

**ĐẠI HỌC CẦN THƠ**

***TRƯỜNG* CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**NIÊN LUẬN CƠ SỞ**

**NGÀNH AN TOÀN THÔNG TIN**

**Đề tài**

**ĐÁNH GIÁ HỆ THỐNG PHÁT HIỆN XÂM NHẬP BẰNG CÁC GIẢI THUẬT HỌC SÂU**

(Evaluation of Intrusion Detection Systems Using Deep Learning Algorithms)

*Giảng viên hướng dẫn:*

**TS. NGUYỄN HỮU VÂN LONG**

*Sinh viên thực hiện:*

**KIM VĂN TRIỆU**

**MSSV: B220345**

**KHÓA 48**

**Cần Thơ, 8/2025**

**ĐẠI HỌC CẦN THƠ**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**NIÊN LUẬN CƠ SỞ**

**NGÀNH AN TOÀN THÔNG TIN**

**Đề tài**

**ĐÁNH GIÁ HỆ THỐNG PHÁT HIỆN XÂM NHẬP BẰNG CÁC GIẢI THUẬT HỌC SÂU**

(Evaluation of Intrusion Detection Systems Using Deep Learning Algorithms)

*Giảng viên hướng dẫn:*

**TS. NGUYỄN HỮU VÂN LONG**

*Sinh viên thực hiện:*

**KIM VĂN TRIỆU**

**MSSV: B2203745**

**KHÓA 48**

**Cần Thơ,8/2025**

MỤC LỤC

[LỜI CẢM ƠN 6](#_Toc20996)

[CHƯƠNG 1: GIỚI THIỆU 1](#_Toc19414)

[1.1. Lý do chọn đề tài. 1](#_Toc14077)

[1.2. Mục tiêu của đề tài. 1](#_Toc20832)

[1.3. Đối tượng nghiên cứu. 2](#_Toc32162)

[1.4. Phạm vi nghiên cứu . 2](#_Toc6602)

[1.5. Phương pháp nghiên cứu. 2](#_Toc26870)

[1.6. Ý nghĩa khoa học và thực tiễn của đề tài. 2](#_Toc15672)

[1.7. Bố cục 3](#_Toc13585)

[Chương 2: Cơ sở lý thuyết 4](#_Toc475)

[2.1. Ngôn ngữ lập trình Python 4](#_Toc2033)

[2.2. Dataset CSE-CIC-IDS2018 5](#_Toc21503)

[2.3. Các giải thuật học sâu. 6](#_Toc15245)

[2.3.1. Deep Neural Network (DNN) 6](#_Toc18736)

[2.3.2. Convolutional Neural Network (CNN) 6](#_Toc8832)

[2.3.3 Recurrent Neural Network (RNN) 6](#_Toc28633)

[2.3.4. Long Short-Term Memory (LSTM) 7](#_Toc7199)

[2.3.5. K-Nearest Neighbors (KNN) 7](#_Toc14796)

[Chương 3: phương pháp thực hiện 8](#_Toc6911)

[3.1. Lựa chọn dữ liệu. 8](#_Toc9127)

[3.2. Tiền xử lí dữ liệu 8](#_Toc13336)

[3.3. Thiết kế mô hình và huấn luyệnmô hình 12](#_Toc13817)

[Chương 4: ĐÁNH GIÁ 14](#_Toc16238)

[4.1. Mô hình CNN 14](#_Toc15284)

[4.2. Mô hình RNN 14](#_Toc27426)

[4.3 Đánh giá chung 16](#_Toc25995)

[Chương 5: Kết luận và Hướng phát triển 18](#_Toc15332)

[5.1. Kết luận. 18](#_Toc2319)

[5.2. Hạn chế & Hướng phát triển. 18](#_Toc3611)

[Tài liệu tham khảo 20](#_Toc25017)

[Phụ lục 20](#_Toc1354)

MỤC LỤC HÌNH ẢNH

[Hình 1 Đọc file và gộp 9](#_Toc27908)

[Hình 2 Lọc và chọn các đặt trưng 9](#_Toc17784)

[Hình 3 Danh sách các đặt trưng 10](#_Toc27905)

[Hình 4 Dữ liệu chưa được cân bằng 11](#_Toc32300)

[Hình 5 Dữ liệu sau khi cân bằng 12](#_Toc9229)

[Hình 6 Ma trận nhầm lẫn của CNN 14](#_Toc5169)

[Hình 7 Ma trận nhầm lẫn mô hình RNN 15](#_Toc5278)

[Hình 8 So sánh các mô hình 16](#_Toc32316)

DANH MỤC TỪ VIẾT TẮT

|  |  |
| --- | --- |
| Từ viết tắt | Viết đầy đủ |
| CNN | Convolutional Neural Network |
| RNN | Recurrent Neural Network |
| LSTM | Long Short-Term Memory |
| MLP | Multi-Layer Perceptron |
| DNN | Deep Neural Network |
| KNN | K-Nearest Neighbors |

# LỜI CẢM ƠN

Lời đầu tiên, tôi xin chân thành cảm ơn Đại học Cần Thơ, Trường Công Nghệ Thông Tin và Truyền Thông đã tạo điều kiện cho tôi thực hiện đề tài này.

Tôi xin cảm ơn gia đình và bạn bè đã luôn ủng hộ và đồng hành cùng tôi trong suốt thời gian học tập. Tôi xin chân thành cảm ơn các anh chị và thầy cô đã giúp đỡ, đóng góp ý kiến và chỉ bảo tôi để tôi có thể hoàn thành nghiên cứu. Đặc biệt, tôi xin gửi lời cảm ơn sâu sắc đến thầy TS Nguyễn Hữu Vân Long người đã tận tình bỏ thời gian hướng dẫn, chỉ bảo giúp tôi hoàn thiện đề tài: “Đánh giá hệ thống phát hiện xâm nhập bằng các giải thuật học sâu”.

Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế, niên luận cơ sở này không thể tránh được những thiếu sót. Tôi rất mong nhận được sự chỉ bảo, đóng góp ý kiến của các thầy cô để tôi có điều kiện bổ sung, nâng cao kiến thức của mình, phục vụ tốt hơn công tác thực tế sau này.

Tôi xin chân thành cảm ơn!

Kim Văn Triệu

Lớp An toàn thông tin K48

# CHƯƠNG 1: GIỚI THIỆU

## Lý do chọn đề tài.

Trong thời đại công nghệ số phát triển mạnh mẽ, các hệ thống mạng máy tính ngày càng trở thành mục tiêu tấn công của nhiều đối tượng với mục đích xâm nhập, đánh cắp dữ liệu, phá hoại hệ thống hoặc gây gián đoạn dịch vụ. Các hình thức tấn công mạng không chỉ gia tăng về số lượng mà còn ngày càng tinh vi, khó nhận diện và có khả năng vượt qua các hệ thống bảo mật truyền thống như tường lửa hay phần mềm chống virus. Điều này đặt ra yêu cầu cấp thiết về việc nghiên cứu và phát triển các phương pháp phát hiện xâm nhập có tính chính xác cao, khả năng thích ứng tốt và phát hiện được cả các hình thức tấn công mới.

Học sâu (Deep Learning) – một lĩnh vực thuộc trí tuệ nhân tạo – đã chứng minh được hiệu quả vượt trội trong nhiều bài toán phức tạp như nhận dạng hình ảnh, xử lý ngôn ngữ tự nhiên và phân tích dữ liệu lớn. Với khả năng tự học các đặc trưng ẩn trong dữ liệu và khai thác mối quan hệ phức tạp giữa các yếu tố, các mô hình học sâu đang ngày càng được ứng dụng rộng rãi trong lĩnh vực an ninh mạng, đặc biệt là trong việc xây dựng hệ thống phát hiện xâm nhập.

Việc kết hợp các giải thuật học sâu như CNN, RNN, DNN, LSTM với dữ liệu mạng có tiềm năng lớn trong việc phát hiện các hành vi bất thường một cách chính xác và kịp thời. Đồng thời, với sự phát triển của các thư viện mã nguồn mở trên Python, việc triển khai các mô hình học sâu trở nên dễ tiếp cận và có thể áp dụng thực tiễn hiệu quả.

Xuất phát từ những lý do nêu trên, nhóm tác giả lựa chọn đề tài "Đánh giá hệ thống phát hiện xâm nhập sử dụng các giải thuật học sâu" nhằm mục tiêu nghiên cứu, đánh giá và thử nghiệm một mô hình phát hiện xâm nhập mạng hiện đại, chính xác và có khả năng ứng dụng thực tế cao.

## Mục tiêu của đề tài.

Đề tài tập trung vào việc đánh giá, so sánh hiệu quả của các mô hình học sâu trong bài toán phát hiện xâm nhập mạng. Mục tiêu chính là đánh giá và phân tích hiệu suất của từng mô hình dựa trên các tiêu chí cụ thể như độ chính xác trong việc nhận diện các loại tấn công và thời gian xử lý – bao gồm cả thời gian huấn luyện mô hình lẫn thời gian dự đoán khi đưa vào vận hành thực tế. Thông qua đó, đề tài hướng đến việc xác định mô hình học sâu nào là tối ưu nhất trong bối cảnh yêu cầu vừa chính xác cao, vừa có khả năng phản hồi nhanh, từ đó có thể đề xuất một giải pháp khả thi cho các hệ thống giám sát an ninh mạng hiện nay.

## Đối tượng nghiên cứu.

Đối tượng nghiên cứu của đề tài là các mô hình học sâu được ứng dụng trong phát hiện xâm nhập mạng, bao gồm các kiến trúc tiêu biểu như Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Deep Neural Network (DNN), Long Short-Term Memory (LSTM) và K-Nearest Neighbors (KNN). Các mô hình này được xây dựng và triển khai bằng ngôn ngữ lập trình Python trong môi trường Google Colab nhằm tận dụng tài nguyên tính toán trên nền tảng đám mây. Tập dữ liệu được sử dụng cho quá trình huấn luyện và đánh giá là bộ dữ liệu CSE-CIC-IDS2018, một trong những tập dữ liệu hiện đại và đầy đủ, mô phỏng nhiều loại tấn công mạng phổ biến hiện nay. Đề tài sẽ tập trung vào việc xử lý dữ liệu, huấn luyện mô hình, và so sánh hiệu quả giữa các mô hình dựa trên độ chính xác, thời gian huấn luyện và thời gian đưa ra kết quả.

## Phạm vi nghiên cứu .

Phạm vi nghiên cứu của đề tài tập trung vào việc áp dụng và so sánh một số mô hình học sâu trong bài toán phát hiện xâm nhập mạng, bao gồm các kiến trúc CNN, RNN, DNN, LSTM và KNN. Quá trình nghiên cứu được thực hiện hoàn toàn trên nền tảng Google Colab, sử dụng ngôn ngữ lập trình Python và các thư viện phổ biến như TensorFlow, Keras, Scikit-learn, v.v. Dữ liệu đầu vào là tập dữ liệu CSE-CIC-IDS2018, trong đó chỉ lựa chọn một phần dữ liệu tiêu biểu để đảm bảo phù hợp với giới hạn tài nguyên xử lý.

## Phương pháp nghiên cứu.

Đề tài sử dụng phương pháp thực nghiệm làm cơ sở chính cho quá trình nghiên cứu. Đầu tiên, dữ liệu từ tập CSE-CIC-IDS2018 sẽ được tiền xử lý bao gồm các bước như làm sạch, mã hóa nhãn, chuẩn hóa các đặc trưng và cân bằng lại tập dữ liệu để tránh tình trạng mất cân đối giữa các lớp. Sau khi xử lý, dữ liệu sẽ được chuyển đổi thành các định dạng phù hợp với từng mô hình học sâu. Chẳng hạn, với mô hình CNN, dữ liệu sẽ được chuyển hóa thành dạng ảnh đầu vào (image dataset), trong khi các mô hình như RNN, DNN, LSTM và KNN sẽ sử dụng dữ liệu dưới dạng vector số (numeric vectors).

Việc huấn luyện và đánh giá mô hình được thực hiện trong môi trường Google Colab, sử dụng ngôn ngữ lập trình Python cùng với các thư viện chuyên dụng như TensorFlow, Keras và Scikit-learn. Mỗi mô hình sẽ được huấn luyện độc lập, sau đó đánh giá hiệu quả dựa trên các tiêu chí như độ chính xác, thời gian huấn luyện và thời gian đưa ra dự đoán. Cuối cùng, kết quả của các mô hình sẽ được so sánh để tìm ra kiến trúc tối ưu nhất cho bài toán phát hiện xâm nhập.

## Ý nghĩa khoa học và thực tiễn của đề tài.

**Ý nghĩa khoa học:**  
Đề tài góp phần làm rõ vai trò và hiệu quả của các mô hình học sâu trong lĩnh vực an ninh mạng, đặc biệt là trong bài toán phát hiện xâm nhập. Việc so sánh giữa các mô hình như CNN, RNN, DNN, LSTM và KNN không chỉ giúp đánh giá khả năng nhận diện tấn công của từng mô hình mà còn mở rộng hiểu biết về cách xử lý và biểu diễn dữ liệu mạng dưới nhiều hình thức khác nhau, như dạng số và dạng ảnh. Kết quả nghiên cứu có thể được sử dụng làm cơ sở tham khảo cho các công trình tiếp theo trong lĩnh vực học máy ứng dụng vào an toàn thông tin.

**Ý nghĩa thực tiễn:**  
Trong bối cảnh các mối đe dọa mạng ngày càng gia tăng về số lượng lẫn mức độ tinh vi, việc xây dựng một hệ thống phát hiện xâm nhập có độ chính xác cao và tốc độ phản hồi nhanh là rất cần thiết. Đề tài cung cấp một hướng tiếp cận khả thi cho việc áp dụng trí tuệ nhân tạo trong giám sát và bảo vệ hệ thống mạng. Việc đánh giá thời gian huấn luyện và thời gian dự đoán của các mô hình cũng mang lại giá trị thực tiễn cao, giúp lựa chọn được giải pháp phù hợp với yêu cầu tài nguyên và hiệu suất trong triển khai thực tế.

## Bố cục

Chương 1: Giới thiệu

Chương 2: Cơ sở lý thuyết

Chương 3: Phương pháp thực hiện

Chương 4: Đánh giá

Chương 5: Kết luận và hướng phát triển

**Tổng kết chương 1:** Trong chương này, niên luận cơ sỡ nêu lên mục tiêu , lý do chọn đề tài, đối tượng nghiên cứu, phạm vi nghiên cứu, các phương pháp nghiên cứu đề tài và ý nghĩa khoa học, thực tiễn của đề tài lựa chọn. Trong chương 2, niên luận cơ sở sẽ trình bày các cơ sở lý thuyết liên quan đến các phương pháp nghiên cứu.

# Chương 2: Cơ sở lý thuyết

Trong chương này, niên luận cơ sở sẽ trình bày các cơ sở lý thuyết chính được sử dụng trong ứng dụng, nội dung bao gồm như sau:

* Ngôn ngữ lập trình Python và các thư viện
* Dataset CSE-CIC-IDS2018
* Các giải thuật học sâu:
* **Deep Neural Network (DNN**
* **Convolutional Neural Network (CNN)**
* **Recurrent Neural Network (RNN)**
* **Long Short-Term Memory (LSTM)**
* **K-Nearest Neighbors (KNN)**

## Ngôn ngữ lập trình Python

Python là một ngôn ngữ lập trình thông dịch, bậc cao. Ngôn ngữ này được sử dụng rộng rãi trong nhiều lĩnh vực khác nhau, đặc biệt là khoa học dữ liệu, trí tuệ nhân tạo, học máy và phát triển web. Python ban đầu được phát triển bởi **Guido van Rossum** và ra mắt lần đầu vào năm 1991 dưới sự bảo trợ của Stichting Mathematisch Centrum (CWI) tại Hà Lan.

Khác với nhiều ngôn ngữ khác, Python được thiết kế với triết lý "code rõ ràng và dễ đọc", với cú pháp đơn giản, dễ tiếp cận cho cả người mới bắt đầu và các lập trình viên chuyên nghiệp. Mặc dù cú pháp của Python có sự tương đồng với các ngôn ngữ như C hay Java, nhưng nó mang tính tối giản và linh hoạt hơn, phù hợp cho việc phát triển nhanh các nguyên mẫu cũng như triển khai các hệ thống lớn.

Python sử dụng phần mở rộng **.py** cho các tập tin mã nguồn, và có thể được thực thi trực tiếp thông qua trình thông dịch Python. Với sự phát triển mạnh mẽ của cộng đồng, Python hiện có một hệ sinh thái thư viện phong phú như **NumPy**, **Pandas**, **Scikit-learn**, **TensorFlow**, **Keras**, giúp hỗ trợ hiệu quả cho việc xử lý dữ liệu và xây dựng mô hình học sâu.

#### Các thư viện được sử dụng

**NumPy** là một thư viện mã nguồn mở trong Python, được thiết kế để hỗ trợ các thao tác tính toán khoa học, đặc biệt là xử lý mảng nhiều chiều (ndarray). NumPy cung cấp các hàm toán học hiệu quả cao, giúp thực hiện các phép tính đại số tuyến tính, thống kê, biến đổi Fourier, v.v. Nó được xem là nền tảng cho hầu hết các thư viện học máy và học sâu trong hệ sinh thái Python.

**Pandas** là một thư viện mạnh mẽ dùng để thao tác và phân tích dữ liệu dạng bảng (dataframe). Pandas cung cấp cấu trúc dữ liệu linh hoạt và các hàm xử lý dữ liệu như lọc, gom nhóm, nối bảng, xử lý dữ liệu thiếu, chuyển đổi định dạng, rất hữu ích trong giai đoạn tiền xử lý dữ liệu trước khi đưa vào mô hình học sâu.

**Matplotlib** là một thư viện vẽ đồ thị phổ biến trong Python, cho phép trực quan hóa dữ liệu dưới dạng biểu đồ đường, biểu đồ cột, biểu đồ phân tán, bản đồ nhiệt, v.v. Thư viện này hỗ trợ quá trình phân tích dữ liệu và trình bày kết quả huấn luyện mô hình một cách trực quan.

**Scikit-learn** là một thư viện học máy được xây dựng trên NumPy, SciPy và matplotlib, cung cấp các công cụ cho tiền xử lý dữ liệu, lựa chọn đặc trưng, huấn luyện và đánh giá các thuật toán học máy truyền thống như KNN, SVM, Decision Tree, Random Forest, v.v. Trong đề tài này, Scikit-learn được sử dụng để xử lý dữ liệu và đánh giá hiệu suất mô hình bằng các chỉ số như độ chính xác, độ nhạy, độ đặc hiệu và F1-score.

**TensorFlow** là một thư viện mã nguồn mở được phát triển bởi Google, chuyên dùng cho xây dựng và huấn luyện các mô hình học sâu. TensorFlow hỗ trợ tính toán trên GPU và TPU, cho phép mô hình học sâu được huấn luyện với hiệu suất cao và tốc độ nhanh.

**Keras** là một API cấp cao chạy trên nền tảng TensorFlow, được thiết kế với mục tiêu đơn giản hóa quá trình xây dựng mạng nơ-ron. Với Keras, người dùng có thể dễ dàng tạo và huấn luyện các mô hình như DNN, CNN, RNN hay LSTM chỉ bằng vài dòng mã, đồng thời theo dõi quá trình huấn luyện và tinh chỉnh siêu tham số một cách thuận tiện.

## Dataset CSE-CIC-IDS2018

### CSE-CIC-IDS2018 là một trong những tập dữ liệu lớn và toàn diện trong lĩnh vực phát hiện xâm nhập mạng, được xây dựng bởi sự hợp tác giữa Canadian Institute for Cybersecurity (CIC) và Communications Security Establishment (CSE) của Canada.

CSE-CIC-IDS2018 được tạo ra bằng cách mô phỏng một hệ thống mạng thực tế bao gồm nhiều loại thiết bị (máy chủ, máy khách, tường lửa, router…) và thực hiện các hoạt động mạng bình thường xen kẽ với các cuộc tấn công có chủ đích. Dữ liệu được thu thập bằng công cụ CICFlowMeter, sau đó được trích xuất ra các tệp CSV chứa thông tin chi tiết về mỗi luồng dữ liệu (flow), bao gồm hơn 80 đặc trưng như: địa chỉ IP, cổng, thời gian bắt đầu và kết thúc, số lượng gói tin, tốc độ truyền, độ trễ, giao thức sử dụng, v.v.

Tập dữ liệu bao gồm nhiều loại hình tấn công phổ biến hiện nay, được tổ chức theo từng ngày, tương ứng với từng tình huống giả lập cụ thể trong môi trường mạng thực tế. Các cuộc tấn công được mô phỏng một cách có chủ đích và đa dạng, bao gồm tấn công **Brute Force** vào các dịch vụ như SSH và FTP, các hình thức **tấn công từ chối dịch vụ** bao gồm cả **DoS** và **DDoS**, các **tấn công web** như **XSS**, **SQL Injection** và **Command Injection**. Ngoài ra, tập dữ liệu còn mô phỏng các hoạt động **botnet**, hành vi **xâm nhập trái phép (infiltration)**, khai thác lỗ hổng **Heartbleed**, và các hoạt động **quét cổng (PortScan)**. Mỗi loại tấn công đều được gắn nhãn cụ thể, giúp việc huấn luyện và đánh giá mô hình trở nên rõ ràng và có hệ thống.

Dữ liệu được gán nhãn rõ ràng, chia thành hai nhóm: **Normal** và **Attack**, trong đó nhóm "Attack" được chia tiếp theo từng loại tấn công. Đây là một điểm mạnh quan trọng, giúp các mô hình học máy và học sâu có thể học được đặc trưng của từng loại hành vi tấn công một cách chính xác hơn.

Với tính đầy đủ, đa dạng và có cấu trúc rõ ràng, CSE-CIC-IDS2018 được xem là một nguồn dữ liệu chuẩn mực cho các nghiên cứu và đánh giá hệ thống phát hiện xâm nhập hiện nay. Trong khuôn khổ đề tài này, tập dữ liệu sẽ được xử lý và trích xuất thành các tập con phù hợp, ví dụ như chuyển đổi thành ảnh (cho mô hình CNN) hoặc vector số (cho các mô hình DNN, LSTM, KNN...) để tiến hành huấn luyện và đánh giá.

## Các giải thuật học sâu.

### 2.3.1. Deep Neural Network (DNN)

DNN là một dạng mạng nơ-ron truyền thống gồm nhiều lớp ẩn (hidden layers). Mỗi lớp trong DNN thực hiện các phép biến đổi phi tuyến lên dữ liệu, từ đó học được các đặc trưng ngày càng trừu tượng. DNN thường hoạt động tốt trên dữ liệu dạng bảng (tabular) và thích hợp cho các tập dữ liệu đã được chuẩn hóa. DNN có thể phát hiện các mẫu tấn công nếu được huấn luyện đầy đủ, tuy nhiên mô hình này dễ bị overfitting nếu không kiểm soát tốt tham số và số lượng lớp.

### 2.3.2. Convolutional Neural Network (CNN)

CNN là mạng nơ-ron được thiết kế đặc biệt cho dữ liệu hình ảnh. Tuy nhiên, trong nghiên cứu này, dữ liệu mạng sẽ được chuyển đổi thành dạng ảnh (image-like data) để tận dụng sức mạnh của CNN. Các lớp tích chập (convolutional layers) trong CNN giúp phát hiện các đặc trưng không gian, lọc nhiễu và phân loại hiệu quả. CNN có ưu điểm về tốc độ huấn luyện và khả năng học đặc trưng tự động từ dữ liệu phức tạp.

### 2.3.3 Recurrent Neural Network (RNN)

RNN là một loại mạng nơ-ron có khả năng xử lý dữ liệu theo chuỗi (sequence data). Nhờ vào cấu trúc kết nối vòng, RNN có thể lưu trữ thông tin từ các bước thời gian trước đó để hỗ trợ quá trình dự đoán hiện tại. Trong bài toán phát hiện xâm nhập, RNN có thể áp dụng để phân tích các chuỗi hoạt động mạng hoặc hành vi tấn công có tính tuần tự. Tuy nhiên, RNN truyền thống thường gặp vấn đề khi xử lý các chuỗi dài do hiện tượng "quên dần" thông tin.

### 2.3.4. Long Short-Term Memory (LSTM)

LSTM là một cải tiến quan trọng của RNN, có khả năng ghi nhớ thông tin dài hạn tốt hơn nhờ vào cơ chế cổng (gates). Với cấu trúc gồm các cổng đầu vào, cổng quên và cổng đầu ra, LSTM kiểm soát việc lưu trữ và loại bỏ thông tin một cách hiệu quả. Điều này giúp mô hình học được các mẫu tấn công diễn ra theo chuỗi dài hoặc có tính phức tạp theo thời gian. LSTM là lựa chọn tốt cho các bài toán cần phát hiện các thay đổi tinh vi hoặc tấn công có hành vi kéo dài.

### 2.3.5. K-Nearest Neighbors (KNN)

KNN là một thuật toán học máy đơn giản, không phải là học sâu, nhưng được đưa vào nghiên cứu nhằm mục đích so sánh. KNN hoạt động dựa trên nguyên tắc "gần giống thì cùng nhóm": khi có một điểm dữ liệu mới, thuật toán tìm **k** điểm gần nhất trong không gian đặc trưng và phân loại dựa trên đa số. KNN dễ hiểu và không cần huấn luyện (lazy learner), nhưng hiệu suất có thể giảm nếu dữ liệu có chiều cao hoặc không được chuẩn hóa tốt.

**Tổng kết chương 2:** Trong chương này, niên luận cơ sở đã trình bày cơ sở lý thuyết về các thành phần có sử dụng để xây dựng hệ thống và đánh giá các mô hình chương tiếp theo sẽ trình bày phương pháp thực hiện.

# Chương 3: phương pháp thực hiện

Trong trong đề tài này sẽ thực hiện các công việc như sau: lựa chọn dữ liệu, tiền xử lí dữ liệu, thiết kế mô hình và huấn luyện mô hình.

Trong chương 3 của niên luận cơ sở sẽ trình bày các phương pháp thực hiện

## Lựa chọn dữ liệu.

Tập dữ liệu CSE-CIC-IDS2018 được chia theo từng ngày, mỗi ngày mô phỏng một kịch bản khác nhau, từ đó tạo điều kiện thuận lợi để kiểm tra mô hình trong nhiều tình huống tấn công khác nhau. Việc dữ liệu đã được gán nhãn rõ ràng giúp dễ dàng trong việc đánh giá kết quả và so sánh giữa các thuật toán. Bên cạnh đó, CSE-CIC-IDS2018 còn cung cấp nhiều đặc trưng mạng có thể khai thác để huấn luyện, đồng thời cân bằng tương đối tốt giữa dữ liệu bình thường và dữ liệu tấn công, phù hợp với yêu cầu của các mô hình học sâu. Chính những lý do trên khiến tập dữ liệu này được lựa chọn làm nền tảng cho toàn bộ quá trình xây dựng và đánh giá hệ thống phát hiện xâm nhập.

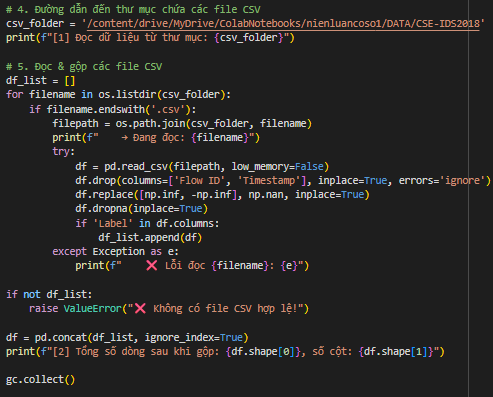
.

## Tiền xử lí dữ liệu

Trước khi đưa dữ liệu vào huấn luyện, cần tiến hành một loạt bước tiền xử lý nhằm chuẩn hóa, làm sạch và chuyển đổi dữ liệu phù hợp với yêu cầu của mô hình học sâu. Các bước này bao gồm:tải dữ liệu, lọc và chọn đặt trưng, mã hóa nhãn và xử lý giá trị thiếu, chuẩn hóa và cân bằng dữ liệu, định dạng dữ liệu theo mô hình.

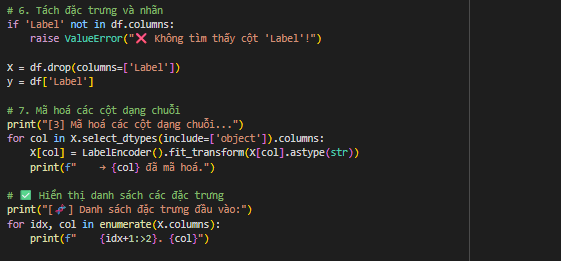
**3.2.1.Tải dữ liệu**

Quá trình bắt đầu bằng việc **tải dữ liệu đầu vào** từ tập CSE-CIC-IDS2018. Các tệp dữ liệu dưới dạng CSV được đọc từ thư mục lưu trữ trên Google Drive, mỗi tệp đại diện cho một ngày thu thập lưu lượng mạng với các loại tấn công khác nhau. Ở bước này, hệ thống chỉ thực hiện thao tác **đọc và tổng hợp dữ liệu gốc**, chưa can thiệp vào nội dung hay cấu trúc dữ liệu. Mục tiêu là tập hợp đầy đủ các nguồn dữ liệu cần thiết để chuẩn bị cho các bước xử lý kế tiếp như lọc đặc trưng, làm sạch và chuẩn hóa.



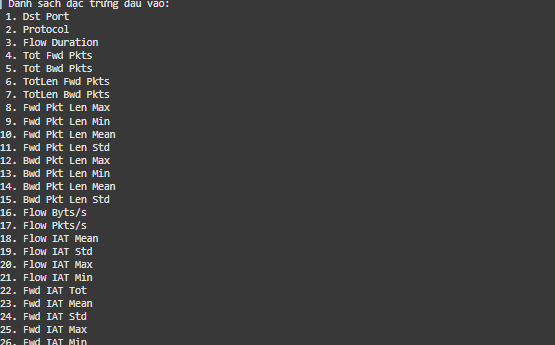
Hình 1 Đọc file và gộp

**3.2.2.Lọc và chọn đặt trưng**



Hình 2 Lọc và chọn các đặt trưng

Trong tập dữ liệu CIC-IDS2018, mỗi phiên kết nối mạng (network flow) được mô tả bằng hàng chục đặc trưng khác nhau, phản ánh đầy đủ các khía cạnh như thời gian truyền, kích thước gói tin, tốc độ luân chuyển dữ liệu, và các cờ điều khiển trong giao thức TCP/IP. Tuy nhiên, không phải tất cả các đặc trưng đều có giá trị như nhau trong việc phát hiện các hành vi bất thường hay các loại tấn công mạng. Vì vậy, việc phân tích, lọc và lựa chọn các đặc trưng quan trọng là bước then chốt trong quá trình xây dựng hệ thống phát hiện xâm nhập.



Hình 3 Danh sách các đặt trưng

Thông qua việc tham khảo các công trình nghiên cứu trước đó, kết hợp với kiến thức chuyên môn về an ninh mạng, có thể xác định một số đặc trưng có vai trò đặc biệt quan trọng trong việc phân loại giữa lưu lượng bình thường và lưu lượng tấn công. Trong đó, các đặc trưng liên quan đến thời gian như **Flow Duration**, **Flow IAT Mean**, hoặc **Idle Mean** thường có sự chênh lệch rõ rệt giữa hành vi hợp lệ của người dùng và các hành vi tấn công tự động. Ví dụ, các cuộc tấn công brute force hoặc port scan thường tạo ra một lượng lớn kết nối ngắn với tốc độ cao, dẫn đến thời gian phiên ngắn và khoảng cách giữa các gói tin rất thấp.

Ngoài ra, các đặc trưng về số lượng và độ dài gói tin như **Tot Fwd Pkts**, **TotLen Fwd Pkts**, **Pkt Len Mean** hay **Flow Byts/s** cũng thể hiện rõ mức độ bất thường của lưu lượng mạng. Những đặc trưng này thường cho thấy sự tăng đột biến trong lượng dữ liệu truyền đi khi xảy ra tấn công từ chối dịch vụ (DoS/DDoS) hoặc botnet. Đặc biệt, các cờ điều khiển trong giao thức TCP như **SYN Flag Count** cũng đóng vai trò quan trọng khi phát hiện các cuộc tấn công lợi dụng lỗ hổng bắt tay ba bước (three-way handshake) như SYN Flood.

Như vậy, có thể khẳng định rằng các đặc trưng thuộc nhóm **thời gian**, **kích thước gói tin**, **tốc độ truyền**, và **các cờ TCP/IP** là những yếu tố quan trọng nhất cần được ưu tiên giữ lại trong quá trình huấn luyện mô hình học sâu. Việc chọn lọc đúng các đặc trưng quan trọng không những giúp giảm độ phức tạp tính toán, mà còn góp phần nâng cao độ chính xác, giảm hiện tượng quá khớp và cải thiện khả năng tổng quát hóa của hệ thống phát hiện xâm nhập.

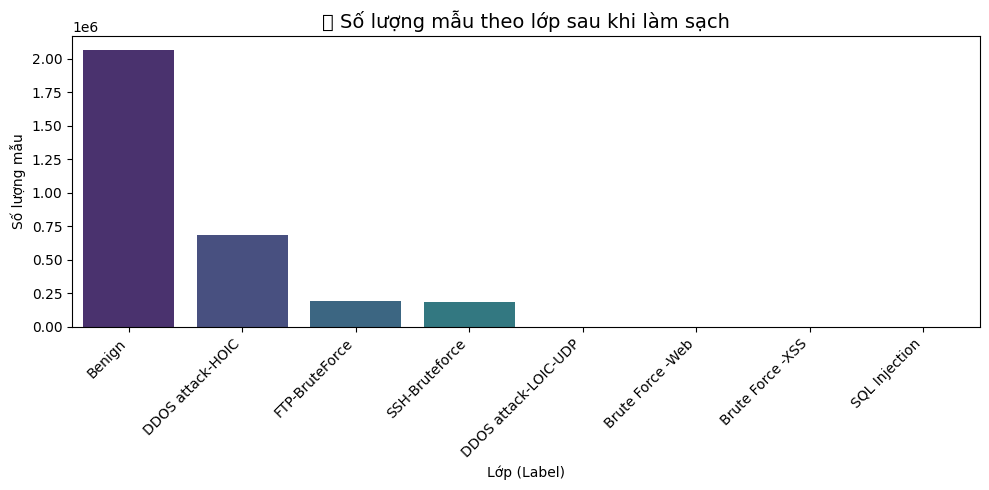
**3.2.3. Mã hóa nhãn và xử lí dữ liệu bị thiếu.**

Sau khi hoàn tất quá trình lọc và chọn đặc trưng, một bước quan trọng tiếp theo trong tiền xử lý dữ liệu là mã hóa nhãn và xử lý các giá trị bị thiếu. Trong tập dữ liệu CIC-IDS2018, mỗi dòng dữ liệu đại diện cho một phiên kết nối mạng được gắn nhãn tương ứng với loại hành vi mạng, bao gồm cả lưu lượng bình thường (Benign) và nhiều loại tấn công khác nhau như DoS, DDoS, Brute Force, PortScan, Botnet, v.v. Tuy nhiên, vì hầu hết các mô hình học sâu yêu cầu đầu vào là dạng số, nên các nhãn chuỗi (dạng văn bản) cần được chuyển đổi sang dạng số nguyên để dễ dàng xử lý và huấn luyện mô hình.

Bên cạnh đó, dữ liệu thô trong CIC-IDS2018 cũng có thể chứa các giá trị bị thiếu (missing values) hoặc không hợp lệ. Những giá trị này nếu không được xử lý đúng cách có thể gây ra lỗi trong quá trình huấn luyện hoặc làm giảm độ chính xác của mô hình. Do đó, tiến hành kiểm tra toàn bộ tập dữ liệu để phát hiện các ô trống, ô chứa giá trị “NaN”, hoặc các giá trị không hợp lệ như “Infinity”, “–1”, hoặc “?”. Tùy theo mức độ và tỷ lệ thiếu hụt, các phương pháp xử lý dữ liệu thiếu bao gồm: loại bỏ hoàn toàn dòng dữ liệu bị lỗi nếu số lượng không đáng kể; hoặc thay thế giá trị thiếu bằng các giá trị trung bình, trung vị (median) hoặc giá trị phổ biến nhất (mode) trong cột tương ứng. Cách tiếp cận này giúp đảm bảo tính toàn vẹn của dữ liệu, đồng thời duy trì chất lượng đầu vào cho các mô hình học sâu mà không làm sai lệch phân bố thống kê ban đầu.

**3.2.4. Chuẩn hóa và cân bằng dữ liệu**

Sau khi thu thập và tổng hợp dữ liệu từ nhiều tệp CSV chứa lưu lượng mạng, quá trình tiền xử lý dữ liệu bắt đầu với việc lọc bỏ các cột không cần thiết như Flow ID và Timestamp, đồng thời loại bỏ các giá trị thiếu và vô hạn để đảm bảo tính sạch sẽ của tập dữ liệu.

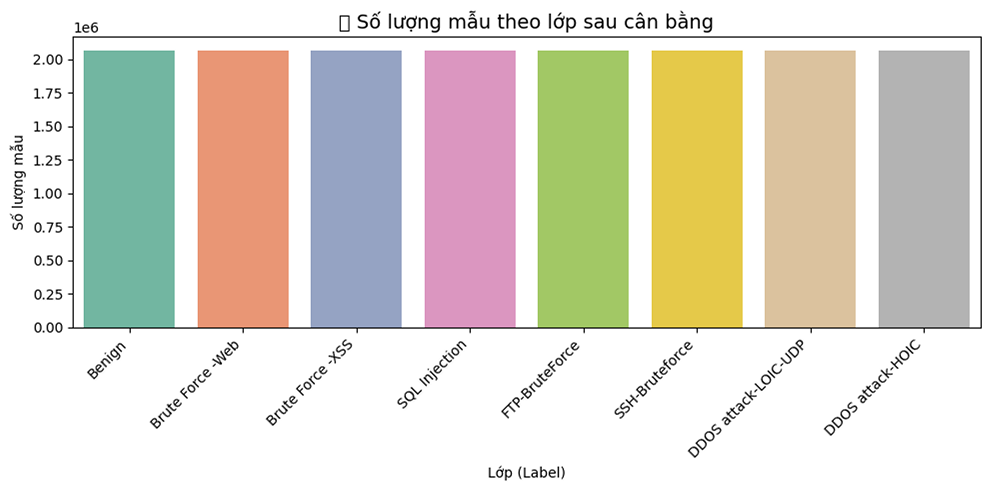


Hình 4 Dữ liệu chưa được cân bằng

Kỹ thuật **Random OverSampling** được áp dụng nhằm nhân bản các mẫu thuộc lớp thiểu số, giúp cân bằng lại phân phối giữa các loại tấn công và dữ liệu bình thường. Sau khi cân bằng, mỗi lớp đều có số lượng mẫu tương đương, tăng khả năng học của mô hình mà không bị thiên lệch về các lớp phổ biến.

Kế đến, các đặc trưng đầu vào được chuẩn hóa bằng StandardScaler – một bước quan trọng nhằm đưa toàn bộ dữ liệu về cùng một phân phối chuẩn, giúp các thuật toán học máy hoạt động ổn định và chính xác hơn. Cuối cùng, dữ liệu được chia thành hai tập: huấn luyện (80%) và kiểm tra (20%) theo phương pháp chia ngẫu nhiên có phân tầng (stratified split) để giữ nguyên tỉ lệ lớp.

Toàn bộ dữ liệu đã xử lý được lưu lại dưới dạng .csv và .npy, giúp dễ dàng sử dụng trong các bước huấn luyện mô hình học sâu sau này. Đồng thời, danh sách các đặc trưng đầu vào cũng được ghi ra một file .txt để tiện theo dõi và tái sử dụng.



Hình 5 Dữ liệu sau khi cân bằng

3.2.5 Định dạng dữ liệu theo mô hình

Sau khi đã mã hóa nhãn, xử lý dữ liệu bị thiếu, chuẩn hóa và cân bằng tập dữ liệu, bước tiếp theo trong quá trình xây dựng hệ thống phát hiện xâm nhập là định dạng lại dữ liệu để phù hợp với các mô hình học sâu.

Mỗi dòng dữ liệu ban đầu là một vector một chiều (1D), chứa khoảng 78 đến 85 đặc trưng sau khi lọc và chọn. Để chuyển đổi các vector này thành ảnh hai chiều (2D), ta tiến hành đưa mỗi vector về dạng ma trận vuông gần nhất, ví dụ như 9x9 hoặc 10x10. Trong trường hợp số lượng đặc trưng không vừa đủ để tạo thành một ma trận vuông hoàn chỉnh, các giá trị còn thiếu sẽ được đệm thêm bằng số 0 (zero padding).

Sau khi định dạng lại, mỗi mẫu dữ liệu sẽ trở thành một ảnh xám (grayscale) có một kênh màu duy nhất, phù hợp với kiến trúc đầu vào chuẩn của mô hình CNN. Việc chuyển dữ liệu sang định dạng ảnh không chỉ giúp tận dụng khả năng trích xuất đặc trưng không gian của mạng CNN, mà còn mở ra hướng tiếp cận mới cho bài toán phát hiện xâm nhập, nơi mà mô hình học sâu có thể tự học ra các mẫu và hành vi bất thường trong dữ liệu mạng.

## Thiết kế mô hình và huấn luyệnmô hình

Trong đề tài này, việc thiết kế mô hình học sâu nhằm phát hiện các hành vi xâm nhập trong hệ thống mạng được thực hiện với nhiều thuật toán khác nhau, nhằm so sánh và đánh giá hiệu quả giữa các phương pháp tiếp cận. Các mô hình được lựa chọn bao gồm CNN (Convolutional Neural Network), RNN (Recurrent Neural Network), LSTM (Long Short-Term Memory), DNN (Deep Neural Network), KNN (K-Nearest Neighbors) và MLP (Multilayer Perceptron). Mỗi mô hình phản ánh một cách xử lý dữ liệu khác nhau, từ phân tích không gian, chuỗi thời gian, đến phân loại dựa trên khoảng cách.

Mô hình CNN được áp dụng trong trường hợp dữ liệu số được chuyển đổi thành ảnh. Dạng biểu diễn này giúp khai thác tốt khả năng học đặc trưng không gian của CNN, từ đó mô hình có thể nhận diện các mẫu bất thường trong lưu lượng mạng dưới dạng trực quan. Trong khi đó, RNN và LSTM lại được sử dụng để xử lý dữ liệu theo chuỗi thời gian, phù hợp với bản chất liên tục và tuần tự của các phiên kết nối mạng. Đặc biệt, LSTM khắc phục được hiện tượng mất nhớ dài hạn, giúp mô hình nắm bắt được các mối quan hệ xa trong chuỗi dữ liệu.

Với dữ liệu vẫn giữ nguyên dưới dạng vector đặc trưng sau khi đã được mã hóa và chuẩn hóa, các mô hình như DNN và MLP được triển khai. Đây là các mạng neural nhiều lớp, có khả năng học các quan hệ phi tuyến giữa các đặc trưng và đưa ra dự đoán chính xác. MLP là một dạng đặc biệt của DNN với kiến trúc đơn giản hơn, thường dùng làm cơ sở để so sánh với các mạng phức tạp hơn. Ngoài ra, mô hình KNN cũng được đưa vào để đóng vai trò là một phương pháp baseline. Dù không phải học sâu, KNN vẫn là một thuật toán truyền thống mạnh trong phân loại, giúp đối chiếu hiệu quả giữa các mô hình hiện đại và cổ điển.

Tất cả các mô hình đều được xây dựng và huấn luyện bằng ngôn ngữ lập trình Python, sử dụng các thư viện mạnh mẽ như TensorFlow, Keras và Scikit-learn. Quá trình huấn luyện được theo dõi thông qua các chỉ số đánh giá như độ chính xác (Accuracy), độ bao phủ (Recall), độ chính xác (Precision), F1-score và đường cong ROC. Việc huấn luyện, đánh giá và so sánh các mô hình giúp xác định phương pháp tối ưu để áp dụng vào hệ thống phát hiện xâm nhập trong thực tế, đồng thời mở ra hướng phát triển cho các nghiên cứu sau này.

**Tổng kết chương 3:** Trong chương 3 này đã đưa ra các bước xử lí dữ liệu cũng như huấn luyện mô hình.

Chương tiếp theo sẽ giới thiệu kết quả đánh giá các mô hình.

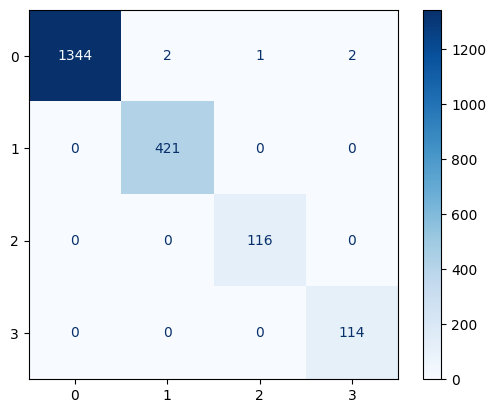
# Chương 4: ĐÁNH GIÁ

## Mô hình CNN

Mô hình CNN được triển khai trong đề tài này nhằm khai thác khả năng học đặc trưng không gian thông qua việc chuyển đổi dữ liệu số thành ảnh. Việc định dạng dữ liệu theo dạng hình ảnh giúp mô hình tận dụng được kiến trúc tích chập – vốn mạnh trong việc phát hiện các mẫu phức tạp và đặc trưng cục bộ. Quá trình huấn luyện CNN được thực hiện trên tập dữ liệu đã được cân bằng và chuẩn hóa, với kích thước ảnh được chuẩn hóa về dạng cố định (ví dụ 28×28 hoặc 32×32) để tương thích với đầu vào của mạng.

Kết quả cho thấy mô hình CNN đạt hiệu suất tốt trong việc phân loại các loại tấn công mạng khác nhau so với dữ liệu bình thường.

Tuy nhiên, CNN cũng bộc lộ một số hạn chế. Đầu tiên, việc chuyển đổi dữ liệu từ dạng vector đặc trưng sang dạng ảnh có thể làm mất một phần ngữ nghĩa ban đầu nếu không thực hiện đúng cách, ảnh hưởng đến hiệu quả của mô hình. Thứ hai, do đặc thù kiến trúc, CNN không xử lý tốt các quan hệ theo chuỗi thời gian – điều có thể quan trọng trong một số loại tấn công cần nắm bắt mối quan hệ theo phiên hoặc theo dòng thời gian.



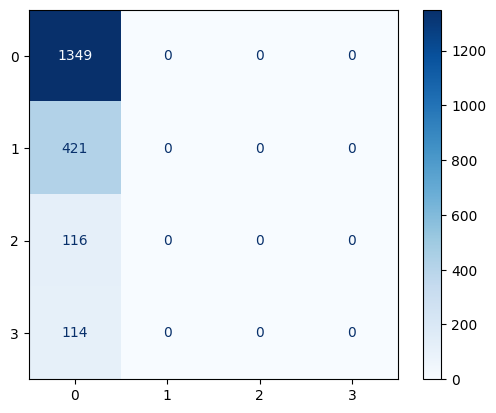
Hình 6 Ma trận nhầm lẫn của CNN

## Mô hình RNN

Mô hình RNN được triển khai trong nghiên cứu này nhằm khai thác đặc trưng theo **trình tự thời gian** trong dữ liệu lưu lượng mạng. Khác với các kiến trúc như CNN hay DNN vốn chỉ xử lý độc lập các mẫu dữ liệu, RNN có khả năng duy trì trạng thái ẩn (hidden state) giúp mô hình ghi nhớ các thông tin từ các bước thời gian trước đó. Đặc điểm này khiến RNN đặc biệt phù hợp với bài toán phát hiện xâm nhập mạng, nơi các sự kiện tấn công thường xảy ra liên tiếp và theo chuỗi có liên quan.

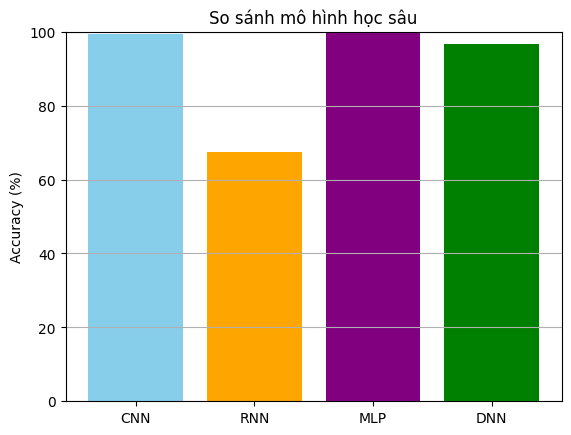
Trong quá trình thực nghiệm, dữ liệu đầu vào được chuẩn hóa và sắp xếp lại dưới dạng chuỗi – chẳng hạn bằng cách gom các gói tin theo phiên (session) hoặc theo cửa sổ thời gian. Mỗi chuỗi sau đó được đưa vào RNN như một dãy các bước thời gian liên tục, giúp mô hình học được mối quan hệ ngữ cảnh giữa các sự kiện mạng.

Kết quả cho thấy RNN đạt hiệu suất tốt trong việc phát hiện các loại tấn công liên tục hoặc có tính thời gian rõ rệt, chẳng hạn như DoS hoặc Brute Force



Hình 7 Ma trận nhầm lẫn mô hình RNN

### **4.3 Đánh giá chun**g



Hình 8 So sánh các mô hình

Dựa trên biểu đồ so sánh độ chính xác giữa các mô hình học sâu, có thể nhận thấy sự khác biệt rõ rệt về hiệu quả phát hiện xâm nhập mạng giữa các thuật toán. Mô hình CNN (Convolutional Neural Network) cho thấy độ chính xác cao nhất, gần như tuyệt đối, khi đạt ngưỡng 100%. Điều này cho thấy rằng CNN đặc biệt phù hợp với cách tiếp cận chuyển đổi dữ liệu số sang dạng ảnh – vốn là cách mà mô hình này phát huy tối đa khả năng học các đặc trưng cục bộ và nhận diện mẫu dữ liệu bất thường. Với khả năng trích xuất đặc trưng mạnh mẽ thông qua các tầng tích chập, CNN trở thành lựa chọn lý tưởng cho việc phát hiện xâm nhập trong không gian dữ liệu hình ảnh.

MLP (Multilayer Perceptron) và DNN (Deep Neural Network) cũng thể hiện kết quả khả quan. Cả hai mô hình đều hoạt động hiệu quả trên dữ liệu số đã được chuẩn hóa, với độ chính xác lần lượt xấp xỉ 100% và 96%. MLP đạt được hiệu suất gần tương đương CNN khi sử dụng kiến trúc nhiều tầng ẩn đơn giản, trong khi DNN với kiến trúc sâu hơn có thể học được các quan hệ phức tạp hơn trong dữ liệu nhưng đồng thời cũng dễ rơi vào trạng thái quá khớp nếu không được điều chỉnh cẩn thận.

Ngược lại, mô hình RNN (Recurrent Neural Network) chỉ đạt độ chính xác khoảng 67%, thấp nhất trong số các mô hình được so sánh. Mặc dù RNN có ưu thế trong xử lý chuỗi thời gian hoặc dữ liệu tuần tự, tuy nhiên trong trường hợp này – nơi dữ liệu không có yếu tố tuần tự rõ ràng hoặc không được xử lý theo dạng chuỗi – hiệu suất của RNN bị hạn chế. Ngoài ra, việc không áp dụng các cải tiến như LSTM hay GRU khiến RNN khó khai thác được các quan hệ phức tạp trong dữ liệu.

Từ kết quả thu được, có thể kết luận rằng CNN là mô hình phù hợp nhất khi dữ liệu được biểu diễn dưới dạng hình ảnh, trong khi MLP và DNN cũng là những lựa chọn hợp lý nếu tiếp cận bài toán theo hướng truyền thống sử dụng dữ liệu dạng số. RNN không đạt hiệu quả cao trong bối cảnh này, trừ khi có thêm các cải tiến phù hợp. Kết quả đánh giá đã cung cấp cơ sở thực nghiệm vững chắc để lựa chọn mô hình phù hợp trong hệ thống phát hiện xâm nhập sử dụng học sâu.

**Tổng kết chương 4:** các mô hình như CNN, RNN, DNN, và MLP được xây dựng và so sánh. Kết quả cho thấy CNN đạt độ chính xác cao nhất, chứng minh hiệu quả của việc chuyển dữ liệu sang dạng ảnh. Các mô hình còn lại như DNN và MLP cũng cho hiệu suất tốt, trong khi RNN không phù hợp với dữ liệu không tuần tự.

# Chương 5: Kết luận và Hướng phát triển

## Kết luận.

Kết quả thực nghiệm cho thấy mô hình CNN mang lại hiệu suất vượt trội so với các mô hình còn lại nhờ khả năng khai thác mối quan hệ giữa các đặc trưng theo dạng không gian. Mô hình DNN và MLP cũng cho thấy độ chính xác khá cao trong việc phân loại các luồng dữ liệu mạng. Trong khi đó, các mô hình RNN và LSTM – vốn thường hiệu quả trong xử lý chuỗi – lại không mang lại kết quả tốt như kỳ vọng, do dữ liệu mạng không mang tính tuần tự rõ rệt. Mô hình KNN, mặc dù đơn giản và dễ triển khai, lại tỏ ra kém hiệu quả khi áp dụng vào bài toán có tập dữ liệu lớn và không đồng đều như CIC-IDS2018.

Từ các kết quả đạt được, có thể khẳng định rằng việc kết hợp các kỹ thuật tiền xử lý phù hợp với các kiến trúc học sâu hiện đại là yếu tố then chốt giúp nâng cao hiệu suất phát hiện xâm nhập. Việc chuyển đổi định dạng dữ liệu từ số sang ảnh cũng được chứng minh là một hướng tiếp cận hiệu quả, đặc biệt khi áp dụng cho các mô hình CNN.

Tuy nhiên, hệ thống vẫn còn một số hạn chế, như thời gian huấn luyện lâu với các mô hình phức tạp, khả năng tổng quát hóa với dữ liệu thực tế chưa được kiểm chứng sâu rộng, và chưa tích hợp khả năng phát hiện theo thời gian thực.

## Hạn chế & Hướng phát triển.

* + 1. Hạn chế.

Trước hết, quá trình đánh giá mô hình chủ yếu dựa vào **ma trận nhầm lẫn** và **chỉ số độ chính xác (accuracy)**, trong khi các chỉ số bổ sung như **Precision**, **Recall**, **F1-score** và **ROC-AUC** lại chưa được phân tích đầy đủ. Điều này có thể gây ra cái nhìn chưa toàn diện, đặc biệt trong các tình huống dữ liệu mất cân bằng, khi một mô hình có thể đạt độ chính xác cao nhưng lại không phát hiện tốt các lớp tấn công hiếm gặp.

Thứ hai, **thời gian huấn luyện** của các mô hình, đặc biệt là với các mạng sâu hoặc có cấu trúc tuần tự như LSTM và RNN, khá lâu do giới hạn về cấu hình phần cứng. Việc thiếu GPU chuyên dụng khiến quá trình huấn luyện trở nên kéo dài, gây ảnh hưởng đến khả năng tinh chỉnh siêu tham số và thử nghiệm nhiều mô hình trong thời gian ngắn.

Ngoài ra, việc tiền xử lý dữ liệu cũng gặp nhiều khó khăn do dung lượng tập dữ liệu lớn và cấu hình máy hạn chế, dẫn đến hiện tượng tiêu tốn nhiều tài nguyên bộ nhớ khi chuẩn hóa, cân bằng và chuyển đổi dữ liệu sang định dạng ảnh. Mô hình hiện tại cũng chưa được kiểm nghiệm trong môi trường thực tế, nơi yêu cầu xử lý thời gian thực và khả năng phản ứng nhanh với các mối đe doạ mới.

* + 1. Hướng phát triển.

Trong tương lai, hệ thống phát hiện xâm nhập có thể được mở rộng và cải tiến theo nhiều hướng nhằm nâng cao hiệu quả và khả năng ứng dụng trong thực tế. Trước hết, cần bổ sung các **chỉ số đánh giá đa dạng hơn** như Precision, Recall, F1-score và ROC-AUC để cung cấp cái nhìn toàn diện về hiệu suất mô hình, đặc biệt trong bối cảnh dữ liệu mất cân bằng. Điều này sẽ giúp đánh giá chính xác hơn khả năng phát hiện tấn công và hạn chế việc mô hình chỉ tối ưu cho độ chính xác tổng thể.

Bên cạnh đó, một hướng phát triển quan trọng là cải tiến khả năng **nhận dạng chi tiết từng loại tấn công** thay vì chỉ phân loại giữa “bình thường” và “tấn công”. Điều này sẽ giúp hệ thống không chỉ phát hiện mà còn **phân tích được loại hình tấn công cụ thể** (ví dụ như DDoS, Brute Force, SQL Injection,...), từ đó hỗ trợ người quản trị mạng đưa ra biện pháp ứng phó thích hợp trong thời gian ngắn nhất.

Cuối cùng, hệ thống nên được **triển khai kiểm thử trong môi trường ứng dụng thực tế**, điển hình là các hệ thống web, nơi thường xuyên phải đối mặt với các mối đe doạ bảo mật. Việc kiểm thử trong môi trường thực tế sẽ giúp đánh giá khả năng phản ứng thời gian thực của mô hình, cũng như phát hiện và xử lý các tình huống không mong muốn trong hoạt động mạng thực tế.

# Tài liệu tham khảo

1. Tìm hiểu về các sử dụng googlecolab

<https://colab.research.google.com/>

1. Tìm hiểu về các mô hình học sâu

<https://viblo.asia/p/recurrent-neural-networkphan-1-tong-quan-va-ung-dung-jvElaB4m5kw>

<https://whitehat.vn/threads/cac-mo-hinh-hoc-sau-deep-learning-su-dung-cho-phat-hien-xam-nhap-mang.16557/>

1. Tìm hiểu về dataset CSE-CIC-IDS2018

<https://www.unb.ca/cic/datasets/ids-2018.html>

<https://www.kaggle.com/datasets/dhoogla/nfcsecicids2018v2>

1. Network Intrusion Detection System using Deep Learning Lirim Ashiku1 Cihan Dagli
2. Cải thiện khả năng phát hiện tấn công mạng bằng kỹ thuật học sâu

Tô Trọng Tín , Trần Văn Lăng

# Phụ lục

Tất cả file code dự án lưu tại:

<https://github.com/kntn88777-pixel/idsdeeplearning.git>

Dataset:

<https://www.kaggle.com/datasets/solarmainframe/ids-intrusion-csv>