HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



Phan Thị Kim Hoàn

NGHIÊN CỨU CÁC PHƯƠNG PHÁP SINH DỮ LIỆU TẤN CÔNG MẠNG

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

HÀ NỘI – 2025

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Phan Thị Kim Hoàn

NGHIÊN CỨU CÁC PHƯƠNG PHÁP SINH DỮ LIỆU TẤN CÔNG MẠNG

Chuyên ngành: Hệ thống thông tin
Mã số: 8.48.01.04

ĐỀ ÁN ĐỒ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

NGƯỜI HƯỚNG DẪN KHOA HỌC

PGS. TS. HOÀNG XUÂN DẬU

HÀ NỘI - 2025

LỜI CAM ĐOAN

Tôi cam đoan đây là công trình nghiên cứu của riêng tôi.

Các số liệu, kết quả nêu trong đề án tốt nghiệp là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.

Tác giả đề án tốt nghiệp

Phan Thị Kim Hoàn

LỜI CẢM ƠN

Trong suốt quá trình học tập và thực hiện đề án Thạc sĩ tại Học viện Công nghệ Bưu chính Viễn thông, em đã nhận được rất nhiều sự quan tâm, hỗ trợ và hướng dẫn quý báu từ các thầy cô, cơ quan, bạn bè và gia đình.

Trước hết, em xin bày tỏ lòng biết ơn sâu sắc tới **PGS. TS. Hoàng Xuân Dậu**, người thầy đã tận tình định hướng, truyền đạt kiến thức khoa học quý báu và luôn theo sát quá trình thực hiện đề tài. Những gợi ý, chỉ dẫn và sự hỗ trợ của thầy đã giúp em tháo gỡ nhiều khó khăn, hoàn thiện đề tài theo đúng mục tiêu nghiên cứu. Em cũng xin chân thành cảm ơn **ThS. Ninh Thị Thu Trang và KS. Mai Xuân Nhật** đã dành nhiều thời gian tư vấn, hỗ trợ, giải đáp những vướng mắc trong quá trình triển khai và thực nghiệm.

Bên cạnh đó, em xin gửi lời cảm ơn đến các thầy cô của Học viện Công nghệ Bưu chính Viễn thông đã giảng dạy, truyền đạt những kiến thức nền tảng và nâng cao trong suốt thời gian học tập.

Cuối cùng, em xin cảm ơn gia đình, bạn bè và đồng nghiệp đã luôn động viên, hỗ trợ và chia sẻ cùng em trong suốt thời gian học tập và thực hiện đề án.

Em xin trân trọng cảm ơn!

MỤC LỤC

I. MỞ ĐẦU 1

1. Lý do chọn đề tài 1
2. Tổng quan về vấn đề nghiên cứu 1
3. Mục đích nghiên cứu 3
4. Đối tượng và phạm vi nghiên cứu 4
5. Phương pháp nghiên cứu 4

II. NỘI DUNG 5

CHƯƠNG 1 : TỔNG QUAN VỀ SINH DỮ LIỆU TẤN CÔNG MẠNG 5

1.1. Tổng quan về dữ liệu tấn công mạng 5
1.2. Các bộ dữ liệu tấn công mạng phổ biến 8
1.3. Mục tiêu và nhu cầu của việc sinh dữ liệu tấn công mạng 12
1.4. Thách thức trong sinh dữ liệu tấn công mạng 13
1.5. Các nghiên cứu liên quan về sinh dữ liệu tấn công mạng 14
1.6. Kết chương 16

CHƯƠNG 2 : CÁC KỸ THUẬT SINH DỮ LIỆU TẤN CÔNG MẠNG 17

2.1. Khái quát về các phương pháp sinh dữ liệu tấn công mạng 17
2.2. Các phương pháp sinh dữ liệu truyền thống 18
2.3. Sinh dữ liệu bằng phương pháp học máy, học sâu 24
2.4. Kết chương 34

CHƯƠNG 3 : THỬ NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ 35

3.1. Mô hình thử nghiệm 35
3.2. Các kịch bản thử nghiệm và kết quả 37
3.3. Nhận xét và đánh giá chung 45
3.4. Kết chương 46

III. KẾT LUẬN 47

1. Kết luận chung 47
2. Hướng nghiên cứu tiếp theo 47

IV. DANH MỤC CÁC TÀI LIỆU THAM KHẢO 49

DANH MỤC BẢNG BIỂU

Bảng 1.1: Các loại tấn công trong UNSW- NB15 8
Bảng 1.2: Mẫu lưu lượng trong tập dữ liệu UNSW-NB15 9
Bảng 1.3: Các loại tấn công trong CIC-UNSW-NB15 Augmented DataSet 9
Bảng 1.4: Các loại tấn công trong CIC-IDS-2017 10
Bảng 1.5: Mẫu lưu lượng trong tập dữ liệu CIC-IDS-2017 10
Bảng 1.6: Bảng so sánh các bộ dữ liệu UNSW-NB15, CIC-IDS-2017, NSL-KDD, BoT-IoT, TON-IoT. 11
Bảng 2.1: Bảng so sánh ưu nhược điểm và độ phổ biến các nhóm phương pháp sinh dữ liệu 18
Bảng 3.1: Các tập ORIGIN_SET trích từ tập dữ liệu UNSW-NB15 38
Bảng 3.2: Hiệu suất phát hiện của mô hình ORIGIN_MODEL và các mô hình GEN_MODEL cho các nhãn tấn công trên tập dữ liệu UNSW-NB15 39
Bảng 3.3: Các tập ORIGIN_SET trích từ tập dữ liệu CIC-UNSW-NB15 40
Bảng 3.4: Hiệu suất phát hiện của mô hình ORIGIN_MODEL và các mô hình GEN_MODEL cho các nhãn tấn công trên CIC-UNSW-NB15 40
Bảng 3.5: Các tập ORIGIN_SET trích từ tập dữ liệu CIC-IDS-2017 41
Bảng 3.6: Hiệu suất phát hiện của mô hình ORIGIN_MODEL và các mô hình GEN_MODEL cho các nhãn tấn công trên tập dữ liệu CIC-IDS-2017 41
Bảng 3.7: Các thành phần của tập dữ liệu UNSW-NB15 42
Bảng 3.8: Các thành phần của các tập dữ liệu UNSW-NB15-ORIGIN 43
Bảng 3.9: Các thành phần của các tập dữ liệu UNSW-NB15-GEN 44
Bảng 3.10: Hiệu suất phát hiện cho các nhãn trên bộ dữ liệu UNSW-NB15-ORIGIN 44
Bảng 3.11: Hiệu suất phát hiện cho các nhãn trên bộ dữ liệu UNSW-NB15-GEN 45

DANH MỤC HÌNH ẢNH

Hình 2.1: Ví dụ thuật toán SMOTE với k = 4 20
Hình 2.2: Mô hình mạng GAN 25
Hình 2.3: Kiến trúc WGAN 27
Hình 2.4: Mô hình generator của cGAN 29
Hình 2.5: Sơ đồ kết hợp giữa generator và discriminator của cGAN 30
Hình 3.1: Sơ đồ mô hình thực hiện đánh giá phương pháp sinh dữ liệu 35

DANH MỤC CÁC THUẬT NGỮ, TỪ VIẾT TẮT

STT	Chữ viết tắt	Nguyên nghĩa	Nghĩa tiếng Việt
1	ACCS	Australian Centre for Cyber Security	Trung tâm An ninh mạng Úc
2	CIC	Canadian Institute for Cybersecurity	Viện An ninh mạng Canada
3	CGAN	Conditional Generative Adversarial Network	Mạng GAN có điều kiện
4	CNN	Convolutional Neural Network	Mạng nơ-ron tích chập
5	CopulaGAN	Copula-based Generative Adversarial Network	GAN sử dụng mô hình Copula để sinh dữ liệu
6	CTGAN	Conditional Tabular Generative Adversarial Network	Mạng GAN có điều kiện dành cho dữ liệu dạng bảng
7	DDoS	Distributed Denial of Service	Tấn công từ chối dịch vụ phân tán
8	FNR	False Negative Rate	Tỷ lệ âm tính giả
9	FPR	False Positive Rate	Tỷ lệ dương tính giả
10	FTP	File Transfer Protocol	Giao thức truyền tệp
11	HTTP	HyperText Transfer Protocol	Giao thức truyền tải siêu văn bản
12	IDS	Intrusion Detection System	Hệ thống phát hiện xâm nhập
13	IoT	Internet of Things	Internet vạn vật
14	LLMs	Large Language Models	Mô hình ngôn ngữ lớn
15	ML	Machine Learning	Học máy
16	NIDS	Network Intrusion Detection System	Hệ thống phát hiện xâm nhập mạng
17	RNN	Recurrent Neural Network	Mạng nơ-ron hồi quy
18	SCADA	Supervisory Control And Data Acquisition	Hệ thống điều khiển giám sát và thu thập dữ liệu
19	SMOTE	Synthetic Minority Over-sampling Technique	Kỹ thuật tăng cường mẫu thiểu số tổng hợp

20	SSH	Secure shell	Giao thức kết nối và điều khiển từ xa an toàn
21	SVM	Support Vector Machine	Máy vector hỗ trợ
22	TCP/IP	Transmission Control Protocol / Internet Protocol	Bộ giao thức điều khiển truyền vận/IP
23	VAEs	Variational Autoencoders	Bộ tự mã biến phân
24	WGAN	Wasserstein Generative Adversarial Network	Mạng GAN dùng khoảng cách Wasserstein
25	WGAN-CP	Wasserstein GAN with Gradient Penalty	WGAN với cơ chế phạt gradient

I. MỞ ĐẦU

1. Lý do chọn đề tài

Trong bối cảnh không gian mạng ngày càng phức tạp và tiềm ẩn nhiều nguy cơ, các cuộc tấn công mạng đang gia tăng nhanh chóng cả về quy mô lẫn mức độ tinh vi. Để đối phó với các mối đe dọa này, các hệ thống phát hiện tấn công, xâm nhập mạng (Intrusion Detection Systems – IDS) đóng vai trò then chốt trong việc giám sát, cảnh báo và bảo vệ hệ thống thông tin khỏi các hành vi bất thường.

Trên thực tế, các hệ thống IDS thường sử dụng các tập luật (rule) để phát hiện các tấn công, xâm nhập. Các luật là các mô tả về các tấn công, xâm nhập đã biết, được mã hóa theo các chuẩn. Các hệ thống IDS dựa trên luật có khả năng phát hiện nhanh và chính xác các tấn công đã biết. Tuy vậy, chúng không thể phát hiện các tấn công, xâm nhập mới, hoặc biến thể của các tấn công đã biết do các luật phát hiện các tấn công này chưa được xây dựng. Ngoài ra, việc xây dựng tập luật thường được các chuyên gia an ninh mạng xây dựng thủ công, nên tốn nhiều thời gian, công sức. Gần đây, hướng tích hợp các kỹ thuật học máy, học sâu để xây dựng các IDS hay các mô hình phát hiện tấn công, xâm nhập đang được quan tâm nghiên cứu và đạt nhiều kết quả khả quan. Các mô hình phát hiện tấn công dựa trên học máy, học sâu có khả năng "học" từ dữ liệu huấn luyện và có khả năng phát hiện các dạng tấn công, xâm nhập đã biết và chưa biết.

Tuy nhiên, một trong những thách thức lớn nhất trong việc phát triển và huấn luyện các hệ thống IDS dựa trên học máy, học sâu là là sự thiếu hụt dữ liệu tấn công có chất lượng cao, có tính đại diện và đa dạng. Dữ liệu tấn công thực tế thường khan hiếm, và không cân bằng giữa các lớp (tấn công – bình thường), hoặc không phản ánh đầy đủ các hình thái tấn công hiện đại. Điều này làm giảm hiệu quả học của các mô hình, đặc biệt là các mô hình học máy và học sâu, vốn phụ thuộc vào dữ liệu đầu vào.

Trước những hạn chế nêu trên, sinh dữ liệu tấn công mạng (network attack data generation) được xem là một hướng tiếp cận khả thi nhằm giải quyết bài toán thiếu dữ liệu và mất cân bằng dữ liệu. Việc tạo ra các mẫu dữ liệu nhân tạo có đặc tính tương tự dữ liệu thật giúp mở rộng tập huấn luyện, cân bằng phân phối dữ liệu và tăng cường khả năng phát hiện tấn công trong các hệ thống IDS.

2. Tổng quan về vấn đề nghiên cứu

2.1. Các phương pháp sinh dữ liệu tấn công mạng

Sự gia tăng nhanh chóng và mức độ tinh vi của các cuộc tấn công mạng đặt ra thách thức lớn cho hệ thống phát hiện xâm nhập (IDS). Việc huấn luyện IDS hiệu quả đòi hỏi bộ dữ liệu đa dạng, cân bằng và cập nhật, nhưng thực tế lại thiếu hụt dữ liệu công khai, khó thu thập mẫu tấn công hiếm và lưu lượng mạng ngày càng được mã hóa. Do đó, nghiên cứu sinh dữ liệu tấn công mạng trở thành hướng tiếp cận quan trọng nhằm

bổ sung hoặc thay thế dữ liệu thực, cải thiện khả năng phát hiện của IDS. Các phương pháp sinh dữ liệu phổ biến và đạt được những kết quả tốt bao gồm:

- Các phương pháp truyền thống – tiêu biểu là SMOTE [2], tạo mẫu mới bằng cách nội suy giữa điểm thuộc lớp thiểu số và các láng giềng gần nhất. Kỹ thuật này đơn giản, hiệu quả trong cân bằng dữ liệu, nhưng hạn chế ở chỗ không sinh được dữ liệu cho các đặc trưng phức tạp – điều cần thiết khi các hình thức tấn công ngày càng tinh vi.
- Các phương pháp dựa trên học sâu với GAN với các biến thể như CTGAN, WGAN, CopulaGAN, cho phép sinh dữ liệu bảng có phân phối gần giống dữ liệu thật và có thể điều khiển theo nhãn tấn công. CTGAN được thiết kế để xử lý cả thuộc tính liên tục và rời rạc, đồng thời hỗ trợ sinh dữ liệu có điều kiện theo nhãn tấn công.
- Các phương pháp dựa trên luật (Rule-based) sinh dữ liệu từ các quy tắc logic mô tả hành vi tấn công, dựa trên kiến thức chuyên gia hoặc phân tích dữ liệu thực. Ưu điểm của phương pháp này là khả năng kiểm soát chặt chẽ kịch bản sinh dữ liệu, phù hợp với các môi trường đặc thù. Tuy nhiên, tính đa dạng bị giới hạn và khó tái tạo các hành vi bất thường chưa từng thấy.
- Các phương pháp dựa trên mô phỏng (Simulation-based) thiết lập môi trường mạng giả lập và triển khai tấn công bằng công cụ như Metasploit, LOIC, Hping; sau đó thu thập lưu lượng qua Wireshark, tcpdump. Dữ liệu thu thập được từ quá trình này thường rất sát với thực tế và có giá trị cao cho huấn luyện IDS. Nhược điểm của phương pháp này là yêu cầu hạ tầng kỹ thuật phức tạp, thời gian triển khai dài và chi phí lớn.

Nhìn chung, mỗi phương pháp sinh dữ liệu mang lại những giá trị riêng, và xu hướng hiện nay là kết hợp nhiều kỹ thuật để tận dụng ưu điểm của từng phương pháp, từ đó tạo ra bộ dữ liệu vừa đa dạng, vừa sát với hành vi tấn công trong thực tế. Dữ liệu tổng hợp từ các phương pháp này, khi kết hợp với các mô hình học máy (Decision Tree, Random Forest, XGBoost, MLP), giúp mô hình phát hiện có độ phủ cao hơn, nâng cao độ chính xác, phát hiện tấn công hiếm và giảm cảnh động giả.

2.2. Khảo sát một số nghiên cứu liên quan

Trong những năm gần đây, nhiều nghiên cứu đã áp dụng các phương pháp sinh dữ liệu tấn công mạng để cải thiện hiệu quả của hệ thống phát hiện xâm nhập (IDS), đặc biệt nhằm giải quyết vấn đề mất cân bằng và thiếu hụt dữ liệu. Dưới đây là khảo sát một số nghiên cứu liên quan đến đề án.

Ferriyan và cộng sự [4] giới thiệu bộ dữ liệu HIKARI-2021, kết hợp lưu lượng mạng thực và tấn công tổng hợp, bao gồm cả lưu lượng mã hóa. Dữ liệu được tạo trong môi trường mạng thật, thu thập bằng Wireshark và tcpdump, với nhiều loại tấn công sinh ra từ các công cụ như Metasploit và Hping3. Kết quả cho thấy HIKARI-2021 giúp

cải thiện đáng kể Recall và F1-score ở các lớp tấn công hiếm, cung cấp nguồn dữ liệu đa dạng và sát thực tế cho nghiên cứu IDS.

Fan và cộng sự [5] nghiên cứu tác động của các kỹ thuật lấy mẫu oversampling (SMOTE, ADASYN) và undersampling (RandomUnderSampler, Instance Hardness Threshold) đối với hiệu quả phát hiện xâm nhập mạng trên bộ dữ liệu IoTID20. Kết quả cho thấy việc không áp dụng lấy mẫu khiến các lớp tấn công thiểu số bị bỏ sót nghiêm trọng, trong khi SMOTE kết hợp với XGBoost đạt hiệu năng tốt nhất, với độ chính xác trên 98% và cải thiện rõ rệt Recall ở các lớp hiếm.

Theo hướng tiếp cận khác sử dụng học máy, học sâu trong sinh dữ liệu tấn công mạng, Zhao và các tác giả [6] tập trung vào việc áp dụng các biến thể của Generative Adversarial Networks (GAN), gồm Vanilla GAN, Wasserstein GAN (WGAN) và Conditional Tabular GAN (CTGAN), để sinh dữ liệu tấn công bổ sung từ bộ CIC-IDS2017. Mục tiêu là giải quyết tình trạng mất cân bằng dữ liệu, đặc biệt ở các lớp tấn công hiếm như botnet. Kết quả thực nghiệm cho thấy dữ liệu sinh ra có phân phối tương đồng với dữ liệu thật và khi kết hợp với dữ liệu gốc để huấn luyện hệ thống phát hiện xâm nhập mạng (NIDS) đã giúp cải thiện đáng kể các chỉ số Precision, Recall và F1-score, đạt Precision gần 1.00 trong nhiều trường hợp. Nghiên cứu khẳng định tiềm năng của GAN trong việc nâng cao hiệu suất phát hiện, nhất là đối với các loại tấn công ít xuất hiện trong dữ liệu huấn luyện.

Một trong những thách thức lớn của hệ thống phát hiện xâm nhập trong môi trường IoT là tình trạng mất cân bằng dữ liệu, khi số lượng mẫu tấn công – đặc biệt là các loại tấn công hiếm – quá ít so với lưu lượng bình thường. Điều này làm suy giảm khả năng huấn luyện và hiệu suất phát hiện của mô hình. Nghiên cứu của Goodfellow và cộng sự [3] đề xuất giải pháp sử dụng GAN để sinh thêm dữ liệu tấn công tổng hợp, giúp cân bằng tập huấn luyện. Phương pháp được áp dụng trên các bộ dữ liệu NSL-KDD, UNSW-NB15 và BoT-IoT, kết hợp dữ liệu gốc và dữ liệu sinh để huấn luyện mô hình ensemble. Kết quả cho thấy độ chính xác đạt tới 99.56% với phân loại nhị phân và 98.14% với phân loại đa lớp, đồng thời cải thiện rõ rệt khả năng phát hiện tấn công hiếm và giảm tỷ lệ báo động giả.

Từ các công trình trên có thể thấy, xu hướng hiện nay là tích hợp nhiều phương pháp sinh dữ liệu nhằm tận dụng ưu điểm của từng kỹ thuật, khắc phục hạn chế của phương pháp đơn lẻ và tạo ra các bộ dữ liệu vừa đa dạng, vừa sát với thực tế, phục vụ hiệu quả cho huấn luyện IDS.

3. Mục đích nghiên cứu

Mục tiêu chính của đề tài bao gồm:

- Hệ thống hóa và phân loại các kỹ thuật sinh dữ liệu tấn công mạng hiện nay.

- Lựa chọn và triển khai các kỹ thuật sinh dữ liệu tiêu biểu theo hướng tiếp cận: phương pháp sinh dữ liệu truyền thống và sinh dữ liệu dựa trên học máy, học sâu.
- Thực nghiệm các kỹ thuật đã chọn trên các bộ dữ liệu thực tế trong lĩnh vực mạng máy tính.
- Đánh giá hiệu quả của từng phương pháp thông qua các tiêu chí định lượng và khả năng hỗ trợ các mô hình phát hiện tấn công mạng.

4. Đối tượng và phạm vi nghiên cứu

- Đối tượng nghiên cứu: Các phương pháp sinh dữ liệu có thể áp dụng trong lĩnh vực phát hiện tấn công mạng, đặc biệt là các kỹ thuật phục vụ cho việc mở rộng và cân bằng dữ liệu huấn luyện.
- Phạm vi nghiên cứu:
    + Tập trung ba kỹ thuật sinh dữ liệu: SMOTE, CGAN và WGAN
    + Áp dụng trên hai bộ dữ liệu mạng công khai phổ biến là UNSW-NB15 và CIC-IDS-2017.
    + Sử dụng mô hình học máy để đánh giá chất lượng dữ liệu sinh ra.

5. Phương pháp nghiên cứu

Đề tài được thực hiện thông qua các phương pháp nghiên cứu chính sau:

- Phân tích tài liệu: Nghiên cứu, tổng hợp các công trình khoa học, bài báo liên quan đến lĩnh vực sinh dữ liệu và an ninh mạng.
- Cài đặt và thực nghiệm: Triển khai các kỹ thuật sinh dữ liệu được lựa chọn trên các bộ dữ liệu thực tế. Tiến hành tiền xử lý, huấn luyện và đánh giá dữ liệu sinh ra.
- Đánh giá và so sánh: Sử dụng các tiêu chí như phân phối dữ liệu, độ đa dạng, độ giống dữ liệu thật và ảnh hưởng đến độ chính xác của mô hình phát hiện tấn công để so sánh các phương pháp.

6. Bố cục của đề án

Đề án này được bố cục thành 3 chương với các nội dung chính như sau:

Chương 1 giới thiệu tổng quan dữ liệu tấn công mạng, thách thức và các nghiên cứu liên quan trong lĩnh vực sinh dữ liệu tấn công mạng.

Chương 2 đề án thực hiện khảo sát các kỹ thuật sinh dữ liệu tấn công mạng bao gồm các phương pháp sinh dữ liệu truyền thống và sinh dữ liệu sử dụng học máy, học sâu.

Chương 3 trình bày mô hình thử nghiệm, các kịch bản và kết quả thử nghiệm.

II. NỘI DUNG

CHƯƠNG 1 : TỔNG QUAN VỀ SINH DỮ LIỆU TẤN CÔNG MẠNG

1.1. Tổng quan về dữ liệu tấn công mạng

1.1.1. Khái quát về tấn công mạng

a, Khái niệm tấn công mạng

Theo Điều 2 Luật An ninh mạng 2018, *"Tấn công mạng là hành vi sử dụng không gian mạng, công nghệ thông tin hoặc phương tiện điện tử để phá hoại, gây gián đoạn hoạt động của mạng viễn thông, mạng Internet, mạng máy tính, hệ thống thông tin, hệ thống xử lý và điều khiển thông tin, cơ sở dữ liệu, phương tiện điện tử" [1]*. Nói cách khác, đây là hình thức lợi dụng điểm yếu, lỗ hổng trong cấu trúc, giao thức hoặc hành vi vận hành của hệ thống thông tin để đạt được mục tiêu không hợp pháp.

Khác với các hình thức tội phạm truyền thống, tấn công mạng có tính ẩn danh, phạm vi rộng và khả năng lây lan toàn cầu. Chỉ cần một thiết bị kết nối Internet, kẻ tấn công có thể thực hiện hành vi phạm tội ở bất kỳ đâu, nhắm đến bất kỳ hệ thống nào. Điều này khiến việc phòng ngừa và ứng phó với các cuộc tấn công mạng trở nên đặc biệt khó khăn.

b, Mục tiêu của các cuộc tấn công mạng

Tùy theo động cơ và bối cảnh, các cuộc tấn công mạng có thể hướng đến nhiều mục tiêu khác nhau. Mục tiêu phổ biến nhất là chiếm đoạt thông tin, đặc biệt là dữ liệu cá nhân, tài khoản ngân hàng hoặc thông tin kinh doanh quan trọng. Bên cạnh đó, một số cuộc tấn công được thiết kế để phá hoại hệ thống hoặc làm gián đoạn hoạt động dịch vụ, như trong các cuộc tấn công từ chối dịch vụ (DoS/DDoS).

Ngoài ra, nhiều tin tặc còn tìm cách chiếm quyền kiểm soát hệ thống, cài đặt phần mềm độc hại hoặc cửa hậu (backdoor) để duy trì quyền truy cập trái phép trong thời gian dài. Các hình thức tấn công lừa đảo (phishing) và mã độc tống tiền (ransomware) cũng ngày càng phổ biến, hướng đến mục tiêu trục lợi tài chính. Ở quy mô lớn hơn, tấn công mạng còn được sử dụng như một công cụ chiến lược trong chiến tranh thông tin, phục vụ mục tiêu chính trị hoặc quân sự giữa các quốc gia.

c, Phân loại tấn công mạng

Có thể phân chia các dạng tấn công mạng theo mức độ can thiệp và đích khai thác. Theo mức độ can thiệp, có thể chia làm hai nhóm chính:

- Tấn công thụ động (Passive attack): Người tấn công chỉ quan sát hoặc nghe lén dữ liệu lưu chuyển trong mạng mà không làm thay đổi nội dung, Ví dụ: phân tích gói tin (packet sniffing) hoặc theo dõi lưu lượng.

- Tấn công chủ động (Active attack): Can thiệp trực tiếp để thay đổi, chèn, xóa hoặc làm gián đoạn dữ liệu, như tấn công DoS, phát tán mã độc, hoặc tấn công giả mạo địa chỉ (spoofing).

Theo đích khai thác, tấn công mạng có thể được chia thành bốn nhóm phổ biến:

- Tấn công ở tầng mạng: Lợi dụng điểm yếu của giao thức truyền thông (ví dụ TCP/IP) để gây nghẽn hoặc chặn luồng dữ liệu.
- Tấn công ở tầng hệ thống: Khai thác lỗ hổng của hệ điều hành hoặc phần mềm, như lỗi tràn bộ đệm (buffer overflow).
- Tấn công ứng dụng: Nhắm vào các ứng dụng web, thường gặp là SQL Injection, Cross-Site Scripting (XSS).
- Tấn công dựa trên yếu tố con người: Dạng tấn công sử dụng kỹ thuật xã hội (social engineering), lừa người dùng cung cấp thông tin hoặc thực hiện hành động vô tình hỗ trợ tin tặc.

c, Tác động và xu hướng hiện tấn công mạng hiện nay.

Tác động của tấn công mạng không chỉ dừng lại ở thiệt hại tài chính mà còn ảnh hưởng nghiêm trọng đến uy tín, niềm tin và an toàn của hệ thống. Một vụ tấn công có thể khiến doanh nghiệp phải ngừng hoạt động, mất dữ liệu quan trọng hoặc bị rò rỉ thông tin khách hàng. Theo các báo cáo gần đây, chi phí khắc phục hậu quả do tội phạm mạng gây ra trên toàn cầu đang tăng nhanh và dự kiến đạt mức hàng nghìn tỷ đô la Mỹ mỗi năm trong giai đoạn tới.

Về xu hướng, các cuộc tấn công hiện nay ngày càng tự động hóa và thông minh hơn, nhờ việc ứng dụng trí tuệ nhân tạo (AI) và học máy (Machine Learning) để dò tìm lỗ hổng hoặc né tránh hệ thống phòng thủ. Bên cạnh đó, hình thức tấn công chuỗi cung ứng (Supply Chain Attack) đang gia tăng, khi tin tặc nhắm vào các nhà cung cấp phần mềm hoặc đối tác liên kết để xâm nhập gián tiếp vào hệ thống mục tiêu. Các thiết bị IoT và hệ thống công nghiệp (ICS/SCADA) cũng trở thành mục tiêu hấp dẫn do khả năng bảo mật hạn chế.

Đặc biệt, trong những năm gần đây, ransomware và phishing được xem là hai hình thức tấn công phổ biến nhất, gây thiệt hại lớn cho cả cá nhân và các tổ chức. Bên cạnh đó, chiến tranh mạng (Cyber Warfare) đang nổi lên như một xu hướng mới, khi các quốc gia sử dụng tấn công mạng như một công cụ chiến lược để gây bất ổn hoặc thu thập thông tin tình báo.

1.1.2. Dữ liệu tấn công mạng

a, Khái niệm dữ liệu tấn công mạng

Dữ liệu tấn công mạng là tập hợp các thông tin thu thập được từ hoạt động giao tiếp trên mạng hoặc trong hệ thống máy tính, phản ánh hành vi của người dùng, ứng

dụng và tiến trình trong quá trình vận hành. Tập dữ liệu này bao gồm cả dữ liệu bình thường và dữ liệu tấn công, được gắn nhãn tương ứng để phục vụ cho việc phân tích, huấn luyện và đánh giá mô hình nhận diện tấn công.

Về bản chất, dữ liệu tấn công mạng có thể bao gồm nhiều loại thông tin khác nhau, chẳng hạn:

- Thông tin mức gói tin: Các đặc trưng về địa chỉ IP, cổng, giao thức, thời gian, độ dài gói tin, …
- Thông tin mức phiên: Tổng hợp hoạt động trong một kết nối, như số lượng gói gửi/nhận, thời gian duy trì kết nối, tỷ lệ lỗi hoặc độ trễ.
- Thông tin mức ứng dụng: Nội dung, hành vi hoặc đặc trưng truy cập của các ứng dụng cụ thể (HTTP, DNS, FTP, …).

Tùy thuộc vào mục tiêu nghiên cứu, dữ liệu có thể được thu thập trực tiếp từ môi trường mạng thực, hệ thống mô phỏng, hoặc các kho dữ liệu công khai như KDD Cup 1999, NSL-KDD, UNSW-NB15, CICIDS2017, … Mỗi bộ dữ liệu mang những đặc trưng riêng về quy mô, độ chi tiết, cấu trúc và số lượng loại tấn công được mô phỏng.

b, Đặc điểm của dữ liệu tấn công mạng

Trong lĩnh vực an ninh mạng, hệ sinh thái dữ liệu được các tổ chức, viện nghiên cứu đóng góp còn hạn chế, tuy nhiên có thể thấy được các đặc điểm nổi bật của dữ liệu tấn công mạng như sau:

- Tính không cân bằng: Dữ liệu bình thường áp đảo so với dữ liệu tấn công, khiến mô hình học máy khó học ranh giới phân lớp.
- Tính đa dạng và phức tạp: Các loại tấn công khác nhau (DDoS, Port Scan, SQL Injection…) biểu hiện qua đặc trưng riêng, khó biểu diễn đầy đủ trong cùng một tập dữ liệu.
- Tính động và thay đổi theo thời gian: Mẫu tấn công liên tục thay đổi để tránh bị phát hiện, làm dữ liệu thu thập nhanh lỗi thời.
- Tính nhạy cảm và giới hạn truy cập: Dữ liệu thật thường bị bảo mật, hạn chế tiếp cận cho nghiên cứu.
- Tính dư thừa và nhiễu: Thông tin thừa hoặc dữ liệu nhiễu có thể xuất hiện, làm giảm độ chính xác mô hình.

c, Vai trò của việc thu thập và chuẩn hoá dữ liệu tấn công mạng

Việc thu thập và xây dựng dữ liệu tấn công mạng giữ vai trò nền tảng trong nghiên cứu an ninh mạng, bởi dữ liệu là yếu tố trực tiếp ảnh hưởng đến hiệu quả, độ chính xác và khả năng khái quát của mô hình phát hiện xâm nhập.

Thứ nhất, dữ liệu tấn công mạng là cơ sở huấn luyện mô hình phát hiện tấn công dự trên học máy. Các thuật toán học máy và học sâu cần lượng lớn dữ liệu có nhãn để học đặc trưng phân biệt hành vi bình thường và tấn công.

Thứ hai, việc thu thập và chuẩn hoá dữ liệu giúp đánh giá và so sánh hiệu năng giữa các phương pháp phát hiện tấn công. Dữ liệu chuẩn giúp so sánh khả năng phát hiện, tỷ lệ cảnh báo giả và bỏ sót, đồng thời thích ứng với nhiều loại tấn công.

Thứ ba, quá trình thu thập dữ liệu còn hỗ trợ phát hiện sớm các xu hướng và mẫu tấn công mới. Dữ liệu cập nhật liên tục giúp phát hiện sớm các hình thức tấn công mới, cải thiện chính sách phòng thủ.

Cuối cùng, dữ liệu thu thập được còn là nguồn tài nguyên quý giá để xây dựng các mô hình sinh dữ liệu (Data Generation Models), nhằm khắc phục vấn đề mất cân bằng và thiếu dữ liệu tấn công thực tế. Đây cũng chính là cơ sở để nghiên cứu các kỹ thuật sinh dữ liệu trong các chương tiếp theo của đề tài.

1.2. Các bộ dữ liệu tấn công mạng phổ biến

1.2.1. Bộ dữ liệu UNSW-NB15

Bộ dữ liệu UNSW-NB15 [20] được phát triển vào năm 2015 bởi Australian Centre for Cyber Security (ACCS), với mục tiêu khắc phục những hạn chế của các bộ dữ liệu cũ như KDD99 và NSL-KDD. Dữ liệu được tạo ra bằng cách kết hợp lưu lượng mạng thực tế và tấn công mô phỏng, nhằm phản ánh các hoạt động trên môi trường mạng hiện đại. Bộ dữ liệu này được thu thập thông qua công cụ IXIA PerfectStorm, một hệ thống tạo lưu lượng hỗn hợp có thể mô phỏng cả hành vi bình thường lẫn các kiểu tấn công khác nhau. Quá trình phân tích và trích xuất đặc trưng được thực hiện bằng công cụ Argus và Bro (nay là Zeek), tạo ra 49 đặc trưng (features) đại diện cho mỗi phiên kết nối (flow). Bảng 1.1 và 1.2 lần lượt cung cấp thông tin về các thành phần và mẫu lưu lượng của bộ dữ liệu UNSW-NB15.

Bảng 1.1: Các loại tấn công trong UNSW- NB15

Loại tấn công	Huấn luyện	Kiểm tra
Normal	*56,000*	*37,000*
Analysis	2,000	677
Backdoor	1,746	583
DoS	12,264	4,089
Exploits	33,393	11,132
Fuzzers	18,184	6,062
Generic	40,000	18,871
Worms	130	44

Reconnaissance	10,491	3,496
Shellcode	1,133	378

Bảng 1.2: Mẫu lưu lượng trong tập dữ liệu UNSW-NB15

id	dur	proto	service	state	…	ct_srv_dst	attack_cat	label
1	0.000011	udp	-	INT		2	Normal	0
2	0.000008	udp	-	INT		2	Normal	0
3	0.000005	udp	-	INT		3	Normal	0
4	0.000006	udp	-	INT		3	Normal	0
247	0.921987	ospf	-	INT		1	DoS	1
288	1.173262	tcp	-	FIN		1	DoS	1
337	1.438237	tcp	http	FIN		1	DoS	1
339	1.465404	tcp	http	FIN		1	DoS	1
360	28.21314	ospf	-	INT		1	DoS	1

1.2.2. Bộ dữ liệu CIC-UNSW-NB15 Augmented DataSet

Tập dữ liệu CIC-UNSW-NB15 Augmented Dataset [19] được phát hành bởi Canadian Institute for Cybersecurity (CIC) – Đại học New Brunswick (Canada), được xây dựng dựa trên bộ UNSW-NB15 gốc do Đại học New South Wales (UNSW), Úc phát triển năm 2015. Để tạo CIC-UNSW-NB15, công cụ CICFlowMeter được sử dụng để trích xuất bộ đặc trưng mới từ dữ liệu lưu lượng mạng được UNSW-NB15 thu thập. Bảng 1.3 cung cấp thông tin về các thành phần của bộ dữ liệu CIC-UNSW-NB15.

Bảng 1.3: Các loại tấn công trong CIC-UNSW-NB15 Augmented DataSet

Loại tấn công	**Tập dữ liệu gốc**	**CICFlowMeter**	**CIC-UNSW-NB15**
Benign	2218764	3450658	358332
Analysis	2677	385	385
Backdoor	2329	452	452
DoS	16353	4467	4467
Exploits	44525	30951	30951
Fuzzers	24246	29613	29613
Generic	215481	4632	4632
Reconnaissance	13987	16735	16735
Shellcode	1511	2102	2102
Worms	174	246	246

1.2.3. Bộ dữ liệu CIC-IDS-2017

Bộ dữ liệu CIC-IDS-2017 [21] được công bố bởi Canadian Institute for Cybersecurity (CIC) vào năm 2017, với mục tiêu tạo ra một bộ dữ liệu gần với thực tế nhất trong việc mô phỏng hoạt động mạng của người dùng. Quá trình thu thập được thực hiện trong vòng năm ngày, bao gồm cả lưu lượng bình thường và nhiều kịch bản tấn công khác nhau.

Mỗi bản ghi được trích xuất 80 đặc trưng, bao gồm cả thông tin về thời gian, giao thức, kích thước gói tin, tỷ lệ lỗi, số lượng gói gửi và nhận, v.v. Bộ dữ liệu này có hơn 2,8 triệu bản ghi, được chia thành nhiều tệp theo ngày để phục vụ cho quá trình phân tích linh hoạt. Bảng 1.4 và 1.5 lần lượt cung cấp thông tin về các thành phần và mẫu lưu lượng của bộ dữ liệu CIC-IDS-2017.

Bảng 1.4: Các loại tấn công trong CIC-IDS-2017

Loại tấn công	Huấn luyện	Kiểm tra
Benign	*219,068*	*153,091*
DDoS	4,184	2,928
DoS Hulk	2,304	6,137
DoS GoldenEye	1,030	720
SSH-Patator	590	413
DoS slowloris	580	406
DoS Slowhttptest	550	385
Bot	195	138
Infiltration	4	3
FTP-Patator		556

Bảng 1.5: Mẫu lưu lượng trong tập dữ liệu CIC-IDS-2017

Destination Port	Flow Duration	Total Fwd Packets	Total Length of Fwd Packets	Fwd Packet Length Max	…	Min Packet Length	Max Packet Length	Attack Type
80	5121659	3	12	6		0	6	Benign
53	128476	1	53	53		53	116	Benign
53	23701	1	53	53		53	69	Benign
54549	4	2	0	0		0	0	Benign
80	23912	3	26	20		0	5840	DDoS

80	80783228	9	62	20		0	8675	DDoS
80	7617540	4	24	6		6	6	DDoS
80	79130	3	26	20		0	5840	DDoS
80	81810201	8	56	20		0	5840	DDoS

1.2.4. Một số bộ khác (NSL-KDD, BoT-IoT, TON_IoT)

Bên cạnh hai bộ dữ liệu tiêu chuẩn nêu trên, còn có một số bộ dữ liệu khác cũng được sử dụng phổ biến trong nghiên cứu an ninh mạng, bao gồm:

- NSL-KDD [22]: Được phát triển nhằm cải tiến bộ KDD Cup 1999. NSL-KDD loại bỏ các bản ghi trùng lặp và giảm sự thiên lệch của dữ liệu, giúp việc huấn luyện mô hình trở nên hiệu quả hơn. Tuy nhiên, do được xây dựng từ môi trường mạng cũ, dữ liệu không còn phù hợp với các loại tấn công hiện đại.

- BoT-IoT [23]: Là bộ dữ liệu được xây dựng năm 2018 bởi Cyber Range Lab (UNSW Canberra), mô phỏng các tấn công trong môi trường Internet of Things (IoT). Dữ liệu bao gồm các loại tấn công như DDoS, DoS, Scan, và Theft, với đặc trưng tập trung vào lưu lượng IoT và thiết bị nhúng.

- TON_IoT [24]: Được phát triển sau BoT-IoT, bộ dữ liệu này mở rộng phạm vi thu thập bao gồm dữ liệu cảm biến, nhật ký hệ thống (log), và gói tin mạng trong hệ thống IoT. TON_IoT đặc biệt hữu ích cho nghiên cứu các giải pháp bảo mật toàn diện trong môi trường IoT và công nghiệp (ICS).

Các bộ dữ liệu này phản ánh xu hướng mở rộng nghiên cứu an ninh mạng sang các lĩnh vực mới như mạng IoT, công nghiệp 4.0 và điện toán biên, nơi mà tấn công mạng đang ngày càng phổ biến.

1.2.5. So sánh các bộ dữ liệu

Mỗi bộ dữ liệu được phát hành ở các thời điểm khác nhau có những ưu nhược điểm phù hợp với từng khía cạnh nghiên cứu trong lĩnh vực an ninh mạng nói chung và lĩnh vực phát hiện xâm nhập nói riêng. Bảng 1.6 dưới đây tổng quan một số đặc điểm của các bộ dữ liệu phổ biến hiện nay.

Bảng 1.6: Bảng so sánh các bộ dữ liệu UNSW-NB15, CIC-IDS-2017, NSL-KDD, BoT-IoT, TON-IoT.

Tiêu chí	**UNSW-NB15**	**CIC-IDS-2017**	**NSL-KDD**	**BoT-IoT**	**TON_IoT**
Số lượng đặc trưng	49	80	41	5	>40

Số lượng bản ghi	~2.5 triệu	~2.8 triệu	~125.000	~70 triệu	>20 triệu
Số loại tấn công	9	14	4	4	10+
Môi trường thu thập	Mô phỏng hỗn hợp	Thực nghiệm thực tế	Mô phỏng	IoT mô phỏng	IoT và công nghiệp
Tính cân bằng dữ liệu	Mất cân bằng	Mất cân bằng	Cải thiện	Mất cân bằng	Mất cân bằng

1.3. Mục tiêu và nhu cầu của việc sinh dữ liệu tấn công mạng

Sinh dữ liệu tấn công mạng là quá trình tạo ra các mẫu dữ liệu mới mô phỏng hoặc mở rộng từ dữ liệu mạng gốc, với mục tiêu phản ánh các hành vi tấn công và lưu lượng mạng trong nhiều kịch bản khác nhau. Các mẫu dữ liệu này có thể được sinh ra dựa trên mô hình thống kê, thuật toán học máy hoặc học sâu như mạng đối sinh GAN hay CTGAN.

Khác với việc thu thập dữ liệu thật trong môi trường thực tế vốn tốn kém, rủi ro và khó kiểm soát, sinh dữ liệu cho phép tái tạo hoặc mở rộng dữ liệu huấn luyện một cách an toàn và linh hoạt. Các mẫu tổng hợp có thể duy trì đặc trưng thống kê và hành vi tương đồng với dữ liệu thật, từ đó giúp hệ thống phát hiện xâm nhập học được đa dạng dạng hành vi tấn công hơn và đạt hiệu quả cao hơn trong thực tế.

1.3.1. Mục tiêu của sinh dữ liệu trong hệ thống IDS

Trong các hệ thống phát hiện xâm nhập dựa trên học máy (ML-IDS), chất lượng và sự cân bằng của dữ liệu huấn luyện là yếu tố then chốt ảnh hưởng trực tiếp đến hiệu suất phát hiện. Tuy nhiên, hầu hết các bộ dữ liệu tấn công mạng hiện nay, chẳng hạn như UNSW-NB15 hay CIC-IDS-2017, đều tồn tại tình trạng mất cân bằng dữ liệu nghiêm trọng giữa lớp “bình thường” và “tấn công”.

Để khắc phục vấn đề này, sinh dữ liệu tấn công mạng hướng tới các mục tiêu chính sau:

- Cân bằng dữ liệu: Bổ sung thêm mẫu cho các lớp tấn công hiếm, giúp mô hình IDS tránh thiên lệch về lớp bình thường.
- Mở rộng dữ liệu huấn luyện: Tạo ra biến thể mới của các mẫu tấn công, giúp mô hình học được đặc trưng tổng quát hơn.
- Tái tạo hành vi tấn công: Giúp mô phỏng kịch bản tấn công trong môi trường an toàn, phục vụ huấn luyện và kiểm thử hệ thống.

- Đảm bảo bảo mật và quyền riêng tư: Dữ liệu tổng hợp có thể được chia sẻ mà không tiết lộ thông tin nhạy cảm.

Nhờ vậy, sinh dữ liệu trở thành một bước quan trọng trong quy trình phát triển hệ thống IDS hiện đại, giúp cải thiện hiệu suất phát hiện, độ tổng quát hóa của mô hình, đồng thời giảm chi phí và rủi ro trong thu thập dữ liệu thật.

1.3.2. Nhu cầu sinh dữ liệu tấn công mạng trong các lĩnh vực

Nhu cầu sinh dữ liệu tổng hợp đang ngày càng mở rộng trong nhiều lĩnh vực, đặc biệt là an ninh mạng, Internet vạn vật (IoT) và công nghiệp thông minh (ICS). Cụ thể:

- Trong nghiên cứu học máy và học sâu: Các nhóm nghiên cứu thường gặp khó khăn khi thu thập đủ dữ liệu tấn công để huấn luyện mô hình. Việc sinh dữ liệu giúp tăng kích thước tập huấn luyện và cải thiện độ chính xác của các mô hình IDS.

- Trong lĩnh vực công nghiệp và doanh nghiệp: Do yếu tố bảo mật, dữ liệu tấn công thật hiếm khi được chia sẻ. Dữ liệu tổng hợp có thể thay thế cho dữ liệu thật trong các bài kiểm thử và huấn luyện, đảm bảo an toàn thông tin.

- Trong môi trường IoT và SCADA: Các hệ thống IoT thường thiếu cơ chế lưu trữ và ghi nhận đầy đủ dữ liệu tấn công. Do đó, sinh dữ liệu tấn công mạng giúp mô phỏng các hành vi tấn công đặc thù cho IoT, chẳng hạn như DDoS hoặc Botnet, hỗ trợ phát triển IDS chuyên biệt cho IoT.

- Trong đào tạo an ninh mạng: Việc sử dụng dữ liệu tổng hợp cho phép xây dựng môi trường huấn luyện mô phỏng thực tế, nơi học viên có thể thực hành phát hiện và ứng phó tấn công mà không ảnh hưởng tới hệ thống thật.

Như vậy, sinh dữ liệu tấn công mạng không chỉ là công cụ hỗ trợ nghiên cứu, mà còn là giải pháp thực tế trong sản xuất, giáo dục và phát triển công nghệ an ninh mạng hiện đại.

1.4. Thách thức trong sinh dữ liệu tấn công mạng

Mặc dù sinh dữ liệu tấn công mạng mang lại nhiều lợi ích trong việc nâng cao hiệu quả phát hiện xâm nhập, song quá trình này vẫn đối mặt với nhiều thách thức kỹ thuật và thực tiễn. Các thách thức này xuất phát từ chính bản chất phức tạp của dữ liệu mạng, sự giới hạn trong thu thập dữ liệu thật, cũng như yêu cầu đảm bảo tính chân thực và bảo mật của dữ liệu tổng hợp. Dưới đây là bốn nhóm vấn đề tiêu biểu.

1.4.1. Thiếu hụt dữ liệu thực tế chất lượng cao

Một trở ngại lớn trong nghiên cứu an ninh mạng là khan hiếm dữ liệu tấn công thực tế chất lượng cao và nhãn chính xác. Dữ liệu thật thường nhạy cảm, khó thu thập do chính sách bảo mật và quyền riêng tư. Nhiều tổ chức thiếu cơ sở hạ tầng để ghi nhận lưu lượng tấn công đa dạng hoặc gán nhãn thủ công. Do đó, các bộ dữ liệu công khai

như UNSW-NB15 hay CIC-IDS-2017, dù hữu ích, vẫn chưa phản ánh đầy đủ hành vi tấn công thực tế, làm giảm tính đại diện của dữ liệu huấn luyện và khó đảm bảo tính chân thực khi sinh dữ liệu tổng hợp.

1.4.2. Mất cân bằng dữ liệu

Mất cân bằng dữ liệu xảy ra khi mẫu "bình thường" áp đảo so với mẫu "tấn công", do các sự kiện tấn công mạng chiếm tỷ lệ rất nhỏ, khiến mô hình học máy thiên về lớp bình thường. Sinh dữ liệu được coi là giải pháp cân bằng các lớp, nhưng việc tạo thêm mẫu tấn công vừa xác thực vừa không lặp lại là thách thức lớn. Các kỹ thuật thống kê như SMOTE có thể gây overfitting hoặc tạo mẫu giả không phù hợp không gian đặc trưng gốc. Do đó, lựa chọn và hiệu chỉnh phương pháp sinh dữ liệu quyết định hiệu quả của hệ thống IDS.

1.4.3. Vấn đề riêng tư và bảo mật

Khi sinh dữ liệu từ các bộ dữ liệu thật, cần tránh rò rỉ thông tin nhạy cảm của người dùng hoặc hệ thống mạng. Các phương pháp học sâu, đặc biệt GAN và biến thể, có thể tái tạo dữ liệu gốc, gây nguy cơ vi phạm quyền riêng tư. Vấn đề nghiêm trọng hơn khi dữ liệu chứa IP, thông tin phiên hoặc payload, có thể tiết lộ cấu trúc hệ thống. Do đó, nghiên cứu sinh dữ liệu tấn công cần áp dụng Differential Privacy hoặc ẩn danh hóa để cân bằng giữa tính hữu ích và bảo vệ thông tin.

1.4.4. Đặc trưng phức tạp và không đồng nhất

Dữ liệu mạng đa chiều, không đồng nhất và biến động cao, gồm đặc trưng định lượng (byte, thời gian kết nối), định tính (protocol, service) và chuỗi (payload). Mô phỏng mối quan hệ phi tuyến và phụ thuộc giữa các thuộc tính này là thách thức lớn. Các mô hình như CTGAN hay VAE thường gặp vấn đề mất tính nhất quán giữa các trường hoặc không sinh chính xác thuộc tính phân loại. Ngoài ra, dữ liệu có tính thời gian (temporal dependency), khiến các mô hình tĩnh khó nắm bắt đầy đủ hành vi tấn công. Do đó, phát triển mô hình sinh dữ liệu hiểu ngữ cảnh tấn công và mối quan hệ đặc trưng là hướng nghiên cứu quan trọng trong tương lai.

1.5. Các nghiên cứu liên quan về sinh dữ liệu tấn công mạng

1.5.1. Các nghiên cứu sử dụng phương pháp truyền thống

Phương pháp truyền thống để giải quyết mất cân bằng trong dữ liệu tấn công mạng chủ yếu dựa trên các kỹ thuật over-sampling, under-sampling và các biến thể lai giữa chúng. Một trong những phương pháp over-sampling được sử dụng rộng rãi là phương pháp SMOTE (Synthetic Minority Over-sampling Technique). Phương pháp SMOTE không nhân bản thô các mẫu thiểu số mà tạo mẫu tổng hợp bằng cách nội suy một số mẫu thiểu số và các láng giềng gần kề thuộc cùng lớp [2].

Trong nghiên cứu kinh điển *"SMOTE: Synthetic Minority Over-sampling Technique"* của Nitesh V. Chawla và cộng sự (2002) [2], nhóm tác giả đề xuất SMOTE

nhằm cải thiện hiệu suất phân loại trong bài toán dữ liệu mất cân bằng. Phương pháp này tạo các mẫu tổng hợp mới từ các điểm thiểu số lân cận thay vì nhân bản dữ liệu cũ, giúp mở rộng vùng quyết định và giảm overfitting. Thực nghiệm trên nhiều tập dữ liệu cho thấy SMOTE kết hợp under-sampling vượt trội hơn so với các kỹ thuật truyền thống như under-sampling đơn thuần, Ripper hay Naive Bayes điều chỉnh. Trong 48 thí nghiệm, các mô hình dùng SMOTE đạt kết quả tốt nhất trong phần lớn trường hợp, khẳng định hiệu quả và tính tổng quát của phương pháp.

Ngoài ra, trong nghiên cứu *"ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning"* của Haibo He và cộng sự (2008) [8], nhóm tác giả đề xuất phương pháp ADASYN nhằm nâng cao khả năng học của mô hình trên dữ liệu mất cân bằng. Phương pháp này sinh thêm mẫu tổng hợp cho lớp thiểu số tùy theo mức độ "khó học" của từng điểm — các mẫu dễ bị nhầm lẫn sẽ được sinh nhiều hơn. Nhờ đó, ADASYN giảm thiên lệch và điều chỉnh biên quyết định theo hướng chú trọng các trường hợp khó. Kết quả thực nghiệm cho thấy ADASYN đạt hiệu quả vượt trội hơn so với SMOTE và nhiều kỹ thuật xử lý mất cân bằng khác trên nhiều tập dữ liệu.

Các phương pháp truyền thống có lợi thế là đơn giản, giải thích được và tiêu tốn ít tài nguyên tính toán. Tuy nhiên, chúng thường kém hiệu quả khi đặc trưng dữ liệu phức tạp (hỗn hợp categorical/continuous, phụ thuộc thời gian) như dữ liệu mạng, và có xu hướng tạo ra mẫu không phản ánh đúng phân phối thực tế nếu áp dụng thiếu cẩn trọng.

1.5.2. Các nghiên cứu sử dụng mô hình học máy, học sâu

Trong vài năm gần đây, các kỹ thuật học máy và học sâu đã được áp dụng rộng rãi để sinh dữ liệu tấn công mạng với mục tiêu mô phỏng phân phối phức tạp hơn và sinh được các mẫu "thực hơn" so với các phép nội suy đơn giản.

Trong nghiên cứu "*G-IDS: Generative Adversarial Networks Assisted Intrusion Detection System*" của Md Hasan Shahriar và các cộng sự (2020) [9], nhóm tác giả đề xuất hệ thống phát hiện xâm nhập có sự hỗ trợ của mạng đối kháng tạo sinh (GAN) dành cho môi trường mạng và IoT. Hệ thống này sử dụng mô hình GAN để tạo ra các mẫu dữ liệu tổng hợp nhằm khắc phục vấn đề dữ liệu thiếu hoặc mất cân bằng trong quá trình huấn luyện IDS. Các thử nghiệm với tập dữ liệu NSL-KDD cho thấy phương pháp này có hiệu suất phát hiện tấn công cao hơn và ổn định hơn so với IDS thông thường chỉ sử dụng dữ liệu gốc.

Trong bài báo "*Packet-Level Adversarial Network Traffic Crafting using Sequence Generative Adversarial Networks*" Qiumei Cheng và các cộng sự [10] đề xuất mô hình Attack-GAN dựa trên SeqGAN để sinh lưu lượng mạng ở cấp độ gói tin nhằm đánh lừa hệ thống phát hiện xâm nhập mạng (NIDS). Mỗi byte của gói tin được xử lý như một phần tử trong chuỗi, giúp mô hình học cách tạo ra lưu lượng hợp lệ giả mạo.

Kết quả cho thấy Attack-GAN có thể vượt qua nhiều hệ thống NIDS dạng hộp đen, mở ra hướng nghiên cứu mới trong việc ứng dụng GAN cho sinh dữ liệu tấn công mạng.

Trong nghiên cứu của Lei Xu và các cộng sự [11], tác giả đề xuất CTGAN, một biến thể của GAN thiết kế riêng cho dữ liệu dưới dạng bảng. CTGAN giới thiệu các kỹ thuật mới như mode-specific normalization để xử lý các cột liên tục đa mô-đun, conditional generator và training-by-sampling để khắc phục mất cân bằng và các đặc trưng hỗn hợp (rời rạc + liên tục) của dữ liệu bảng. Thử nghiệm trên bộ benchmark gồm nhiều tập dữ liệu giả lập và thực tế cho thấy CTGAN vượt trội hơn các mô hình GAN trước đó và nhiều phương pháp mạng Bayes trong việc tái tạo phân phối và hiệu năng học máy từ dữ liệu tổng hợp.

Các phương pháp học sâu (đặc biệt GANs và các biến thể dành cho dữ liệu bảng/chuỗi) đang trở thành hướng nghiên cứu chủ đạo cho sinh dữ liệu tấn công mạng do khả năng mô phỏng phân phối phức tạp. Tuy nhiên, việc đảm bảo tính chân thực, tránh rò rỉ dữ liệu gốc, ổn định khi huấn luyện và kiểm soát chất lượng mẫu sinh vẫn là những bài toán mở lớn.

1.6. Kết chương

Chương 1 đã trình bày tổng quan về bài toán sinh dữ liệu tấn công mạng, một hướng nghiên cứu quan trọng trong bối cảnh các hệ thống phát hiện xâm nhập (IDS) phụ thuộc ngày càng nhiều vào dữ liệu lớn để huấn luyện. Do việc thu thập dữ liệu thực tế gặp hạn chế về quyền riêng tư, chi phí và sự hiếm của các loại tấn công, sinh dữ liệu nhân tạo trở thành giải pháp hiệu quả giúp mở rộng tập huấn luyện, cân bằng dữ liệu và nâng cao khả năng tổng quát của mô hình.

Ngoài ra, chương cũng làm rõ các khái niệm nền tảng về dữ liệu tấn công mạng, các yêu cầu cho học máy và những thách thức đặc thù như mất cân bằng, thiếu dữ liệu chất lượng cao và vấn đề bảo mật. Cuối cùng, chương đã giới thiệu các hướng nghiên cứu tiêu biểu gồm phương pháp truyền thống (SMOTE) và phương pháp học sâu (GAN và các biến thể).

CHƯƠNG 2 : CÁC KỸ THUẬT SINH DỮ LIỆU TẤN CÔNG MẠNG

2.1. Khái quát về các phương pháp sinh dữ liệu tấn công mạng

Sinh dữ liệu tấn công mạng là quá trình tạo ra các mẫu dữ liệu giả lập phản ánh đặc tính của các hoạt động tấn công thực tế. Mục tiêu là mở rộng tập dữ liệu huấn luyện, cân bằng các lớp dữ liệu mất cân bằng, hoặc tạo ra các tình huống kiểm thử trong môi trường an toàn. Việc sinh dữ liệu trở nên đặc biệt quan trọng trong lĩnh vực an ninh mạng, nơi dữ liệu thật thường khan hiếm, khó thu thập, hoặc chứa thông tin nhạy cảm.

Các phương pháp sinh dữ liệu đã phát triển đa dạng, từ các kỹ thuật xử lý thống kê đơn giản đến các mô hình học sâu phức tạp. Mỗi hướng tiếp cận phản ánh một triết lý khác nhau về cách mô phỏng và tái tạo hành vi tấn công trong dữ liệu.

2.1.1. Phân loại các phương pháp sinh dữ liệu

Trong nghiên cứu sinh dữ liệu tấn công mạng, việc phân loại phương pháp theo nguyên lý tạo mẫu và nguồn tri thức giúp hệ thống hóa tiếp cận, so sánh ưu nhược điểm và lựa chọn hướng phù hợp. Nhìn chung, các kỹ thuật được chia thành ba nhóm chính: (i) dựa trên lấy mẫu và mô hình thống kê, (ii) dựa trên mô phỏng và quy tắc chuyên ngành, và (iii) dựa trên học máy/học sâu.

Nhóm thống kê – lấy mẫu (như SMOTE, ADASYN) điều chỉnh tần suất lớp bằng cách tạo hoặc loại bớt mẫu để giảm mất cân bằng, thường dùng như bước tiền xử lý do đơn giản và ít tốn tài nguyên.

Nhóm mô phỏng – rule-based sử dụng kiến thức chuyên môn về giao thức và hành vi tấn công để sinh dữ liệu có ngữ nghĩa rõ ràng, phù hợp cho đánh giá kịch bản xác định nhưng thiếu tính ngẫu nhiên và đòi hỏi chuyên môn cao.

Nhóm học máy/học sâu (như GAN, VAE) học phân phối dữ liệu để sinh mẫu mới tương đồng thống kê, có khả năng tạo dữ liệu đa dạng và phi tuyến nhưng yêu cầu dữ liệu lớn, tài nguyên và kỹ thuật huấn luyện phức tạp.

Ngoài ra, phương pháp lai (hybrid) đang được quan tâm, kết hợp giữa mô phỏng, rule-based và các mô hình sinh học máy hoặc sampling để tận dụng ưu điểm của từng hướng. Việc lựa chọn phương pháp phụ thuộc vào đặc trưng dữ liệu, mục tiêu nghiên cứu và nguồn lực thực tế.

2.1.2. Ưu nhược điểm, mức độ phổ biến

Các nhóm phương pháp trên có những ưu thế và hạn chế riêng, phụ thuộc vào mục tiêu nghiên cứu, dạng dữ liệu, và nguồn lực sẵn có. Bảng 2.1 cung cấp so sánh khái quát so sánh ưu nhược điểm và độ phổ biến các nhóm phương pháp sinh dữ liệu.

Bảng 2.1: Bảng so sánh ưu nhược điểm và độ phổ biến các nhóm phương pháp sinh dữ liệu

Nhóm phương pháp	Ưu điểm chính	Hạn chế	Mức độ phổ biến trong an ninh mạng
Truyền thống	- Đơn giản, dễ cài đặt, chi phí tính toán thấp. - Hiệu quả trong bài toán mất cân bằng dữ liệu.	- Dễ gây *overfitting* hoặc sinh mẫu không tự nhiên. - Không mô phỏng được các quan hệ phức tạp giữa các đặc trưng.	Rất phổ biến trong giai đoạn tiền xử lý dữ liệu, đặc biệt với các bộ dữ liệu như UNSW-NB15, CICIDS2017.
Mô phỏng/ Dựa trên luật	- Kiểm soát tốt hành vi tấn công, dễ giải thích. - Có thể tạo kịch bản hiếm hoặc mới.	- Phụ thuộc nhiều vào kiến thức chuyên môn. - Tốn công thiết kế, khó mở rộng, dễ thiếu tính ngẫu nhiên.	Phổ biến trong môi trường thử nghiệm hoặc khi xây dựng tập dữ liệu tấn công có cấu trúc (DDoS, fuzzing).
Học máy/ Học sâu	- Mô phỏng tốt các phân phối phức tạp. - Tạo dữ liệu đa dạng, sát thực tế. - Có khả năng thích ứng cho nhiều dạng dữ liệu khác nhau.	- Yêu cầu dữ liệu huấn luyện lớn và tài nguyên tính toán cao. - Khó kiểm soát, có thể gặp vấn đề về ổn định khi huấn luyện.	Xu hướng mới, đang được nghiên cứu mạnh trong lĩnh vực sinh dữ liệu mạng.

2.2. Các phương pháp sinh dữ liệu truyền thống

2.2.1. Phương pháp sinh dữ liệu dựa trên lấy mẫu (sampling)

Các phương pháp sinh dữ liệu dựa trên lấy mẫu là hướng tiếp cận truyền thống để xử lý mất cân bằng lớp trong tập huấn luyện. Trong dữ liệu tấn công mạng, mẫu "bình thường" thường chiếm ưu thế, còn mẫu tấn công rất ít, khiến mô hình dễ thiên lệch và nhận dạng kém với các tấn công hiếm.

Nhìn chung, kỹ thuật lấy mẫu gồm hai nhóm chính: oversampling – tăng số lượng mẫu lớp thiểu số, và undersampling – giảm số lượng mẫu lớp đa số.

a, Oversampling

Oversampling là một kỹ thuật tăng cường dữ liệu khi xử lý các tập dữ liệu mất cân bằng, kỹ thuật này cân bằng sự phân bố bằng cách tăng số lượng mẫu trong lớp thiểu số. Nguyên lý hoạt động chính của kỹ thuật này là việc sao chép các điểm hiện có hoặc tạo ra các điểm mới bằng cách tạo dữ liệu tổng hợp.

Một trong những kỹ thuật oversampling tiêu biểu và được sử dụng rộng rãi nhất là SMOTE (Synthetic Minority Over-sampling Technique). Phương pháp SMOTE được đề xuất bởi Chawla (2002) [2], với ý tưởng tạo ra các mẫu tổng hợp (synthetic samples) cho lớp thiểu số thay vì sao chép trực tiếp các mẫu hiện có. Nguyên lý hoạt động của SMOTE dựa trên phép nội suy tuyến tính giữa một mẫu thiểu số và các láng giềng gần của nó.

Cụ thể, với mỗi mẫu thiểu số x_i, thuật toán chọn ngẫu nhiên một trong k láng giềng gần nhất cùng lớp (với k thường bằng 5), sau đó tạo ra mẫu mới x_{new} theo công thức:

$$x_{new} = x_i + \delta \text{ x } (x_{nm} - x_i),\ \delta \in [0,1] \qquad (2.1)$$

Trong đó, x_{nm} là láng giềng được chọn, còn δ là hệ số ngẫu nhiên. Cách tiếp cận này giúp mở rộng không gian biểu diễn của lớp thiểu số, tạo ra các điểm dữ liệu mới có ý nghĩa thống kê so với việc nhân bản mẫu cũ.

Công thức này đảm bảo rằng mẫu mới nằm trên đoạn nối giữa hai điểm thiểu số, qua đó mở rộng biên phân lớp của lớp thiểu số, giúp mô hình học máy nhận diện tốt hơn các vùng dữ liệu hiếm.

Thuật toán SMOTE có thể được mô tả qua các bước chính sau:

Bước 1: Xác định lớp tối thiểu trong tập dữ liệu.

Bước 2: Với mỗi mẫu x_i thuộc lớp thiểu số:

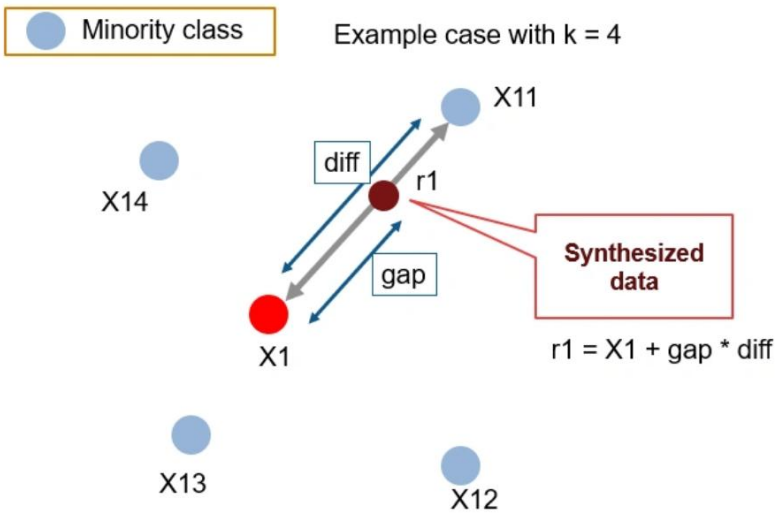
- Tìm k láng giềng gần nhất của x_i trong không gian đặc trưng.
- Chọn ngẫu nhiên một số láng giềng để sinh mẫu tổng hợp.
- Tạo mẫu mới x_{new} theo công thức nội suy tuyến tính (2.1) ở trên.

Bước 3: Hợp tập dữ liệu gốc và các mẫu tổng hợp thành tập huấn luyện mới.

SMOTE mang lại nhiều lợi thế so với các phương pháp oversampling truyền thống:

- Giảm overfitting: Tạo mẫu mới thay vì sao chép giúp mô hình học tốt hơn và tăng khả năng khái quát.
- Cân bằng dữ liệu: Bổ sung số lượng mẫu lớp thiểu số, giúp phân bố dữ liệu đồng đều hơn.

- Cải thiện hiệu suất mô hình: Dữ liệu cân bằng giúp mô hình giảm thiên lệch và tăng các chỉ số Accuracy, Precision, Recall và F1-score.



Hình 2.1: Ví dụ thuật toán SMOTE với k = 4

Bên cạnh đó, SMOTE cũng vẫn còn tồn tại một số hạn chế như:

- Dễ tạo dữ liệu nhiễu: Nếu lớp thiểu số có nhiễu hoặc ngoại lai, SMOTE có thể sinh thêm các mẫu không tốt và làm giảm chất lượng mô hình.
- Mẫu tổng hợp thiếu tính thực tế: Dữ liệu được tạo theo phép nội suy tuyến tính nên đôi khi không phản ánh đúng sự phức tạp thật của lớp thiểu số.
- Tốn thời gian và tài nguyên: Việc tạo mẫu mới, đặc biệt trên các tập dữ liệu lớn, có thể làm tăng chi phí tính toán.

Sau công trình của Chawla [2], nhiều biến thể của SMOTE đã được đề xuất nhằm cải thiện hiệu năng và khả năng thích ứng với các dạng dữ liệu khác nhau.

- Borderline-SMOTE [25]: chỉ sinh mẫu gần biên phân lớp, giúp tăng khả năng phân biệt giữa hai lớp.
- SMOTE-NC [25]: mở rộng SMOTE cho dữ liệu hỗn hợp gồm cả thuộc tính số và thuộc tính danh mục.
- ADASYN [25]: điều chỉnh tỷ lệ sinh mẫu dựa trên độ khó học của từng điểm (điểm gần biên sinh nhiều hơn).
- SVM-SMOTE [33]: sử dụng siêu phẳng của SVM để xác định vùng cần sinh mẫu.
- KMeans-SMOTE [33]: kết hợp phân cụm để sinh dữ liệu có cấu trúc tốt hơn, tránh sinh trong vùng nhiễu.

Trong lĩnh vực phát hiện tấn công mạng, SMOTE được ứng dụng rộng rãi nhằm cân bằng các loại tấn công hiếm gặp trong các bộ dữ liệu kinh điển như KDDCup'99, NSL-KDD, và UNSW-NB15. Các nghiên cứu của Zhang cùng cộng sự [12] và nhóm tác giả từ China Information Technology Security Evaluation Center [13] chỉ ra rằng khi áp dụng SMOTE vào các tập dữ liệu này, độ chính xác trung bình của các mô hình phân loại tăng từ 3–15% so với khi không xử lý mất cân bằng, đặc biệt hiệu quả đối với các loại tấn công có tần suất thấp như User-to-Root (U2R) hoặc Fuzzers. Ngoài ra, khi kết hợp SMOTE với các kỹ thuật giảm nhiễu (như Tomek Links hoặc Edited Nearest Neighbors), mô hình có thể đạt được sự cân bằng tốt hơn giữa độ chính xác và độ nhạy. Nhờ những ưu điểm về tính đơn giản, hiệu quả và khả năng mở rộng, SMOTE vẫn là phương pháp nền tảng được sử dụng phổ biến trong nhiều nghiên cứu và hệ thống thực tế, đồng thời là cơ sở cho các kỹ thuật cải tiến hiện đại như Borderline-SMOTE, ADASYN hay KMeans-SMOTE.

b, Undersampling

Bên cạnh các kỹ thuật oversampling như SMOTE hay ADASYN, undersampling là một hướng tiếp cận truyền thống khác nhằm xử lý vấn đề mất cân bằng lớp trong dữ liệu huấn luyện. Thay vì tăng cường số lượng mẫu của lớp thiểu số, phương pháp này giảm bớt số lượng mẫu của lớp đa số để đạt được tỉ lệ cân bằng hơn giữa các lớp. Cách tiếp cận đơn giản và phổ biến nhất trong nhóm này là Random Undersampling (RUS), được xem như một kỹ thuật "lọc dữ liệu" giúp giảm thiên lệch học tập về phía lớp chiếm ưu thế.

Nguyên lý của Random Undersampling [26] khá trực quan: từ lớp đa số trong tập dữ liệu, thuật toán chọn ngẫu nhiên một tập con có kích thước nhỏ hơn (thường bằng hoặc gần bằng số lượng mẫu của lớp thiểu số), sau đó loại bỏ phần dữ liệu dư thừa còn lại. Phương pháp này giúp mô hình học máy không bị chi phối bởi sự xuất hiện quá nhiều của một lớp, đồng thời rút ngắn thời gian huấn luyện nhờ giảm kích thước dữ liệu. Một biến thể của phương pháp này là Cluster Centroids Undersampling, trong đó các mẫu bị loại bỏ dựa trên kết quả phân cụm, nhằm giữ lại các điểm "đại diện" cho toàn bộ lớp đa số.

Ưu điểm nổi bật của Random Undersampling nằm ở tính đơn giản, hiệu quả tính toán cao và khả năng áp dụng nhanh cho nhiều loại dữ liệu khác nhau. Việc giảm kích thước tập dữ liệu giúp rút ngắn đáng kể thời gian huấn luyện mô hình và giảm chi phí lưu trữ, đặc biệt hữu ích trong các hệ thống xử lý dữ liệu mạng có quy mô lớn. Ngoài ra, khi được kết hợp với các kỹ thuật lựa chọn mẫu thông minh (ví dụ: NearMiss, Tomek Links), RUS còn có thể loại bỏ nhiễu và vùng chồng lấn không cần thiết, giúp cải thiện chất lượng biên phân lớp.

2.2.2. Phương pháp sinh dữ liệu dựa trên mô phỏng

Phương pháp mô phỏng tái tạo hành vi, môi trường hoặc quy trình sinh dữ liệu thực bằng mô hình toán học hoặc phần mềm, dựa trên quy tắc xác suất, đặc tính vật lý hoặc kịch bản hệ thống thay vì học trực tiếp từ dữ liệu. Mô phỏng có thể thực hiện ở mức sự kiện rời rạc hoặc mức hệ thống, ví dụ mô phỏng lưu lượng mạng, hành vi người dùng hay tiến trình tấn công, và sinh dữ liệu phản ánh các tình huống giả định để huấn luyện, kiểm thử IDS.

Ưu điểm là khả năng kiểm soát cao: có thể tạo kịch bản hiếm/nguy hiểm khó thu thập, điều chỉnh tham số linh hoạt để đánh giá phản ứng mô hình trong tình huống bất thường — điều rất hữu ích trong an toàn mạng khi dữ liệu thật bị hạn chế.

Hạn chế là phụ thuộc vào độ chính xác của mô hình giả định; nếu mô phỏng không phản ánh đúng hành vi thực tế thì dữ liệu sinh ra thiếu đại diện, dẫn đến sai lệch mô hình học máy. Ngoài ra, chi phí thiết kế và hiệu chỉnh mô phỏng cao và đòi hỏi chuyên môn sâu về hệ thống và công cụ.

Trong lĩnh vực an ninh mạng, nhiều nghiên cứu đã áp dụng mô phỏng để sinh dữ liệu tấn công mạng có kiểm soát. Ví dụ, Mirkovic [16] đã phát triển công cụ DDoS Simulator để tạo ra các mẫu tấn công từ chối dịch vụ có thể tùy chỉnh cường độ và mô hình lưu lượng. Ngoài ra, trong bài báo “A Taxonomy of DDoS Attack and DDoS Defense Mechanisms” [15] đã mô phỏng hành vi người dùng và phần mềm độc hại để sinh ra tập dữ liệu kiểm thử IDS trong môi trường an toàn. Những nghiên cứu này cho thấy mô phỏng là công cụ mạnh mẽ để mở rộng dữ liệu phục vụ huấn luyện và đánh giá các hệ thống phòng thủ mạng.

2.2.3. Phương pháp sinh dữ liệu dựa trên luật

Phương pháp sinh dữ liệu dựa trên luật là hướng tiếp cận dựa vào tập hợp các quy tắc hoặc mẫu hành vi (rules/patterns) được xác định trước để tạo ra dữ liệu mới. Khác với mô phỏng (dựa trên mô hình hệ thống) hay học máy (dựa vào thống kê từ dữ liệu thật), phương pháp này dựa trên tri thức chuyên gia và quy tắc xác định sẵn, cho phép kiểm soát chặt chẽ các đặc tính của dữ liệu sinh ra.

Cốt lõi của phương pháp là tập luật sinh dữ liệu (generation rules), trong đó mỗi luật xác định mối quan hệ giữa các thuộc tính và giá trị khả dĩ của chúng. Khi thực thi, hệ thống áp dụng các luật này để tạo ra các mẫu dữ liệu phù hợp với những ràng buộc logic hoặc ngữ cảnh cụ thể. Ví dụ, trong phát hiện xâm nhập mạng, có thể sinh ra gói tin “giả lập” tấn công dựa theo mẫu:

Nếu protocol = TCP và dst_port = 80 thì có thể sinh hành vi tấn công kiểu HTTP Flood.

Ưu điểm của phương pháp dựa trên luật là minh bạch và kiểm soát cao: người thiết kế có thể xác định rõ dữ liệu sinh ra và điều chỉnh luật theo nhu cầu nghiên cứu.

Phương pháp này ít phụ thuộc dữ liệu đầu vào, đồng thời tạo dữ liệu tuân thủ các quy định hoặc tiêu chuẩn nghiệp vụ.

Nhược điểm là phụ thuộc chuyên gia miền, việc xây dựng và hiệu chỉnh bộ luật yêu cầu kiến thức sâu. Khi số lượng luật tăng, độ phức tạp và chi phí bảo trì cũng tăng, dễ gây mâu thuẫn hoặc chồng chéo. Dữ liệu sinh ra từ luật cố định thường thiếu ngẫu nhiên và đa dạng, khiến mô hình học máy dễ overfitting vào các mẫu quy tắc có sẵn.

Trong các hệ thống IDS, phương pháp sinh dữ liệu dựa trên luật thường được sử dụng để tái tạo mẫu tấn công chuẩn hoặc kiểm thử hệ thống phát hiện xâm nhập. Ví dụ, QI Liu và các cộng sự [18] đã áp dụng phương pháp rule-based để sinh dữ liệu tấn công mạng trên cơ sở các quy tắc Snort nhằm huấn luyện và đánh giá mô hình phát hiện. Ngoài ra, một số công cụ như IDSGAN Generator hoặc DARPA Traffic Generator cũng tích hợp các tập luật để sinh ra luồng lưu lượng mạng giả định, phục vụ huấn luyện mô hình học sâu.

2.2.4. So sánh các kỹ thuật

Trong các phương pháp sinh dữ liệu truyền thống, ba nhóm kỹ thuật tiêu biểu đã được trình bày gồm: (i) phương pháp lấy mẫu (sampling), (ii) phương pháp mô phỏng (simulation), và (iii) phương pháp dựa trên luật (rule-based).

Mỗi nhóm lại có nhiều kỹ thuật cụ thể, trong đó SMOTE, ADASYN và Random Undersampling là ba đại diện phổ biến nhất trong nhóm sampling, được ứng dụng rộng rãi trong xử lý mất cân bằng dữ liệu mạng.

Xét trong nội bộ nhóm lấy mẫu, SMOTE (Synthetic Minority Over-sampling Technique) tạo dữ liệu mới bằng cách nội suy tuyến tính giữa các mẫu thiểu số gần nhau, giúp duy trì cấu trúc dữ liệu gốc. ADASYN mở rộng SMOTE bằng cách tăng cường sinh dữ liệu ở các vùng khó phân loại, qua đó cải thiện khả năng học của mô hình. Trong khi đó, Random Undersampling giảm kích thước lớp đa số để cân bằng dữ liệu, nhưng làm mất thông tin và thường gây suy giảm hiệu quả huấn luyện.

So với nhóm mô phỏng và nhóm dựa trên luật, các kỹ thuật lấy mẫu có ưu thế về độ đơn giản, tính tự động và khả năng tích hợp vào quy trình học máy. Tuy nhiên, chúng chỉ tạo ra dữ liệu "mới" về mặt hình thức, không phản ánh được hành vi tấn công thực tế như các kỹ thuật mô phỏng hay rule-based. Ngược lại, các kỹ thuật mô phỏng và dựa trên luật cho phép sinh dữ liệu có ngữ nghĩa tấn công cụ thể hơn, nhưng đòi hỏi nhiều tri thức miền và công sức thiết lập.

Nhìn chung, các phương pháp truyền thống mang lại hiệu quả nhất định trong việc cân bằng và mở rộng dữ liệu cho các hệ thống phát hiện xâm nhập. Tuy nhiên, chúng bị giới hạn bởi tính tuyến tính và quy tắc cố định, khó mô hình hóa được các hành vi tấn công ngày càng tinh vi. Do đó, các kỹ thuật học sâu tạo sinh đã được đề xuất nhằm khắc phục các hạn chế này.

2.3. Sinh dữ liệu bằng phương pháp học máy, học sâu

2.3.1. Tổng quan về học máy, học sâu.

a, Khái niệm, ứng dụng, một số thuật toán học máy và học sâu.

Học máy (Machine Learning – ML) là lĩnh vực con của trí tuệ nhân tạo, tập trung vào việc xây dựng mô hình có khả năng học từ dữ liệu để thực hiện dự đoán hoặc phân loại mà không cần lập trình thủ công từng quy tắc. Học sâu (Deep Learning – DL) là một nhánh đặc biệt của học máy, sử dụng các mạng nơ-ron nhiều tầng (deep neural networks) để học các biểu diễn phức tạp từ dữ liệu, đặc biệt hữu ích với dữ liệu lớn và phi cấu trúc.

Các mô hình học máy và học sâu hiện được ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau. Chúng bao gồm các tác vụ như phân loại và dự đoán dữ liệu, nhận diện hình ảnh, video, âm thanh, xử lý ngôn ngữ tự nhiên, cũng như sinh dữ liệu tổng hợp. Trong bối cảnh tấn công mạng, các mô hình này đặc biệt hữu ích để tạo ra dữ liệu giả có đặc tính thống kê và cấu trúc gần giống dữ liệu thật, hỗ trợ huấn luyện các hệ thống phát hiện tấn công hoặc đánh giá bảo mật.

Một số thuật toán học máy phổ biến gồm hồi quy tuyến tính, cây quyết định, rừng ngẫu nhiên (Random Forest) và máy vector hỗ trợ (SVM). Trong khi đó, học sâu sử dụng các kiến trúc mạng phức tạp hơn như mạng nơ-ron tích chập (CNN), mạng hồi tiếp (RNN), Autoencoder, Generative Adversarial Networks (GAN) cùng các biến thể nâng cao như VAE, CTGAN, CopulaGAN. Những thuật toán này cho phép mô hình học các phân phối dữ liệu phức tạp, từ dữ liệu rời rạc, liên tục đến dữ liệu tuần tự.

b, Lợi thế của học máy, học sâu so với truyền thống.

So với các phương pháp thống kê truyền thống, học máy và học sâu mang lại nhiều lợi thế nổi bật:

- Tự động học các mối quan hệ phức tạp: Nhận biết các tương tác phi tuyến và phụ thuộc đa chiều trong dữ liệu.
- Xử lý dữ liệu lớn và đa dạng: Thích hợp với dữ liệu nhiều chiều, dữ liệu rời rạc, liên tục hoặc tuần tự, giảm thiểu yêu cầu tiền xử lý.
- Cải thiện độ chính xác trong dự đoán và sinh dữ liệu: Sinh dữ liệu tổng hợp chất lượng cao, đặc biệt khi dữ liệu gốc bị mất cân bằng hoặc thiếu hụt.
- Tính linh hoạt và khả năng mở rộng: Áp dụng cho nhiều loại dữ liệu và bài toán khác nhau, từ hình ảnh, âm thanh, văn bản đến dữ liệu tấn công mạng.
- Bảo toàn cấu trúc và phân phối dữ liệu: Dữ liệu giả gần giống dữ liệu thật, giữ được đặc trưng thống kê và cấu trúc dữ liệu gốc.

Nhìn chung, học máy và học sâu không chỉ nâng cao khả năng phân tích dữ liệu mà còn tạo ra các giải pháp sinh dữ liệu tổng hợp hiệu quả, giúp mở rộng và cải thiện

chất lượng tập dữ liệu, đặc biệt trong các ứng dụng mạng và an ninh mạng, nơi dữ liệu thật thường hạn chế hoặc mất cân bằng.

2.3.2. Mạng đối kháng tạo sinh (Generative Adversarial Networks – GANs)

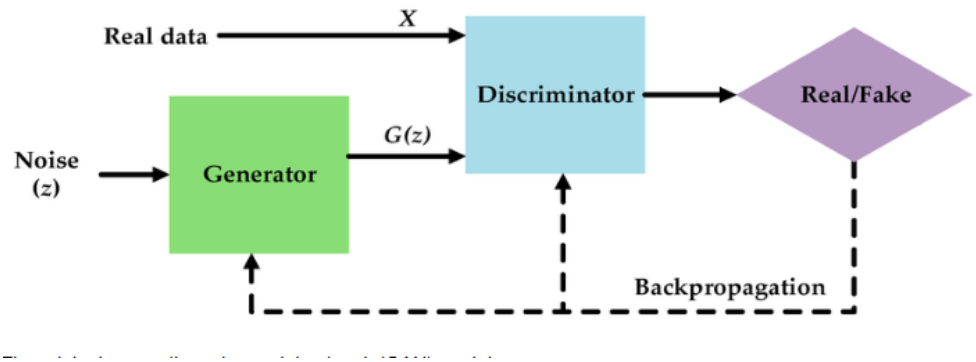
a, Mạng Generative Adversarial Networks

Mạng đối kháng tạo sinh (Generative Adversarial Network – GAN) được giới thiệu bởi Ian Goodfellow (2014) [3], được xem là bước ngoặt lớn trong lĩnh vực học sâu, mở ra hướng tiếp cận mới cho các bài toán sinh dữ liệu.

Cấu trúc của GAN bao gồm hai mạng nơ-ron chính hoạt động theo cơ chế đối kháng:

- Bộ sinh (Generator – G): Nhận đầu vào là một vector ngẫu nhiên $z \sim P_z(z)$, thường được lấy từ phân phối Gaussian hoặc Uniform. Mục tiêu của G là ánh xạ từ không gian nhiễu z sang không gian dữ liệu thật x, sao cho dữ liệu sinh G(z) có đặc trưng thống kê tương tự dữ liệu thật.
- Bộ phân biệt (Discrimination – D): Là một mạng nơ-ron đóng vai trò bộ phân loại nhị phân, nhận đầu vào là một mẫu dữ liệu (có thể thật hoặc giả), và đầu ra là xác suất mẫu đó là thật. Nếu D (x) =1 nghĩa là mô hình tin rằng mẫu x này là thật; ngược lại D(G(x)) = 0 thể hiện mẫu giả bị phát hiện.

Hai mạng này được huấn luyện đồng thời có mục tiêu trái ngược nhau, G cố gắng sinh dữ liệu sao cho D không thể phân biệt thật giả, còn D cố gắng nhận diện chính xác mẫu thật và mẫu sinh. Khi trạng thái cân bằng Nash, tức là D không thể phân biệt được mẫu thật hay giả (xác suất gần 0.5), khi đó G được xem là đã học thành công phân phối dữ liệu thật.



Hình 2.2: Mô hình mạng GAN

Hàm mất mát của GAN [28] là kết hợp giữa hàm mất mát của hai thành phần là bộ sinh và bộ phân biệt.

$$min_G min_D V(D,G) = E_{x\sim P_{data}(x)}[logD(x)] + E_{z\sim P_z(z)}[\log(1 - D(G(z)))] \quad (2.2)$$

Trong đó:

$D(x)$: Là xác suất mà bộ phân biệt – D cho rằng mẫu x là thật.

$G(z)$: Là mẫu dữ liệu được tạo ra bởi mô hình sinh – G từ nhiễu đầu vào z.

$p_{data}(x)[log\ D\ (x)]$: Là phân phối dữ liệu thật.

$p_z(z)[log(\ 1 - D(G(z)))]$: Là phân phối của nhiễu đầu vào.

Trong thực tế, việc tối ưu trực tiếp hàm mất mát trên có thể dẫn đến gradient vanish (mất gradient) khi D quá mạnh. Vì vậy Goodfellow đề xuất thay đổi hàm mục tiêu của G để tối qua hóa log(D(G(z))) thay vì tối tiểu hóa log (1- D(G(z))). [3]

b, Mạng WGAN, WGAN-GP.

Mặc dù mạng đối kháng tạo sinh (GAN) đã thể hiện hiệu quả vượt trội trong nhiều bài toán tạo dữ liệu, nhưng mô hình gốc vẫn tồn tại những hạn chế đáng kể trong quá trình huấn luyện. Cụ thể, quá trình tối ưu trong GAN truyền thống thường gặp hiện tượng mất cân bằng giữa hai mạng, khiến bộ sinh (Generator – G) không nhận được tín hiệu gradient ổn định từ bộ phân biệt (Discriminator – D). Hậu quả là mô hình dễ rơi vào tình trạng mode collapse, tức là chỉ sinh ra một số ít dạng mẫu, đồng thời hàm mất mát không phản ánh đúng chất lượng dữ liệu sinh. Để giải quyết các vấn đề này, Arjovsky (2017) [16] đã đề xuất Wasserstein GAN (WGAN) – một biến thể quan trọng của GAN, sử dụng khái niệm khoảng cách Wasserstein (Earth Mover's Distance – EMD) để đo độ khác biệt giữa hai phân phối thay vì sử dụng Jensen–Shannon Divergence (JSD) như trong GAN gốc.

Khoảng cách Wasserstien

Khoảng cách Wasserstein được định nghĩa là chi phí tối thiểu cần thiết để "vận chuyển" khối lượng xác suất từ phân phối P_g sang P_r. Công thức tổng quát được biểu diễn như sau:

$$W(\mathbf{p}_r, \mathbf{p}_g) = \inf_{\gamma \in \prod(\mathbf{p}_r, \mathbf{p}_g)} \mathbf{E}_{(x,y)\sim\gamma}[\| x - y \|] \qquad (2.3)$$

Khoảng cách này thể hiện trực quan "độ gần" giữa hai phân phối và có gradient liên tục, giúp mô hình dễ hội tụ hơn trong quá trình huấn luyện.

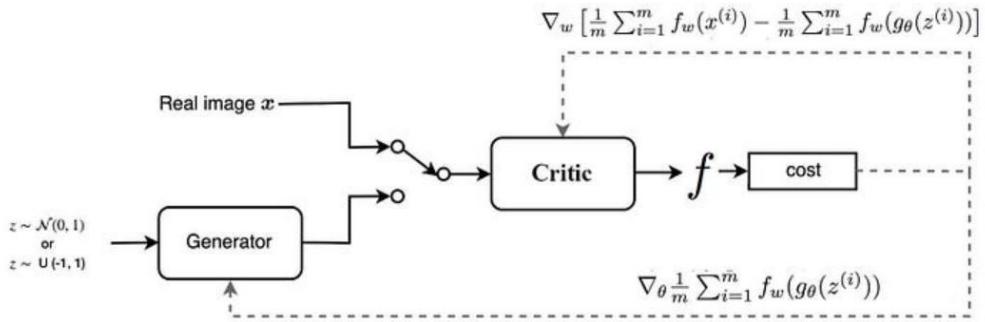
Dựa trên định lý Kantorovich–Rubinstein, khoảng cách Wasserstein [27] có thể được viết lại dưới dạng tối ưu hóa:

$$W\ (P_r, P_g) = \frac{sup}{||f||_L \leq 1} E_{x\sim P_r}[f(x)] - E_{x\sim P_g}[f(x)] \qquad (2.4)$$

Trong đó, hàm f được ràng buộc bởi điều kiện 1-Lipschitz, nghĩa là đạo hàm của nó không vượt quá 1.

Nhờ đặc tính này, khoảng cách Wasserstein mang lại gradient ổn định và có ý nghĩa thực tế hơn so với JSD, giúp cải thiện đáng kể khả năng huấn luyện của các mô hình tạo sinh.

Kiến trúc mạng WGAN



Hình 2.3: Kiến trúc WGAN [27]

Về kiến trúc của WGAN, cơ bản bao gồm 2 mạng chính là Generator và Critic:

- *Generator*: có đầu vào là vector ngẫu nhiên z lấy từ phân phối chuẩn hoặc đồng đều. Mạng này có thể được xây dựng từ các lớp Fully Connected hoặc Transposed Convolution (đối với dữ liệu hình ảnh), thường sử dụng hàm kích hoạt ReLU và tanh ở lớp cuối để đưa dữ liệu về miền [-1, 1].
- *Critic*: nhận đầu vào là mẫu dữ liệu thật hoặc giả và xuất ra một giá trị thực biểu thị "điểm số Wassertein", thể hiện mức độ khác biệt giữa hai phân phối. Critic không sử dụng hàm kích hoạt sigmoid ở đầu ra như Discriminator trong GAN truyền thống, và thường loại bỏ các kỹ thuật như Batch Normalization để không vi phạm điều kiện Lipschitz liên tục. Thay vào đó, WGAN sử dụng weight clipping, tức là giới hạn trọng số của Critic trong khoảng $[-c, c]$ nhằm duy trì tính Lipschitz cần thiết cho việc tính khoảng cách Wasserstein.

Quá trình huấn luyện WGAN

Trong quá trình huấn luyện, Critic thường được cập nhật nhiều lần hơn Generator (ví dụ 5 bước cập nhật Critic cho mỗi lần cập nhật Generator). Mục tiêu của Critic là tối đa hóa sự khác biệt giữa giá trị trung bình mà nó gán cho dữ liệu thật và dữ liệu giả, trong khi Generator cố gắng tối thiểu hóa giá trị đó bằng cách sinh ra mẫu dữ liệu khiến Critic khó phân biệt. Thuật toán huấn luyện như sau:

- Huấn luyện Critic (k lần):

  + Lấy mẫu minibatch từ dữ liệu thật $x \sim P_r$

  + Sinh minibatch từ Generator $x^{\sim} = G(z)$

  + Cập nhật Critic bằng cách tối đa hóa:

  $$L_D = E_{x \sim P_r}[D(x)] - E_{z \sim P_z}[D(z)] \qquad (2.5)$$

  + Clip trọng số của Critic: $w \leftarrow clip\ (w, -c, c)$

- Huấn luyện Generator (1 lần):

+ Cập nhật G để tối thiểu hóa:

$$L_G = -E_{z\sim P_z}[D(G(z))] \qquad (2.6)$$

Cơ chế Weight Clipping trong WGAN [29]

Trong WGAN gốc, Critic cần là hàm Lipschitz với hằng số 1 để khoảng cách Wasserstein được tính chính xác. Để đảm bảo điều này, WGAN sử dụng weight clipping, tức giới hạn giá trị trọng số của Critic trong một khoảng cố định [−c,c] sau mỗi bước cập nhật. Nhờ cơ chế này, Critic không học các hàm quá dốc, giúp quá trình huấn luyện ổn định hơn.

Tuy nhiên, weight clipping cũng có hạn chế. Nếu khoảng cắt quá nhỏ, Critic bị hạn chế khả năng biểu diễn, còn nếu quá lớn, điều kiện Lipschitz có thể bị vi phạm. Cơ chế này đôi khi dẫn đến gradient vanishing hoặc exploding, ảnh hưởng đến hiệu quả huấn luyện. Dù đơn giản và dễ triển khai, weight clipping chưa tối ưu hoàn toàn, từ đó thúc đẩy sự ra đời của WGAN-GP với cơ chế phạt gradient, cải thiện độ ổn định và chất lượng dữ liệu sinh.

Những cải tiến chính của WGAN

WGAN đưa ra một số cải tiến quan trọng nhằm khắc phục hạn chế của GAN truyền thống:

- Ứng dụng khoảng cách Wasserstein để đo mức độ khác biệt giữa dữ liệu thật và dữ liệu sinh, giúp mô hình Generator học hiệu quả và ổn định hơn.
- Hạn chế hiện tượng mất gradient, nhờ đó quá trình huấn luyện trở nên mượt mà và ít gặp lỗi hội tụ.
- Loại bỏ nhu cầu dùng hàm sigmoid ở Discriminator; thay vào đó áp dụng cơ chế ràng buộc như weight clipping hoặc gradient penalty để kiểm soát việc cập nhật trọng số.

Nhờ các cải tiến này, WGAN có khả năng tạo ra dữ liệu tổng hợp có chất lượng và độ chân thực cao hơn, đặc biệt phù hợp trong các bài toán sinh ảnh.

WGAN-GP (Wasserstein GAN with Gradient Penalty)

Trong WGAN gốc, điều kiện Lipschitz được duy trì bằng phương pháp weight clipping, tức là giới hạn toàn bộ trọng số của Critic trong một khoảng cố định [-c, c]. Cách làm này đơn giản nhưng có nhiều hạn chế: nếu giá trị ccc quá nhỏ, Critic trở nên quá "phẳng", làm mất khả năng học; ngược lại, nếu ccc quá lớn, tính Lipschitz không được đảm bảo, dẫn đến gradient không ổn định. Ngoài ra, weight clipping còn khiến quá trình tối ưu trở nên khó khăn và có thể làm Critic hội tụ kém.

WGAN-GP, được giới thiệu bởi Gulrajani [17], thay thế cơ chế clipping bằng Gradient Penalty – một cách mềm dẻo và hiệu quả hơn để duy trì tính Lipschitz. Thay vì ép trọng số về một khoảng, WGAN-GP thêm vào hàm mất mát của Critic một thành

phần phạt (penalty term), buộc chuẩn của gradient của Critic theo đầu vào phải gần bằng 1. Cụ thể, hàm mất mát của Critic trong WGAN-GP được viết là:

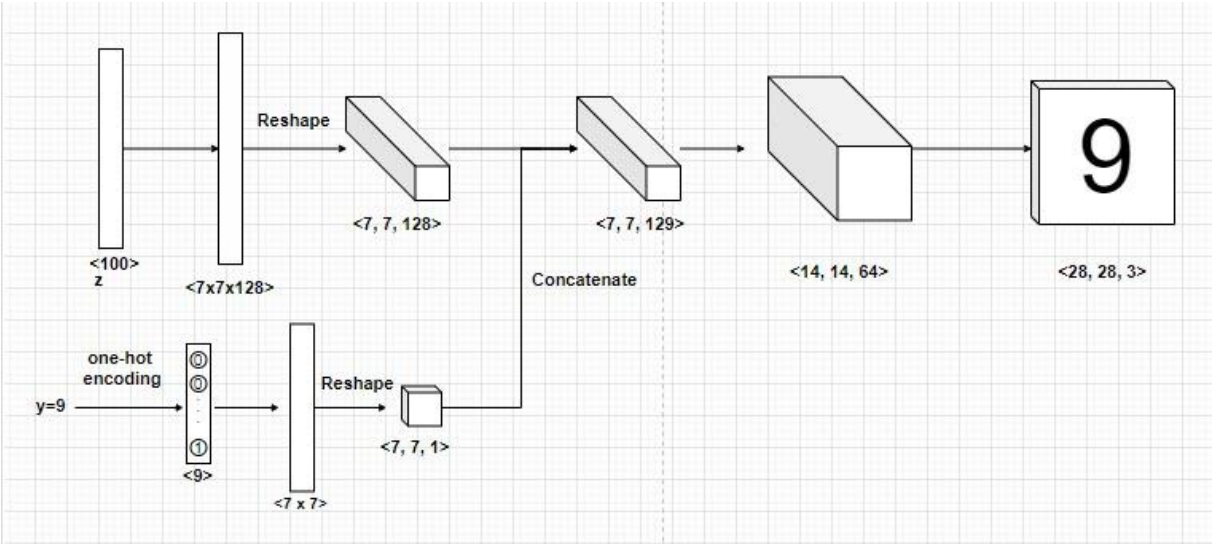
$$L_D = E_{x^{\sim} \sim P_g}[D(x^{\sim})] - E_{x \sim P_{r^{\sim}}}[D(x)] + \lambda E_{x^{\sim} \sim P_{x^{\sim}}}[(||\nabla x^{\sim} D(x^{\sim})||_2 - 1)^2] \quad (2.7)$$

Trong đó $x^{\sim}$ là điểm được lấy ngẫu nhiên trên đoạn nối giữa mẫu thật và mẫu giả, còn λ là hệ số điều chỉnh độ mạnh của ràng buộc (thường chọn λ=10). Nhờ đó, Critic được khuyến khích duy trì gradient có độ lớn xấp xỉ 1, đảm bảo điều kiện 1-Lipschitz một cách tự nhiên mà không làm giảm năng lực biểu diễn của mạng.

c, Mạng Conditional GAN (cGAN)

Conditional GAN (cGAN) là một biến thể quan trọng của GAN, được đề xuất bởi Mirza và Osindero (2014) trong bài báo "Conditional Generative Adversarial Nets" [21]. Mục tiêu của cGAN là điều khiển quá trình sinh dữ liệu thông qua một biến điều kiện y chẳng hạn như nhãn lớp, loại tấn công, hoặc thuộc tính dữ liệu. Điều này giúp mô hình sinh ra các mẫu có đặc điểm cụ thể thay vì ngẫu nhiên như trong GAN gốc.

Kiến trúc cGAN [30]



Hình 2.4: Mô hình generator của cGAN

Generator có điều kiện: Generator trong cGAN được thiết kế để sinh dữ liệu phụ thuộc vào biến điều kiện y.

- Đầu vào: Nhiễu ngẫu nhiên z và biến điều kiện y. Hai thành phần này có thể được ghép nối hoặc nhân theo từng phần tử.
- Mạng ẩn: Bao gồm các lớp Fully Connected hoặc Deconvolution (nếu dữ liệu là ảnh), kết hợp với các kỹ thuật như BatchNorm hoặc Conditional BatchNorm.
- Đầu ra: Mẫu dữ liệu giả $x^{\sim} = G(z, y)$.

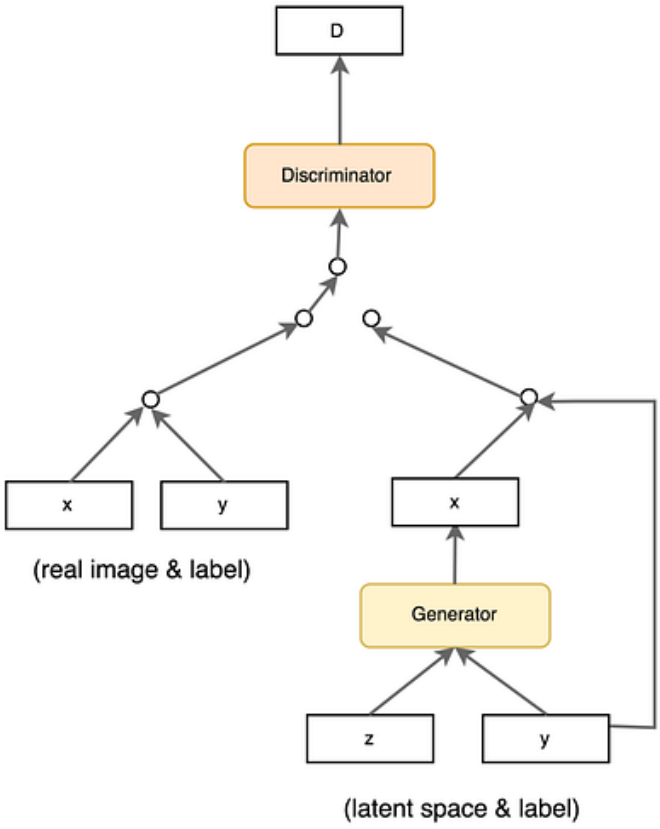
Discrimminator có điều kiện: Discriminator được huấn luyện để phân biệt dữ liệu thật có điều kiện (x, y) và dữ liệu giả có điều kiện (G (z, y), y).

- Đầu vào: Cặp (x, y).
- Kết hợp điều kiện: Có thể thực hiện theo nhiều cách:

  + Concatenation: Nối trực tiếp vector y và x hoặc với đặc trưng trích xuất từ x.

  + Projection Discriminator: Dùng tích vô hướng giữa embedding của y và vector đặc trưng cuối cùng của x.
- Đầu ra: Một giá trị thực biểu thị khả năng "thật" của mẫu x dưới điều kiện y.

Sơ đồ quá trình kết hợp giữa generator và discriminatior được thể hiện tổng quan dưới hình sau:



Hình 2.5: Sơ đồ kết hợp giữa generator và discriminator của cGAN [30]

Hàm mất mát (Loss function)

Trong Conditional GAN, cả Generator (G) và Discriminator (D) đều được cung cấp thêm thông tin điều kiện *y* (nhãn lớp hoặc đặc trưng phụ trợ). Điều này giúp mô hình sinh dữ liệu phù hợp với từng điều kiện cụ thể.

Hàm mất mát của CGAN được mở rộng từ GAN truyền thống bằng cách đưa y vào cả hai mô hình:

- Hàm mất mát của Discriminator:

$$L_D = -E_{x,y\sim p_{data}}[\log D(x,y)] - E_{z\sim P_z,y\sim P_{data}}[\log(1-D(G(z,y),y))] \quad (2.8)$$

- Hàm mất mát của Generator:

$$L_G = -E_{z\sim P_z,y\sim P_{data}}[\log D(G(z,y),y)] \quad (2.9)$$

Trong các biến thể hiện đại, cGAN còn có thể sử dụng Wasserstein loss, hinge loss, hoặc gradient penalty (như trong WGAN-GP) để ổn định quá trình huấn luyện.

So với GAN truyền thống, Conditional GAN (CGAN) được thiết kế để khắc phục hạn chế trong việc kiểm soát dữ liệu sinh, nhờ đó mang lại khả năng tạo mẫu có định hướng hơn. Những khác biệt nổi bật của CGAN gồm:

- Sử dụng thêm thông tin điều kiện (y): CGAN đưa nhãn lớp hoặc thuộc tính bổ sung vào cả Generator và Discriminator, trong khi GAN gốc chỉ dựa trên nhiễu đầu vào.
- Sinh dữ liệu theo lớp hoặc thuộc tính cụ thể: Việc bổ sung điều kiện cho phép mô hình tạo ra dữ liệu thuộc đúng một lớp nhất định, thay vì sinh ngẫu nhiên và khó kiểm soát như GAN truyền thống.
- Hàm mất mát có tích hợp điều kiện: Cả hai hàm mất mát của G và D đều kết hợp thông tin điều kiện y, giúp mô hình học phân phối có điều kiện p(x|y), từ đó cải thiện độ chính xác của dữ liệu sinh.
- Tăng khả năng kiểm soát chất lượng dữ liệu sinh: Nhờ cơ chế điều kiện, CGAN giảm khả năng sinh sai lớp và tạo ra dữ liệu phù hợp hơn với mục tiêu cụ thể của bài toán.

d, Mạng Conditional Tabular GAN (CTGAN)

CTGAN (Conditional Tabular GAN) là một biến thể đặc biệt của mô hình GAN được thiết kế để sinh dữ liệu dạng bảng (tabular data), tức là dữ liệu có cả thuộc tính rời rạc và thuộc tính liên tục. CTGAN được giới thiệu bởi Lei Xu trong bài báo "Modeling Tabular Data using Conditional GAN" [22] và hiện là một trong những mô hình sinh dữ liệu tổng hợp khá tốt cho bài toán dữ liệu cấu trúc.

Kiến trúc mạng CTGAN

CTGAN gồm hai thành phần chính: Generator và Discriminator, nhưng khác GAN thông thường, nó sử dụng conditional sampling để học đúng phân phối của từng cột rời rạc.

- Generator tạo dữ liệu tổng hợp phản ánh mối quan hệ giữa các thuộc tính, đầu vào là vector z ~ N(0,1) kèm điều kiện c. Mạng là Fully Connected với

Batch Normalization và LeakyReLU; đầu ra gồm các thuộc tính rời rạc qua Gumbel-Softmax và thuộc tính liên tục qua Mode-Specific Normalization.

- Discriminator phân biệt dữ liệu thật và giả, nhận mẫu kèm vector điều kiện, là MLP nhiều tầng với LeakyReLU và Dropout, xuất ra độ "thật" của mẫu. CTGAN dùng Wasserstein loss với Gradient Penalty (WGAN-GP) để ổn định huấn luyện và tránh mode collapse.

Cơ chế huấn luyện

- Condition Sampling: Mỗi lần huấn luyện, CTGAN chọn ngẫu nhiên một cột rời rạc và một giá trị cụ thể (như Protocol = TCP) để sinh dữ liệu có điều kiện. Cách này giúp mô hình học tốt hơn phân phối của các giá trị hiếm và giảm mất cân bằng dữ liệu.
- Mode – Specific Normalization: Với thuộc tính liên tục, CTGAN dùng *Gaussian Mixture Model (GMM)* để tách các mode trong phân phối, rồi chuẩn hóa từng mode riêng, giúp nắm bắt cấu trúc dữ liệu phức tạp chính xác hơn.
- Sử dụng Gumbel – Softmax: Hỗ trợ sinh biến rời rạc có thể lan truyền ngược, giúp Generator học phân phối xác suất của các giá trị rời rạc mượt mà hơn.

Nhờ các cơ chế trên cùng WGAN-GP, CTGAN xử lý tốt dữ liệu hỗn hợp, giữ quan hệ giữa các thuộc tính và đảm bảo huấn luyện ổn định, dữ liệu sinh ra sát thực tế.Tuy nhiên, CTGAN tốn thời gian huấn luyện hơn các phương pháp như SMOTE, cần tinh chỉnh nhiều siêu tham số, và hiệu quả phụ thuộc vào chất lượng vector điều kiện cũng như khả năng tách các mode của dữ liệu gốc.

d, Mạng CopulaGAN

CopulaGAN nổi bật ở việc tách riêng học phân phối từng biến và học mối quan hệ phụ thuộc đa biến thông qua copula. Khác với GAN truyền thống, dữ liệu được chuẩn hóa về không gian copula trước khi huấn luyện, giúp mô hình học được cấu trúc phụ thuộc mà không bị chi phối bởi phân phối biên.

Generator sinh dữ liệu trong không gian copula, Discriminator phân biệt giữa mẫu thật và giả, sau đó các mẫu được biến đổi ngược về miền dữ liệu gốc, giữ nguyên phân phối biên và mối quan hệ giữa các thuộc tính. Thuật toán còn bao gồm bước đánh giá và điều chỉnh chất lượng dữ liệu tổng hợp, đảm bảo mẫu sinh ra phản ánh đúng cấu trúc dữ liệu thật. So với CTGAN, CopulaGAN ưu tiên bảo toàn phụ thuộc đa biến, trong khi CTGAN mạnh hơn trong xử lý dữ liệu rời rạc kết hợp liên tục.

Thuật toán CopulaGAN

Thuật toán CopulaGAN [34] được triển khai qua bốn bước cơ bản: chuẩn hóa dữ liệu bằng copula, huấn luyện mạng GAN, chuyển dữ liệu trở về không gian gốc, và đánh giá điều chỉnh chất lượng dữ liệu sinh.

- Biến đổi dữ liệu bằng copula: Chuẩn hóa các cột dữ liệu về không gian đồng đều, tách rời phân phối biên và phụ thuộc đa biến.
- Huấn luyện GAN: Generator sinh dữ liệu giả trong không gian copula, Discriminator phân biệt giữa dữ liệu thật và giả; hai mạng cạnh tranh để cải thiện chất lượng mẫu.
- Biến đổi ngược về dữ liệu gốc: Sử dụng phép inverse copula để chuyển dữ liệu tổng hợp về miền gốc, giữ nguyên phân phối biên và các mối quan hệ giữa thuộc tính.
- Đánh giá và điều chỉnh: So sánh dữ liệu tổng hợp với dữ liệu thật về phân phối và phụ thuộc giữa các biến; nếu cần, mô hình được tinh chỉnh để nâng cao chất lượng mẫu sinh.

2.3.3. Bộ mã hóa tự động biến thiên (Variational Autoencoders -VAEs)

Bộ mã hóa tự động biến thiên (VAEs) [31] là một mô hình sinh dữ liệu dựa trên học biểu diễn tiềm ẩn (latent representation), cho phép tạo ra dữ liệu mới có phân phối gần giống dữ liệu gốc. VAE gồm hai thành phần chính:

- Encoder: Chuyển đổi dữ liệu đầu vào x thành một phân phối xác suất trong không gian latent z, thay vì ánh xạ cố định. Thông thường, phân phối này được giả định là Gaussian.
- Decoder: Sinh lại dữ liệu từ các mẫu z được rút ra từ phân phối latent.

Khác với Autoencoder truyền thống, VAE học phân phối tiềm ẩn thay vì một điểm duy nhất, nhờ đó có thể sinh dữ liệu ngẫu nhiên nhưng vẫn hợp lý. Hàm mất mát của VAE bao gồm:

- Lỗi tái tạo (reconstruction loss): Đo mức độ giống nhau giữa dữ liệu gốc và dữ liệu sinh ra.
- KL divergence: Đo sự khác biệt giữa phân phối latent học được và phân phối chuẩn, giúp latent space có cấu trúc liên tục và dễ điều hướng.

Nhờ những đặc điểm này, VAE có thể sinh dữ liệu ảnh, văn bản hoặc dữ liệu bảng, đồng thời duy trì các đặc trưng thống kê quan trọng của dữ liệu gốc, phù hợp với các bài toán sinh dữ liệu tổng hợp.

2.3.4. Mô hình ngôn ngữ lớn (Large Language Models -LLMs)

Mô hình ngôn ngữ lớn (LLMs) [32] là các mạng nơ-ron sâu được huấn luyện trên khối lượng văn bản cực lớn để học phân phối ngôn ngữ tự nhiên. Điển hình là các mô hình Transformer, sử dụng cơ chế attention để học các mối quan hệ dài hạn giữa các token.

Các đặc điểm chính của LLMs bao gồm:

- Khả năng sinh dữ liệu: Dự đoán và tạo token tiếp theo dựa trên ngữ cảnh, từ đó sinh văn bản mới hoặc log hệ thống.
- Hiểu ngữ nghĩa: Mô hình học biểu diễn ngữ nghĩa của từ, câu và đoạn văn, giúp tạo dữ liệu hợp lý và đa dạng.
- Ứng dụng linh hoạt: Trong sinh dữ liệu tấn công mạng, LLMs có thể được dùng để sinh log, câu lệnh, traffic pattern dạng văn bản, hoặc các tập dữ liệu mô phỏng hành vi mạng, hỗ trợ tăng kích thước và đa dạng tập dữ liệu huấn luyện cho các mô hình phát hiện tấn công.

Nhờ khả năng học mối quan hệ phức tạp trong dữ liệu tuần tự, LLMs trở thành công cụ mạnh trong việc sinh dữ liệu dạng chuỗi hoặc văn bản có cấu trúc, đặc biệt khi dữ liệu thật bị hạn chế hoặc nhạy cảm.

2.4. Kết chương

Chương này đã trình bày tổng quan về các phương pháp sinh dữ liệu, tập trung vào ba nhóm chính: phương pháp thống kê truyền thống, mô hình học máy và học sâu, và các mô hình nâng cao cho dữ liệu bảng và tuần tự. Các kỹ thuật thống kê như SMOTE giúp xử lý mất cân bằng dữ liệu nhưng gặp hạn chế về tính đa dạng và khả năng sinh dữ liệu phức tạp. Trong khi đó, các mô hình học sâu như GAN, WGAN, CTGAN, CopulaGAN, VAEs và LLMs cho phép sinh dữ liệu tổng hợp với cấu trúc và phân phối gần với dữ liệu thật, đồng thời xử lý tốt dữ liệu rời rạc, dữ liệu liên tục, hoặc dữ liệu tuần tự.

Chương này cũng nêu rõ cấu trúc, cơ chế hoạt động, thuật toán và ưu nhược điểm của từng phương pháp, từ đó làm cơ sở lý luận vững chắc cho các thí nghiệm sinh dữ liệu tấn công mạng trong chương tiếp theo.

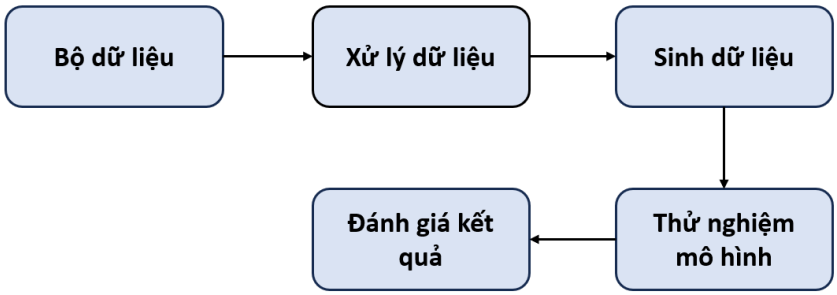
CHƯƠNG 3 : THỬ NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

3.1. Mô hình thử nghiệm

3.1.1. Giới thiệu mô hình thử nghiệm

Để nghiên cứu và đánh giá hiệu quả của các phương pháp sinh dữ liệu tấn công mạng, đề tài xây dựng một quy trình thực hiện gồm năm giai đoạn chính: (1) Chuẩn bị dữ liệu đầu vào, (2) Xử lý dữ liệu, (3) Sinh dữ liệu tấn công, (4) Thử nghiệm mô hình phân loại, và (5) Đánh giá kết quả.

Quy trình tổng quát được minh họa trong hình dưới đây.



Hình 3.1: Sơ đồ mô hình thực hiện đánh giá phương pháp sinh dữ liệu

Input: Bộ dữ liệu tấn công mạng.

Output: Kết quả thử nghiệm của mô hình phân loại.

Quy trình thực hiện:

Bước 1: Xử lý dữ liệu: Làm sạch, chuẩn hóa dữ liệu hoặc hoặc xử lý các đặc trưng dữ liệu gây nhiễu (nếu có).

Bước 2: Sinh dữ liệu: Áp dụng các kỹ thuật sinh dữ liệu nhằm mở rộng và cân bằng tập dữ liệu.

Bước 3: Thử nghiệm mô hình: Sử dụng tập dữ liệu để thực hiện huấn luyện và thử nghiệm trên các mô hình phân loại. Trong bước này, thực hiện huấn luyện và kiểm thử 2 mô hình phát hiện sử dụng dữ liệu thực và dữ liệu sinh của cùng 1 dạng tấn công để so sánh, đánh giá.

Bước 4: Đánh giá kết quả.

3.1.2. Thiết lập môi trường thực nghiệm

a, Môi trường thực nghiệm

Đề án được thực hiện trong môi trường tính toán sử dụng Kaggle Docker Image và tận dụng sức mạnh của GPU (CUDA) thông qua thư viện PyTorch để huấn luyện mô

hình học sâu. Các thư viện cốt lõi Numpy, Pandas, Sklearn, Torch được sử dụng để xây dựng mô hình trong đề án này.

Đề án được thực hiện trên hệ thống máy tính với cấu hình cụ thể như sau:

- Bộ nhớ: RAM 16GB.
- Ổ cứng: SSD 512GB
- Hệ điều hành: Windows 10 Pro

b, Cài đặt tham số

Đối với quá trình sinh dữ liệu tấn công mạng, các tham số được điều chỉnh để đảm bảo chất lượng dữ liệu sinh. Một số thông số quan trọng như: epochs = 200; learning_rate = 0.0001.

Trong mô hình phân loại dữ liệu, đề án sử dụng thuật toán học máy Random Forest (RF) được cấu hình các tham số cụ thể như sau:

```
rf_params = dict(
        n_estimators=400,
        random_state=RANDOM_STATE,
        n_jobs=-1,
        class_weight="balanced_subsample"
)
```

Trong đó:

- n-estimators: Số lượng cây quyết định (decision trees) trong rừng ngẫu nhiên.
- ramdom-state: Seed để kiểm soát tính ngẫu nhiên của mô hình, đảm bảo kết quả lặp lại.
- n-jobs: Sử dụng tất cả các nhân xử lý có sẵn để tăng tốc độ huấn luyện.
- class-weight: Điều chỉnh trọng số của các lớp dựa trên tần suất của chúng trong mẫu bootstrap (subsample), nhằm giảm thiểu tác động của sự mất cân bằng lớp (class imbalance) tiềm ẩn.

3.1.3. Các độ đo đánh giá

Để đánh giá hiệu quả của các mô hình phát hiện hoặc sinh dữ liệu tấn công mạng, người ta thường sử dụng các độ đo phổ biến được tính toán dựa trên các tham số của ma trận nhầm lẫn (confusion matrix) gồm: TP (True Positive – dương tính thật), TN (True Negative – âm tính thật), FP (False Positive – dương tính giả) và FN (False Negative – âm tính giả). Các độ đo cụ thể như sau:

Accuracy (Độ chính xác tổng thể)

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3.1)$$

Ý nghĩa: Đo tỷ lệ dự đoán đúng trên tổng số mẫu, phản ánh hiệu quả tổng thể của mô hình.

Precision (Độ chính xác dương tính)

$$Precision = \frac{TP}{TP+FP} \quad (3.2)$$

Ý nghĩa: Xác định tỷ lệ các dự đoán dương tính là chính xác, tức mức độ "đáng tin cậy" của các cảnh báo dương tính.

Recall (Độ nhạy, Tỷ lệ phát hiện dương tính)

$$Recall = \frac{TP}{TP+FN} \quad (3.3)$$

Ý nghĩa: Đo khả năng phát hiện đúng tất cả các mẫu dương tính thực sự, tức mức độ "không bỏ sót" các tấn công.

F1-score (Điểm hài hòa giữa Precision và Recall)

$$F1 = 2.\frac{Precision.Recall}{Precision+Recall} \quad (3.4)$$

Ý nghĩa: Kết hợp cả độ chính xác và độ nhạy vào một chỉ số duy nhất, phản ánh cân bằng giữa cảnh báo sai và bỏ sót.

FPR (False Positive Rate – Tỷ lệ dương tính giả)

$$FPR = \frac{FP}{FP+TN} \quad (3.5)$$

Ý nghĩa: Tỷ lệ mẫu âm tính bị dự đoán nhầm thành dương tính, phản ánh mức độ cảnh báo sai.

FNR (False Negative Rate – Tỷ lệ âm tính giả)

$$FNR = \frac{FN}{FN+TP} \quad (3.6)$$

Ý nghĩa: Tỷ lệ mẫu dương tính thực sự bị bỏ sót, phản ánh mức độ rủi ro do bỏ sót các tấn công.

Nhìn chung, việc kết hợp các độ đo giúp đánh giá toàn diện hiệu suất của mô hình, từ khả năng dự đoán chính xác, khả năng phát hiện các sự kiện quan trọng, đến rủi ro cảnh báo sai, từ đó đưa ra nhận xét chính xác hơn về tính hiệu quả của mô hình.

3.2. Các kịch bản thử nghiệm và kết quả

Như đề cập trong mục 3.1, các kịch bản thử nghiệm được thực hiện theo 4 bước với 3 phương pháp sinh dữ liệu SMOTE, CGAN và WGAN trên 3 tập dữ liệu mở, gồm UNSW-NB15, CIC-UNSW-NB15 và CIC-IDS-2017 nhằm đưa ra phương pháp sinh dữ

liệu phù hợp nhất trên đa số các tập dữ liệu và sau đó sử dụng phương pháp sinh dữ liệu phù hợp nhất để khắc phục vấn đề mất cân bằng dữ liệu. Hai kịch bản thử nghiệm được triển khai thực hiện như sau:

- Kịch bản 1: Thử nghiệm và đánh giá các phương pháp sinh dữ liệu. Để thực hiện kịch bản này, đề án chọn các nhãn tấn công, xâm nhập trong các tập dữ liệu có nhiều mẫu để thử nghiệm và đánh giá. Cụ thể:
  + Lựa chọn một nhãn tấn công có nhiều mẫu thực (chẳng hạn Exploits, Fuzzers) kết hợp với dữ liệu bình thường thực theo tỷ lệ 1:1 để tạo tập gốc dữ liệu thực (ORIGIN_SET). Từ tập gốc thực chia ngẫu nhiên 75% cho huấn luyện (ORIGIN_TRAIN) và 25% cho kiểm tra (ORIGIN_TEST);
  + Sử dụng dữ liệu bình thường thực và dữ liệu tấn công thực từ tập ORIGIN_TRAIN để sinh dữ liệu mới tạo tập GEN_TRAIN với số mẫu mỗi nhãn tương đương ORIGIN_TRAIN. Các mẫu dữ liệu sinh gồm cả mẫu thuộc nhãn bình thường và mẫu thuộc nhãn tấn công;
  + Huấn luyện và kiểm tra mô hình gốc (ORIGIN_MODEL) sử dụng (ORIGIN_TRAIN, ORIGIN_TEST) và mô hình sinh (GEN_MODEL) sử dụng (GEN_TRAIN, ORIGIN_TEST);
  + Mô hình ORIGIN_MODEL huấn luyện từ dữ liệu gốc và mô hình GEN_MODEL huấn luyện từ dữ liệu sinh đều được kiểm tra trên dữ liệu gốc (ORIGIN_TEST) để so sánh hiệu suất. Hiệu suất của mô hình dựa trên các độ đo phân loại là căn cứ để đánh giá sự phù hợp của mô hình và phương pháp sinh dữ liệu.
- Kịch bản 2: Áp dụng phương pháp sinh dữ liệu phù hợp nhất có được từ Kịch bản 1 để sinh thêm dữ liệu cho các nhãn hiếm dữ liệu nhằm khắc phục vấn đề mất cân bằng dữ liệu. Huấn luyện và kiểm thử mô hình phát hiện xâm nhập sử dụng tập dữ liệu cân bằng.

3.2.1. Kịch bản 1: Thử nghiệm và đánh giá các phương pháp sinh dữ liệu

a, Thử nghiệm trên tập dữ liệu UNSW-NB15

Bảng 3.1 cung cấp các thành phần của tập gốc dữ liệu thực (ORIGIN_SET) trích từ tập dữ liệu UNSW-NB15.

Bảng 3.1: Các tập ORIGIN_SET trích từ tập dữ liệu UNSW-NB15

TT	Thành phần tập dữ liệu thử nghiệm	Số lượng mẫu huấn luyện	Số lượng mẫu kiểm tra
1	Exploits, Benign	71240	17810
2	Fuzzers, Benign	38793	9699

3	Reconnaissance, Benign	22379	5595
4	DoS, Benign	26164	6542

Theo đó các mẫu dữ liệu của 4 nhãn tấn công, gồm Exploits, Fuzzers, Reconnaissance và DoS được chọn để lần lượt kết hợp với các mẫu dữ liệu Benign để tạo thành các tập gốc ORIGIN_SET cho sinh các tập dữ liệu, huấn luyện xây dựng mô hình phát hiện. Từng cặp mô hình huấn luyện từ ORIGIN_SET và GEN_SET cho mỗi nhãn tấn công được kiểm tra trên tập ORIGIN_TEST để đánh giá hiệu suất phát hiện. Bảng 3.2 cung cấp các độ đo phát hiện của mô hình ORIGIN_MODEL huấn luyện từ dữ liệu gốc/thực và các mô hình GEN_MODEL huấn luyện từ dữ liệu sinh cho từng nhãn tấn công, bao gồm Exploits, Fuzzers, Reconnaissance và DoS.

Bảng 3.2: Hiệu suất phát hiện của mô hình ORIGIN_MODEL và các mô hình GEN_MODEL cho các nhãn tấn công trên tập dữ liệu UNSW-NB15

TT	Loại tấn công	Phương Pháp sinh	Accuracy	Precision	Recall	F1	FPR	FNR
1	Exploits	Thực	0.9969	0.9942	0.9996	0.9969	0.0058	0.0004
2	Exploits	SMOTE	0.9967	0.994	0.9996	0.9968	0.0061	0.0004
3	Exploits	CGAN	0.8028	0.9833	0.6159	0.7574	0.0104	0.3841
4	Exploits	WGAN	0.9905	0.9835	0.9978	0.9906	0.0167	0.0022
5	Reconnissance	Thực	0.9991	0.9982	1	0.9991	0.0018	0
6	Reconnissance	SMOTE	0.9986	0.9975	0.9996	0.9986	0.0025	0.0004
7	Reconnissance	CGAN	0.9430	0.9836	0.9010	0.9405	0.0150	0.0990
8	Reconnissance	WGAN	0.9887	0.9844	0.9932	0.9888	0.0157	0.0068
9	Fuzzers	Thực	0.9935	0.9878	0.9994	0.9935	0.0124	0.0006
10	Fuzzers	SMOTE	0.9931	0.9876	0.9988	0.9931	0.0126	0.0012
11	Fuzzers	CGAN	0.9902	0.9831	0.9975	0.9903	0.0171	0.0025
12	Fuzzers	WGAN	0.9907	0.9853	0.9963	0.9908	0.0148	0.0037
13	Dos	Thực	0.9974	0.9954	0.9994	0.9974	0.0046	0.0006
14	Dos	SMOTE	0.9972	0.9954	0.9991	0.9973	0.0046	0.0009
15	Dos	CGAN	0.9888	0.981	0.9969	0.9889	0.0193	0.0031
16	Dos	WGAN	0.9879	0.9825	0.9936	0.9880	0.0177	0.0064

b, Thử nghiệm trên tập dữ liệu CIC-UNSW-NB15

Bảng 3.3 cung cấp các thành phần của tập gốc dữ liệu thực (ORIGIN_SET) trích từ tập dữ liệu CIC-UNSW-NB15.

Bảng 3.3: Các tập ORIGIN_SET trích từ tập dữ liệu CIC-UNSW-NB15

TT	Thành phần tập dữ liệu thử nghiệm	Số lượng mẫu huấn luyện	Số lượng mẫu kiểm tra
1	Exploits, Benign	49521	12381
2	Fuzzers, Benign	47380	11846
3	Reconnaissance, Benign	26776	6694

Theo đó các mẫu dữ liệu của 3 nhãn tấn công, gồm Exploits, Fuzzers, và Reconnaissance được chọn để lần lượt kết hợp với các mẫu dữ liệu Benign để tạo thành các tập gốc ORIGIN_SET cho sinh các tập dữ liệu, huấn luyện xây dựng mô hình phát hiện. Từng cặp mô hình huấn luyện từ ORIGIN_SET và GEN_SET cho mỗi nhãn tấn công được kiểm tra trên tập ORIGIN_TEST để đánh giá hiệu suất phát hiện. Bảng 3.4 cung cấp các độ đo phát hiện của mô hình ORIGIN_MODEL huấn luyện từ dữ liệu gốc/thực và các mô hình GEN_MODEL huấn luyện từ dữ liệu sinh cho từng nhãn tấn công, bao gồm Exploits, Fuzzers và Reconnaissance.

Bảng 3.4: Hiệu suất phát hiện của mô hình ORIGIN_MODEL và các mô hình GEN_MODEL cho các nhãn tấn công trên CIC-UNSW-NB15

TT	Loại tấn công	Phương Pháp sinh	Accuracy	Precision	Recall	F1	FPR	FNR
1	Exploits	Thực	0.9888	0.9808	0.9971	0.9889	0.0195	0.0029
2	Exploits	SMOTE	0.9877	0.9813	0.9943	0.9878	0.0189	0.0057
3	Exploits	CGAN	0.8899	0.9889	0.7887	0.8775	0.0089	0.2113
4	Exploits	WGAN	0.9805	0.9707	0.9908	0.9807	0.0299	0.0092
5	Reconnissance	Thực	0.9901	0.9821	0.9985	0.9902	0.0182	0.0015
6	Reconnissance	SMOTE	0.9879	0.9829	0.9931	0.9880	0.0173	0.0069
7	Reconnissance	CGAN	0.9836	0.9693	0.9988	0.9838	0.0317	0.0012
8	Reconnissance	WGAN	0.9849	0.9724	0.9982	0.9851	0.0284	0.0018
9	Fuzzers	Thực	0.9874	0.9777	0.9976	0.9875	0.0228	0.0024
10	Fuzzers	SMOTE	0.9873	0.9778	0.9973	0.9875	0.0226	0.0027
11	Fuzzers	CGAN	0.9806	0.9743	0.9872	0.9807	0.026	0.0128
12	Fuzzers	WGAN	0.9849	0.9717	0.9988	0.9851	0.029	0.0012

c, Thử nghiệm trên tập dữ liệu CIC-IDS-2017

Bảng 3.5 cung cấp các thành phần của tập gốc dữ liệu thực (ORIGIN_SET) trích từ tập dữ liệu CIC-IDS-2017.

Bảng 3.5: Các tập ORIGIN_SET trích từ tập dữ liệu CIC-IDS-2017

TT	Thành phần tập dữ liệu thử nghiệm	Số lượng mẫu huấn luyện	Số lượng mẫu kiểm tra
1	DoS GoldenEye, Benign	16468	4118
2	FTP-Patator, Benign	12700	3176
3	SSH-Patator, Benign	9435	2359
4	DoS slowloris, Benign	9273	2319
5	DoS Slowhttptest, Benign	8798	2200

Theo đó các mẫu dữ liệu của 5 nhãn tấn công, gồm DoS GoldenEye, FTP-Patator, SSH-Patator, DoS slowloris và DoS Slowhttptest được chọn để lần lượt kết hợp với các mẫu dữ liệu Benign để tạo thành các tập gốc ORIGIN_SET cho sinh các tập dữ liệu, huấn luyện xây dựng mô hình phát hiện. Từng cặp mô hình huấn luyện từ ORIGIN_SET và GEN_SET cho mỗi nhãn tấn công được kiểm tra trên tập ORIGIN_TEST để đánh giá hiệu suất phát hiện. Bảng 3.6 cung cấp các độ đo phát hiện của mô hình ORIGIN_MODEL huấn luyện từ dữ liệu gốc/thực và các mô hình GEN_MODEL huấn luyện từ dữ liệu sinh cho từng nhãn tấn công, bao gồm DoS GoldenEye, FTP-Patator, SSH-Patator, DoS slowloris và DoS Slowhttptest.

Bảng 3.6: Hiệu suất phát hiện của mô hình ORIGIN_MODEL và các mô hình GEN_MODEL cho các nhãn tấn công trên tập dữ liệu CIC-IDS-2017

TT	Loại tấn công	Phương Pháp sinh	Accuracy	Precision	Recall	F1	FPR	FNR
1	DoS GoldenEye	Thực	0.9998	1	0.9995	0.9998	0	0.0005
2	DoS GoldenEye	SMOTE	0.9988	1	0.9976	0.9988	0	0.0024
3	DoS GoldenEye	CGAN	0.973	0.9594	0.9879	0.9734	0.0418	0.0121
4	DoS GoldenEye	WGAN	0.9954	0.9985	0.9922	0.9954	0.0015	0.0078
5	FTP-Patator	Thực	0.9997	1	0.9994	0.9997	0	0.0006
6	FTP-Patator	SMOTE	0.9997	1	0.9994	0.9997	0	0.0006
7	FTP-Patator	CGAN	0.9821	0.9676	0.9975	0.9823	0.0334	0.0025
8	FTP-Patator	WGAN	0.9981	1	0.9962	0.9981	0	0.0038
9	SSH-Patator	Thực	0.9996	1	0.9992	0.9996	0	0.0008
10	SSH-Patator	SMOTE	0.9996	1	0.9992	0.9996	0	0.0008

11	SSH-Patator	CGAN	0.9517	0.9216	0.9873	0.9533	0.0839	0.0127
12	SSH-Patator	WGAN	0.9771	0.9991	0.955	0.9766	0.0008	0.045
13	DoS slowloris	Thực	0.9991	1	0.9983	0.9991	0	0.0017
14	DoS slowloris	SMOTE	0.9978	1	0.9957	0.9978	0	0.0043
15	DoS slowloris	CGAN	0.9491	0.9312	0.9698	0.9501	0.0716	0.0302
16	DoS slowloris	WGAN	0.9875	0.9956	0.9793	0.9874	0.0043	0.0207
17	DoS Slowhttptest	Thực	0.9982	1	0.9964	0.9982	0	0.0036
18	DoS Slowhttptest	SMOTE	0.9959	1	0.9918	0.9959	0	0.0082
19	DoS Slowhttptest	CGAN	0.8564	0.9804	0.7273	0.8351	0.0145	0.2727
20	DoS Slowhttptest	WGAN	0.9927	0.9901	0.9955	0.9927	0.01	0.0045

d, Nhận xét về kết quả thử nghiệm của Kịch bản 1

Các kết quả thử nghiệm cho trên các Bảng 3.2, 3.4 và 3.6 cho thấy các mô hình GEN_MODEL sử dụng dữ liệu sinh bởi các phương pháp SMOTE và WGAN cho các độ đo phân loại ổn định và bám sát các độ đo của mô hình ORIGIN_MODEL sử dụng dữ liệu thực với tất cả các nhãn trên cả 3 tập dữ liệu sử dụng. Trong khi đó các mô hình GEN_MODEL sử dụng dữ liệu sinh bởi các phương pháp CGAN cho kết quả không ổn định – chưa phù hợp với các tập dữ liệu thử nghiệm. Mặc dù mô hình dựa trên SMOTE cho hiệu suất phát hiện cao hơn so với mô hình dựa trên WGAN, nhưng khác biệt là không lớn. Đồng thời, nhiều nghiên cứu đã xác nhận SMOTE thường hoạt động tốt trên các đặc trưng rời rạc và có phân phối đơn giản, nhưng không hoạt động tốt trên các đặc trưng liên tục hoặc có phân phối phức tạp [8][16], *nên đề án lựa chọn WGAN là phương pháp sinh dữ liệu tấn công mạng cho thử nghiệm Kịch bản 2 do WGAN có khả năng khắc phục được các vấn đề trên của SMOTE [16]*.

3.2.2. Kịch bản 2: Áp dụng WGAN để khắc phục vấn đề mất cân bằng dữ liệu

Trong kịch bản này, đề án sử dụng phương pháp WGAN đã được đánh giá là phương pháp phù hợp nhất theo Kịch bản 1 để khắc phục vấn đề mất cân bằng dữ liệu của tập dữ liệu UNSW-NB15.

Bảng 3.7: Các thành phần của tập dữ liệu UNSW-NB15

TT	Nhãn dữ liệu	Số lượng mẫu
1	Benign	2218764
2	Generic	215481
3	Exploits	44525
4	Fuzzers	24246
5	DoS	16353
6	Reconnaissance	13987
7	Analysis	2677
8	Backdoor	1795

9	Shellcode	1511
10	Backdoors	534
11	Worms	174

Có thể thấy tập dữ liệu UNSW-NB15 cho trên Bảng 3.7, dữ liệu bình thường (Benign) có số lượng mẫu vượt trội so với số lượng mẫu của các nhãn tấn công. Đặc biệt, một số nhãn tấn công, như Analysis, Backdoor, Shellcode, Backdoors và Worms có rất ít mẫu dữ liệu. Điều này ảnh hưởng nghiêm trọng đến hiệu suất phát hiện của mô hình IDS dựa trên học máy với các dạng tấn công có ít mẫu dữ liệu. Đề án thực hiện 2 thử nghiệm với kịch bản này như sau:

- Sử dụng dữ liệu gốc/thực từ tập dữ liệu UNSW-NB15 (bình thường + tấn công) để huấn luyện và kiểm tra. Số mẫu của các nhãn nhiều dữ liệu được giới hạn ở 13000 mẫu để giảm ảnh hưởng của các nhãn nhiều dữ liệu lên nhóm nhãn ít dữ liệu. Tập dữ liệu này được đặt tên là UNSW-NB15-ORIGIN và các thành phần của nó cho trên Bảng 3.8;
- Sử dụng dữ liệu gốc/thực từ tập dữ liệu UNSW-NB15 (bình thường + tấn công) với 6 nhãn (Benign, Generic, Exploits, Fuzzers, DoS, Reconnaissance) và sinh thêm dữ liệu cho 5 nhãn (Analysis, Backdoor, Shellcode, Backdoors và Worms) để khắc phục vấn đề mất cân bằng dữ liệu, sau đó sử dụng tập kết hợp để huấn luyện và kiểm tra. Số mẫu mỗi nhãn lấy từ tập gốc và sinh mới là 13000 để đảm bảo tập dữ liệu là cân bằng giữa tất cả các nhãn. Tập dữ liệu này được đặt tên là UNSW-NB15-GEN và các thành phần của nó cho trên Bảng 3.9.

Bảng 3.8: Các thành phần của các tập dữ liệu UNSW-NB15-ORIGIN

TT	**Nhãn dữ liệu**	**Số mẫu dữ liệu**				
		Gốc	Sinh	Tổng	Huấn luyện	Kiểm tra
1	Benign	13000	0	13000	10000	3000
2	Generic	13000	0	13000	10000	3000
3	Exploits	13000	0	13000	10000	3000
4	Fuzzers	13000	0	13000	10000	3000
5	DoS	13000	0	13000	10000	3000
6	Reconnaissance	13000	0	13000	10000	3000
7	Analysis	2677	0	2677	2059	618
8	Backdoor	1795	0	1795	1380	415
9	Shellcode	1511	0	1511	1162	349
10	Backdoors	534	0	534	411	123
11	Worms	174	0	174	134	40

Bảng 3.9: Các thành phần của các tập dữ liệu UNSW-NB15-GEN

TT	Nhãn dữ liệu	Số mẫu dữ liệu				
		Gốc	Sinh	Tổng	Huấn luyện	Kiểm tra
1	Benign	13000	0	13000	10000	3000
2	Generic	13000	0	13000	10000	3000
3	Exploits	13000	0	13000	10000	3000
4	Fuzzers	13000	0	13000	10000	3000
5	DoS	13000	0	13000	10000	3000
6	Reconnaissance	13000	0	13000	10000	3000
7	Analysis	2677	10323	13000	10000	3000
8	Backdoor	1795	11205	13000	10000	3000
9	Shellcode	1511	11489	13000	10000	3000
10	Backdoors	534	12466	13000	10000	3000
11	Worms	174	12826	13000	10000	3000

Bảng 3.10: Hiệu suất phát hiện cho các nhãn trên bộ dữ liệu UNSW-NB15-ORIGIN

TT	Nhãn dữ liệu	Các độ đo hiệu suất phát hiện		
		Precision	***Recall***	***F1-Score***
1	Benign	0.9993	0.9837	0.9914
2	Generic	0.9962	0.9730	0.9845
3	Exploits	0.7329	0.5277	0.6136
4	Fuzzers	0.9155	0.8700	0.8922
5	DoS	0.4246	0.3603	0.3898
6	Reconnaissance	0.9242	0.8330	0.8762
7	Analysis	0.2412	0.2217	0.2310
8	Backdoor	0.0231	0.1108	0.0383
9	Shellcode	0.7849	0.7736	0.7792
10	Backdoors	0.1836	0.6911	0.2901
11	Worms	0.7222	0.6500	0.6842

Bảng 3.11: Hiệu suất phát hiện cho các nhãn trên bộ dữ liệu UNSW-NB15-GEN

TT	Nhãn dữ liệu	Các độ đo hiệu suất phát hiện		
		Precision	***Recall***	***F1-Score***
1	Benign	0.9990	0.9840	0.9914
2	Generic	0.9959	0.9730	0.9843
3	Exploits	0.6277	0.5523	0.5876
4	Fuzzers	0.8868	0.8773	0.8820
5	DoS	0.4815	0.7143	0.5752
6	Reconnaissance	0.9211	0.8287	0.8724
7	Analysis	0.9506	0.8333	0.8881
8	Backdoor	0.9858	0.8813	0.9307
9	Shellcode	0.9710	0.9813	0.9761
10	Backdoors	0.9574	0.9600	0.9587
11	Worms	0.9947	0.9960	0.9953

Bảng 3.10 cung cấp các độ đo phát hiện của mô hình IDS dựa trên thuật toán RF trên các nhãn sử dụng bộ dữ liệu gốc UNSW-NB15-ORIGIN. Có thể thấy, hiệu suất phát hiện của mô hình với các nhãn ít dữ liệu (Analysis, Backdoor, Shellcode, Backdoors và Worms) đều khá thấp. Đặc biệt, mô hình gần như không thể phát hiện tấn công Backdoor, với độ đo F1 chỉ là 3,83%. Bảng 3.11 cung cấp các độ đo phát hiện của mô hình IDS dựa trên thuật toán RF trên các nhãn sử dụng bộ dữ liệu gốc UNSW-NB15-GEN. Có thể thấy, hiệu suất phát hiện của mô hình với các nhãn ít dữ liệu (Analysis, Backdoor, Shellcode, Backdoors và Worms) đều được cải thiện đáng kể, với độ đo F1 đạt từ 88 – trên 99%. Đặc biệt, mô hình này phát hiện hiệu quả tấn công Backdoor với độ đo F1 đạt 93,07%.

3.3. Nhận xét và đánh giá chung

Đề án đã thử nghiệm 3 phương pháp sinh dữ liệu tiêu biểu bao gồm SMOTE, CGAN và WGAN trên 3 tập dữ liệu UNSW-NB15, CIC-UNSW-NB15 và CIC-IDS2017. Từ các kết quả cho bởi Kịch bản thử nghiệm 1 tại các Bảng 3.2, 3.4 và 3.6, có thể rút ra một số nhận xét:

– Các mô hình huấn luyện sử dụng dữ liệu sinh bởi các phương pháp SMOTE và WGAN đạt các độ đo phát hiện ổn định và bám sát mô hình huấn luyện sử dụng dữ liệu gốc trên tất cả các thử nghiệm với các nhãn tấn công khác nhau của 3 tập dữ liệu tấn công mạng. Điều này càng khẳng định hiệu quả của 2 phương pháp sinh dữ liệu trên. Mô hình dựa trên WGAN cho các độ đo thấp hơn một chút so với SMOTE, nhưng WGAN đã được chứng minh là có khả năng hoạt động ổn định trên cả các đặc trưng rời rạc có phân bố đơn giản cũng như phức tạp.

– Các mô hình huấn luyện sử dụng dữ liệu sinh bởi phương pháp CGAN đạt các độ đo phát hiện không ổn định và thấp hơn đáng kể mô hình huấn luyện sử dụng dữ liệu gốc. Lý do có thể do CGAN mô hình hoạt động của phương pháp này không phù hợp với phân phối của các đặc trưng của 3 tập dữ liệu thử nghiệm.

Đề án sử dụng các nhãn có nhiều mẫu dữ liệu để đánh giá hiệu suất và độ tin cậy của các phương pháp sinh dữ liệu là phù hợp do có đủ lượng lớn dữ liệu huấn luyện mô hình gốc/cơ sở làm căn cứ so sánh với các mô hình huấn luyện sử dụng dữ liệu sinh. Ngoài ra, việc thử nghiệm với nhiều nhãn tấn công trong 3 tập dữ liệu phổ biến đảm bảo được tính ổn định và tin cậy của phương pháp lựa chọn.

Dựa trên kết quả thử nghiệm Kịch bản 1, Đề án thực hiện thử nghiệm Kịch bản 2, trong đó sử dụng phương pháp WGAN để sinh thêm dữ liệu cho 5 nhãn ít dữ liệu trong tập UNSW-NB15. Các kết quả thử nghiệm cho trên các Bảng 3.10 và 3.11 cho thấy, hiệu suất phát hiện 5 nhãn ít dữ liệu, gồm Analysis, Backdoor, Shellcode, Backdoors và Worms trong tập UNSW-NB15 đã được cải thiện đáng kể khi bổ sung thêm dữ liệu sinh bởi WGAN. Điều này khẳng định tiềm năng của phương pháp sinh dữ liệu tấn công mạng giúp tăng hiệu quả của mô hình IDS dựa trên học máy, giảm công sức thu thập và làm sạch dữ liệu.

3.4. Kết chương

Trong chương này, luận văn đã trình bày quy trình thực nghiệm bao gồm: mô hình thử nghiệm, kịch bản xây dựng, môi trường cài đặt và đánh giá kết quả. Mô hình thử nghiệm được thiết kế nhằm đánh giá hiệu quả của ba kỹ thuật sinh dữ liệu tấn công mạng: SMOTE, CGAN và WGAN, thông qua các chỉ số Accuracy, Precision, Recall, F1-score, FPR và FNR.

Các thử nghiệm được triển khai trên ba bộ dữ liệu UNSW-NB15, CIC-UNSW-NB15 và CIC-IDS2017. Kết quả cho thấy dữ liệu thật vẫn đạt hiệu năng cao nhất, trong khi WGAN tái tạo dữ liệu tốt và cho độ chính xác tiệm cận dữ liệu gốc. SMOTE ổn định nhưng hạn chế về đa dạng, còn CGAN sinh dữ liệu phong phú hơn song độ ổn định chưa cao. Nhìn chung, chương này đã chứng minh tính khả thi và hiệu quả của việc ứng dụng các phương pháp sinh dữ liệu trong phát hiện tấn công mạng.

III. KẾT LUẬN

1. Kết luận chung

Đề án với đề tài "Nghiên cứu các phương pháp sinh dữ liệu tấn công mạng" đã tập trung khảo sát, phân tích và thực nghiệm ba kỹ thuật tiêu biểu đại diện cho ba nhóm phương pháp sinh dữ liệu: SMOTE (thống kê truyền thống), CGAN và WGAN (học sâu). Đề án đã hoàn thành các nội dung sau:

- Khảo sát các bộ dữ liệu tấn công mạng, vấn đề mất cân bằng dữ liệu, một số nghiên cứu liên quan về sinh dữ liệu tấn công mạng;
- Khảo sát các kỹ thuật, phương pháp sinh dữ liệu tấn công mạng;
- Xây dựng mô hình, kịch bản thử nghiệm, thử nghiệm và đánh giá phương pháp sinh dữ liệu tấn công mạng;
- Áp dụng phương pháp sinh dữ liệu tấn công mạng phù hợp nhất để khắc phục vấn đề mất cân bằng dữ liệu.

2. Các hạn chế của đề án

Mặc dù đã đạt được một số kết quả, Đề án còn có một số hạn chế sau:

- Đề án mới thử nghiệm sinh dữ liệu trên 3 tập dữ liệu với 4 dạng tấn công mạng phổ biến có các trường dữ liệu rời rạc và phân phối tương đối đơn giản. Cần có các thử nghiệm trên các tập dữ có kích thước lớn với nhiều dạng tấn công để có đánh giá toàn diện hơn.
- Đề án mới sử dụng 6 độ đo đánh giá phổ biến, gồm Độ chính xác chung (Accuracy), Độ chính xác dương tính (Precision), Độ nhạy (Recall), Tỷ lệ dương tính giả (FPR) và Tỷ lệ âm tính giả (FNR) để so sánh, đánh giá hiệu suất phát hiện của các mô hình phân loại tấn công dựa trên dữ liệu gốc và dữ liệu sinh. Các độ đo khác cho tập dữ liệu sinh, cũng như so sánh với tập dữ liệu gốc, như giá trị trung bình, trung vị, phương sai, phân phối... chưa được sử dụng.
- Đề án mới khảo sát 3 phương pháp sinh dữ liệu bao gồm: SMOTE, CGAN và WGAN, mà chưa thử nghiệm các phương pháp sinh dữ liệu khác, hoặc kết hợp các phương pháp sinh dữ liệu.

3. Hướng nghiên cứu tiếp theo

Từ kết quả đạt được, các hướng nghiên cứu tiếp theo có thể bao gồm:

- Mở rộng phạm vi thử nghiệm với các bộ dữ liệu mới, đa dạng hơn về loại hình tấn công và quy mô mạng.

- Thử nghiệm thêm các phương pháp sinh dữ liệu khác, hoặc kết hợp nhiều phương pháp sinh dữ liệu (ví dụ: lai giữa thống kê và học sâu) để tối ưu hóa giữa độ chính xác và tính đa dạng.
- Tích hợp dữ liệu sinh vào mô hình IDS thực tế, đánh giá hiệu quả trong môi trường mạng động, giúp kiểm chứng khả năng áp dụng thực tiễn.
- Nghiên cứu cơ chế đánh giá chất lượng dữ liệu sinh bằng các độ đo chuyên biệt phù hợp hơn, thay vì chỉ sử dụng các thước đo phân loại truyền thống. Các độ đo có thể bổ sung, như giá trị trung bình, trung vị, phương sai, phân phối chuẩn, phân phối Poisson...

Kết quả đạt được cho thấy hướng tiếp cận này có thể cải thiện hiệu quả huấn luyện hệ thống phát hiện xâm nhập. Trong tương lai, việc mở rộng nghiên cứu sang các mô hình sinh tiên tiến và thử nghiệm trên môi trường mạng thực tế sẽ giúp phát triển lĩnh vực phát hiện xâm nhập nói riêng và an ninh thông tin nói chung.

IV.DANH MỤC CÁC TÀI LIỆU THAM KHẢO

[1] Q. hội, "Luật An ninh mạng, Luật số: 24/2018/QH14," Quốc hội Việt Nam, Hà Nội, 2018.

[2] K. W. B. L. O. H. W. P. K. Nitesh V. Chawla, "SMOTE: Synthetic Minority Over-sampling Technique," Journal of Artificial Intelligence Research, 2002.

[3] J. P.-A. M. M. B. X. D. W.-F. S. O. A. C. Y. B. Ian J. Goodfellow, "Generative Adversarial Networks," 2014.

[4] A. H. T. K. T. Andrey Ferriyan, "Generating Network Intrusion Detection Dataset Based on Real and Encrypted Synthetic Attack Traffic," *Applied Sciences,* tập 11, số 17, 2021.

[5] S. S. F. S. Zongwen Fan, "Sampling-Based Machine Learning Models for Intrusion Detection in Imbalanced Dataset," *Electronics ,* tập 13, số 10, 2024.

[6] K. W. F. a. V. L. L. T. Xinxing Zhao, "Enhancing Network Intrusion Detection Performance using Generative," Engineering, 2024.

[7] S. W. D. S. D. L. A. H. Markus Ring, "A survey of network-based intrusion detection data sets," *Computers & Security,* tập 86, pp. 147-167, 2019.

[8] Y. B. E. A. G. a. S. L. Haibo He, "ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced," 2008.

[9] N. I. H. M. A. R. M. A. J. Md Hasan Shahriar, "G-IDS: Generative Adversarial Networks Assisted Intrusion Detection System," *Cryptography and Security,* 2006.

[10] S. Z. Y. S. D. K. C. W. Qiumei Cheng, "Packet-Level Adversarial Network Traffic Crafting using Sequence Generative Adversarial Networks," *Cryptography and Security,* 2021.

[11] M. S. A. C.-I. K. V. Lei Xu, "Modeling Tabular Data using Conditional GAN," 2019.

[12] L. H. C. Q. W. Z. L. Hongpo Zhang, "An effective convolutional neural network based on SMOTE and Gaussian mixture model for intrusion detection in imbalanced dataset," *Computer Networks,* tập 177, 2020.

[13] Z. T. C. J. W. Y. Z. Q. ZHANG Yang, "Research on Network Intrusion Detection Based on SMOTE Algorithm and Machine Learning," *ransactions of Beijing institute of Technology,* tập 39, số 12, 2019.

[14] P. R. Jelena Mirkovic, "A Taxonomy of DDoS Attack," *ACM SIGCOMM Computer Communications Review,* tập 34, 2004.

[15] F. M. J. I. O. Carlos Sarraute, "Simulation of Computer Network Attacks," *Core Security Technologies,* 2010.

[16] S. C. L. B. Martin Arjovsky, “Wasserstein Generative Adversarial Networks,” *Machine Learning,* pp. 214-223, 2017.

[17] F. A. M. A. V. D. Ishaan Gulrajan, “Improved Training of Wasserstein GANs,” 2017.

[18] S. O. Mehdi Mirza, “Conditional Generative Adversarial Nets,” *Machine Learning,* 2014.

[19] CIC, “CIC UNSW-NB15 Augmented Dataset,” 2015.

[20] UNSW, “The UNSW-NB15 Dataset,” 2015.

[21] CIC, “Intrusion detection evaluation dataset (CIC-IDS2017),” 2017.

[22] CIC, “ISCX NSL-KDD dataset 2009,” 2009.

[23] UNSW, “The Bot-IoT Dataset,” 2018.

[24] UNSW, “The TON_IoT Datasets,” 2020.

[25] C. Y. Wijaya, ,"7 SMOTE Variations for Oversampling” 2023.

[26] Master's in Data Science“What is Undersampling,” 2022.

[27] P. D. Khanh, “GANWasserstein”, 2020.

[28] H. Dwivedi, “gan loss functions,” 2023.

[29] “Advanced GAN techiques” "https://apxml.com/courses/generative-adversarial-networks-gans/chapter-3-gan-training-stabilization/wgan-weight-clipping".

[30] P. D. Khanh, "ConditionalGAN" “https://phamdinhkhanh.github.io/2020/08/09/ConditionalGAN.html”, 2020.

[31] K. Pykes, “Variational Autoencoders: How They Work and Why They Matter,” 2024.

[32] N. M. Hải, “LLM là gì? Khám phá ứng dụng và xu hướng của mô hình ngôn ngữ lớn,” 2025.

[33] imbalanced-learn.org , “KMeansSMOTE”, https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.KMeansSMOTE.html.

[34] S. A. M. D. Sanket Kamthe, “Copula Flows for Synthetic Data Generation,” 2021.