNR_{dB} = 20dB$

7 Tổng kết

7.1. Kết quả đạt được

Trong luận văn này, chúng tôi đã xây dựng một mô hình tấn công đối kháng vào mô hình hộp trắng nhận diện và phân loại giọng nói tiếng Việt. Mô hình tạo các mẫu âm thanh đối kháng được dựa trên giải thuật IFGSM [8] cải tiến lại giải thuật dựa trên các yếu tố về âm lượng và nội dung âm thanh. Qua các thí nghiệm, chúng tôi nhận thấy yếu tố về âm lượng có ảnh hưởng khá lớn đến quá trình thêm nhiễu vào các mẫu âm thanh. Vì vậy chúng tôi đã đề xuất cải tiến và mang lại những kết quả đáng mong đợi.

Kết quả mà chúng tôi đạt được đó là tạo thành công 100% các mẫu tấn công đối kháng với lượng nhiễu âm lượng nhỏ hơn $15dB$ đến $20dB$ so với âm thanh gốc trong cả hai cuộc tấn công có mục tiêu và không mục tiêu với giải thuật IFGSM cải tiến. Với lượng nhiễu âm lượng nhỏ tai người vẫn dễ dàng nhận ra nội dung gốc nhưng vẫn khiến các mô hình nhận diện sai lệch. So sánh với kết quả tạo mẫu từ giải thuật IFGSM ban đầu cho thấy phương pháp của chúng tôi hiệu quả hơn. Tuy nhiên, so sánh trên là tương đối bởi vì tập dữ liệu của chúng tôi và các nghiên cứu được dựa trên các ngôn ngữ khác nhau.

7.2. Hạn chế và hướng phát triển

Trong quá trình nghiên cứu, các cuộc tấn công đối kháng trên các mô hình nhận diện giọng nói tiếng Việt còn quá mới mẻ, theo khảo sát của chúng tôi hiện nay chưa có một tài liệu tham khảo nào. Vì vậy, các nghiên cứu được tham khảo trong luận văn này đều được thực hiện trên các mô hình đối với tiếng Anh. Bên cạnh đó, các mô hình nhận diện giọng nói tiếng Việt dùng cho việc nghiên cứu còn nhiều hạn chế làm cho việc nghiên cứu bị giới hạn. Các hạn chế này sẽ được chúng tôi cải thiện trong các nghiên cứu sắp tới.

Ngoài các mô hình hộp trắng nhận diện phân loại giọng nói, thì trên thực tế hiện nay các mô hình nhận diện chuyển đổi giọng nói thành văn bản và các mô hình nhận diện giọng nói hộp đen được sử dụng khá phổ biến. Vì vậy bài toán có thể mở rộng để thực hiện tấn công đối kháng trên các mô hình đó. Từ đó, ta có thể nghiên cứu và đề xuất một số phương pháp phòng chống các cuộc tấn công có thể xảy ra trong tương lai. Đây là hai hướng nghiên cứu quan trọng sẽ được chúng tôi thực hiện nghiên cứu sắp tới. Ngoài ra, chúng tôi còn hướng đến ứng dụng quá trình tạo mẫu đối kháng như một quá trình mã hóa dữ liệu.

Tài liệu tham khảo

- [1] Xuejing Yuan et al. “Commandersong: a systematic approach for practical adversarial voice recognition”. In: *Proceedings of the 27th USENIX Conference on Security Symposium*. USENIX Association. 2018, pp. 49–64.
- [2] Yuxuan Chen et al. “Devil’s whisper: A general approach for physical adversarial attacks against commercial black-box speech recognition devices”. In: *29th USENIX Security Symposium (USENIX Security 20)*. 2020, pp. 2667–2684.
- [3] Moustafa Alzantot, Bharathan Balaji, and Mani Srivastava. “Did you hear that? adversarial examples against automatic speech recognition”. In: *arXiv preprint arXiv:1801.00554* (2018).
- [4] Kevin Eykholt et al. “Robust physical-world attacks on deep learning visual classification”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 1625–1634.
- [5] Yiming Li et al. “Backdoor learning: A survey”. In: *arXiv preprint arXiv:2007.08745* (2020).
- [6] Ali Shafahi et al. “Poison frogs! targeted clean-label poisoning attacks on neural networks”. In: *arXiv preprint arXiv:1804.00792* (2018).
- [7] Martin Abadi et al. “Deep learning with differential privacy”. In: *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 2016, pp. 308–318.
- [8] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. “Explaining and Harnessing Adversarial Examples”. In: (2015). URL: <http://arxiv.org/abs/1412.6572>.
- [9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems* 25 (2012), pp. 1097–1105.
- [10] Lea Schönherr et al. “Adversarial Attacks Against Automatic Speech Recognition Systems via Psychoacoustic Hiding”. In: *Network and Distributed System Security Symposium (NDSS)*. 2019.
- [11] Andrew Maas. *Spoken Language Processing*. 2017.

- [12] Vivek Tyagi and Christian Wellekens. “On desensitizing the Mel-Cepstrum to spurious spectral components for Robust Speech Recognition”. In: *Proceedings.(ICASSP’05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005*. Vol. 1. IEEE. 2005, pp. I–529.
- [13] Thomas H Cormen et al. *Introduction to algorithms*. MIT press, 2009.
- [14] Kehtarnavaz Nasser. *Digital Signal Processing System Design: LabVIEW Based Hybrid Programming*. 2008.
- [15] Paul S Addison. “Wavelet transforms and the ECG: a review”. In: *Physiological measurement* 26.5 (2005), R155.
- [16] Walid A Zgallai. *Biomedical Signal Processing and Artificial Intelligence in Healthcare*. Academic Press, 2020.
- [17] Tsai Wei-Yu et al. “Always-on speech recognition using truenorth, a reconfigurable, neurosynaptic processor”. In: *IEEE Transactions on Computers* 66.6 (2016), pp. 996–1007.
- [18] *Introduction to Speech Processing*. <https://wiki.aalto.fi/display/ITSP/Introduction+to+Speech+Processing>. Accessed: 2020-11-24.
- [19] James MacQueen et al. “Some methods for classification and analysis of multivariate observations”. In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. 14. Oakland, CA, USA. 1967, pp. 281–297.
- [20] Dong Yu and Li Deng. *Automatic Speech Recognition*. Springer.
- [21] *How to handle the seo by Markov chains*. <http://www.vincenzomusumeci.com/findability-seo/how-to-handle-seo-by-markov-chains/>. Accessed: 2020-12-28.
- [22] *File:Recurrent neural network unfold.svg*. https://commons.wikimedia.org/wiki/File:Recurrent_neural_network_unfold.svg. Accessed: 2021-03-30.
- [23] *Simple RNN vs GRU vs LSTM :- Difference lies in More Flexible control*. <https://medium.com/@saurabh.rathor092/simple-rnn-vs-gru-vs-lstm-difference-lies-in-more-flexible-control-5f33e07b1e57>. Accessed: 2021-03-30.
- [24] Ian Goodfellow et al. “Generative adversarial nets”. In: *Advances in neural information processing systems* 27 (2014).
- [25] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. “Neural machine translation by jointly learning to align and translate”. In: *arXiv preprint arXiv:1409.0473* (2014).
- [26] Hadi Abdullah et al. “Practical Hidden Voice Attacks against Speech and Speaker Recognition Systems”. In: *NDSS’19*. 2019, pp. 1369–1378.
- [27] Bhagwandas P Lathi. *Modern digital and analog communication systems*. Oxford University Press, Inc., 1990.

- [28] Guoming Zhang et al. “Dolphinattack: Inaudible voice commands”. In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 2017, pp. 103–117.
- [29] *Signal-to-noise ratio*. https://en.wikipedia.org/wiki/Signal-to-noise_ratio. Accessed: 2021-03-30.
- [30] Nicolas Papernot et al. “Practical black-box attacks against machine learning”. In: *Proceedings of the 2017 ACM on Asia conference on computer and communications security*. 2017, pp. 506–519.
- [31] Yinpeng Dong et al. “Boosting adversarial attacks with momentum”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 9185–9193.
- [32] Phan Duy Hung et al. “Vietnamese speech command recognition using recurrent neural networks”. In: *Int. J. Adv. Comput. Sci. Appl.(IJACSA)* 10.7 (2019).
- [33] Douglas Coimbra de Andrade et al. “A neural attention model for speech command recognition”. In: *arXiv preprint arXiv:1808.08929* (2018).
- [34] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).