

ĐẠI HỌC QUỐC GIA HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG

HỒ NGỌC THIỆN
PHÙNG ĐỨC LƯƠNG

ĐỒ ÁN CHUYÊN NGÀNH
NGHIÊN CỨU TÍNH CHUYỂN GIAO CỦA MÃU
LUỒNG MẠNG ĐỐI KHÁNG TRÊN CÁC HỆ THỐNG
PHÁT HIỆN XÂM NHẬP DỰA TRÊN HỌC SÂU
**A STUDY ON TRANSFERABILITY OF ADVERSARIAL
NETWORK FLOW AGAINST DEEP LEARNING-BASED
INTRUSION DETECTION SYSTEM**

TP. Hồ Chí Minh, 2025

ĐẠI HỌC QUỐC GIA HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG

HỒ NGỌC THIỆN - 21522620
PHÙNG ĐỨC LƯƠNG - 21522312

ĐỒ ÁN CHUYÊN NGÀNH
NGHIÊN CỨU TÍNH CHUYỂN GIAO CỦA MÃU
LUỒNG MẠNG ĐỐI KHÁNG TRÊN CÁC HỆ THỐNG
PHÁT HIỆN XÂM NHẬP DỰA TRÊN HỌC SÂU
**A STUDY ON TRANSFERABILITY OF ADVERSARIAL
NETWORK FLOW AGAINST DEEP LEARNING-BASED
INTRUSION DETECTION SYSTEM**

GIẢNG VIÊN HƯỚNG DẪN:
ThS. Phan Thế Duy

TP.Hồ Chí Minh - 2025

LỜI CẢM ƠN

Trong quá trình nghiên cứu và hoàn thành đồ án chuyên ngành, nhóm đã nhận được sự định hướng, giúp đỡ, các ý kiến đóng góp quý báu và những lời động viên của các giáo viên hướng dẫn và giáo viên bộ môn. Nhóm xin bày tỏ lời cảm ơn tới thầy Phan Thế Duy đã tận tình trực tiếp hướng dẫn, giúp đỡ trong quá trình nghiên cứu.

Hồ Ngọc Thiện
Phùng Đức Lương

MỤC LỤC

LỜI CẢM ƠN	i
MỤC LỤC	ii
DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT	v
DANH MỤC CÁC HÌNH VẼ	vi
DANH MỤC CÁC BẢNG BIỂU	vii
MỞ ĐẦU	1
 CHƯƠNG 1. TỔNG QUAN	 3
1.1 Dặt vấn đề	3
1.2 Giới thiệu những nghiên cứu liên quan	4
1.2.1 Tấn công đối kháng vào hệ thống phát hiện xâm nhập . .	4
1.2.2 Nghiên cứu các phương pháp phòng thủ cho hệ thống phát hiện xâm nhập chống lại các mẫu đối kháng	5
1.3 Tính ứng dụng	6
1.4 Những thách thức	6
1.5 Mục tiêu và cấu trúc đồ án chuyên ngành	7
1.5.1 Mục tiêu nghiên cứu	7
1.5.2 Cấu trúc đồ án chuyên ngành	7
 CHƯƠNG 2. CƠ SỞ LÝ THUYẾT	 8
2.1 Hệ thống phát hiện xâm nhập (IDS)	8
2.2 Giới thiệu về học sâu	9
2.2.1 Khái niệm học sâu	9
2.2.2 Một số khái niệm trong học sâu	10
2.3 Các mô hình học sâu sử dụng trong đề tài	12
2.3.1 Mô hình Convolutional neuron network (CNN)	12

2.3.2	Mô hình Long Short-Term Memory (LSTM)	14
2.3.3	Mô hình Gated Recurrent Unit kết hợp với Convolutional Neural Network (CNN+GRU)	15
2.4	Mô hình mạng sinh đối kháng (GAN)	16
2.4.1	GAN là gì?	16
2.4.2	Cấu trúc mạng GAN	17
2.5	Tấn công đối kháng (Adversarial Attack)	18
2.5.1	Nguyên lý hoạt động	18
2.5.2	Các loại tấn công đối kháng	19
2.6	Các phương pháp phòng thủ mẫu đối kháng cho hệ thống phát hiện xâm nhập	20
2.6.1	Kernel Density Estimation	20
2.6.2	MANDA	21
2.6.3	Adversarial Training	25
CHƯƠNG 3. THIẾT KẾ HỆ THỐNG		27
3.1	Phát sinh dữ liệu đối kháng bằng tấn công đối kháng	27
3.2	Xây dựng các hệ thống phát hiện xâm nhập dựa trên các mô hình học sâu	30
3.3	Tổng quan mô hình đề xuất	33
CHƯƠNG 4. THÍ NGHIỆM VÀ ĐÁNH GIÁ		37
4.1	Môi trường thực nghiệm	37
4.1.1	Tài nguyên	37
4.1.2	Tập dữ liệu	37
4.1.3	Tiền xử lý dữ liệu	38
4.2	Kết quả thí nghiệm	39
4.2.1	Kết quả xây dựng các mô hình học sâu phát hiện tấn công	39
4.2.2	Tỉ lệ trốn tránh của các mẫu đối kháng được tạo ra từ CTGAN khi chưa có biện pháp phòng thủ	40

4.2.3 Kết quả về hiệu suất của các mô hình phát hiện xâm nhập dựa trên học sâu và tỉ lệ trốn tránh của các mẫu đối kháng khi áp dụng các biện pháp phòng thủ	41
CHƯƠNG 5. KẾT LUẬN	45
5.1 Kết luận	45
5.2 Hướng phát triển	46
TÀI LIỆU THAM KHẢO	47

DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT

DL	Deep Learning
IDS	Intrusion Detection System
NIDS	Network-based IDS
HIDS	Host-based IDS
GAN	Generative Adversarial Network
CTGAN	Conditional Tabular Generative Adversarial Network
CNN	Convolutional Neural Network
RNN	Recurrent Neural Networks
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Unit
ReLU	Rectified Linear Unit
DNN	Deep Neural Network
ASR	Attack Success Rate
KDE	Kernel Density Estimation
AE	Adversarial Example
MANDA	MANifold and Decision boundary-based AE

DANH MỤC CÁC HÌNH VẼ

Hình 2.1	Cấu trúc mạng nơ-ron	10
Hình 2.2	Ảnh mặt người sinh bởi GAN	17
Hình 2.3	Các mạng trong GAN	18
Hình 2.4	Kết quả mô hình đưa ra kết quả phân loại sai sau khi thêm nhiều vào mẫu ban đầu	19
Hình 2.5	Phân loại AEs dựa trên 2 TH A và B	22
Hình 2.6	Hàm Score-Compute() cho tiêu chí 1 và 2	22
Hình 2.7	Thuật toán 2 Manifold	23
Hình 2.8	Thuật toán 3 DB	24
Hình 2.9	Thuật toán 4 MANDA	25
Hình 3.1	Phân phối Gauss so với Phân phối MultiModal	28
Hình 3.2	Ví dụ về mode-specific normalization	29
Hình 3.3	CTGAN Model	30
Hình 3.4	Kiến trúc CNN 3 lớp	31
Hình 3.5	Kiến trúc CNN 5 lớp	31
Hình 3.6	Kiến trúc CNN 7 lớp	32
Hình 3.7	Kiến trúc LSTM	32
Hình 3.8	Kiến trúc CNN kết hợp với GRU	33
Hình 3.9	Mô Hình về quá trình tấn công đối kháng vào mô hình học sâu được trang bị Adversarial Training	34
Hình 3.10	Mô Hình về quá trình tấn công đối kháng vào mô hình học sâu được trang bị Adversarial Detection	34
Hình 4.1	Quá trình tiền xử lý tập dữ liệu Edge-IIoT	39

DANH MỤC CÁC BẢNG BIỂU

Bảng 4.1	Kết quả đánh giá của mô hình học sâu cho tập dữ liệu DNN-Edge-IIoT	40
Bảng 4.2	Tỉ lệ trốn tránh của các mẫu đối kháng đối với các mô hình IDS	41
Bảng 4.3	Tỉ lệ trốn tránh của các mẫu đối kháng trước và sau khi áp dụng Adversarial Retraining	42
Bảng 4.4	Tỉ lệ trốn tránh của các mẫu đối kháng khi áp dụng KDE và MANDA cho mô hình IDS	43

TÓM TẮT ĐỒ ÁN CHUYÊN NGÀNH

Với sự bùng nổ của mạng, số lượng thiết bị trong mạng tăng theo cấp số nhân, đi kèm với các mối đe dọa bảo mật nghiêm trọng hơn đối với các thiết bị mạng. Các mối đe dọa bảo mật này có thể gây nguy hiểm cho tính bảo mật, tính toàn vẹn và tính khả dụng của tài sản. Do đó, việc bảo vệ mạng và hệ thống thông tin khỏi các cuộc tấn công mạng tiềm ẩn là rất quan trọng. IDS đã nổi lên như một trong những lựa chọn tốt nhất để cung cấp các giải pháp bảo mật chống lại nhiều cuộc tấn công xâm nhập mạng. Với sự tiến bộ nhanh chóng của công nghệ phần cứng và sự xuất hiện của nhiều GPU có sức mạnh tính toán cao, nhiều nhà nghiên cứu đang khám phá khả năng áp dụng các kỹ thuật học sâu (DL) để phát hiện lưu lượng đáng ngờ trong luồng mạng. Tuy nhiên, mặc cho hiệu quả của chúng, IDS dựa trên DL dễ bị tấn công đối kháng, trong đó kẻ tấn công tạo ra nhiều loạn tinh vi đến dữ liệu đầu vào, dẫn đến hệ thống phân loại sai. Các mẫu đối nghịch (AEs) này khai thác các lỗ hổng vốn có trong các mô hình học sâu, gây ra mối đe dọa đáng kể đến độ tin cậy của IDS. Nghiên cứu này tập trung vào tính mạnh mẽ của IDS dựa trên DL khi đối mặt với các cuộc tấn công đối kháng được tạo ra bằng GAN. Cụ thể, nghiên cứu đánh giá khả năng chuyển giao của các mẫu đối kháng - khả năng đánh lừa mô hình này và cũng có thể đánh lừa các mô hình khác. Bằng cách phân tích một cách có hệ thống tỷ lệ thành công và tác động của các cuộc tấn công như vậy, nghiên cứu này nhằm mục đích định lượng mức độ đe dọa do các mẫu đối kháng gây ra cho IDS dựa trên DL. Phương pháp nghiên cứu bao gồm thiết kế một khuôn khổ dựa trên GAN để tạo ra các luồng mạng đối kháng. Sau đó, các luồng này được thử nghiệm với nhiều mô hình IDS dựa trên DL khác nhau để đo khả năng chuyển giao và hiệu quả của chúng trong việc tránh phát hiện. Các phát hiện cho thấy các cuộc tấn công đối kháng có thể làm giảm đáng kể hiệu suất

của IDS dựa trên DL. Để giải quyết thách thức này, nghiên cứu đề xuất ba chiến lược phòng thủ nhằm giảm thiểu thành công của các cuộc tấn công đối địch. Các giải pháp này bao gồm đào tạo đối nghịch (Adversarial Training) để tăng cường khả năng phục hồi của mô hình, kết hợp các cơ chế phát hiện dựa trên thống kê (Statistic) và dựa trên biến đổi (Mutation) để xác định các mẫu bất thường trong luồng mạng. Các biện pháp phòng thủ được đề xuất chứng minh tỷ lệ thành công của các cuộc tấn công đối địch giảm đáng kể, do đó cải thiện độ tin cậy và tính mạnh mẽ của IDS dựa trên DL. Nghiên cứu này đóng góp vào lĩnh vực an ninh mạng bằng cách cung cấp phân tích toàn diện về các mối đe dọa đối địch đối với IDS dựa trên DL và giới thiệu các chiến lược giảm thiểu hiệu quả. Bằng cách tăng cường hiểu biết về các cơ chế tấn công đối kháng và các biện pháp đối phó của chúng, nghiên cứu này mở đường cho việc phát triển các hệ thống phát hiện xâm nhập mạng an toàn và đáng tin cậy hơn.

CHƯƠNG 1. TỔNG QUAN

Chương này giới thiệu về ngữ cảnh và các nghiên cứu liên quan. Đồng thời, trong chương này nhóm cũng trình bày phạm vi và cấu trúc của Đồ án chuyên ngành.

1.1. Đặt vấn đề

Trong bối cảnh an ninh mạng ngày càng trở thành mối quan tâm hàng đầu, hệ thống phát hiện xâm nhập (Intrusion Detection System - IDS) dựa trên học máy và học sâu đã nổi lên như một giải pháp hiệu quả nhờ khả năng tự động học và phát hiện các mẫu hành vi bất thường trong lưu lượng mạng. Các kỹ thuật học sâu như Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), hoặc sự kết hợp của chúng đã chứng minh hiệu quả vượt trội trong việc phân loại lưu lượng mạng và phát hiện tấn công. Tuy nhiên, tính an toàn của các mô hình học sâu này trước các cuộc tấn công tinh vi, đặc biệt là tấn công đối kháng (adversarial attack), vẫn còn là một câu hỏi lớn.

Adversarial attack tạo ra các mẫu dữ liệu được tối ưu hóa để đánh lừa mô hình học sâu bằng cách thực hiện các thay đổi nhỏ nhưng có chủ đích vào dữ liệu gốc. Trong ngữ cảnh IDS, những thay đổi này có thể gây ra hậu quả nghiêm trọng khi các cuộc tấn công bị bỏ qua hoặc phân loại sai. Đáng lo ngại hơn, tính chuyển giao của các mẫu đối kháng – khả năng một mẫu đối kháng được tạo ra cho một mô hình có thể đánh lừa các mô hình khác – đã mở ra những nguy cơ mới trong an ninh mạng. Với sự phát triển của các kỹ thuật tấn công đối kháng, nghiên cứu về khả năng chuyển đổi của các mẫu đối kháng trong môi trường lưu lượng mạng là vô cùng cần thiết. Việc này không chỉ giúp hiểu rõ hơn về mức độ an toàn của các IDS dựa trên học sâu mà còn đặt nền móng cho

việc phát triển các cơ chế phòng thủ hiệu quả hơn. Tuy nhiên, hiện tại vẫn còn thiếu các nghiên cứu chuyên sâu về việc đánh giá các mẫu đối kháng được tạo ra khi đánh lừa nhiều mô hình IDS cùng lúc.

Do đó, nhóm tiến hành khám phá sâu hơn về tính chất chuyển đổi của mẫu đối kháng trong hệ thống phát hiện xâm nhập dựa trên học sâu. Nhóm sử dụng CTGAN (Conditional Tabular Generative Adversarial Network) - một mô hình dựa trên mạng sinh đối kháng (GAN) được thiết kế đặc biệt để tạo dữ liệu tổng hợp cho các tập dữ liệu dạng bảng (tabular data) để sinh ra các mẫu dữ liệu đối kháng nhằm đánh lừa các mô hình IDS. Các mô hình IDS dựa trên học sâu được nhóm lựa chọn để đánh giá bao gồm: CNN 3 lớp, CNN 5 lớp, CNN 7 lớp, CNN kết hợp với GRU, LSTM. Sau khi đưa các mẫu đối kháng vào các mô hình IDS để phân loại, ta thu được kết quả là tỉ lệ tấn công thành công (Attack Success Rate) của mẫu đối kháng so với từng mô hình IDS nhằm đánh giá khả năng chuyển giao của mẫu đối kháng đó đối với các mô hình IDS dựa trên học sâu khác nhau. Bên cạnh đó, nhóm tiến hành đánh giá lại hiệu suất tấn công thành công của các mẫu đối kháng khi áp dụng các phương pháp phòng thủ bao gồm Adversarial Training, KDE (Kernel Density Estimation) và MANDA[11] cho các mô hình IDS dựa trên học sâu. Công việc này giúp đánh giá sâu hơn tính chuyển giao của các mẫu đối kháng đối với các mô hình IDS dựa trên học sâu kể cả khi áp dụng các phương pháp phòng thủ

1.2. Giới thiệu những nghiên cứu liên quan

1.2.1. Tấn công đối kháng vào hệ thống phát hiện xâm nhập

Tấn công đối kháng vào hệ thống phát hiện xâm nhập (IDS) là một trong những thách thức lớn đối với an ninh mạng hiện đại, đặc biệt trong bối cảnh các hệ thống này ngày càng dựa vào học sâu để phát hiện các mối đe dọa. Các cuộc tấn công đối kháng lợi dụng những điểm yếu trong mô hình học sâu để tạo ra các mẫu dữ liệu được tinh chỉnh nhằm làm giảm hiệu năng của IDS, khiến hệ

thống không thể nhận diện chính xác các cuộc tấn công hoặc nhầm lẫn chúng với các mẫu hợp lệ. Theo nghiên cứu của N. Wang và các cộng sự [11], tấn công đối kháng được phân loại thành ba dạng chính: tấn công hộp trăng, hộp xám, và hộp đen. Trong các trường hợp tấn công hộp đen, tính chuyển giao của các mẫu đối kháng giữa các mô hình khác nhau đóng vai trò then chốt, cho phép kẻ tấn công sử dụng một mô hình thay thế để đánh lừa hệ thống mục tiêu.

Những cuộc tấn công này không chỉ làm giảm độ chính xác của IDS mà còn gây nguy hiểm cho toàn bộ hệ thống mạng, cho phép các cuộc xâm nhập diễn ra mà không bị phát hiện. Các thách thức đặt ra đối với việc phòng thủ trước các tấn công đối kháng trong IDS bao gồm việc xác định các mẫu đối kháng một cách chính xác, duy trì hiệu năng trong môi trường thực tế, và giảm thiểu chi phí tính toán để đảm bảo tính khả thi của hệ thống.

1.2.2. Nghiên cứu các phương pháp phòng thủ cho hệ thống phát hiện xâm nhập chống lại các mẫu đối kháng

Đã có nhiều nghiên cứu về tấn công đối kháng được thực hiện thành công nhằm qua mặt các hệ thống phát hiện xâm nhập dựa trên học máy. Các cuộc tấn công này sử dụng mẫu đối kháng để giảm hiệu năng của các mô hình một cách tinh vi và hiệu quả bằng cách tạo ra các thay đổi nhỏ trong lưu lượng mạng bình thường. Được nghiên cứu bởi N. Wang và các cộng sự [11], nhóm tác giả đã tập trung vào tấn công hộp đen và đã đề xuất MANDA - một khung phòng thủ hiệu quả được thiết kế để bảo vệ các hệ thống phát hiện xâm nhập (IDS) trước các tấn công đối kháng, đặc biệt trong bối cảnh tấn công hộp đen để đối phó với loại tấn công này.

Bên cạnh đó, nghiên cứu của Goodfellow và cộng sự [3] giới thiệu và mô tả phương pháp phòng thủ Adversarial Training, một kỹ thuật kết hợp các mẫu đối kháng trực tiếp vào quá trình huấn luyện mô hình. Phương pháp này hoạt động bằng cách bổ sung các mẫu đối kháng vào tập huấn luyện nhằm cải thiện khả năng tổng quát của mô hình trước các cuộc tấn công. Adversarial Training

không chỉ giúp mô hình trở nên bền vững hơn trước các nhiễu loạn đối kháng mà còn làm tăng khả năng phát hiện các mẫu bất thường trong môi trường thực tế. Một trong những phương pháp phòng thủ hiệu quả khác là Kernel Density Estimation (KDE) [8] được Feinman đề cập đến, một kỹ thuật thống kê được sử dụng để mô hình hóa phân phối xác suất của dữ liệu. KDE hoạt động bằng cách ước tính mật độ xác suất của các điểm dữ liệu trong không gian đầu vào, từ đó xác định các điểm dữ liệu bất thường hoặc không phù hợp với phân phối của tập dữ liệu gốc. Phương pháp này được ứng dụng trong phòng thủ trước các mẫu đối kháng bằng cách sử dụng KDE để phân biệt các mẫu hợp lệ và các mẫu đối kháng tiềm năng. Các mẫu có mật độ thấp trong không gian đầu vào, theo KDE, thường bị coi là đối kháng và có thể bị loại bỏ hoặc gán cờ để xử lý thêm..

1.3. Tính ứng dụng

Đề tài này tập trung vào việc nghiên cứu hiện tượng chuyển giao của các mẫu đối kháng trong mạng lưu lượng (network flow) nhằm đánh bại các hệ thống phát hiện xâm nhập (IDS) dựa trên học sâu. Nghiên cứu này mang ý nghĩa quan trọng trong lĩnh vực an ninh mạng, đặc biệt trong việc đánh giá và cải thiện khả năng chống chịu của các hệ thống IDS trước các cuộc tấn công đối kháng phức tạp.

1.4. Những thách thức

Nghiên cứu về tính chuyển giao của các mẫu đối kháng trong IDS đối mặt với nhiều thách thức, bao gồm sự đa dạng và phức tạp của lưu lượng mạng, đòi hỏi các mẫu đối kháng phải thực tế và khó bị phát hiện. Tính đa dạng về kiến trúc và dữ liệu huấn luyện của IDS khiến việc đảm bảo tính chuyển giao giữa các hệ thống trở nên khó khăn. Trong kịch bản hộp đen, việc thiếu thông tin về mô hình mục tiêu làm tăng độ phức tạp và chi phí tính toán.

1.5. Mục tiêu và cấu trúc đồ án chuyên ngành

1.5.1. Mục tiêu nghiên cứu

- Phân tích tính chuyển giao của các mẫu đối kháng: Nghiên cứu khả năng các mẫu đối kháng được tạo từ một mô hình có thể đánh lừa các hệ thống IDS khác với kiến trúc hoặc dữ liệu huấn luyện khác nhau.
- Đánh giá mức độ hiệu quả của tấn công hộp đen: Xác định hiệu suất của các mẫu đối kháng trong kịch bản không có thông tin về mô hình mục tiêu.
- Phát triển hiểu biết sâu sắc về lỗ hổng của IDS: Tìm hiểu những điểm yếu tiềm ẩn của các mô hình IDS dựa trên học sâu trước các chiến lược tấn công đối kháng.
- Định hướng thiết kế giải pháp phòng thủ: Cung cấp cơ sở khoa học để phát triển các chiến lược phòng thủ nhằm giảm khả năng thành công của các cuộc tấn công đối kháng.

1.5.2. Cấu trúc đồ án chuyên ngành

Cấu trúc của đồ án chuyên ngành như sau:

- Chương 1: Giới thiệu tổng quan về đề tài của Đồ án và những nghiên cứu liên quan.
- Chương 2: Trình bày cơ sở lý thuyết và kiến thức nền tảng liên quan đến đề tài.
- Chương 3: Trình bày mô hình nghiên cứu đề xuất.
- Chương 4: Trình bày thực nghiệm và đánh giá.
- Chương 5: Kết luận và hướng phát triển của đề tài.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

Chương này trình bày cơ sở lý thuyết của nghiên cứu: Bao gồm hệ thống phát hiện xâm nhập, mô hình học sâu cho hệ thống tìm kiếm, phát hiện xâm nhập, các kỹ thuật sinh đôi kháng và tấn công đôi kháng.

2.1. Hệ thống phát hiện xâm nhập (IDS)

Hệ thống phát hiện xâm nhập (IDS) là hệ thống phần mềm hoặc phần cứng tự động thực hiện quy trình phát hiện xâm nhập bao gồm theo dõi các sự kiện diễn ra trong một hệ thống máy tính hoặc mạng máy tính và phân tích để nhận biết các dấu hiệu của sự bất thường (hành vi xâm nhập - intrusion). IDS chủ yếu được phân loại dựa theo nguồn dữ liệu và kỹ thuật phân tích:

- NIDS (Network-based IDS): thường được triển khai ở biên mạng như gần firewall hoặc router biên, server VPN, server remote access và mạng không dây. NIDS có chức năng theo dõi lưu lượng mạng cho một phần mạng (network segment) hoặc các thiết bị, phân tích các hoạt động mạng và các giao thức ứng dụng để xác định hành vi bất thường.
- HIDS (Host-based IDS): thường được triển khai ở các host quan trọng như các server có thể truy cập internet và các server chứa thông tin quan trọng. HIDS có chức năng theo dõi các đặc điểm của một host và các sự kiện xảy ra trong host đó (trong mạng LAN) để nhận biết các hành vi đáng ngờ.
- Signature-based IDS: hay còn được gọi là knowledge-based, đây là IDS hoạt động giống các phần mềm diệt virus, dựa vào các chữ ký (signature) và các dấu hiệu của nguy cơ cũng như tấn công đã biết từ trước. Loại IDS này

không thể phát hiện các bất thường chưa biết trước hoặc biến đổi nhỏ trong những tấn công đã biết.

- Anomaly-based IDS: hay còn được gọi là profile-based, IDS này hoạt động bằng cách dựa trên các profile về các hành vi bình thường, dự kiến được thiết lập trước (giống như một whitelist). Bất kỳ hành vi nào nằm ngoài profile đều được xem là bất thường nên có thể phát hiện các tấn công đã biết và chưa biết. Tuy nhiên có tỉ lệ false positive cao vì nhầm hành vi bình thường thành tấn công (do profile chưa được xây dựng chặt chẽ).

2.2. Giới thiệu về học sâu

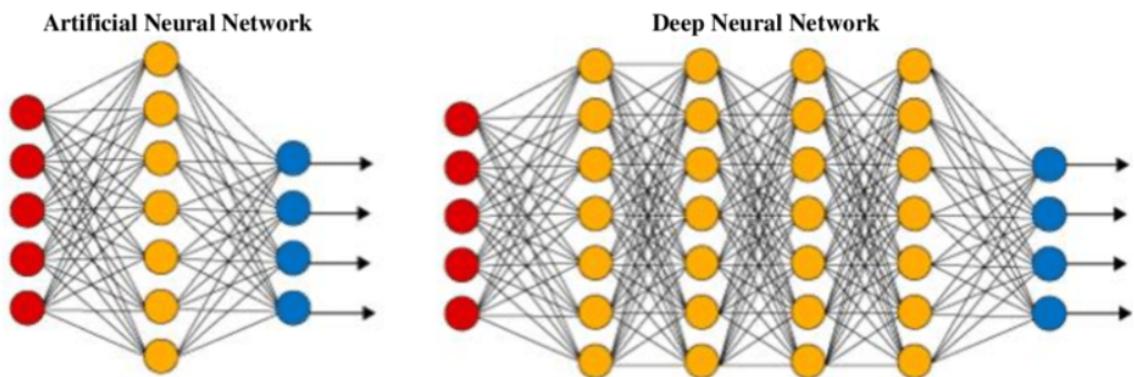
2.2.1. Khái niệm học sâu

Học sâu là một lĩnh vực của học máy, nơi máy tính được đào tạo để học một cách tự nhiên giống như con người. Học sâu được áp dụng chủ yếu trong các ứng dụng như xe tự lái, cho phép chúng tham gia giao thông mà không cần sự can thiệp của con người. Ngoài ra, học sâu cũng được áp dụng trong các thiết bị thông minh như trợ lý ảo trên loa thông minh, máy tính bảng và điện thoại thông minh. Vì những lợi ích này, học sâu đang trở thành một xu hướng quan trọng, thu hút sự quan tâm rất lớn và đạt được những thành tựu đáng kể, với tiềm năng phát triển tiếp theo.

Kiến trúc của học sâu bao gồm nhiều lớp dữ liệu được gán nhãn và sử dụng nhiều kiến trúc mạng nơ-ron nhân tạo. Dữ liệu được đưa qua các lớp mạng từ lớp đầu vào, đi qua các lớp ẩn và kết thúc tại lớp đầu ra. Các lớp mạng ẩn trong kiến trúc học sâu cung cấp khả năng học mạnh mẽ, giúp thuật toán học sâu đạt được kết quả tốt hơn so với các mô hình học máy truyền thống.

2.2.2. Một số khái niệm trong học sâu

- Mạng nơ-ron: Mạng nơ-ron trong học sâu mô phỏng lại cấu trúc mạng lưới nơ-ron trong não người, trong đó các nơ-ron được kết nối với nhau. Các nơ-ron trong mạng nơ-ron được chia thành ba loại là lớp đầu vào, lớp ẩn và lớp đầu ra.



Hình 2.1: Cấu trúc mạng nơ-ron

- Tế bào thần kinh (perceptron): Một tế bào thần kinh có thể được hiểu đơn giản như một hàm toán học, nơi nó nhận đầu vào từ một hoặc nhiều số và thực hiện các phép toán để tính toán kết quả đầu ra. Trọng số của tế bào thần kinh là các giá trị mà chúng ta cần tìm và được xác định thông qua quá trình huấn luyện.
- Hàm kích hoạt (activation functions): Trong một mô hình, các nơ-ron trong lớp ẩn sử dụng các hàm phi tuyến tính để tính toán đầu ra của chúng và chuyển tiếp nó cho lớp tiếp theo. Có một số hàm kích hoạt phổ biến được sử dụng, bao gồm Sigmoid, Tanh và Rectified Linear Unit (ReLU). Các hàm kích hoạt này giúp tạo ra tính phi tuyến tính và khả năng học linh hoạt cho mô hình.
 - Sigmoid: Hàm sigmoid hay còn được gọi là đường cong sigmoid, là một hàm liên tục mà ánh xạ đầu vào từ các số thực vào các giá trị trong khoảng từ 0 đến 1. Hàm này được sử dụng trong học máy để chuyển

đổi đầu vào thành xác suất hoặc những giá trị có ý nghĩa xác suất. Giá trị trả về được biểu diễn dưới dạng một hàm số.

- Tanh: Hàm tanh là một hàm kích hoạt được sử dụng trong học máy, với đặc điểm là đầu ra của nó nằm trong khoảng $(-1, 1)$. Điều này làm cho hàm tanh phù hợp cho các mô hình có đầu ra với ba giá trị: âm, trung tính (0) và dương. Hàm tanh giúp biểu diễn các mức độ khác nhau của đầu vào và tạo ra một phản ứng tương tự như hàm sigmoid, nhưng với khoảng giá trị mở rộng hơn. Hàm tanh cũng là một hàm liên tục và có thể biểu diễn dưới dạng một hàm số.
- ReLU: Hàm ReLU (Rectified Linear Unit) được xây dựng dựa trên ý tưởng loại bỏ các tham số không quan trọng trong quá trình huấn luyện, nhằm tạo ra một mô hình mạng nhẹ, nhanh chóng và hiệu quả hơn. Hàm ReLU thực hiện việc giữ nguyên các giá trị đầu vào lớn hơn 0, trong khi đối với các giá trị nhỏ hơn 0, chúng được coi như là 0. Điều này giúp hàm ReLU đơn giản hóa tính toán và giảm độ phức tạp của mạng. Hàm ReLU không có đạo hàm tại 0, nhưng trong thực tế, điều này ít ảnh hưởng đến quá trình huấn luyện và đã được chứng minh là rất hiệu quả trong nhiều mô hình mạng nơ-ron.
- Softmax: Hàm softmax, còn được gọi là hàm trung bình mũ, được sử dụng để tính toán xác suất của một sự kiện, thường được áp dụng trong bài toán phân loại đa lớp. Hàm softmax tính toán khả năng xuất hiện của mỗi lớp trong tổng số các lớp có thể xuất hiện, sau đó sử dụng xác suất này để xác định lớp mục tiêu cho đầu vào. Hàm softmax giúp chúng ta hiểu mức độ đáng tin cậy của các lớp và thường được sử dụng để tạo ra phân phối xác suất đa lớp.
- Dropout: Dropout là một kỹ thuật được sử dụng để ngăn chặn hiện tượng overfitting (quá khớp) trong mô hình học máy. Kỹ thuật này hoạt động bằng cách ngẫu nhiên loại bỏ một số đơn vị (neuron) trong quá trình huấn luyện. Khi loại bỏ một đơn vị, nó sẽ không được sử

dụng trong quá trình tính toán và cập nhật các trọng số trong mạng. Dropout giúp giảm sự phụ thuộc quá mức giữa các đơn vị trong mạng nơ-ron kết nối đầy đủ (fully-connected) trong mô hình học sâu. Điều này có tác dụng giúp mô hình trở nên chống lại overfitting và tổng quát hóa tốt hơn trên dữ liệu mới.

- One-hot Coding: One-hot encoding là một phương pháp được sử dụng để biểu diễn các biến hoặc lớp đầu ra trong các bài toán phân loại. Phương pháp này chuyển đổi các giá trị thành các đặc trưng nhị phân chỉ có giá trị 1 hoặc 0. Mỗi mẫu trong tập dữ liệu sẽ được chuyển thành một vector có kích thước n , trong đó giá trị 1 chỉ ra trạng thái "active" và giá trị 0 cho trạng thái "inactive" của đặc trưng tương ứng. One-hot encoding giúp đưa thông tin về sự hiện diện hoặc vắng mặt của một đặc trưng trong một mẫu cụ thể.
- Max pooling: Max pooling là một lớp được áp dụng giữa các lớp tích chập trong mô hình học sâu nhằm giảm kích thước của dữ liệu thông qua quá trình lấy mẫu. Quá trình này thực hiện bằng cách chia dữ liệu thành các ô nhỏ và chọn giá trị lớn nhất (max) trong mỗi ô làm giá trị đại diện. Kỹ thuật max pooling giúp giảm kích thước dữ liệu, giữ lại các đặc trưng quan trọng và giảm hiện tượng overfitting (quá khớp) trong mô hình học sâu.

2.3. Các mô hình học sâu sử dụng để tài

2.3.1. Mô hình *Convolutional neuron network (CNN)*

Convolutional Neural Network là một loại mạng nơ-ron, thường được áp dụng cho các bài toán phân loại và thị giác máy tính. Nó cung cấp một phương pháp tiếp cận tốt và có khả năng mở rộng bằng cách sử dụng các nguyên tắc từ đại số tuyến tính, đặc biệt là phép nhân ma trận để xác định các mẫu nằm trong dữ liệu. Đồng thời, so với các mạng nơ-ron khác, CNN có hiệu suất vượt trội

khi xử lý các đầu vào là tín hiệu hình ảnh, giọng nói hoặc âm thanh. CNN sử dụng ba lớp chính để xử lý dữ liệu:

- Lớp tích chập (convolutional): Là thành phần chính và nơi quan trọng trong quá trình học và tính toán của mạng nơ-ron. Nó sử dụng các bộ lọc, còn được gọi là bộ phát hiện đặc trưng, để quét qua từng vùng của dữ liệu đầu vào và xác định sự xuất hiện của các đặc trưng. Ta cũng phải xem xét cẩn thận các siêu tham số (hyperparameters) của bộ lọc, vì chúng ảnh hưởng đến kích thước của dữ liệu đầu ra. Bên cạnh đó, việc chia sẻ các trọng số giữa các vùng của đầu vào giúp bộ lọc không bị thay đổi khi di chuyển qua từng vùng khác nhau của dữ liệu.
- Lớp pooling: Lớp này được sử dụng để giảm kích thước không gian của dữ liệu và giảm số lượng tham số đầu vào. Điều này giúp làm giảm độ phức tạp của mô hình, nâng cao hiệu quả tính toán và hạn chế rủi ro overfitting. Giống như lớp tích chập, lớp pooling cũng sử dụng một bộ lọc để quét qua từng vùng của đầu vào. Tuy nhiên, bộ lọc này không có trọng số như lớp tích chập. Thay vào đó, nó sử dụng một hàm tổng hợp trên từng vùng tiếp nhận của đầu vào và đưa ra một giá trị duy nhất cho mỗi vùng, sau đó ghi kết quả này vào một mảng đầu ra. Điều này giúp giảm kích thước của dữ liệu mà không ảnh hưởng quá nhiều đến thông tin quan trọng trong dữ liệu.
- Lớp fully-connected: Đây là lớp có nhiệm vụ biến đầu ra của lớp trước đó thành một vector và thực hiện phân loại dựa trên các đặc trưng đã được trích xuất qua các lớp trước đó và bộ lọc tương ứng. Mỗi nút trong lớp fully-connected được kết nối trực tiếp với tất cả các nút trong lớp trước đó và sử dụng các hàm kích hoạt như sigmoid hoặc softmax để tính toán đầu ra và phân loại. Các hàm sigmoid được sử dụng trong trường hợp phân loại nhị phân, trong khi hàm softmax thường được sử dụng trong bài toán phân loại đa lớp, để xác định xác suất của mỗi lớp đầu ra.

Convolutional Neural Network bắt đầu với lớp convolutional làm lớp đầu tiên.

Các lớp sau đó có thể bao gồm các lớp convolutional bổ sung, lớp pooling hoặc lớp fully-connected. Các lớp đầu tiên trong mạng này giúp xác định các tính năng đơn giản trong dữ liệu. Khi qua mỗi lớp, Convolutional Neural Network tăng độ phức tạp của nó để xác định các tính năng lớn hơn và phức tạp hơn. Việc sử dụng Convolutional Neural Network mang đến một số lợi ích:

- Không cần giám sát của con người trong việc xác định các tính năng quan trọng: Mạng nơ-ron convolutional có khả năng tự động học và trích xuất các đặc trưng quan trọng từ dữ liệu, không yêu cầu sự can thiệp của con người trong việc xác định các đặc trưng cụ thể.
- Giảm thiểu số lượng tính toán so với các mạng thần kinh thông thường: Việc sử dụng các lớp convolutional và pooling giúp giảm kích thước không gian dữ liệu, từ đó giảm số lượng tính toán cần thiết, làm cho mạng nơ-ron convolutional có hiệu quả tính toán cao hơn so với các mạng thần kinh thông thường.
- Chia sẻ các trọng số trên các vùng tiếp nhận của một lớp: Một trong những đặc điểm đáng chú ý của mạng nơ-ron convolutional là khả năng chia sẻ các trọng số giữa các vùng tiếp nhận của một lớp. Điều này giúp giảm số lượng tham số trong mô hình, từ đó giúp mô hình trở nên hiệu quả và dễ huấn luyện hơn.

2.3.2. Mô hình Long Short-Term Memory (LSTM)

LSTM (Long Short-Term Memory) là một loại mạng nơ-ron hồi quy (RNN) được thiết kế để khắc phục vấn đề độ dốc biến mất (vanishing gradient), vốn thường xảy ra trong các RNN truyền thống. Điểm mạnh nổi bật của LSTM là khả năng xử lý và ghi nhớ thông tin trong khoảng thời gian dài mà không bị mất hiệu quả, giúp nó vượt trội hơn so với các mô hình như mạng RNN cơ bản, mô hình Markov ẩn (HMM), hoặc các phương pháp học chuỗi khác. LSTM hoạt

động như một bộ nhớ ngắn hạn nhưng có khả năng lưu trữ thông tin hàng ngàn bước thời gian, phù hợp để dự đoán cả ngắn hạn và dài hạn.

Cấu trúc chính của LSTM bao gồm một bộ nhớ (cell) để lưu trữ thông tin và ba cổng điều khiển: cổng quên (Forget Gate), cổng đầu vào (Input Gate), và cổng đầu ra (Output Gate). Cổng quên quyết định thông tin nào từ trạng thái trước đó cần được loại bỏ, dựa trên đầu vào hiện tại và trạng thái cũ, thông qua việc gán giá trị từ 0 (loại bỏ hoàn toàn) đến 1 (giữ lại toàn bộ). Cổng đầu vào xác định thông tin nào từ dữ liệu mới sẽ được thêm vào bộ nhớ. Cổng đầu ra điều chỉnh thông tin nào từ bộ nhớ hiện tại sẽ được sử dụng để tạo ra kết quả đầu ra, đảm bảo chỉ các thông tin quan trọng và phù hợp được chọn.

Nhờ khả năng chọn lọc và điều chỉnh thông tin qua các cổng, LSTM có thể duy trì các mối quan hệ dài hạn trong dữ liệu và thực hiện tốt các nhiệm vụ như dự đoán chuỗi thời gian hoặc xử lý ngôn ngữ tự nhiên (NLP). Điều này làm cho LSTM trở thành một giải pháp hiệu quả trong các bài toán yêu cầu phân tích dữ liệu chuỗi và học dài hạn.

2.3.3. Mô hình Gated Recurrent Unit kết hợp với Convolutional Neural Network (CNN+GRU)

CNN+GRU là một mô hình mạng nơ-ron kết hợp giữa Convolutional Neural Networks (CNN) và Gated Recurrent Units (GRU), được thiết kế để tận dụng điểm mạnh của cả hai loại mạng nhằm giải quyết các bài toán phức tạp, đặc biệt là những bài toán liên quan đến phân tích dữ liệu không gian và thời gian. Mô hình này hoạt động bằng cách sử dụng CNN để trích xuất đặc trưng không gian từ dữ liệu đầu vào, sau đó GRU sẽ xử lý các đặc trưng tuần tự, giúp mô hình học được các mối quan hệ theo thời gian và đưa ra dự đoán chính xác.

CNN đảm nhận vai trò trích xuất các đặc trưng quan trọng từ dữ liệu, đặc biệt là các dữ liệu có cấu trúc không gian như hình ảnh hoặc các chuỗi thời gian được ánh xạ không gian. Với các lớp convolutional và pooling, CNN có khả năng tự động học các mẫu (patterns) phức tạp, giảm kích thước dữ liệu và loại bỏ

nhiều không cần thiết, tạo ra một tập hợp các đặc trưng tối ưu để tiếp tục xử lý.

GRU, một dạng cải tiến của Recurrent Neural Networks (RNN), được sử dụng để mô hình hóa thông tin tuần tự và học các mối quan hệ theo thời gian. So với các RNN truyền thống, GRU khắc phục được vấn đề độ dốc biến mất (vanishing gradient), đồng thời đơn giản hóa kiến trúc so với LSTM bằng cách giảm số lượng cổng điều khiển. Cụ thể, GRU sử dụng hai cổng chính: cổng cập nhật (Update Gate) và cổng xóa (Reset Gate). Cổng cập nhật quyết định mức độ thông tin từ trạng thái trước đó cần được giữ lại để duy trì mối quan hệ dài hạn, trong khi cổng xóa kiểm soát việc xóa thông tin không cần thiết từ trạng thái trước đó. Nhờ đó, GRU có khả năng xử lý dữ liệu tuần tự hiệu quả mà không yêu cầu bộ nhớ phức tạp như LSTM.

Sự kết hợp giữa CNN và GRU trong một mô hình duy nhất cho phép tận dụng khả năng học sâu đặc trưng không gian của CNN và khả năng xử lý mối quan hệ thời gian của GRU. Điều này làm cho mô hình CNN+GRU trở thành một giải pháp lý tưởng cho các bài toán như phát hiện xâm nhập mạng, nhận dạng hình ảnh có yếu tố thời gian, hoặc các ứng dụng phân tích dữ liệu chuỗi phức tạp.

2.4. Mô hình mạng sinh đối kháng (GAN)

2.4.1. GAN là gì?

Mô hình mạng sinh đối kháng (Generative Adversarial Networks - GAN) là một mô hình sinh mẫu trong học máy, được quan tâm rất nhiều trong xu hướng áp dụng trí tuệ nhân tạo vào giải quyết các vấn đề trong đời sống hiện nay, từ nhận diện xử lý ảnh, tới các vấn đề bảo mật, an toàn thông tin cho các hệ thống. GAN thuộc nhóm generative model, trong đó generative là tính từ nghĩa là khả năng sinh ra, model nghĩa là mô hình. Vậy hiểu đơn giản generative model nghĩa là mô hình có khả năng sinh ra dữ liệu. Hay nói cách khác, GAN là mô hình

có khả năng sinh ra dữ liệu mới. Ví dụ như những ảnh mặt người ở Hình 2.2 chúng ta thấy là do GAN sinh ra, không phải mặt người thật. Dữ liệu sinh ra nhìn như thật nhưng không phải thật.



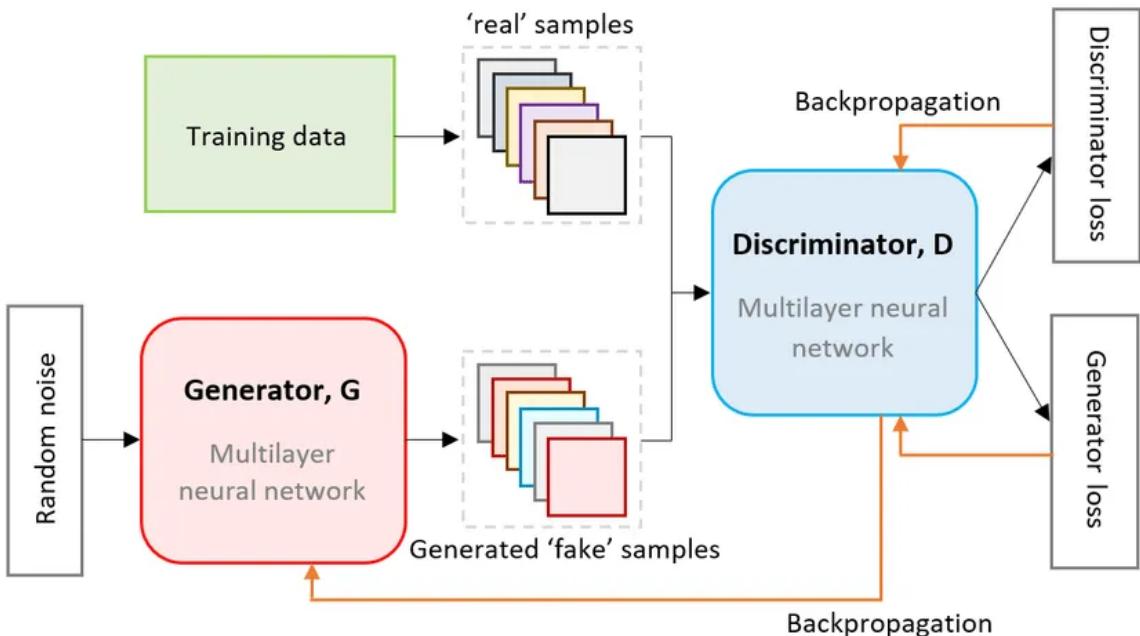
Hình 2.2: Ảnh mặt người sinh bởi GAN

2.4.2. Cấu trúc mạng GAN

Mô hình mạng GAN được cấu tạo bởi 2 mạng nơ-ron luôn hoạt động đối nghịch nhau: bộ sinh (Generator) và bộ phân biệt (Discriminator).

- Generator: Học cách sinh ra dữ liệu giả để lừa mô hình Discriminator. Để có thể đánh lừa được Discriminator thì đòi hỏi mô hình sinh ra output phải thực sự tốt. Do đó chất lượng dữ liệu phải càng như thật càng tốt.
- Discriminator: Học cách phân biệt giữa dữ liệu giả được sinh từ mô hình Generator với dữ liệu thật. Discriminator như một giáo viên chấm điểm cho Generator biết cách nó sinh dữ liệu đã đủ tinh xảo để qua mặt được Discriminator chưa và nếu chưa thì Generator cần tiếp tục phải học để tạo

ra ảnh thật hơn. Đồng thời Discriminator cũng phải cải thiện khả năng phân biệt của mình vì chất lượng ảnh được tạo ra từ Generator càng ngày càng giống thật hơn. Thông qua quá trình huấn luyện thì cả Generator và Discriminator cùng cải thiện được khả năng của mình.



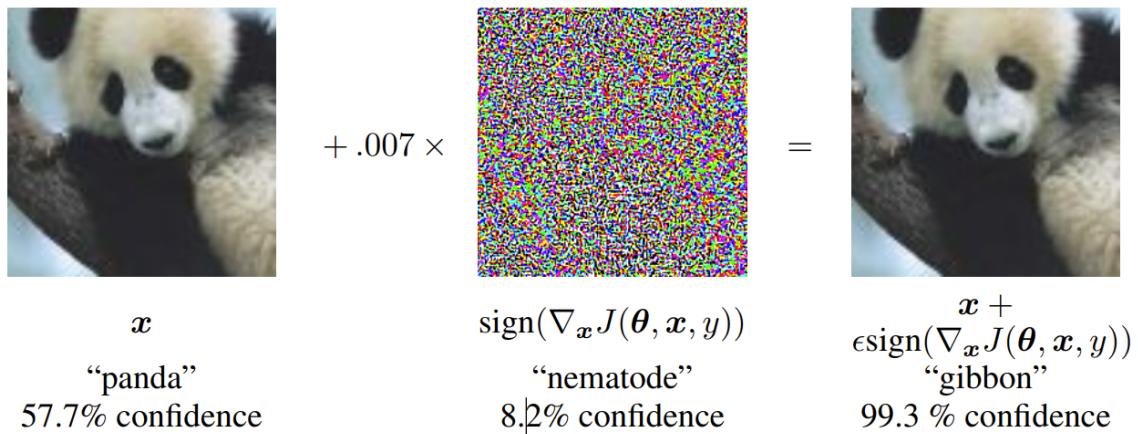
Hình 2.3: Các mạng trong GAN

2.5. Tấn công đối kháng (Adversarial Attack)

2.5.1. Nguyên lý hoạt động

Tấn công đối kháng là loại tấn công bằng cách thay đổi không đáng kể các giá trị đầu vào của tập dữ liệu nhằm làm suy yếu hoặc đánh lừa khả năng phân loại của IDS nhằm làm giảm hiệu năng của mô hình, đặc biệt là các mô hình học sâu (Deep Learning). Nó hoạt động bằng cách thêm một lượng nhiễu nhỏ, tinh vi vào dữ liệu đầu vào, khiến mô hình đưa ra dự đoán sai hoặc phản ứng không mong muốn, ngay cả khi dữ liệu đã bị thay đổi trông vẫn gần như không thay đổi đối với con người. Ví dụ như hình 2.4 bên dưới, ta có thể thấy bên trái

là hình ảnh một chú gấu trúc, mà mạng nơ-ron của chúng ta nhận dạng đúng là “gấu trúc” với độ tin cậy 57.7%. Sau khi thêm vào một chút nhiễu, mạng nơ-ron đó giờ đây nghĩ rằng đây là hình ảnh của một con vượn với độ tin cậy 99,3%!



Hình 2.4: Kết quả mô hình đưa ra kết quả phân loại sai sau khi thêm nhiễu vào mẫu ban đầu

2.5.2. Các loại tấn công đối kháng

- Dựa trên mức độ hiểu biết của kẻ tấn công, được phân thành:

- White-box Attack (Tấn công hộp trắng): Kẻ tấn công có toàn bộ thông tin về mô hình, như kiến trúc, trọng số, và hàm mất mát.
- Black-box Attack (Tấn công hộp đen): Kẻ tấn công không biết chi tiết về mô hình và chỉ có thể quan sát đầu vào và đầu ra để xây dựng tấn công.
- Gray-box Attack: Kẻ tấn công chỉ biết một phần thông tin về mô hình.

- Dựa trên phương thức thực hiện, được phân thành:

- Poisoning Attacks: Đây là loại tấn công nhắm vào giai đoạn huấn luyện của mô hình. Kẻ tấn công chèn các dữ liệu bị thao túng (nhiều hoặc độc hại) vào tập dữ liệu huấn luyện nhằm làm suy giảm hiệu suất của mô hình hoặc thậm chí thao túng hành vi của mô hình.

- Evasion Attacks: Đây là loại tấn công xảy ra trong giai đoạn suy luận (inference phase). Kẻ tấn công tinh chỉnh dữ liệu đầu vào để đánh lừa mô hình mà không cần thay đổi bản chất của dữ liệu. Mục đích làm mô hình đưa ra dự đoán sai (misclassification) hoặc né tránh được phát hiện.
- Model Extraction Attacks: Loại tấn công này nhầm tái tạo hoặc sao chép mô hình học máy thông qua các truy vấn, mà không cần truy cập trực tiếp vào mô hình hoặc dữ liệu huấn luyện của nó.

2.6. Các phương pháp phòng thủ mẫu đối kháng cho hệ thống phát hiện xâm nhập

2.6.1. Kernel Density Estimation

Kernel Density Estimation (KDE) [8] là một kỹ thuật thống kê được sử dụng để ước lượng hàm mật độ xác suất của dữ liệu dựa trên các mẫu quan sát. Trong bối cảnh phòng thủ trước tấn công đối kháng, KDE được áp dụng để phân biệt giữa các mẫu hợp lệ và các mẫu đối kháng bằng cách phân tích mật độ xác suất của dữ liệu đầu vào trong không gian đặc trưng.

Nguyên lý hoạt động:

- KDE xây dựng một hàm mật độ xác suất từ các mẫu huấn luyện hợp lệ. Hàm này được sử dụng để đánh giá khả năng xuất hiện của một mẫu đầu vào.
- Các mẫu đầu vào có mật độ xác suất thấp hơn một ngưỡng định trước (tức nằm ngoài vùng mật độ cao của dữ liệu hợp lệ) sẽ bị xem là bất thường hoặc đối kháng.

Hàm mật độ:

$$\hat{f}(x) = \frac{1}{|X_t|} \sum_{x_i \in X_t} k(x, x_i) \quad (2.1)$$

với $k(x, x_i)$ là hàm kernel, thường sử dụng Gaussian kernel:

$$k_\sigma(x, y) \sim \exp\left(-\frac{\|x - y\|^2}{\sigma^2}\right) \quad (2.2)$$

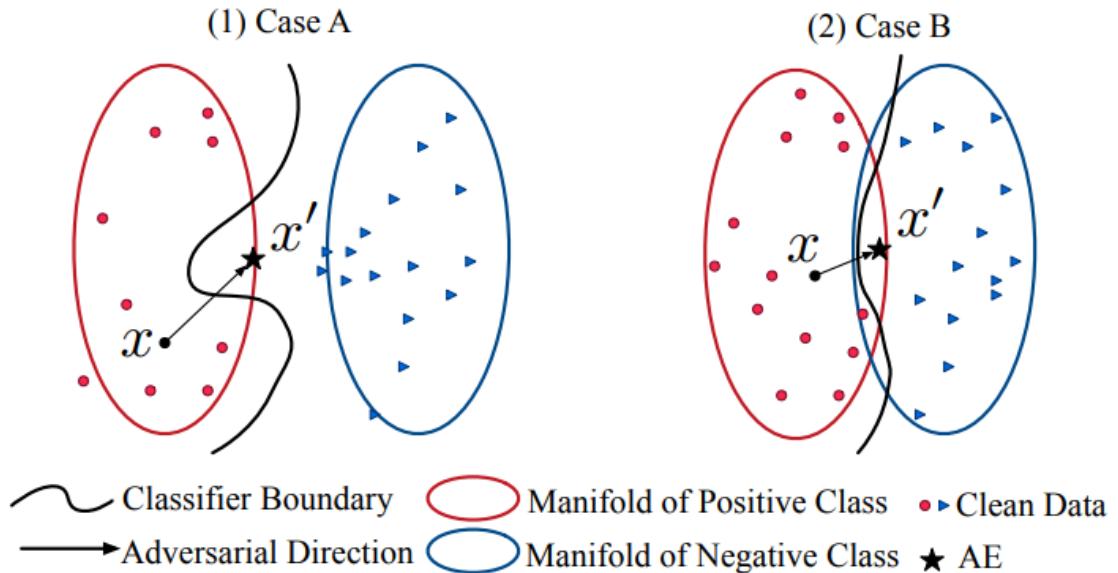
Trong đó, σ là băng thông được tinh chỉnh để tối ưu hóa hiệu suất.

2.6.2. MANDA

Theo như N. Wang và các cộng sự [11], MANDA là một cơ chế phát hiện AEs dựa trên sự phân loại AEs. Giả sử AEs cần giữ lại thuộc tính thiết yếu của lớp thực sự của nó, các AE thành công có thể được phân loại thành hai trường hợp như hình 2.5 (trong đó : x' là mẫu đối kháng (AE) được tạo ra từ đầu vào x sau khi thêm nhiễu).

- Trường hợp A: x' gần với đa tạp (manifold) của lớp 'malicious' nhưng xa đa tạp của lớp 'benign'. Điều này thường xảy ra khi các đa tạp ác tính và lành tính có thể tách biệt hoàn toàn với nhau.
- Trường hợp B: x' gần với cả hai đa tạp. Điều này xảy ra khi cả hai đa tạp gần nhau hoặc thậm chí chòng chéo (tức là không thể tách biệt hoàn toàn với nhau).

MANDA bao gồm hai thành phần, Manifold và DB (Decision Boundary). **Manifold** kết hợp cả kết quả phân loại và đánh giá đa tạp của x để phát hiện AE. Nếu hai đầu ra của x không nhất quán, thì x rất có thể là AE. **DB** kiểm tra xem đầu vào x có gần ranh giới quyết định của mô hình IDS để phát hiện AE hay không. Cụ thể, nếu chúng ta thêm nhiễu Gauss nhỏ vào x và y (y là confidence vector) tương ứng (và do đó kết quả dự đoán lớp thay đổi thường xuyên), thì x rất có thể là AE.



Hình 2.5: Phân loại AEs dựa trên 2 TH A và B

Algorithm 1 Score-Compute() for Criterion 1 & 2

Input: input $x \in \mathbb{R}^n$, IDS model $\mathcal{F}(\theta)$, learned manifold \mathcal{M}
Output: $score_1, score_2$

- 1: $p \leftarrow \mathcal{M}(x)$ # Confidence vector of manifold evaluation
- 2: $q \leftarrow \mathcal{F}(\theta, x)$ # Classifier output
- 3: $score_1 \leftarrow \|p\| + \|q\| - \|p + q\|$ # Criterion 1
- 4: **for** $i = 0$ to N **do**
- 5: $x_i = x + \mathcal{N}(0, \sigma^2)$
- 6: $p_i \leftarrow \mathcal{F}(\theta, x_i)$
- 7: **end for**
- 8: $score_2 \leftarrow \frac{1}{N} \sum_{i=1}^N \|p_i\| - \frac{1}{N} \left\| \sum_{i=1}^N p_i \right\|$ # Criterion 2
- 9: **return** $score_1, score_2$

Hình 2.6: Hàm Score-Compute() cho tiêu chí 1 và 2

2.6.2.1. Manifold

Để thực hiện, trước tiên cần tính toán score_1 bằng cách so sánh vector độ tin cậy từ việc đánh giá manifold và vector từ mô hình IDS như đã trình bày trong Thuật toán 1 (hình 2.6). Tiếp theo, so sánh score_1 với một ngưỡng tối ưu τ_1 để quyết định xem một mẫu đầu vào có phải là AE hay không (xem Thuật toán 2 hình 2.7). Ngưỡng τ_1 được chọn bằng cách kiểm tra các thống kê của các điểm dữ liệu sạch. Trong phần đánh giá, chúng tôi chọn phần trăm thứ 95 của các điểm dữ liệu sạch làm τ_1 .

Algorithm 2 Manifold

Input: input $x \in \mathbb{R}^n$, IDS model $\mathcal{F}(\theta)$, learned manifold \mathcal{M} , threshold τ_1

Output: $isAdversarial \in \{False, True\}$

- 1: $score_1, \sim \leftarrow \text{Score-Compute}(x, \mathcal{F}, \mathcal{M})$
 - 2: **if** ($score_1 > \tau_1$) **then**
 - 3: $isAdversarial \leftarrow True$
 - 4: **end if**
 - 5: **return** $isAdversarial$
-

Hình 2.7: Thuật toán 2 Manifold

2.6.2.2. DB(Decision Boundary)

DB cần đánh giá xem đầu vào có phải là mẫu gần ranh giới trong không gian nhiều chiều hay không. Để đạt được mục tiêu này, chúng tôi đánh giá độ không chắc chắn của đầu ra của mô hình IDS khi đầu vào được áp dụng với nhiễu động nhỏ. Đối với mẫu gần ranh giới, nhiễu động nhỏ như vậy có thể khiến nó đi qua ranh giới quyết định. Do đó, đầu ra của mô hình IDS trở nên rất không ổn định khi đầu vào được áp dụng với nhiễu động. Ngược lại, nhiễu động nhỏ trên đầu vào cách xa ranh giới khó có thể dẫn đến thay đổi như vậy. Chúng tôi tính toán

độ không chắc chắn của mô hình trên đầu vào với nhiễu động Gaussian $N(0, \sigma^2)$ trong Thuật toán 1. Đối với một đầu vào $x_i = x \pm N(0, \sigma^2)$, độ không chắc chắn của đầu ra từ mô hình IDS được đánh giá dưới dạng phương sai của vector độ tin cậy $\mathcal{F}(\theta, x_i)$:

$$\text{score}_2 = \frac{1}{N} \sum_{i=1}^N \|\mathcal{F}(\theta, x_i)\| - \frac{1}{N} \sum_{i=1}^N \|\mathcal{F}(\theta, x_i)\| . \quad (1)$$

Cũng như thuật toán Manifold, thuật toán DB (hình 2.8) đầu tiên tính toán score_2 bằng cách sử dụng Công thức (1). Tiếp theo, DB so sánh score_2 với ngưỡng tối ưu τ_2 để quyết định xem mẫu đầu vào có phải là AE hay không. Ngưỡng τ_2 được chọn tương tự như τ_1 .

Algorithm 3 DB

Input: input $x \in \mathbb{R}^n$, IDS model $\mathcal{F}(\theta)$, learned manifold \mathcal{M} , threshold τ_2

Output: $isAdversarial \in \{False, True\}$

- 1: $\sim, score_2 \leftarrow \text{Score-Compute}(x, \mathcal{F}, \mathcal{M})$
 - 2: **if** ($score_2 > \tau_2$) **then**
 - 3: $isAdversarial \leftarrow True$
 - 4: **end if**
 - 5: **return** $isAdversarial$
-

Hình 2.8: Thuật toán 3 DB

2.6.2.3. MANDA (Manifold+DB)

MANDA được thiết kế để tận dụng tối đa Manifold và DB cho việc phát hiện AE. Chúng tôi trước tiên kết hợp một số AE với tập dữ liệu sạch. Chúng tôi trộn các AE này với các đầu vào sạch và sử dụng X để biểu thị cho các đầu vào đã trộn. Tiếp theo, MANDA thu được $[score_1, score_2]$ của mỗi đầu vào bằng cách thực hiện Thuật toán 1. Chúng tôi gán nhãn 1 cho $[score_1, score_2]$ nếu đầu vào

là một AE và nhãn 0 nếu đầu vào không phải là AE. Sau đó, chúng tôi tạo ra một tập dữ liệu mới $[S_1, S_2, Y_{adv}]$ trong đó $[S_1, S_2]$ biểu thị cho điểm số cho mỗi đầu vào. Cuối cùng, MANDA huấn luyện một mô hình hồi quy logistic trên tập dữ liệu mới và sử dụng nó để phát hiện AE. Thuật toán 4 (hình 2.9) mô tả quy trình của MANDA.

Algorithm 4 MANDA

Input: IDS model $\mathcal{F}(\theta)$, learned manifold \mathcal{M} , test data point $x_{test} \in \mathbb{R}^n$, input dataset \mathbf{X} , AE flag \mathbf{Y}_{adv}

Output: $isAdversarial \in \{False, True\}$

```

1: if training then
2:    $S_1, S_2 \leftarrow \text{Score-Compute}(\mathbf{X}, \mathcal{F}, \mathcal{M})$ 
3:    $model \leftarrow \text{LogisticRegression}(S_1, S_2, \mathbf{Y}_{adv})$ 
4: else
5:    $score_1, score_2 \leftarrow \text{Score-Compute}(x_{test}, \mathcal{F}, \mathcal{M})$ 
6:    $isAdversarial \leftarrow model(score_1, score_2)$ 
7: end if
8: return  $isAdversarial$ 

```

Hình 2.9: Thuật toán 4 MANDA

2.6.3. Adversarial Training

Một trong những phương pháp phòng thủ phổ biến và hiệu quả là Adversarial Training được đề xuất bởi Goodfellow [3], một kỹ thuật được sử dụng để cải thiện khả năng chống chịu của mô hình học sâu trước các tấn công đối kháng. Adversarial Training hoạt động bằng cách bổ sung các mẫu đối kháng vào quá trình huấn luyện, giúp mô hình học cách xử lý dữ liệu bất thường một cách hiệu quả.

Phương pháp này được áp dụng bằng cách tạo các mẫu đối kháng từ dữ liệu gốc (trong nghiên cứu này, nhóm sử dụng CTGAN [12] để tạo ra các mẫu đối

kháng). Các mẫu này sau đó được sử dụng để huấn luyện mô hình cùng với dữ liệu ban đầu, nhằm đảm bảo rằng mô hình không chỉ học được các đặc trưng của dữ liệu hợp lệ mà còn học cách phân loại chính xác các mẫu bị tấn công.

Tuy Adversarial Training mang lại hiệu quả cao trong việc tăng cường độ bền vững của mô hình, phương pháp này cũng đi kèm một số hạn chế. Quá trình tạo và huấn luyện với mẫu đối kháng làm tăng đáng kể chi phí tính toán, đặc biệt khi áp dụng trên các tập dữ liệu lớn hoặc phức tạp. Ngoài ra, hiệu quả của phương pháp này phụ thuộc vào loại tấn công được mô phỏng, và mô hình có thể không hoạt động tốt trước các tấn công mà nó chưa được huấn luyện để chống lại. Dù vậy, Adversarial Training vẫn được coi là một phương pháp nền tảng để tăng cường khả năng phòng thủ của các hệ thống học sâu, đặc biệt trong các ứng dụng yêu cầu độ an toàn cao trước các tấn công tinh vi.

CHƯƠNG 3. THIẾT KẾ HỆ THỐNG

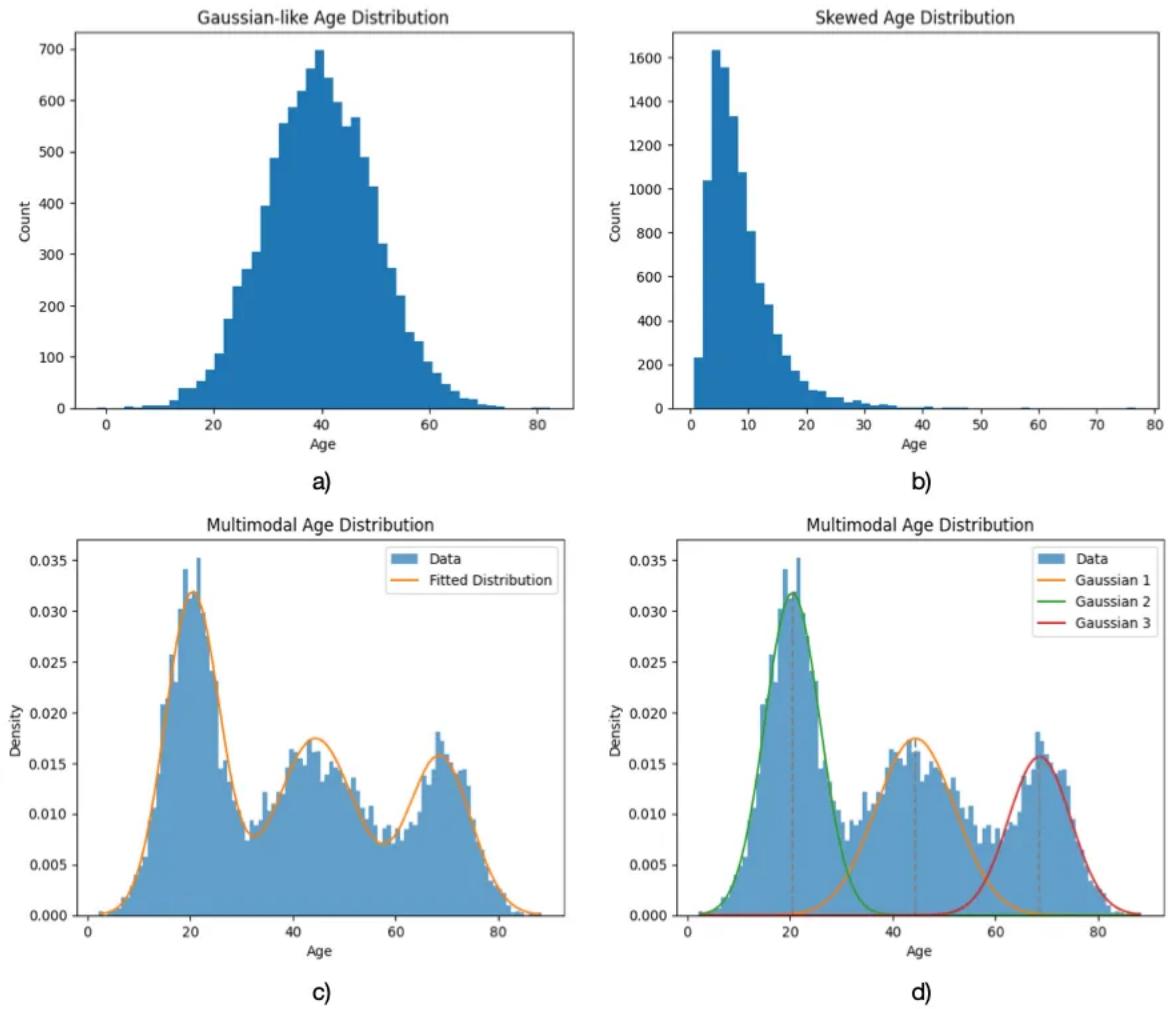
Ở chương này nhóm sẽ trình bày cách tạo ra dữ liệu đối kháng cũng như mô hình phát hiện tấn công đối kháng bằng mô hình học sâu.

3.1. Phát sinh dữ liệu đối kháng bằng tấn công đối kháng

Theo hướng tiếp cận của tấn công đối kháng, nhóm thực hiện dựa trên mô hình tấn công đối kháng là *Conditional Tabular Generative Adversarial Network (CTGAN)* [12]. Trước khi thực hiện đánh giá hiệu năng mô hình trước các cuộc tấn công đối kháng, nhóm thực hiện đưa ra lí thuyết tổng quan cho mô hình tấn công này.

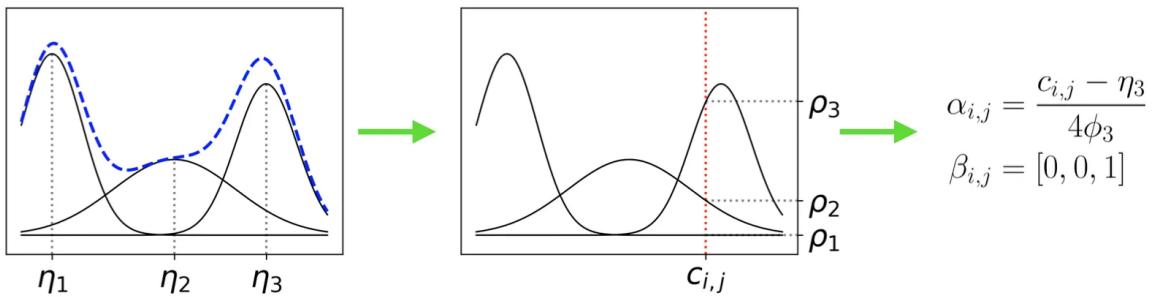
Thông thường nhất, GAN được sử dụng để tạo hình ảnh hoặc thậm chí là âm thanh/lời nói. Tuy nhiên, có rất nhiều lĩnh vực trong thế giới thực thường được mô tả bằng "dữ liệu dạng bảng", tức là dữ liệu có thể được cấu trúc và sắp xếp theo định dạng giống bảng. Theo tiêu chuẩn, các đặc trưng được biểu diễn trong các cột, trong khi các giá trị (hoặc "bản ghi") tương ứng với các hàng. Ngoài ra, dữ liệu trong thế giới thực thường bao gồm cả các đặc trưng số và phân loại. Các đặc trưng số (còn được gọi là "liên tục") là các đặc trưng mã hóa các giá trị định lượng, trong khi các đặc trưng phân loại (còn được gọi là "rời rạc") biểu diễn các phép đo định tính. CTGAN (Mạng đối kháng tạo bảng có điều kiện) được khái niệm hóa để "nắm bắt" một phần tính không đồng nhất này của dữ liệu trong thế giới thực và so với các kiến trúc khác như WGAN [1] và WGAN-GP [4], đã chứng minh là mạnh mẽ hơn và có thể khai quật hóa cho nhiều tập dữ liệu khác nhau. Phần "conditional" của CTGAN đề cập đến thực tế là nó cũng có thể tạo ra dữ liệu tổng hợp có điều kiện dựa trên một số biến đầu vào nhất định, chẳng hạn như giới tính hoặc độ tuổi.

Ngược lại với dữ liệu hình ảnh, trong đó các giá trị pixel thường tuân theo phân phối Gauss, các đặc trưng liên tục trong dữ liệu dạng bảng thường không phải là Gauss. Hơn nữa, chúng có xu hướng tuân theo phân phối đa đỉnh (như được mô tả trong Hình 3.1), tức là có nhiều "đỉnh" hoặc "cực đại cục" trong đồ thị phân phối xác suất.



Hình 3.1: Phân phối Gauss so với Phân phối MultiModal

Để nắm bắt những hành vi này, CTGAN sử dụng chuẩn hóa theo chế độ cụ thể (mode-specific normalization). Sử dụng mô hình VGM (Hỗn hợp Gauss biến đổi), mỗi giá trị trong một đặc trưng liên tục được biểu diễn bằng một vectơ one-hot biểu thị chế độ lấy mẫu của nó và một số vô hướng biểu diễn giá trị được chuẩn hóa theo chế độ đó:

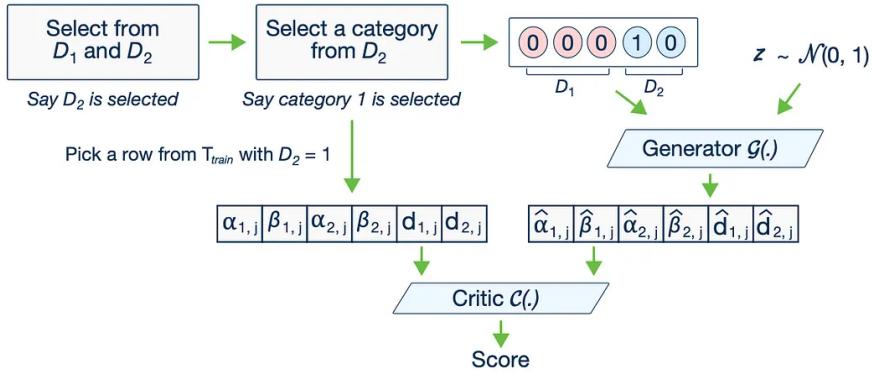


Hình 3.2: Ví dụ về mode-specific normalization

Mặt khác, CTGAN hướng đến giải quyết về cơ bản hai thách thức chính do các đặc trưng phân loại đưa ra. Thách thức đầu tiên là tính thừa thớt của các vectơ one-hot-encoded trong dữ liệu thực tế. Trong khi bộ tạo đưa ra các phân phối xác suất trên tất cả các giá trị phân loại có thể, thì các giá trị phân loại "thực" ban đầu được mã hóa trực tiếp trong một vectơ one-hot. Chúng có thể dễ dàng phân biệt được bằng bộ phân biệt bằng cách so sánh độ thừa thớt phân phối giữa dữ liệu thực và dữ liệu tổng hợp. Thách thức thứ hai là sự mất cân bằng liên quan đến một số đặc trưng phân loại. Nếu một số loại của một đặc trưng không được thể hiện đầy đủ, thì bộ tạo không thể học được chúng một cách đầy đủ. Nếu đây là một bài toán dự đoán phân loại, chúng ta có thể sử dụng các phương pháp như oversampling để khắc phục sự mất cân bằng. Tuy nhiên, mục tiêu của việc tạo dữ liệu tổng hợp không phải là cải thiện mô hình dự đoán mà là tái tạo lại các đặc điểm của dữ liệu gốc, nên oversampling không phải là giải pháp phù hợp.

CTGAN giới thiệu một bộ sinh có Điều Kiện (Conditional Generator) để giải quyết những thách thức do sự mất cân bằng giữa các đặc trưng phân loại gây ra, điều mà thường dẫn đến hiện tượng "mode collapse" tai tiếng của GAN. Tuy nhiên, các kiến trúc có điều kiện cũng gây ra sự đánh đổi đó là: dữ liệu đầu vào cần được chuẩn bị kỹ lưỡng để bộ sinh có thể hiểu và áp dụng các điều kiện, đồng thời các hàng dữ liệu được sinh ra cũng phải đảm bảo tuân thủ các điều kiện đã định.

Để đạt được mục đích này, CTGAN xem xét một vectơ có điều kiện, khi được sử dụng trong sample-by-sample training, sẽ tạo ra nhiều khác biệt liên quan đến việc sử dụng CTGAN:

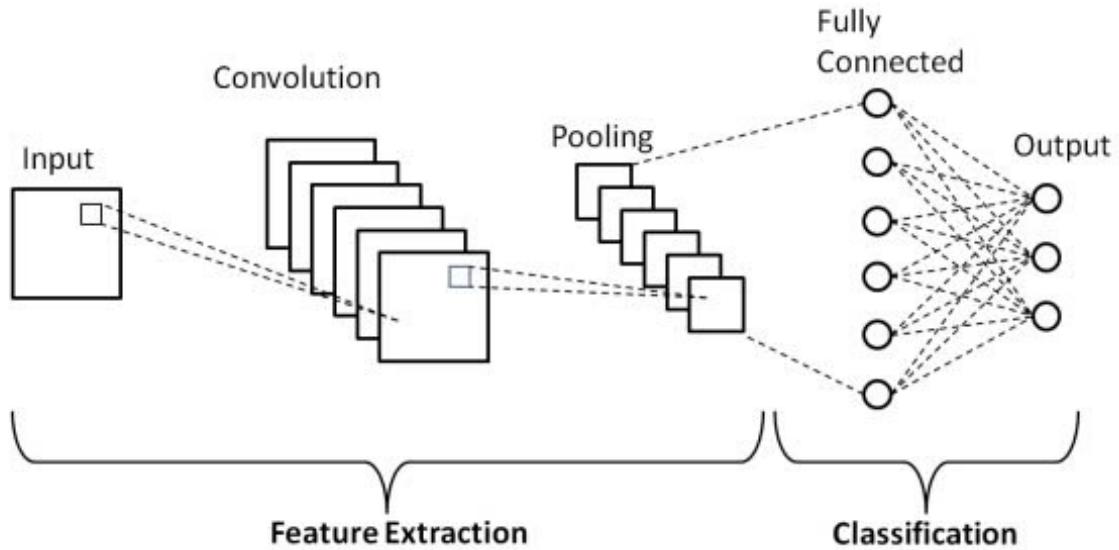


Hình 3.3: CTGAN Model

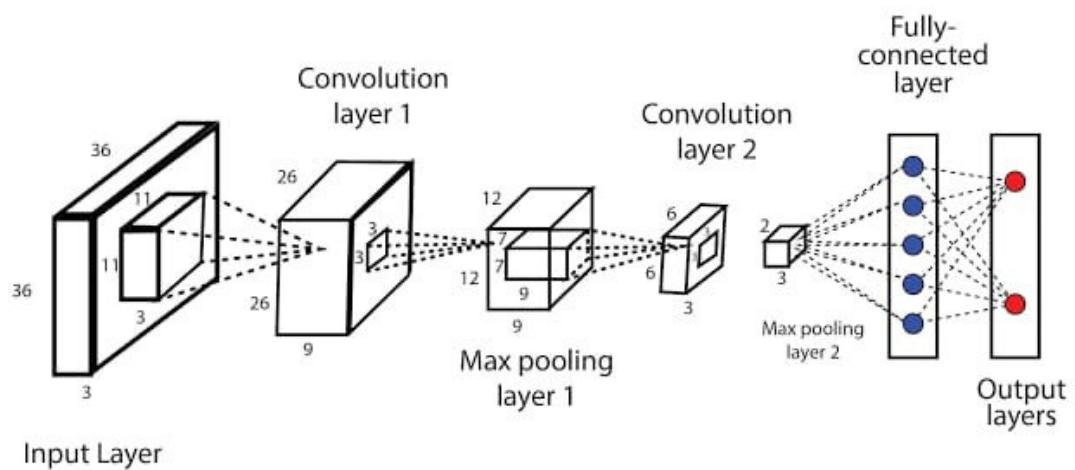
3.2. Xây dựng các hệ thống phát hiện xâm nhập dựa trên các mô hình học sâu

Nhóm xây dựng 5 mô hình học sâu cho việc phân loại tập dữ liệu Edge-IIoT, bao gồm CNN 3 layers, CNN 5 layers, CNN 7 layers, LSTM, CNN+GRU.

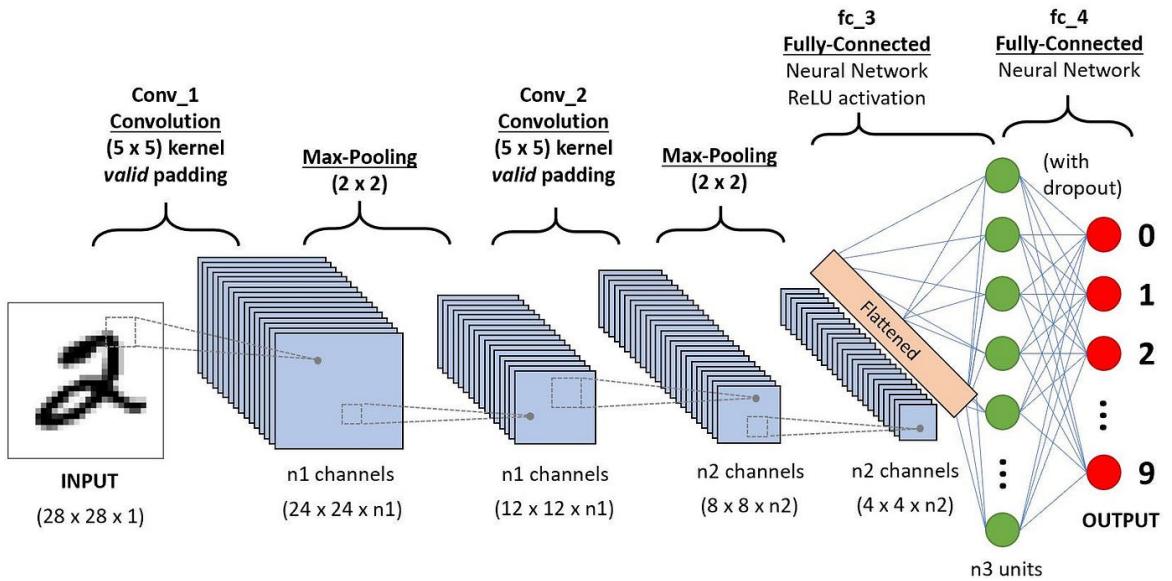
1. **CNN 3 Layers:** Kiến trúc CNN 3 lớp [7] được mô tả ở hình 3.4. Mô hình CNN 3 lớp được huấn luyện với tổng cộng 15 epochs, $batch_size = 256$, hàm Adam được dùng làm hàm tối ưu với $learningrate = 0.001$.
2. **CNN 5 Layers:** Kiến trúc CNN 5 lớp [9] được mô tả ở hình 3.5. Mô hình CNN 5 lớp được huấn luyện với tổng cộng 15 epochs, $batch_size = 256$, hàm Adam được dùng làm hàm tối ưu với $learningrate = 0.001$.
3. **CNN 7 Layers:** Kiến trúc CNN 7 lớp [10] được mô tả ở hình 3.6. Mô hình CNN 7 lớp được huấn luyện với tổng cộng 10 epochs, $batch_size = 64$, hàm Adam được dùng làm hàm tối ưu với $learningrate = 0.001$.



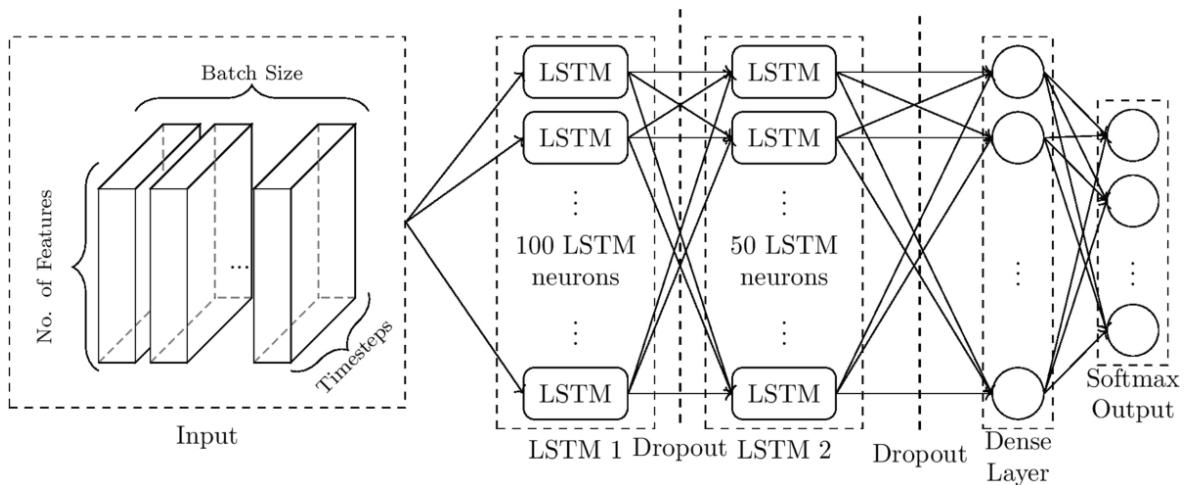
Hình 3.4: Kiến trúc CNN 3 lớp



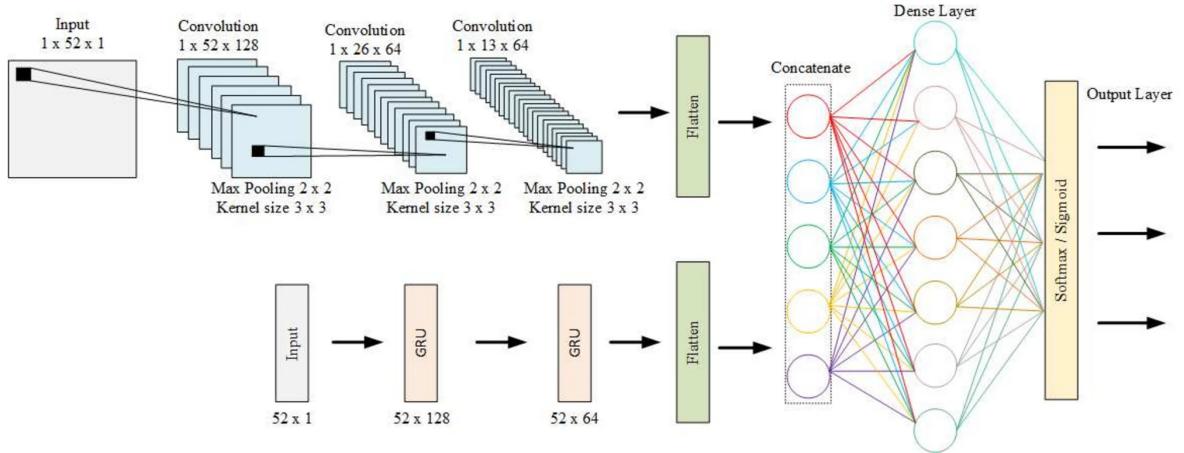
Hình 3.5: Kiến trúc CNN 5 lớp



Hình 3.6: Kiến trúc CNN 7 lớp



Hình 3.7: Kiến trúc LSTM



Hình 3.8: Kiến trúc CNN kết hợp với GRU

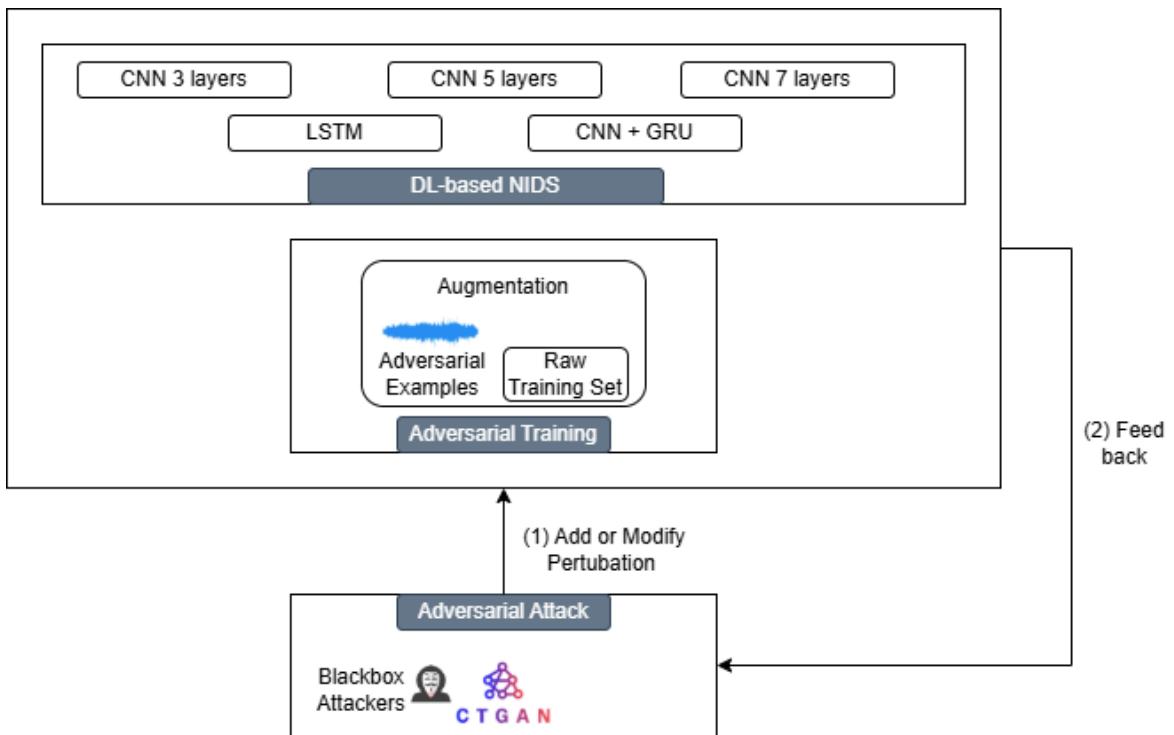
4. **LSTM:** Kiến trúc của LSTM [6] được mô tả ở hình 3.7. Mô hình LSTM được xây dựng với các lớp chính như sau: hai lớp LSTM liên tiếp và lớp Dense được sử dụng với số lượng đầu ra bằng số lớp cần phân loại, kèm theo hàm kích hoạt softmax để tạo phân phối xác suất cho các nhãn, được huấn luyện với tổng cộng 15 epochs, $batch_size = 16$. Mô hình hỗ trợ hai trình tối ưu hóa chính: SGD (Stochastic Gradient Descent) với $learningrate = 0.01$ và Adam Optimizer với $learningrate = 0.001$.

5. **CNN+GRU:** Kiến trúc của CNN+GRU [5] được mô tả ở hình 3.8. Mô hình CNN+GRU được huấn luyện với tổng cộng 10 epochs, $batch_size = 256$, hàm Adam được dùng làm hàm tối ưu với $learningrate = 0.001$.

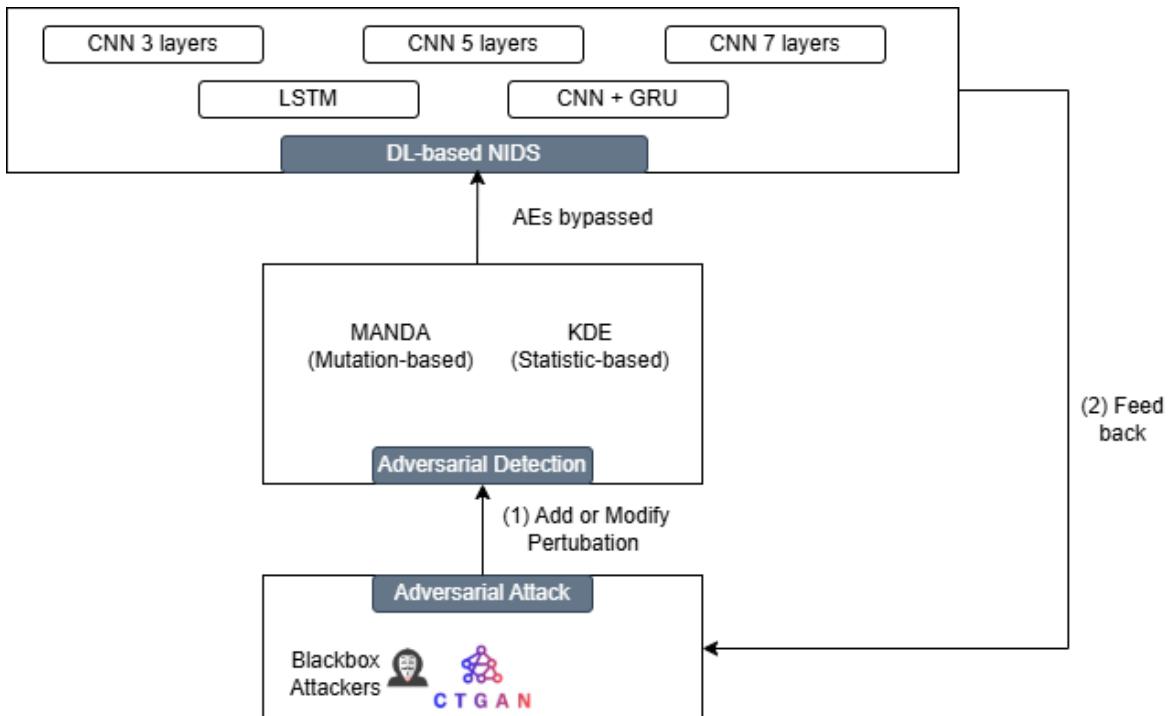
3.3. Tổng quan mô hình đề xuất

Bằng những trình bày ở mục 2.6, 3.1 và 3.2, nhóm đề xuất mô hình tấn công đối kháng vào các mô hình học sâu cùng với đó là các phương pháp phòng thủ kết hợp.

Cụ thể, mô hình bao gồm hệ thống phát hiện xâm nhập dựa trên học sâu (DL-based NIDS) với các thành phần chính như các kiến trúc mạng nơ-ron (CNN 3 layers, CNN 5 layers, CNN 7 layers, LSTM, và CNN + GRU). Hệ thống phải



Hình 3.9: Mô Hình về quá trình tấn công đối kháng vào mô hình học sâu được trang bị Adversarial Training



Hình 3.10: Mô Hình về quá trình tấn công đối kháng vào mô hình học sâu được trang bị Adversarial Detection

đối mặt với cuộc tấn công đối kháng được tạo ra từ CTGAN - được sử dụng để tạo ra các mẫu đối nghịch (AEs) nhằm vượt qua các mô hình NIDS.

Hệ thống bao gồm hai chiến lược chính để chống lại tấn công:

- **Adversarial Training:** Các mẫu đối kháng được tạo ra sẽ được thêm vào tập dữ liệu huấn luyện thông qua quá trình tăng cường dữ liệu (augmentation). Sau đó, mô hình được huấn luyện lại nhằm cải thiện khả năng chống chịu trước các cuộc tấn công.
- **Adversarial Detection:** Sử dụng hai phương pháp phát hiện mẫu đối kháng:
 - MANDA: Dựa trên sự kết hợp giữa manifold và decision boundary (phương pháp dựa trên biến đổi - Mutation-based).
 - KDE: Dựa trên phương pháp ước lượng mật độ xác suất (phương pháp dựa trên thống kê - Statistics-based).

Từ đó dẫn đến hai kịch bản như sau:

- **Kịch bản 01:** Theo hình 3.9, ban đầu kẻ tấn công đưa dữ liệu luồng mạng đầu vào là các mẫu đối kháng được tạo ra từ CTGAN tới mô hình IDS học sâu. Sau đó, kẻ tấn công sẽ nhận được feedback từ mô hình, ví dụ như một gói tin ACK phản ánh liệu rằng luồng mạng có được phân loại là bất thường hay không. Và dựa trên feedback này, kẻ tấn công có thể thêm nhiều hoặc điều chỉnh nhiều trên dữ liệu đầu vào để tối ưu hóa việc né tránh phát hiện. Những mẫu đối kháng này sau đó cũng được đưa vào tập huấn luyện để tăng cường khả năng chống chịu của mô hình IDS học sâu.
- **Kịch bản 02:** Theo hình 3.10, ban đầu kẻ tấn công cũng đưa dữ liệu luồng mạng đầu vào là các mẫu đối kháng được tạo ra từ CTGAN tới mô hình IDS học sâu. Lúc này, trước khi dữ liệu đi vào IDS, nó sẽ phải đi qua bộ phát hiện đối kháng là MANDA hoặc KDE. Những mẫu đối kháng bị phát

hiện sẽ bị loại bỏ, còn những mẫu vượt qua được sẽ tiếp tục đi vào mô hình IDS, sau đó kẻ tấn công nhận được feedback từ mô hình IDS và thực hiện việc tinh chỉnh tinh vi dữ liệu đầu vào để tối ưu hóa khả năng né tránh.

CHƯƠNG 4. THÍ NGHIỆM VÀ ĐÁNH GIÁ

Ở chương này nhóm tiến hành tạo môi trường, cài đặt và đưa ra các tiêu chí đánh giá về mức độ hiệu quả của mô hình.

4.1. Môi trường thực nghiệm

4.1.1. Tài nguyên

- CPU: Intel Core i5-13500H
- RAM: 16 GB
- Hệ điều hành: Windows 11
- Bộ nhớ SSD 512 GB

Môi trường phát triển

- Trình soạn thảo: Visual Studio Code.
- Ngôn ngữ lập trình: Python3.
- Nền tảng học sâu: Keras Tensorflow.
- Nền tảng CTGAN: Synthetic Data Vault (SDV).
- Thư viện sử dụng: numpy, pandas, tensorflow, sklearn,...

4.1.2. Tập dữ liệu

Trong phần thực nghiệm, nhóm chúng tôi chọn một trong số các bộ dữ liệu mới nhất dùng để huấn luyện các mô hình IDS học sâu, có tên là Edge-IIoT dataset.

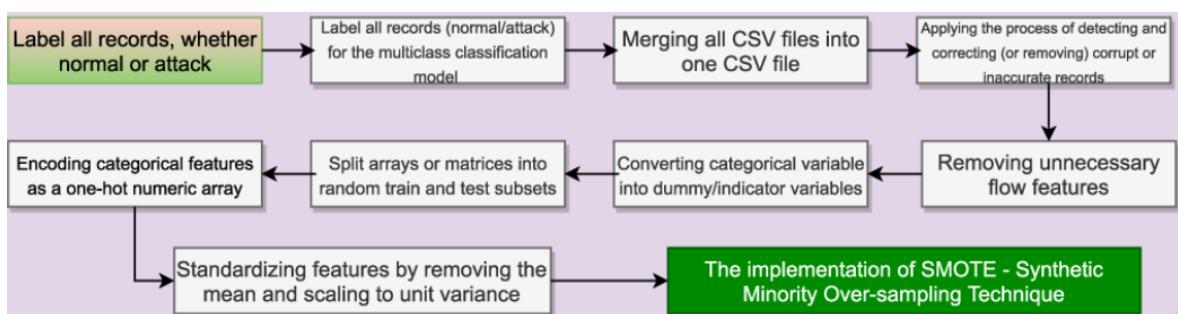
Cụ thể nhóm sử dụng tập DNN-Edge-IIoT, đây là tập các lưu lượng mạng được trích xuất bằng công cụ Tshark và Zeek, có chứa 63 thuộc tính (bao gồm cả nhãn) với tổng cộng hơn 2.2 triệu bản ghi. Trong đó, số bản ghi chứa lưu lượng mạng tấn công (attack) chiếm gần 53% và 47% còn lại là các mẫu lành tính (benign). Các mẫu lưu lượng mạng độc hại trên được thu thập dựa trên 14 loại tấn công khác nhau như: Backdoor, DDoS_HTTP, DDos_TCP, DDos_UDP, DDoS_ICMP, Uploading, Man-In-The-Middle, Password, Ransomware, Port Scanning, XSS, Fingerprinting, SQL_injection và Vulnerability_scanner.

4.1.3. Tiền xử lý dữ liệu

Quá trình xử lý và phân tích tập dữ liệu DNN-Edge-IIoT [2] (mô tả trong hình 4.1) sẽ bao gồm các bước sau:

- Bước 1: Thêm một nhãn mới có tên là **Attack_label** để gắn nhãn cho tất cả các bản ghi, phân loại chúng thành hai nhóm: normal hoặc attack. Nhãn Attack_label có giá trị 0 hoặc 1, được sử dụng trong mô hình phân loại nhị phân.
- Bước 2: Thêm một nhãn mới có tên là **Attack_type**, biểu thị các loại tấn công, được sử dụng trong mô hình phân loại đa lớp.
- Bước 3: Loại bỏ các bản ghi trùng lặp và các giá trị bị thiếu như **NaN** (Not A Number) hoặc **INF** (Infinite Value).
- Bước 4: Loại bỏ các thuộc tính không cần thiết như địa chỉ IP, cổng, dấu thời gian và thông tin payload.
- Bước 5: Sử dụng gói **pandas.get_dummies** để chuyển đổi các biến phân loại thành các biến dummy/indicator
- Bước 6: Sử dụng **train_test_split** từ gói **sklearn.model_selection** để chia dữ liệu thành tập huấn luyện và tập kiểm tra một cách ngẫu nhiên.

- Bước 7: Sử dụng ***OneHotEncoder*** từ gói ***sklearn.preprocessing*** để mã hóa các biến phân loại thành mảng số dạng one-hot.
- Bước 8: Áp dụng ***StandardScaler*** từ gói ***sklearn.preprocessing*** để chuẩn hóa các đặc trưng bằng cách loại bỏ giá trị trung bình và tỷ lệ theo phương sai đơn vị.
- Bước 9: Triển khai SMOTE - Synthetic Minority Over-sampling Technique để giải quyết việc mất cân bằng lớp.



Hình 4.1: Quá trình tiền xử lý tập dữ liệu Edge-IIoT

Kết thúc quá trình xử lý dữ liệu, tập DNN-Edge-IIoT chúng tôi sử dụng chứa 48 thuộc tính bao gồm cả thuộc tính Label.

4.2. Kết quả thí nghiệm

Ở mục này, tôi sẽ trình bày các kết quả thực nghiệm và đưa ra đánh giá.

Chúng tôi tập trung trả lời câu hỏi sau:

Sự khác biệt về hiệu suất của các mô hình phát hiện xâm nhập dựa trên học sâu và tỉ lệ trốn tránh của các mẫu đối kháng trước và sau khi thực hiện tấn công đối kháng, cũng như trước và sau khi áp dụng các biện pháp phòng thủ.

4.2.1. Kết quả xây dựng các mô hình học sâu phát hiện tấn công

Chúng tôi thực hiện đánh giá hiệu năng học sâu khi thực hiện phân loại các mẫu từ tập dữ liệu ban đầu trong bộ dữ liệu DNN-Edge-IIoT. Chúng tôi thực

hiện chia bộ dữ liệu DNN-Edge-IIoT thành các thành phần như sau:

- 80% dữ liệu dành cho tập Train
- 20% dữ liệu dành cho tập Test, phục vụ cho mục đích đánh giá hiệu năng của mô hình và sinh mẫu đối kháng.

Bảng 4.1 đánh giá kết quả của mô hình IDS dựa trên 4 tiêu chí: *Accuracy*, *Precision*, *Recall* và *F1-Score*. Các mô hình học sâu được sử dụng để đánh giá bao gồm CNN 3 layers, CNN 5 layers, CNN 7 layers, LSTM, CNN + GRU.

Model	Accuracy	Precision	Recall	F1-Score
CNN 3 Layers	0.9454	0.9569	0.9454	0.9397
CNN 5 Layers	0.9493	0.9547	0.9493	0.9460
CNN 7 Layers	0.9482	0.9613	0.9482	0.9427
LSTM	0.9650	0.9600	0.9650	0.9427
CNN + GRU	0.7143	0.5102	0.7143	0.5952

Bảng 4.1: Kết quả đánh giá của mô hình học sâu cho tập dữ liệu DNN-Edge-IIoT

Quan sát ta có thể thấy hiệu suất của các mô hình học sâu đều cao khi tất cả các chỉ số đánh giá đều trung bình trên 0.94. Tuy nhiên, đối với mô hình CNN kết hợp GRU, lại có các chỉ số hơi thấp so với các mô hình còn lại, điều này có thể do các siêu tham số dùng để huấn luyện mô hình chưa được tối ưu hoặc do mô hình CNN kết hợp GRU bị overfitting khi huấn luyện đối với tập dữ liệu Edge-IIoT. Nhóm sẽ cố gắng tối ưu hóa hiệu suất của mô hình này trong tương lai.

4.2.2. Tỉ lệ trốn tránh của các mẫu đối kháng được tạo ra từ CT-GAN khi chưa có biện pháp phòng thủ

Chúng tôi sử dụng chỉ số ASR(Attack Success Rate) để đánh giá về tỉ lệ trốn tránh của các mẫu đối kháng. ASR được tính như sau:

$$ASR = \frac{S}{N} \times 100$$

trong đó S biểu thị số lượng cuộc tấn công thành công (tức là các AE bị phân loại sai) và N đại diện cho tổng số AE sử dụng để thực hiện tấn công.

Bảng 4.2 mô tả tỉ lệ trốn tránh thành công của các mẫu đối kháng sinh ra từ CTGAN, có thể thấy tỉ lệ ở mức khá cao, tỉ lệ cao nhất là 59.10% cho mô hình CNN 3 lớp, và thấp nhất là 33.94% đối với mô hình LSTM. Có thể thấy khi chưa có các biện pháp phòng thủ đối với các mẫu đối kháng thì mô hình IDS có khả năng cao bị đánh lừa dẫn đến việc phân loại sai.

Model	ASR
CNN 3 Layers	59.10%
CNN 5 Layers	35.46%
CNN 7 Layers	40.37%
LSTM	33.94%
CNN + GRU	54.11%

Bảng 4.2: Tỉ lệ trốn tránh của các mẫu đối kháng đối với các mô hình IDS

4.2.3. Kết quả về hiệu suất của các mô hình phát hiện xâm nhập dựa trên học sâu và tỉ lệ trốn tránh của các mẫu đối kháng khi áp dụng các biện pháp phòng thủ

4.2.3.1. Adversarial Training

Để thực hiện phương pháp phòng thủ Adversarial Training, chúng tôi thực hiện sinh 381935 mẫu đối kháng từ CTGAN nhằm phục vụ cho mục đích thêm vào tập dữ liệu huấn luyện các mô hình IDS và kiểm tra khả năng chống lại các mẫu đối kháng của phương pháp phòng thủ này. Tỉ lệ chia tập huấn luyện và kiểm tra là 8:2, tương đương với 305548 mẫu huấn luyện và 76387 mẫu kiểm tra.

Kết quả bảng 4.3 cho thấy tỷ lệ trốn tránh của các mẫu đối kháng tăng nhẹ ở các mô hình CNN 5 Layers (35.46% lên 35.79%) và LSTM (33.94% lên 36.20%), có thể do kiến trúc chưa đủ phức tạp hoặc khả năng xử lý tuần tự của LSTM làm tăng tính nhạy cảm với mẫu đối kháng mới. Ngược lại, tỷ lệ này giảm nhiều ở

Model	Without Retrain	Retrain
CNN 3 Layers	59.10%	59.49%
CNN 5 Layers	35.46%	35.79%
CNN 7 Layers	40.37%	35.17%
LSTM	33.94%	36.20%
CNN + GRU	54.11%	44.49%

Bảng 4.3: Tỉ lệ trốn tránh của các mẫu đối kháng trước và sau khi áp dụng Adversarial Retraining

CNN 7 Layers (40.37% xuống 35.17%) và CNN + GRU (54.11% xuống 44.49%), cho thấy các mô hình phức tạp hơn đã tận dụng tốt Adversarial Training để cải thiện khả năng chống lại các mẫu đối kháng, từ đó tăng cường tính ổn định và hiệu quả của mô hình trong các môi trường đối kháng.

Model	Without retrain				Retrain			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
CNN 3 layers	0.9454	0.9569	0.9454	0.9397	0.9420	0.9541	0.9420	0.9349
CNN 5 layers	0.9493	0.9547	0.9493	0.9460	0.9439	0.9536	0.9439	0.9388
CNN 7 layers	0.9482	0.9613	0.9482	0.9427	0.9438	0.9475	0.9438	0.9422
LSTM	0.9485	0.9600	0.9485	0.9436	0.9481	0.9506	0.9481	0.9470
CNN + GRU	0.7143	0.5102	0.7143	0.5952	0.6748	0.4554	0.6748	0.5438

Hình 4.2: Hiệu suất chung của các mô hình IDS dựa trên học sâu trước và sau khi áp dụng phương pháp Retraining

Về hiệu suất của các mô hình IDS dựa trên học sâu trước và sau khi Retraining được mô tả trong hình 4.2, xu hướng chung là tất cả các chỉ số hiệu suất (Accuracy, Precision, Recall, F1) của các mô hình đều giảm. Đối với các mô hình CNN (3, 5, 7 layers), sự giảm sút là tương đối nhỏ, nhưng vẫn đáng chú ý, đặc biệt ở chỉ số Precision và F1. Với LSTM, dù chỉ số Precision giảm nhẹ (từ

0.9600 xuống 0.9506), nhưng các chỉ số khác như F1 thì tăng nhẹ. Đáng chú ý nhất là mô hình CNN + GRU, nơi hiệu suất giảm mạnh, đặc biệt ở Precision (từ 0.5102 xuống 0.4554), dẫn đến sự suy giảm đáng kể trong F1. Tuy nhiên, việc retraining đã góp phần làm giảm tỷ lệ trốn tránh của các mẫu đối kháng, như đã được thể hiện ở bảng 4.3. Điều này cho thấy rằng retraining, mặc dù có thể làm suy giảm hiệu suất tổng thể trên dữ liệu thông thường, lại giúp tăng cường khả năng phòng thủ của mô hình trước các mẫu đối kháng. Đây là một sự đánh đổi phổ biến trong việc áp dụng các biện pháp bảo vệ như Adversarial Training, khi ưu tiên tính mạnh mẽ của mô hình trong các môi trường đối kháng hơn là hiệu suất trên dữ liệu sạch.

4.2.3.2. Adversarial Detection

Model	KDE	MANDA
CNN 3 Layers	53.2%	55.3%
CNN 5 Layers	36.5%	50.2%
CNN 7 Layers	33.5%	31.8%
LSTM	30.5%	29.1%
CNN + GRU	51.1%	40.2%

Bảng 4.4: Tỉ lệ trốn tránh của các mẫu đối kháng khi áp dụng KDE và MANDA cho mô hình IDS

Kết quả trong bảng 4.4 thể hiện tỉ lệ trốn tránh của các mẫu đối kháng được tạo ra từ CTGAN khi áp dụng hai biện pháp phòng thủ phát hiện adversarial là KDE và MANDA trên các mô hình IDS khác nhau. Phương pháp MANDA, dựa trên sự kết hợp giữa manifold và decision boundary, đạt tỉ lệ trốn tránh cao nhất trên CNN 3 Layers (55.3%), nhưng hiệu quả giảm dần khi áp dụng trên các mô hình phức tạp hơn, với tỉ lệ trốn tránh thấp nhất là 29.1% trên LSTM. Trong khi đó, phương pháp KDE, dựa trên ước lượng mật độ xác suất, cũng cho kết quả cao nhất trên CNN 3 Layers (53.2%) và thấp nhất cũng ở mô hình LSTM với (30.5%). Đáng chú ý, các mô hình nhiều lớp hơn (CNN 5 Layers và CNN 7 Layers) có tỉ lệ trốn tránh thấp hơn đáng kể so với mô hình CNN 3

Layers, đặc biệt khi áp dụng phương pháp KDE (lần lượt 36.5% và 33.5%), cho thấy rằng việc tăng số lớp có thể góp phần làm suy giảm khả năng tránh của các mẫu đối kháng.

Dựa vào kết quả, ta thấy rằng khả năng phát hiện các mẫu đối kháng của MANDA có phần vượt trội hơn so với KDE, nguyên nhân có thể là do KDE hoạt động dựa trên việc ước lượng mật độ xác suất của dữ liệu huấn luyện gốc và so sánh với mật độ của các mẫu mới, điều đó dẫn tới việc là nếu dữ liệu gốc có phân phối hẹp, KDE có thể không phát hiện được các mẫu đối kháng gần với phân phối gốc. Do đó, KDE thích hợp sử dụng khi tập dữ liệu nhỏ, có phân phối đơn giản và không nhiều chiều. Đối với trường hợp của MANDA, nó khai thác hiệu quả hai yếu tố chính: manifold và decision boundary. MANDA sử dụng manifold để nắm bắt cấu trúc nội tại của dữ liệu, trong đó các mẫu đối kháng thường tạo ra sự không nhất quán giữa kết quả đánh giá manifold và kết quả phân loại của hệ thống phát hiện xâm nhập (IDS). Cụ thể, trong trường hợp các manifold của hai lớp (malicious và benign) tách biệt (Case A), mẫu đối kháng thường nằm gần manifold của lớp “malicious” nhưng cách xa manifold của lớp “benign”. MANDA có khả năng phát hiện sự không nhất quán này, điều mà KDE – chỉ tập trung vào mật độ xác suất – không thể làm tốt được.

Ngoài ra, MANDA còn đánh giá gần đúng khoảng cách của mẫu đầu vào đến decision boundary, đặc biệt hiệu quả trong các trường hợp manifold của hai lớp chồng lấn hoặc gần nhau (Case B). Trong trường hợp này, các mẫu đối kháng thường rất gần ranh giới quyết định và dễ bị tác động bởi các nhiễu nhỏ. MANDA đánh giá mức độ bất ổn định của đầu ra mô hình IDS khi đầu vào bị nhiễu nhẹ, từ đó phát hiện các mẫu đối kháng gần ranh giới. Điều này giúp MANDA có khả năng phát hiện chính xác hơn so với KDE, vốn chỉ dựa vào phân phối mật độ và không tính đến cấu trúc manifold hoặc ranh giới quyết định. Nhờ việc kết hợp cả hai yếu tố trên, MANDA có khả năng phát hiện các mẫu đối kháng hiệu quả hơn và giảm tỷ lệ trốn tránh của chúng, như được minh chứng qua kết quả thực nghiệm.

CHƯƠNG 5. KẾT LUẬN

Ở chương này, chúng tôi đưa ra những kết luận về nghiên cứu và đồng thời đưa ra hướng phát triển trong tương lai dựa trên nghiên cứu này.

5.1. Kết luận

Nghiên cứu về tính chuyển giao của mẫu đối kháng được sinh ra từ CTGAN góp phần đánh giá độ mạnh mẽ của các mô hình phát hiện xâm nhập dựa trên học sâu. Ngoài ra, nghiên cứu này cũng giúp đánh giá các phương pháp phòng thủ chống lại mẫu đối kháng được đề xuất trước đó, nhằm đem lại một kết quả khách quan về các phương pháp phòng thủ đang hiện hữu.

Qua việc xây dựng và đánh giá tính chuyển giao của các mẫu đối kháng nhằm đánh lừa hệ thống phát hiện xâm nhập và các phương pháp phòng thủ, chúng tôi đã hiểu sâu hơn về các hướng nghiên cứu liên quan, hiểu được các hạn chế để góp phần cải thiện dần. Đồ án này đã đạt được những kết quả sau:

- Tìm hiểu về các mô hình phát hiện xâm nhập dựa trên học sâu và cách xây dựng chúng
- Tạo các mẫu đối kháng từ CTGAN với đầu vào là các mẫu luồng tấn công mạng của tập dữ liệu Edge-IIoT
- Nghiên cứu các phương pháp phòng thủ chống lại mẫu đối kháng và tiến hành đánh giá tính chuyển giao của mẫu đối kháng đối với các phương pháp phòng thủ đó

Kết quả thu được qua thực nghiệm cho thấy tỉ lệ trốn tránh ở mức cao của các mẫu đối kháng khi chưa có các biện pháp phòng thủ, chứng tỏ độ hiệu quả của

các mẫu đối kháng sinh ra từ CTGAN hoạt động tốt trên các mô hình phát hiện xâm nhập. Ngoài ra, khi áp dụng các phương pháp phòng thủ, tỉ lệ trốn tránh của các mẫu đối kháng có xu hướng giảm, chứng tỏ độ hiệu quả của các phương pháp phòng thủ.

5.2. Hướng phát triển

- Nâng cao và phát triển thêm các phương pháp phòng thủ chống tấn công đối kháng hiệu quả hơn.
- Phát triển các phương pháp tấn công mới, nghiên cứu các kỹ thuật tấn công tiên tiến hơn như tấn công không gian đặc trưng (feature space attacks) hoặc các tấn công dựa trên mạng đối kháng phức tạp hơn (multi-modal GANs) để tạo ra adversarial examples có tính chuyên giao cao.
- Áp dụng thêm các phương pháp bảo vệ trước các sự tấn công đối kháng như Ensemble Learning, Gradient Masking, ...

TÀI LIỆU THAM KHẢO

Tiếng Anh:

- [1] Martin Arjovsky, Soumith Chintala, and Léon Bottou, *Wasserstein GAN*, 2017, arXiv: 1701.07875 [stat.ML], URL: <https://arxiv.org/abs/1701.07875>.
- [2] Mohamed Amine Ferrag et al. (2022), “Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning”, *IEEE Access*, 10, pp. 40281–40306, DOI: 10.1109/ACCESS.2022.3165809.
- [3] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy (2014), “Explaining and harnessing adversarial examples”, *arXiv preprint arXiv:1412.6572*.
- [4] Ishaan Gulrajani et al., *Improved Training of Wasserstein GANs*, 2017, arXiv: 1704.00028 [cs.LG], URL: <https://arxiv.org/abs/1704.00028>.
- [5] Beibei Li et al. (2021), “DeepFed: Federated Deep Learning for Intrusion Detection in Industrial Cyber–Physical Systems”, *IEEE Transactions on Industrial Informatics*, 17 (8), pp. 5615–5624, DOI: 10.1109/TII.2020.3023430.
- [6] Roghayeh Mojarad et al. (2023), “A hybrid and context-aware framework for normal and abnormal human behavior recognition”, *Soft Comput.*, 28 (6), 4821–4845, ISSN: 1432-7643, DOI: 10.1007/s00500-023-09188-4, URL: <https://doi.org/10.1007/s00500-023-09188-4>.

- [7] Van Hiep Phung and Eun Joo Rhee (2019), “A High-Accuracy Model Average Ensemble of Convolutional Neural Networks for Classification of Cloud Image Patches on Small Datasets”, *Applied Sciences*, URL: <https://api.semanticscholar.org/CorpusID:207975537>.
- [8] Saurabh Shintre Andrew B. Gardner Reuben Feinman Ryan R. Curtin (2017), “Detecting Adversarial Samples from Artifacts”, *Adversarial attacks against deep learning-based network intrusion detection systems and defense mechanisms*.
- [9] Iqbal H. Sarker (2021), “Machine Learning: Algorithms, Real-World Applications and Research Directions”, *SN Comput. Sci.*, 2 (3), DOI: 10.1007/s42979-021-00592-x, URL: <https://doi.org/10.1007/s42979-021-00592-x>.
- [10] Fathma Siddique, Shadman Sakib, and Md. Abu Bakr Siddique, “Recognition of Handwritten Digit using Convolutional Neural Network in Python with Tensorflow and Comparison of Performance for Various Hidden Layers”, in: *2019 5th International Conference on Advances in Electrical Engineering (ICAEE)*, 2019, pp. 541–546, DOI: 10.1109/ICAEE48663.2019.8975496.
- [11] Ning Wang et al. (2022), “Manda: On adversarial example detection for network intrusion detection system”, *IEEE Transactions on Dependable and Secure Computing*, 20 (2), pp. 1139–1153.
- [12] Lei Xu et al., “Modeling Tabular data using Conditional GAN”, in: *Neural Information Processing Systems*, 2019, URL: <https://api.semanticscholar.org/CorpusID:195767064>.