Tạp chí Khoa học Đại học Cần Thơ
Phần A: Khoa học tự nhiên, Công nghệ và Môi trường
ISSN 1859-2333 | e-ISSN 2815-5599

DOI:10.22144/ctujos.2025.093

GIẢI PHÁP PHÁT HIỆN XÂM NHẬP MẠNG SỬ DỤNG MÔ HÌNH HỌC SÂU

Lê Anh Quân*, Trần Minh Quang, Lý Phương Khải, Trần Trung Nguyễn và Phan Thượng Cang
Trường Công Nghệ Thông Tin và Truyền Thông, Trường Đại học Cần Thơ, Việt Nam
**Tác giả liên hệ (Corresponding author): quanB2105684@student.ctu.edu.vn*

Thông tin chung (Article Information)

Nhận bài (Received): 23/10/2024
Sửa bài (Revised): 26/11/2024
Duyệt đăng (Accepted): 20/02/2025

Title: *Network intrusion detection using deep learning model*

Author: *Le Anh Quan*, Tran Minh Quang, Ly Phuong Khai, Tran Trung Nguyen and Phan Thuong Cang*

Affiliation(s): *College of Information and Communication Technology, Can Tho University, Viet Nam*

TÓM TẮT

Trong nghiên cứu này, việc phân tích hiệu suất của các mô hình học sâu trong phát hiện xâm nhập mạng dựa trên dữ liệu UNSW-NB15 đã được thực hiện. Các mô hình được triển khai bao gồm MLP, RNN, CNN, LSTM, BiLSTM, GRU, Autoencoder và Transformer. Kết quả cho thấy BiLSTM và GRU đạt độ chính xác cao nhất (98,78%) với thời gian huấn luyện ngắn (656,20 và 497,89 phút), trong khi Transformer cũng đạt độ chính xác cao (98,76%) nhưng yêu cầu thời gian huấn luyện dài nhất (1.010,49 phút). RNN và Autoencoder có thời gian huấn luyện ngắn nhất nhưng độ chính xác thấp hơn. BiLSTM và GRU là lựa chọn tối ưu nhờ sự cân bằng giữa độ chính xác và thời gian huấn luyện, Transformer phù hợp với hệ thống không giới hạn tài nguyên. Trong nghiên cứu, tiềm năng của các mô hình học sâu trong phát hiện các mối xâm nhập mạng hiện đại và khả năng ứng dụng vào các hệ thống an ninh mạng thực tiễn được nhấn mạnh.

Từ khóa: *BiLSTM, GRU, học sâu, UNSW-NB15, phát hiện xâm nhập mạng, Transformer*

ABSTRACT

This study analyzes the performance of deep learning models in network intrusion detection using the UNSW-NB15 dataset. The evaluated models include MLP, RNN, CNN, LSTM, BiLSTM, GRU, Autoencoder, and Transformer. Experimental results show that BiLSTM and GRU achieved the highest accuracy (98.78%) with relatively short training times (656.20 and 497.89 minutes). Meanwhile, the Transformer also achieved high accuracy (98.76%) but required the longest training time (1010.49 minutes). RNN and Autoencoder had the shortest training times but slightly lower accuracy (98.64% and 98.72%). BiLSTM and GRU emerged as optimal choices due to their balance between accuracy and training time, whereas Transformer is suitable for systems with abundant computational resources. This study highlights the potential of deep learning models in detecting modern network intrusions and their applicability to practical cybersecurity systems.

Keywords: *BiLSTM, deep learning, GRU, intrusion detection, Transformer, UNSW-NB15*

1. GIỚI THIỆU

An ninh mạng đã trở thành một thách thức lớn trong thời đại kỹ thuật số khi các cuộc tấn công ngày càng trở nên tinh vi và phức tạp. Hệ thống phát hiện xâm nhập mạng (IDS - Intrusion Detection System) là một trong những giải pháp chủ chốt để bảo vệ các hệ thống mạng khỏi các hành vi xâm nhập trái phép. Tuy nhiên, các phương pháp truyền thống thường gặp phải nhiều hạn chế, như tỷ lệ cảnh báo sai cao, độ chính xác không cao khi xử lý các dạng dữ liệu mạng phức tạp và khả năng thích ứng kém với các mối đe dọa mới. Việc ứng dụng các mô hình học sâu, một nhánh của trí tuệ nhân tạo, trong phát hiện xâm nhập mạng đã mở ra cơ hội nâng cao hiệu quả và độ chính xác. Ở các nghiên cứu trước đây, các kỹ thuật học sâu như CNN, RNN và LSTM đã được áp dụng trong phát hiện xâm nhập, nhưng chưa đánh giá đầy đủ về hiệu quả phân loại đa lớp và khả năng phát hiện các kiểu tấn công mới hoặc ít xuất hiện.

Trong nghiên cứu này, một loạt các mô hình học sâu đã được triển khai như RNN, Autoencoder, MLP, CNN, BiLSTM, LSTM, GRU và Transformer nhằm nâng cao khả năng nhận diện các hành vi xâm nhập trên tập dữ liệu UNSW-NB15. Tính mới của nghiên cứu nằm ở việc đánh giá toàn diện hiệu quả của các mô hình trong phân loại đa lớp các tấn công khác nhau, như Exploits, Fuzzers và Generic trên cùng một môi trường huấn luyện. Mục tiêu được đặt ra trong nghiên cứu là xác định mô hình tối ưu với độ chính xác cao nhất, giảm thiểu thời gian huấn luyện, và cải thiện khả năng phát hiện các loại tấn công mạng phức tạp.

2. CÁC NGHIÊN CỨU LIÊN QUAN

2.1. Các loại tấn công mạng

2.1.1. Analysis

Analysis là cuộc tấn công bằng cách thâm nhập ứng dụng web bằng chức năng quét cổng, thư rác và tập lệnh web (Moustafa & Slay, 2000).

2.1.2. DoS

DoS là loại tấn công làm gián đoạn, quá tải tài nguyên của một hệ thống hoặc dịch vụ thông qua bộ nhớ, khiến nó không thể phục vụ các yêu cầu vào hệ thống (Moustafa & Slay, 2000).

2.1.3. Exploit

Exploit là một chuỗi các hướng dẫn lợi dụng lỗi, sự cố, hoặc lỗ hổng bảo mật do hành vi không mong muốn hoặc không được dự đoán trên máy chủ hoặc mạng gây ra (Moustafa & Slay, 2000).

2.1.4. Fuzzers

Fuzzers Attack là một dạng tấn công mạng nhằm vào các hệ thống và máy chủ bằng cách sử dụng một lượng lớn dữ liệu ngẫu nhiên, hoặc "fuzz", để làm tràn và gây sập hệ thống. Cuộc tấn công này liên quan đến việc tạo và gửi dữ liệu ngẫu nhiên hoặc dữ liệu không hợp lệ đến một ứng dụng hoặc hệ thống mục tiêu để tìm ra các lỗ hổng có thể bị khai thác. Mục tiêu của Fuzzers Attack là phát hiện ra các điểm yếu như lỗi tràn bộ nhớ, lỗi logic hoặc các lỗ hổng bảo mật khác trong hệ thống (More et al., 2024).

2.1.5. Generic

Generic là một loại tấn công không nhắm vào một hệ thống, ứng dụng, hoặc cấu trúc cụ thể, mà có thể áp dụng đối với nhiều loại hệ thống hoặc dịch vụ khác nhau. Loại tấn công này thường dựa trên các kỹ thuật hoặc phương pháp đã được biết đến và khai thác các lỗ hổng bảo mật, sai sót trong thiết kế, hoặc các điểm yếu chung trong cách thức hoạt động của hệ thống mạng. Mục tiêu của tấn công tổng quát là gây gián đoạn hoặc làm suy yếu tính bảo mật mà không phụ thuộc vào cấu trúc nội tại của hệ thống bị tấn công (More et al., 2024).

2.1.6. Reconnaissance

Đây là loại xâm nhập nhằm thăm dò thông tin của mạng máy tính mục tiêu nhằm trốn tránh các bảo mật của nó (Moustafa & Slay, 2000).

2.1.7. Worms

Worm là loại xâm nhập sẽ tự nhân bản và lây lan dựa trên các mạng máy tính khác tùy thuộc vào các lỗi bảo mật trên máy mục tiêu mà không cần phụ thuộc vào tương tác của người dùng (Moustafa & Slay, 2000).

2.1.8. Backdoor

Tấn công cửa sau (Backdoor) trong học máy là khi kẻ tấn công cài đặt một cửa sau vào mô hình, cho phép mô hình thực hiện cả nhiệm vụ chính và nhiệm vụ phụ của kẻ tấn công. Mô hình hoạt động bình thường khi không có "kích hoạt", khiến nó khó bị phát hiện. Nhưng khi nó được "kích hoạt" bí mật xuất hiện trong đầu vào, mô hình sẽ thực hiện nhiệm vụ phụ của kẻ tấn công, bất kể nội dung ban đầu của đầu vào (Gao et al., 2020).

2.1.9. Shellcode

Shellcode là một loại xâm nhập mà kẻ tấn công bắt đầu bằng Shellcode (một đoạn mã nhỏ) để điều khiển máy bị xâm nhập (Moustafa & Slay, 2000).

2.2. Các giải pháp phát hiện xâm nhập mạng hiện có

Các nghiên cứu gần đây khi được thực hiện đã tập trung vào việc ứng dụng các mô hình học sâu nhằm nâng cao hiệu quả của hệ thống phát hiện và phân loại các loại xâm nhập mạng, đặc biệt là trên tập dữ liệu UNSW-NB15. Tuy nhiên, các cách tiếp cận này còn thiếu sự so sánh toàn diện giữa các mô hình trong cùng một môi trường thử nghiệm, khiến việc đánh giá tổng thể và khách quan trở nên khó khăn.

Stein et al. (2024) đã sử dụng mô hình Transformer để tận dụng cơ chế tự chú ý, cho phép nắm bắt các mối quan hệ phức tạp trong dữ liệu. Mô hình đạt độ chính xác 79,57% trong phân loại nhị phân và 74,24% trong phân loại đa lớp, vượt qua CNN và LSTM trong cùng thử nghiệm. Tuy nhiên, độ chính xác này chưa đạt mức cao để đáp ứng yêu cầu thực tiễn, đặc biệt là trong các tình huống dữ liệu mất cân bằng. Bên cạnh đó, Psathas et al. (2024) đề xuất một hệ thống lai kết hợp DNN, CNN, và LSTM chạy song song, sử dụng chiến lược bỏ phiếu trọng số và đa số. Hệ thống đạt độ chính xác vượt trội, lên tới 97,5% trong giai đoạn kiểm tra. Tuy nhiên, cấu trúc lai này yêu cầu tài nguyên tính toán lớn, và nghiên cứu không làm rõ hiệu quả tương đối của từng thành phần trong hệ thống.

Pansari et al. (2024) và More et al. (2024) áp dụng phương pháp học máy trên tập dữ liệu UNSW-NB15. Pansari et al. (2024) sử dụng XGBoost để chọn lọc đặc trưng, giảm số lượng thuộc tính, từ đó cải thiện hiệu suất các mô hình như Random Forest (độ chính xác 95,18% cho phân loại nhị phân và 84,01% cho phân loại đa lớp) và giảm thời gian huấn luyện hơn 50%.

More et al. (2024) tập trung vào giảm cảnh báo sai, với Random Forest đạt độ chính xác 98,63% và tỷ lệ cảnh báo sai 1,36%. Dù hiệu quả, cả hai nghiên cứu đều bị giới hạn ở các mô hình học máy truyền thống, chưa tận dụng hết tiềm năng của học sâu trong việc xử lý các mối quan hệ phức tạp của dữ liệu.

Điểm nổi bật của nghiên cứu này là thực hiện so sánh toàn diện giữa các mô hình học sâu trong cùng một môi trường thử nghiệm công bằng, trên cùng một tập dữ liệu (UNSW-NB15) và với các siêu tham số được áp dụng đồng nhất cho tất cả các mô hình. Cách tiếp cận này giúp đánh giá khách quan hiệu quả thực sự của từng mô hình, bao gồm MLP, RNN, CNN, LSTM, BiLSTM, GRU, Autoencoder và Transformer.

Kết quả nghiên cứu không chỉ làm rõ được ưu nhược điểm của từng mô hình trong việc phát hiện và phân loại các loại tấn công mà còn chỉ ra những yếu tố quan trọng như khả năng xử lý dữ liệu mất cân bằng, mức độ phức tạp của mô hình, và hiệu quả tính toán. Đóng góp này giúp định hướng việc lựa chọn mô hình phù hợp hơn cho các hệ thống phát hiện xâm nhập mạng, đồng thời tạo nền tảng cho các nghiên cứu tiếp theo trong lĩnh vực an ninh mạng

2.3. Simple Imputer

Theo Bertsimas et al. (2021), Simple Imputer là một kỹ thuật xử lý dữ liệu trong thư viện scikit-learn, cho phép thay thế các giá trị thiếu trong dữ liệu bằng các chiến lược khác nhau, như trung bình (mean), trung vị (median) hoặc giá trị xuất hiện nhiều nhất (most frequent). Việc sử dụng Simple Imputer giúp đảm bảo tính nhất quán của dữ liệu và ngăn ngừa các lỗi do giá trị thiếu gây ra trong quá trình huấn luyện. Công thức Simple Imputer như sau:

$$\widehat{x_i} = \frac{1}{n} \times \sum_{i=1}^{N} x_i \quad (1)$$

Trong đó:

- $\widehat{x_i}$ là giá trị được thay thế,
- x_i là các giá trị quan sát được của đặc trưng đó,
- n là số lượng các giá trị quan sát được

2.4. Log Transformation

Theo Feng et al. (2014), Log Transformation là một kỹ thuật biến đổi phổ biến giúp chuyển đổi dữ liệu lệch thành dạng gần với phân phối chuẩn (normal distribution), từ đó giúp mô hình học máy hoạt động ổn định hơn. Công thức của Log Transformation là:

$$log(x) = log_e(x + 1)$$

Trong đó:

x là giá trị gốc của một đặc trưng số, chẳng hạn như số lượng byte, thời gian kết nối, hoặc bất kỳ đặc trưng nào có giá trị số trong tập dữ liệu.

2.5. Isolation Forest

Theo Liu et al. (2012), Isolation Forest là một phương pháp học không giám sát được sử dụng rộng rãi để phát hiện các điểm dữ liệu bất thường dựa trên nguyên tắc cách ly các điểm này trong không gian nhiều chiều. Isolation Forest hoạt động bằng cách xây dựng nhiều cây quyết định (decision trees) và các điểm dữ liệu nằm ở mức độ cách ly cao hơn sẽ được xem là ngoại lệ. Isolation Forest có lợi thế là không yêu cầu giả định phân phối dữ liệu, giúp mô

hình dễ dàng phát hiện các ngoại lệ trong các tập dữ liệu lớn và phức tạp như UNSW-NB15. Công thức Isolation Forest:

Độ đo mức độ cách ly của một điểm dữ liệu x được xác định dựa trên số lượng các cạnh từ gốc đến nút lá chứa x trong cây quyết định. Công thức của Isolation Forest dựa trên độ sâu của nút lá là:

$$c(n) = 2H(n-1) - \left(\frac{2(n-1)}{n}\right) \quad (2)$$

Trong đó:

n là số lượng điểm dữ liệu,

H(n) là giá trị điều hòa (Harmonic number).

Điểm cách ly được xác định là:

$$s(x,n) = 2^{-\frac{E(h(x))}{c(n)}} \quad (3)$$

Trong đó:

- E(h(x)) là độ sâu trung bình của điểm x qua tất cả các cây quyết định trong rừng,
- c(n) là giá trị điều chỉnh độ sâu tối đa.

Các điểm có giá trị s(x,n) gần với 1 được coi là ngoại lệ, trong khi các điểm có giá trị gần 0 được coi là bình thường.

2.6. Focal Loss

Một biến thể của Cross-Entropy Loss được thiết kế để tập trung vào các mẫu dữ liệu khó phân loại

Focal Loss được định nghĩa bằng công thức sau:

$$FL(p_t) = -\alpha \times (1-pt)^{\gamma} \times log(pt) \quad (4)$$

Trong đó:

p_t Là xác suất được dự đoán cho lớp thực sự của một mẫu dữ liệu. Nếu mẫu dữ liệu thuộc lớp 1, thì p_t là xác suất được mô hình dự đoán cho lớp 1, tương tự nếu mẫu thuộc lớp 0.

α: Là hệ số cân bằng, được thiết lập để giảm thiểu tác động của sự chênh lệch giữa các lớp. Nó giúp điều chỉnh sự mất cân bằng trong tập dữ liệu bằng cách giảm thiểu ảnh hưởng của các lớp có tần suất xuất hiện cao.

γ: Là hệ số trọng số điều chỉnh, giúp kiểm soát tầm quan trọng của các mẫu dữ liệu khó phân loại. Nếu p_t càng nhỏ, tức là mẫu càng khó phân loại, thì trọng số của mẫu này trong hàm mất mát sẽ càng lớn.

2.7. Standard Scaler

Theo de Amorim et al. (2022), Standard Scaler là một kỹ thuật chuẩn hóa dữ liệu, thực hiện theo phương pháp Z-score normalization. Kỹ thuật này chuẩn hóa các thuộc tính bằng cách trừ đi giá trị trung bình của chúng và chia kết quả cho độ lệch chuẩn của thuộc tính đó, dẫn đến một phân phối với trung bình bằng 0 và phương sai bằng 1.

Công thức Standard Scaler:

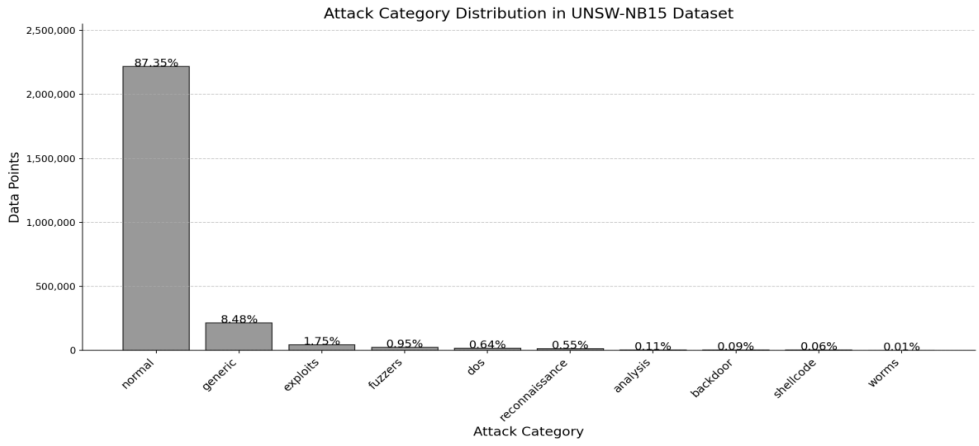
$$Z = \frac{(X-\mu)}{\sigma} \quad (5)$$

Trong đó:

- X là giá trị gốc,
- μ là giá trị trung bình,
- σ là độ lệch chuẩn.

2.8. Tập dữ liệu

Bộ dữ liệu UNSW-NB15 được tạo ra bằng công cụ IXIA PerfectStorm tại phòng thí nghiệm Cyber Range của UNSW Canberra, mô phỏng hoạt động mạng hiện đại và các hành vi tấn công. Với 100 GB dữ liệu mạng thu thập bằng tcpdump, bộ dữ liệu gồm 9 loại tấn công: Fuzzers, Analysis, Backdoors,DoS,Exploits, Generic, Reconnaissance, Shellcode, và Worms. Công cụ Argus, Bro-IDS và 12 thuật toán đã tạo ra 49 đặc trưng với nhãn lớp. Tổng cộng có 2.540.044 bản ghi trong bốn tệp CSV, chia thành bộ huấn luyện (175.341 bản ghi) và bộ kiểm thử (82.332 bản ghi). Tập dữ liệu bao gồm normal (2.218.760), generic (215.481), exploits (44.525), fuzzers (24.246), dos (16.353), reconnaissance (13.987), analysis (2677), backdoor (2329), shellcode (1511) và worms (174). Kết quả phân bố mẫu các loại tấn công được trình bày ở Hình 1.



Hình 1. Cung cấp thông tin về số lượng mẫu của các lớp trong tập dữ liệu UNSW-NB15

2.9. Đặc trưng dữ liệu

Để phục vụ cho việc phân tích và xây dựng mô hình, tập dữ liệu được biểu diễn thông qua một bộ các đặc trưng được lựa chọn và trích xuất. Các đặc trưng này đóng vai trò quan trọng trong việc mô tả các thuộc tính khác nhau của lưu lượng mạng. Trong nghiên cứu này, việc phân tích được thực hiện dựa trên bốn nhóm đặc trưng chính: đặc trưng luồng, đặc trưng cơ bản và đặc trưng nội dung, đặc trưng thời gian, cùng với các đặc trưng bổ sung được tạo ra và nhãn dữ liệu. Chi tiết về từng nhóm đặc trưng được mô tả trong các bảng sau đây:

Bảng 1. Đặc trưng luồng (Flow features)

Tên	Mô tả
scrip	Địa chỉ IP nguồn
sport	Số cổng nguồn
dstip	Địa chỉ IP đích
dsport	Số cổng đích
proto	Giao thức giao dịch

Bảng 2. Đặc trưng cơ bản và đặc trưng nội dung (Basic features and Content features)

Tên	Mô tả	Tên	Mô tả
state	Trạng thái, giao thức phụ thuộc	dur	Tổng thời gian
sbytes	Byte từ nguồn đến đích	dbytes	Byte từ đích tới nguồn
sttl	Thời gian tồn tại từ nguồn tới đích	dttl	Thời gian tồn tại từ đích tới nguồn
sloss	Gói tin nguồn bị truyền lại hoặc bị mất	dloss	Gói tin đích bị truyền lại hoặc bị mất
service	http, ftp, ssh, dns,..	sload	Số bit nguồn mỗi giây
dsload	Số bit đích mỗi giây	spkts	Số lượng gói từ nguồn tới đích
dpkts	Số lượng gói từ đích tới nguồn	swin	Quảng cáo cửa sổ TCP của nguồn
dwin	Quảng cáo cửa số TCP của đích	stcpb	Số thứ tự TCP của nguồn

Bảng 3. Đặc trưng thời gian (Time features)

Tên	Mô tả	Tên	Mô tả
sjit	Độ dao động của nguồn	djit	Độ dao động của đích
stime	Thời gian bắt đầu ghi	ltime	thời gian ghi cuối cùng
sintpkt	Thời gian giữa các gói tin liên tiếp từ nguồn	dintpkt	Thời gian giữa các gói tin liên tiếp từ đích
tcprtt	Tổng của 'synack' và 'ackdat' trong TCP	synack	Thời gian giữa các gói SYN và SYN-ACK của TCP
ackdat	Thời gian giữa các gói SYN-ACK và ACK của TCP		

Bảng 4. Đặc trưng được tạo bổ sung (Additional generated features) và Đặc trưng nhãn (Label)

Tên	Mô tả	Tên	Mô tả
is_sm_ips_ports	Nếu srcip (1) bằng dstip (3) và sport (2) bằng dsport (4), biến này sẽ được gán giá trị 1, nếu không sẽ gán giá trị 0.	ct_state_ttl	Số lượng cho mỗi trạng thái (6) dựa theo khoảng giá trị cụ thể của sttl (10) và dttl (11).
ct_flw_http_mthd	Số lượng luồng có phương thức như Get và Post trong dịch vụ HTTP.	is_ftp_login	Nếu phiên FTP được truy cập bằng tên người dùng và mật khẩu thì gán 1, nếu không thì gán 0.
ct_ftp_cmd	Số lượng luồng có lệnh trong phiên FTP.	ct_srv_src	Số lượng bản ghi chứa cùng một dịch vụ (14) và srcip (1) trong 100 bản ghi dựa theo ltime (26).
ct_srv_dst	Số lượng kết nối chứa cùng một dịch vụ (14) và địa chỉ đích (3) trong 100 kết nối dựa theo thời gian cuối cùng (26).	ct_dst_ltm	Số lượng kết nối có cùng địa chỉ đích (3) trong 100 kết nối dựa theo thời gian cuối cùng (26).
ct_src_ltm	Số lượng kết nối có cùng địa chỉ nguồn (1) trong 100 kết nối dựa theo thời gian cuối cùng (26)	ct_src_dport_ltm	Số lượng kết nối có cùng địa chỉ nguồn (1) và cổng đích (4) trong 100 kết nối dựa theo thời gian cuối cùng (26).
ct_dst_port_ltm	Số lượng kết nối có cùng địa chỉ đích (3) và cổng nguồn (2) trong 100 kết nối dựa theo thời gian cuối cùng (26).	ct_dst_src_ltm	Số lượng kết nối có cùng địa chỉ nguồn (1) và địa chỉ đích (3) trong 100 kết nối dựa theo thời gian cuối cùng (26).
attack_cat	Tên của mỗi loại tấn công. Trong tập dữ liệu này	label	0 cho các bản ghi bình thường và 1 cho các bản ghi tấn công.

Đặc trưng luồng (Flow Features) Nhóm này gồm thông tin cơ bản như địa chỉ IP, cổng và giao thức, giúp xác định đường đi và loại kết nối giữa nguồn và đích. Vai trò chính là phân biệt kết nối hợp lệ và tấn công, đặc biệt hiệu quả với các tấn công như giả mạo IP hoặc quét cổng. Các đặc trưng này cung cấp thông tin quan trọng giúp mô hình phát hiện các tấn công mạng cơ bản. Chúng đóng vai trò quan trọng trong việc nhận diện sớm các cuộc tấn công có liên quan đến việc thay đổi thông tin kết nối cơ bản, từ đó tăng cường khả năng phân loại kết nối mạng chính xác.

Đặc trưng cơ bản và nội dung (Content Features) Nhóm này gồm các mô tả trạng thái, kích thước gói tin và dữ liệu được truyền qua kết nối. Các đặc trưng như service và dur hữu ích trong phát hiện tấn công dịch vụ cụ thể, trong khi sbytes và dbytes hỗ trợ nhận diện tấn công DDoS với lưu lượng bất thường. Các đặc trưng này giúp mô hình phát hiện các hành vi bất thường như tấn công DDoS hoặc các tấn công nhắm vào dịch vụ cụ thể (như HTTP, FTP). Việc sử dụng các đặc trưng này có thể tăng cường khả năng phân biệt giữa các kết nối hợp lệ và các cuộc tấn công phức tạp hơn, từ đó nâng cao hiệu quả phát hiện xâm nhập.

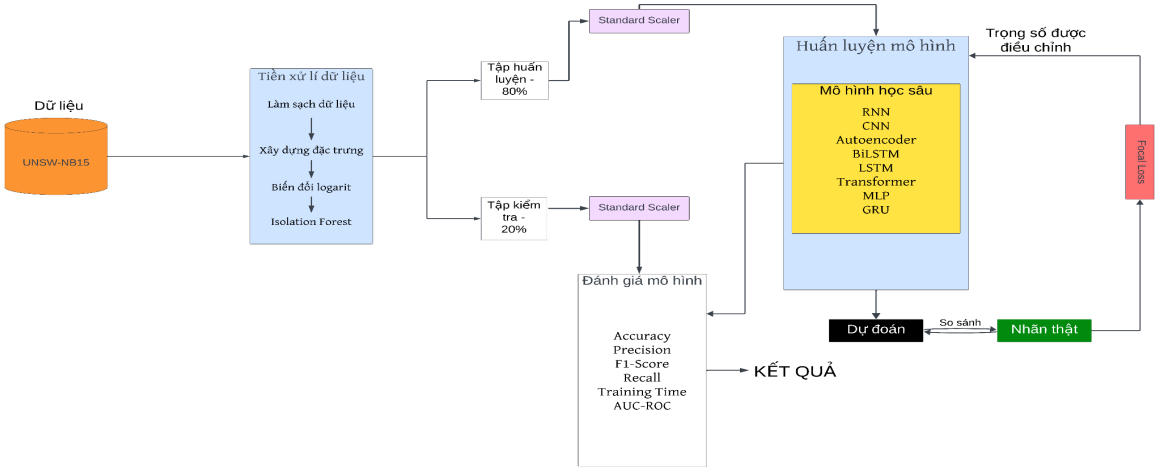
Đặc trưng thời gian (Time Features) Nhóm này bao gồm việc theo dõi khoảng thời gian và độ trễ giữa các gói tin. Những đặc trưng như stime và dintpkt rất quan trọng trong nhận diện các tấn công chậm, nơi thời gian phản hồi giữa các gói tin không nhất quán. Các đặc trưng này đóng vai trò quan trọng trong việc phát hiện các tấn công chậm (slow attacks), khi mà độ trễ hoặc thời gian phản hồi bất thường có thể chỉ ra một cuộc tấn công. Mô hình có thể phân biệt giữa các hành vi hợp lệ và bất thường dựa trên thời gian truyền tải dữ liệu, từ đó cải thiện khả năng phát hiện các loại tấn công tinh vi.

Đặc trưng được tạo bổ sung (Additional Generated Features) Các đặc trưng này được tổng hợp từ dữ liệu ban đầu để tăng cường khả năng phát hiện hành vi đáng ngờ. Ví dụ, ct_state_ttl và ct_srv_src hỗ trợ phát hiện các hành vi tấn công lặp lại như quét mạng hoặc dò mật khẩu. Chúng giúp mô hình phát hiện các hành vi tấn công lặp đi lặp lại hoặc các cuộc tấn công lan rộng, đặc biệt trong các môi trường có lưu lượng mạng lớn. Việc bổ sung các đặc trưng này có thể cải thiện độ chính xác của mô hình và giúp phát hiện các tấn công tinh vi, như tấn công dò mật khẩu hoặc quét mạng.

Nhãn và danh mục tấn công (Label and Attack Category) Các nhãn này đóng vai trò then chốt trong việc phân loại và nhận diện các hành vi tấn công mạng so với các kết nối hợp lệ. Những nhãn này không chỉ giúp mô hình học máy nhận diện được loại tấn công cụ thể, mà còn tối ưu hóa khả năng phân loại chính xác trong các tình huống thực tế. Việc phân bổ nhãn một cách hợp lý có thể cải thiện đáng kể khả năng tổng quát của mô hình, đặc biệt trong việc phát hiện các tấn công hiếm gặp hoặc có tính chất phức tạp. Nhờ vào sự rõ ràng và chi tiết của các nhãn, mô hình có thể đạt được độ chính xác cao hơn và cải thiện hiệu suất khi xử lý những tình huống tấn công đa dạng và phức tạp.

3. MÔ HÌNH ĐỀ XUẤT

Quy trình nghiên cứu được thể hiện ở Hình 2, sử dụng dữ liệu UNSW-NB15, trải qua các bước tiền xử lý (làm sạch, trích chọn đặc trưng, logarit hóa, phát hiện bất thường bằng Isolation Forest), chia dữ liệu, chuẩn hóa và huấn luyện nhiều mô hình học sâu (RNN, CNN, Transformer, Autoencoder, BiLSTM, LSTM, MLP, GRU) với Focal Loss. Hiệu quả được đánh giá qua các chỉ số như Accuracy, F1-Score, Precision, Recall, Trainning Time và AUC-ROC.



Hình 2. Sơ đồ quy trình hoạt động của hệ thống

3.1. Tiền xử lí dữ liệu

Trong nghiên cứu này, tập dữ liệu được sử dụng là UNSW-NB15 để huấn luyện và đánh giá mô hình. Trước khi tiến hành huấn luyện, dữ liệu được làm sạch và chuẩn hóa thông qua một loạt các bước tiền xử lý nhằm cải thiện chất lượng và tính nhất quán của dữ liệu.

Đầu tiên, các tập con dữ liệu của UNSW-NB15 được kết hợp lại và các nhãn của cột attack_cat được chuẩn hóa về lớp 'normal'. Các giá trị thiếu trong các cột quan trọng như ct_flw_http_mthd và is_ftp_login được thay thế bằng 0. Đồng thời, nhãn của cột attack_cat được ánh xạ thành các giá trị số để phục vụ cho bài toán phân loại. Đối với các cột đặc trưng số, phương pháp Simple Imputer được sử dụng để thay thế các giá trị thiếu bằng giá trị trung bình (mean) trong dữ liệu trước khi tiến hành các bước tiếp theo như chuẩn hóa hoặc huấn luyện mô hình, bởi các cột số (numeric columns) có thể chứa các giá trị NaN (giá trị bị thiếu), điều này sẽ gây ra lỗi trong quá trình huấn luyện mô hình hoặc làm giảm chất lượng dữ liệu.

Sau khi hoàn thành các bước xử lý giá trị thiếu, các kỹ thuật biến đổi dữ liệu như log transformation đối với các đặc trưng số được áp dụng để giảm thiểu độ lệch phân phối (nếu có).

Đồng thời, các đặc trưng mới như duration (khoảng thời gian kết nối), byte_ratio, pkt_ratio, load_ratio,

jit_ratio, và tcp_setup_ratio được tạo ra để nâng cao khả năng phân biệt của mô hình. Sau đó, Isolation Forest được sử dụng để phát hiện và loại bỏ các mẫu ngoại lệ với tỷ lệ ngoại lệ được thiết lập là 0,01, điều này cho phép mô hình tập trung vào các mẫu dữ liệu bình thường và giảm thiểu ảnh hưởng của các ngoại lệ.

Sau khi xử lý các mẫu ngoại lệ, dữ liệu được chia thành tập huấn luyện và tập kiểm tra bằng cách sử dụng hàm train_test_split. Hàm này phân chia dữ liệu thành hai phần ngẫu nhiên với tỷ lệ 80% cho tập huấn luyện và 20% cho tập kiểm tra, đồng thời đảm bảo duy trì tỷ lệ của các lớp thông qua tham số stratify.

Cuối cùng, các đặc trưng số được tiến hành chuẩn hóa bằng phương pháp StandardScaler. Quá trình chuẩn hóa giúp đảm bảo rằng các đặc trưng có cùng phân phối, giúp mô hình học được các đặc trưng mà không bị ảnh hưởng bởi sự khác biệt về đơn vị hoặc giá trị tuyệt đối của dữ liệu.

3.2. Mô hình học sâu

Theo Saabith et al. (2023), học sâu là một tập hợp con của các thuật toán học có giám sát và không giám sát, các thuật toán học sâu sử dụng mạng nơ-ron nhiều lớp để phát hiện bất thường. Các thuật toán này đặc biệt giỏi trong việc phát hiện những bất thường trong dữ liệu có độ phức tạp cao vì chúng có khả năng tự động học các biểu diễn phức tạp của dữ liệu đầu vào. Một số ví dụ về các kỹ thuật học sâu để nhận diện bất thường là mạng nơ-ron tích chập (CNN), mạng nơ-ron hồi quy (RNN) và mạng đối kháng sinh (GAN).

3.2.1. MLP

Perceptron đa lớp(MLP) là một mạng nơ-ron truyền thẳng đơn giản gồm lớp đầu vào, các lớp ẩn và lớp đầu ra. Lớp đầu vào xử lý dữ liệu thô, các lớp ẩn thực hiện biến đổi phi tuyến và trích xuất đặc trưng, còn lớp đầu ra dựa trên đặc trưng để dự đoán kết quả. MLP được ứng dụng rộng rãi trong nhiều nhiệm vụ như nhận diện giọng nói, hình ảnh và phân đoạn ngữ nghĩa (Yuan et al., 2024).

3.2.2. RNN

Mạng nơ-ron hồi tiếp (RNNs) là một mở rộng của mạng nơ-ron truyền thẳng, có khả năng truyền thông tin qua các bước thời gian. Tính năng này giúp RNNs đặc biệt phù hợp cho các nhiệm vụ liên quan đến chuỗi dữ liệu có sự phụ thuộc thời gian, như xử lý chuỗi đầu vào và đầu ra (Lipton et al., 2015).

3.2.3. Autoencoder

Bộ mã hóa tự động là mạng thần kinh nhằm mục đích tìm hiểu các biểu diễn dữ liệu nén bằng cách mã hóa dữ liệu đầu vào vào không gian tiềm ẩn có chiều thấp hơn và sau đó tái tạo lại đầu vào ban đầu từ không gian đó thông qua quy trình giải mã (Hinton & Salakhutdinov, 2006).

3.2.4. CNN

Mạng nơ-ron tích chập (CNN) là một kỹ thuật học sâu phổ biến với cấu trúc đa lớp yêu cầu ít tiền xử lý. CNN bao gồm lớp đầu vào, lớp đầu ra và các lớp ẩn như lớp tích chập, lớp gộp và lớp kết nối đầy đủ. So với các thuật toán phân loại khác, CNN có lợi thế nhờ sử dụng ít tiền xử lý và không phụ thuộc vào thiết kế tính năng trước đó (Wu et al., 2020).

3.2.5. LSTM

Bộ nhớ dài hạn ngắn hạn (LSTM) là một loại mạng nơ-ron hồi quy, có khả năng học và dự đoán dữ liệu tuần tự tốt. LSTM được phát triển để khắc phục hạn chế của RNN trong việc duy trì bộ nhớ dài hạn bằng cách thêm cấu trúc bộ nhớ, giúp duy trì trạng thái theo thời gian, với các cổng để quản lý việc nhớ, quên và xuất dữ liệu. LSTM đạt hiệu quả cao trong các ứng dụng như nhận dạng giọng nói, tổng hợp giọng nói, mô hình hóa ngôn ngữ, dịch thuật và nhận dạng chữ viết tay (Al-jabery et al., 2020).

3.2.6. Transformer

Transformer là một kiến trúc mô hình mới trong xử lý ngôn ngữ tự nhiên (NLP), không sử dụng cơ chế tuần tự như RNN hay LSTM truyền thống, mà hoàn toàn dựa vào cơ chế tự chú ý. Điều này giúp mô hình học các mối quan hệ phụ thuộc trong dữ liệu một cách hiệu quả hơn so với các mô hình tuần tự truyền thống. Transformer nổi bật trong việc quản lý các phụ thuộc dài hạn giữa các phần tử trong chuỗi đầu vào và hỗ trợ xử lý song song (Vaswani et al., 2017; Islam et al., 2024).

3.2.7. BiLSTM

BiLSTM (LSTM hai chiều) là một mô hình xử lý chuỗi gồm hai mạng LSTM, trong đó một mạng xử lý theo hướng tiến và mạng kia theo hướng lùi. Điều này giúp tăng cường khả năng học các mối quan hệ phụ thuộc trong dữ liệu, cải thiện hiệu quả của mô hình (TS & Shrinivasacharya, 2021).

3.2.8. GRU

GRU (Đơn vị hồi quy có cổng) là một loại đơn vị hồi quy trong mạng nơ-ron hồi quy (RNN), được thiết kế để giải quyết vấn đề biến mất gradient thường gặp trong các RNN truyền thống. GRU sử dụng các cổng để điều tiết dòng chảy thông tin bên trong mỗi đơn vị. Cấu trúc của GRU đơn giản hơn so với LSTM vì không có bộ nhớ riêng biệt, thay vào đó, các cổng trong GRU quyết định giữ lại bao nhiêu thông tin từ bước trước và kết hợp với thông tin mới từ bước hiện tại, giúp duy trì thông tin qua các bước thời gian mà không gặp vấn đề biến mất gradient (Cho et al., 2014).

3.3. Xử lý mất cân bằng dữ liệu

Trong nghiên cứu này, các phương pháp cân bằng dữ liệu trong bước tiền xử lý không được sử dụng mà lựa chọn áp dụng Focal Loss để giải quyết vấn đề mất cân bằng rõ rệt giữa các lớp tấn công trong tập dữ liệu UNSW-NB15.

Sự mất cân bằng này dẫn đến tình trạng các mô hình học sâu có thể đạt độ chính xác cao trên các lớp phổ biến nhưng lại bỏ sót các tấn công hiếm gặp, tiềm ẩn nhiều nguy cơ nghiêm trọng đối với an ninh mạng. Thay vì can thiệp vào tập dữ liệu, Focal Loss điều chỉnh trọng số các mẫu trong quá trình huấn luyện, cho phép mô hình tập trung hơn vào các mẫu khó phân loại hoặc các lớp ít xuất hiện, từ đó cải thiện khả năng phát hiện các loại tấn công nguy hiểm nhưng ít gặp.

Trong nghiên cứu này, hệ số α = 1 và γ = 2 được chọn dựa trên các nghiên cứu trước đây và thử nghiệm thực nghiệm trên tập dữ liệu UNSW-NB15. Hệ số γ giúp tăng cường trọng số của các mẫu khó phân loại, đồng thời giảm sự tác động của các mẫu dễ, qua đó cải thiện hiệu quả tổng thể của mô hình mà không làm mất cân bằng trong việc học.

Cách tiếp cận này không chỉ giúp duy trì tính nguyên bản của dữ liệu mà còn mang lại một giải pháp hiệu quả và tự nhiên hơn để xử lý sự mất cân bằng, đảm bảo hiệu suất cao và đáng tin cậy trong việc phát hiện các tấn công mạng phức tạp.

3.4. Các phương pháp đánh giá

Độ chính xác (Accuracy) là một thước đo phổ biến để đánh giá hiệu suất của các mô hình phân lớp. Tuy nhiên, việc chỉ dựa vào độ chính xác có thể không cung cấp cái nhìn toàn diện về hiệu quả hoạt động của tất cả các mô hình. Do đó, bên cạnh độ chính xác, các chỉ số khác cũng được sử dụng thêm như độ chính xác (Precision), khả năng truy xuất (Recall), và điểm F1 (F1-score), AUC. Những thước đo này giúp cung cấp một đánh giá toàn diện và chính xác hơn về khả năng phân loại, đặc biệt là trong các trường hợp dữ liệu mất cân bằng hoặc khi có sự chênh lệch giữa các lớp.

Accuracy là tỷ lệ giữa số mẫu được phân loại đúng và tổng số mẫu trong tập dữ liệu đánh giá (Hicks et al., 2022).

$$Accuracy = \frac{Tổng\ số\ mẫu\ dự\ đoán\ đúng}{Tổng\ số\ mẫu} \quad (6)$$

Recall là tỷ lệ số mẫu dự đoán đúng (True Positives) của một lớp cụ thể so với tổng số mẫu thực sự thuộc lớp đó (TP + FN).

$$Recall_i = \frac{True\ Positives_i}{True\ Positives_i + False\ Negatives_i} (7)$$

Trong đó:

- $True\ Positives_i$ (TP) là số lượng mẫu thuộc lớp i mà mô hình dự đoán đúng là thuộc lớp i,
- $False\ Negatives_i$ (FN) là số lượng mẫu thuộc lớp i mà mô hình dự đoán sai.

Weighted Recall: Trung bình có trọng số của Recall, trong đó trọng số là số lượng mẫu của từng lớp:

$$Weighted\ Recall = \frac{\sum_{i=1}^{N} Recall_i \times Support_i}{\sum_{i=1}^{N} Support_i} (7)$$

Trong đó:

- $Support_i$ **:** Là số lượng mẫu thực sự trong mỗi lớp. Support giúp đánh giá tầm quan trọng của từng lớp trong quá trình tính toán Weighted Recall.

Precision là tỷ lệ số mẫu dự đoán đúng (True Positives - TP) của một lớp cụ thể so với tổng số mẫu được dự đoán là lớp đó (TP + FP).

$$Precision_i = \frac{True\ Positives_i}{True\ Positives_i + False\ Positives_i} (8)$$

Trong đó:

- $True\ Positives_i$: Là số lượng mẫu thuộc lớp i mà mô hình dự đoán đúng là thuộc lớp i,
- $False\ Positives_i$: Số lượng các mẫu được dự đoán là lớp iii nhưng thực tế thuộc lớp khác.

Weighted Precision: Trung bình có trọng số của Precision, trong đó trọng số là số lượng mẫu của từng lớp:

$$Weighted\ Precision = \frac{\sum_{i=1}^{N} Precision_i \times Support_i}{\sum_{i=1}^{N} Support_i} \quad (9)$$

F1-Score là trung bình điều hòa giữa Precision và Recall, nhằm cân bằng giữa hai chỉ số này.

$$F1 - Score_i = 2 \times \frac{Precision_i \times Recall_i}{Precision_i + Recall_i} (10)$$

Weighted F1-Score: Trung bình có trọng số của F1-Score, trong đó trọng số là số lượng mẫu của từng lớp:

$$Weighted\ F1 - Score = \frac{\sum_{i=1}^{N} F1 - Score_i \times Support_i}{\sum_{i=1}^{N} Support_i} \quad (11)$$

Training Loss là giá trị tổn thất của mô hình tính toán trên tập huấn luyện. Nó đại diện cho sai số giữa đầu ra dự đoán của mô hình và giá trị nhãn thực tế trong tập huấn luyện. Mục tiêu là giảm giá trị này để mô hình học tốt hơn từ dữ liệu.

$$Training\ Loss = \frac{1}{N}\sum_{i=1}^{N} L(\hat{y}_i, y_i) \quad (12)$$

Trong đó:

- N là số mẫu trong tập huấn luyện,
- $\hat{y}_i$ là số mẫu dự đoán cho mẫu thứ i,
- y_i là nhãn thực tế của mẫu thứ i,
- L là hàm mất mát, như Cross-Entropy Loss hoặc Mean Squared Error Loss.

Validation Loss là giá trị tổn thất của mô hình tính toán trên tập kiểm tra (validation set). Nó dùng để đánh giá khả năng tổng quát hóa của mô hình.

$$Validation\ Loss = \frac{1}{M}\sum_{i=1}^{M} L(\hat{y}_i, y_i) \quad (13)$$

Trong đó:

- M là số mẫu trong tập kiểm tra,
- $\hat{y}_i$ là số mẫu dự đoán cho mẫu thứ i,
- y_i là nhãn thực tế của mẫu thứ i,
- L là hàm mất mát, tương tự như hàm tình toán trong Training Loss.

Training Time per Epoch là thời gian mà mô hình cần để hoàn thành một epoch huấn luyện (epoch là một vòng lặp qua toàn bộ dữ liệu huấn luyện). Thời gian này phụ thuộc vào kích thước tập dữ liệu, số lượng tham số của mô hình, và tài nguyên tính toán (GPU/CPU).

$$\frac{Training\ Time}{Epoch} = End\ Time\ of\ Epoch - Start\ Time\ of\ Epoch \quad (10)$$

AUC (Area Under the Curve) đo lường khả năng phân biệt giữa các lớp trong bài toán phân loại và được tóm tắt từ đường cong ROC (Receiver Operator Characteristic), một biểu đồ trực quan hóa hiệu suất dự đoán của mô hình. AUC được xác định dựa trên True Positive Rate (TPR) và False Positive Rate (FPR). Các chỉ số này được tính theo các công thức sau:

- TPR là tỷ lệ các mẫu dương tính được dự đoán đúng,

- FPR là tỷ lệ các mẫu âm tính bị dự đoán sai thành dương tính.

$$TPR = \frac{TP}{TP + FN} \quad (14)$$

$$FPR = \frac{FP}{FP + TN} \quad (15)$$

Trong đó:

- TP (True Positive): Số lượng mẫu dương tính được phân loại đúng là dương tính.

- FN (False Negative): Số lượng mẫu dương tính bị phân loại sai là âm tính.

- FP (False Positive): Số lượng mẫu âm tính bị phân loại sai là dương tính.

- TN (True Negative): Số lượng mẫu âm tính được phân loại đúng là âm tính.

Giá trị AUC nằm trong khoảng từ 0 đến 1:

- AUC = 1,0: Mô hình hoàn hảo, không có lỗi dự đoán.

- AUC = 0,5: Mô hình không tốt hơn việc đoán ngẫu nhiên.

- AUC < 0,5: Mô hình hoạt động tệ hơn việc đoán ngẫu nhiên.

AUC càng cao thì mô hình phân biệt giữa các lớp dương tính và âm tính càng tốt.

4. KẾT QUẢ VÀ THẢO LUẬN

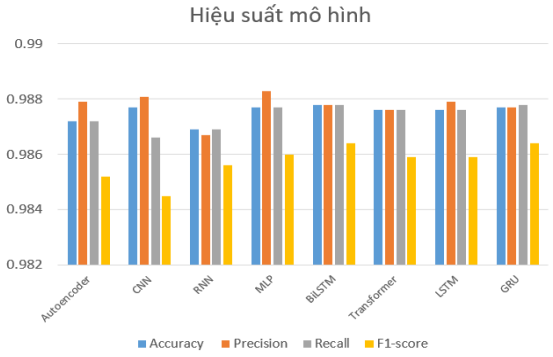
4.1. Thiết lập thực nghiệm

Các thực nghiệm được tiến hành trên hệ thống tính toán hiệu năng cao với CPU 20 nhân, RAM 48GB và ổ lưu trữ 300GB. Hệ thống có GPU NVIDIA Tesla P40 với 3840 nhân CUDA và 24GB bộ nhớ đồ họa, tối ưu cho xử lý song song. Hệ thống còn trang bị ổ SSD 512GB giúp truy xuất dữ liệu nhanh hơn. Môi trường thực nghiệm chạy trên Linux, điều này đảm bảo quản lý tài nguyên hiệu quả và ổn định khi xử lý công việc lớn.

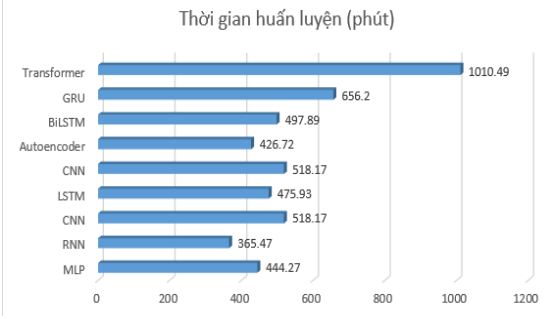
4.2. Kết quả

Trong nghiên cứu này, các mô hình Transformer, GRU và BiLSTM đã thể hiện hiệu suất ấn tượng, với những điểm mạnh riêng biệt. Mô hình Transformer đạt Accuracy 98,76% và F1-Score 98,59%, kết quả cho thấy khả năng vượt trội trong việc xử lý các tập dữ liệu phức tạp. Tuy nhiên, thời gian huấn luyện dài là hạn chế chính, làm giảm tính thực tiễn khi cần phát hiện nhanh chóng và kịp thời các cuộc tấn công mạng hiện đại. Ngược lại, mô hình GRU nổi bật với hiệu suất toàn diện, đạt Accuracy 98,77%, Precision 98,77%, Recall 98,78% và F1-Score 98,64%. Đặc biệt, thời gian huấn luyện của GRU nhanh hơn đáng kể, giúp mô hình này trở thành lựa chọn ưu tiên trong các bài toán đòi hỏi sự cân bằng giữa độ chính xác và hiệu quả thời gian. BiLSTM cũng là một mô hình mạnh mẽ, đạt Accuracy 98,78% và F1-Score 98,64%. Với khả năng xử lý dữ liệu chuỗi theo cả hai chiều thời gian, BiLSTM đặc biệt phù hợp với các dữ liệu có cấu trúc chuỗi phức tạp. Tóm lại, mỗi mô hình đều

có ưu điểm riêng. GRU và BiLSTM được đánh giá cao nhờ sự cân đối giữa hiệu suất và tính khả thi trong ứng dụng thực tế, trong khi Transformer thể hiện tiềm năng vượt trội đối với các bài toán phức tạp nhưng yêu cầu tài nguyên tính toán lớn.



Hình 3. So sánh hiệu suất của từng mô hình trên tập kiểm thử

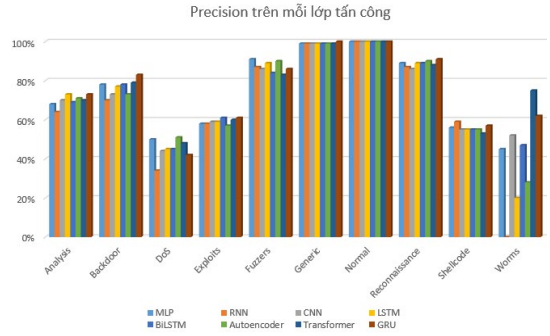


Hình 4. Thời gian huấn luyện mô hình

Kết quả từ các mô hình khác nhau được trình bày trong Hình 5 cung cấp cái nhìn tổng quan về khả năng phân loại của từng mô hình đối với các loại tấn công khác nhau.

Đối với các lớp Analysis, DoS và Shellcode, các mô hình như BiLSTM đạt precision lần lượt là 69%, 45% và 55%, trong khi MLP đạt 68%, 50% và 56% và RNN đạt 64%, 34% và 59%. Điều này cho thấy khả năng phát hiện các tấn công trong các lớp này không đủ chính xác và có thể dẫn đến nhiều trường hợp False Positives. Ngược lại, kết quả các mô hình như Transformer và GRU cho thấy hiệu quả tốt hơn với precision lần lượt là 70% và 73% cho lớp Analysis. Ở lớp Backdoor, GRU đạt precision cao nhất (83%), tiếp theo là Transformer (79%) và BiLSTM (78%). Trong khi đó, các mô hình khác như LSTM và Autoencoder đạt precision thấp hơn, chỉ lần lượt là 77% và 73%. Với lớp Fuzzers, các mô hình như Autoencoder và GRU đạt kết quả khá cao với precision lần lượt là 90% và 86%, trong khi Transformer thấp hơn một chút (83%). Đối với lớp Generic và Normal, tất cả các mô hình đều đạt precision tối đa (99% - 100%), phản ánh khả năng phân loại tốt ở hai lớp này. Tuy nhiên, một số lớp như Worms và Shellcode có mức precision rất thấp. Đặc biệt, ở lớp Worms, RNN hoàn toàn không thể phân loại với precision 0%, trong khi Transformer dẫn đầu với 75% và GRU đạt 62%. Ở lớp Shellcode, GRU có kết quả tốt nhất (57%), tiếp theo là RNN (59%), trong khi các mô hình như CNN và Autoencoder chỉ đạt khoảng 55%.

Nhìn chung, các mô hình như GRU, BiLSTM và Transformer cho thấy hiệu quả tốt hơn trên các lớp khó phân loại, trong khi các mô hình truyền thống như RNN hoặc MLP gặp nhiều khó khăn hơn trong việc phân loại chính xác trên các lớp đó.

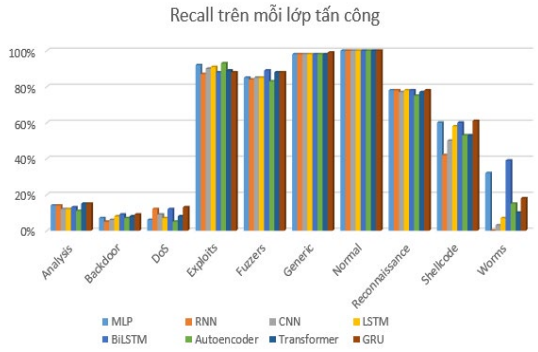


Hình 5. Biểu đồ Precision của các mô hình theo từng lớp tấn công

Recall là một thông số quan trọng, đo lường tỷ lệ các mẫu dương tính thực sự (true positives) được mô hình dự đoán đúng, phản ánh khả năng phát hiện các mẫu thuộc lớp dương tính. Dựa trên các chỉ số từ bảng, chỉ số recall cao nhất ở các loại tấn công Generic, Exploits, Fuzzers và Reconnaissance. Với loại tấn công Generic, hầu hết các mô hình đều đạt recall 98%, trong khi AutoEncoder, LSTM, GRU và Transformer đạt ngưỡng cao tới 98% - 99%. Đối với Exploits, recall dao động từ 87% đến 93%, nổi bật nhất là AutoEncoder với 93%. Các chỉ số của Fuzzers nằm trong khoảng 83% đến 89%, với LSTM đạt cao nhất là 89%. Trong khi đó, loại tấn công Reconnaissance có kết quả ngang bằng nhau dao động trong khoảng từ 77% đến 78% trên các mô hình. Đặc biệt, tất cả các mô hình đều đạt recall 100% đối với lớp không bị tấn công (Normal), cho thấy khả năng nhận diện hoàn hảo.

Tuy nhiên, các loại tấn công khác lại không đạt kết quả tốt. Với Analysis, recall chỉ đạt tối đa 15% đối với mô hình Transformer và GRU, trong khi các mô hình khác dao động từ 11% đến 14%. Loại tấn công Backdoor có recall rất thấp, chỉ từ 5% đến 9%,

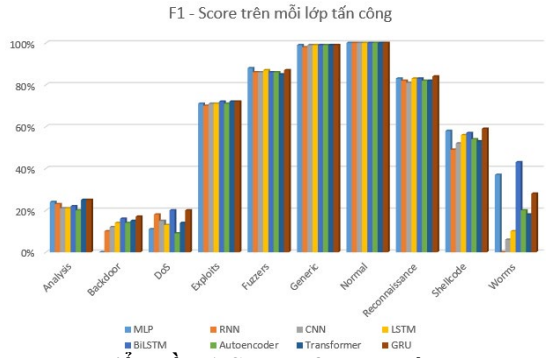
trong đó BiLSTM và GRU đạt cao nhất với 9%. Worms là lớp có recall thấp nhất, với phần lớn các mô hình dưới 20% và BiLSTM đạt cao nhất là 39%. Đối với DoS, các mô hình có recall từ 5% (AutoEncoder) đến 13% (GRU), nhưng vẫn không khả quan. Những kết quả này cho thấy các mô hình hoạt động tốt ở một số loại tấn công như Generic, Exploits, Fuzzers và Reconnaissance nhưng gặp khó khăn với các lớp như Backdoor, Worms và DoS.



Hình 6. Biểu đồ thông số Recall của từng mô hình trên từng lớp tấn công

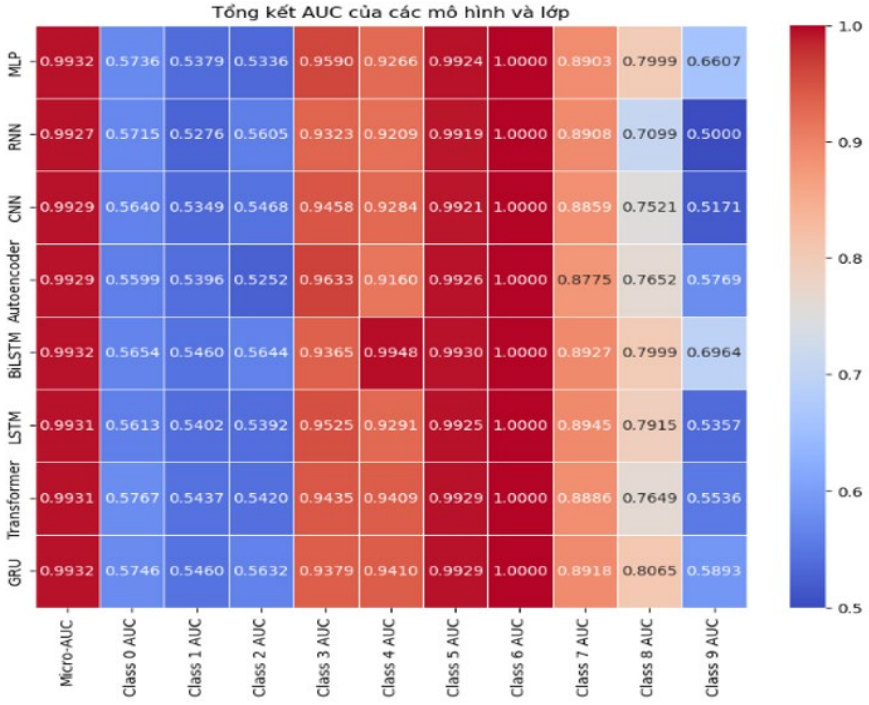
F1-score là một thông số đo lường hiệu suất của mô hình bằng cách kết hợp precision và recall thông qua trung bình điều hòa của chúng. F1-score phản ánh sự cân bằng giữa khả năng mô hình dự đoán chính xác các mẫu dương tính (precision) và khả năng phát hiện đầy đủ các mẫu dương tính thực sự (recall). Do đó, F1-score là một thông số quan trọng trong các bài toán mà cả precision và recall đều cần được tối ưu, như phát hiện gian lận, chẩn đoán bệnh, hay phát hiện bất thường (anomaly detection).

Kết quả được trình bày ở Hình 7 là kết quả tổng hợp F1-score của các mô hình trên 9 lớp tấn công và lớp Normal. Trên lớp Analysis, các mô hình Transformer và GRU đạt F1-score cao nhất (25%), cho thấy khả năng phân loại tương đối tốt. Đối với lớp Backdoor, GRU đạt F1-score cao nhất (17%), nhưng giá trị vẫn thấp, điều này cho thấy cần cải thiện. Trên lớp DoS, BiLSTM và GRU đạt F1-score cao nhất (20%), trong khi các mô hình khác có F1-score thấp hơn, kết quả này phản ánh hiệu suất hạn chế trong việc phát hiện tấn công này. Trên lớp Exploits, các mô hình BiLSTM, Transformer và GRU đạt F1-score cao nhất (72%), cho thấy khả năng phân loại tốt. Trên lớp Fuzzers, MLP, GRU và LSTM đạt F1-score cao nhất (88% và 87%), điều này thể hiện hiệu suất tốt trong việc nhận diện tấn công này. Trên lớp Generic, F1-score giữa các mô hình gần như tương đương với mức 99%, cho thấy độ chính xác cao trong phân loại. Tất cả các mô hình đều đạt F1-score 100% trên lớp Normal, biểu thị hiệu suất hoàn hảo trong việc nhận diện lưu lượng bình thường. Đối với lớp Reconnaissance, GRU đạt F1-score cao nhất (84%), kết quả cho thấy hiệu suất ổn định. Trên lớp Shellcode, GRU đạt F1-score cao nhất (59%), cải thiện so với các mô hình khác. Cuối cùng, trên lớp Worms, BiLSTM đạt F1-score cao nhất (43%), mặc dù giá trị còn thấp, nhưng cao hơn đáng kể so với các mô hình khác, cho thấy tiềm năng cải thiện trong việc phát hiện tấn công hiếm gặp này.



Hình 7. Biểu đồ F1-Score của các mô hình theo từng lớp tấn công

Kết quả tổng quan giá trị AUC được thể hiện ở Hình 8 cho thấy các mô hình học sâu có hiệu suất tổng thể tốt với Micro-AUC dao động từ 99,23% đến 99,30%. Các mô hình như BiLSTM, LSTM, GRU và CNN thể hiện khả năng phân loại tốt và cân bằng giữa các lớp, đặc biệt là BiLSTM và Transformer đạt AUC cao ở nhiều lớp. Tuy nhiên, các lớp như Analysis, Backdoor, Worms có chỉ số AUC thấp hơn, chỉ ra rằng việc nhận diện các tấn công này còn thách thức, đặc biệt với các mô hình như Autoencoder và RNN. Đặc biệt, Transformer và GRU có chỉ số AUC cao hơn ở một số lớp như class Fuzzers và class Reconnaissance, nhấn mạnh khả năng mô hình hóa tốt hơn của các mô hình này trong việc phát hiện các loại tấn công đa dạng. Tất cả các mô hình đều đạt AUC tuyệt đối 100% cho class Normal, cho thấy hiệu suất tối ưu trong việc phân loại lớp này.



Hình 8. Biểu đồ tổng quan giá trị AUC của các mô hình qua các lớp tấn công

4.3. Thảo luận

Trong lĩnh vực phát hiện xâm nhập mạng, việc thu thập và xử lý dữ liệu mạng thực tế đóng vai trò quan trọng trong việc cải thiện hiệu quả của các mô hình học sâu. Tuy nhiên, các tập dữ liệu hiện có như UNSW-NB15, mặc dù được thiết kế chi tiết và chứa các hành vi tấn công đa dạng, là dữ liệu mô phỏng và có hạn chế về số lượng mẫu cho các lớp tấn công hiếm và phức tạp như Worms, Shellcode và Backdoor. Để khắc phục vấn đề này, việc sử dụng Honeypot đã được chứng minh là một giải pháp hiệu quả, giúp thu thập dữ liệu thực tế và tạo ra dữ liệu giàu thông tin nhằm tăng cường khả năng nhận diện của các mô hình học sâu.

Honeypot là công cụ hữu hiệu trong việc mô phỏng các dịch vụ dễ bị tấn công (HTTP, FTP, SSH), ghi nhận hành vi của kẻ tấn công và tạo ra dữ liệu thực tế phục vụ huấn luyện mô hình học sâu. Dữ liệu thu thập được xử lý bằng các công cụ như Argus hoặc Bro/Zeek để tạo ra các luồng dữ liệu (flows) và trích xuất các đặc trưng cần thiết. Các đặc trưng được trích xuất bao gồm: xác định giao thức (TCP, UDP, ICMP), phân tích cổng nguồn và cổng đích, tính toán thời gian giữa các gói tin, thời gian tồn tại kết nối và khai thác nội dung payload hoặc các lệnh bất thường. Quá trình này còn được bổ sung các đặc trưng thống kê như tần suất gói tin và phân bố thời gian để phản ánh hành vi bất thường. Sau đó, các đặc trưng này được mã hóa (one-hot encoding hoặc embeddings) và chuẩn hóa để đảm bảo tính tương thích giữa các đặc trưng số và danh mục.

Dữ liệu đã được xử lý và trích xuất đặc trưng sẽ được tích hợp vào tập dữ liệu hiện có như UNSW-NB15, giúp cân bằng mẫu và làm giàu các đặc trưng hiếm, đặc biệt đối với các lớp tấn công khó như Worms và Backdoor. Sau khi hoàn thiện, các đặc trưng này trở thành đầu vào cho các mô hình học sâu, hỗ trợ huấn luyện hiệu quả hơn và tăng khả năng nhận diện các hành vi tấn công phức tạp.

Trong hệ thống thực tế, Honeypot đóng vai trò thu thập dữ liệu liên tục từ hành vi tấn công mới, đảm bảo mô hình luôn được cập nhật với các mối đe dọa hiện đại. Dữ liệu mạng thời gian thực được xử lý qua pipeline trích xuất đặc trưng tương tự, sau đó đưa vào mô hình học sâu để phân loại tấn công và phát hiện xâm nhập. Việc tích hợp này không chỉ cải thiện khả năng phát hiện tấn công mà còn tạo ra các cảnh báo sớm và hỗ trợ hệ thống thích nghi linh hoạt với các mối đe dọa mạng mới.

5. KẾT LUẬN

Trong nghiên cứu này, các mô hình học sâu như MLP, RNN, CNN, LSTM, BiLSTM, Autoencoder, GRU và Transformer trong việc phát hiện xâm nhập mạng trên tập dữ liệu UNSW-NB15 đã được tiến hành đánh giá.

Kết quả cho thấy GRU, BiLSTM và Transformer vượt trội hơn so với các mô hình khác nhờ khả năng mô hình hóa các mối quan hệ tuần tự, xử lý hiệu quả các đặc trưng thời gian và nội dung, cũng như tận dụng tốt tính đa chiều của dữ liệu. GRU và BiLSTM, với cơ chế cổng (gates), tập trung vào các đặc trưng quan trọng, giúp cải thiện hiệu suất nhận dạng các loại tấn công như Exploits, Shellcode và Reconnaissance. Đặc biệt, BiLSTM với kiến trúc hai chiều cho phép phân tích cả ngữ cảnh trước và sau, tăng cường khả năng phát hiện. Transformer với cơ chế self-attention, nổi bật trong việc xử lý các đặc trưng phức tạp và không tuần tự, đạt hiệu suất gần như hoàn hảo trên các lớp tấn công phổ biến như Generic và Normal.

Tuy nhiên, các mô hình này vẫn gặp khó khăn trong nhận dạng các lớp như Worms và Backdoor do sự mất cân bằng dữ liệu nghiêm trọng và sự thiếu rõ ràng trong đặc trưng. Lớp Worms với số lượng mẫu cực kỳ hạn chế, dẫn đến việc mô hình không đủ dữ liệu để học các đặc trưng đại diện. Hành vi của Worms như gửi gói tin với thời gian ngắn hoặc kết nối đến nhiều cổng khác nhau, thiếu các đặc điểm rõ ràng khiến ngay cả các mô hình tiên tiến như Transformer cũng khó phân biệt. Lớp Backdoor dù có số lượng mẫu lớn hơn, lại có các đặc trưng nội dung và thời gian tương đồng với lưu lượng mạng bình thường, dẫn đến tỷ lệ phân loại sai cao. Ngoài ra, các mô hình đơn giản hơn như MLP và CNN không đủ khả năng xử lý các mối quan hệ tuần tự và đa chiều trong dữ liệu, dẫn đến hiệu suất thấp hơn, đặc biệt trên các lớp tấn công phức tạp.

Kết quả ở các mô hình này cho thấy tiềm năng của nó bởi hiệu suất cao trong việc nhận diện các hình thức tấn công mạng phổ biến, đặc biệt là khả năng phân loại chính xác các lớp tấn công như Reconnaissance, Generic và Normal, trong khi vẫn xử lý tốt các tấn công phức tạp như Exploits và Shellcode. Việc phân loại đa lớp thành công này không chỉ nâng cao khả năng phát hiện mà còn giúp cải thiện độ chính xác và độ tin cậy của hệ thống phát hiện xâm nhập trong môi trường thực tế, khắc phục hạn chế của các phương pháp phát hiện dựa trên quy tắc truyền thống. Kết quả này góp phần củng cố cơ sở khoa học cho việc áp dụng các giải pháp dựa trên học sâu vào an ninh mạng, hướng tới phát triển các hệ thống phát hiện tấn công chủ động và hiệu quả hơn.

Để tiếp tục nâng cao hiệu quả, nghiên cứu có thể được tập trung vào tối ưu hóa thời gian huấn luyện mô hình thông qua mô hình nhẹ (lightweight model) và lượng tử hóa trọng số, kết hợp các mô hình lai giữa các kiến trúc học sâu hoặc giữa học sâu và học máy để tăng cường khả năng phát hiện. Bên cạnh đó, các phương pháp tăng cường dữ liệu như SMOTE hoặc GANs cần được áp dụng để cân bằng dữ liệu, mở rộng thử nghiệm trên các tập dữ liệu mới nhằm đánh giá tính tổng quát và kết hợp với các kỹ thuật học tăng cường để tối ưu hóa tự động và giảm chi phí tính toán.

TÀI LIỆU THAM KHẢO (REFERENCES)

Al-jabery, K. K., Obafemi-Ajayi, T., Olbricht, G. R., & Wunsch, II, D. C. (2020). Selected approaches to supervised learning. In K. A. Khalid, O. Tayo, R. O. Gayla, & C. W. Donald (Eds.), *Computational Learning Approaches to Data Analytics in Biomedical Applications.* (pp. 101-123). Elsevier. https://doi.org/10.1016/B978-0-12-814482-4.00004-8.

Bertsimas, D., Delarue, A., & Pauphilet, J. (2021). Simple Imputation Rules for Prediction with Missing Data: Contrasting Theoretical Guarantees with Empirical Performance. *arXiv preprint arXiv:2104.03158*. https://doi.org/10.48550/arXiv.2104.03158

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1724–1734). Association for Computational Linguistics. https://doi.org/10.3115/v1/D14-1179

de Amorim, L. B. V., Cavalcanti, G. D. C., & Cruz, R. M. O. (2022). The choice of scaling technique matters for classification performance. *Applied Soft Computing*, *133,* 109924. https://doi.org/10.1016/j.asoc.2022.109924

Feng, C., Wang, H., Lu, N., Chen, T., He, H., Lu, Y., & Tu, X. M. (2014). Log-transformation and its implications for data analysis. *Shanghai Archives of Psychiatry*, *26*(1), 16-20. https://doi.org/10.3969/j.issn.1002-0829.2014.02.009

Gao, Y., Doan, B. G., Zhang, Z., Ma, S., Zhang, J., Fu, A., Nepal, S., & Kim, H. (2020). Backdoor attacks and countermeasures on deep learning: A comprehensive review. *arXiv preprint arXiv:2007.10760*. https://doi.org/10.48550/arXiv.2007.10760

Hicks, S. A., Strümke, I., Thambawita, V., Hammou, M., Riegler, M. A., Halvorsen, P., & Parasa, S.

(2022). On evaluation metrics for medical applications of artificial intelligence. *Scientific Reports, 12*(1), 5979. https://doi.org/10.1038/s41598-022-09954-8

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science, 313*(5786), 504-507. https://doi.org/10.1126/science.1127647

Islam, S., Elmekki, H., Elsebai, A., Bentahar, J., Drawel, N., Rjoub, G., & Pedrycz, W. (2024). A comprehensive survey on applications of transformers for deep learning tasks. *Expert Systems with Applications, 241,* 122666. https://doi.org/10.1016/j.eswa.2024.122666

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2012). Isolation-Based Anomaly Detection. *ACM Transactions on Knowledge Discovery from Data (TKDD), 6*(1), 1-39. https://doi.org/10.1145/2133360.2133363

Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019v4*. https://doi.org/10.48550/arXiv.1506.00019

More, S., Idrissi, M., Mahmoud, H., & Asyhari, AT. (2024) Enhanced Intrusion Detection Systems Performance with UNSW-NB15 Data Analysis. *Algorithms 2024, 17*(2), 64. https://doi.org/10.3390/a17020064

Moustafa, N., & Slay, J. (2015). UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In *Military Communications and Information Systems Conference (MilCIS)* (pp. 1-6). IEEE. https://doi.org/10.1109/MilCIS.2015.7348942

Psathas, A. P., Iliadis, L., Papaleonidas, A., & Pimenidis, E. (2024). HEDL-IDS2: An innovative hybrid ensemble deep learning prototype for cyber intrusion detection. In L. Iliadis, I. Maglogiannis, A. Papaleonidas, E. Pimenidis, & C. Jayne (Eds.), *Engineering applications of neural networks* (pp. 191-206). Springer. https://doi.org/10.1007/978-3-031-62495-7_15

Pansari, N., Srivastava, S., R R, H., & Agarwal, M. (2024). Attack classification using machine learning on UNSW-NB 15 dataset using XGBoost feature selection & ablation analysis. In *2024 IEEE 9th International Conference for Convergence in Technology (I2CT)* (pp. 1-9). IEEE. https://doi.org/10.1109/I2CT61223.2024.10543523

Saabith, S., Thangarajah, V., & Fareez, M. (2023) A survey of machine learning techniques for anomaly detection in cybersecurity. *International Journal of Research in Engineering and Science, 11*(10), 183-193.

Stein, K., Mahyari, A., Francia, G., & El-Sheikh, E. (2024). A Transformer-Based Framework for Payload Malware Detection and Classification *2024 IEEE World AI IoT Congress (AIIoT)* (pp. 105-111). IEEE. http://doi.org/10.1109/AIIoT61789.2024.10579000

TS, P., & Shrinivasacharya, P. (2021). Evaluating neural networks using Bi-Directional LSTM for network IDS (intrusion detection systems) in cyber security. *Global Transitions Proceedings, 2*(2), 448-454. https://doi.org/10.1016/j.gltp.2021.08.017

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need.

Conference on Neural Information Processing. https://doi.org/10.48550/arXiv.1706.03762 Systems (NIPS 2017). Curran Associates Inc.

Wu, Y., Wei, D., & Feng, J. (2020). Network Attacks Detection Methods Based on Deep Learning Techniques: A Survey. *Security and Communication Networks, 2020*(1), Article 8872923. https://doi.org/10.1155/2020/8872923

Yuan, L., Jiang, P., Hou, W., & Huang, W. (2024). G-MLP: Graph multi-layer perceptron for node classification using contrastive learning. *IEEE Access,12,*104909-104919 https://doi.org/10.1109/ACCESS.2024.3432583