

BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC CẦN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



LUẬN VĂN TỐT NGHIỆP
NGÀNH TRUYỀN THÔNG VÀ MẠNG MÁY TÍNH

Đề tài

ĐÁNH GIÁ CÁC KỸ THUẬT HỌC SÂU
TRONG PHÁT HIỆN XÂM NHẬP TRÊN CÁC
MÔI TRƯỜNG MẠNG KHÔNG ĐỒNG NHẤT

Giảng viên hướng dẫn:

Ts. Nguyễn Hữu Văn Long

Sinh viên thực hiện:

Dương Thị Kiều Trâm

MSSV: B2110953

Khóa 47

Cần Thơ, 8/2025

BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC CẦN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



LUẬN VĂN TỐT NGHIỆP
NGÀNH TRUYỀN THÔNG VÀ MẠNG MÁY TÍNH

Đề tài

ĐÁNH GIÁ CÁC KỸ THUẬT HỌC SÂU
TRONG PHÁT HIỆN XÂM NHẬP TRÊN CÁC
MÔI TRƯỜNG MẠNG KHÔNG ĐỒNG NHẤT

Giảng viên hướng dẫn:

Ts. Nguyễn Hữu Văn Long

Sinh viên thực hiện

Dương Thị Kiều Trâm

MSSV: B2110953

Khóa 47

Cần Thơ, 8/2025

XÁC NHẬN CHỈNH SỬA LUẬN VĂN
THEO YÊU CẦU CỦA HỘI ĐỒNG

Tên luận văn (tiếng Việt và tiếng Anh): Đánh giá các kỹ thuật học sâu trong phát hiện xâm nhập trên các môi trường mạng không đồng nhất (Performance and Robustness Evaluation of Deep Learning Models for Intrusion Detection Across Heterogeneous Network Environments).

Họ tên sinh viên: Dương Thị Kiều Trâm

MSSV: B2110953

Mã lớp: DI21T9A1

Đã báo cáo tại hội đồng ngành Mạng máy tính và Truyền thông dữ liệu.

Ngày báo cáo: 22/08/2025

Luận văn đã được chỉnh sửa theo góp ý của Hội đồng.

Cần Thơ, ngày tháng năm 2025

Giáo viên hướng dẫn

(Ký và ghi họ tên)

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN



NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN



LỜI CẢM ƠN

Trong suốt những năm học đại học đã qua và hơn bốn tháng thực hiện đề tài Luận Văn Tốt Nghiệp, em đã nhận được sự quan tâm, hỗ trợ tận tình từ Quý Thầy Cô, gia đình và bạn bè. Đây là nguồn động lực to lớn, giúp em có cơ hội học tập, trau dồi và ngày càng trưởng thành như hôm nay.

Em xin bày tỏ lòng biết ơn chân thành đến Quý Thầy Cô tại Trường Công nghệ Thông tin và Truyền thông, Trường Đại học Cần Thơ, những người không chỉ trang bị cho em kiến thức chuyên môn vững chắc mà còn truyền cảm hứng về tinh thần trách nhiệm và niềm đam mê trong học tập cũng như nghiên cứu.

Em muốn bày tỏ lòng biết ơn chân thành Thầy Nguyễn Hữu Vân Long, người đã nhiệt tình hướng dẫn và chỉ bảo em trong suốt quá trình làm Luận Văn Tốt Nghiệp. Nhờ những kiến thức và kinh nghiệm mà Thầy chia sẻ, em có thể định hướng được rõ ràng hướng đi của mình và hoàn thành tốt các nhiệm vụ trong quá trình thực hiện. Sự hỗ trợ quý báu từ Thầy đã giúp em vượt qua thử thách ban đầu và hoàn thiện bài báo cáo này một cách tốt nhất.

Bên cạnh đó, em muốn gửi lời cảm ơn đến gia đình và bạn bè, những người đã luôn đồng hành, động viên tinh thần và cổ vũ em trong suốt quá trình làm Luận Văn.

Trong quá trình hoàn thành bài báo cáo, em đã cố gắng hết sức để hoàn thành tốt nhất có thể, nhưng không tránh khỏi những sai sót và thiếu sót. Em rất mong nhận được sự thông cảm và ý kiến đóng góp từ Quý Thầy Cô để em có thể rút kinh nghiệm và hoàn thiện hơn.

Dương Thị Kiều Trâm

Lớp Truyền thông & Mạng máy tính A1 K47

TÓM TẮT

An ninh mạng ngày càng đối mặt với nhiều hình thức tấn công phức tạp và đa dạng, đòi hỏi các hệ thống phát hiện xâm nhập (Intrusion Detection System – IDS) phải có khả năng nhận diện chính xác và kịp thời. Luận văn này tập trung nghiên cứu và đánh giá hiệu suất của các mô hình học sâu CNN, LSTM, GRU, MLP và các mô hình kết hợp CNN–LSTM, CNN–MLP trong nhiệm vụ phát hiện tấn công mạng. Các tập dữ liệu chuẩn được sử dụng bao gồm CSE-CIC-IDS-2018, CICIoT2023 và ICS-Flow, trong các tập dữ liệu này chứa các lưu lượng mạng bình thường và bất thường, phản ánh đa dạng các kịch bản tấn công thực tế như DoS/DDoS, Brute Force, tấn công Web, Reconnaissance, Spoofing, Replay và nhiều biến thể khác. Để khắc phục tình trạng mất cân bằng dữ liệu vốn phô biến trong các bài toán phát hiện tấn công, nghiên cứu áp dụng các bước tiền xử lý sử dụng kỹ thuật cân bằng dữ liệu là SMOTE kết hợp Tomek Links. Các mô hình được huấn luyện và đánh giá dựa trên các chỉ số thông thường như Accuracy, Precision, Recall, F1-score mà còn đo lường cả hiệu quả tính toán thông qua mức tiêu thụ tài nguyên GPU/CPU. Kết quả đánh giá cho thấy mỗi mô hình đều có những ưu, nhược điểm riêng khi triển khai trong các môi trường mạng khác nhau. Từ đó, nghiên cứu này cung cấp một tài liệu tham khảo có giá trị về các kỹ thuật học sâu (Deep Learning), có thể hỗ trợ triển khai thực tế các hệ thống phát hiện xâm nhập (IDS) thế hệ mới với hiệu quả và đáng tin cậy hơn.

ABSTRACT

Cybersecurity is increasingly challenged by sophisticated and diverse forms of attacks, requiring Intrusion Detection Systems (IDS) to accurately and promptly identify threats. This thesis investigates and evaluates the effectiveness of deep learning models—including CNN, LSTM, GRU, MLP , and hybrid architectures such as CNN–LSTM and CNN–MLP —for network attack detection tasks. The benchmark datasets employed in this study include CSE-CIC-IDS-2018, CICIoT2023, and ICS-Flow, which contain both normal and malicious network traffic, reflecting a wide range of real-world attack scenarios such as DoS/DDoS, Brute Force, Web attacks, Reconnaissance, Spoofing, Replay, and other variants. To address the inherent data imbalance problem in intrusion detection, preprocessing steps incorporating data balancing techniques such as SMOTE combined with Tomek Links are applied. The models are trained and evaluated not only on standard performance metrics (Accuracy, Precision, Recall, and F1-score) but also on computational efficiency, measured through GPU/CPU resource utilization. The evaluation results reveal that each model demonstrates distinct strengths and weaknesses when deployed across different network environments. Consequently, this research provides a valuable reference on the application of deep learning techniques, offering practical insights for developing next-generation IDS with enhanced efficiency and reliability.

Keywords: Deep learning , IDS, NIDS , IoT ,ICS

MỤC LỤC

CHƯƠNG 1: GIỚI THIỆU	1
1.1. Giới thiệu đề tài	1
1.2. Mục tiêu đề tài	3
1.3. Những nghiên cứu liên quan	3
1.4. Đối tượng và phạm vi nghiên cứu.....	6
1.5. Phương pháp nghiên cứu.....	7
1.6. Bố cục luận văn.....	7
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	8
2.1. Tổng quan về hệ thống phát hiện xâm nhập	8
2.1.1. Hệ thống phát hiện xâm nhập dựa trên phương pháp triển khai (IDS Based on Deployment Methods)	8
2.1.2. Hệ thống phát hiện xâm nhập dựa trên phương pháp phát hiện (IDS Based on Detection Methods)	9
2.2. Các môi trường ứng dụng hệ thống phát hiện xâm nhập (IDS)	11
2.2.1. Mạng doanh nghiệp (Enterprise network).....	11
2.2.2. Mạng IoT (Internet of things network).....	12
2.2.3. Mạng công nghiệp (Industrial Network)	13
2.3. Deep Learning	14
2.3.1. CNN (Convolutional Neural Network)	15
2.3.2. LSTM (Long Short-Term Memory).....	17
2.3.3. GRU (Gated Recurrent Unit).....	18
2.3.4. MLP (Multi-Layer Perceptron)	19
2.3.5. Mô hình kết hợp (Hybird Models)	20
2.4. Nền tảng Kaggle.....	22

CHƯƠNG 3: PHƯƠNG PHÁP THỰC HIỆN	23
3.1. Mô hình hoá bài toán.....	23
3.2. Tổng quan về các tập dữ liệu	24
3.2.1. Tập dữ liệu CSE-CIC-IDS-2018	25
3.2.2. Tập dữ liệu CIC - IoT -2023.....	29
3.2.3. Tập dữ liệu ICS-Flow	32
3.3. Data Pre-Processing (Chuẩn bị và tiền xử lý dữ liệu)	35
3.3.1. Dataset Merging (Gộp dữ liệu).....	35
3.3.2. Data Cleaning (Làm sạch dữ liệu).....	37
3.3.3. Data Encoding (Mã hóa dữ liệu)	37
3.3.4. Data Normalization (Chuẩn hóa dữ liệu)	38
3.3.5. Feature Selection (Lựa chọn đặc trưng)	39
3.3.6. Data Splitting (Chia dữ liệu)	39
3.3.7. Data Balancing (Cân bằng dữ liệu)	40
3.4. Huấn luyện các mô hình (Models Training)	43
3.4.1. Thông số cấu hình ban đầu của các kỹ thuật học sâu.....	46
3.4.2. Cấu trúc các mô hình áp dụng	47
3.4.3. Các Tiêu chí đánh giá mô hình.....	51
3.5. Các công cụ hỗ trợ hỗ trợ dùng để huấn luyện mô hình học sâu	53
CHƯƠNG 4: KẾT QUẢ THỰC NGHIỆM	54
4.1. Thiết lập môi trường.....	54
4.2. Thiết kế và triển khai.....	54
4.2.1. Thiết kế	54
4.2.2. Triển khai.....	55
4.3. Kết quả thực nghiệm	56
4.3.1. Kết quả trên kịch bản 1	57

4.3.2. Kết quả trên kịch bản 2.....	68
4.3.3. Kết quả trên kịch bản 3.....	79
4.4. . Đánh giá tổng hợp	90
4.4.1. Đánh giá tổng hợp các kịch bản	90
4.4.2. Đánh giá tổng hợp hiệu quả của từng kiến trúc mô hình	91
4.4.3. Đánh giá tổng hợp mức độ sử dụng tài nguyên của các mô hình	93
CHƯƠNG 5: KẾT LUẬN.....	95
5.1. Kết quả đạt được	95
5.2. Hạn chế.....	96
5.3. Hướng phát triển	96
TÀI LIỆU THAM KHẢO	97

MỤC LỤC HÌNH ẢNH

Hình 1:	Sơ đồ hình phân loại Hệ thống phát hiện xâm nhập (IDS)	8
Hình 2:	Mô tả mạng doanh nghiệp (Enterprise network) [15].....	11
Hình 3:	Mô tả mạng IoT (Internet of Things Network) [16]	12
Hình 4:	Mô tả mạng ICS [17]	13
Hình 5:	Kiến trúc mạng CNN (Convolutional Neural Network) [23]	15
Hình 6:	Kiến trúc mạng LSTM (Long Short-Term Memory) [24].....	17
Hình 7:	Kiến trúc mạng GRU (Gated Recurrent Unit) [25]	18
Hình 8:	Kiến trúc MLP (Multi-layer Perceptron)[26].....	19
Hình 9:	Mô hình tổng quan	24
Hình 10:	Cấu trúc mạng mô phỏng của tập dữ liệu CSE-CIC-IDS-2018.....	25
Hình 11:	Thống kê số lượng mẫu trong tập dữ liệu CSE-CIC-IDS-2018[27].....	26
Hình 12:	Các thiết bị IoT được triển khai của tập dữ liệu CIC-IoT-2023 [28].....	29
Hình 13:	Thống kê số lượng mẫu trong tập dữ liệu CIC-IoT-2023.....	30
Hình 14:	Cấu trúc mạng mô hình thử nghiệm hệ thống ICS [29].....	32
Hình 15:	Thống kê số lượng mẫu tập dữ liệu ICS-Flow[29].....	33
Hình 16:	Mô hình hoá chi tiết cho bước Tiền xử lý dữ liệu.....	35
Hình 17:	Cấu hình kiến trúc CNN với 5 lớp ẩn.	47
Hình 18:	Cấu hình kiến trúc LSTM với 5 lớp ẩn.	48
Hình 19:	Cấu hình kiến trúc GRU với 5 lớp ẩn.	48
Hình 20:	Cấu hình kiến trúc MLP với 5 lớp ẩn.....	49
Hình 21:	Cấu hình kiến trúc CNN-LSTM với 5 lớp ẩn.	49
Hình 22:	Cấu hình kiến trúc CNN-MLP với 5 lớp ẩn.....	50
Hình 23:	Trước và sau khi cân bằng của tập dữ liệu (1)CSE-CIC-IDS-2018.	58
Hình 24:	Biểu đồ chính xác (Accuracy) tổng quát của các mô hình trên kịch bản 1.....	58

Hình 25: Biểu đồ thời gian huấn luyện và suy luận tổng quát của các mô hình trên kịch bản 1	59
Hình 26: Lịch sử huấn luyện (training history) trên kịch bản 1	61
Hình 27: Ma trận nhầm lẫn (confusion matrix) trên kịch bản 1.	63
Hình 28: Trước và sau khi cân bằng của tập dữ liệu (2)CIC-IoT-2023.....	69
Hình 29: Biểu đồ chính xác (Accuracy) tổng quát của các mô hình trên kịch bản 2.....	69
Hình 30: Biểu đồ thời gian huấn luyện và suy luận tổng quát của các mô hình trên kịch bản 2.....	70
Hình 31: Lịch sử huấn luyện (training history) trên kịch bản 2.....	72
Hình 32: Ma trận nhầm lẫn (confusion matrix) trên kịch bản 2.	74
Hình 33: Trước và sau khi cân bằng của tập dữ liệu (3) ICS-Flow.	80
Hình 34: Biểu đồ chính xác (Accuracy) tổng quát của các mô hình trên kịch bản 3.....	80
Hình 35: Biểu đồ thời gian huấn luyện và suy luận tổng quát của các mô hình trên kịch bản 3.....	81
Hình 36: Lịch sử huấn luyện (training history) trên kịch bản 3.....	83
Hình 37: Ma trận nhầm lẫn (confusion matrix) trên kịch bản 3.	85

MỤC LỤC BẢNG

Bảng 1:	Bảng so sánh các loại IDS: ưu điểm, thách thức, các công cụ phổ biến [14].....	10
Bảng 2:	Bảng đặc trưng tập dữ liệu CSE-CIC-IDS-2018.	27
Bảng 3:	Bảng đặc trưng tập dữ liệu CIC-IoT-2023	31
Bảng 4:	Bảng đặc trưng tập dữ liệu ICS-Flow.	34
Bảng 5:	Phân bố số lượng mẫu ban đầu của Tập dữ liệu (1):CSE-CIC-IDS-2018.	36
Bảng 6:	Phân bố số lượng mẫu ban đầu của Tập dữ liệu (2):CIC-IoT-2023.	36
Bảng 7:	Phân bố số lượng mẫu ban đầu của Tập dữ liệu (3):ICS-Flow.....	36
Bảng 8:	Bảng tóm tắt các hàm và ý nghĩa trong bước tiền xử lý.	55
Bảng 9:	Bảng tóm tắt các hàm và ý nghĩa trong bước huấn luyện và đánh giá mô hình.....	55
Bảng 10:	Tổng hợp đánh giá mức độ sử dụng tài nguyên của các mô hình.....	94

CHƯƠNG 1: GIỚI THIỆU

1.1. Giới thiệu đề tài

Ngày nay, Internet đã trở thành một phần không thể thiếu trong cuộc sống hàng ngày, được sử dụng rộng rãi trên đa dạng thiết bị từ máy tính để bàn, thiết bị cá nhân đến điện thoại thông minh. Internet ứng dụng sâu rộng trong những lĩnh vực then chốt như giáo dục, y tế, kinh tế và quốc phòng, cùng với sự phát triển nhanh chóng mạng 5G, mạng IoT ,điện toán đám mây,... các vấn đề về bảo mật và an ninh mạng ngày càng trở nên cấp bách và phức tạp hơn bao giờ hết. An ninh mạng không chỉ là vấn đề bảo vệ môi trường làm việc của cá nhân, tổ chức mà còn là yếu tố sống còn đối với nhiều doanh nghiệp và cơ quan nhà nước. Thiệt hại do mất mát hoặc đánh cắp thông tin có thể lên tới hàng triệu đô la. Theo báo cáo năm 2023 của IBM(International Business Machines) ghi nhận chi phí trung bình cho một vụ rò rỉ dữ liệu lên tới 4,24 triệu USD đối với doanh nghiệp toàn cầu, với nhiều trường hợp tại Mỹ, con số này lên tới 9 triệu USD mỗi vụ. Trong khi đó rò rỉ dữ liệu mật tại các cơ quan nhà nước có thể gây ra nguy cơ nghiêm trọng đối với an ninh quốc gia [1].

Mỗi ngày, các hệ thống mạng trên khắp các lĩnh vực đều phải đối mặt với nhiều hình thức tấn công đa dạng, nhắm vào các mục tiêu như dữ liệu cá nhân và tổ chức, tài khoản ngân hàng, phần mềm, người dùng và mạng nội bộ. Trước thực trạng này, các giải pháp bảo mật và hệ thống phòng chống mạng được phát triển và hoàn thiện không ngừng nhằm đối phó hiệu quả với phần mềm độc hại và các mối đe dọa mạng phức tạp. Yêu cầu đặt ra là các giải pháp này phải tiên tiến, linh hoạt và bền vững để bảo vệ tốt hơn trước sự đa dạng và gia tăng nhanh chóng của các nguy cơ an ninh trong môi trường số hóa hiện nay.

Các mối đe dọa an ninh mạng thông thường đều bắt nguồn từ hành vi xâm nhập trái phép vào hệ thống mạng. Xâm nhập mạng trái phép là mọi hành vi truy cập, khai thác hoặc can thiệp vào hệ thống mạng mà không có sự đồng ý hoặc ủy quyền hợp pháp từ chủ sở hữu hệ thống. Đây là dạng tấn công mạng có mục đích xấu như đánh cắp thông tin, phá hoại dữ liệu, tống tiền, gây gián đoạn dịch vụ... Những hành vi này thường sử dụng các phương thức như khai thác lỗ hổng bảo mật, chiếm quyền truy cập bất hợp pháp, phát tán phần mềm độc hại đều bị coi là vi phạm pháp luật.

Trong bối cảnh đó, Phát hiện xâm nhập (Intrusion Detection) là tuyến phòng thủ thứ hai sau tường lửa giữ vai trò then chốt trong việc bảo vệ hạ tầng số khỏi các hành vi

trái phép. Khái niệm phát hiện xâm nhập lần đầu tiên được James Anderson đề xuất vào năm 1980 [2]. Sau đó, một số học giả đã áp dụng các phương pháp học máy (Machine Learning) vào việc phát hiện xâm nhập. Tuy nhiên, do giới hạn về bộ nhớ và năng lực tính toán của máy tính vào thời điểm đó, học máy chưa nhận được nhiều sự chú ý. Với sự phát triển mạnh mẽ của công nghệ và sự ra đời của Trí tuệ nhân tạo (AI) cùng các công nghệ liên quan, ngày càng có nhiều nghiên cứu áp dụng học máy vào lĩnh vực an ninh mạng và đã đạt được những kết quả nhất định.

Phát hiện xâm nhập là quá trình ứng dụng các kỹ thuật phân tích tiên tiến để nhận diện sớm dấu hiệu bất thường hoặc hành vi đáng ngờ trên mạng, từ đó giúp tổ chức phát hiện kịp thời các cuộc tấn công và chủ động triển khai biện pháp ứng phó nhằm giảm thiểu thiệt hại và rủi ro cho hệ thống thông tin. Trong các môi trường mạng hiện đại, Hệ thống phát hiện xâm nhập (Intrusion Detection System – IDS) đóng vai trò quan trọng trong việc bảo vệ hạ tầng thông tin trước sự gia tăng về số lượng và tính phức tạp ngày càng cao của các cuộc tấn công mạng. Tuy nhiên, khi hệ thống mạng trở nên đa dạng hơn về kiến trúc, thiết bị và lưu lượng đặc biệt trong các môi trường mạng không đồng nhất như Mạng doanh nghiệp (Enterprise Network Environment), Mạng IoT (Internet of Things) và Mạng công nghiệp (Industrial Control Network),... Thì các mô hình phát hiện xâm nhập truyền thống như phương pháp phát hiện dựa trên chữ ký (signature-based), dựa trên luật đặc tả (specification-based),... dần bộc lộ những hạn chế. Mặc dù đơn giản và hiệu quả trong các môi trường tĩnh, các phương pháp này thường gặp khó khăn trong việc phát hiện các cuộc tấn công chưa từng biết đến (zero-day attacks). Điều này khiến chúng gặp những thách thức lớn trong việc đáp ứng các yêu cầu ngày càng phức tạp và đa dạng như hiện nay.

Hiện nay, nhiều nghiên cứu đã hướng đến việc ứng dụng học sâu (Deep learning) trong các hệ thống phát hiện xâm nhập nhằm nâng cao độ chính xác và khả năng thích nghi [3]. Điều này đã mang lại nhiều cải tiến đáng kể cho lĩnh vực an ninh mạng. Tuy nhiên, phần lớn các nghiên cứu vẫn tập trung đánh giá hiệu quả mô hình trên từng môi trường hoặc tập dữ liệu riêng lẻ, hoặc chỉ ghi nhận kết quả trong những trường hợp thử nghiệm có giới hạn. Trong khi đó, cách tiếp cận này hạn chế khả năng tổng quát hóa, khả năng thích nghi thực tế của các mô hình và hạn chế tính ứng dụng rộng rãi của các phương pháp khi triển khai trên nhiều loại môi trường mạng khác nhau.

Vì vậy đề tài "**Đánh giá các kỹ thuật học sâu trong phát hiện xâm nhập trên các môi trường mạng không đồng nhất**" được thực hiện, nhằm cung cấp cái nhìn chi tiết về hiệu quả của các kỹ thuật học sâu trong phát hiện xâm nhập trên các môi trường mạng thực tế, qua đó đóng góp vào việc phát triển các hệ thống an ninh mạng tiên tiến, linh hoạt hơn có thể bảo vệ các hệ thống trước các mối đe dọa mạng ngày càng tinh vi trong kỷ nguyên kết nối liên tục.

1.2. Mục tiêu đề tài

Đề tài "**Đánh giá các kỹ thuật học sâu trong phát hiện xâm nhập trên các môi trường mạng không đồng nhất**" sẽ tập trung vào nghiên cứu kỹ thuật học sâu và đánh giá chi tiết về hiệu năng của các kỹ thuật học sâu trong phát hiện xâm nhập khi áp dụng trên các môi trường mạng khác nhau, bao gồm Mạng doanh nghiệp (Enterprise Network Environment), Mạng IoT (Internet of Things) và Mạng công nghiệp (Industrial Control Network).

1.3. Những nghiên cứu liên quan

Trong những năm gần đây, các hệ thống phát hiện xâm nhập (IDS) đã không ngừng được cải tiến, đặc biệt với sự phát triển của các phương pháp dựa trên học máy, trong đó các kỹ thuật học sâu đã thu hút sự chú ý mạnh mẽ từ cộng đồng nghiên cứu.

Phương pháp này chủ yếu dựa vào hai bước cơ bản: (1) Thu thập một lượng lớn dữ liệu liên quan đến lưu lượng mạng (network traffic), trong tập dữ liệu này bao gồm các lưu lượng mạng bình thường (benign) và lưu lượng mạng bất thường (attack);(2) Sau khi có dữ liệu, quá trình huấn luyện mô hình được thực hiện, cho phép các thuật toán học sâu tự động phân loại lưu lượng mạng.Việc áp dụng thành công các kỹ thuật học sâu trong hệ thống phát hiện xâm nhập (IDS) phụ thuộc đáng kể vào chất lượng và đặc điểm của tập dữ liệu huấn luyện. Các nghiên cứu đã tập trung khai thác các tập dữ liệu công khai phổ biến trong lĩnh vực phát hiện xâm nhập mạng. Các tập dữ liệu được sử dụng rộng rãi nhất trong nhiều nghiên cứu như KDD Cup 1999 (KDD99), NSL-KDD, UNSW-NB15 [4]. Tiêu biểu nghiên cứu của A. M. Amine [5] về NIDS cho thấy rằng việc ứng dụng các kỹ thuật học máy có thể nâng cao khả năng phát hiện và giảm thiểu tỷ lệ cảnh báo giả.Tập dữ liệu thử nghiệm được sử dụng là KDD Cup 1999, một chuẩn đánh giá phổ biến cho các hệ thống phát hiện xâm nhập mạng. Trong nghiên cứu, các

tác giả đã xây dựng mô hình phát hiện xâm nhập sử dụng mạng nơ-ron tích chập (Convolutional Neural Network – CNN). Kết quả thực nghiệm cho thấy mô hình đạt độ chính xác 99,90%, recall 99,89%, và specificity 100%, cho thấy tiềm năng lớn của CNN trong bài toán phát hiện các mối đe dọa mạng. Quan et al. [6] này nói việc phân tích hiệu suất của các mô hình học sâu trong phát hiện xâm nhập mạng dựa trên dữ liệu UNSW-NB15. Các mô hình được triển khai bao gồm MLP, RNN, CNN, BiLSTM, LSTM, GRU và Transformer. Kết quả cho thấy GRU đạt độ chính xác cao nhất (98,78%) với thời gian huấn luyện ngắn nhất. Nghiên cứu của Mohammed [7] đã tập trung cải thiện hiệu suất của hệ thống IDS bằng cách tích hợp các kỹ thuật học sâu với dữ liệu chuỗi thời gian. Nghiên cứu này đánh giá hiệu năng của các kỹ thuật học sâu như RNN (Recurrent Neural Network), CNN (Convolutional Neural Network) và LSTM (Long Short-Term Memory). Kết quả thực nghiệm cho thấy các mô hình kết hợp (Hybrid models) cho kết quả tiềm năng hơn các mô hình riêng lẻ, đặc biệt là mô hình kết hợp CNN+RNN+LSTM, đạt hiệu suất tốt với điểm F1 là 86%, chỉ số precision là 92% và chỉ số recall là 79%.

Trong khi đó, Hệ thống phát hiện xâm nhập (IDS) cho IoT cũng trở thành một chủ đề hấp dẫn, do tính chất đa dạng và phức tạp của các thiết bị IoT. Các nghiên cứu chỉ ra rằng các thiết bị này thường có khả năng bảo mật yếu, dẫn đến nhu cầu cấp bách về các giải pháp IDS chuyên biệt. Asgharzadeh et.al [8] đã phát triển mô hình kết hợp CNN-BMEGTO-KNN, thử nghiệm trên tập ToN-IoT và đạt độ chính xác lên đến 99.99%, cho thấy tiềm năng mạnh mẽ của các kỹ thuật học sâu với dữ liệu đa dạng từ IoT trong việc phát hiện các kiểu tấn công phức tạp, mới xuất hiện. Ngoài ra, các nghiên cứu đề xuất các khung giải pháp nâng cao như tiền xử lý dữ liệu lớn, lựa chọn đặc trưng, và mô hình học bán giám sát để thích ứng với dữ liệu IoT thực tế ngày càng đa dạng và ngẫu nhiên. Sagu et.al và nhóm tác giả [9] đề xuất một giải pháp kết hợp giữa mạng nơ-ron tích chập (CNN) và mô hình học sâu GRU (Gated Recurrent Unit) nhằm phát hiện hiệu quả các mối đe dọa mạng IoT. CNN được dùng để trích xuất đặc trưng theo không gian từ dữ liệu mạng, còn GRU đảm nhiệm phát hiện mối liên kết theo thời gian. Để tăng hiệu năng mô hình, nhóm phát triển thuật toán Self-Upgraded Cat and Mouse Optimization (SUCMO) để tối ưu siêu tham số tự động, nâng cao độ chính xác phân loại. Kết quả cho thấy đạt hiệu quả cao trong phát hiện nhiều loại tấn công như DoS, Botnet trong môi trường IoT đa dạng và thực tiễn hơn. Kết quả này khẳng định tiềm

năng ứng dụng của các mô hình học sâu lai, đặc biệt khi được kết hợp với thuật toán tối ưu hiện đại, cho việc xây dựng hệ thống an ninh mạng thông minh cho IoT. Alotaibi và cộng sự [10] đã nghiên cứu các mô hình học sâu như CNN và RNN được huấn luyện và đánh giá trên dữ liệu NF-BoT-IoT, hướng tới nhận diện hành vi botnet trong mạng IoT. Hệ thống đề xuất đạt độ chính xác trên 96%, đồng thời giảm đáng kể tỷ lệ báo động giả, thể hiện khả năng thích ứng cao và hiệu quả khi triển khai trên các mạng thiết bị kết nối đa dạng. Bakhsh et al [11], các tác giả đã xây dựng framework IDS dựa trên ba mô hình học sâu nhằm phát hiện và phân loại xâm nhập trong môi trường IoT. Các mô hình được huấn luyện trên tập dữ liệu CIC IoT 2022, thử nghiệm ở cả chế độ phân loại nhị phân và đa lớp, đạt hiệu quả vượt trội về khả năng phát hiện các loại tấn công như Blackhole, DDoS, Sinkhole,... phương pháp đề xuất cho thấy hiệu suất vượt trội khi thử nghiệm trên tập dữ liệu CIC-IoT22. Kết quả đạt được độ chính xác là 99.93% với mô hình FFNN, 99.85% với mô hình LSTM, và 96.42% với mô hình Rand-NN . Hệ thống phát hiện xâm nhập (IDS) sau huấn luyện được tích hợp vào hệ sinh thái IoT bao gồm thiết bị, cảm biến, ứng dụng, máy chủ và nền tảng đám mây, hệ thống sẽ thực hiện giám sát lưu lượng mạng theo thời gian thực, giúp nhanh chóng nhận diện, phân loại xâm nhập và phát cảnh báo, giúp tổ chức kịp thời triển khai các biện pháp ứng phó. Nhờ đó, độ tin cậy và hiệu quả bảo vệ của toàn bộ hệ thống IoT hiện đại được nâng cao rõ rệt.

Đối với hệ thống phát hiện xâm nhập (IDS) trong các Hệ thống điều khiển công nghiệp (Industrial Control System - ICS), lĩnh vực nghiên cứu đã chứng kiến sự phát triển mạnh mẽ và đa dạng trong những năm gần đây, phản ánh tầm quan trọng ngày càng tăng của việc bảo vệ cơ sở hạ tầng công nghiệp quan trọng như nhà máy điện, hệ thống cấp thoát nước, mạng lưới giao thông, và các nhà máy sản xuất,... khỏi các mối đe dọa mạng hiện đại. Nghiên cứu của Le et al [12], họ xem xét rộng rãi sự phát triển của hệ thống điều khiển công nghiệp (ICS), tập trung cụ thể vào bộ điều khiển logic lập trình (Programmable Logic Controllers - PLC) trong cơ sở hạ tầng quan trọng, đặc biệt là các trạm tròn và cơ sở xử lý nhiệt. Nghiên cứu làm rõ những rủi ro về an ninh mạng phát sinh từ sự hội tụ giữa bộ điều khiển logic lập trình (PLC) và công nghệ thông tin, khi các hệ thống chuyển đổi từ dạng độc lập sang tích hợp cùng công nghệ điện tử-dám mây. Những đóng góp nổi bật từ cả ngành công nghiệp và giới học thuật đã nhấn mạnh vai trò then chốt của các kỹ thuật máy học và học sâu trong việc gia tăng độ an toàn cho các hệ thống dựa trên bộ điều khiển logic lập trình (PLC). Bài viết tối ưu hóa

năm kỹ thuật học máy cổ điển và ba kỹ thuật học sâu, đạt độ chính xác ấn tượng trên 97%. Đặc biệt, mô hình kết hợp được đề xuất đã đạt được độ chính xác trên 99% khi thử nghiệm trên các tập dữ liệu thực tế Hardware-In-the-Loop thuộc Augmented ICS (HAI). Nghiên cứu của Bozdal và các cộng sự [13] tập trung vào vấn đề mất cân bằng lớp trong học máy. Họ nghiên cứu khái niệm chồng chéo lớp (class overlapping) và ảnh hưởng của nó đến độ chính xác và khả năng phân loại, đặc biệt trong bối cảnh hệ thống điều khiển công nghiệp (ICS). Hiện tượng chồng chéo lớp xảy ra khi các điểm dữ liệu ngoại lai (outlier) làm dịch chuyển ranh giới phân lớp về phía lớp chiếm đa số, từ đó khiến các mô hình học máy trở nên thiên lệch hơn và dễ mắc lỗi phân loại âm sai (false negative). Mức độ chịu đựng hiện tượng chồng chéo lớp của các mô hình học máy là khác nhau, trong đó SVM (Support vector machine) thể hiện kém linh hoạt hơn đáng kể so với KNN (K-nearest neighbors). Ngoài ra, các nhà nghiên cứu còn thực hiện đánh giá so sánh các vấn đề bảo mật trên tập dữ liệu xử lý nước an toàn SwaT (Secure Water Treatment). Trong nghiên cứu, họ đề xuất mô hình ID-CNN kết hợp với GRU (Gated Recurrent Unit) và đạt được kết quả ấn tượng với F1-score lên đến 98,69%.

Tổng hợp các nghiên cứu về phát hiện xâm nhập trong các môi trường và hệ thống mạng khác nhau cho thấy tiềm năng lớn của các kỹ thuật học sâu trong việc nâng cao hiệu quả nhận dạng và có thể ngăn chặn các cuộc tấn công mạng. Trong bối cảnh môi trường mạng ngày càng đa dạng, phức tạp, việc đánh giá chi tiết hiệu năng của các kỹ thuật học sâu là điều cần thiết để đảm bảo khả năng thích ứng, đáp ứng và vận hành ổn định trong thực tiễn. Chỉ khi các giải pháp phát hiện xâm nhập sử dụng học sâu đáp ứng đầy đủ các yêu cầu thực tiễn, chúng mới có thể phát huy tối đa hiệu quả khi chuyển giao từ môi trường phòng thí nghiệm ra ứng dụng thực tế, qua đó nâng cao đáng kể năng lực phòng thủ mạng trong kỷ nguyên kết nối liên tục và không ngừng phát triển.

1.4. Đối tượng và phạm vi nghiên cứu.

- **Đối tượng nghiên cứu:** Các kỹ thuật học sâu bao gồm CNN (Convolutional Neural Network), MLP (Multi-Layer Perceptron), LSTM (Long Short-Term Memory), GRU (Gated Recurrent Unit) và kỹ thuật học sâu kết hợp (Hybird Models).
- **Phạm vi thực nghiệm:** Các tập dữ liệu chuẩn đại diện cho từng môi trường bao gồm CSE-CIC-IDS-2018 (Mạng doanh nghiệp); CIC-IoT-2023 (Mạng IoT); ICS-Flow (Mạng công nghiệp).

1.5. Phương pháp nghiên cứu

- Luận văn thực hiện tham khảo các tài liệu liên quan đến nghiên cứu về phát hiện xâm nhập (Instruction Detection), Hệ thống phát hiện xâm nhập (IDS), Các kỹ thuật học sâu (Deep learning).
- Tiền xử lý, chuẩn bị và trích xuất đặc trưng dữ liệu từ các tập dữ liệu.
- Xây dựng, huấn luyện và đánh giá các kỹ thuật học sâu trên từng tập dữ liệu.
- Tổng hợp, đánh giá kết quả thực nghiệm.

1.6. Bộ cục luận văn.

Nội dung chính của luận văn được chia thành 5 chương, cụ thể như sau:

Chương 1: Giới Thiệu

Chương 2: Cơ Sở Lý Thuyết

Chương 3: Phương Pháp Thực Hiện

Chương 4: Kết Quả Thực Nghiệm

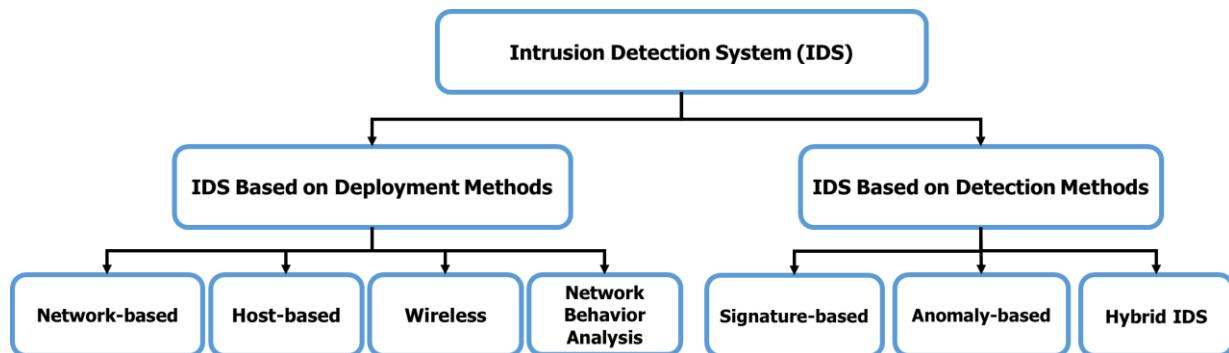
Chương 5: Kết Luận

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Tổng quan về hệ thống phát hiện xâm nhập

Hệ thống phát hiện xâm nhập (Intrusion Detection System - IDS) là tuyến phòng thủ thứ hai chỉ sau tường lửa, dùng để giám sát lưu lượng mạng hoặc hoạt động của hệ thống máy tính nhằm phát hiện các hành vi đáng ngờ, truy cập trái phép hoặc các cuộc tấn công độc hại. IDS không trực tiếp ngăn chặn tấn công mà tập trung vào việc phát hiện, ghi lại (log) các sự kiện và gửi cảnh báo cho quản trị viên để họ có thể đưa ra biện pháp xử lý kịp thời. Với khả năng giám sát liên tục và phân tích dữ liệu thời gian thực, IDS giúp nâng cao khả năng phòng thủ mạng, đảm bảo an toàn cho hệ thống trước các mối đe dọa tiềm ẩn.

Hệ thống phát hiện xâm nhập (IDS) có thể được phân loại dựa trên Phương pháp triển khai (IDS Based on Deployment Methods) hoặc Phương pháp phát hiện (IDS Based on Detection Methods).



Hình 1: Sơ đồ hình phân loại Hệ thống phát hiện xâm nhập (IDS)

2.1.1. Hệ thống phát hiện xâm nhập dựa trên phương pháp triển khai (IDS Based on Deployment Methods)

Là phân loại dựa trên vị trí và phạm vi giám sát của IDS trong hệ thống.

- **Network-based IDS (NIDS):** Giám sát toàn bộ lưu lượng mạng tại một điểm chiến lược. Nó phân tích các gói tin (packets) ra vào mạng để tìm kiếm dấu hiệu tấn công.
- **Host-based IDS (HIDS):** Được cài đặt trên từng máy tính (host) riêng lẻ. Nó giám sát các hoạt động bên trong máy đó như thay đổi tệp tin hệ thống, log hệ thống, và các tiến trình đang chạy.
- **Wireless IDS (WIDS):** Chuyên giám sát mạng không dây (Wifi) để phát hiện

các mối đe dọa đặc thù như các điểm truy cập giả mạo, nghe lén, hoặc tấn công từ chối dịch vụ trên mạng không dây.

- **Network Behavior Analysis (NBA):** Phân tích hành vi chung của lưu lượng mạng. Hệ thống này sẽ xây dựng một mô hình "bình thường" (baseline) của mạng và cảnh báo khi có những thay đổi đột ngột hoặc bất thường so với mô hình đó.

2.1.2. Hệ thống phát hiện xâm nhập dựa trên phương pháp phát hiện (IDS Based on Detection Methods)

Là phân loại dựa trên kỹ thuật mà IDS sử dụng để xác định một cuộc tấn công.

- **Dựa trên dấu hiệu (Signature-based):** Hoạt động giống như phần mềm diệt virus. Nó có một cơ sở dữ liệu chứa "dấu hiệu" (signatures) của các loại mã độc và các kiểu tấn công đã biết. IDS sẽ so sánh các hoạt động đang giám sát với cơ sở dữ liệu này để phát hiện tấn công.
- **Dựa trên sự bất thường (Anomaly-based):** Phương pháp này xây dựng một mô hình về trạng thái "bình thường" của hệ thống hoặc mạng. Bất kỳ hoạt động nào lệch khỏi mô hình này sẽ bị coi là đáng ngờ và bị cảnh báo.
- **Hệ thống IDS lai (Hybrid IDS):** Hệ thống kết hợp cả hai phương pháp dựa trên dấu hiệu và dựa trên bất thường trên để tận dụng ưu điểm của cả hai loại, giúp tăng độ chính xác và khả năng bao phủ, giảm thiểu cảnh báo sai. Tuy nhiên có độ phức tạp cao, chi phí tính toán lớn.

Bảng so sánh ưu điểm, thách thức, các công cụ phổ biến của các loại hệ thống phát hiện xâm nhập được trình bày ở **Bảng 1**.

Loại IDS	Ưu điểm	Thách thức	Công cụ phổ biến
Host-Based IDS (HIDS)	<ul style="list-style-type: none"> - Phát hiện tốt các tấn công nội bộ - Giám sát thay đổi hệ thống tập tin - Có thể cung cấp bằng chứng pháp lý 	<ul style="list-style-type: none"> - Chỉ bảo vệ được máy chủ cài đặt - Khó mở rộng quy mô - Tiêu tốn tài nguyên hệ thống 	<ul style="list-style-type: none"> - OSSEC - Tripwire - Samhain
Network-Based IDS (NIDS)	<ul style="list-style-type: none"> - Phát hiện tấn công trên toàn bộ mạng - Giám sát lưu lượng trong thời gian thực 	<ul style="list-style-type: none"> - Khó phát hiện các tấn công được mã hóa - Có thể bị tấn công từ chính luồng mạng 	<ul style="list-style-type: none"> - Snort - Suricata - Bro/Zeek
Wireless IDS (WIDS)	<ul style="list-style-type: none"> - Bảo vệ mạng không dây - Phát hiện các điểm truy cập giả mạo (rogue AP), tấn công từ chối dịch vụ không dây 	<ul style="list-style-type: none"> - Bị giới hạn bởi khoảng cách tín hiệu - Khó phát hiện thiết bị ẩn danh - Dễ bị nhiễu 	<ul style="list-style-type: none"> - Kismet - WIDS tính năng trong Aruba/Cisco
Network Behavior Analysis (NBA)	<ul style="list-style-type: none"> - Phát hiện các hành vi bất thường chưa biết - Tốt trong phát hiện DDoS, botnet 	<ul style="list-style-type: none"> - Có thể tạo báo động giả - Phụ thuộc vào baseline hành vi 	<ul style="list-style-type: none"> - Darktrace - Flowmon
Signature-Based IDS	<ul style="list-style-type: none"> - Độ chính xác cao với tấn công đã biết - Ít báo động giả 	<ul style="list-style-type: none"> - Không phát hiện được các tấn công mới (zero-day) - Cần cập nhật chữ ký thường xuyên 	<ul style="list-style-type: none"> - Snort - Suricata
Anomaly-Based IDS	<ul style="list-style-type: none"> - Phát hiện được tấn công chưa biết - Có khả năng thích ứng 	<ul style="list-style-type: none"> - Tỷ lệ báo động giả cao - Cần dữ liệu huấn luyện chất lượng 	<ul style="list-style-type: none"> - Bro/Zeek - Splunk - Mô hình học máy
Hybrid IDS	<ul style="list-style-type: none"> - Kết hợp ưu điểm của nhiều phương pháp - Tăng độ chính xác và khả năng bao phủ 	<ul style="list-style-type: none"> - Độ phức tạp cao - Chi phí tính toán lớn 	<ul style="list-style-type: none"> - OSSEC + Snort - AI kết hợp signature và anomaly

Bảng 1: Bảng so sánh các loại IDS: ưu điểm, thách thức, các công cụ phổ biến [14].

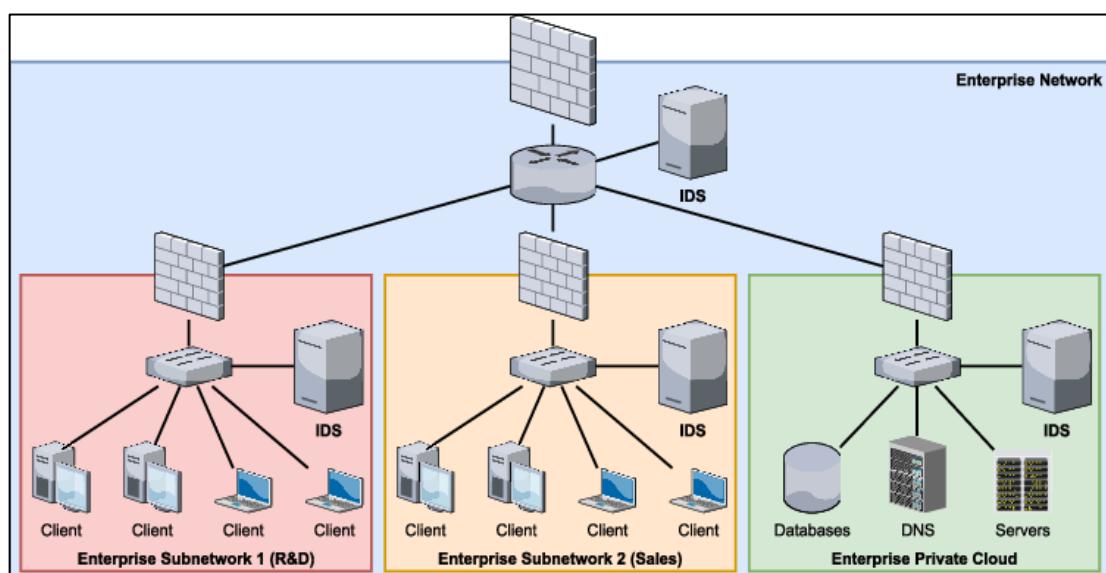
2.2. Các môi trường ứng dụng hệ thống phát hiện xâm nhập (IDS)

Các môi trường ứng dụng hệ thống phát hiện xâm nhập (IDS) phổ biến nhất là Mạng doanh nghiệp, Mạng IoT, Mạng công nghiệp.

2.2.1. Mạng doanh nghiệp (Enterprise network)

Mạng doanh nghiệp là một mạng máy tính được thiết kế để kết nối các máy tính, máy chủ, trung tâm dữ liệu và các thiết bị hệ thống của một tổ chức hay một công ty, hỗ trợ các hoạt động kinh doanh và trao đổi thông tin.

Việc bảo vệ mạng doanh nghiệp cũng là một trong những thách thức lớn hiện nay. Các mạng này thường có quy mô lớn và kiến trúc đa dạng, khiến công tác quản lý và giám sát trở nên phức tạp. Khối lượng dữ liệu truyền tải lớn và liên tục đặt ra yêu cầu cho các giải pháp bảo mật phải xử lý được dữ liệu trong thời gian thực mà vẫn duy trì hiệu suất hệ thống ổn định. Mạng doanh nghiệp còn phải đối mặt với nhiều mối đe dọa như tấn công từ chối dịch vụ (DDoS), lây nhiễm mã độc (malware), lừa đảo (phishing), truy cập trái phép, trong đó truy cập trái phép từ bên trong thuộc loại tấn công từ nội bộ (insider threats) là một vấn đề phức tạp và khó kiểm soát, do bắt nguồn từ những người đã có quyền truy cập hợp pháp. Do đó, việc đảm bảo an ninh mạng cho doanh nghiệp ngày càng trở thành ưu tiên hàng đầu trong chiến lược quản lý hạ tầng của các tổ chức. Để ứng phó, các tổ chức thường triển khai hệ thống phát hiện xâm nhập (IDS), dưới dạng IDS dựa trên mạng (NIDS) để giám sát toàn bộ lưu lượng, hoặc IDS dựa trên máy chủ (HIDS) để bảo vệ các máy chủ riêng lẻ.

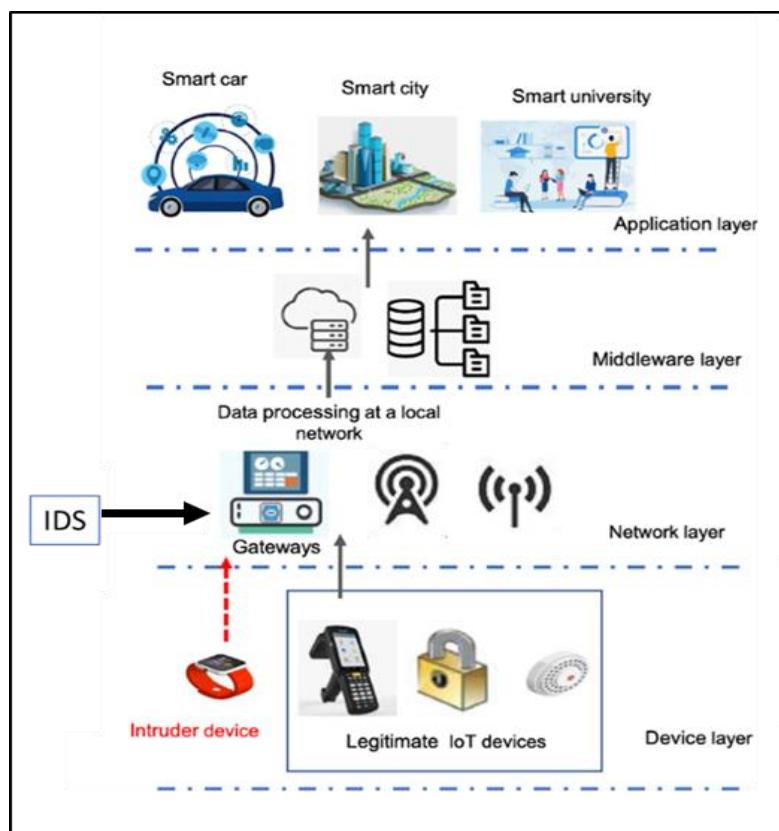


Hình 2: Mô tả mạng doanh nghiệp (Enterprise network) [15]

2.2.2. Mạng IoT (Internet of things network)

Mạng IoT (Internet of Things) là hệ thống kết nối các thiết bị thông minh (cảm biến, camera, thiết bị gia dụng...) thông qua internet để thu thập, trao đổi và xử lý dữ liệu tự động. Sự phát triển mạnh mẽ của IoT mang lại nhiều tiện ích nhưng đồng thời cũng tạo ra một môi trường phức tạp và tiềm ẩn nhiều rủi ro về an ninh mạng.

Với sự bùng nổ về số lượng thiết bị và những lỗ hổng bảo mật tiềm ẩn, mạng IoT nhanh chóng trở thành mục tiêu hấp dẫn cho các cuộc tấn công mạng. Thách thức lớn nhất trong việc bảo vệ IoT xuất phát từ đặc điểm của chính các thiết bị: chúng thường có tài nguyên hạn chế (CPU, bộ nhớ), sử dụng nhiều giao thức truyền thông đa dạng như MQTT, CoAP, Zigbee, Bluetooth... và ít khi được cập nhật bảo mật định kỳ. Điều này khiến cho việc triển khai các giải pháp bảo vệ trở nên khó khăn và tốn nhiều nỗ lực. Các mối đe dọa phổ biến đối với IoT bao gồm lỗ hổng phần mềm, tấn công từ chối dịch vụ (DDoS), tấn công vật lý vào thiết bị và khai thác dữ liệu riêng tư. Một giải pháp khả thi là triển khai hệ thống phát hiện xâm nhập (IDS) tại các điểm kết nối giữa mạng IoT và mạng chính, hoặc tại cổng kết nối internet. Cách tiếp cận này cho phép giám sát lưu lượng từ nhiều thiết bị mà không cần cài đặt phần mềm trực tiếp trên từng thiết bị, góp phần nâng cao hiệu quả bảo mật tổng thể.

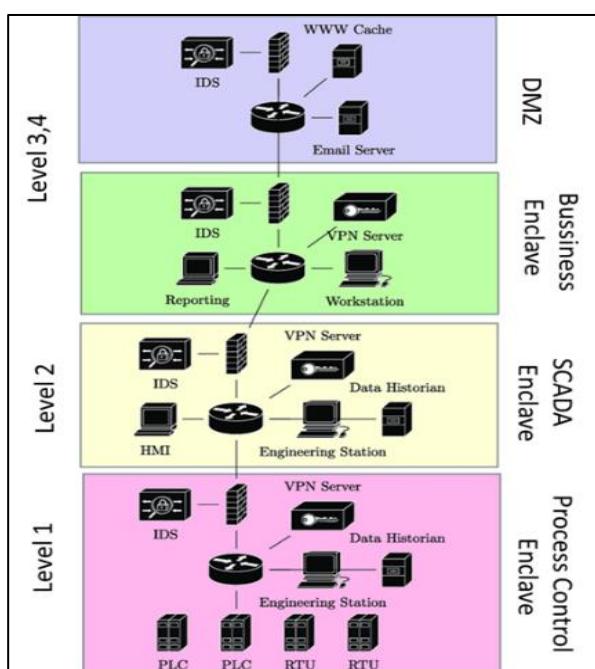


Hình 3: Mô tả mạng IoT (Internet of Things Network) [16]

2.2.3. Mạng công nghiệp (Industrial Network)

Mạng công nghiệp (Industrial Network) là nền tảng kết nối trong các hệ thống điều khiển công nghiệp (Industrial Control System – ICS), cho phép trao đổi dữ liệu và điều khiển giữa các thiết bị. ICS bao gồm các hệ thống như SCADA(Supervisory Control and Data Acquisition), DCS(Distributed Control System), cùng các thiết bị tự động hóa như bộ điều khiển logic lập trình (PLC), cảm biến và máy móc, được thiết kế để giám sát và điều khiển các quy trình sản xuất trong thời gian thực.

Môi trường này cũng tiềm ẩn nhiều mối đe dọa nghiêm trọng. Các cuộc tấn công có thể trực tiếp nhắm vào thiết bị điều khiển gây gián đoạn hoạt động, khai thác lỗ hổng từ những giao thức truyền thông chưa được mã hóa như Modbus ,DNP3 hoặc triển khai tấn công ransomware (mã độc tống tiền). Khác với các mạng thông thường, bảo mật trong ICS không chỉ tập trung vào dữ liệu, mà còn gắn liền với sự an toàn của máy móc, quy trình vật lý và tính mạng cả con người.Việc bảo vệ mạng công nghiệp càng trở nên thách thức do đặc thù của ICS là có nhiều thiết bị cũ khó cập nhật bảo mật, yêu cầu vận hành liên tục 24/7, cùng sự hạn chế về khả năng xử lý của các thiết bị điều khiển. Bất kỳ gián đoạn nào cũng có thể dẫn đến thiệt hại lớn, từ ngừng sản xuất, hư hỏng dây chuyền cho tới rủi ro tai nạn nghiêm trọng. Để ứng phó với các mối đe dọa ngày càng gia tăng, việc triển khai hệ thống phát hiện xâm nhập (IDS) có thể giám sát lưu lượng mạng công nghiệp, phát hiện bất thường và cảnh báo kịp thời trước các mối đe dọa tiềm ẩn, từ đó có thể bảo vệ doanh nghiệp khỏi tổn thất tài chính cũng như rủi ro an toàn.



Hình 4: Mô tả mạng ICS [17]

2.3. Deep Learning

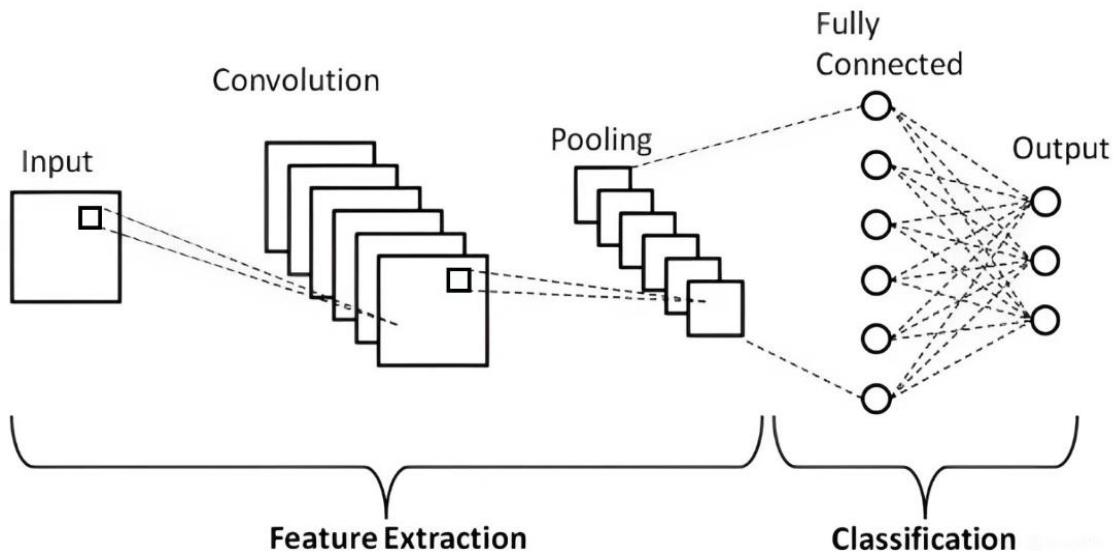
Lý thuyết học sâu (Deep learning) kể từ khi được Hinton và cộng sự đề xuất [18], là một nhánh quan trọng của học máy (Machine Learning) và cũng thuộc lĩnh vực trí tuệ nhân tạo (AI), học sâu đã thể hiện hiệu suất vượt trội trong các lĩnh vực như thị giác máy tính (Computer Vision - CV) và xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP). Khác với các phương pháp Machine Learning (Học máy) truyền thống, Deep Learning sử dụng mạng nơ-ron nhân tạo nhiều tầng để tự động học ra các đặc trưng từ dữ liệu đầu vào từ đơn giản đến trừu tượng mà không cần trích xuất thủ công. Nhờ khả năng xử lý dữ liệu phi cấu trúc cho phép máy tính xử lý những bài toán phức tạp như nhận diện hình ảnh, xử lý ngôn ngữ tự nhiên, hoặc dự đoán chính xác với dữ liệu lớn và phi cấu trúc [19]. Deep Learning đã trở thành công cụ cốt lõi trong nhiều ứng dụng hiện đại như: Nhận dạng khuôn mặt và vật thể trong ảnh, chatbot, phân tích cảm xúc, phát hiện gian lận, tấn công mạng, xe tự lái, robot thông minh, trợ lý ảo, ứng dụng trong y tế để phát hiện bệnh, hỗ trợ phát triển thuốc mới và phân tích dữ liệu y tế lớn.

Các nhà nghiên cứu đã khám phá ứng dụng các kỹ thuật trí tuệ nhân tạo (AI) như học sâu học sâu (Deep Learning) và máy học (Machine Learning) để giải quyết mối quan tâm về bảo mật của các hệ thống phát hiện xâm nhập (IDS) [20]. Chúng học thông tin hữu ích từ dữ liệu quy mô lớn và trở nên phổ biến trong thập kỷ qua do những tiến bộ trong bộ xử lý như bộ xử lý đồ họa mạnh mẽ (GPU). Nhờ khả năng học đặc trưng mạnh mẽ và tự động, học sâu đã được xem là một lựa chọn đầy hứa hẹn trong việc phát triển các hệ thống phát hiện xâm nhập (IDS) hiện đại [21].

Với tiềm năng đó, có rất nhiều bài nghiên cứu ứng dụng máy học, học sâu trong các hệ thống phát hiện xâm nhập (IDS). Theo kết quả nghiên cứu [14], khi phân tích các mô hình học sâu được áp dụng trong các hệ thống phát hiện xâm nhập, Convolutional Neural Network (CNN) được ghi nhận là thuật toán được sử dụng phổ biến nhất, chiếm khoảng 21% tổng số công trình được khảo sát. Đứng thứ hai là Long Short-Term Memory (LSTM) với tỷ lệ sử dụng khoảng 19%. Ngoài ra, các phương pháp khác như Gated Recurrent Unit (GRU), Multi-Layer Perceptron (MLP), cùng với các mô hình kết hợp (Hybrid models) cũng được triển khai trong nhiều nghiên cứu, phản ánh xu hướng khai thác đa dạng và hiệu quả của kiến trúc học sâu. Do đó, luận văn lựa chọn các thuật toán nền tảng (baseline) nêu trên để tiến hành thực nghiệm.

2.3.1. CNN (Convolutional Neural Network)

CNN (Convolutional Neural Network) là một trong những kiến trúc mạng nơ-ron sâu phổ biến và hiệu quả, đặc biệt trong các bài toán xử lý dữ liệu có cấu trúc không gian như hình ảnh, chuỗi thời gian, hoặc văn bản. Thuật ngữ "Mạng nơ-ron" dùng để chỉ một lĩnh vực con của trí tuệ nhân tạo. Nhờ mạng nơ-ron nhân tạo, nhiều vấn đề được giải quyết bởi máy tính mà không cần quá nhiều sự trợ giúp từ con người. Tên và cấu trúc được lấy cảm hứng từ bộ não con người, bắt chước cách tế bào thần kinh sinh học truyền tín hiệu cho nhau. Thuật toán này được thiết kế để tự động học và trích xuất các đặc trưng phân cấp từ dữ liệu đầu vào thông qua các lớp tích chập (convolutional layers), giúp giảm thiểu số lượng tham số học và tăng khả năng khai quát hóa của mô hình. CNN hoạt động dựa trên nguyên lý sử dụng các bộ lọc để quét qua dữ liệu đầu vào, từ đó phát hiện các đặc trưng cục bộ như cạnh, góc, hoặc họa tiết đặc trưng trong hình ảnh, hoặc các mẫu ngữ nghĩa trong chuỗi văn bản. Các đặc trưng này sau đó được kết hợp thông qua các lớp kích hoạt, lớp chuẩn hóa, và lớp gộp để tạo ra biểu diễn trừu tượng hơn ở các tầng sâu hơn của mạng [22]. **Hình 5** dưới đây thể hiện kiến trúc của một mạng nơ-ron tích chập CNN.



Hình 5: Kiến trúc mạng CNN (Convolutional Neural Network) [23].

Kiến trúc của Mạng Nơ-ron Tích chập (CNN)

1. Lớp Tích Chập (Convolutional Layer)

Lớp tích chập đóng vai trò là nền tảng cốt lõi của CNN, nơi các trọng số (kernel hoặc filter) được quét tuần tự trên không gian dữ liệu đầu vào, thực hiện phép tích chập để tạo ra các bản đồ đặc trưng (feature maps). Cơ chế này giúp mô hình tự động phát hiện và học các đặc trưng không gian quan trọng khác.

2. Lớp Kích Hoạt (Activation Layer)

Sau mỗi phép tích chập, một hàm kích hoạt phi tuyến tính, điển hình là ReLU (Rectified Linear Unit), được áp dụng nhằm đưa đầu ra về dạng phi tuyến. Tác dụng của lớp này là giúp mạng học được các đặc trưng phi tuyến phức tạp, tăng cường khả năng nhận diện các đặc điểm đặc thù hoặc bất thường trong dữ liệu, như các mẫu tấn công đột biến trong an ninh mạng. Đặc biệt, ReLU còn giúp giải quyết hiệu quả vấn đề "vanishing gradient" (độ dốc biến mất), từ đó đẩy nhanh quá trình huấn luyện và cải thiện hiệu năng của các mô hình sâu.

3. Lớp Pooling (Pooling Layer)

Lớp gộp là thành phần quan trọng nhằm giảm chiều dữ liệu cũng như số lượng tham số, nhờ đó tiết kiệm tài nguyên tính toán và hạn chế hiện tượng quá khớp. Hai kỹ thuật phổ biến nhất là max pooling (chọn giá trị lớn nhất trong vùng local patch) và average pooling (tính giá trị trung bình), qua đó giữ lại các đặc trưng quan trọng nhất trong bản đồ đặc trưng đồng thời giảm nhiễu.

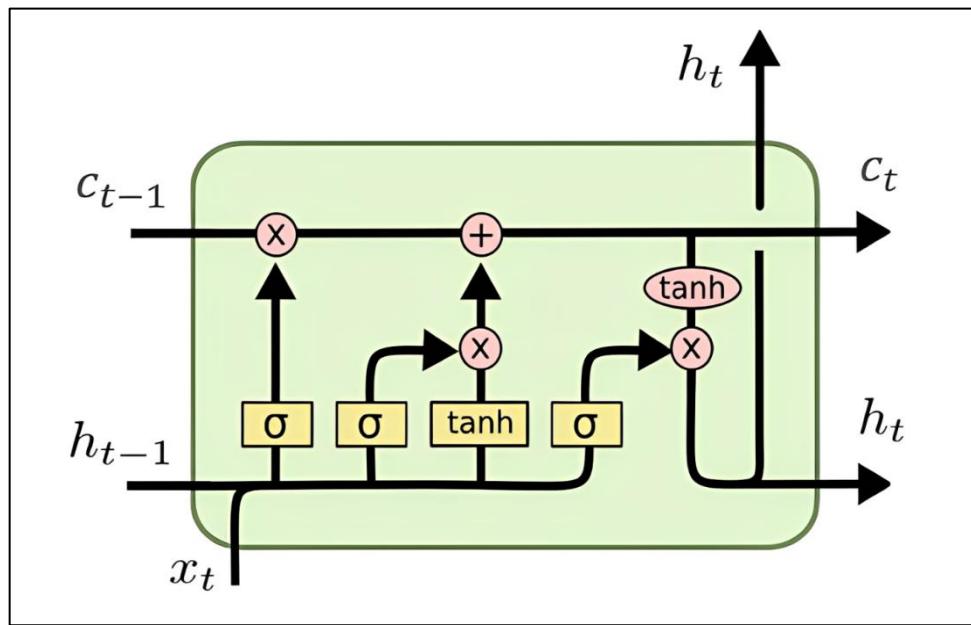
4. Lớp Kết nối Toàn phần (Fully Connected Layer)

Sau khi các đặc trưng đã được trích xuất và giảm chiều, chúng được chuyển sang các lớp kết nối toàn phần (FC) – mô phỏng mạng nơ-ron truyền thống. Tại đây, các đặc trưng được kết hợp lại để thực hiện các nhiệm vụ cao hơn, như phân loại hoặc dự đoán. Đầu ra cuối cùng thường được xử lý qua hàm softmax, cho phép mô hình tính toán xác suất thuộc về từng lớp, từ đó xác định nhãn lớp với xác suất cao nhất. Cơ chế này đặc biệt hữu ích trong việc phân loại chính xác các đối tượng hoặc các dạng tấn công đa dạng trong các ứng dụng thực tế.

Nhờ có sự phối hợp linh hoạt giữa các lớp, CNN có thể tự động học và tối ưu hóa các đặc trưng đa cấp độ, mang lại hiệu quả vượt trội trong các bài toán nhận diện, phân loại và phát hiện bất thường.

2.3.2. LSTM (Long Short-Term Memory)

Long Short-Term Memory (Bộ nhớ ngắn-dài hạn) là một kiến trúc mạng nơ-ron hồi quy RNN (Recurrent Neural Network) tiên tiến, được thiết kế nhằm khắc phục hiện tượng gradient biến mất trong quá trình huấn luyện và giúp mô hình có khả năng học các phụ thuộc dài hạn trong dữ liệu chuỗi. LSTM đã chứng minh hiệu quả vượt trội trong nhiều lĩnh vực như xử lý ngôn ngữ tự nhiên, phân tích chuỗi thời gian, nhận dạng giọng nói, và các hệ thống dự báo phức tạp.



Hình 6: Kiến trúc mạng LSTM (Long Short-Term Memory) [24]

Kiến trúc của Bộ nhớ ngắn dài hạn (LSTM)

Kiến trúc của LSTM được xây dựng dựa trên một cơ chế bộ nhớ tinh vi cho phép mô hình lưu trữ, cập nhật và truy xuất thông tin qua nhiều bước thời gian. Các thành phần chính bao gồm:

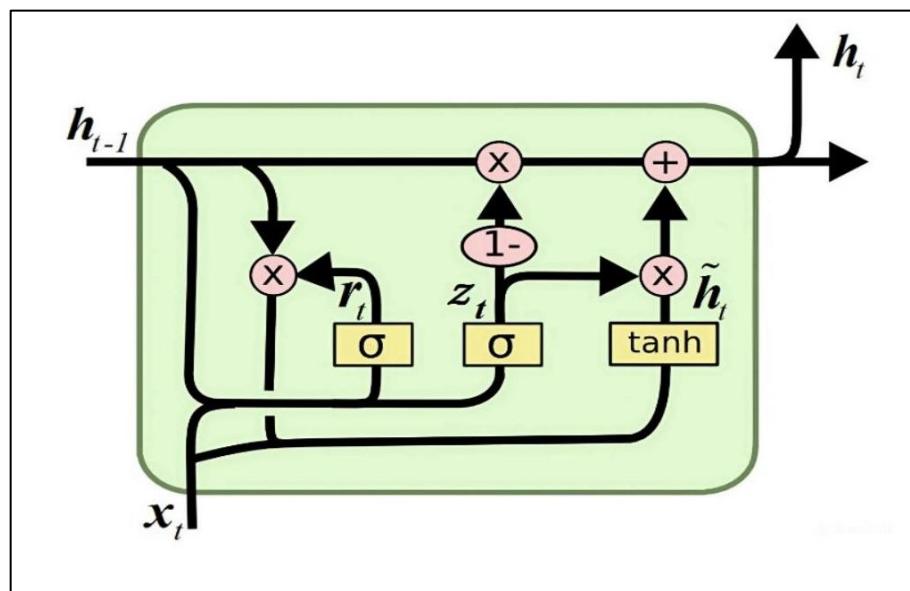
- Trạng thái ô (Cell State: C_t):** Đây là bộ nhớ chính của mô hình, lưu giữ thông tin trong thời gian dài được điều chỉnh bởi các cổng để giữ hoặc loại bỏ dữ liệu tại từng thời điểm.
- Trạng thái ẩn (Hidden State, h_t):** Đóng vai trò là đầu ra tạm thời của LSTM tại mỗi bước thời gian, kết nối trạng thái bộ nhớ với các lớp khác trong mạng.
- Cổng vào (Input Gate, i_t):** Điều khiển lượng thông tin mới được ghi vào bộ nhớ, cho phép mô hình học cách chọn lọc thông tin đầu vào.
- Cổng quên (Forget Gate, f_t):** Xác định thông tin nào từ trạng thái bộ nhớ trước đó sẽ bị loại bỏ, giúp mô hình loại trừ các thông tin không còn giá trị.

5. Cổng ra (Output Gate, O_t): Điều khiển lượng thông tin từ bộ nhớ được chuyển sang đầu ra, ảnh hưởng trực tiếp đến trạng thái ẩn và đầu ra cuối cùng của LSTM tại thời điểm hiện tại.

LSTM đã trở thành một trong những thuật toán Deep Learning không thể thiếu trong lĩnh vực học sâu, đặc biệt là trong các bài toán liên quan đến dữ liệu chuỗi và phụ thuộc theo thời gian. Khả năng ghi nhớ dài hạn, linh hoạt trong việc chọn lọc thông tin, cùng với hiệu suất cao trong các ứng dụng thực tế đã đưa LSTM trở thành một tiêu chuẩn quan trọng trong nhiều lĩnh vực như dịch máy, phân tích cảm xúc, dự báo tài chính, và phát hiện bất thường.

2.3.3. GRU (Gated Recurrent Unit)

GRU (Gated Recurrent Unit) cũng là một kiến trúc thuộc mạng nơ-ron hồi quy RNN (Recurrent Neural Network) được phát triển nhằm giải quyết hiệu quả các bài toán xử lý dữ liệu tuần tự như chuỗi thời gian, văn bản, tín hiệu cảm biến hoặc lưu lượng mạng. GRU được ra đời nhằm đơn giản hóa kiến trúc của LSTM, đồng thời vẫn duy trì hiệu quả trong việc ghi nhớ các phụ thuộc dài hạn trong dữ liệu tuần tự. Vì kiến trúc nhẹ hơn nên GRU thường có tốc độ huấn luyện nhanh hơn LSTM, sử dụng ít tài nguyên tính toán hơn, và phù hợp với các bài toán có dữ liệu vừa và nhỏ hoặc yêu cầu hiệu suất thời gian thực [25].



Hình 7: Kiến trúc mạng GRU (Gated Recurrent Unit) [25]

Kiến trúc của GRU

Kiến trúc của GRU tương tự kiến trúc của LSTM, được xây dựng dựa trên cơ chế công điều khiển hiệu quả, sử dụng hai công điều khiển giúp giám độ phức tạp của mô hình và số lượng tham số cần học so với LSTM. Hai công này bao gồm:

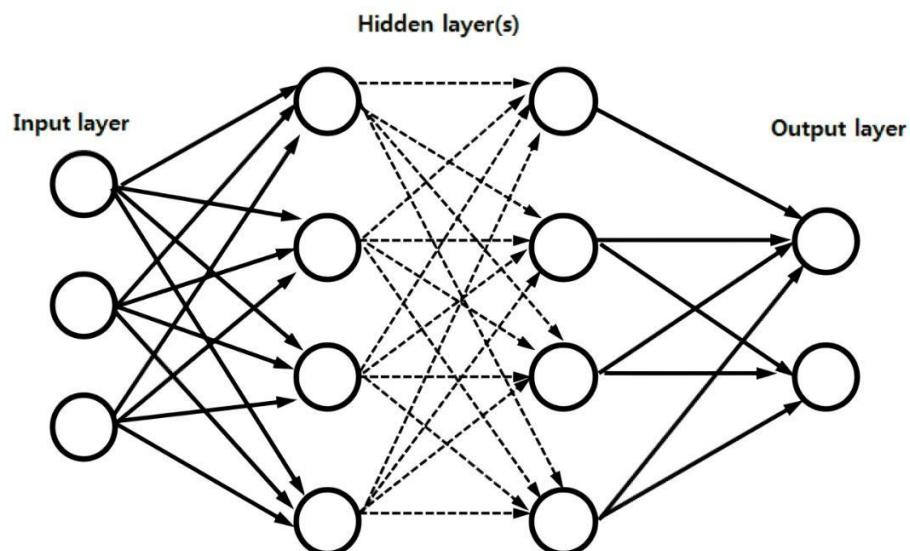
- **Công cập nhật (Update Gate):** kiểm soát lượng thông tin từ trạng thái ẩn trước đó được giữ lại và lượng thông tin mới được thêm vào. Công này kết hợp chức năng của cả công vào và công quên trong LSTM.
- **Công đặt lại (Reset Gate):** xác định mức độ thông tin từ trạng thái ẩn trước đó sẽ bị loại bỏ khi tính toán trạng thái ẩn mới. Điều này giúp GRU "quên" có chọn lọc thông tin cũ không còn hữu ích.

2.3.4. MLP (Multi-Layer Perceptron)

Mạng nơ-ron MLP (Multi-Layer Perceptron) là một mô hình học máy thuộc nhóm mạng nơ-ron truyền thẳng với nhiều lớp ẩn (hidden layer). Nhờ cấu trúc nhiều tầng, MLP có khả năng học và biểu diễn các đặc trưng phức tạp, phi tuyến từ dữ liệu đầu vào, và đã đạt được những thành tựu vượt trội trong các bài toán phân loại, hồi quy và xử lý dữ liệu phức tạp.

Kiến trúc của mạng nơ-ron MLP

Kiến trúc MLP được xây dựng dựa trên việc xếp chồng nhiều lớp nơ-ron, cho phép mô hình học và trích xuất các đặc trưng phân cấp từ dữ liệu đầu vào.



Hình 8: Kiến trúc MLP (Multi-layer Perceptron)[26].

Các thành phần chính bao gồm:

- **Lớp đầu vào (Input Layer):** Chịu trách nhiệm tiếp nhận dữ liệu thô. Số lượng nơ-ron trong lớp này tương ứng với số chiều của véc-tơ đặc trưng đầu vào.
- **Các lớp ẩn (Hidden Layers):** Là thành phần cốt lõi của mạng, thực hiện các phép biến đổi phi tuyến tính trên dữ liệu. Mỗi nơ-ron trong lớp ẩn nhận đầu vào từ tất cả các nơ-ron của lớp trước đó, áp dụng một hàm kích như ReLU, Sigmoid, hoặc Tanh để tạo ra đầu ra. Số lượng lớp ẩn và số nơ-ron trong mỗi lớp quyết định độ sâu và độ rộng của mạng, từ đó ảnh hưởng đến khả năng biểu diễn của mô hình.
- **Lớp đầu ra (Output Layer):** Đưa ra dự đoán cuối cùng. Tùy vào bài toán, lớp này có thể chứa một nơ-ron với bài toán hồi quy hoặc nhiều nơ-ron phân loại đa lớp với hàm softmax.

Quá trình huấn luyện của MLP thường dựa trên thuật toán lan truyền ngược để tính đạo hàm của hàm lỗi theo các trọng số mạng. Gradient này sau đó được sử dụng bởi các thuật toán tối ưu như Gradient Descent, Adam hoặc RMSprop để cập nhật trọng số, nhằm giảm thiểu sai số dự đoán. Tuy hiệu quả, quá trình này có thể gặp thách thức như hiện tượng gradient biến mất khiến việc cập nhật ở các lớp đầu trở nên chậm hoặc kém hiệu quả. Các giải pháp như sử dụng hàm kích hoạt ReLU hay khởi tạo trọng số tốt hơn đã góp phần khắc phục vấn đề này, qua đó nâng cao hiệu suất của MLP trong nhiều bài toán thực tế.

2.3.5. Mô hình kết hợp (Hybird Models)

Mô hình kết hợp hoạt động dựa trên ý tưởng rằng việc kết hợp các dự đoán từ nhiều mô hình sẽ giảm thiểu sai số và tăng khả năng tổng quát hóa. Trong học sâu (Deep Learning), mô hình kết hợp là phương pháp tận dụng sức mạnh của nhiều kiến trúc mạng khác nhau để nâng cao hiệu quả học và khả năng dự đoán. Thay vì chỉ sử dụng một mạng đơn lẻ, mô hình kết hợp cho phép khai thác khả năng trích xuất đặc trưng, xử lý dữ liệu tuần tự và học mối quan hệ phi tuyến giữa các đặc trưng. Các lợi ích chính của mô hình kết hợp bao gồm:

- + Tăng độ chính xác: Kết hợp nhiều mô hình giúp giảm thiểu lỗi dự đoán.
- + Giảm overfitting: Các mô hình khác nhau có thể bù trừ sai lầm của nhau.
- + Tăng độ ổn định: Mô hình tổng hợp thường ít nhạy cảm với nhiễu trong dữ liệu.

Mạng nơ-ron tích chập (Convolutional Neural Network – CNN) là kiến trúc học sâu mạnh mẽ trong việc tự động trích xuất đặc trưng từ dữ liệu thô. CNN đặc biệt hiệu quả với dữ liệu có cấu trúc dạng lưới hoặc ma trận, như hình ảnh hay lưu lượng mạng. CNN có khả năng nhận diện các mẫu cục bộ và giảm nhiễu trong dữ liệu. Tuy nhiên, đôi khi CNN chỉ tập trung vào trích xuất đặc trưng và không được tối ưu để học các phụ thuộc theo thời gian hoặc các mối quan hệ phi tuyến phức tạp. Để bổ sung và nâng cao khả năng này, CNN thường được kết hợp với các mô hình khác:

- + *Nhóm mạng hồi quy RNN (LSTM, GRU, SimpleRNN)*: Bổ sung khả năng học các phụ thuộc tuần tự giữa các đặc trưng do CNN trích xuất, giúp phát hiện các mẫu hành vi xâm nhập có tính thời gian.
- + *Nhóm mạng truyền thẳng MLP* : Học các mối quan hệ phi tuyến giữa các đặc trưng, hỗ trợ phân loại dữ liệu hiệu quả, đặc biệt với những mẫu không có tính tuần tự rõ ràng.

Việc kết hợp mạng nơ-ron tích chập CNN với mạng hồi quy RNN hoặc mạng truyền thẳng MLP cho phép tận dụng thế mạnh trích xuất đặc trưng của CNN, đồng thời bổ sung khả năng học tuần tự hoặc mô hình hóa quan hệ phi tuyến từ các kiến trúc khác.

2.4. Nền tảng Kaggle

Kaggle là một nền tảng trực tuyến thuộc sở hữu của Google, được phát triển với mục tiêu hỗ trợ cộng đồng khoa học dữ liệu, học máy và trí tuệ nhân tạo trên toàn cầu. Với hơn hàng triệu thành viên, Kaggle đã trở thành một trong những môi trường nghiên cứu và học tập phổ biến nhất hiện nay, cung cấp không chỉ tài nguyên tính toán mà còn cả kho dữ liệu, mã nguồn, và các cuộc thi học máy quy mô quốc tế.



Một số đặc điểm nổi bật của Kaggle có thể kể đến:

- **Tài nguyên tính toán mạnh mẽ:** Người dùng có thể tận dụng GPU (như Tesla P100, T4) hoặc TPU để huấn luyện các mô hình học sâu với quy mô lớn mà không cần đầu tư hạ tầng phần cứng.
- **Kho dữ liệu phong phú:** Kaggle cung cấp hàng nghìn bộ dữ liệu mở ở nhiều lĩnh vực khác nhau như y tế, tài chính, mạng máy tính, an ninh mạng, ..., tạo điều kiện thuận lợi cho việc nghiên cứu và thực nghiệm.
- **Môi trường Notebook trực tuyến:** Kaggle Notebook hỗ trợ chạy mã nguồn trực tiếp trên nền tảng đám mây, tích hợp sẵn nhiều thư viện khoa học dữ liệu và học máy như TensorFlow, PyTorch, Scikit-learn, ... giúp tiết kiệm thời gian cài đặt.

Trong phạm vi luận văn này, Kaggle được lựa chọn làm môi trường thí nghiệm chính để triển khai và đánh giá các mô hình học sâu. Nền tảng này không chỉ cung cấp sức mạnh tính toán cần thiết (GPU/TPU) mà còn đảm bảo sự linh hoạt trong việc quản lý và theo dõi quá trình huấn luyện. Nhờ đó, các mô hình có thể được so sánh một cách khách quan cả về hiệu năng phân loại lẫn hiệu quả sử dụng tài nguyên, từ đó mang lại kết quả đánh giá toàn diện hơn.

CHƯƠNG 3: PHƯƠNG PHÁP THỰC HIỆN

3.1. Mô hình hoá bài toán

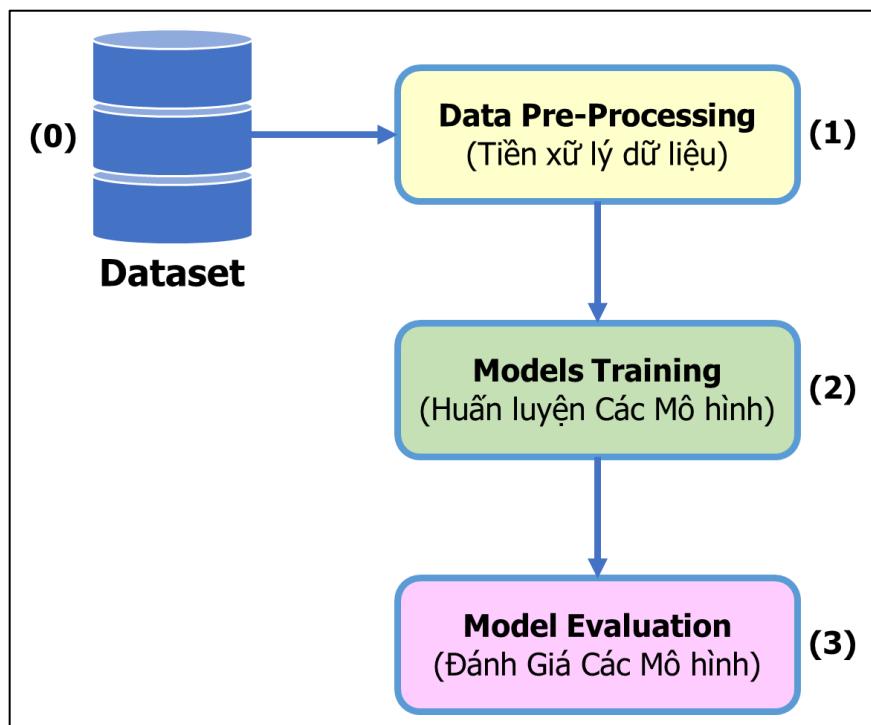
Luận văn tập trung vào thực hiện huấn luyện các mô hình học sâu CNN, LSTM, GRU, MLP và các mô hình học sâu kết hợp CNN–LSTM, CNN–MLP. Mỗi mô hình sở hữu những ưu điểm riêng trong việc khai thác và học các đặc trưng khác nhau của dữ liệu, từ khả năng nắm bắt quan hệ không gian, chuỗi thời gian đến khả năng trích xuất đặc trưng phi tuyến phức tạp. Để nâng cao hiệu suất của các hệ thống phát hiện xâm nhập (IDS), đặc biệt là phát hiện các cuộc tấn công vào các hệ thống mạng khác nhau như: tấn công từ chối dịch vụ (Dos/DDos), tấn công vén cạn (Brute Force Attack), tấn công Web, tấn công thăm dò (Reconnaissance Attack), tấn công giả mạo (Spoofing Attack), tấn công phát lại (Replay Attack) và cùng nhiều biến thể và phương thức tấn công khác.

Các tập dữ liệu chuẩn đại diện cho từng môi trường được sử dụng trong luận văn này bao gồm **CSE-CIC-IDS-2018**, **CIC-IoT-2023** và **ICS-Flow**. Đây là những bộ dữ liệu quy mô lớn, chứa cả lưu lượng mạng bình thường và lưu lượng mạng bất thường, phản ánh đa dạng các kịch bản tấn công mạng trong thực tế. Để giải quyết thách thức mất cân bằng dữ liệu (data imbalance) rất phổ biến trong các bài toán phát hiện tấn công mạng, luận văn này đã áp dụng các kỹ thuật tiền xử lý dữ liệu (Data Processing) phù hợp và các phương pháp xử lý mất cân bằng dữ liệu như Random Under-Sampling (RUS), SMOTE kết hợp Tomek Link. Những kỹ thuật này giúp cải thiện khả năng nhận diện các loại tấn công có số lượng mẫu thấp, giảm tác động của dữ liệu nhiễu, đồng thời duy trì tính đa dạng và sự cân bằng hợp lý giữa các lớp dữ liệu. Trước khi đưa vào huấn luyện, dữ liệu sẽ được tiền xử lý nhằm loại bỏ nhiễu, chuẩn hóa về định dạng phù hợp cho các mô hình học sâu. Tiếp theo, các kỹ thuật cân bằng dữ liệu sẽ được áp dụng để khắc phục sự chênh lệch giữa các lớp, giúp mô hình học hiệu quả hơn và cải thiện khả năng phát hiện các tấn công hiếm gặp. Tập dữ liệu sau khi được xử lý và đã cân bằng sẽ được sử dụng để huấn luyện và đánh giá hiệu suất của các mô hình đề xuất. Hiệu suất này được đo lường thông qua các chỉ số Accuracy, Precision, Recall, F1-score và mức tiêu thụ tài nguyên trên GPU/CPU. Trên cơ sở đó, nghiên cứu sẽ tiến hành phân tích, so sánh ưu và nhược điểm của từng phương pháp học sâu khi áp dụng trên các môi trường mạng khác nhau.

3.2. Tổng quan về các tập dữ liệu

Để phù hợp cho việc thực hiện đề tài của luận văn này, quá trình thực nghiệm được tiến hành khảo sát trên 3 tập dữ liệu (dataset) của các môi trường mạng khác nhau bao gồm **CSE-CIC-IDS-2018**, **CICIoT2023**, **ICS-Flow** tương ứng với 3 môi trường khác nhau là **Mạng doanh nghiệp**, **Mạng IoT** và **Mạng công nghiệp**.

Để đảm bảo đánh giá khách quan và nhất quán hiệu quả của các mô hình phát hiện xâm nhập, một quy trình thực nghiệm chung được thiết kế để áp dụng cho các tập dữ liệu. Quy trình này được áp dụng nhất quán cho cả ba tập dữ liệu: **CSE-CIC-IDS-2018**, **CIC-IoT-2023** và **ICS-Flow**, nhằm đảm bảo tính đồng bộ trong quá trình phân tích và đánh giá. Toàn bộ quy trình bao gồm ba giai đoạn chính được thể hiện qua mô hình tổng quan tại **Hình 9** đó là Tiền xử lý dữ liệu (Data Pre-Processing), Huấn luyện các mô hình (Models Training), Đánh giá các mô hình (Model Evaluation).

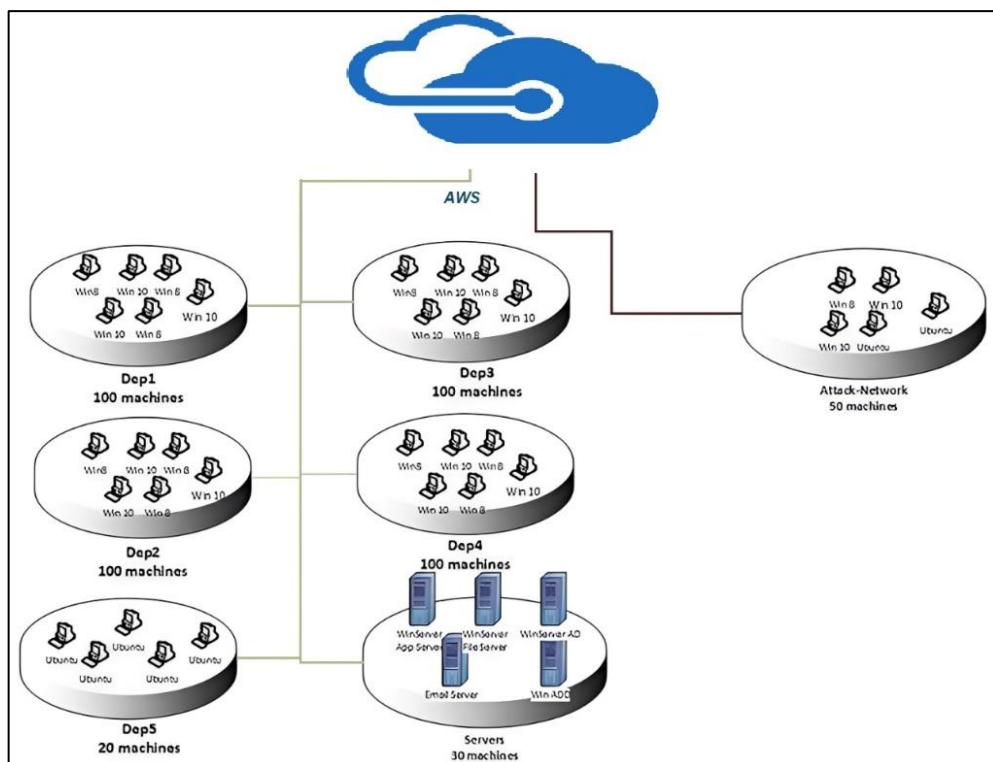


Hình 9: Mô hình tổng quan

3.2.1. Tập dữ liệu CSE-CIC-IDS-2018

Tập dữ liệu CSE-CIC-IDS-2018 [27] là một tập hợp dữ liệu toàn diện và thực tế về an ninh mạng, được phát triển thông qua sự hợp tác và phối hợp giữa Trung tâm An ninh Viễn thông Canada (CSE), Viện An ninh mạng Canada (CIC) và Amazon Web Services (AWS). Tập dữ liệu này được tạo ra một cách có hệ thống bằng cách mô phỏng các hoạt động mạng thông thường và các cuộc tấn công mạng đa dạng, nhằm mục đích hỗ trợ phát triển và đánh giá các hệ thống phát hiện xâm nhập (IDS).

Đây là một tập dữ liệu lớn, dữ liệu được thu thập trong 10 ngày làm việc liên tục từ ngày 14/02/2018 đến ngày 02/03/2018 bao gồm cả các tệp nhật ký thô với tổng dung lượng khoảng 450 GB và các tệp CSV đã được trích xuất đặc trưng bằng phương pháp CIC để phục vụ cho việc phân tích có tổng dung lượng khoảng 16-20 GB. Kịch bản mô phỏng có cơ sở hạ tầng tấn công bao gồm 50 máy tính và tổ chức nạn nhân có 5 phòng ban, bao gồm 420 máy tính và 30 máy chủ được thể hiện ở **Hình 10**. Tập dữ liệu bao gồm lưu lượng mạng và nhật ký hệ thống của mỗi máy tính, cùng với 80 tính năng được trích xuất từ lưu lượng đã thu thập bằng CICFlowMeter-V3 được trình bày ở **Bảng 2**.



Hình 10: Cấu trúc mạng mô phỏng của tập dữ liệu CSE-CIC-IDS-2018.

Tập dữ liệu CSE-CIC-IDS-2018 được xây dựng với nội dung mô phỏng nhiều kịch bản tấn công mạng thực tế trong môi trường doanh nghiệp. Dữ liệu bao gồm hai nhóm chính: lưu lượng bình thường (Benign) và lưu lượng tấn công (Attacks), trong đó các loại tấn công được chia thành các nhóm tiêu biểu như: DDoS (Tấn công từ chối dịch vụ phân tán), DoS (Tấn công từ chối dịch vụ), Botnet, Brute-force, và Web Attack (Tấn công web). Thống kê số lượng mẫu được thể hiện ở **Hình 11**. về các loại tấn công và các loại nhóm tấn công chính trong tập dữ liệu.

Attack type	Rows	Attack type	Rows
Begin	9,533,886	DoS attacks-Slowloris	9,908
DDoS attacks-LOIC-HTTP	575,364	DDoS attacks-LOIC-UDP	1,730
DDoS attacks-HOIC	198,861	Bruteforce-Web	555
DDoS attacks-Hulk	145,456	Bruteforce-XSS	228
Bot	144,535	DoS attacks-SlowHTTPTest	98
Infiltration	116,738	SQL Injection	84
SSH-Bruteforce	94,048	FTP Bruteforce	54
DoS attacks-GoldenEye	41,406		
Attack Category	Rows	Attack type	
Benign	9,533,886	Benign	
DDoS	775,955	DDoS attacks-LOIC-HTTP, DDoS-LOIC-UDP, and DDOS-HOIC	
DoS	196,868	DoS-GoldenEye, DoS-Slowloris, DoS-SlowHTTPTest, and DoS-Hulk	
Bot	144,535	Bot	
Brute-force	94,102	Brute Force-Web, Brute Force-XSS, and SQL Injection	
Web attack	867	FTP-Bruteforce and SSH-Bruteforce	

Hình 11: Thống kê số lượng mẫu trong tập dữ liệu CSE-CIC-IDS-2018[27].

STT	Nhóm đặc trưng	Ghi chú các đặc trưng liên quan
1-4	Các đặc trưng cơ bản của kết nối mạng	Destination Port, Flow Duration, Total Fwd Packets, Total Backward Packets
5-16	Đặc trưng các gói mạng	Total Length Fwd/Bwd Packets, Packet Length Min/Max/Mean/Std, Flow Bytes/s, Flow Packets/s

17–30	Đặc trưng về thời gian luồng	Flow IAT Mean/Std/Max/Min, Fwd/Bwd IAT Total, Mean, Std, Max, Min
31–34	Đặc trưng về cờ TCP	Fwd/Bwd PSH Flags, URG Flags
35–38	Đặc trưng về header và tốc độ	Fwd/Bwd Header Length, Fwd/Bwd Packets/s
39–42	Đặc trưng về kích thước gói	Packet Length Min/Max/Mean/Std
43–50	Đặc trưng về cờ flag	FIN, SYN, RST, PSH, ACK, URG, CWE, ECE Flag
51–61	Đặc trưng về tỉ lệ, segment, bulk rate	Down/Up Ratio, Avg Packet Size, Avg Segment Size (Fwd/Bwd), Bulk Rate (Fwd/Bwd), Avg Bytes/Bulk
62–65	Đặc trưng về subflow	Subflow Fwd/Bwd Packets, Bytes
66–67	Đặc trưng về TCP Window Size	Init_Win_bytes_forward/backward
68–69	Đặc trưng về dữ liệu thực tế	act_data_pkt_fwd, min_seg_size_forward
70–73	Đặc trưng về thời gian hoạt động	Active Mean/Std/Max/Min
74–77	Đặc trưng về thời gian rảnh	Idle Mean/Std/Max/Min
78–80	Đặc trưng nhãn và thông tin bổ sung	Label, Protocol, Timestamp

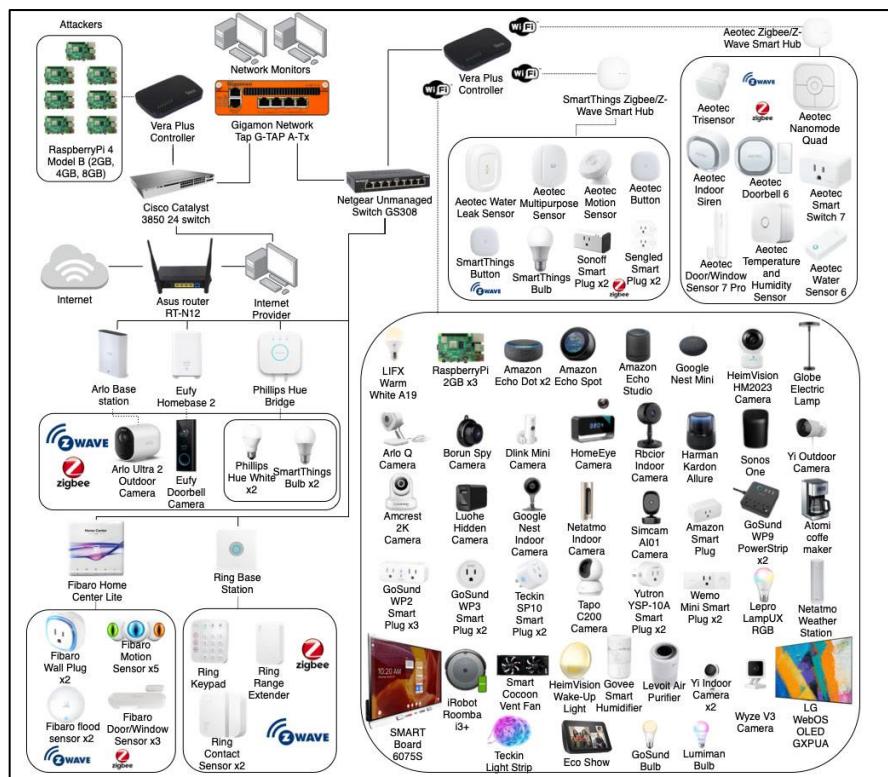
Bảng 2: Bảng đặc trưng tập dữ liệu CSE-CIC-IDS-2018.

Tập dữ liệu bao gồm lưu lượng mạng bình thường và lưu lượng mạng bất thường thể hiện các loại tấn công được mô phỏng theo các kịch bản sau:

- **Brute-force:** Các tấn công kiểu brute-force nhằm dò đoán mật khẩu dịch vụ bằng cách (SSH/FTP) thử liên tục nhiều tổ hợp username/password, dẫn đến số lượng lớn phiên đăng nhập thất bại trong thời gian ngắn hoặc nhiều kết nối ngắn bị reset. *Tập dữ liệu bao gồm nhiều biến thể như: SSH-Bruteforce, FTP-Bruteforce.*
- **DoS (Denial of Service):** Tấn công từ một hoặc số ít nguồn nhằm làm nghẽn hoặc gián đoạn dịch vụ mục tiêu bằng cách gửi lượng lớn yêu cầu hoặc kết nối, khiến server không thể đáp ứng user hợp lệ. *Tập dữ liệu bao gồm nhiều biến thể như: Slowloris, SlowHTTPTest, Hulk, GoldenEye.*
- **DDoS (Distributed Denial of Service):** Nhiều nguồn (máy/botnet) phối hợp gửi lượng lớn lưu lượng hoặc yêu cầu đến mục tiêu cùng lúc, làm cạn kiệt tài nguyên và gây tê liệt dịch vụ. *Tập dữ liệu bao gồm nhiều biến thể như: LOIC-UDP, LOIC-TCP, LOIC-HTTP, HOIC.*
- **Web attack:** Tấn công vào các dịch vụ web nhằm khai thác lỗ hổng ứng dụng như chèn mã độc, dò mật khẩu quản trị, hoặc gửi truy vấn nguy hiểm. *Tập dữ liệu bao gồm nhiều biến thể như: Brute Force-Web, Brute Force-XSS, SQL Injection.*
- **Botnet:** Máy bị nhiễm malware tiến hành kết nối tới server điều khiển (C&C), nhận lệnh và có thể phối hợp thực hiện tấn công khác (ví dụ: DDoS, phát tán mã độc).

3.2.2. Tập dữ liệu CIC - IoT -2023

Tập dữ liệu CIC-IoT-2023 [28] là một tập dữ liệu lớn về mạng IoT được phát triển bởi Viện An ninh mạng Canada (CIC) có vị thế nổi bật trong hệ sinh thái an ninh mạng và bề dày lịch sử đóng góp to lớn cho ngành công nghiệp và học thuật. Tập dữ liệu được thu thập trong năm 2023, nhằm đáp ứng nhu cầu ngày càng cao trong việc nghiên cứu và phát triển các hệ thống phát hiện xâm nhập (IDS) trong môi trường mạng IoT. Tập dữ liệu mô phỏng được các kịch bản tấn công thực tế thường xảy ra trên mạng IoT bao gồm nhiều thiết bị thông minh khác nhau như camera an ninh, máy in, cảm biến và các thiết bị điều khiển thông minh trong nhà. Cấu trúc các thiết bị IoT được triển khai để sản xuất tập dữ liệu CIC-IoT-2023 được minh họa trong **Hình 12**.



Hình 12: Các thiết bị IoT được triển khai của tập dữ liệu CIC-IoT-2023 [28].

Tập dữ liệu này bao gồm 47 đặc trưng được trích xuất bởi công cụ CICFlowMeter được thể hiện ở **Bảng 3**. Dữ liệu thu gọn hoặc lưu trữ nội bộ bởi CIC vào khoảng tổng 27 GB tính cả các file CSV đặc trưng trích xuất, cung cấp một nguồn tài nguyên lớn và có cấu trúc rõ ràng giúp thuận tiện cho việc huấn luyện mô hình học máy và học sâu. Tập dữ liệu này không chỉ cung cấp một nguồn dữ liệu phong phú để đánh giá các thuật toán, mà còn là một trong những tập dữ liệu đầu tiên tích hợp nhiều dạng tấn công IoT phức tạp phản ánh sát với thực tế. Thông kê số lượng mẫu được thể hiện ở **Hình 13**.

Attack type	Rows	Attack type	Rows
DDoS-ICMP_Flood	7,200,504	DoS-TCP_Flood	2,671,445
DDoS-UDP_Flood	5,412,287	DoS-SYN_Flood	2,028,834
DDoS-TCP_Flood	4,497,667	BenignTraffic	1,098,195
DDoS-PSHACK_Flood	4,094,755	Mirai-greeth_flood	991,866
DDoS-SYN_Flood	4,059,190	Mirai-udpplain	890,576
DDoS-RSTFINFlood	4,045,285	Mirai-greib_flood	751,682
DDoS-SynonymousIP_Flood	3,598,138	DDoS-ICMP_Fragmentation	452,489
DoS-UDP_Flood	3,318,595	MITM-ArpSpoofing	307,593
Recon-PingSweep	2,262	Uploading_Attack	1,252
DDoS-UDP_Fragmentation	286,925	DoS-HTTP_Flood	28,790
DDoS-ACK_Fragmentation	285,104	DDoS-SlowLoris	23,426
DNS_Spoofing	178,911	DictionaryBruteForce	13,064
Recon-HostDiscovery	134,378	BrowserHijacking	5,859
Recon-OSScan	98,259	CommandInjection	5,409
Recon-PortScan	82,284	SqlInjection	5,245
DoS-HTTP_Flood	71,864	XSS	3,846
VulnerabilityScan	37,382	Backdoor_Malware	3,218
Attack Category	Rows	Attack type	
DDoS	33,984,560	DDoS-ICMP_Flood, DDoS-UDP_Flood, DDoS-TCP_Flood, DDoS-PSHACK_Flood,....	
DoS	8,090,738	DoS-UDP_Flood, DoS-TCP_Flood, DoS-SYN_Flood, DoS-HTTP_Flood	
Mirai	2,634,124	Mirai-greeth_flood, Mirai-udppain, Mirai-greib_flood	
Benign	1,098,195	BenignTraffic	
Spoofing	486,504	DNS_Spoofing, MITM-ArpSpoofing	
Recon	354,565	Recon-PingSweep, Recon-HostDiscovery, Recon-OSScan, Recon-PortScan	
BruteForce	13,064	DictionaryBruteForce, Backdoor_Malware	

Hình 13: Thống kê số lượng mẫu trong tập dữ liệu CIC-IoT-2023.

STT	Nhóm đặc trưng	Ghi chú các đặc trưng liên quan
1–4	Các đặc trưng cơ bản của kết nối mạng	ts, flow duration, Header Length, Protocol Type
5–15	Đặc trưng của các gói mạng	Duration, Rate, Srate, Drate, fin/ syn/ rst/ psh/ ack/ ece/ cwr flag number
16–20	Đặc trưng đếm gói theo loại cờ	ack count, syn count, fin count, urg count, rst count
21–34	Đặc trưng về giao thức mạng	HTTP, HTTPS, DNS, Telnet, SMTP, SSH, IRC, TCP, UDP, DHCP, ARP, ICMP, IPv, LLC

35-42	Đặc trưng về kích thước và số lượng gói	Tot sum, Min, Max, AVG, Std, Tot size, IAT, Number
43-47	Đặc trưng thống kê nâng cao về lưu lượng	Magnitude, Radius, Covariance, Variance, Weight

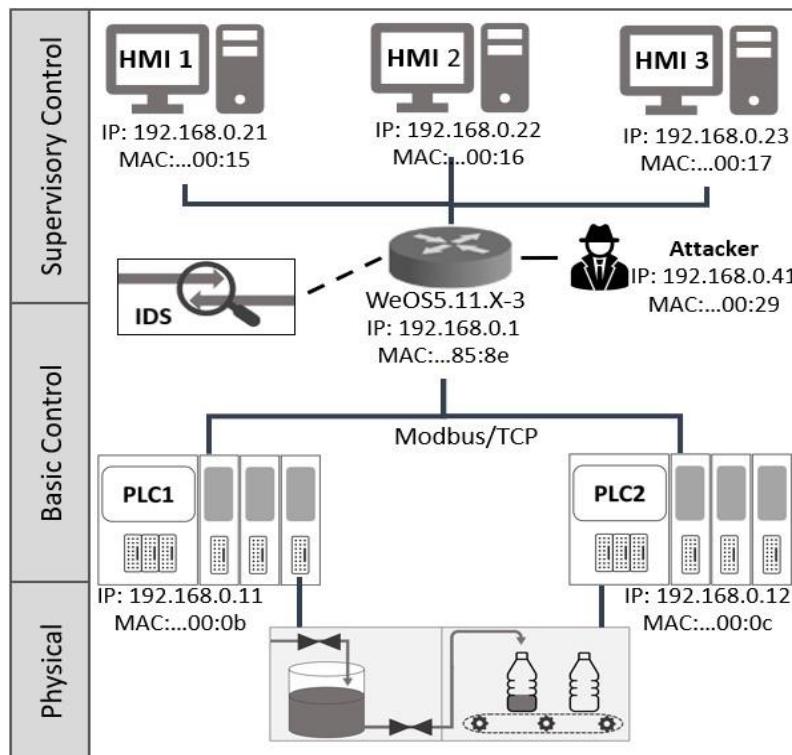
Bảng 3: Bảng đặc trưng tập dữ liệu CIC-IoT-2023

Tập dữ liệu bao gồm lưu lượng mạng bình thường và lưu lượng mạng bất thường thể hiện các loại tấn công được mô phỏng theo các kịch bản sau:

- **DDoS (Distributed Denial of Service):** Các tấn công từ chối dịch vụ phân tán nhằm làm nghẽn mạng hoặc thiết bị mục tiêu qua lượng lớn lưu lượng hoặc yêu cầu. Nhiều thiết bị độc hại phối hợp gửi lượng lớn lưu lượng tới mục tiêu nhằm làm kiệt tài nguyên hoặc làm quá tải, khiến dịch vụ bị gián đoạn. *Tập dữ liệu bao gồm nhiều biến thể như: ICMP_Flood, UDP_Flood, TCP_Flood, PSHACK_Flood, SYN_Flood, RSTFINFlood, SynonymousIP_Flood, UDP_Fragmentation, ACK_Fragmentation, ICMP_Fragmentation, SlowLoris.*
- **DoS (Denial of Service):** Tương tự DDoS nhưng thường xuất phát từ một nguồn, gây gián đoạn dịch vụ. Các thiết bị tấn công mục tiêu bằng cách gửi liên tục yêu cầu đến nạn nhân từ một nguồn duy nhất, làm nghẽn hoặc ngừng dịch vụ. *Tập dữ liệu bao gồm nhiều biến thể như: UDP_Flood, TCP_Flood, SYN_Flood, HTTP_Flood.*
- **Reconnaissance (Tấn công thăm dò) :** Tấn công này nhằm thu thập thông tin về mạng hoặc các thiết bị IoT để chuẩn bị cho các hành động xâm nhập tiếp theo, thường qua kỹ thuật quét cổng, phát hiện địa chỉ IP, dịch vụ chạy trên thiết bị. *Tập dữ liệu bao gồm nhiều biến thể như: PingSweep, HostDiscovery, OSScan, PortScan.*
- **Brute-force:** Hay còn gọi là tấn công vén cạn thường nhắm vào dịch vụ từ xa như SSH/Telnet bằng cách thử nhiều cặp username/password mặc định hoặc yếu. Kiểu tấn công này có thể dẫn tới chiếm quyền thiết bị và cài đặt mã độc. *Tập dữ liệu bao gồm nhiều biến thể như DictionaryBruteForce và Backdoor_Malware.*
- **Spoofing (Giả mạo) :** Tấn công mạo danh danh tính để đánh lừa hệ thống. Kẻ tấn công giả mạo địa chỉ IP/MAC hoặc bản ghi DNS để đánh lừa các thiết bị IoT, thực hiện MITM hoặc chiếm quyền kiểm soát luồng dữ liệu. *Tập dữ liệu bao gồm nhiều biến thể như: ARP Spoofing, DNS Spoofing.*

3.2.3. Tập dữ liệu ICS-Flow

Tập dữ liệu ICS-Flow [29] là tập dữ liệu được tạo ra từ mô phỏng Hệ thống điều khiển công nghiệp (Industrial Control System - ICS) bằng công cụ ICSSIM (Industrial Control System Simulator), mô phỏng nhà máy đóng chai với các thiết bị điều khiển như van nước, bồn chứa và băng tải. Kiến trúc mô hình thử nghiệm trên hệ thống điều khiển ICS được thể hiện qua **Hình 14**. Tập dữ liệu thu được tổng cộng 2GB lưu lượng mạng thô, khoảng 25 triệu gói tin bao gồm bản ghi lưu lượng dạng flow (luồng) và nhật ký biến đổi trạng thái quá trình, chứa cả lưu lượng bình thường (Benign) lẫn bất thường đồng thời chứa các tình huống tấn công mạng thực tế như DDoS (Tấn công từ chối dịch vụ), MitM (Man-In-Middle), Replay (Tấn công phát lại) và Reconnaissance (Tấn công thăm dò), thống kê số lượng mẫu được thể hiện ở **Hình 15**. Tập dữ liệu này bao gồm 60 đặc trưng được trích xuất từ công cụ mã nguồn mở ICSFlowGenerator được thể hiện qua **Bảng 4**, công cụ này hỗ trợ trích xuất đặc trưng từ dữ liệu thô để phục vụ huấn luyện các hệ thống phát hiện bất thường và xâm nhập trong môi trường hệ thống công nghiệp. ICS-Flow là tập dữ liệu quan trọng giúp nghiên cứu và phát triển giải pháp an ninh mạng cho hệ thống công nghiệp.



Hình 14: Cấu trúc mạng mô hình thử nghiệm hệ thống ICS [29].

Attack Category	Rows
Normal (Benign)	30,236
Replay	4,300
DDoS	4,221
Port-Scan	3,235
MitM	3,014
IP-Scan	712

Hình 15: Thống kê số lượng mẫu tập dữ liệu ICS-Flow[29].

STT	Nhóm đặc trưng	Ghi chú các đặc trưng liên quan
1–7	Các đặc trưng cơ bản của kết nối mạng	sAddress, rAddress, sMACs, rMACs, sIPs, rIPs, protocol
8–13	Đặc trưng về thời gian luồng	startDate, endDate, start, end, startOffset, endOffset
14–15	Đặc trưng về thời lượng luồng	duration, tổng số packet (sPackets, rPackets)
16–24	Đặc trưng về kích thước gói & tải	sBytesSum, rBytesSum, sBytesMax, rBytesMax, sBytesMin, rBytesMin, sBytesAvg, rBytesAvg, sLoad, rLoad
25–33	Đặc trưng về payload	sPayloadSum, rPayloadSum, sPayloadMax, rPayloadMax, sPayloadMin, rPayloadMin, sPayloadAvg, rPayloadAvg, sInterPacketAvg, rInterPacketAvg
34–35	Đặc trưng thời gian sống của gói tin	sttl, ttl
36–43	Đặc trưng về tỷ lệ cờ TCP	sAckRate, rAckRate, sUrgRate, rUrgRate, sFinRate, rFinRate, sPshRate, rPshRate
44–49	Đặc trưng về tỷ lệ cờ TCP mở rộng	sSynRate, rSynRate, sRstRate, rRstRate, sWinTCP, rWinTCP
50–51	Đặc trưng về Fragment	sFragmentRate, rFragmentRate

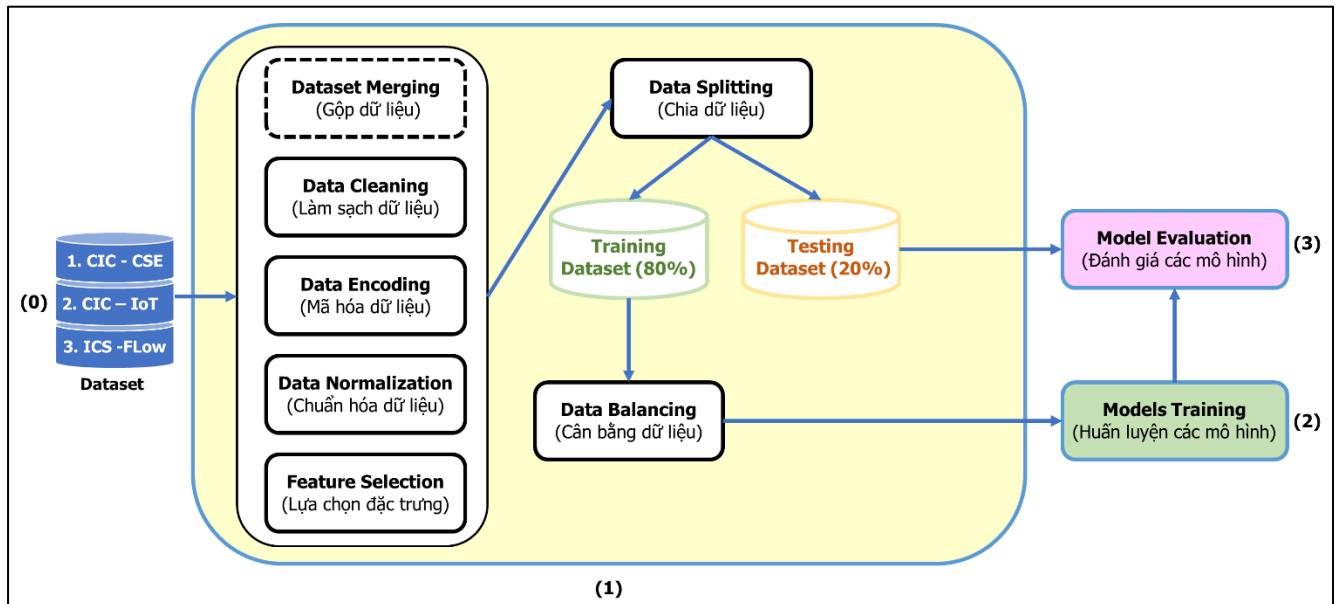
52–57	Đặc trưng về độ trễ ACK	sAckDelayMax,rAckDelayMax,sAckDelayMin, rAckDelayMin, sAckDelayAvg, rAckDelayAvg
58–60	Nhãn phân loại	IT_B_Label, IT_M_Label

Bảng 4: Bảng đặc trưng tập dữ liệu ICS-Flow.

Tập dữ liệu bao gồm lưu lượng mạng bình thường và lưu lượng mạng bất thường thể hiện các loại tấn công được mô phỏng theo các kịch bản sau:

- **Reconnaissance (Tấn côn thăm dò):** Kẻ tấn công sử dụng công cụ như Ettercap để thu thập thông tin hệ thống (IP, MAC, cổng mở) mà không gây gián đoạn hoạt động, thường là bước chuẩn bị cho các tấn công phức tạp hơn. *Tập dữ liệu bao gồm hai biến thể là IP-Scan , Port-Scan.*
- **Replay Attack (Tấn công phát lại):** Phát lại các gói tin hợp lệ đã thu thập trước đó để gây ra hành vi bất thường trong hệ thống, không đòi hỏi kiến thức sâu về cấu trúc gói tin.
- **DDoS (Distributed Denial of Service):** Làm tràn ngập PLC với lượng lớn yêu cầu, gây tắc nghẽn và chậm trễ giao tiếp trong mạng ICS.
- **MitM (Man-in-the-Middle):** Chặn và chỉnh sửa dữ liệu Modbus trong quá trình truyền, đưa dữ liệu sai lệch vào hệ thống điều khiển, gây sai số hoặc gián đoạn quá trình sản xuất.

3.3. Data Pre-Processing (Chuẩn bị và tiền xử lý dữ liệu)



Hình 16: Mô hình hóa chi tiết cho bước Tiền xử lý dữ liệu.

3.3.1. Dataset Merging (Gộp dữ liệu)

Việc gộp file là một bước xử lý dữ liệu cần thiết đối với các tập dữ liệu lớn trong một tập dữ liệu gồm nhiều file nhỏ có thể phân loại theo thời gian, loại thiết bị, hoặc loại tấn công. Cần chọn ra các file phù hợp và kết hợp nhiều phần dữ liệu thành một tập thống nhất để đảm bảo đầy đủ thông tin và kích thước mẫu huấn luyện phù hợp.

Tuy nhiên, việc gộp file chỉ áp dụng với một số tập dữ liệu nhất định. Trong thực nghiệm này các tập dữ liệu lớn như CSE-CIC-IDS-2018 và CIC-IoT-2023 thường bao gồm nhiều file nhỏ [24][25]. Để có một tập dữ liệu thống nhất, đủ lớn cho việc huấn luyện, cần phải chọn và gộp các file này lại. Đối với các tập dữ liệu khác (trong thực nghiệm này là tập dữ liệu ICS-Flow) đã được cung cấp dưới dạng thống nhất, bước này sẽ không cần thực hiện.

Phân bố số lượng mẫu của 3 tập dữ liệu ban đầu được sử dụng gồm:

Bảng 5: Phân bố số lượng mẫu ban đầu của Tập dữ liệu (1):CSE-CIC-IDS-2018.

STT	Loại Malware	Số lượng mẫu tin
0	Benign	4.133.778
1	DDoS	200.591
2	DoS	196.568
3	Bot	144.535
4	Brute-force	94.102
5	Web attack	867
Tổng cộng		4.770.441

Bảng 6: Phân bố số lượng mẫu ban đầu của Tập dữ liệu (2):CIC-IoT-2023.

STT	Loại Malware	Số lượng mẫu tin
0	Benign	1.098.195
1	DDoS	3.397.878
2	DoS	809.101
3	Bot-Mirai	263.965
4	Spoofing	486.504
5	Recon	82.284
6	Brute-Force	13.064
Tổng cộng		6.150.991

Bảng 7: Phân bố số lượng mẫu ban đầu của Tập dữ liệu (3):ICS-Flow.

STT	Loại Malware	Số lượng mẫu tin
0	Normal(Benign)	30.236
1	IP-Scan	712
2	Port-Scan	3.235
3	Replay	4.300
4	DDoS	4.221
5	MitM	3.014
Tổng cộng		45.718

3.3.2. Data Cleaning (Làm sạch dữ liệu)

Làm sạch dữ liệu là một bước tiền xử lý quan trọng nhằm nâng cao chất lượng dữ liệu đầu vào. Trong một tập dữ liệu, các cột thể hiện các đặc trưng (features) của lưu lượng mạng, còn các dòng là các bản ghi (records) về từng luồng dữ liệu (flow) cụ thể. Các bước làm sạch dữ liệu bao gồm:

- Loại bỏ các cột không liên quan đến việc phân loại tấn công. Các cột này thường chứa các giá trị duy nhất cho mỗi bản ghi như mã định danh (ID), địa chỉ IP, dấu thời gian (timestamp) hoặc các trường thông tin không mang lại giá trị phân tích cho mô hình.
- Xử lý các giá trị bị thiếu (Null/NaN), giá trị vô hạn (infinity) hoặc các giá trị không hợp lệ (invalid values) như ký tự không đúng định dạng, giá trị vượt quá phạm vi chấp nhận được hoặc không có ý nghĩa về mặt ngữ cảnh dữ liệu.
- Loại bỏ dòng trùng lặp (Duplicates) và các dòng không có ý nghĩa phân tích.
- Loại bỏ các đặc trưng chỉ có một giá trị duy nhất, ví dụ toàn cột chỉ có giá trị 0.
- Loại bỏ các cột trùng nhau giữ lại một cột.

Đây là các bước làm sạch dữ liệu chuẩn [30] và mục tiêu chính của việc làm sạch dữ liệu là đảm bảo tập dữ liệu có tính nhất quán, đầy đủ và chính xác. Điều này giúp loại bỏ nhiễu, giảm thiểu sai sót, từ đó nâng cao hiệu suất và độ tin cậy của mô hình.

3.3.3. Data Encoding (Mã hóa dữ liệu)

Mã hóa dữ liệu là một bước nền tảng trong tiền xử lý dữ liệu, nơi chúng ta chuyển đổi các dữ liệu phi số, chẳng hạn như văn bản (text) hay danh mục (category), thành định dạng số (numeric) mà các mô hình học máy và học sâu có thể xử lý. Điều này đặc biệt quan trọng để mô hình có thể "hiểu" và học từ dữ liệu.

Trong các tập dữ liệu dùng cho bài toán học có giám sát (Supervised Learning), cột chứa nhãn (Label) hay còn gọi là cột đặc trưng mục tiêu (Target feature) chứa thông tin về lớp hoặc trạng thái của từng mẫu dữ liệu. Trong bài toán phát hiện xâm nhập, cột nhãn (Label) chứa thông tin xác định lưu lượng mạng đó là bình thường (Benign) hay thuộc một loại tấn công cụ thể (Bruteforce, DoS, Web Attack, Botnet hoặc DDoS,...). Đây chính là giá trị đầu ra mà mô hình cần dự đoán dựa trên các đặc trưng đầu vào, từ đó đánh giá hiệu suất của mô hình. Do các thuật toán học sâu không thể xử lý trực tiếp dữ liệu dạng văn bản, cột nhãn cần được mã hóa. Quá trình mã hóa dữ liệu trong này

gồm 2 bước chính là Label Encoding (Mã hóa nhãn) và One-Hot Encoding:

- **Label Encoding (Mã hóa nhãn):** Đây là quá trình gán một giá trị số duy nhất cho mỗi nhãn phân loại. Trong luận văn này thực hiện bài toán phân loại đa lớp (Multi-class classification), nhãn "Benign" sẽ luôn được gán số 0, các loại tấn công khác sẽ được gán số tăng dần 1, 2, 3,... Việc gán số này phụ thuộc vào số lượng nhãn phân loại duy nhất có trong tập dữ liệu.
- **One-Hot Encoding:** là một kỹ thuật mã hóa dữ liệu quan trọng, được sử dụng để xử lý các nhãn phân loại sau khi mã hóa nhãn (Label Encoding). Mục đích chính của nó là chuyển đổi các giá trị số này thành các vector nhị phân, giúp các mô hình học sâu không hiểu sai mối quan hệ trong cột đặc trưng mục tiêu (Target feature) cụ thể là cột “Label” là thứ tự giữa các nhãn. Ví dụ, nếu tập dữ liệu có ba nhãn đã được mã hóa số là 0 (Benign), 1 (DDoS), và 2 (Bot), One-Hot Encoding ta được các vector như sau:
 - + Nhãn 0 đại diện cho (Benign) sẽ được mã hóa thành : [1, 0, 0].
 - + Nhãn 1 đại diện cho (DDoS) sẽ được mã hóa thành : [0, 1, 0].
 - + Nhãn 2 đại diện cho (Bot) sẽ được mã hóa thành : [0, 0, 1].

3.3.4. Data Normalization (Chuẩn hóa dữ liệu)

Chuẩn hóa dữ liệu là một kỹ thuật tiền xử lý được sử dụng để đưa dữ liệu số về một khoảng giá trị nhất định, thường là từ 0 đến 1 hoặc từ -1 đến 1 [30]. Mục tiêu của quá trình này là đảm bảo rằng các đặc trưng (feature) khác nhau đóng góp ngang nhau vào mô hình học máy, ngăn không cho các đặc trưng có miền giá trị lớn lấn át các đặc trưng có miền giá trị nhỏ hơn.

Có nhiều kỹ thuật chuẩn hóa dữ liệu khác nhau như Min-Max Scaling, Z-Score Normalization, hoặc Robust Scaling. Trong luận văn này, chúng tôi lựa chọn sử dụng kỹ thuật Min-Max Scaling để chuẩn hóa tất cả các giá trị đặc trưng về khoảng [0, 1]. Kỹ thuật này đặc biệt hiệu quả trong việc xử lý các đặc trưng có giá trị lớn và phân bố không đồng đều, đảm bảo rằng không có đặc trưng nào chiếm ưu thế quá mức trong quá trình huấn luyện mô hình [31].

Công thức của kỹ thuật Min-Max Scaling được định nghĩa như sau:

$$X_{Normalized} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

- $X_{Normalized}$: Giá trị sau khi chuẩn hoá
- X: Giá trị gốc của đặc trưng
- X_{min} : Giá trị nhỏ nhất của đặc trưng
- X_{max} : Giá trị lớn nhất của đặc trưng

3.3.5. Feature Selection (Lựa chọn đặc trưng)

Lựa chọn đặc trưng (Feature Selection) là quá trình đánh giá và trích chọn các đặc trưng quan trọng từ tập dữ liệu, nhằm loại bỏ các đặc trưng dư thừa, ít đóng góp hoặc gây nhiễu cho mô hình. Việc này giúp tối ưu hóa hiệu suất, cải thiện khả năng tổng quát hóa từ đó giúp mô hình học tốt hơn và tránh tình trạng “học vẹt” (overfitting) từ đó nâng cao chất lượng dự đoán.

Đối với các kỹ thuật học sâu (deep learning), lựa chọn đặc trưng không phải là bước bắt buộc, vì các mô hình này có khả năng tự động trích xuất và học biểu diễn đặc trưng trực tiếp từ dữ liệu thô như thuật toán mạng nơ ron tích chập (CNN) học bộ lọc để trích xuất đặc trưng không gian, Bộ nhớ ngắn-dài hạn (LSTM) học đặc trưng theo chuỗi thời gian. Tuy nhiên, trong khuôn khổ luận văn này, để phục vụ việc đánh giá trên ba tập dữ liệu khác nhau, đồng thời giảm yêu cầu tài nguyên tính toán và rút ngắn thời gian huấn luyện, tôi vẫn áp dụng bước lựa chọn đặc trưng sử dụng thuật toán **Random Forest** để đánh giá mức độ quan trọng của các đặc trưng, đây là một phương pháp lựa chọn đặc trưng đơn giản nhưng hiệu quả [31], vừa đảm bảo khả năng so sánh kết quả giữa các mô hình trên nhiều tập dữ liệu, vừa nâng cao hiệu quả và tốc độ huấn luyện.

3.3.6. Data Splitting (Chia dữ liệu)

Chia dữ liệu là bước cơ bản trong quy trình xây dựng mô hình học máy, nhằm phân tách tập dữ liệu ban đầu thành các phần riêng biệt như huấn luyện (training), xác thực (validation) và kiểm thử (testing). Mục tiêu chính là bảo đảm việc đánh giá mô hình được thực hiện trên dữ liệu mà mô hình chưa từng thấy, qua đó phản ánh chính xác khả năng tổng quát hóa và hạn chế hiện tượng overfitting. Trong nghiên cứu này, dữ liệu được chia theo tỷ lệ 80% – 20%, trong đó 80% được sử dụng làm tập huấn luyện (training set) và còn 20% còn lại được sử dụng làm tập kiểm thử (testing set) để đánh giá khách quan hiệu suất cuối cùng của mô hình.

Trong nghiên cứu [32], tác giả đã lựa chọn ngẫu nhiên 40.000 mẫu hợp lệ từ tổng

số hơn 13 triệu mẫu cùng với 20.000 mẫu tấn công để tiến hành thí nghiệm. Cách tiếp cận này cho thấy rằng việc lựa chọn một tập mẫu phù hợp, dù nhỏ hơn rất nhiều so với tổng thể, vẫn có thể đảm bảo hiệu quả và tính đại diện cho quá trình huấn luyện mô hình, đồng thời tiết kiệm đáng kể tài nguyên tính toán.

Để xử lý hiệu quả hai tập dữ liệu lớn (1)CSE-CIC-IDS-2018 và (2)CIC-IoT-2023 có kích thước lớn, tôi áp dụng kỹ thuật Random Under-Sampling (RUS) (*được trình bày chi tiết ở phần 3.3.7*) sau các bước tiền xử lý và trước khi chia dữ liệu. Kỹ thuật này giảm số lượng mẫu của các lớp đa số xuống còn 150.000, trong khi giữ nguyên số lượng mẫu của các lớp thiểu số nhằm duy trì đặc điểm phân bố dữ liệu. Sau đó, dữ liệu được chia thành tập huấn luyện và tập kiểm tra theo tỷ lệ 80%-20%. Quy trình này đảm bảo sự cân bằng giữa các lớp và tối ưu hóa hiệu quả huấn luyện mô hình, đồng thời giảm yêu cầu về tài nguyên tính toán.

3.3.7. Data Balancing (Cân bằng dữ liệu)

Cân bằng dữ liệu là bước tiền xử lý dữ liệu quan trọng nhằm xử lý các tập dữ liệu mất cân bằng, khi mà một hoặc một số lớp có số lượng mẫu vượt trội so với các lớp khác, khiến mô hình có xu hướng thiên lệch về lớp chiếm đa số và bỏ qua lớp thiểu số, dẫn đến hiệu suất phát hiện kém đối với các mẫu quan trọng nhưng hiếm gặp.

Trong các tập dữ liệu về lưu lượng mạng hay trong thực tế, lưu lượng tấn công mạng thường chiếm tỷ lệ rất nhỏ so với lưu lượng bình thường. Sự mất cân bằng này khiến mô hình học sâu có xu hướng thiên lệch về lớp chiếm đa số, từ đó làm giảm độ chính xác trong việc phát hiện các loại tấn công hiếm gặp [33]. Vì vậy, việc áp dụng các kỹ thuật cân bằng dữ liệu là cần thiết để cải thiện khả năng nhận diện và nâng cao hiệu suất của mô hình.

Để xử lý dữ liệu mất cân bằng, các kỹ thuật lấy mẫu (Data Sampling) được sử dụng nhằm điều chỉnh phân phối giữa các lớp, giúp cân bằng số lượng mẫu và cải thiện hiệu quả, độ chính xác của mô hình học máy. Quá trình này tạo ra phiên bản dữ liệu mới với tỷ lệ lớp hợp lý hơn, từ đó nâng cao khả năng học và khả năng phát hiện của mô hình [34]. Các phương pháp lấy mẫu dữ liệu thường được chia thành ba nhóm chính:

- Under-Sampling (Lấy mẫu giảm): Giảm số lượng mẫu của lớp đa số để cân bằng dữ liệu.
- Over-Sampling (Lấy mẫu tăng): Tăng số lượng mẫu của lớp thiểu số nhằm cân bằng phân phối.

- Kết hợp lấy mẫu tăng và giảm.

Một số kỹ thuật lấy mẫu (Data Sampling) phổ biến bao gồm:

- **Random Over-Sampling (ROS)**: Tăng số lượng mẫu của lớp thiểu số bằng cách sao chép ngẫu nhiên các mẫu hiện có. Phương pháp này giúp cân bằng dữ liệu nhanh chóng nhưng dễ dẫn đến overfitting.

$$\text{While } |D_{\text{minor}}| < N_{\text{target}}: D_{\text{minor}} += \text{Sample}(D_{\text{minor}}, 1)$$

- **Synthetic Minority Over-sampling Technique (SMOTE)**: phương pháp này giúp giảm nguy cơ trùng lặp và cải thiện khả năng học của mô hình so với ROS. SMOTE tạo ra các mẫu tổng hợp mới cho lớp thiểu số bằng cách nội suy tuyến tính giữa một mẫu s và một trong các láng giềng gần nhất s^R .

$$n = s + d \cdot (s^R - s), 0 \leq d \leq 1$$

Trong đó: s : vector đặc trưng của một mẫu thuộc lớp thiểu số.

s^R : một mẫu khác thuộc k-láng giềng gần nhất của s .

d : số ngẫu nhiên trong khoảng [0,1].

- **Random Under-Sampling (RUS)**: Giảm số lượng mẫu của lớp chiếm đa số bằng cách chọn ngẫu nhiên một tập con. Phương pháp này đơn giản nhưng có thể làm mất đi thông tin quan trọng.

$$D_{\text{major}'} = \text{Sample}(D_{\text{major}}, N_{\text{target}})$$

$$D_{\text{balanced}} = D_{\text{major}'} \cup D_{\text{minor}}$$

Trong đó: D_{major} : tập dữ liệu của lớp đa số.

D_{minor} : tập dữ liệu của lớp thiểu số.

N_{target} : số lượng mẫu mục tiêu.

- **Tomek Link**: là một kỹ thuật undersampling được đề xuất bởi Tomek (1976), nhằm loại bỏ các mẫu nằm sát ranh giới phân lớp và thuộc các lớp khác nhau. Kỹ thuật này giúp làm sạch biên phân lớp và giảm nhiễu, nhưng không tăng số lượng mẫu cho lớp thiểu số, do đó thường được kết hợp với các kỹ thuật oversampling.

Một cặp (x, y) được gọi là **Tomek Link** nếu:

$$\text{NN}(x) = y \text{ và } \text{NN}(y) = x$$

Trong đó : x và y : là hai mẫu dữ liệu thuộc hai lớp khác nhau.

$d(x, y)$: là hàm đo khoảng cách (thường dùng Euclidean).

$NN(x)$: là láng giềng gần nhất của x .

Điều kiện để (x,y) được xem là Tomek Link nếu thỏa:

$$d(x,y) < d(x,i) \text{ hoặc } d(x,y) < d(y,i), \forall i$$

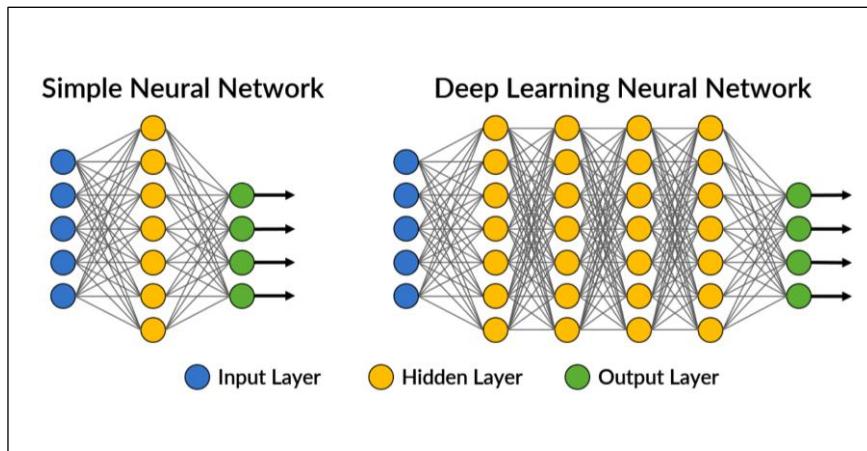
Các kỹ thuật cân bằng dữ liệu đơn lẻ như đã trình bày ở trên, mặc dù mang lại hiệu quả nhất định đối với các tập dữ liệu nhỏ, nhưng khi áp dụng độc lập vẫn còn hạn chế trong việc xử lý những tập dữ liệu phức tạp hoặc chứa nhiều nhiễu, khiến hiệu quả chưa đạt mức tối ưu. Trong đó, kỹ thuật SMOTE là một trong những phương pháp phổ biến nhất nhờ khả năng tạo mẫu tổng hợp cho lớp thiểu số. Tuy nhiên, khi sử dụng riêng lẻ, SMOTE vẫn có thể gây ra hiện tượng chồng lấn giữa các lớp hoặc không loại bỏ được các điểm nhiễu, từ đó ảnh hưởng đến chất lượng phân loại. Chính vì vậy, việc kết hợp SMOTE với các kỹ thuật làm sạch dữ liệu khác trở nên cần thiết để nâng cao hiệu quả cân bằng và cải thiện kết quả mô hình. Gần đây có nghiên cứu đã chỉ ra rằng việc kết hợp kỹ thuật SMOTE với các phương pháp làm sạch dữ liệu khác có thể cải thiện hiệu quả cân bằng dữ liệu và nâng cao chất lượng mô hình phân loại [35]. Một số phương pháp phổ biến bao gồm:

- **SMOTE-RUS** (Random Under-Sampling): Kết hợp SMOTE với giảm mẫu ngẫu nhiên ở lớp chiếm đa số để cân bằng giữa oversampling và undersampling.
- **SMOTE-ENN** (Edited Nearest Neighbors): Sau khi tạo mẫu nhân tạo bằng SMOTE, ENN được dùng để loại bỏ mẫu nhiễu dựa trên lân cận gần nhất (cả mẫu thiểu số và đa số) làm sạch dữ liệu cho mô hình học.
- **SMOTE-Tomek Links**: Kết hợp SMOTE với kỹ thuật Tomek Links nhằm loại bỏ các cặp điểm dữ liệu gần nhau nhưng thuộc hai lớp khác biệt. Khác với ENN, Tomek Links chủ yếu loại bỏ các điểm ở lớp đa số trong cặp, giúp làm sạch ranh giới phân lớp và giảm nhiễu mà vẫn giữ được nhiều mẫu thiểu số.

Trong số các phương pháp kết hợp cân bằng dữ liệu, tôi lựa chọn áp dụng kỹ thuật SMOTE + Tomek Links để xử lý tập huấn luyện trong luận văn này. Phương pháp này được lựa chọn bởi khả năng cân bằng dữ liệu hiệu quả và làm sạch vùng biên phân lớp, giúp cải thiện đáng kể độ chính xác của mô hình. Đồng thời, so với các phương pháp kết hợp khác, thời gian huấn luyện với **SMOTE + Tomek Links** nhanh hơn đáng kể so với **SMOTE + ENN**, và đạt hiệu quả phân loại tốt hơn so với sự kết hợp của **SMOTE + RUS** [35]. Nhờ vậy, phương pháp này tối ưu cả về hiệu suất mô hình lẫn chi phí tính toán trong quá trình xây dựng và huấn luyện.

3.4. Huấn luyện các mô hình (Models Training)

Trong các mô hình học sâu (Deep Learning), mạng nơ-ron thường được xây dựng từ nhiều lớp liên kết tuần tự, trong đó mỗi lớp đảm nhận một vai trò riêng trong việc trích xuất và biến đổi đặc trưng dữ liệu. Ngoài lớp đầu vào (input layer) và lớp đầu ra (output layer), một mô hình học sâu thường bao gồm một số lớp ẩn (hidden layers).



Việc huấn luyện mạng học sâu không chỉ đơn giản là tăng số lượng lớp nơ-ron. Trên thực tế, một kiến trúc mạng với số lớp ẩn hợp lý, kết hợp cùng các kỹ thuật như Dropout (vô hiệu hóa ngẫu nhiên một tỷ lệ nơ-ron trong quá trình huấn luyện nhằm giảm nguy cơ quá khớp) và Batch Normalization (chuẩn hóa đầu ra của mỗi lớp để ổn định và tăng tốc quá trình huấn luyện), có thể mang lại hiệu quả cao hơn, đồng thời giảm nguy cơ quá khớp (overfitting – mô hình học quá kỹ dữ liệu huấn luyện, dẫn đến giảm hiệu quả trên dữ liệu mới) và cải thiện tốc độ hội tụ. Hiện nay chưa tồn tại công thức chung xác định chính xác số lớp ẩn và số lượng nơ-ron tối ưu cho mọi bài toán. Nếu số lượng nơ-ron quá lớn, mô hình dễ bị quá khớp, làm suy giảm khả năng tổng quát hóa. Ngược lại, nếu mạng quá nhỏ, mô hình có thể gặp tình trạng thiếu khớp (underfitting – mô hình không học đủ các đặc trưng cần thiết từ dữ liệu), dẫn đến hiệu suất thấp trên cả dữ liệu huấn luyện và dữ liệu mới.

Trong luận văn này, mục tiêu không phải là tìm kiếm cấu hình tối ưu cho từng mô hình, mà là thử nghiệm nhiều cấu hình với số lượng lớp ẩn (hidden layers) và số lượng nơ-ron khác nhau để đánh giá và so sánh hiệu quả của các thuật toán học sâu khi áp dụng cho bài toán phát hiện xâm nhập. Số lượng nút nơ-ron sẽ tỉ lệ thuận với số lượng tham số của mô hình. Tốc độ học (learning rate) ảnh hưởng trực tiếp tới mức điều chỉnh trọng số trong quá trình học, từ đó tác động tới tốc độ hội tụ của mô hình, cũng như thời gian huấn luyện. Tốc độ học (Learning rate), hàm kích hoạt (Activation function) và bộ tối

uru (Optimizer) được lựa chọn phù hợp để đánh giá hiệu năng của từng thuật toán với các cấu hình lớp ẩn khác nhau. Các mô hình dự đoán được áp dụng trong đề tài này bao gồm CNN, LSTM, GRU, MLP, CNN-LSTM và CNN-MLP.

Các Thuật ngữ và khái niệm liên quan đến quá trình huấn luyện mô hình:

- **Tham số mô hình (Model parameters):** Là các giá trị được mô hình học trực tiếp từ dữ liệu trong quá trình huấn luyện, bao gồm trọng số và độ chệch (biases).
- **Siêu tham số (Hyperparameters):** Các giá trị được thiết lập trước khi huấn luyện, không học trực tiếp từ dữ liệu,
 - + **Tốc độ học (Learning rate):** Điều chỉnh mức độ thay đổi trọng số sau mỗi lần cập nhật.
 - + **Số vòng lặp huấn luyện (Epoch):** Số lần toàn bộ tập dữ liệu huấn luyện được đưa qua mô hình.
 - + **Kích thước lô (Batch size):** Số lượng mẫu dữ liệu được sử dụng trong mỗi lần cập nhật trọng số.
 - + **Số lớp ẩn (Hidden layer):** Các lớp nằm giữa đầu vào và đầu ra, chịu trách nhiệm trích xuất và biến đổi đặc trưng. Số lớp (layer) và số lượng nơ-ron trong mỗi lớp ảnh hưởng trực tiếp đến khả năng học và mức độ phức tạp của mô hình.
- **Hàm kích hoạt (Activation function)** Hàm phi tuyến tại mỗi nơ-ron, giúp mô hình học các quan hệ phức tạp trong dữ liệu.
 - + **Sigmoid :** Đầu ra trong khoảng (0,1). Thường dùng cho bài toán phân loại nhị phân hoặc trong các cổng (gate) của LSTM.

$$f(x) = \frac{1}{1 + e^{-x}}$$

- + **Tanh (Hyperbolic Tangent) :** Đầu ra trong khoảng (-1,1). Thường dùng trong LSTM/GRU để biểu diễn trạng thái ẩn.

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

- + **ReLU (Rectified Linear Unit)** Đơn giản, tính toán nhanh, giảm vanishing gradient. Được dùng phổ biến trong CNN, MLP.

$$f(x) = \max(0, x)$$

- + **Softmax** :Chuyển đổi vector thành phân phối xác suất.Thường dùng ở lớp đầu ra trong bài toán phân loại đa lớp.

$$f(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

- **Hàm mất mát (Loss function)**: Thước đo sai lệch giữa dự đoán và giá trị thực tế, làm cơ sở để tối ưu hóa mô hình.
- **Bộ tối ưu (Optimizer)**: Thuật toán cập nhật trọng số dựa trên đạo hàm của hàm mất mát, tối ưu hóa hiệu suất mô hình.
 - + **Momentum** (quán tính): Bổ sung “quán tính” cho quá trình cập nhật trọng số bằng cách kết hợp gradient hiện tại với gradient trước đó.Giúp mô hình giảm dao động, di chuyển mượt hơn trên bề mặt hàm mất mát và hội tụ nhanh hơn.
 - + **RMSProp** (tốc độ học thích nghi): Điều chỉnh tốc độ học bằng cách chia gradient cho căn bậc hai của trung bình bình phương gradient gần đây.Giúp cân bằng bước nhảy của các tham số, đặc biệt hiệu quả với dữ liệu không đồng nhất.
 - + **Adam (Adaptive Moment Estimation)**: Kết hợp ưu điểm của Momentum và RMSProp (tốc độ học thích nghi). Được sử dụng phổ biến nhờ khả năng hội tụ nhanh, ổn định và thường là lựa chọn mặc định trong nhiều mô hình học sâu.
- **Regularization**: Trong quá trình huấn luyện các mô hình học sâu, đặc biệt là khi số lượng tham số rất lớn, mô hình dễ gặp phải hiện tượng overfitting – tức là mô hình học quá chi tiết từ dữ liệu huấn luyện và không tổng quát tốt trên dữ liệu mới. Regularization là tập hợp các kỹ thuật giúp hạn chế overfitting, tăng khả năng khai quát hóa và độ tin cậy của mô hình. Một số kỹ thuật phổ biến gồm:
 - + **Dropout**: Kỹ thuật vô hiệu hóa ngẫu nhiên một số nơ-ron trong quá trình huấn luyện, giúp giảm sự phụ thuộc quá mức vào một số đặc trưng cụ thể và tăng tính đa dạng của mạng nơ-ron.
 - + **Batch Normalization**: Kỹ thuật chuẩn hóa đầu ra của một lớp bằng cách điều chỉnh giá trị trung bình và phương sai, giúp ổn định quá trình huấn luyện, tăng tốc độ hội tụ và giảm nguy cơ overfitting.
- **Learning curve**: Biểu đồ thể hiện sự thay đổi của hàm mất mát hoặc độ chính xác theo số epoch. Đó cũng chính là biểu đồ thể hiện lịch sử huấn luyện của mô hình

3.4.1. Thông số cấu hình ban đầu của các kỹ thuật học sâu

Quá trình thiết kế mô hình sử dụng các siêu tham số chính sau:

- Hidden Layer (Số lớp ẩn) : [1, 2, 3, 4, 5]												
- Số lượng nơ-ron trong mỗi lớp ẩn : Tổng cộng 256 nơ-ron, phân bổ như bảng dưới:												
<table border="1"><thead><tr><th>Hidden Layer</th><th>Cấu hình số lượng nơ-ron</th></tr></thead><tbody><tr><td>1 lớp</td><td>[256]</td></tr><tr><td>2 lớp</td><td>[128, 128]</td></tr><tr><td>3 lớp</td><td>[64, 64, 128]</td></tr><tr><td>4 lớp</td><td>[32, 32, 64, 128]</td></tr><tr><td>5 lớp</td><td>[32, 32, 64, 64]</td></tr></tbody></table>	Hidden Layer	Cấu hình số lượng nơ-ron	1 lớp	[256]	2 lớp	[128, 128]	3 lớp	[64, 64, 128]	4 lớp	[32, 32, 64, 128]	5 lớp	[32, 32, 64, 64]
Hidden Layer	Cấu hình số lượng nơ-ron											
1 lớp	[256]											
2 lớp	[128, 128]											
3 lớp	[64, 64, 128]											
4 lớp	[32, 32, 64, 128]											
5 lớp	[32, 32, 64, 64]											
- Learning rate (Tốc độ học) : 0,001												
- Optimizer (Bộ tối ưu) : Adam												
- Batch size : 256												
- Epoch (số lần huấn luyện) : 100												
- Regularization (chuẩn hoá) + Dropout : 0,2												
- + BatchNormalization												
- Loss function (Hàm mất mát) : categorical_crossentropy (Với bài toán phân loại đa lớp (multi-class classification))												

Các cấu hình mô hình được thiết kế dựa trên những **layer cơ bản và đặc trưng nhất** của từng thuật toán, với mục tiêu đánh giá hiệu quả cốt lõi của từng kiến trúc mạng.

Cụ thể:

- **CNN**: sử dụng lớp **Conv1D** để trích xuất đặc trưng cục bộ.
- **LSTM, GRU**: triển khai trực tiếp thông qua các lớp **LSTM** và **GRU**, nhằm mô hình hóa quan hệ tuần tự.
- **MLP**: được hình thành từ các lớp **Dense**, đại diện cho cấu trúc kết nối đầy đủ giữa các nơ-ron.

3.4.2. Cấu trúc các mô hình áp dụng

Quá trình thực nghiệm được tiến hành trên 6 thuật toán khác nhau, với số lớp ẩn (hidden layer) thay đổi từ 1 đến 5, nhằm đánh giá tác động của độ sâu mạng đến hiệu suất mô hình. Mỗi thuật toán được huấn luyện 5 lần tương ứng với số lớp ẩn từ 1 đến 5, do đó cần thực hiện tổng cộng 30 lượt huấn luyện cho cả 6 thuật toán và quá trình này được lặp lại với 3 tập dữ liệu, khiến tổng số lần thực nghiệm tăng lên gấp ba là 90 lượt huấn luyện. Do đó số lượng cấu hình thử nghiệm khá lớn, báo cáo này chỉ trình bày cấu hình đối với cấu hình 5 lớp ẩn (hidden layer) cho từng thuật toán (CNN, MLP, LSTM, GRU, CNN-MLP, CNN-LSTM) nhằm đảm bảo tính gọn gàng và tập trung vào các so sánh quan trọng.

Các hình minh họa cấu trúc mạng của từng mô hình với 5 lớp ẩn được trình bày ở phần này được tạo ra từ lệnh `model.summary()` của thư viện Keras (TensorFlow), các hình này thể hiện rõ số lượng tham số, kích thước đầu vào (Input shape), kích thước đầu ra (Output shape) của từng lớp, cùng các đặc điểm kiến trúc của mỗi cấu hình. Cấu trúc mạng này được thực hiện trên tập dữ liệu (1) CSE-CIC-IDS-2018, các tập dữ liệu khác sẽ có “Output Shape” và “Param #” khác nhau và có “Layer(type)” tương tự.

3.4.2.1 Cấu hình kiến trúc CNN với 5 lớp ẩn

Layer (type)	Output Shape	Param #
conv1d_1 (Conv1D)	(None, 51, 32)	96
dropout_1 (Dropout)	(None, 51, 32)	0
conv1d_2 (Conv1D)	(None, 51, 32)	2,080
dropout_2 (Dropout)	(None, 51, 32)	0
conv1d_3 (Conv1D)	(None, 51, 64)	4,160
dropout_3 (Dropout)	(None, 51, 64)	0
conv1d_4 (Conv1D)	(None, 51, 64)	8,256
dropout_4 (Dropout)	(None, 51, 64)	0
conv1d_5 (Conv1D)	(None, 51, 64)	8,256
dropout_5 (Dropout)	(None, 51, 64)	0
batch_norm (BatchNormalization)	(None, 51, 64)	256
flatten (Flatten)	(None, 3264)	0
dense_output (Dense)	(None, 6)	19,590

Hình 17: Cấu hình kiến trúc CNN với 5 lớp ẩn.

3.4.2.2 Cấu hình kiến trúc LSTM với 5 lớp ẩn

Layer (type)	Output Shape	Param #
lstm_1 (LSTM)	(None, 51, 32)	4,352
dropout_1 (Dropout)	(None, 51, 32)	0
lstm_2 (LSTM)	(None, 51, 32)	8,320
dropout_2 (Dropout)	(None, 51, 32)	0
lstm_3 (LSTM)	(None, 51, 64)	24,832
dropout_3 (Dropout)	(None, 51, 64)	0
lstm_4 (LSTM)	(None, 51, 64)	33,024
dropout_4 (Dropout)	(None, 51, 64)	0
lstm_5 (LSTM)	(None, 64)	33,024
dropout_5 (Dropout)	(None, 64)	0
batch_normalization (BatchNormalization)	(None, 64)	256
dense_output (Dense)	(None, 6)	390

Hình 18: Cấu hình kiến trúc LSTM với 5 lớp ẩn.

3.4.2.3 Cấu hình kiến trúc GRU với 5 lớp ẩn

Layer (type)	Output Shape	Param #
gru_1 (GRU)	(None, 51, 32)	3,360
dropout_1 (Dropout)	(None, 51, 32)	0
gru_2 (GRU)	(None, 51, 32)	6,336
dropout_2 (Dropout)	(None, 51, 32)	0
gru_3 (GRU)	(None, 51, 64)	18,816
dropout_3 (Dropout)	(None, 51, 64)	0
gru_4 (GRU)	(None, 51, 64)	24,960
dropout_4 (Dropout)	(None, 51, 64)	0
gru_5 (GRU)	(None, 64)	24,960
dropout_5 (Dropout)	(None, 64)	0
batch_normalization (BatchNormalization)	(None, 64)	256
dense_output (Dense)	(None, 6)	390

Hình 19: Cấu hình kiến trúc GRU với 5 lớp ẩn.

3.4.2.4 Cấu hình kiến trúc MLP với 5 lớp ẩn

Layer (type)	Output Shape	Param #
dense_DNN_1 (Dense)	(None, 32)	1,664
dropout_1 (Dropout)	(None, 32)	0
dense_2 (Dense)	(None, 32)	1,056
dropout_2 (Dropout)	(None, 32)	0
dense_3 (Dense)	(None, 64)	2,112
dropout_3 (Dropout)	(None, 64)	0
dense_4 (Dense)	(None, 64)	4,160
dropout_4 (Dropout)	(None, 64)	0
dense_5 (Dense)	(None, 64)	4,160
dropout_5 (Dropout)	(None, 64)	0
batch_normalization (BatchNormalization)	(None, 64)	256
output (Dense)	(None, 6)	390

Hình 20: Cấu hình kiến trúc MLP với 5 lớp ẩn.

3.4.2.5 Cấu hình kiến trúc CNN-LSTM với 5 lớp ẩn

Layer (type)	Output Shape	Param #
conv1d_1 (Conv1D)	(None, 50, 32)	96
conv1d_2 (Conv1D)	(None, 49, 64)	4,160
lstm_1 (LSTM)	(None, 49, 32)	12,416
dropout_1 (Dropout)	(None, 49, 32)	0
lstm_2 (LSTM)	(None, 49, 32)	8,320
dropout_2 (Dropout)	(None, 49, 32)	0
lstm_3 (LSTM)	(None, 49, 64)	24,832
dropout_3 (Dropout)	(None, 49, 64)	0
lstm_4 (LSTM)	(None, 49, 64)	33,024
dropout_4 (Dropout)	(None, 49, 64)	0
lstm_5 (LSTM)	(None, 64)	33,024
dropout_5 (Dropout)	(None, 64)	0
batch_normalization (BatchNormalization)	(None, 64)	256
dense_output (Dense)	(None, 6)	390

Hình 21: Cấu hình kiến trúc CNN-LSTM với 5 lớp ẩn.

3.4.2.6 Cấu hình kiến trúc CNN-MLP với 5 lớp ẩn

Layer (type)	Output Shape	Param #
conv1d_1 (Conv1D)	(None, 50, 32)	96
conv1d_2 (Conv1D)	(None, 49, 64)	4,160
flatten (Flatten)	(None, 3136)	0
dense_1 (Dense)	(None, 32)	100,384
dropout_1 (Dropout)	(None, 32)	0
dense_2 (Dense)	(None, 32)	1,056
dropout_2 (Dropout)	(None, 32)	0
dense_3 (Dense)	(None, 64)	2,112
dropout_3 (Dropout)	(None, 64)	0
dense_4 (Dense)	(None, 64)	4,160
dropout_4 (Dropout)	(None, 64)	0
dense_5 (Dense)	(None, 64)	4,160
dropout_5 (Dropout)	(None, 64)	0
dense_output (Dense)	(None, 6)	390

Hình 22: Cấu hình kiến trúc CNN-MLP với 5 lớp ẩn.

3.4.3. Các Tiêu chí đánh giá mô hình

- **Accuracy (Độ chính xác)** là một trong những chỉ số phổ biến nhất để đánh giá hiệu suất của mô hình học máy, đặc biệt trong các bài toán phân loại. Nó được định nghĩa là tỷ lệ giữa số lượng dự đoán đúng (bao gồm cả lớp dương và lớp âm) trên tổng số mẫu trong tập dữ liệu.

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

Trong đó:

- + **TP (True Positive)**: Số mẫu dương được dự đoán đúng.
- + **TN (True Negative)**: Số mẫu âm được dự đoán đúng.
- + **FP (False Positive)**: Số mẫu âm bị dự đoán sai thành dương.
- + **FN (False Negative)**: Số mẫu dương bị dự đoán sai thành âm.
- **Confusion Matrix (Ma trận nhầm lẫn)** là một công cụ đánh giá hiệu quả phân loại phổ biến, đặc biệt hữu ích trong các bài toán phát hiện bất thường (anomaly detection) dựa trên học máy, học sâu. Trong khi các chỉ số tổng hợp như độ chính xác (Accuracy) chỉ phản ánh tỷ lệ tổng thể các mẫu được phân loại đúng, chúng không cung cấp thông tin chi tiết về hiệu suất phân loại đối với từng lớp cụ thể. Tùy theo bài toán phân loại ta có phân loại đa lớp (multi-class classification) hoặc phân loại nhị phân (binary classification). Ma trận nhầm lẫn có thể có kích thước $N \times N$ (với N là số lượng lớp mà mô hình cần phân loại). Đối với bài toán phân loại nhị phân (binary classification), ma trận nhầm lẫn có dạng 2×2 được thể hiện ở bảng dưới đây:

	Nhận dự đoán: dương	Nhận dự đoán: âm
Nhận thực tế: dương	TP	FN
Nhận thực tế: âm	FP	TN

Mỗi hàng trong ma trận biểu thị nhãn thực tế (actual labels), trong khi mỗi cột tương ứng với nhãn dự đoán (predicted labels). Việc sử dụng ma trận nhầm lẫn cho phép phân tích chi tiết các loại lỗi mà mô hình gặp phải, từ đó hỗ trợ điều chỉnh và tối ưu hóa hiệu năng.

- **Precision (Độ chính xác dương)** Độ đo precision là độ đo tính tỉ lệ dương tính thật (TP) trên tổng số dự đoán là dương tính bao gồm dương tính thật (TP) và dương tính giả (FP). Công thức tính precision là:

$$Precision = \frac{TP}{TP + FP}$$

- **Recall (Khả năng bao phủ):** Độ đo xác định tỉ lệ dương tính thật (TP) trên tổng số các mẫu dương tính thật sự trong tập dữ liệu bao gồm dương tính thật (TP) và âm tính giả (FN). Công thức tính recall là:

$$Recall = \frac{TP}{TP + FN}$$

- **F1-score:** Điểm F1 được mô tả là trung bình hài hòa giữa Presicion và Recall khi F1-score trở thành chỉ số đánh giá hiệu suất mô hình tổng thể mang lại độ tinh cậy cao. Công thức tính F1-score là:

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

- **Mức độ sử dụng tài nguyên:** Tiêu chí này phản ánh hiệu quả khai thác phần cứng của mô hình trong quá trình huấn luyện và suy luận. Việc theo dõi giúp đánh giá khả năng tận dụng CPU, GPU và bộ nhớ, từ đó tối ưu hóa hiệu suất và chi phí vận hành. Ngoài ra, phân tích mức độ sử dụng tài nguyên còn hỗ trợ xác định tính khả thi khi triển khai mô hình trên các hệ thống thực tế, đặc biệt trong môi trường có giới hạn về phần cứng hoặc yêu cầu tiết kiệm năng lượng. Trong nghiên cứu này:

- **CPU :** Được theo dõi bằng thư viện Psutil, bao gồm mức độ sử dụng CPU và dung lượng RAM.
- **GPU:** Được theo dõi bằng thư viện Nvsmi, ghi nhận mức sử dụng GPU, dung lượng VRAM, nhiệt độ vận hành và công suất tiêu thụ.

3.5. Các công cụ hỗ trợ học sâu

- **Python** là ngôn ngữ lập trình phổ biến nhất trong lĩnh vực học máy và học sâu nhờ cú pháp đơn giản, cộng đồng phát triển mạnh mẽ và khả năng tích hợp tốt với các thư viện khoa học dữ liệu, xử lý số liệu và trí tuệ nhân tạo.
- **Pandas** là thư viện mã nguồn mở giúp thao tác, tổ chức và xử lý dữ liệu dạng bảng (DataFrame), hỗ trợ rất mạnh cho khâu tiền xử lý dữ liệu trước khi đưa vào mô hình học sâu. Pandas dễ dàng xử lý dữ liệu bị thiếu, tổng hợp, biến đổi và chuẩn hóa dữ liệu.
- **TensorFlow** là nền tảng học máy, học sâu mã nguồn mở do Google phát triển, hỗ trợ xây dựng, huấn luyện và triển khai các mô hình deep learning với hiệu suất cao, linh hoạt trên cả CPU, GPU và TPU. TensorFlow tích hợp nhiều API cho cả người mới và chuyên gia, hỗ trợ hàng loạt ứng dụng thực tiễn từ xử lý ảnh, ngôn ngữ tự nhiên đến tự động hóa.
- **Keras** là thư viện API cấp cao, giúp việc xây dựng, huấn luyện mô hình mạng nơ-ron (Multi-layer Perceptrons) trở nên trực quan, đơn giản hơn rất nhiều. Keras thường được xây dựng trên nền tảng TensorFlow, Theano, hoặc CNTK, với giao diện dễ dùng, hỗ trợ mạnh các mô hình học sâu hiện đại (CNN, RNN...).
- **Scikit-learn** là thư viện machine learning mã nguồn mở nổi tiếng trên Python, hỗ trợ phong phú các thuật toán cơ bản (phân loại, hồi quy, phân cụm, tiền xử lý dữ liệu...). Scikit-learn thường dùng để xử lý dữ liệu, chọn mô hình, đánh giá hiệu năng và xây dựng các pipeline kết hợp với các thư viện học sâu khác như TensorFlow/Keras.

CHƯƠNG 4: KẾT QUẢ THỰC NGHIỆM

4.1. Thiết lập môi trường

Các thực nghiệm được triển khai trên nền tảng Kaggle, một môi trường tính toán đám mây miễn phí, phổ biến trong nghiên cứu học máy. Môi trường này cung cấp cấu hình phần cứng gồm CPU Intel(R) Xeon(R) (2 core, 4 threads, 2 GHz), bộ nhớ RAM 30 GB và GPU NVIDIA Tesla P100 với dung lượng 16 GB VRAM. Hệ điều hành sử dụng là Linux, cho phép tối ưu hóa quá trình quản lý tài nguyên và cài đặt các thư viện học máy. Kaggle hỗ trợ truy cập dễ dàng, thuận tiện trong việc triển khai mô hình, lưu trữ kết quả và chia sẻ với cộng đồng.

4.2. Thiết kế và triển khai

4.2.1. Thiết kế

- **Quá trình diễn ra hai quy trình :**

- **Tiền xử lý dữ liệu :** Các hàm được thiết kế để đọc dữ liệu từ các tập dữ liệu đầu vào, thực hiện các bước tiền xử lý đã được trình bày tại *Mục 3.3*, sau đó áp dụng phương pháp cân bằng dữ liệu. Kết quả cuối cùng của bước tiền xử lý là hai tệp:
 - *train_balanced.parquet*
 - *test.parquet*

Hai tệp này được sử dụng trực tiếp trong giai đoạn huấn luyện mô hình, đảm bảo không cần lặp lại quá trình tiền xử lý và duy trì tính nhất quán của dữ liệu.

- **Huấn luyện và đánh giá mô hình:** Sử dụng kết quả của bước tiền xử lý, xây dựng các mô hình cho các thuật toán học sâu (CNN, LSTM, GRU, MLP, CNN-LSTM, CNN-MLP), đồng thời thiết kế các hàm phục vụ việc đánh giá hiệu suất và lưu trữ kết quả huấn luyện.

4.2.2. Triển khai

Bảng 8: Bảng tóm tắt các hàm và ý nghĩa trong bước tiền xử lý.

Tên hàm	Chức năng chính
<code>add_library()</code>	-Nạp các thư viện cần thiết
<code>processing_data()</code>	-Đọc dữ liệu thô, thực hiện các bước tiền xử lý như làm sạch, chuẩn hóa, mã hóa và tách tập dữ liệu
<code>balance_data()</code>	-Áp dụng kết hợp SMOTE và Tomek Link để cân bằng dữ liệu, giảm nhiễu và xuất ra hai tệp: +train_balanced.parquet +test.parquet.
<code>save_result()</code>	-Lưu trữ các tệp dữ liệu đã xử lý để phục vụ giai đoạn huấn luyện mô hình.

Bảng 9: Bảng tóm tắt các hàm và ý nghĩa trong bước huấn luyện và đánh giá mô hình.

Tên hàm	Chức năng chính
<code>add_library()</code>	-Khai báo và nạp các thư viện cần thiết cho quá trình huấn luyện và đánh giá mô hình.
<code>load_data()</code>	-Đọc dữ liệu đã được xử lý huấn luyện train_balanced.parquet và test.parquet.
<code>pre_process()</code>	-Tiền xử lý dữ liệu nhằm đảm bảo định dạng và cấu trúc phù hợp với yêu cầu đầu vào của các mô hình học sâu.
<code>built_models()</code>	-Xây dựng và biên dịch các kiến trúc mô hình học sâu gồm CNN, LSTM, GRU, MLP, CNN-LSTM và CNN-MLP .
<code>monitor_CPU_GPU()</code>	-Giám sát và ghi nhận mức sử dụng CPU và GPU trong quá trình huấn luyện.
<code>train_model()</code>	-Huấn luyện mô hình trên tập train_balanced.parquet với các siêu tham số đã cài đặt chung ban đầu.
<code>evaluate_model()</code>	-Đánh giá hiệu suất mô hình trên tập test.parquet
<code>save_result()</code>	-Lưu trữ kết quả huấn luyện và đánh giá gồm số liệu, đồ thị trực quan và báo cáo.

4.3. Kết quả thực nghiệm

Kết quả thực nghiệm được trình bày qua ba kịch bản sau. Trong cả ba kịch bản, các mô hình đều được huấn luyện với cùng cấu hình tham số, nhằm đảm bảo tính công bằng trong so sánh và đánh giá:

- Kịch bản 1: Huấn luyện các mô hình sử dụng tập dữ liệu CSE-CIC-IDS-2018.
- Kịch bản 2: Huấn luyện các mô hình sử dụng tập dữ liệu CIC-IoT-2023.
- Kịch bản 3: Huấn luyện các mô hình sử dụng tập dữ liệu ICS-Flow.

Mỗi một kịch bản sẽ được trình bày các phần sau đây:

1. Kết quả tiền xử lý.

2. Kết quả huấn luyện:

- + **Đánh giá tổng quát:** Trình bày kết quả tổng quan, bao gồm độ chính xác (Accuracy), thời gian huấn luyện (Training time) và thời gian suy luận của mô hình (Inference time) của kịch bản trên tất cả các cấu hình.
- + **Đánh giá chi tiết:** Trình bày các chỉ số chi tiết như Accuracy, Precision, Recall, F1-score, hiệu suất của các mô hình trên cấu hình 5 lớp ẩn, lịch sử huấn luyện (training history), ma trận nhầm lẫn (confusion matrix) và mức độ sử dụng tài nguyên.

(Do số lượng cấu hình quá nhiều như tôi đã trình bày ở mục 3.4.2 trước đó, để thuận tiện cho việc phân tích các chỉ số khác, nên phần này chỉ trình bày kết quả chi tiết đối với **cấu hình 5 lớp ẩn** cho các mô hình học sâu)

Các mô hình học sâu được sử dụng bao gồm: CNN, LSTM, GRU, MLP, CNN-LSTM và CNN-MLP. Để thuận tiện cho việc so sánh và phân tích, các mô hình này được phân thành bốn nhóm dựa trên đặc trưng kiến trúc mạng:

- Nhóm 1 – CNN: mạng nơ-ron tích chập, chuyên xử lý dữ liệu có cấu trúc không gian và trích xuất đặc trưng không gian.
- Nhóm 2 – LSTM và GRU: thuộc nhóm mạng nơ-ron hồi quy RNN (Recurrent Neural Network) chuyên xử lý dữ liệu chuỗi theo thời gian.
- Nhóm 3 – MLP: thuộc nhóm mạng truyền thẳng nhiều lớp, xử lý dữ liệu dạng vector đặc trưng.
- Nhóm 4 – CNN-LSTM và CNN-MLP: mô hình kết hợp, kết hợp ưu điểm của nhiều kiến trúc để nâng cao khả năng trích xuất và học đặc trưng.

4.3.1. Kết quả trên kịch bản 1

4.3.1.1 Kết quả tiền xử lý

- Tập dữ liệu (1) : CSE-CIC-IDS-2018.
- Làm sạch dữ liệu :
 - + Các đặc trưng không đóng góp cho việc phân loại được loại bỏ là các thông tin định danh: Timestamp, Flow ID, Src IP, Src Port, Dst IP, Dst Port.
 - + Các đặc trưng chỉ có một giá trị duy nhất: Bwd PSH Flags, Bwd URG Flags, Fwd Byts/b Avg, Fwd Pkts/b Avg, Fwd Blk Rate Avg, Bwd Byts/b Avg, Bwd Pkts/b Avg, Bwd Blk Rate Avg.
 - + Loại bỏ các dòng có các giá trị NaN hoặc Inf.
 - + Loại bỏ được nhiều dòng trùng nhau ở những dòng có nhãn Benign, DDoS.
- Thực hiện phương pháp lấy mẫu (Data Sampling) Random Under-Sampling (RUS) (*đã trình bày ở mục 3.3.6*).
- Mã hóa các nhãn cho tập dữ liệu (1) với bài toán phân loại đa lớp và số lượng mẫu cho mỗi nhãn sau khi làm sạch dữ liệu :

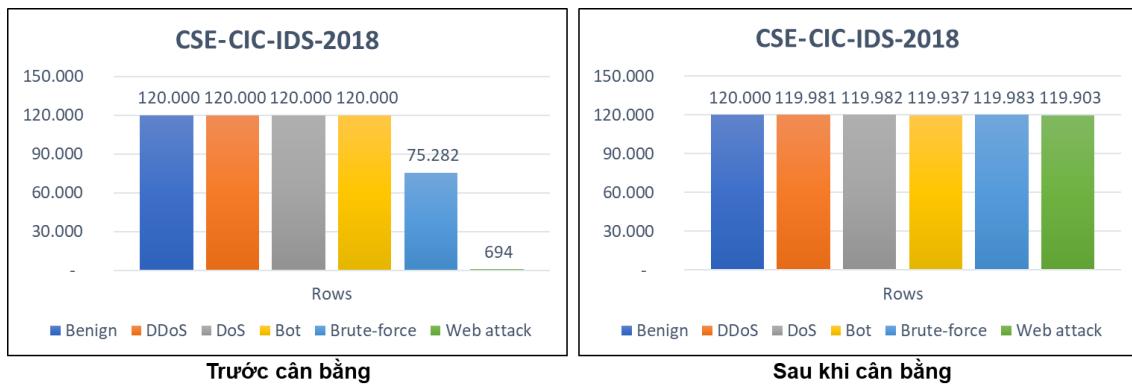
Label_encoding	Attack Category	Rows
0	Benign	150.000
1	DDoS	150.000
2	DoS	150.000
3	Bot	144.535
4	Brute-force	94.094
5	Web attack	867

- Các cột đặc trưng được thuật toán Random Forest lựa chọn là những cột đặc trưng có độ quan trọng (feature importance) > 0,01. Trong tập dữ liệu (1), sau khi áp dụng ngưỡng này, ta thu được tổng cộng 53 cột, bao gồm 51 cột đặc trưng vào và 2 cột đặc trưng mục tiêu “Label” và “Label_encode”, được thể hiện ở hình bên dưới :

```
['Flow Duration', 'Tot Fwd Pkts', 'TotLen Fwd Pkts', 'Fwd Pkt Len Max', 'Fwd Pkt Len Min', 'Fwd Pkt Len Mean',  
 'Fwd Pkt Len Std', 'Bwd Pkt Len Max', 'Bwd Pkt Len Min', 'Bwd Pkt Len Mean', 'Bwd Pkt Len Std', 'Flow Bytes/s',  
 'Flow Pkts/s', 'Flow IAT Mean', 'Flow IAT Std', 'Flow IAT Max', 'Flow IAT Min', 'Fwd IAT Tot', 'Fwd IAT Mean',  
 'Fwd IAT Std', 'Fwd IAT Max', 'Fwd IAT Min', 'Bwd IAT Tot', 'Bwd IAT Mean', 'Bwd IAT Std', 'Bwd IAT Max', 'Bwd IAT Min',  
 'Fwd Header Len', 'Bwd Header Len', 'Fwd Pkts/s', 'Bwd Pkts/s', 'Pkt Len Min', 'Pkt Len Max', 'Pkt Len Mean', 'Pkt Len Std',  
 'Pkt Len Var', 'FIN Flag Cnt', 'PSH Flag Cnt', 'ACK Flag Cnt', 'Pkt Size Avg', 'Subflow Bwd Byts', 'Init Fwd Win Byts', 'Init Bwd Win Byts',  
 'Fwd Act Data Pkts', 'Fwd Seg Size Min', 'Active Mean', 'Active Max', 'Active Min', 'Idle Mean', 'Idle Max', 'Idle Min', 'Label_encode',  
 'Label'],
```

- Sau khi chia tập dữ liệu (1) thành 2 phần tập huấn luyện (80%) và tập kiểm tra (20%).
- Ta đệm tập huấn luyện cân bằng dữ liệu với kỹ thuật **SMOTE+Tomek-Link**.

- Kết quả trước và sau khi cân bằng của **tập huấn luyện (training set)** trên tập dữ liệu (1) được thể hiện ở **Hình 23**.



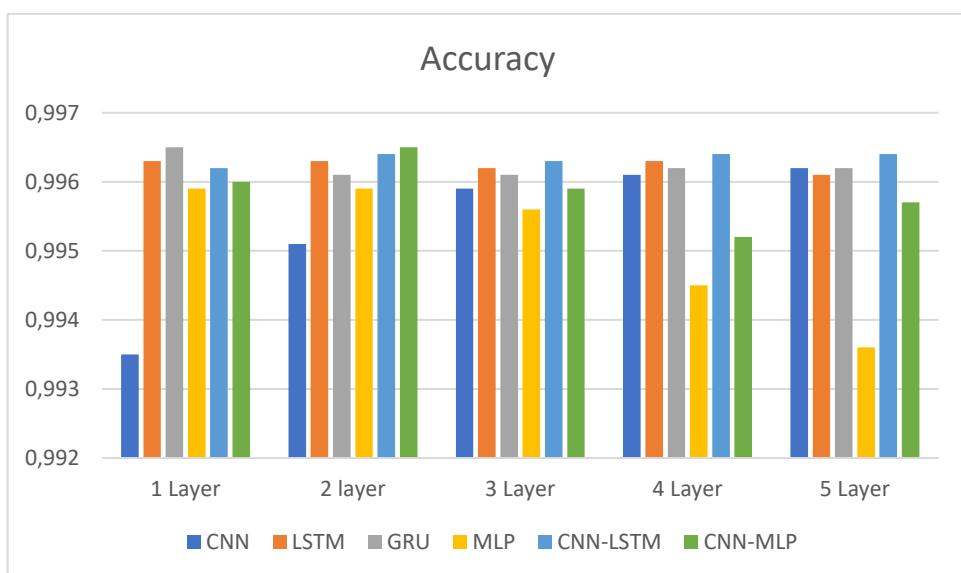
Hình 23: Trước và sau khi cân bằng của tập dữ liệu (1)CSE-CIC-IDS-2018.

4.3.1.2 Kết quả huấn luyện

- **Đánh giá tổng quát:** Hình dưới đây trình bày kết quả huấn luyện tổng quát của các thuật toán, thể hiện qua độ chính xác (Accuracy).

Model	Accuracy				
	1 Layer	2 layer	3 Layer	4 Layer	5 Layer
CNN	0,9935	0,9951	0,9959	0,9961	0,9962
LSTM	0,9963	0,9963	0,9962	0,9963	0,9961
GRU	0,9965	0,9961	0,9961	0,9962	0,9962
MLP	0,9959	0,9959	0,9956	0,9945	0,9936
CNN-LSTM	0,9962	0,9964	0,9963	0,9964	0,9964
CNN-MLP	0,996	0,9965	0,9959	0,9952	0,9957

- Biểu đồ thể hiện sự so sánh độ chính xác (Accuracy) tổng quát của các mô hình:

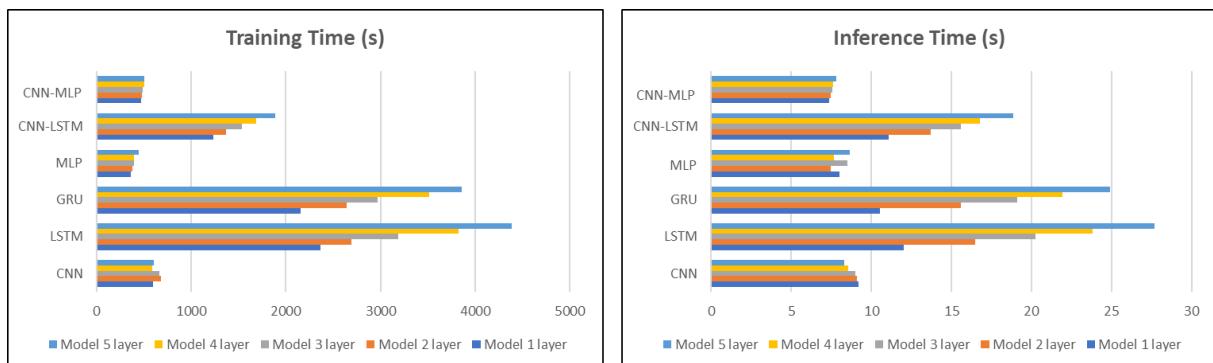


Hình 24: Biểu đồ chính xác (Accuracy) tổng quát của các mô hình trên kích thước 1.

-Hình dưới đây trình bày tổng quát thời gian huấn luyện và thời gian suy luận :

Model	1 Layer		2 layer		3 Layer		4 Layer		5 Layer	
	Train Time (s)	Inference Time (s)								
CNN	590.79	9.2	681	9.11	665.11	8.99	588.89	8.56	604.14	8.31
LSTM	2365.29	12	2693.1	16.47	3190.54	20.24	3823.39	23.83	4390.5	27.68
GRU	2156.16	10.55	2640.77	15.61	2969.86	19.13	3518.03	21.96	3864.42	24.94
MLP	355.86	8.01	375.86	7.47	390.64	8.48	391.14	7.64	442.34	8.65
CNN-LSTM	1228.44	11.09	1364.27	13.7	1535.15	15.58	1687.36	16.8	1885.58	18.89
CNN-MLP	470.5	7.36	473.66	7.47	485.12	7.55	499.21	7.6	500.36	7.79

-Biểu đồ thể hiện sự so sánh thời gian huấn luyện và thời gian suy luận tổng quát của các mô hình:



Hình 25: Biểu đồ thời gian huấn luyện và suy luận tổng quát của các mô hình trên kịch bản 1.

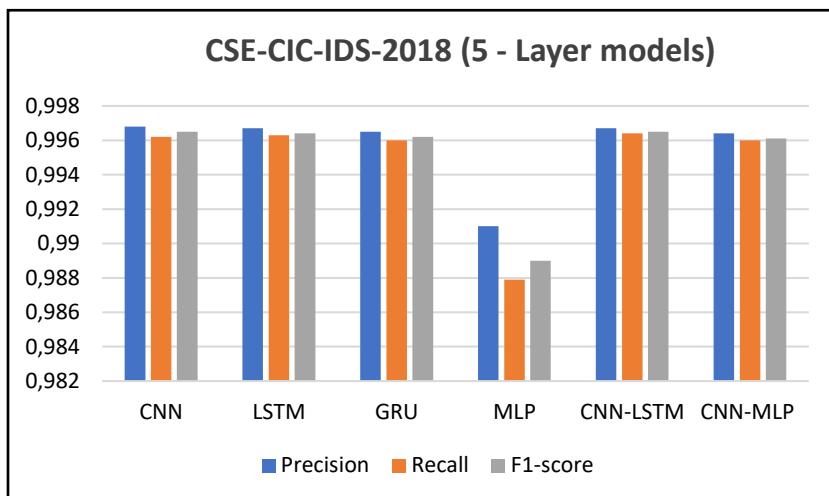
Nhận xét :

- Tất cả các mô hình đều đạt độ chính xác cao dao động từ 0.993 đến 0.996, có sự chênh lệch không quá lớn.
- Quan sát thời gian huấn luyện của các mô hình cho thấy sự khác biệt rõ rệt giữa các nhóm thuật toán. Cụ thể, GRU và LSTM có thời gian huấn luyện dài nhất ở tất cả số lớp, cụ thể tại cấu hình 1-Layer, thời gian huấn luyện của hai mô hình này cao gấp khoảng 4-5 lần so với CNN và gấp 6-7 lần so với mô hình MLP. Nhóm mô hình kết hợp CNN-LSTM và CNN-MLP có thời gian huấn luyện trung bình, ngắn hơn so với LSTM nhưng vẫn cao hơn đáng kể so với nhóm mô hình thuần CNN hoặc MLP, tuy nhiên mô hình kết hợp này lại đạt độ chính xác cao ở nhiều cấu hình (0.9963-0.9965).
- Quan sát thời gian suy luận của các mô hình còn lại không có sự chênh lệch quá nhiều. Trong số các mô hình, mô hình MLP và mô hình kết hợp CNN-MLP đạt thời gian suy luận nhanh nhất (7-8 giây).

- **Đánh giá chi tiết:** Trình bày kết quả chi tiết đối với cấu hình gồm 5 lớp ẩn.

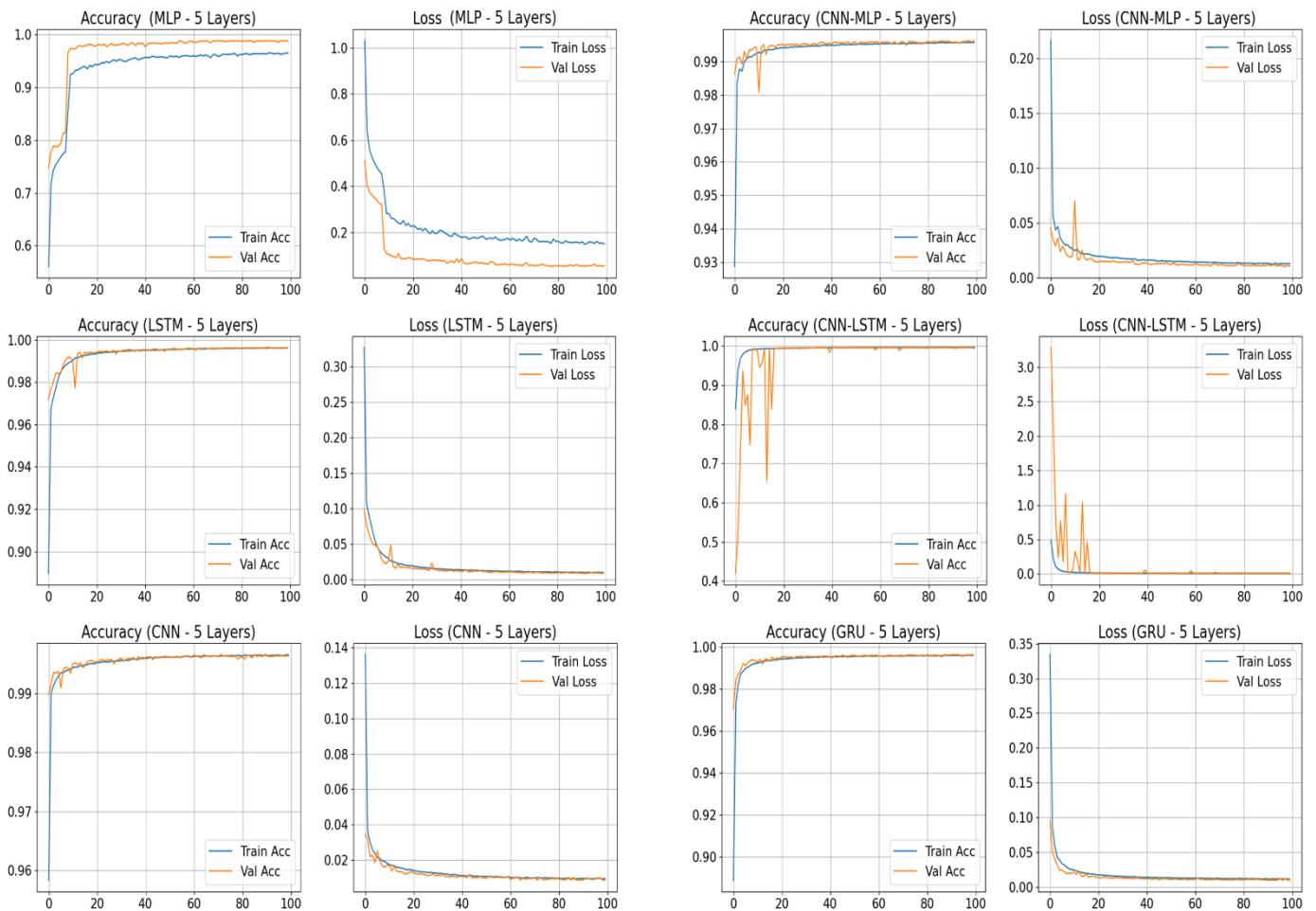
- Dưới đây là kết quả chi tiết thể hiện sự so sánh các chỉ số Precision, Recall, F1-score:

5-Layer model	Precision	Recall	F1-score
CNN	0.9968	0.9962	0.9965
LSTM	0.9967	0.9963	0.9964
GRU	0.9965	0.996	0.9962
MLP	0.991	0.9879	0.989
CNN-LSTM	0.9967	0.9964	0.9965
CNN-MLP	0.9964	0.996	0.9961



Nhận xét : Nhìn chung, tất cả các mô hình đều đạt độ chính xác rất cao với Precision, Recall và F1-score đều trên 0.99, chứng tỏ khả năng phân loại lưu lượng mạng trong tập dữ liệu CSE-CIC-IDS-2018 là rất tốt. Trong đó, mô hình CNN và CNN-LSTM cho thấy hiệu suất nổi bật với F1-score cao nhất (0.9965), thể hiện sự cân bằng tốt giữa Precision và Recall. Đặc biệt, CNN có Precision cao nhất (0.9968), cho thấy khả năng phân loại chính xác và ít nhầm lẫn hơn so với các mô hình còn lại. Tuy nhiên mô hình MLP, hiệu suất nhìn chung thấp hơn các mô hình khác, đặc biệt ở Recall (0.9879), cho thấy khả năng nhận diện đầy đủ các mẫu tấn công còn nhiều nhầm lẫn.

-Để quan sát được quá trình học của các mô hình, các hình dưới đây trình bày lịch sử huấn luyện (training history) của từng mô hình:



Hình 26: Lịch sử huấn luyện (training history) trên kịch bản 1.

Nhận xét :

+Nhóm (1) và nhóm (2) – (CNN , LSTM ,GRU):

Các mô hình đạt độ chính xác gần 99.5% trên cả tập huấn luyện và kiểm thử, Đồ thị Accuracy và đồ thị Loss cho thấy đường train và đường validation gần như trùng nhau, Loss giảm ổn định về mức rất thấp (gần 0.01). Hiệu suất ổn định chứng tỏ mô hình có khả năng học tốt.

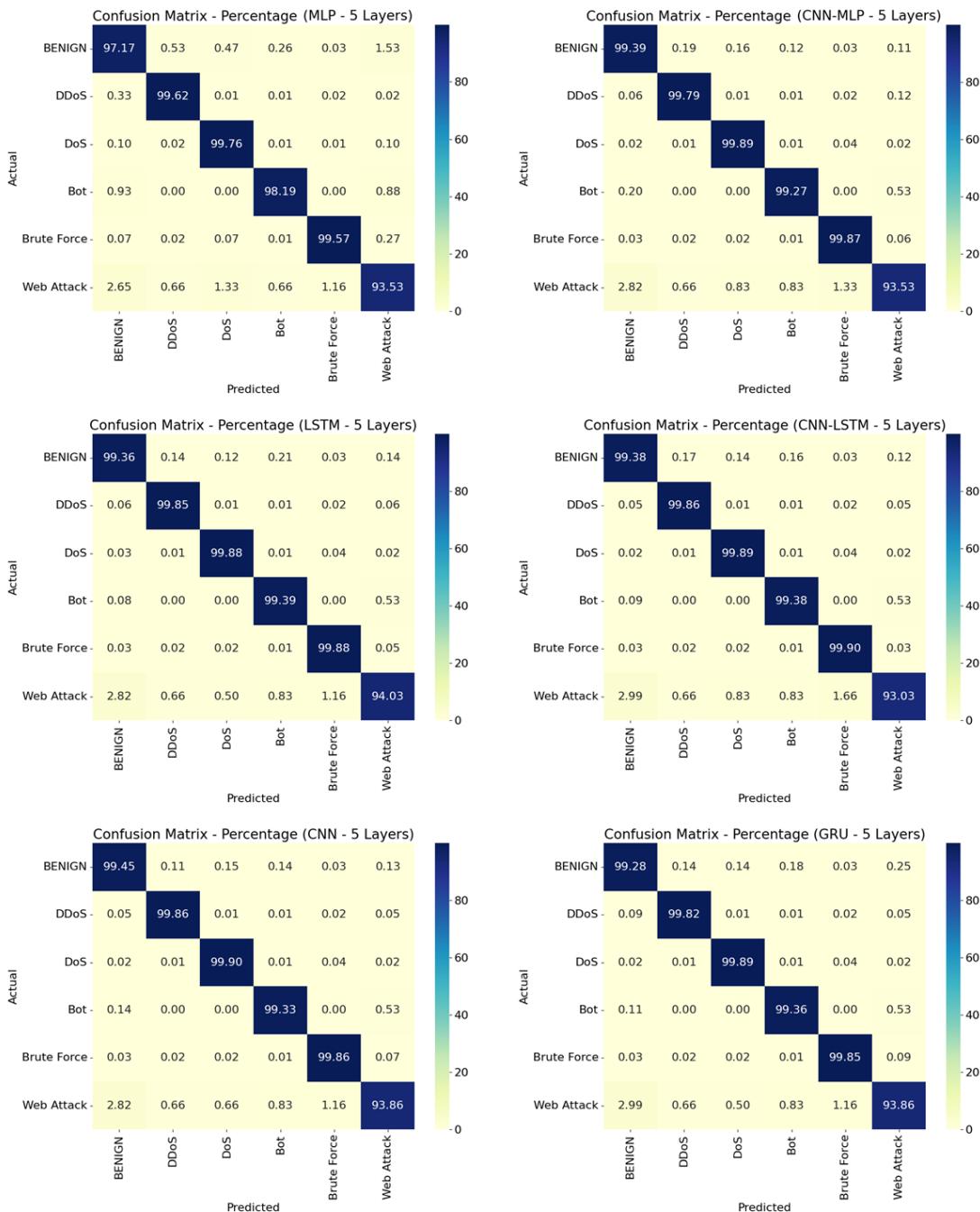
+Nhóm (3) – (MLP):

Mô hình MLP đạt độ chính xác gần 99.1%, Loss giảm ổn định về mức rất thấp (gần 0.02). Tuy nhiên, hiệu suất vẫn kém hơn so với các nhóm khác.

+Nhóm (4) – (CNN-LSTM , CNN-MLP):

Cả hai mô hình CNN-LSTM và CNN-MLP đều đạt độ chính xác cao trên cả tập huấn luyện và kiểm thử. Đồ thị Accuracy và Loss của train và validation bám sát nhau, Loss giảm ổn định về mức rất thấp với CNN-LSTM (gần 0.02) và CNN-MLP (gần 0.03).

-Để đánh giá khả năng phân loại của từng mô hình, các hình dưới đây trình bày ma trận nhầm lẫn (confusion matrix) tương ứng cho từng trường hợp:

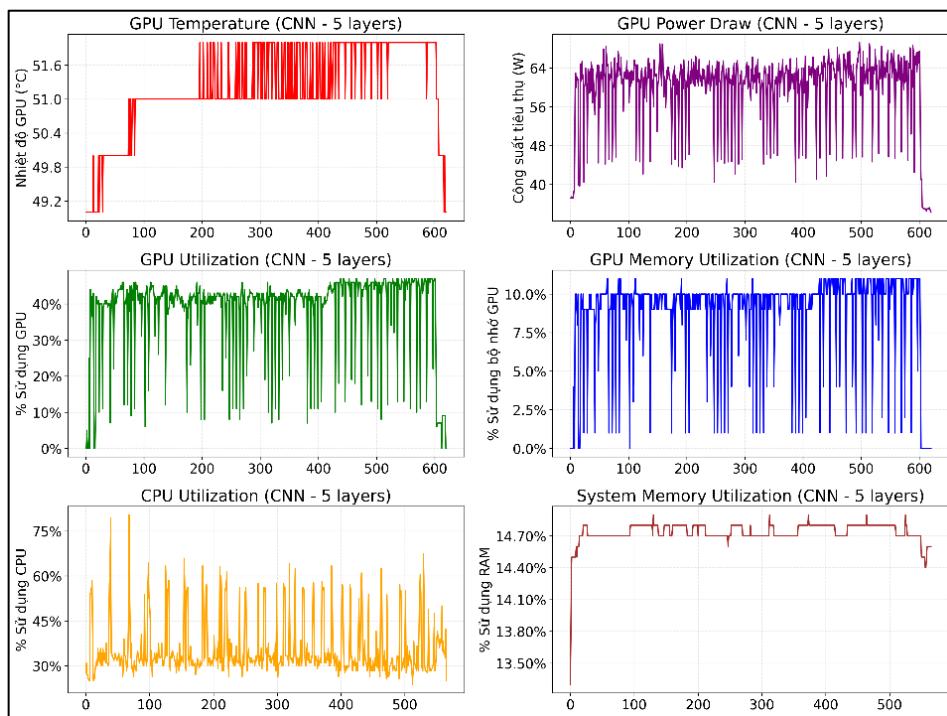


Hình 27: Ma trận nhầm lẫn (confusion matrix) trên kịch bản 1.

Nhận xét : Nhìn chung, tất cả các mô hình đều thể hiện hiệu suất cao trong việc phân loại các loại tấn công BENIGN, DDoS, DoS, Bot, Brute Force, Web Attack với độ chính xác dao động từ 93% đến 99,9%. Các mô hình có xu hướng phân loại tốt nhất với các lớp BENIGN và DDoS/DoS, đạt trên 98% ở hầu hết các trường hợp. Tuy nhiên các mô hình cho thấy sự nhầm lẫn còn nhiều với lớp thiểu số Web Attack.

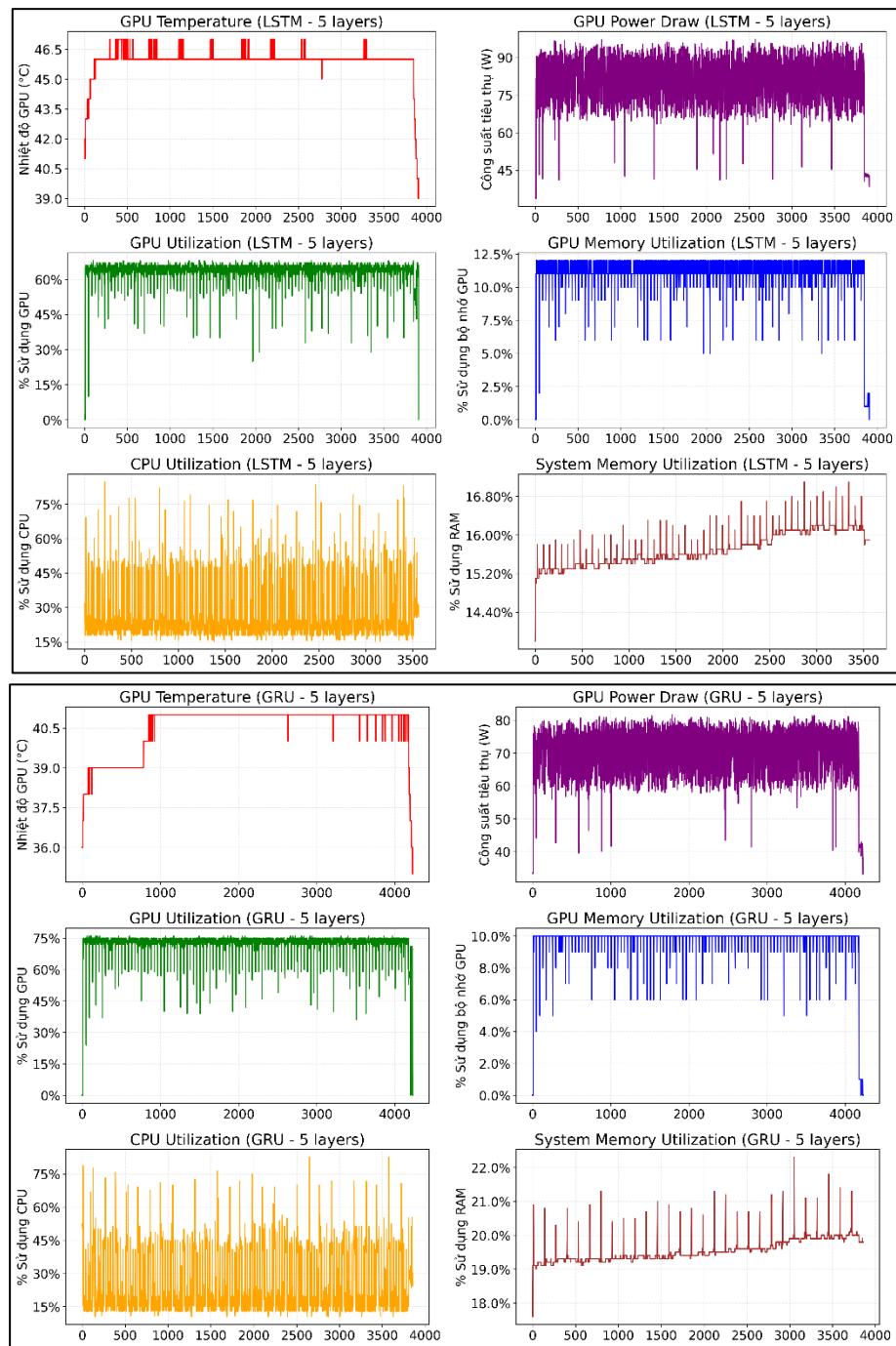
-Để đánh giá mức độ sử dụng tài nguyên CPU và GPU các hình dưới đây trình bày biểu đồ tương ứng cho các trường hợp:

+Nhóm (1) - (CNN) :



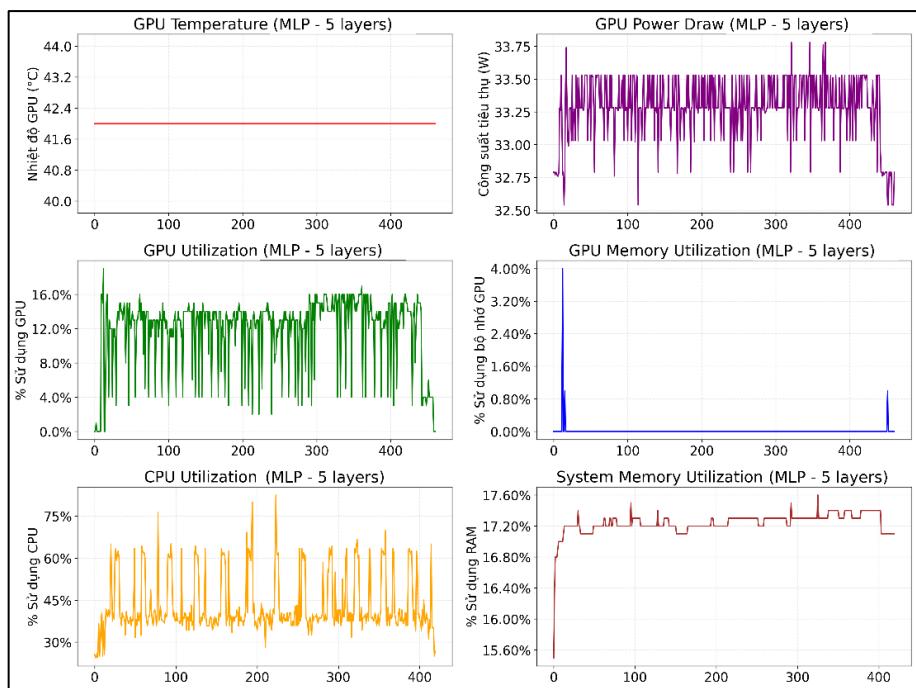
Nhận xét: Nhóm này cho thấy tận dụng hiệu quả khả năng xử lý của GPU, thể hiện qua mức sử dụng GPU ổn định quanh 40-45%, công suất tiêu thụ khoảng 60–65W, nhiệt độ GPU duy trì ở mức 51–52°C, bộ nhớ GPU được khai thác khoảng 10-12%, CPU hoạt động ở mức trung bình 30–40% và có thời điểm đạt đỉnh lên tới 75%, và RAM hệ thống ổn định quanh 14.7%.

+Nhóm (2) – (LSTM, GRU):



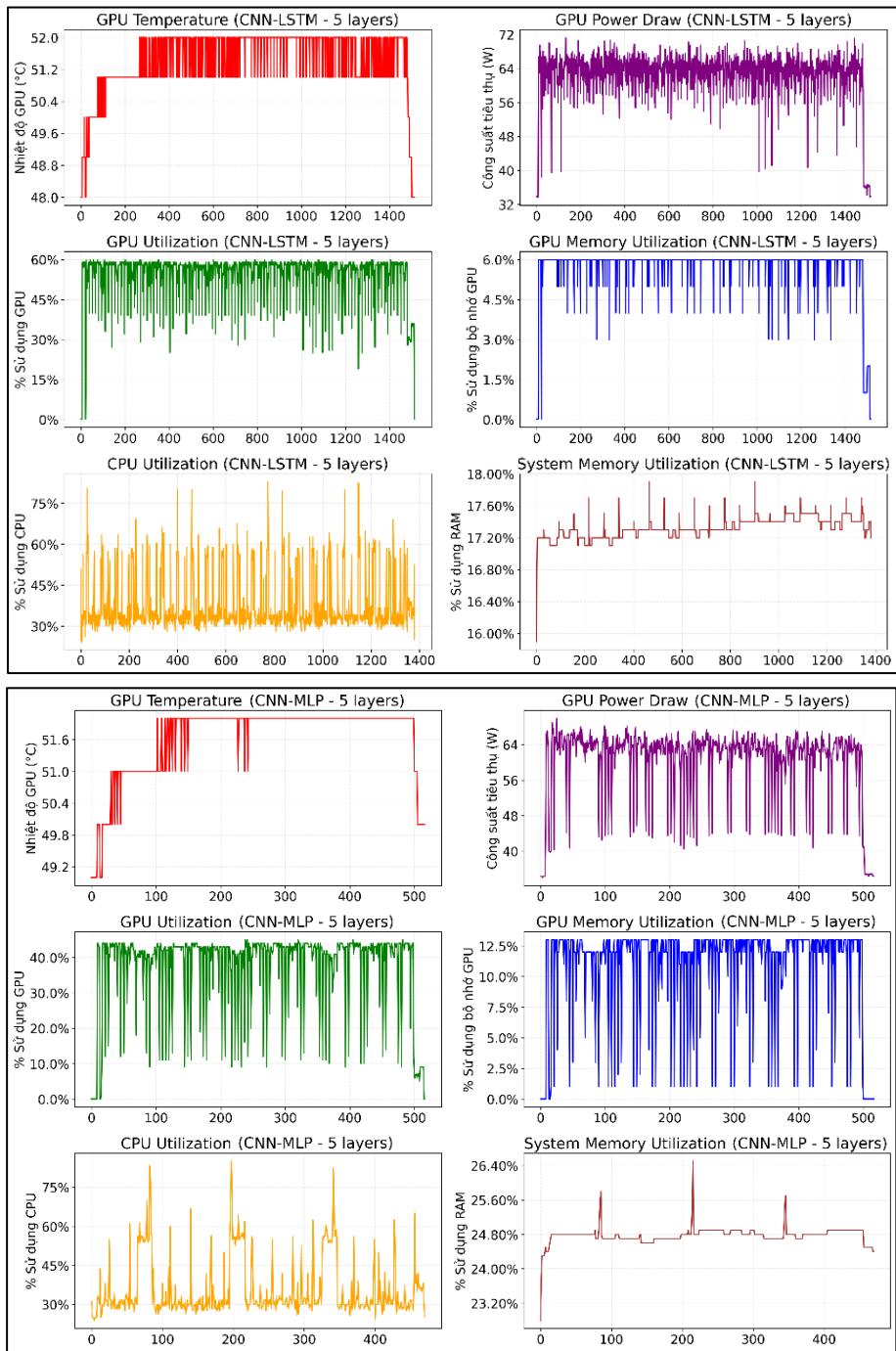
Nhận xét: Nhóm này cho thấy tiêu thụ năng lượng rất cao trong quá trình huấn luyện, với công suất GPU thường xuyên trên 75W và nhiều thời điểm vượt 90W, mức sử dụng GPU ổn định trên 60-75%, bộ nhớ GPU duy trì ở mức hơn 10-12%, CPU hoạt động cao trung bình khoảng 25-50% và biến động mạnh có nhiều lúc đạt 75%, trong đó RAM hệ thống dao động tăng dần theo thời gian, nhóm mô hình phản ánh rõ khối lượng tính toán lớn và yêu cầu xử lý song song cao giữa CPU và GPU.

+Nhóm (3) – (MLP):



Nhận xét: Nhóm này có mức tiêu thụ tài nguyên thấp trong quá trình huấn luyện, với công suất GPU duy trì quanh 33,5W, nhiệt độ ổn định khoảng 40,8°C, mức sử dụng GPU dao động nhẹ trong khoảng 16-20% , bộ nhớ GPU gần như thấp gần như bằng 0%. Ngược lại, CPU được khai thác ở mức cao hơn trung bình khoảng 40–60% đôi khi tới 75%, trong khi RAM hệ thống duy trì ổn định 27–29% với vài biến động nhỏ. Mô hình này phản ánh phần lớn khói lượng tính toán được xử lý trên CPU.

+Nhóm (4)- (CNN-LSTM ,CNN-MLP):



Nhận xét : Đối với mô hình CNN-LSTM, GPU duy trì ổn định với nhiệt độ 51–52°C, tiêu thụ điện năng 60–70W và mức độ sử dụng khoảng 45–60%, thấp hơn LSTM thuần, CPU lại được khai thác tương đối nhiều dao động trung bình khoảng 35% nhiều lúc đạt 75%. Trong khi đó, CNN-MLP cho thấy mức sử dụng GPU thấp hơn một chút dao động trong khoảng 30–45%, còn mức độ sử dụng CPU tương tự CNN-LSTM khoảng 30%-60%, và RAM hệ thống trung bình khoảng 24.8-26%, cao hơn CNN-LSTM nhưng vẫn thấp hơn MLP thuần. Nhóm mô hình kết hợp cho thấy khả năng khai thác tài nguyên cân đối và hiệu quả hơn so với các mô hình thuần.

4.3.2. Kết quả trên kịch bản 2

4.3.2.1 Kết quả tiền xử lý

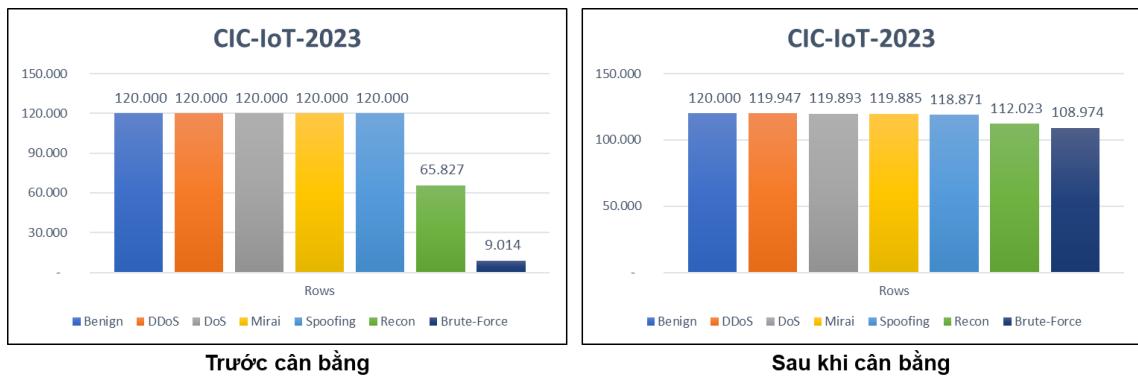
- Tập dữ liệu (2) : CIC-IoT-2023
- Làm sạch dữ liệu loại bỏ :
 - + Loại bỏ các dòng có các giá trị NaN hoặc Inf.
 - + Loại bỏ được nhiều dòng trùng nhau ở những dòng có nhãn Benign , Bot-Mirai, DDoS.
- Thực hiện phương pháp lấy mẫu (Data Sampling) Random Under-Sampling (RUS) (đã trình bày ở mục 3.3.6).
- Mã hóa các nhãn cho tập dữ liệu (2) với bài toán phân loại đa lớp và số lượng mẫu cho mỗi nhãn sau khi làm sạch dữ liệu :

Label_encoding	Attack Category	Rows
0	Benign	150.000
1	DDoS	150.000
2	DoS	150.000
3	Bot-Mirai	150.000
4	Spoofing	150.000
5	Recon	82.284
6	Brute-Force	11.268

- Các cột đặc trưng được thuật toán Random Forest lựa chọn là những cột đặc trưng có độ quan trọng (feature importance) > 0,01. Trong tập dữ liệu (2), sau khi áp dụng ngưỡng này, ta thu được tổng cộng 32 cột, bao gồm 30 cột đặc trưng vào và 2 cột đặc trưng mục tiêu “Label” và “Label_encode”, được thể hiện ở hình bên dưới :

```
['flow_duration', 'Header_Length', 'Protocol Type', 'Duration', 'Rate',
 'syn_flag_number', 'psh_flag_number', 'ack_count', 'syn_count',
 'fin_count', 'urg_count', 'rst_count', 'HTTP', 'HTTPS', 'SSH', 'TCP',
 'UDP', 'Tot sum', 'Min', 'Max', 'AVG', 'Std', 'Tot size', 'IAT',
 'Number', 'Magnitue', 'Radius', 'Covariance', 'Variance', 'Weight',
 'Label_encode', 'Label']
```

- Sau khi chia tập dữ liệu (2) thành 2 phần tập huấn luyện (80%) và tập kiểm tra (20%).
- Ta đếm tập huấn luyện cân bằng dữ liệu với kỹ thuật **SMOTE+Tomek-Link**.
- Kết quả trước và sau khi cân bằng của tập huấn luyện trên tập dữ liệu (2) được thể hiện ở **Hình 28**.



Hình 28: Trước và sau khi cân bằng của tập dữ liệu (2)CIC-IoT-2023.

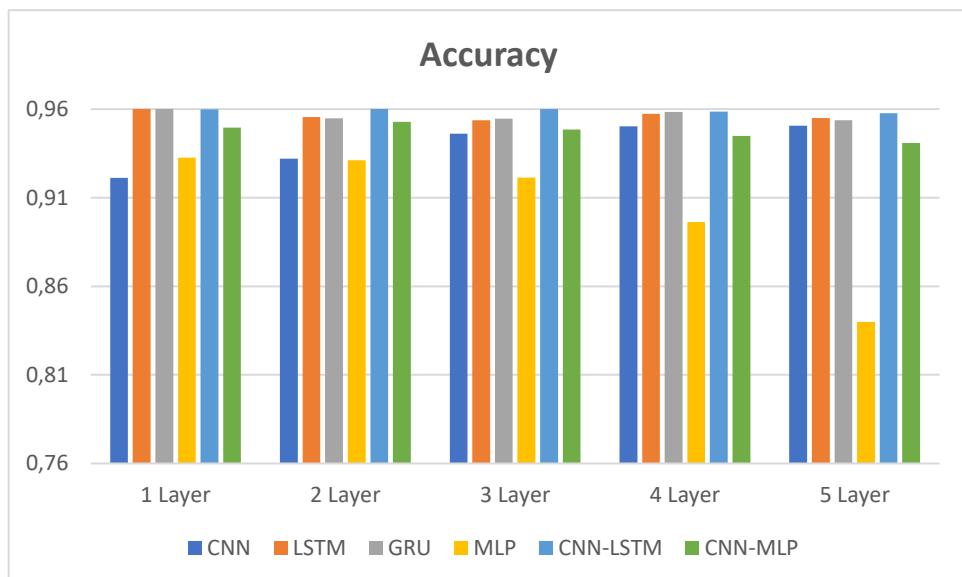
4.3.2.2 Kết quả huấn luyện

- Đánh giá tổng quát:

-Hình dưới đây trình bày kết quả huấn luyện tổng quát của các thuật toán, thể hiện qua độ chính xác (Accuracy) và thời gian huấn luyện

Model	Accuracy				
	1 Layer	2 layer	3 Layer	4 Layer	5 Layer
CNN	0,9212	0,9321	0,9462	0,9503	0,9506
LSTM	0,9601	0,9555	0,9538	0,9574	0,955
GRU	0,9602	0,9548	0,9547	0,9584	0,9537
MLP	0,9325	0,9311	0,9214	0,8963	0,8399
CNN-LSTM	0,9599	0,9625	0,9601	0,9586	0,9576
CNN-MLP	0,9496	0,9528	0,9484	0,9449	0,9409

- Biểu đồ thể hiện sự so sánh độ chính xác (Accuracy) tổng quát của các mô hình:

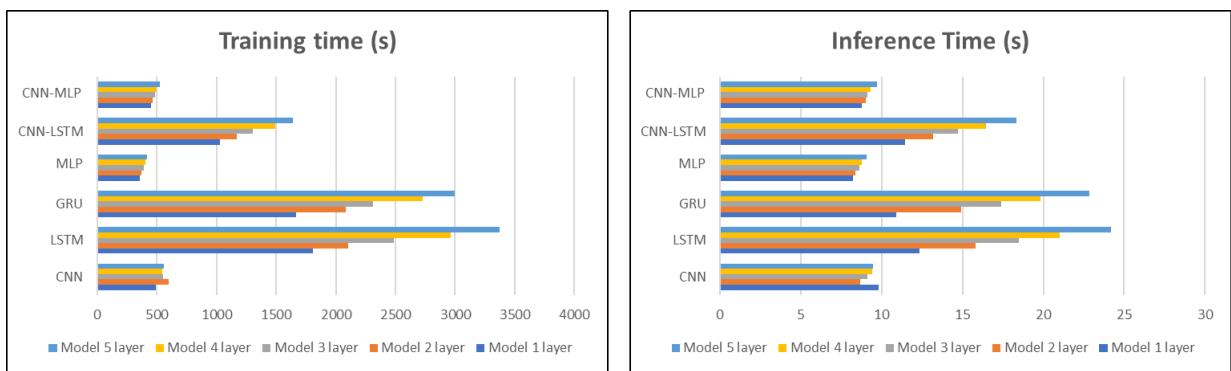


Hình 29: Biểu đồ chính xác (Accuracy) tổng quát của các mô hình trên kịch bản 2.

-Hình dưới đây trình bày tổng quát thời gian huấn luyện và thời gian suy luận :

Model	1 Layer		2 layer		3 Layer		4 Layer		5 Layer	
	Train Time (s)	Inference Time (s)								
CNN	494.77	9.8	601.17	8.69	553.17	9.09	542.25	9.41	555.25	9.44
LSTM	1810.31	12.33	2103.3	15.82	2485.61	18.47	2966.13	20.99	3371.69	24.17
GRU	1668.27	10.92	2085.58	14.92	2314.45	17.4	2730.34	19.81	2996.91	22.86
MLP	357.29	8.2	372.18	8.37	390.44	8.63	404.31	8.76	415.98	9.07
CNN-LSTM	1026.03	11.44	1168.72	13.18	1304.46	14.73	1492.48	16.46	1641.33	18.36
CNN-MLP	453.19	8.78	465.7	9.02	486.85	9.13	501.18	9.29	523.1	9.69

-Biểu đồ thể hiện sự so sánh thời gian huấn luyện và thời gian suy luận tổng quát của các mô hình:



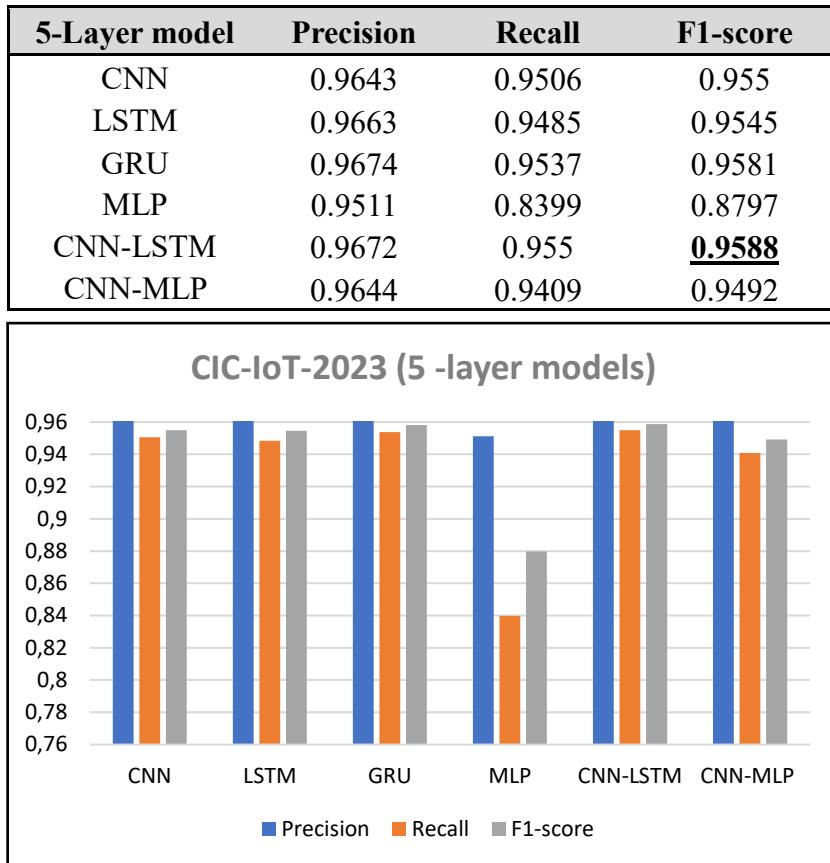
Hình 30: Biểu đồ thời gian huấn luyện và suy luận tổng quát của các mô hình trên kịch bản 2.

Nhận xét :

- Tất cả các mô hình đều đạt độ chính xác tương đối cao, dao động trong khoảng 0.92–0.96, có sự khác biệt nhẹ giữa các mô hình. Trong đó, LSTM, GRU và đặc biệt là CNN-LSTM thể hiện hiệu suất nổi bật và ổn định, đạt độ chính xác cao nhất ở nhiều cấu hình (tối đa 0.9625). Ngược lại, MLP cho kết quả kém hơn rõ rệt và suy giảm mạnh khi tăng số lớp, chỉ còn 0.8399 ở cấu hình 5 lớp.
- Quan sát thời gian huấn luyện cho thấy sự khác biệt rõ rệt giữa các nhóm mô hình: CNN, MLP và CNN-MLP có thời gian huấn luyện ngắn nhất, trong khi LSTM và GRU lại mất nhiều thời gian hơn đáng kể ở mọi cấu hình số lớp. Tuy nhiên, LSTM và GRU vẫn duy trì độ chính xác ổn định, thường xếp ở vị trí thứ hai hoặc thứ ba, chỉ sau mô hình kết hợp CNN-LSTM – vốn gần như luôn dẫn đầu với độ chính xác dao động quanh 0.95–0.96 ở hầu hết các cấu hình.
- Quan sát thời gian suy luận giữa các mô hình không có sự chênh lệch lớn, với MLP và CNN-MLP là nhanh nhất, còn LSTM và GRU thường chậm hơn một chút khoảng 3-4 giây do độ phức tạp cao hơn.

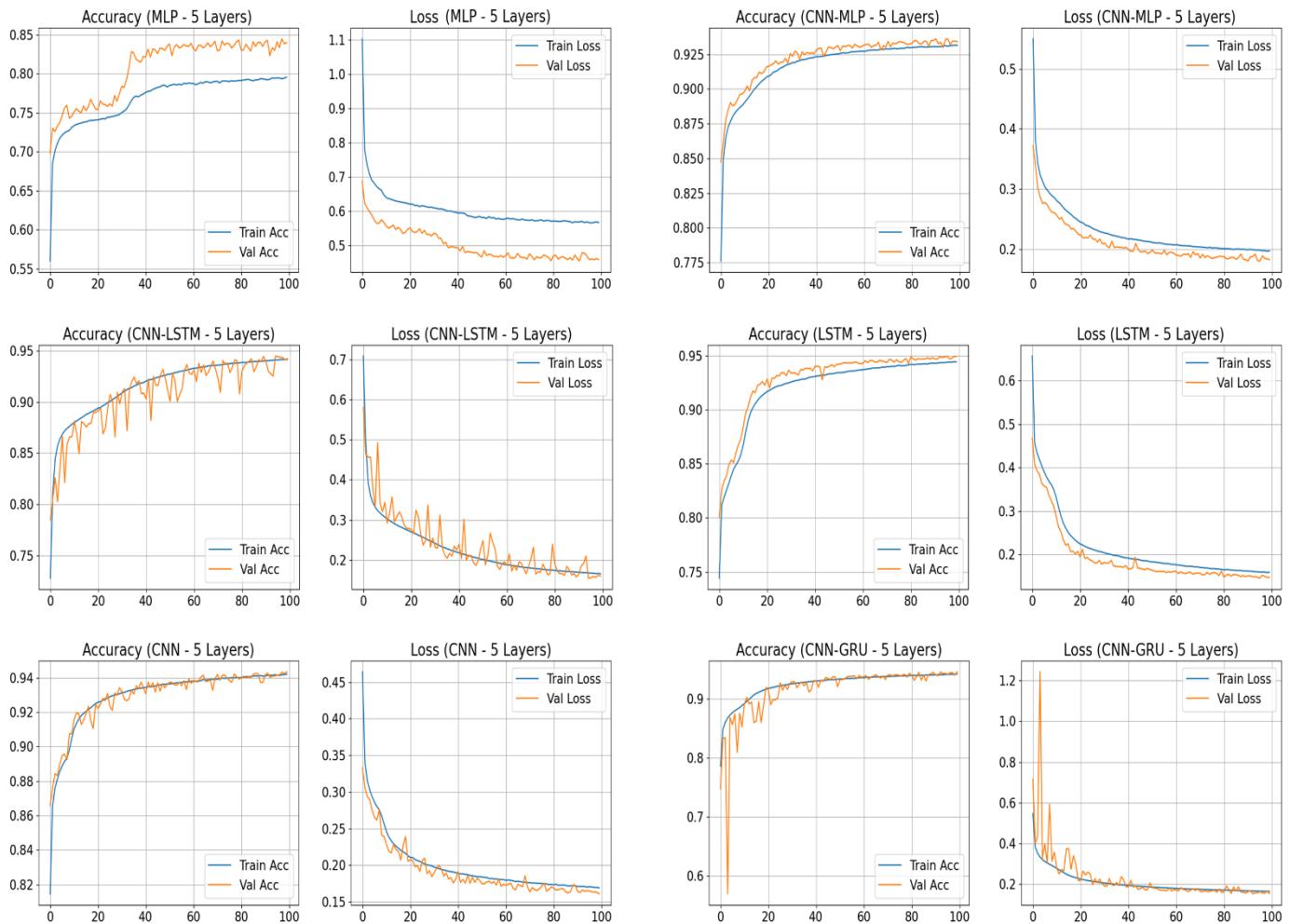
- **Đánh giá chi tiết:** Trình bày kết quả chi tiết đối với cấu hình gồm 5 lớp ẩn.

- Dưới đây là kết quả chi tiết thể hiện sự so sánh các chỉ số Precision, Recall, F1-score:



Nhận xét : Nhìn chung, tất cả các mô hình đều đạt kết quả tốt với Precision, Recall và F1-score đều trên 0.87, phản ánh khả năng phân loại lưu lượng mạng trên tập dữ liệu CIC-IoT-2023 ở mức tương đối tốt. Trong đó, GRU đạt hiệu suất nổi bật nhất với F1-score 0.9581, theo sát là LSTM với 0.9588, cho thấy hai mô hình này tận dụng tốt đặc trưng tuần tự của dữ liệu. CNN cũng cho kết quả ổn định với F1-score 0.955, trong khi CNN-LSTM và CNN-MLP giữ mức cân bằng khá tốt nhưng thấp hơn một chút so với GRU và LSTM. Ngược lại, MLP thể hiện hiệu suất kém nhất (F1-score 0.8797), đặc biệt ở Recall, cho thấy khả năng bỏ sót mẫu tấn công cao hơn so với các mô hình còn lại.

-Để quan sát được quá trình học của các mô hình, hình dưới đây trình bày lịch sử huấn luyện (training history) của từng mô hình:



Hình 31: Lịch sử huấn luyện (training history) trên kịch bản 2.

-Nhận xét :

+Nhóm (1) – (CNN):

CNN đạt độ chính xác khoảng 94% trên cả train và validation, Loss giảm đều về mức thấp (gần 0.05). Đường train và validation bám sát nhau, tuy nhiên các đường vẫn còn dao động nhẹ, do đó vẫn có thể tăng thêm số epoch huấn luyện để giúp mô hình hội tụ ổn định hơn.

+Nhóm (2) – (LSTM ,GRU):

LSTM đạt độ chính xác gần 95%, GRU khoảng 94% trên cả tập huấn luyện và kiểm thử. Đồ thị Accuracy và Loss cho thấy đường train và validation gần như trùng nhau, Loss giảm nhanh và ổn định (LSTM gần 0.02, GRU gần 0.06).

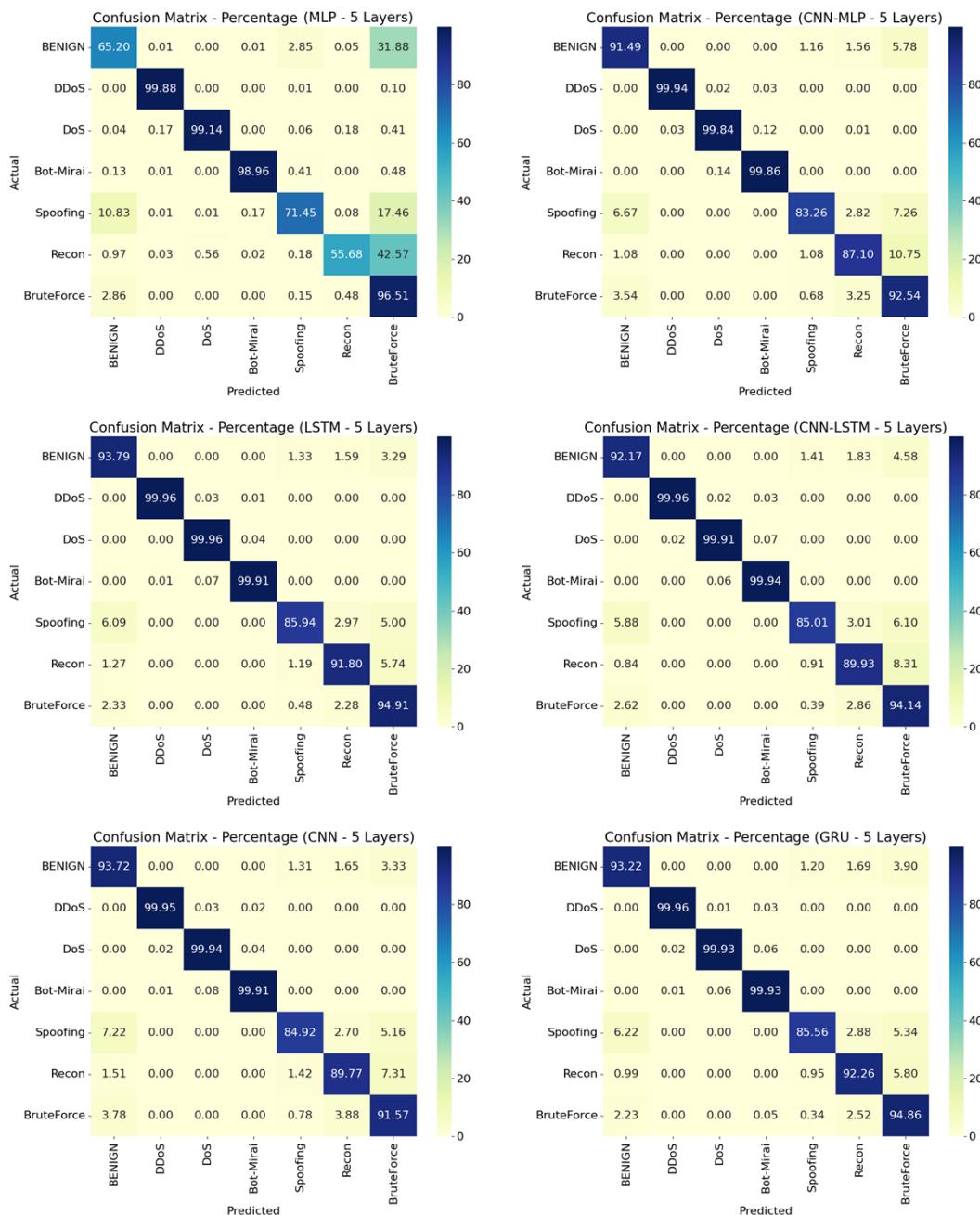
+Nhóm (3) – (MLP):

MLP đạt khoảng 92% trên tập huấn luyện và khoảng 91% trên tập kiểm thử, loss giảm ổn định về mức gần 0.12 .Tuy nhiên có dấu hiệu nhẹ bị overfitting và vẫn kém hơn đáng kể so với các mô hình CNN kết hợp hoặc mô hình tuần tự như LSTM và GRU.

+Nhóm (4) – (CNN-MLP , CNN-LSTM):

Cả hai đạt độ chính xác cao, tiệm cận 95% trên cả train và validation. Loss giảm nhanh và ổn định (CNN-LSTM gần 0.02, CNN-MLP gần 0.03), đường train và validation gần như trùng nhau, cho thấy khả năng học và tổng quát hóa rất tốt.

-Để đánh giá khả năng phân loại của từng mô hình, các hình dưới đây trình bày ma trận nhầm lẫn (confusion matrix) tương ứng cho từng trường hợp:

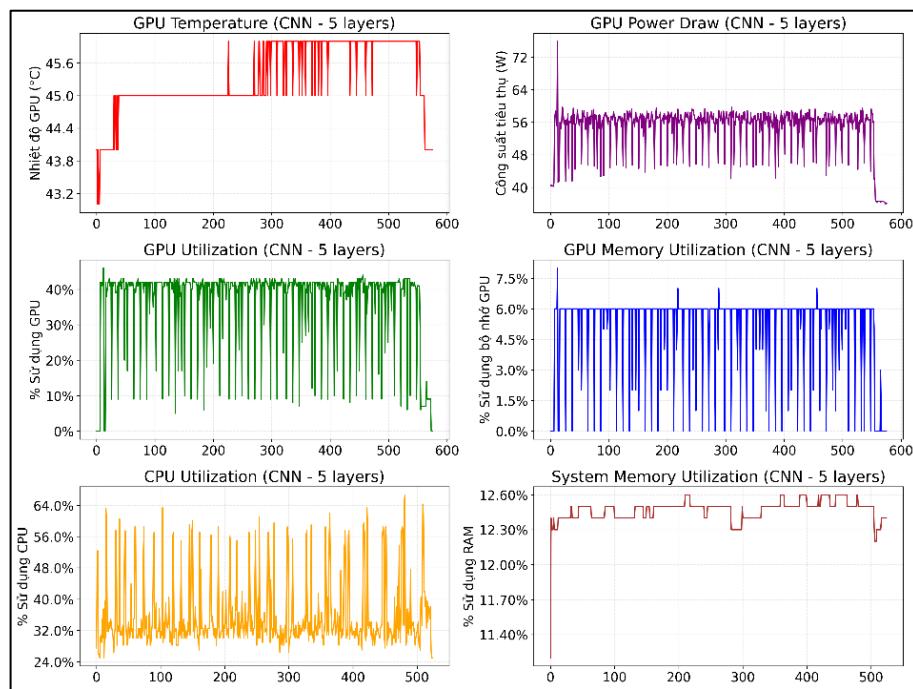


Hình 32: Ma trận nhầm lẫn (confusion matrix) trên kích bản 2.

Nhận xét : Nhìn chung các mô hình nhìn chung đạt hiệu suất phân loại tương đối tốt, với các lớp BENIGN, DDoS, DoS và Bot-Mirai được nhận diện chính xác, ít xảy ra nhầm lẫn với tỷ lệ đúng đạt trên 95% ở hầu hết mô hình. Tuy nhiên, đối với các lớp Spoofing, Reconnaissance và đặc biệt là BruteForce, tỷ lệ nhầm lẫn vẫn còn đáng kể, có trường hợp sai lệch lên đến hơn 10%. Trong số các mô hình, CNN và CNN-LSTM thể hiện ưu thế rõ rệt khi duy trì khả năng phân loại ổn định ở các lớp khó, trong khi đó mô hình MLP lại xuất hiện nhiều nhầm lẫn hơn ở các lớp Benign, Spoofing và Reconnaissance.

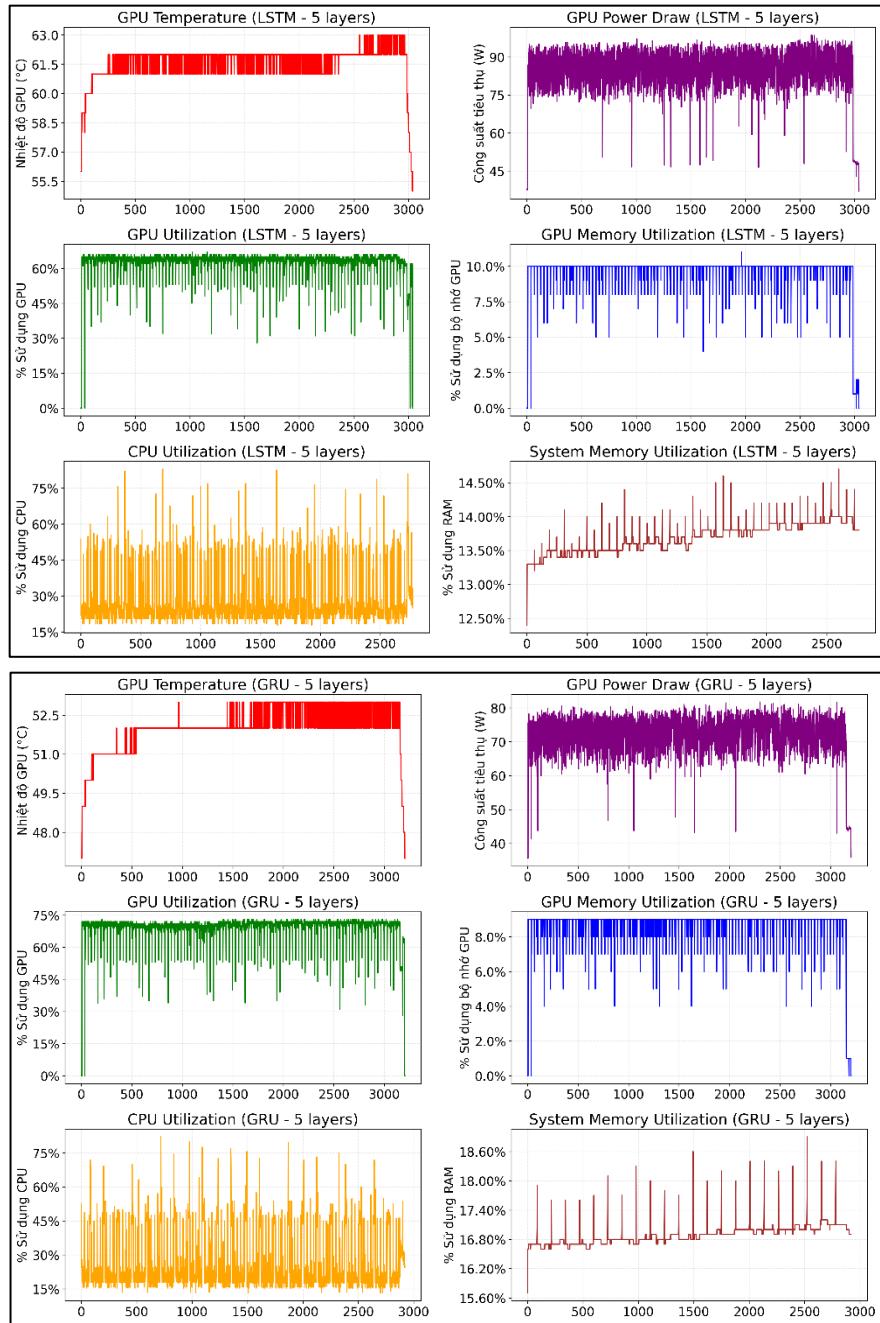
-Để đánh giá mức độ sử dụng tài nguyên CPU và GPU các hình dưới đây trình bày biểu đồ tương ứng cho các trường hợp:

+Nhóm (1) – (CNN):



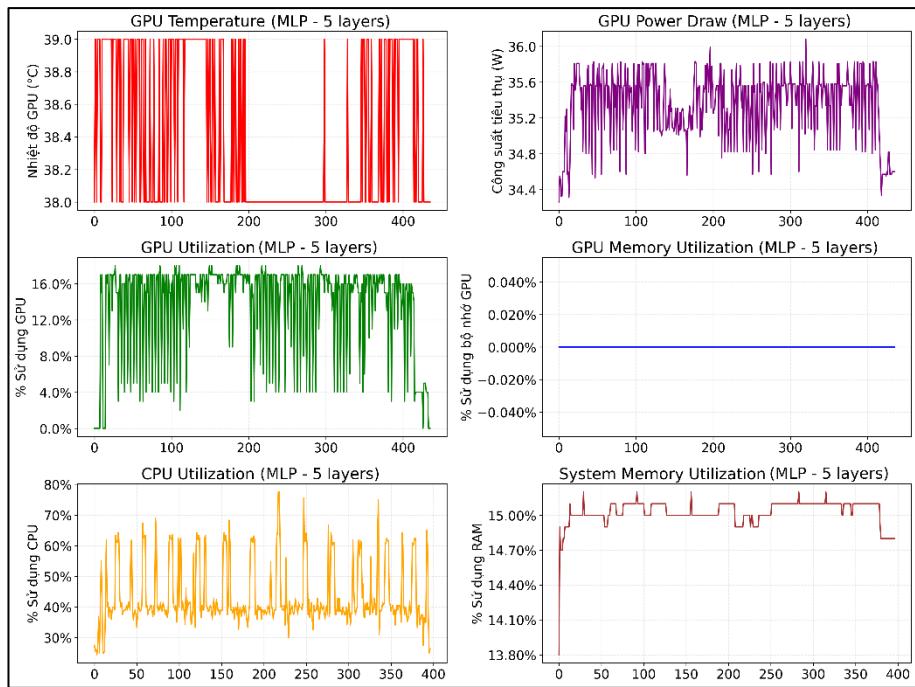
Nhận xét: Nhóm này cho thấy tận dụng hiệu quả khả năng xử lý của GPU, thể hiện qua mức sử dụng GPU ổn định quanh 35-40%, công suất tiêu thụ trung bình 48-56W, nhiệt độ GPU duy trì ở mức 45°C , bộ nhớ GPU được khai thác ở mức 6%, CPU hoạt động ở mức trung bình 35–40% và có thời điểm đạt đỉnh lên tới 60%, và RAM hệ thống ổn định quanh 12%.

+Nhóm (2) – (LSTM, GRU): LSTM là mô hình tiêu biểu đại diện cho nhóm này:



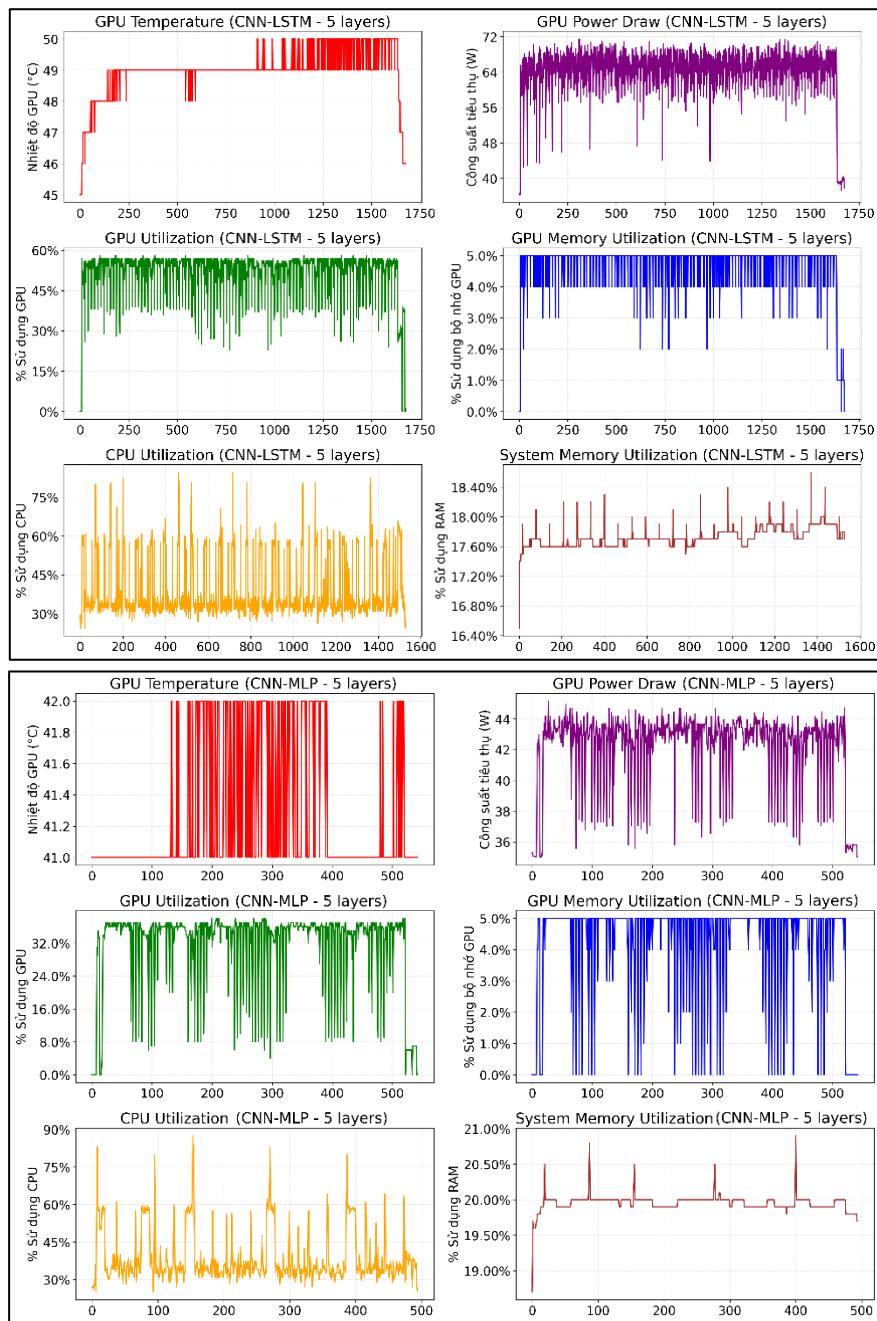
Nhận xét: Nhóm này cho thấy tiêu thụ năng lượng rất cao trong quá trình huấn luyện với công suất tiêu thụ GPU rất cao, thường xuyên trên 75W và có thời điểm vượt 90W. Mức sử dụng GPU duy trì ổn định trên 60-65%, kéo theo nhiệt độ GPU tăng cao trung bình khoảng 62°C (LSTM) và 52°C (GRU) và bộ nhớ GPU duy trì ở mức hơn 8-10% cao hơn so với các mô hình còn lại. CPU hoạt động ở cường độ khá cao trung bình khoảng 25-45% với biến động lớn có lúc đạt tới hơn 75%, và RAM hệ thống dao động tăng dần, phản ánh khối lượng xử lý lớn cùng sự phối hợp chặt chẽ giữa CPU và GPU trong suốt quá trình huấn luyện.

+Nhóm (3) – (MLP):



Nhận xét: Nhóm này cho thấy hầu như không tận dụng khả năng xử lý của GPU. Mức sử dụng GPU duy trì ở mức rất thấp, chủ yếu dưới 17%, và bộ nhớ GPU gần như thấp gần như bằng 0%. Công suất tiêu thụ GPU ổn định quanh gần 35,6W, nhiệt độ duy trì ở mức thấp dao động 38-39°C, phản ánh tải xử lý nhẹ trên GPU. Ngược lại, CPU được khai thác ở mức cao hơn, trung bình khoảng 40–60% đôi khi tới 80%, RAM hệ thống được khai thác ở mức ổn định trên 17%. Mô hình này phản ánh phần lớn khối lượng tính toán được xử lý trên CPU.

+Nhóm (4)- (CNN-LSTM , CNN-MLP):



Nhận xét : Đối với mô hình CNN-LSTM, GPU duy trì ổn định với nhiệt độ 49–50°C, tiêu thụ điện năng 64–70W và mức độ sử dụng khoảng 45–60%, thấp hơn LSTM thuần, CPU lại được khai thác tương đối nhiều dao động trung bình khoảng 35% nhiều lúc đạt trên 75%. Trong khi đó, CNN-MLP cho thấy mức sử dụng GPU thấp hơn một chút dao động trong khoảng 32–35%, còn mức độ sử dụng CPU tương tự CNN-LSTM khoảng 35%-60%, và RAM hệ thống trung bình khoảng 20-21%, cao hơn CNN-LSTM nhưng vẫn thấp hơn MLP thuần. Nhóm mô hình kết hợp cho thấy khả năng khai thác tài nguyên cân đối và hiệu quả hơn so với các mô hình thuần.

4.3.3. Kết quả trên kịch bản 3

4.3.3.1 Kết quả tiền xử lý

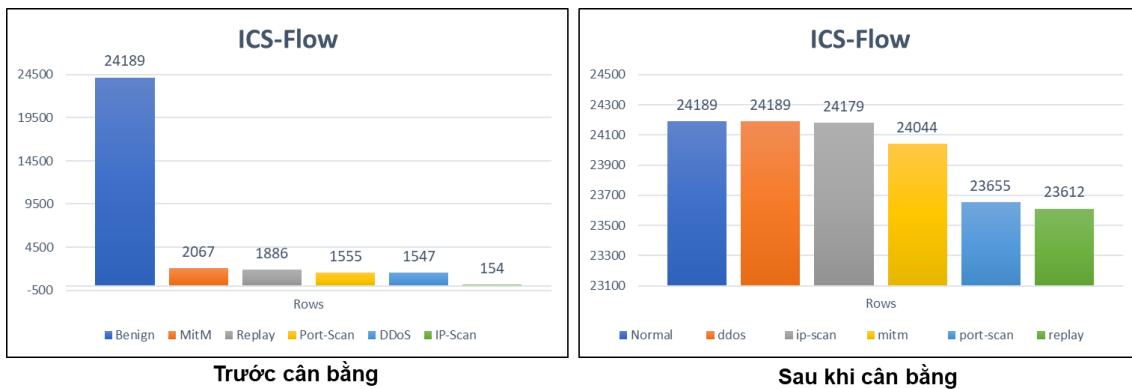
- Tập dữ liệu (3) : ICS-Flow
- Làm sạch dữ liệu:
 - + Các đặc trưng không đóng góp cho việc phân loại được loại bỏ là các thông tin định danh: sAddress, rAddress, sMACs, rMACs, sIPs, rIPs, protocol, startDate, endDate, start, end, startOffset, endOffset.
 - + Các đặc trưng chỉ có một giá trị duy nhất : sUrgRate, rUrgRate, sFragmentRate, rFragmentRate.
 - + Loại bỏ được nhiều dòng trùng nhau ở những dòng có nhãn Benign.
- Mã hoá các nhãn cho tập dữ liệu (3) với bài toán phân loại đa lớp và số lượng mẫu cho mỗi nhãn sau khi làm sạch dữ liệu :

Label_encoding	Attack Category	Rows
0	Normal(Benign)	30236
1	DDoS	1934
2	IP-Scan	192
3	MitM	2584
4	Port-Scan	1944
5	Replay	2358

- Các cột đặc trưng được thuật toán Random Forest lựa chọn là những cột đặc trưng có độ quan trọng (feature importance) > 0,01. Trong tập dữ liệu (3), sau khi áp dụng ngưỡng này, ta thu được tổng cộng 32 cột, bao gồm 30 cột đặc trưng vào và 2 cột đặc trưng mục tiêu “Label” và “Label_encode”, được thể hiện ở hình bên dưới :

```
['duration', 'sPackets', 'rPackets', 'sBytesSum', 'rBytesSum',
 'sBytesMax', 'sBytesMin', 'rBytesMin', 'sBytesAvg', 'rBytesAvg',
 'sLoad', 'rLoad', 'sPayloadSum', 'rPayloadSum', 'rPayloadMax',
 'rPayloadAvg', 'sInterPacketAvg', 'rInterPacketAvg', 'sttl', 'rttl',
 'rAckRate', 'rFinRate', 'rPshRate', 'rSynRate', 'rRstRate', 'sWinTCP',
 'rWinTCP', 'rAckDelayMax', 'sAckDelayAvg', 'rAckDelayAvg',
 'Label_encode', 'Label']
```

- Sau khi chia tập dữ liệu (3) thành 2 phần tập huấn luyện (80%) và tập kiểm tra (20%).
- Ta đem tập huấn luyện cân bằng dữ liệu với kỹ thuật **SMOTE+Tomek-Link**.
- Kết quả trước và sau khi cân bằng của tập huấn luyện trên tập dữ liệu (3) được thể hiện ở **Hình 33**.



Hình 33: Trước và sau khi cân bằng của tập dữ liệu (3) ICS-Flow.

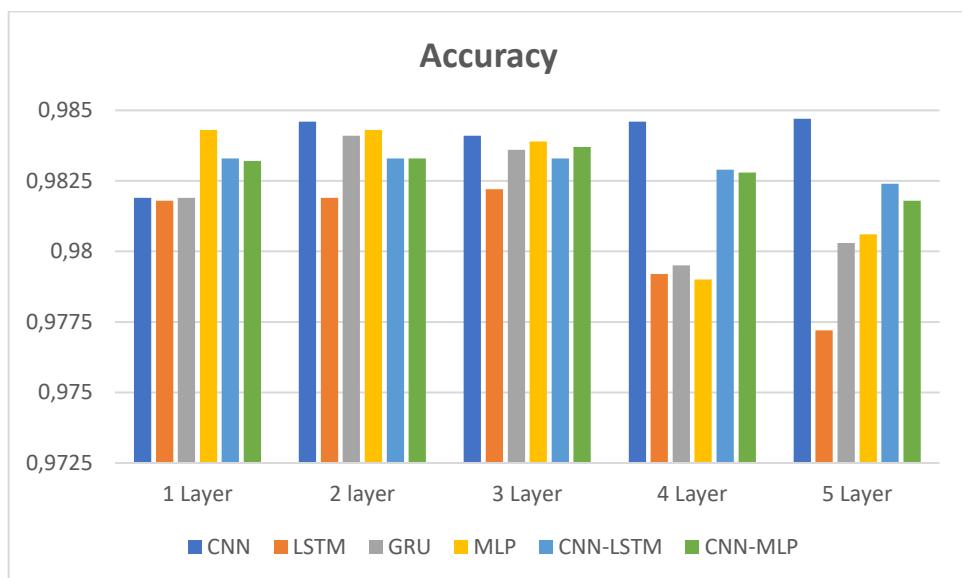
4.3.3.2 Kết quả huấn luyện

- Đánh giá tổng quát:

-Hình dưới đây trình bày kết quả huấn luyện tổng quát của các thuật toán, thể hiện qua độ chính xác (accuracy) :

Model	Accuracy				
	1 Layer	2 layer	3 Layer	4 Layer	5 Layer
CNN	0,9819	0,9846	0,9841	0,9846	0,9847
LSTM	0,9818	0,9819	0,9822	0,9792	0,9772
GRU	0,9819	0,9841	0,9836	0,9795	0,9803
MLP	0,9843	0,9843	0,9839	0,979	0,9806
CNN-LSTM	0,9833	0,9833	0,9833	0,9829	0,9824
CNN-MLP	0,9832	0,9833	0,9837	0,9828	0,9818

- Biểu đồ thể hiện sự so sánh độ chính xác (accuracy) tổng quát của các mô hình:

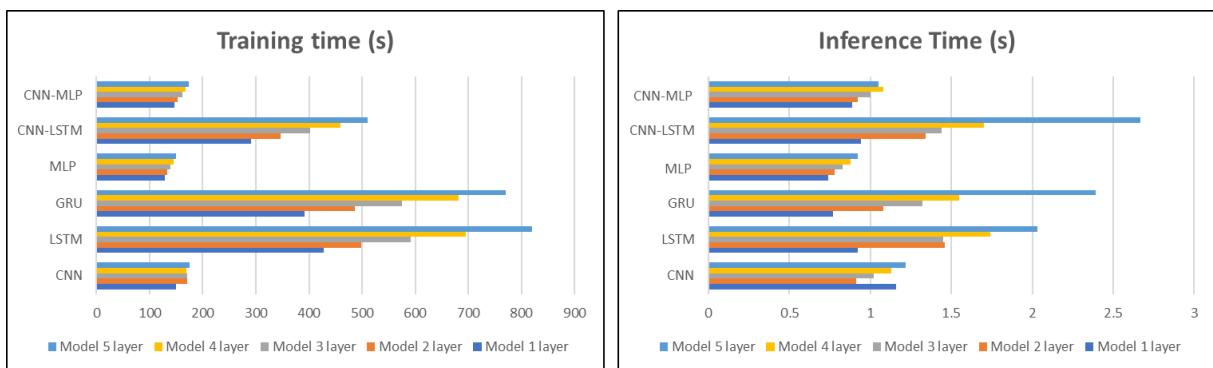


Hình 34: Biểu đồ chính xác (Accuracy) tổng quát của các mô hình trên kịch bản 3.

-Hình dưới đây trình bày tổng quát thời gian huấn luyện và thời gian suy luận :

Model	1 Layer		2 layer		3 Layer		4 Layer		5 Layer	
	Train Time (s)	Inference Time (s)								
CNN	149.14	1.16	171.33	0.91	170.21	1.02	169.14	1.13	175.75	1.22
LSTM	428.43	0.92	498.17	1.46	592.27	1.45	696.07	1.74	820.18	2.03
GRU	391.82	0.77	486.12	1.08	575.34	1.32	681.49	1.55	770.91	2.39
MLP	128.53	0.74	133.01	0.78	139.13	0.83	144.54	0.88	149.32	0.92
CNN-LSTM	291.72	0.94	346.53	1.34	401.72	1.44	459.51	1.7	510.13	2.67
CNN-MLP	146.39	0.89	152.42	0.92	161.3	1	168.24	1.08	174.2	1.05

-Biểu đồ thể hiện sự so sánh thời gian huấn luyện và thời gian suy luận tổng quát của các mô hình:



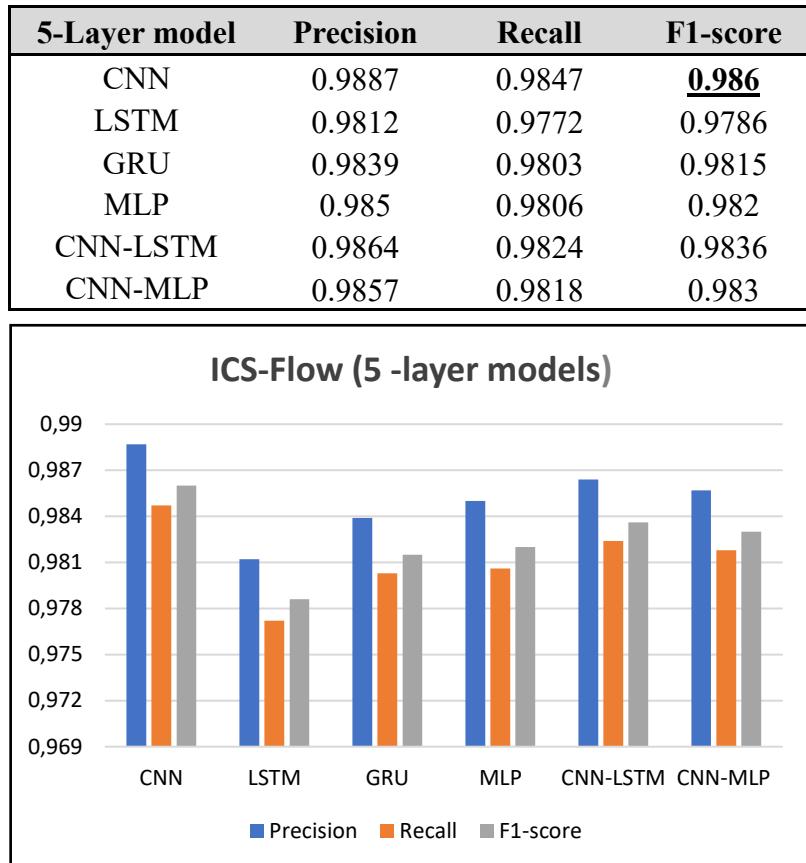
Hình 35: Biểu đồ thời gian huấn luyện và suy luận tổng quát của các mô hình trên kịch bản 3.

Nhận xét :

- Tất cả các mô hình đều đạt độ chính xác tương đối cao, dao động trong khoảng 0.981–0.985, có sự chênh lệch không quá lớn. Trong đó, CNN thể hiện sự nổi bật và ổn định nhất, đạt độ chính xác cao nhất ở nhiều cấu hình và cao nhất ở cấu hình 5 layer (0.9847). Ngoài ra, mô hình MLP cũng đạt kết quả khá tốt ở 3 cấu hình đầu và cao nhất ở cấu hình 1 layer (0.9843), nhưng độ ổn định kém hơn so với CNN khi số lớp tăng. Các mô hình kết hợp như CNN-LSTM và CNN-MLP duy trì hiệu suất khá cân bằng, với độ chính xác dao động trong khoảng 0.983–0.984. Ngược lại, LSTM và GRU lại có xu hướng suy giảm khi tăng số lớp.
- Quan sát thời gian huấn luyện có sự khác biệt lớn giữa các nhóm mô hình. CNN, MLP và CNN-MLP có thời gian huấn luyện ngắn nhất. Trong khi đó LSTM và GRU tốn nhiều thời gian hơn đáng kể ở mọi cấu hình do đặc trưng tuần tự, cụ thể là gấp khoảng 5-6 lần đối với các mô hình khác ở cùng cấu hình số lớp.
- Quan sát thời gian suy luận thì sự khác biệt giữa các mô hình là không quá lớn. MLP vẫn là mô hình cho tốc độ suy luận nhanh nhất dao đ từ 0.74-092 giây.

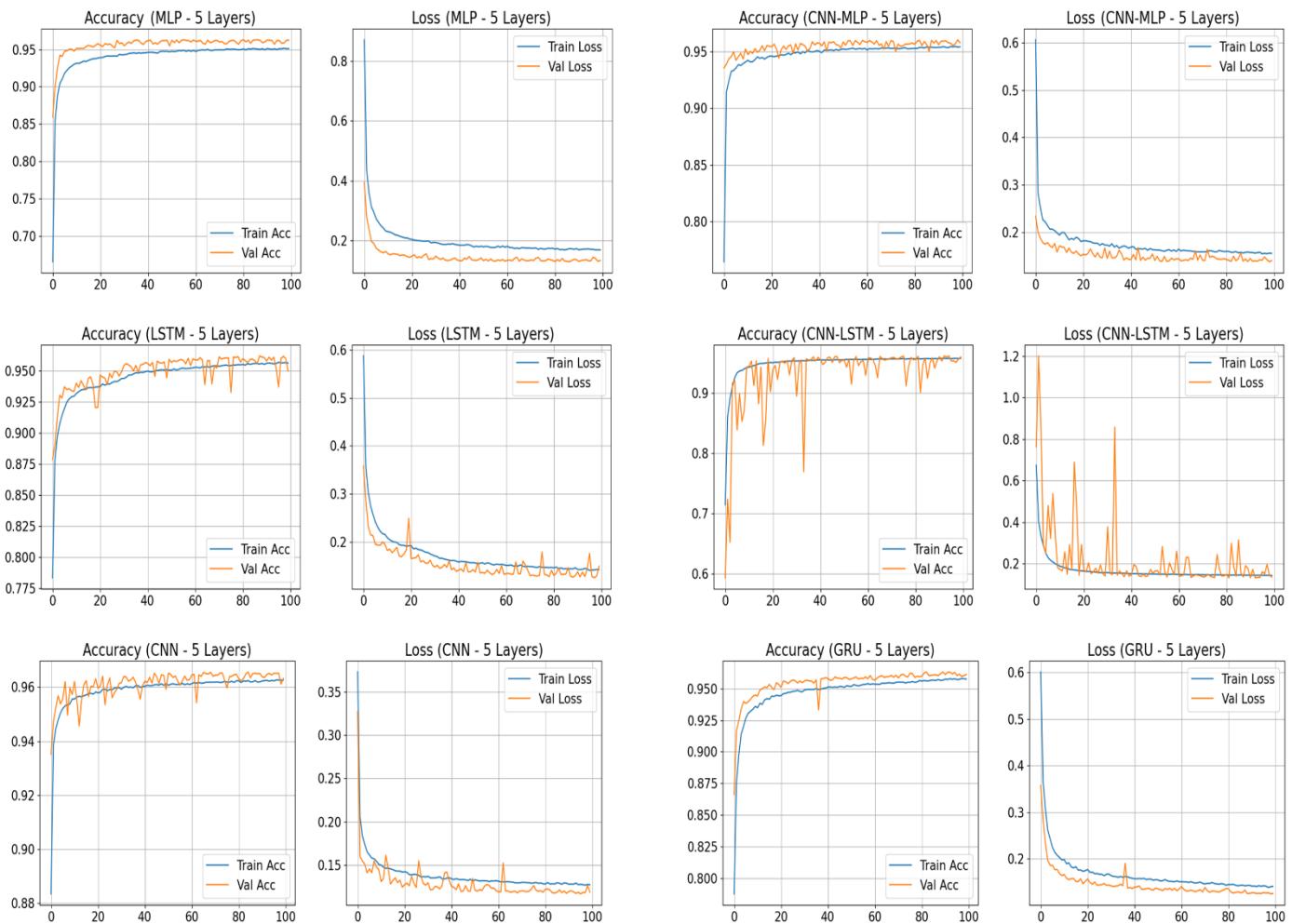
- **Đánh giá chi tiết:** Trình bày kết quả chi tiết đối với cấu hình gồm 5 lớp ẩn.

- Dưới đây là kết quả chi tiết thể hiện sự so sánh các chỉ số Precision, Recall, F1-score



Nhận xét : Nhìn chung, tất cả các mô hình đều đạt kết quả cao với Precision, Recall và F1-score đều trên 0.97, phản ánh khả năng phân loại lưu lượng mạng trên tập dữ liệu ICS-Flow ở mức rất tốt. Trong đó, CNN đạt hiệu suất nổi bật nhất với F1-score 0.986, thể hiện sự cân bằng giữa Precision (0.9887) và Recall (0.9847). Theo sau là CNN-LSTM và CNN-MLP với F1-score lần lượt 0.9836 và 0.983, cho thấy các mô hình kết hợp vẫn duy trì hiệu quả ổn định. GRU cũng đạt mức khá tốt với F1-score 0.9815, nhỉnh hơn LSTM (0.9786) nhờ Recall cao hơn. Ngược lại, MLP cho kết quả thấp nhất (F1-score 0.9828), mặc dù vẫn vượt ngưỡng 0.98 nhưng kém ổn định hơn so với CNN và các mô hình lai, phản ánh hạn chế trong việc khai thác đặc trưng phức tạp của dữ liệu.

-Để quan sát được quá trình học của các mô hình, hình dưới đây trình bày lịch sử huấn luyện (training history) của từng mô hình:



Hình 36: Lịch sử huấn luyện (training history) trên kịch bản 3.

Nhận xét :

+Nhóm (1) – (CNN):

CNN đạt độ chính xác khoảng gần 95–96% trên cả train và validation, đường cong Accuracy mượt, hội tụ nhanh và ổn định. Loss giảm đều về mức thấp (gần 0.15), train và validation bám sát nhau, không xuất hiện dấu hiệu overfitting.

+Nhóm (2) – (LSTM, GRU):

LSTM đạt độ chính xác khoảng gần 94–95%, GRU đạt gần 95% trên cả train và validation. Các đường Accuracy và Loss của train/validation gần như trùng nhau, Loss giảm nhanh và ổn định (LSTM gần 0.25, GRU gần 0.2), không xuất hiện dấu hiệu overfitting. GRU có tốc độ hội tụ nhanh và ổn định hơn, trong khi LSTM có độ dao động nhẹ ở validation nhưng vẫn duy trì kết quả tốt, phù hợp cho dữ liệu chuỗi.

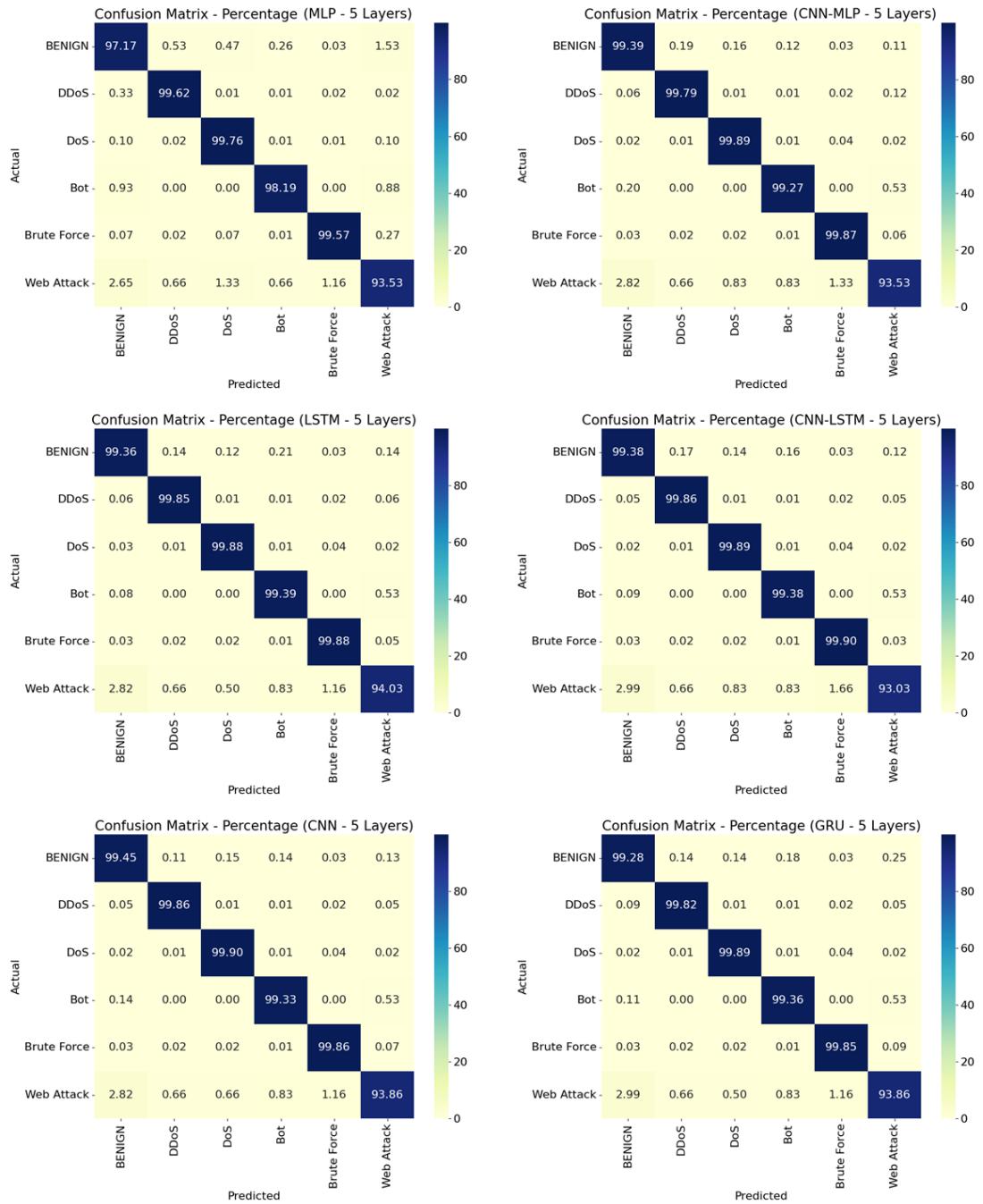
+Nhóm (3) – (MLP):

MLP đạt khoảng gần 95% trên train và gần 94.5% trên validation. Accuracy tăng nhanh, đường cong khá mượt, Loss giảm ổn định về mức gần 0.15, train và validation bám sát nhau, ít dao động.

+Nhóm (4) – (CNN-MLP, CNN-LSTM):

CNN-MLP đạt độ chính xác gần 95–95.5% trên train và validation, Loss giảm ổn định về mức gần 0.15–0.2, cả hai đường bám sát nhau, thể hiện tính ổn định tốt. CNN-LSTM đạt Accuracy khoảng gần 90–91%, nhưng Validation dao động mạnh, Loss biến thiên nhiều, đặc biệt là val loss, cho thấy mô hình chưa ổn định, có thể cần huấn luyện thêm.

-Để đánh giá khả năng phân loại của từng mô hình, các hình dưới đây trình bày ma trận nhầm lẫn (confusion matrix) tương ứng cho từng trường hợp:

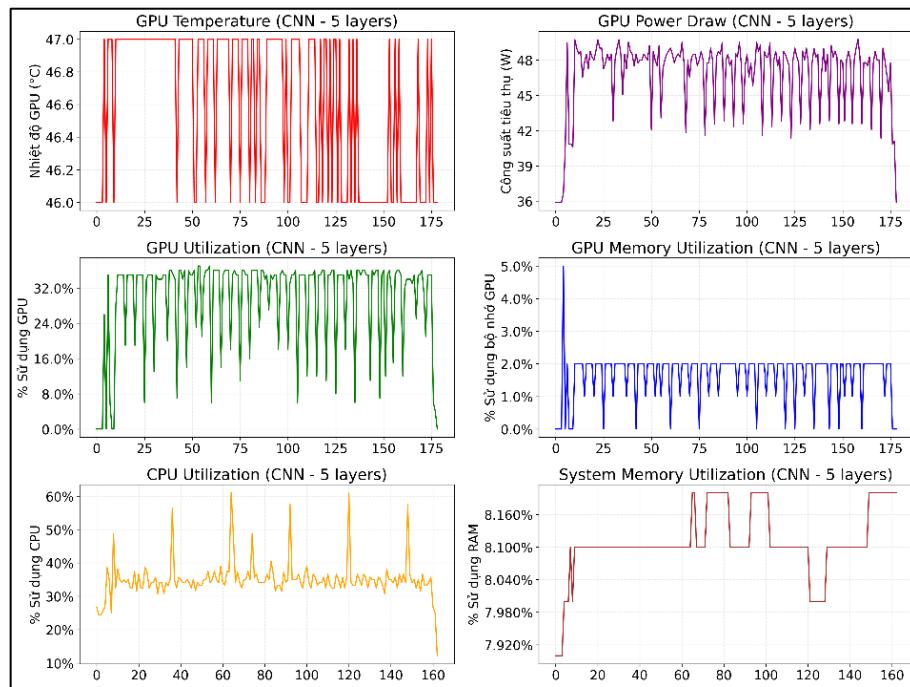


Hình 37: Ma trận nhầm lẫn (confusion matrix) trên kích thước 3.

Nhận xét : Nhìn chung, tất cả các mô hình đều đạt hiệu suất cao trong việc phân loại các loại tấn công Normal, DDoS, DoS, Bot, Port Scan, Replay với độ chính xác dao động từ 95% đến gần 100%. Các mô hình có xu hướng phân loại tốt nhất đối với các lớp Normal và DoS/DDoS, đạt trên 98–100% ở hầu hết các trường hợp. Tuy nhiên, vẫn còn xuất hiện nhầm lẫn ở các lớp Port Scan và Replay, do đặc trưng gần giống nhau, dẫn đến độ chính xác thấp hơn so với các lớp còn lại.

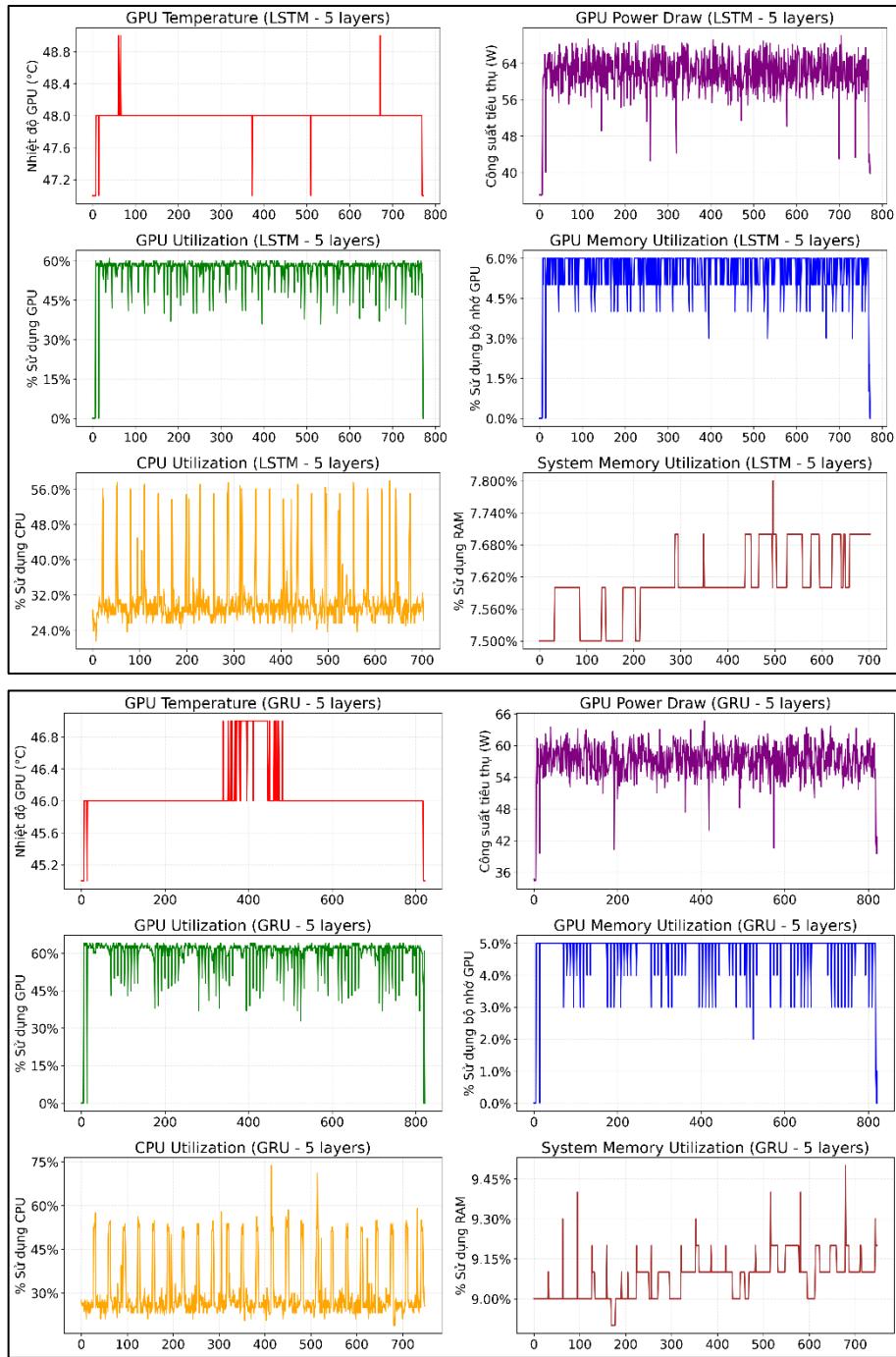
-Để đánh giá mức độ sử dụng tài nguyên CPU và GPU các hình dưới đây trình bày biểu đồ tương ứng cho các trường hợp:

+Nhóm (1) – (CNN):



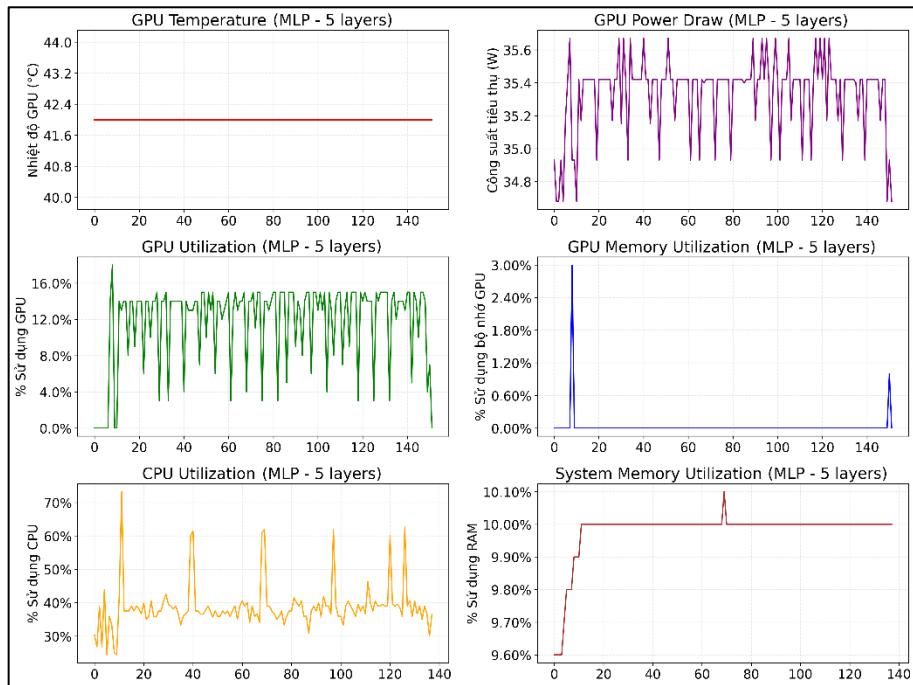
Nhận xét: Nhóm này cho thấy tận dụng hiệu quả khả năng xử lý của GPU, thể hiện qua mức sử dụng GPU ổn định quanh 30-32%, công suất tiêu thụ khoảng 40–48W, nhiệt độ GPU duy trì ở mức 48°C, bộ nhớ GPU được khai thác khoảng 2%, CPU hoạt động ở mức trung bình 35–40% và có thời điểm đạt đỉnh lên tới 60%, và RAM hệ thống ổn định quanh 8.1%.

+Nhóm (2) – (LSTM, GRU):



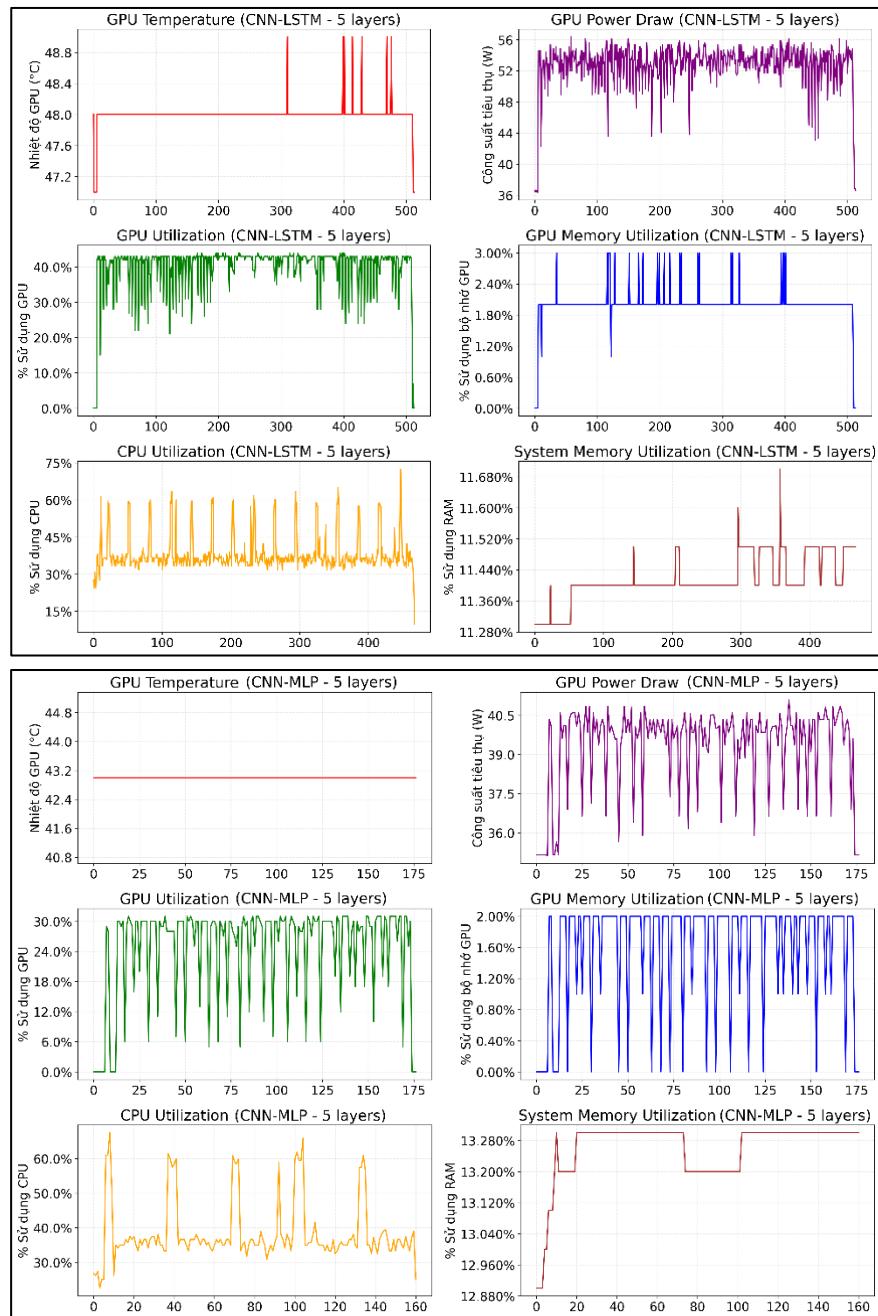
Nhận xét: Nhóm này được ghi nhận là tiêu thụ nhiều năng lượng nhất trong quá trình huấn luyện. Công suất tiêu thụ GPU rất cao, thường xuyên trên 60W. Mức sử dụng GPU duy trì ổn định khoảng 59-62%, bộ nhớ GPU duy trì ở mức hơn 5-6%. CPU cũng hoạt động ở cường độ cao với biến động lớn dao động ở mức % có nhiều biến động đạt tới 56%, và RAM hệ thống dao động tăng nhẹ theo thời gian. Nhóm này phản ánh rõ khối lượng tính toán lớn và yêu cầu xử lý song song cao giữa CPU và GPU.

+Nhóm (3) – (MLP):



Nhận xét: Nhóm này cho thấy hầu như không tận dụng khả năng xử lý của GPU, cho thấy mức tiêu thụ tài nguyên ở mức khá thấp. Công suất GPU duy trì quanh 35–36W, không xuất hiện các đột biến lớn. Nhiệt độ GPU giữ ở mức rất ổn định 42°C. Mức sử dụng GPU dao động trong khoảng 10–14%, trong khi bộ nhớ GPU hầu như không được khai thác. Ngược lại, CPU được khai thác ở mức cao hơn, trung bình khoảng 40–60% đôi khi tới 70%, trong khi RAM hệ thống duy trì ổn định ở mức 10%. Mô hình này phản ánh phần lớn khối lượng tính toán được xử lý trên CPU.

+Nhóm (4)- (CNN-LSTM ,CNN-MLP):



Nhận xét : Đối với mô hình CNN-LSTM, GPU duy trì ổn định với nhiệt độ 48°C, tiêu thụ điện năng 52–54W và mức độ sử dụng trung bình khoảng 42%, thấp hơn LSTM thuần, CPU lại được khai thác tương đối nhiều dao động trung bình khoảng 32-60% nhiều lúc đạt 75%. Trong khi đó, CNN-MLP cho thấy mức sử dụng GPU thấp hơn một chút dao động trong khoảng 30%, còn mức độ sử dụng CPU tương tự CNN-LSTM và RAM hệ thống trung bình khoảng 13.2%, cao hơn CNN-LSTM nhưng vẫn thấp hơn MLP thuần. Nhóm mô hình kết hợp cho thấy khả năng khai thác tài nguyên cân đối và hiệu quả hơn so với các mô hình thuần.

4.4. . Đánh giá tổng hợp

4.4.1. Đánh giá tổng hợp các kịch bản

- **Mạng Doanh nghiệp (Kịch bản 1 - CSE-CIC-IDS-2018):** Đây là môi trường mà tất cả các mô hình đều đạt hiệu suất cao nhất, với độ chính xác vượt trội đều trên 99%. Điều này cho thấy dữ liệu từ mạng doanh nghiệp có các đặc trưng rõ ràng, giúp các mô hình dễ dàng phân biệt giữa lưu lượng bình thường và tấn công.
- **Mạng IoT (Kịch bản 2 - CIC-IoT-2023):** Đây là môi trường thách thức nhất. Độ chính xác của các mô hình giảm xuống đáng kể, nằm trong khoảng 92% đến 96% xuất hiện nhiều trường hợp nhầm lẫn giữa các lớp. Điều này phản ánh tính chất phức tạp, đa dạng và nhiều nhiễu của lưu lượng mạng IoT, đòi hỏi các mô hình phải có khả năng nắm bắt các đặc trưng tinh vi hơn.
- **Mạng Công nghiệp (Kịch bản 3 - ICS-Flow):** Mặc dù quy mô tập dữ liệu là nhỏ hơn hai tập dữ liệu còn lại, kết quả đánh giá có thể phần nào chịu ảnh hưởng từ tập kiểm tra (testing dataset) hạn chế về số lượng mẫu. Tuy nhiên, các mô hình học sâu vẫn đạt độ chính xác khá cao 97,72% – 98,5% và chưa ghi nhận dấu hiệu quá khớp rõ rệt, cho thấy khả năng khai thác tốt đặc trưng dữ liệu. Đây được xem là một điểm cộng đáng ghi nhận, bởi nó minh chứng rằng ngay cả trong điều kiện dữ liệu giới hạn, học sâu vẫn giữ được tính hiệu quả và tiềm năng ứng dụng trong phát hiện xâm nhập ở môi trường ICS.

Tổng hợp ba kịch bản trên cho thấy hiệu quả của các mô hình học sâu chịu ảnh hưởng mạnh từ đặc thù môi trường mạng. Trong khi dữ liệu doanh nghiệp mang lại độ chính xác vượt trội nhờ đặc trưng rõ ràng, thì môi trường IoT đặt ra thách thức lớn do tính phức tạp và nhiễu cao, còn môi trường công nghiệp dù hạn chế về dữ liệu vẫn cho thấy tiềm năng áp dụng ổn định. Điều này khẳng định tầm quan trọng của việc lựa chọn kiến trúc mô hình phù hợp và tùy chỉnh chiến lược huấn luyện theo đặc thù từng loại mạng.

4.4.2. Đánh giá tổng hợp hiệu quả của từng kiến trúc mô hình

- **Mô hình CNN:** Mô hình mạng nơ-ron tích chập này thể hiện khả năng trích xuất đặc trưng hiệu quả, duy trì hiệu quả cao và ổn định trên nhiều loại dữ liệu, từ tập nhỏ như ICS-Flow đến các tập lớn và phức tạp hơn. Nhờ cơ chế trích xuất theo tầng, CNN có thể khai thác tốt cả đặc trưng bề mặt lẫn đặc trưng sâu, đặc biệt khi số lớp được thiết kế hợp lý. Tuy nhiên ở cấu hình 1-layer, hiệu suất của CNN thấp hơn rõ rệt so với các cấu hình từ 2-layer trở lên, nhiều khả năng do hiện tượng học nồng – số lớp quá ít chỉ cho phép mô hình nắm bắt các đặc trưng bề mặt, chưa đủ khả năng trích xuất và kết hợp các đặc trưng phức tạp. Khi tăng số lớp lên mức phù hợp, CNN tận dụng hiệu quả hơn các tầng học biểu diễn, giúp cải thiện rõ rệt độ chính xác và chứng minh vai trò quan trọng trong các kịch bản phân tích dữ liệu mạng.
- **Mô hình LSTM và GRU :** Đây là nhóm kiến trúc mạng hồi quy (RNN) luôn duy trì độ chính xác cao, thường xuyên giữ vị trí thứ hai hoặc thứ ba trong các kịch bản thử nghiệm. Kết quả này phản ánh rõ khả năng học và khai thác đặc trưng theo chuỗi thời gian cũng như năng lực tổng quát hóa mạnh mẽ, đặc biệt phát huy hiệu quả trong các môi trường dữ liệu có tính tuần tự cao như IoT. Tuy vậy, hạn chế đáng kể của LSTM và GRU là thời gian huấn luyện tương đối dài do đặc thù tính toán tuần tự theo từng bước thời gian, dẫn đến tốc độ xử lý chậm hơn và ảnh hưởng đến tính khả thi khi triển khai trên các tập dữ liệu lớn hoặc yêu cầu thời gian thực.
- **Mô hình MLP:** mô hình này thuộc nhóm mạng nơ-ron truyền thẳng, cho thấy hiệu quả không ổn định ở một số cấu hình 1-Layer có thể đạt kết quả cao, nhưng khi tăng số layer thì hiệu suất lại giảm, cho thấy mô hình chưa tận dụng tốt độ sâu mạng để cải thiện khả năng học. Khả năng phân loại nhìn chung chỉ ở mức tương đối tốt, chưa đạt đến độ chính xác và độ ổn định như các mô hình CNN hoặc các kiến trúc kết hợp, đặc biệt khi xử lý các đặc trưng phức tạp của dữ liệu.
- **Nhóm các mô hình kết hợp CNN-LSTM và CNN-MLP** cho thấy hiệu quả vượt trội so với các mô hình đơn lẻ. Cụ thể, CNN-LSTM không chỉ đạt độ chính xác cao hơn so với việc sử dụng riêng lẻ CNN hoặc LSTM, mà còn rút ngắn thời gian huấn luyện của LSTM xuống khoảng một nửa nhờ CNN đảm nhiệm giai đoạn trích xuất đặc trưng ban đầu. Trong khi đó, CNN-MLP khai thác thế mạnh của

CNN trong việc trích xuất đặc trưng không gian, giúp nâng hiệu suất phân loại lên đáng kể so với MLP thuần, đồng thời duy trì thời gian huấn luyện hợp lý. Điều này chứng tỏ các kiến trúc kết hợp có khả năng tận dụng ưu điểm của từng thành phần để cải thiện cả độ chính xác lẫn hiệu quả tính toán.

Từ kết quả thực nghiệm các mô hình học sâu trên ta thấy được dù cùng số lớp và cùng số nơ-ron, các mô hình cho thấy hiệu quả và độ ổn định rất khác biệt. Sự khác biệt này không nằm ở quy mô hay độ phức tạp về mặt số lượng, mà bắt nguồn từ kiến trúc và cơ chế học đặc trưng vốn có của chúng. Mỗi kiến trúc được thiết kế để giải quyết một loại vấn đề cụ thể: CNN vượt trội trong việc nhận dạng các mẫu không gian theo tầng, LSTM/GRU lại mạnh về việc nắm bắt các mối quan hệ phụ thuộc theo thời gian, trong khi MLP truyền thống chỉ đơn thuần học các mối liên hệ phi tuyến tính mà không có sự chuyên biệt hóa. Vì vậy, lựa chọn kiến trúc phù hợp với bản chất dữ liệu quan trọng hơn việc chỉ tăng quy mô mô hình. Các mô hình kết hợp minh chứng rõ điều này khi tận dụng ưu điểm từng thành phần để đạt hiệu quả tốt hơn.

4.4.3. Đánh giá tổng hợp mức độ sử dụng tài nguyên của các mô hình

- **Mô hình CNN:** mô hình này tận dụng GPU hiệu quả nhờ khả năng xử lý song song, CNN vận hành ổn định với mức tải đều, giữ nhiệt độ và công suất an toàn. CPU và RAM chỉ đóng vai trò hỗ trợ nhẹ. Thời gian huấn luyện nhanh–trung bình, trong khi suy luận rất nhanh và ổn định, phù hợp cho yêu cầu thời gian thực. Nhìn chung, CNN cân bằng tốt giữa độ chính xác, tốc độ và chi phí, thích hợp triển khai trên GPU phổ thông trong môi trường mạng doanh nghiệp hoặc giám sát an ninh thời gian thực.
- **Mô hình LSTM và GRU:** Đây là nhóm tiêu thụ tài nguyên nặng nhất. Mặc dù GRU đã được tinh gọn hơn so với LSTM, song do đặc thù xử lý tuần tự, cả hai mô hình đều buộc GPU hoạt động ở mức tải rất cao, thường xuyên chạm ngưỡng tối đa, kéo theo công suất tiêu thụ và nhiệt độ tăng mạnh, CPU chịu nhiều đinh tải, còn RAM tăng dần theo thời gian huấn luyện. Đặc trưng này phản ánh bản chất RNN vốn nặng tính toán tuần tự, khiến chi phí huấn luyện lớn nhưng bù lại khả năng mô hình hóa quan hệ chuỗi mạnh mẽ, phù hợp với các hệ thống cần phân tích dữ liệu theo thời gian dài hạn.
- **Mô hình MLP:** Đây là kiến trúc tiêu tốn ít tài nguyên nhất, với các phép tính chủ yếu dựa trên nhân ma trận – vector nên đơn giản và dễ tối ưu hóa. Nhờ đặc điểm này, MLP gần như không gây áp lực lên GPU, gánh nặng xử lý được chuyển phần lớn sang CPU và RAM, đó là do cấu trúc đơn giản nên mức sử dụng các tài nguyên này khiêm tốn. Thời gian huấn luyện nhanh và suy luận cũng rất nhanh, tạo lợi thế trong các ứng dụng yêu cầu phản hồi tức thì. Tuy nhiên sự đơn giản này có thể phải trả giá bằng hiệu năng kém ổn định hơn so với các mô hình phức tạp khác. Kiến trúc này thích hợp triển khai trong môi trường hạn chế tài nguyên như IoT, mạng cảm biến hoặc hệ thống bảo mật nhỏ gọn.
- **Nhóm mô hình kết hợp (CNN–MLP, CNN–LSTM):** Tiêu thụ tài nguyên ở mức trung bình, cao hơn CNN nhưng thấp hơn LSTM. CNN–MLP giữ sự gọn nhẹ của CNN và cải thiện độ chính xác nhờ tầng phân loại MLP mà không tăng nhiều chi phí. CNN–LSTM tận dụng CNN để giảm tải cho LSTM, rút ngắn thời gian huấn luyện và duy trì năng lượng ở mức vừa phải. Nhờ đó, nhóm mô hình này vừa đạt độ chính xác cao vừa tối ưu tốc độ và tài nguyên, phù hợp triển khai trong hệ thống mạng quy mô lớn, trung tâm dữ liệu hoặc giám sát an ninh thời gian thực.

Bảng dưới đây tổng hợp mức độ sử dụng tài nguyên của các mô hình đã phân tích:

Mô hình	Mức độ sử dụng tài nguyên	Đặc điểm nổi bật	Thời gian xử lý	Ứng dụng phù hợp
CNN	Trung bình	- Tận dụng GPU hiệu quả, ổn định	- Huấn luyện: Trung bình đến nhanh - Suy luận: nhanh	- Môi trường có GPU phổ thông đến mạnh - Hệ thống giám sát thời gian thực
LSTM và GRU	Cao	- GPU và CPU hoạt động ở mức rất cao	- Huấn luyện: Chậm - Suy luận: Trung bình	- Phân tích chuỗi sự kiện dài hạn
MLP	Rất thấp	- Chủ yếu sử dụng CPU, không gây áp lực lên GPU	- Huấn luyện: Rất nhanh - Suy luận: Rất nhanh	- Môi trường hạn chế tài nguyên (IoT) - Ứng dụng yêu cầu phản hồi nhanh.
Mô hình kết hợp	Trung bình	- Cân bằng giữa hiệu quả và chi phí - Giảm thời gian huấn luyện	- Huấn luyện: Trung bình - Suy luận: Nhanh	- Mạng quy mô lớn, trung tâm dữ liệu - Cần cân bằng giữa hiệu năng và tài nguyên

Bảng 10: Tổng hợp đánh giá mức độ sử dụng tài nguyên của các mô hình.

CHƯƠNG 5: KẾT LUẬN

5.1. Kết quả đạt được

- Luận văn đã hoàn thành mục tiêu đề ra là đánh giá chi tiết hiệu quả của các kỹ thuật học sâu tiêu biểu trong bài toán phát hiện xâm nhập trên các môi trường mạng không đồng nhất. Quá trình thực nghiệm đã triển khai và đánh giá 6 kiến trúc học sâu là CNN, LSTM, GRU, MLP, CNN-LSTM, CNN-MLP trên ba môi trường mạng không đồng nhất là Mạng doanh nghiệp, Mạng IoT, Mạng công nghiệp với ba tập dữ liệu tương ứng là CSE-CIC-IDS-2018, CIC-IoT-2023, ICS-Flow với quy trình tiền xử lý và cân bằng dữ liệu nhất quán. Các mô hình được thiết kế với nhiều cấu hình khác nhau về số lượng lớp ẩn và nơ-ron, sau đó các mô hình được huấn luyện và đánh giá dựa trên các chỉ số thông thường như Accuracy, Precision, Recall, F1-score và còn đo lường cả hiệu quả tính toán thông qua mức tiêu thụ tài nguyên GPU/CPU.
- Kết quả thực nghiệm khẳng định hiệu suất vượt trội của các mô hình học sâu, với độ chính xác hầu hết đều trên 94% ở cả ba môi trường. Trong đó, CNN nổi bật nhờ sự ổn định, hiệu năng cao và cân bằng tốt giữa tốc độ cùng độ chính xác. Nhóm mô hình hồi quy (RNN) gồm LSTM và GRU cho thấy ưu thế rõ rệt trong xử lý dữ liệu tuần tự, song phải đánh đổi bằng chi phí tài nguyên và thời gian huấn luyện lớn, trong đó GRU tiêu thụ ít tài nguyên hơn LSTM nhưng vẫn cao hơn hầu hết các mô hình khác. MLP là kiến trúc gọn nhẹ nhất, song hiệu suất kém ổn định và thường thấp hơn khi xử lý dữ liệu phức tạp. Các mô hình kết hợp như CNN-LSTM và CNN-MLP thể hiện sự vượt trội rõ rệt; trong đó CNN-LSTM được xem là toàn diện nhất, vừa đạt độ chính xác cao trên nhiều cấu hình, vừa giảm đáng kể gánh nặng tính toán so với LSTM thuần túy.
- Đánh giá đồng thời hiệu quả và mức độ sử dụng tài nguyên của mô hình học sâu mang lại cái nhìn toàn diện về ưu, nhược điểm của từng kiến trúc, tạo nền tảng cho việc tối ưu hóa và triển khai trong các môi trường hạn chế tài nguyên từ IoT, ngôi nhà thông minh, Drone đến trung tâm dữ liệu, đảm bảo hiệu suất cao và tính ứng dụng thực tiễn.

5.2. Hạn chế

Mặc dù đã nỗ lực để đạt được các mục tiêu đề ra, luận văn vẫn còn một số hạn chế nhất định:

- Giới hạn về dữ liệu: Tập dữ liệu ICS-Flow có quy mô nhỏ hơn đáng kể so với hai tập còn lại, điều này có thể ảnh hưởng đến khả năng tổng quát hóa của các mô hình khi đánh giá trên môi trường công nghiệp. Hơn nữa, các bộ dữ liệu dù thực tế nhưng vẫn là dữ liệu offline, chưa phản ánh hết được sự biến động và độ trễ của một hệ thống mạng thời gian thực.
- Phạm vi tối ưu hóa mô hình: Luận văn này chỉ tập trung vào việc so sánh các kiến trúc dưới cùng một cấu hình chung về số lớp và nơ-ron để đảm bảo tính công bằng. Việc tinh chỉnh sâu hơn các siêu tham số (hyperparameter tuning) cho từng mô hình riêng lẻ và chọn ra cấu hình tốt nhất có thể giúp cải thiện hơn nữa hiệu suất của chúng và chọn được mô hình tốt và phù hợp nhất để ứng dụng thực tiễn.
- Môi trường triển khai: Các thử nghiệm chỉ thực hiện trên Kaggle với GPU mạnh, chưa đánh giá trên thiết bị tài nguyên hạn chế như gateway IoT hay máy tính nhúng.

5.3. Hướng phát triển

Từ những hạn chế trên, một số hướng nghiên cứu tiếp theo có thể được đề xuất:

- Mở rộng thu thập dữ liệu từ nhiều nguồn thực tế, đồng thời triển khai thử nghiệm trong các môi trường hạn chế tài nguyên nhằm phản ánh sát hơn điều kiện ứng dụng. Nghiên cứu áp dụng các kiến trúc tối ưu hiệu năng và thử nghiệm với mô hình học sâu mới để nâng cao hiệu suất cũng như khả năng thích ứng.
- Nghiên cứu Học liên tục (Continual Learning) để giúp mô hình tự cập nhật và thích nghi với các kiểu tấn công mới mà không cần huấn luyện lại từ đầu.
- Nghiên cứu tính giải thích của mô hình (Explainable AI – XAI): Phát triển các cơ chế giúp giải thích quyết định của mô hình IDS, từ đó tăng độ tin cậy và khả năng áp dụng trong môi trường công nghiệp yêu cầu tính minh bạch cao.

TÀI LIỆU THAM KHẢO

- [1] “IBM Report: Escalating Data Breach Disruption Pushes Costs to New Highs,” IBM Newsroom.
Available: <https://newsroom.ibm.com/2024-07-30-ibm-report-escalating-data-breach-disruption-pushes-costs-to-new-highs>
- [2] D. E. Denning, “An Intrusion-Detection Model,” *IEEE Trans. Softw. Eng.*, vol. SE-13, no. 2, pp. 222–232, Feb. 1987, doi: 10.1109/TSE.1987.232894.
- [3] L. Ashiku and C. Dagli, “Network Intrusion Detection System using Deep Learning,” *Procedia Comput. Sci.*, vol. 185, pp. 239–247, Jan. 2021, doi: 10.1016/j.procs.2021.05.025.
- [4] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, “Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study,” *J. Inf. Secur. Appl.*, vol. 50, p. 102419, Feb. 2020, doi: 10.1016/j.jisa.2019.102419.
- [5] A. M. Amine and E. Y. I. Khamlich, “Advancing Intrusion Detection: Application of Distributed Deep Learning on the KDD Cup 99 Dataset,” *Int. J. Electron. Commun. Eng.*, vol. 11, no. 6, pp. 107–113, June 2024, doi: 10.14445/23488549/IJECE-V11I6P109.
- [6] Quân L. A., Quang T. M., Khải L. P., Nguyễn T. T., and Cang P. T., “Giải pháp phát hiện xâm nhập mạng sử dụng mô hình học sâu,” *Tạp Chí Khoa Học Đại Học Cần Thơ*, vol. 61, no. 3, Art. no. 3, June 2025, doi: 10.22144/ctujos.2025.093.
- [7] A. A. A. Mohammed, “Improving Intrusion Detection Systems by using Deep Learning Methods on Time Series Data,” *Eng. Technol. Appl. Sci. Res.*, vol. 15, no. 1, Art. no. 1, Feb. 2025, doi: 10.48084/etasr.9417.
- [8] H. Asgharzadeh, A. Ghaffari, M. Masdari, and F. S. Gharehchopogh, “An Intrusion Detection System on The Internet of Things Using Deep Learning and Multi-objective Enhanced Gorilla Troops Optimizer,” *J. Bionic Eng.*, vol. 21, no. 5, pp. 2658–2684, Sept. 2024, doi: 10.1007/s42235-024-00575-7.
- [9] A. Sagu, N. S. Gill, P. Gulia, N. Alduaiji, P. K. Shukla, and M. A. Shah, “Advances to IoT security using a GRU-CNN deep learning model trained on SUCMO algorithm,” *Sci. Rep.*, vol. 15, no. 1, p. 16485, May 2025, doi: 10.1038/s41598-025-99574-9.
- [10] F. A. Alotaibi and S. Mishra, “Cyber Security Intrusion Detection and Bot Data Collection using Deep Learning in the IoT,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 15, no. 3, 2024, doi: 10.14569/IJACSA.2024.0150343.
- [11] S. A. Bakhsh, M. A. Khan, F. Ahmed, M. S. Alshehri, H. Ali, and J. Ahmad, “Enhancing IoT network security through deep learning-powered Intrusion Detection System,” *Internet Things*, vol. 24, p. 100936, Dec. 2023, doi: 10.1016/j.IoT.2023.100936.
- [12] H. A. Le, L. D. A. Tran, S. H. Hoang, and T. H. Nguyen, “Advanced Machine Learning and Deep Learning Techniques for Anomaly Detection in Industrial Control System,” *JST Smart Syst. Devices*, vol. 34, no. 3, Art. no. 3, Sept. 2024, doi: 10.51316/jst.176.ssad.2024.34.3.2.
- [13] M. Bozdal, K. Ileri, and A. Ozhakraman, “Comparative Analysis of Dimensionality Reduction Techniques for Cybersecurity in the SWaT Dataset,” May 10, 2023, *Research Square*. doi: 10.21203/rs.3.rs-2904250/v1.

- [14] R. Chinnasamy, M. Subramanian, S. V. Easwaramoorthy, and J. Cho, “Deep learning-driven methods for network-based intrusion detection systems: A systematic review,” *ICT Express*, vol. 11, no. 1, pp. 181–215, Feb. 2025, doi: 10.1016/j.icte.2025.01.005.
- [15] “Fig. 2. Representation of an enterprise network consisting of a...,” ResearchGate. Available: https://www.researchgate.net/figure/Representation-of-an-enterprise-network-consisting-of-a-three-tier-IDS-and-three-local_fig2_369432566
- [16] S. Mahadik, P. M. Pawar, and R. Muthalagu, “Efficient Intelligent Intrusion Detection System for Heterogeneous Internet of Things (HetIoT),” *J. Netw. Syst. Manag.*, vol. 31, no. 1, p. 2, Oct. 2022, doi: 10.1007/s10922-022-09697-x.
- [17] “Figure 1: Traditional ICS network diagram.,” ResearchGate. Available: https://www.researchgate.net/figure/Traditional-ICS-network-diagram_fig1_347628957
- [18] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.
- [19] “(PDF) A Survey of Deep Learning Algorithms and its Applications,” ResearchGate, Aug. 2025, doi: 10.21608/njccs.2022.139054.1000.
- [20] “Deep Learning vs. Machine Learning for Intrusion Detection in Computer Networks: A Comparative Study.” Available: <https://www.mdpi.com/2076-3417/15/4/1903>
- [21] “(PDF) The Role of GPU Architecture in Accelerating Deep Learning Workloads,” ResearchGate. Available: https://www.researchgate.net/publication/393787841_The_Role_of_GPU_Architecture_in_Accelerating_Deep_Learning_Workloads
- [22] “Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions.” Available: <https://www.mdpi.com/2079-3197/11/3/52>
- [23] H. Kaur, S. Bansal, M. Kumar, A. Mittal, and K. Kumar, “Worddeepnet: handwritten gurumukhi word recognition using convolutional neural network,” *Multimed. Tools Appl.*, vol. 82, no. 30, pp. 46763–46788, Dec. 2023, doi: 10.1007/s11042-023-15527-2.
- [24] “Fig. 10. A comparison between RNN neuron and an LSTM neuron,” ResearchGate. Available: https://www.researchgate.net/figure/A-comparison-between-RNN-neuron-and-an-LSTM-neuron_fig3_363568278
- [25] F. Pan *et al.*, “Stacked-GRU Based Power System Transient Stability Assessment Method,” *Algorithms*, vol. 11, no. 8, p. 121, Aug. 2018, doi: 10.3390/a11080121.
- [26] S. Lee *et al.*, “Multi-layer Perceptron Approach for Prediction of Heating Energy Consumption in Old Houses,” *Energies*, vol. 14, no. 1, p. 122, Jan. 2021, doi: 10.3390/en14010122.
- [27] “IDS 2018 || Datasets | Research | Canadian Institute for Cybersecurity | UNB.” Available: <https://www.unb.ca/cic/datasets/ids-2018.html>
- [28] “IoT Dataset 2023 | Datasets | Research | Canadian Institute for Cybersecurity | UNB.” Available: <https://www.unb.ca/cic/datasets/IoTdataset-2023.html>
- [29] A. Dehlaghi-Ghadim, M. H. Moghadam, A. Balador, and H. Hansson, “Anomaly Detection Dataset for Industrial Control Systems,” May 11, 2023, *arXiv*: arXiv:2305.09678. doi: 10.48550/arXiv.2305.09678.
- [30] “7.3. Preprocessing data,” scikit-learn.

- Available: <https://scikit-learn/stable/modules/preprocessing.html>
- [31] A. D. Vibhute and V. Nakum, “Deep learning-based network anomaly detection and classification in an imbalanced cloud environment,” *Procedia Comput. Sci.*, vol. 232, pp. 1636–1645, Jan. 2024, doi: 10.1016/j.procs.2024.01.161.
- [32] “Intrusion Detection of Imbalanced Network Traffic Based on Machine Learning and Deep Learning | IEEE Journals & Magazine | IEEE Xplore.” Available: <https://ieeexplore.ieee.org/document/9311173>
- [33] “A Review of Deep Learning Applications in Intrusion Detection Systems: Overcoming Challenges in SpatioTemporal Feature Extraction and Data Imbalance.” Available: <https://www.mdpi.com/2076-3417/15/3/1552>
- [34] Hải L. T. T. and Giàu P. N., “Vấn đề mất cân bằng dữ liệu và một số phương pháp xử lý dữ liệu mất cân bằng trong mô hình học sâu,” *Tạp Chí Khoa Học Đại Học Cần Thơ*, vol. 60, no. 5, pp. 50–58, Oct. 2024, doi: 10.22144/ctujos.2024.407.
- [35] S. Matharaarachchi, M. Domaratzki, and S. Muthukumarana, “Enhancing SMOTE for imbalanced data with abnormal minority instances,” *Mach. Learn. Appl.*, vol. 18, p. 100597, Dec. 2024, doi: 10.1016/j.mlwa.2024.100597.