TRƯỜNG ĐẠI HỌC CMC

KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

CMC UNIVERSITY
Aspire to Inspire the Digital World

BÀI TẬP LỚN

ỨNG DỤNG HỌC MÁY NHẬN DẠNG XÂM NHẬP MẠNG DOANH NGHIỆP

Học phần	**:**	**HỌC MÁY VÀ KHAI PHÁ DỮ LIỆU**
Giảng viên hướng dẫn	**:**	**PGS. NGUYỄN THANH TÙNG**
Nhóm sinh viên	**:**	**BÙI KHẮC KHÁNH – BIT220084**
		PHẠM QUỐC AN – BIT220006
		VŨ VĂN TIẾN – BIT220248
		NGUYỄN TRUNG DU – BIT220031
Ngành	**:**	**CÔNG NGHỆ THÔNG TIN**
Lớp	**:**	**22IT3**
Khóa	**:**	**1**

Hà Nội, tháng 11 năm 2024

LỜI CẢM ƠN

Nhóm em xin bày tỏ lòng biết ơn sâu sắc và chân thành nhất đến **PGS. Nguyễn Thanh Tùng**, giảng viên môn **Học máy và Khai phá dữ liệu**, người đã tận tình hướng dẫn và đồng hành cùng nhóm em trong suốt quá trình thực hiện đề tài "***Ứng dụng học máy nhận dạng xâm nhập mạng doanh nghiệp.***"

Trong suốt học kỳ, thầy không chỉ truyền đạt những kiến thức chuyên môn sâu sắc về học máy, an ninh mạng và phân tích dữ liệu mà còn chia sẻ nhiều kinh nghiệm thực tiễn quý giá trong lĩnh vực khai phá dữ liệu và xây dựng mô hình dự đoán. Nhờ sự hướng dẫn tận tâm của thầy, nhóm em đã từng bước nắm vững các khái niệm phức tạp, từ việc tiền xử lý dữ liệu, chọn lựa thuật toán phù hợp, xây dựng mô hình đến đánh giá hiệu quả và triển khai giải pháp.

Đặc biệt, thầy luôn kiên nhẫn xem xét và góp ý chi tiết cho từng giai đoạn nghiên cứu của nhóm. Những nhận xét sắc sảo và gợi ý mang tính định hướng của thầy đã giúp nhóm em kịp thời nhận ra các vấn đề kỹ thuật, tối ưu hóa thuật toán và cải thiện khả năng nhận dạng xâm nhập của hệ thống. Qua đó, nhóm em không chỉ nâng cao khả năng áp dụng lý thuyết vào thực tiễn mà còn rèn luyện được kỹ năng phân tích và giải quyết vấn đề, những yếu tố vô cùng quan trọng cho sự nghiệp trong tương lai.

Không chỉ là người truyền đạt kiến thức, thầy còn giúp nhóm em phát triển tư duy logic, kỹ năng làm việc nhóm và phong thái làm việc chuyên nghiệp. Những bài học này không chỉ giúp nhóm hoàn thành đề tài một cách tốt nhất mà còn là hành trang quý giá trong quá trình học tập và phát triển sau này.

Nhóm em xin gửi lời cảm ơn chân thành đến thầy vì sự tận tâm, nhiệt huyết và những đóng góp to lớn cho sự thành công của đề tài. Kính chúc thầy dồi dào sức khỏe, hạnh phúc và ngày càng thành công trong sự nghiệp giáo dục.

MỤC LỤC

BẢNG CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT

STT	Từ viết tắt	Diễn giải
1	Activation Function	Hàm kích hoạt, được sử dụng trong các mạng nơ-ron để xác định đầu ra của một nơ-ron dựa trên trọng số và đầu vào. Các hàm phổ biến gồm ReLU, sigmoid và tanh.
2	Adam (Optimizer)	Một thuật toán tối ưu hóa dựa trên tính toán trung bình trọng số và điều chỉnh học tốc, thường dùng trong học sâu để tăng hiệu quả huấn luyện
3	Backpropagation	Thuật toán lan truyền ngược, được sử dụng để cập nhật trọng số của mạng nơ-ron bằng cách lan truyền lỗi từ đầu ra về đầu vào.
4	Batch Size	Số lượng mẫu dữ liệu được xử lý trong một lần cập nhật trọng số trong quá trình huấn luyện mô hình.
5	Deep Learning	Một lĩnh vực của học máy sử dụng các mạng nơ-ron sâu với nhiều lớp ẩn để học các biểu diễn phức tạp từ dữ liệu.
6	DoS (Denial of Service)	Một loại tấn công mạng nhằm làm gián đoạn dịch vụ bằng cách làm ngập mạng hoặc hệ thống với lưu lượng truy cập bất hợp pháp.
7	Dropout	Kỹ thuật regularization trong học sâu, trong đó một tỷ lệ nơ-ron ngẫu nhiên được "tắt" trong quá trình huấn luyện để giảm overfitting.
8	Epoch	Một vòng lặp qua toàn bộ tập dữ liệu trong quá trình huấn luyện mô hình.
9	GaussianNB	Gaussian Naive Bayes, một thuật toán phân loại dựa trên lý thuyết Bayes và giả định dữ liệu tuân theo phân phối Gaussian.

10	IDS (Intrusion Detection System)	Hệ thống phát hiện xâm nhập, dùng để giám sát mạng hoặc hệ thống nhằm phát hiện các hoạt động đáng ngờ hoặc vi phạm chính sách bảo mật.
11	KDD-CUP 1999	Một tập dữ liệu phổ biến được sử dụng trong nghiên cứu phát hiện xâm nhập, bao gồm nhiều loại tấn công mạng và dữ liệu thông thường.
12	Kernel RBF (Radial Basis Function)	Hàm nhân RBF, một loại kernel được sử dụng trong thuật toán SVM để ánh xạ dữ liệu không tuyến tính vào không gian đặc trưng cao hơn.
13	Loss Function	Hàm mất mát, được sử dụng để đánh giá sự khác biệt giữa dự đoán của mô hình và giá trị thực tế, từ đó điều chỉnh mô hình để giảm lỗi.
14	MLPClassifier (Multi-Layer Perceptron Classifier)	Một thuật toán học máy dựa trên mạng nơ-ron nhiều lớp (Multi-Layer Perceptron), được sử dụng cho các bài toán phân loại.
15	Margin	Trong SVM, đây là khoảng cách giữa siêu phẳng phân tách và các điểm dữ liệu gần nhất, cần được tối đa hóa để tăng hiệu quả phân loại.
16	NB (Naive Bayes)	Thuật toán phân loại dựa trên lý thuyết xác suất Bayes với giả định độc lập mạnh mẽ giữa các đặc trưng.
17	NSL-KDD	Một phiên bản cải tiến của tập dữ liệu KDD-CUP 1999, khắc phục các lỗi dư thừa và phân phối không cân bằng trong tập dữ liệu gốc.
18	Overfitting	Hiện tượng mô hình hoạt động rất tốt trên tập huấn luyện nhưng kém trên tập kiểm tra do học quá kỹ các đặc điểm không quan trọng hoặc nhiễu trong dữ liệu.

19	Probe	Một loại tấn công mạng nhằm thu thập thông tin về mạng hoặc hệ thống để chuẩn bị cho các tấn công khác.
20	R2L (Remote-to-Local)	Một loại tấn công mạng mà kẻ tấn công từ xa cố gắng đạt được quyền truy cập cục bộ vào máy mục tiêu.
21	ReLU (Rectified Linear Unit)	Một hàm kích hoạt phổ biến trong học sâu, cho giá trị đầu ra là 0 nếu đầu vào âm và giá trị đầu vào nếu đầu vào dương.
22	Regularization	Kỹ thuật giảm overfitting bằng cách thêm một hình phạt vào hàm mất mát để ngăn mô hình trở nên quá phức tạp.
23	ROC (Receiver Operating Characteristic)	Đường cong ROC là biểu đồ đánh giá hiệu quả của một mô hình phân loại bằng cách biểu diễn mối quan hệ giữa tỷ lệ dương tính thật (TPR) và tỷ lệ dương tính giả (FPR).
24	SGD (Stochastic Gradient Descent)	Một thuật toán tối ưu hóa phổ biến trong học máy, thực hiện cập nhật trọng số cho từng mẫu hoặc từng batch nhỏ thay vì toàn bộ tập dữ liệu.
25	Sigmoid	Một hàm kích hoạt cho đầu ra giá trị trong khoảng (0,1), thường được sử dụng trong các bài toán phân loại nhị phân.
26	SVM (Support Vector Machine)	Một thuật toán học máy được sử dụng để phân loại và hồi quy bằng cách tìm siêu phẳng tối ưu phân tách dữ liệu trong không gian đặc trưng.
27	Tanh (Hyperbolic Tangent)	Một hàm kích hoạt cho đầu ra giá trị trong khoảng (-1,1), thường được sử dụng trong các mạng nơ-ron để xử lý dữ liệu có giá trị âm.

28	U2R (User-to-Root)	Một loại tấn công mạng mà kẻ tấn công cố gắng nâng quyền hạn từ người dùng thông thường lên quyền root hoặc quản trị viên.

DANH MỤC CÁC BẢNG

DANH MỤC CÁC HÌNH ẢNH

MỞ ĐẦU

Trí tuệ nhân tạo đang là một trong những lĩnh vực được quan tâm mạnh mẽ trong nhiều ngành công nghiệp khác nhau. Trong lĩnh vực An ninh mạng, một trong những thách thức lớn mà các doanh nghiệp phải đối mặt là phát hiện và ngăn chặn các cuộc tấn công mạng, từ các cuộc tấn công DDoS đến các cuộc tấn công tinh vi như xâm nhập vào hệ thống, trộm cắp dữ liệu.

Để đối phó với những mối đe dọa này, các hệ thống IDS (Intrusion Detection System) giúp phát hiện sớm các hành vi xâm nhập vào hệ thống mạng phát hiện những hành vi này trước khi chúng gây ra hậu quả nghiêm trọng.

Nội dung chính của báo cáo gồm các mục sau đây:

- **Phần 1**: Tổng quan về vấn đề: Giới thiệu về mục tiêu của hệ thống IDS trong bảo vệ mạng doanh nghiệp, ngữ cảnh ứng dụng của IDS trong thực tế. tổng quan về tập dữ liệu, bao gồm các dạng tấn công phổ biến và các nguồn tài liệu tham khảo.
- **Phần 2**: Phương pháp học máy và mô hình đề xuất: Giới thiệu về các phương pháp học máy phổ biến được sử dụng trong phát hiện xâm nhập, bao gồm các thuật toán như Support Vector Machine (SVM) , Naive Bayes , Mạng Nơ-ron Nhân tạo (Artificial Neural Network - ANN) đề xuất sử dụng một mô hình cụ thể, có thể là một biến thể của một thuật toán học máy hiệu quả, để giải quyết bài toán phân loại và phát hiện xâm nhập mạng.
- **Phần 3:** Thực nghiệm và kết quả: Thực hiện thử nghiệm với các tập dữ liệu mô phỏng hoặc thực tế để đánh giá hiệu suất của hệ thống IDS, Phân tích kết quả, đưa ra nhận xét và đánh giá về hiệu suất của mô hình đề xuất so với các phương pháp khác.

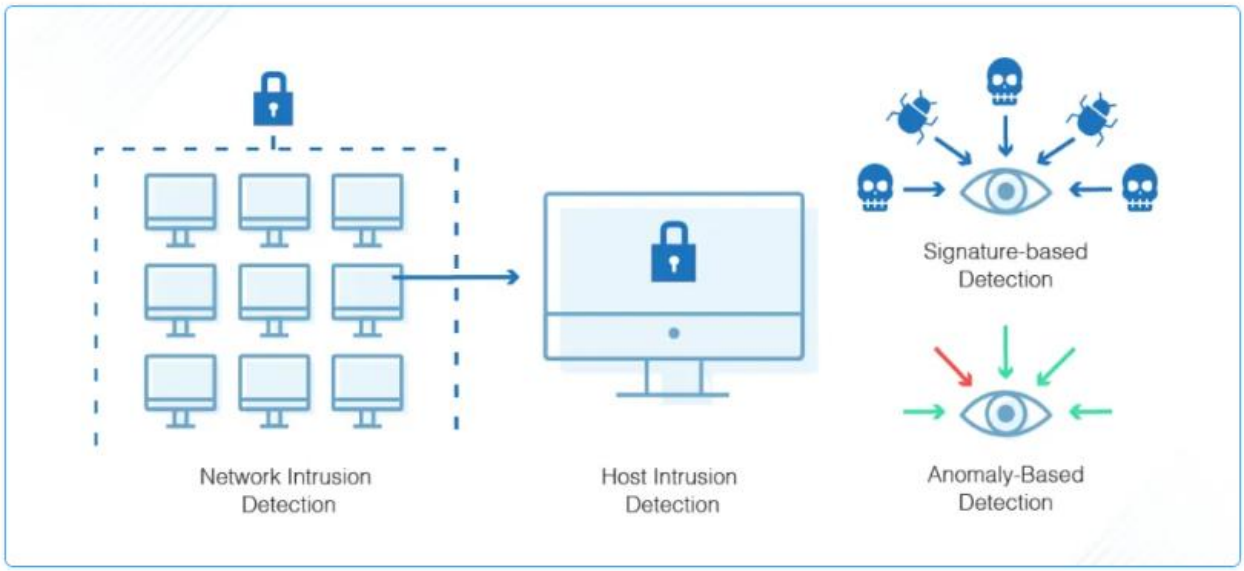
Báo cáo sẽ kết thúc bằng một phần kết luận, trong đó sẽ đánh giá các công việc đã thực hiện được, tổng kết về kết quả đạt được, và đề xuất một số hướng tìm hiểu và nghiên cứu tiếp theo trong lĩnh vực phát hiện và phân loại xâm nhập mạng doanh nghiệp.

PHẦN I. GIỚI THIỆU TỔNG QUAN

1. Giới thiệu bài toán

Trong môi trường kinh doanh ngày nay, việc bảo vệ mạng máy tính của doanh nghiệp trở nên cực kỳ quan trọng để đảm bảo an toàn thông tin và dữ liệu của khách hàng. Khía cạnh quan trọng trong việc này là Hệ thống Phát hiện Xâm nhập (IDS) công nghệ được thiết kế để phát hiện các hoạt động xâm nhập mạng không mong muốn.

IDS, được xem như là "mắt" của hệ thống mạng, chịu trách nhiệm theo dõi và phát hiện các mẫu hoạt động không bình thường hoặc có hại trong dữ liệu mạng. Một IDS hiệu quả sẽ sử dụng các phương pháp phân tích dữ liệu mạng để nhận biết các hành vi xâm nhập, bao gồm phân tích gói tin, dòng lưu lượng, và hành vi người dùng.



Hình 1. Hệ thống IDS

❖ **Các tính năng quan trọng nhất của IDS bao gồm:**

- **Về giám sát traffic mạng và các hoạt động khả nghi:** IDS theo dõi và phân tích lưu lượng mạng, xác định các hoạt động không bình thường và có thể là mối đe dọa đến hệ thống mạng. Bằng trí tuệ nhân tạo có thể học từ dữ liệu lưu lượng mạng lịch sử để nhận diện các dấu hiệu tiềm ẩn của tấn công mạng, bao gồm các cuộc tấn công phân tán, tấn công từ chối dịch vụ (DoS), và phần mềm độc hại.

- **Về học từ kinh nghiệm và cải thiện liên tục**: Quan trọng nhất, hệ thống IDS được cải thiện bằng trí tuệ nhân tạo có thể học từ kinh nghiệm và cải thiện liên tục qua thời gian. Bằng cách thu thập dữ liệu về các mối đe dọa mạng và kết quả của các biện pháp phòng chống, AI có thể cung cấp các thông tin quý báu để nâng cao hiệu suất và hiệu quả của hệ thống.

Sự phát triển của trí tuệ nhân tạo, đặc biệt là các phương pháp học máy và học sâu, đã tạo ra cơ hội mới để cải thiện hiệu suất của IDS. Trí tuệ nhân tạo có thể được sử dụng để phát hiện các mẫu hoạt động xâm nhập mạng phức tạp hơn và tinh vi hơn, đồng thời cung cấp các giải pháp ngăn chặn tự động và linh hoạt hơn.

Với các phân tích ở trên, nhóm thực hiện lựa chọn chủ đề Nghiên cứu về học máy cho xử lý phát hiện và phân loại xâm nhập mạng ứng dụng trong doanh nghiệp .

2. Bộ dữ liệu sử dụng

Trong bài toán này, nhóm thực hiện sử dụng các tập dữ liệu NSL-KDD được rút gọn từ KDD Cup 1999:

Kích thước: Tập dữ liệu gốc KDD Cup 1999 chứa khoảng 4.9 triệu mẫu. Tuy nhiên, trong dự án này, sử dụng phiên bản thu nhỏ của tập dữ liệu, tập dữ liệu có sẵn tại https://kdd.ics.uci.edu/databases/kddcup99/kddcup.data_5_percent.gz, chỉ chứa khoảng 0.2 triệu mẫu.

Nguồn gốc: Tập dữ liệu KDD Cup 1999 được tạo ra để phát triển và kiểm thử các phương pháp phát hiện xâm nhập mạng. Dữ liệu được thu thập từ hệ thống mạng thực tế và chứa các giao thức mạng khác nhau.

Nhận xét về tập dữ liệu: Tập dữ liệu KDD Cup 1999 có kích thước lớn và đa dạng, với nhiều loại các cuộc tấn công và giao thức mạng khác nhau.

3. Kết luận

Trong quá trình nghiên cứu và sử dụng các tập dữ liệu từ NSL-KDD, ta nhận thấy rằng các tập dữ liệu này rất hữu ích để nghiên cứu về phát hiện xâm nhập mạng. Tuy nhiên, cần phải lưu ý rằng các tập dữ liệu này không phản ánh hoàn toàn các tình huống mạng

thực tế hiện nay do thời gian thu thập và sự phát triển của công nghệ mạng. Mặc dù vậy, chúng vẫn cung cấp một bức tranh tổng quan về các loại tấn công và hoạt động đáng ngờ trong môi trường mạng.

PHẦN II. PHƯƠNG PHÁP THỰC HIỆN

1. Tổng quan

Hiện nay học máy là một trong những công cụ phổ biến giải quyết bài toán phân loại dữ liệu. Trong học máy có nhiều loại mô hình học máy khác nhau , tuy nhiên nhóm thực hiện lựa chọn mô hình học có giám sát SVM (Suport Vector Machine), Navie Bayes , Mạng Nơ-ron Nhân tạo (Artificial Neural Network - ANN) cho bài toán của mình.

1.1 Giới thiệu các kiến trúc mô hình

SVM (Suport Vector Machine) là một thuật toán học máy phổ biến được sử dụng cho cả bài toán phân loại và hồi quy. Ý tưởng cơ bản của SVM là tìm ra một siêu mặt phẳng (hyperplane) tốt nhất để phân chia dữ liệu thành các lớp khác nhau. SVM cố gắng tối đa hóa khoảng cách giữa các điểm dữ liệu gần siêu mặt phẳng (các điểm gọi là support vectors) từ các lớp khác nhau.

Naive Bayes là một mô hình phân loại dựa trên nguyên tắc của định lý Bayes. Mặc dù đơn giản, nhưng Naive Bayes thường cho kết quả tốt đối với các bài toán phân loại văn bản, email spam, và nhiều lĩnh vực khác. Thuật toán Naive Bayes giả định rằng các đặc trưng là độc lập với nhau, mặc dù trong thực tế không phải lúc nào cũng đúng.

Học Sâu (Deep Learning) với mô hình Mạng Nơ-ron Nhân tạo (Artificial Neural Network - ANN) để phát hiện và phân loại các cuộc tấn công mạng dựa trên bộ dữ liệu NSL-KDD. Bộ dữ liệu này được coi là tiêu chuẩn trong nghiên cứu an ninh mạng, chứa các thông tin chi tiết về nhiều loại tấn công khác nhau.

1.2 Quy trình huấn luyện

- **Quá trình huấn luyện SVM** bao gồm việc tìm ra siêu mặt phẳng tối ưu phân chia các lớp dữ liệu bằng cách sử dụng các điểm dữ liệu hỗ trợ (support vectors).
- **Quá trình huấn luyện Naive Bayes** đơn giản hóa là tính toán xác suất có điều kiện cho mỗi lớp dựa trên các đặc trưng của dữ liệu huấn luyện.

- **Quá trình huấn luyện ANN (Artificial Neural Network)** bao gồm việc tối ưu hóa các trọng số của mạng bằng cách sử dụng lan truyền ngược lỗi (backpropagation) và thuật toán tối ưu hóa (như SGD, Adam). Mục tiêu là giảm thiểu hàm mất mát (loss function) để mạng có thể dự đoán chính xác nhất có thể trên tập dữ liệu huấn luyện.

1.3 Các tham số

SVM có nhiều tham số quan trọng như kernel, độ rộng của vùng hỗ trợ (margin), và siêu tham số C (điều chỉnh độ quan trọng của việc phân loại sai sót).

Naive Bayes thường không có nhiều tham số để điều chỉnh. Tuy nhiên, trong một số trường hợp, bạn có thể cần điều chỉnh các tham số smoothing như alpha.

ANN có nhiều tham số quan trọng cần điều chỉnh, bao gồm: Số lớp và số lượng nơ-ron trong mỗi lớp (mô hình hóa độ phức tạp). Hàm kích hoạt (activation function) như ReLU, sigmoid, hoặc tanh. Tỷ lệ học (learning rate) và thuật toán tối ưu hóa (như SGD, Adam). Số epoch và batch size (kiểm soát quá trình huấn luyện). Dropout rate (để giảm overfitting) và các tham số regularization khác như L1/L2. Việc tinh chỉnh các tham số này có ảnh hưởng lớn đến hiệu suất của mạng.

2. Tiền xử lý dữ liệu

Tiền xử lý dữ liệu là bước chuẩn bị dữ liệu trước khi đưa vào mô hình, nhằm cải thiện chất lượng và hiệu quả phân tích.

- **Xử lý dữ liệu thiếu**: Điền giá trị hoặc loại bỏ dữ liệu thiếu.
- **Làm sạch dữ liệu**: Xóa trùng lặp, xử lý giá trị bất thường.
- **Mã hóa dữ liệu**: Chuyển dữ liệu phân loại thành dạng số.
- **Chuẩn hóa/Chuẩn hóa**: Đưa dữ liệu về cùng phạm vi hoặc phân phối chuẩn.
- **Tách dữ liệu**: Chia thành tập huấn luyện, kiểm tra, xác nhận.

```
Loading datasets...
Training set shape: (125973, 42)
Test set shape: (22544, 42)

Mapping attack types to categories...
Preprocessing data...
Warning: Found unmapped attack types in test set: ['snmpgetattack' 'httptunnel' 'ps' 'snmpguess' 'named' 'sendmail' 'worm'
 'xlock' 'xsnoop']

Encoding categorical features...
Processing numeric features...
Feature matrix shape - Training: (125973, 122), Test: (22544, 122)

Processing labels...
Unique classes found: ['DoS' 'Probe' 'R2L' 'U2R' 'normal']

Cleaning test data...
Final test set shape: (22544, 122)

Scaling features...

Splitting training data...
Training set size: 100778
Validation set size: 25195

Initializing models...
SVM model initialized
Naive Bayes model initialized
Neural Network model initialized
```

Hình 2. Các bước tiền xử lý dữ liệu

- **Tải dữ liệu:**
  - Tập huấn luyện có 125.973 mẫu với 42 đặc trưng.
  - Tập kiểm tra có 22.544 mẫu với 42 đặc trưng.
- **Ánh xạ các loại tấn công vào danh mục:**
  - Có các loại tấn công chưa được ánh xạ trong tập kiểm tra, bao gồm: snmpgetattack, httptunnel, ps, snmpguess, named, sendmail, worm, xlock, and xsnoop.
- **Mã hóa các đặc trưng phân loại:**
  - Ma trận đặc trưng cho cả tập huấn luyện và tập kiểm tra có 122 đặc trưng.
- **Xử lý nhãn:**
  - Các lớp duy nhất được xác định là: DoS, Probe, R2L, U2R, and normal.
- **Làm sạch dữ liệu kiểm tra:**

- Hình dạng cuối cùng của tập kiểm tra là 22.544 mẫu với 122 đặc trưng.

➢ **Chuẩn hóa đặc trưng**

➢ **Chia tập dữ liệu huấn luyện:**

- Kích thước tập huấn luyện được điều chỉnh còn 100.778 mẫu.
- Kích thước tập xác thực là 25.195 mẫu.

➢ **Khởi tạo mô hình:**

- Các mô hình bao gồm SVM, Naive Bayes, và Mạng Nơ-ron đang được khởi tạo để thực hiện nhiệm vụ phát hiện xâm nhập.

Quy trình này nhằm chuẩn bị dữ liệu và thực hiện tiền xử lý cho các mô hình học máy cho huấn luyện.

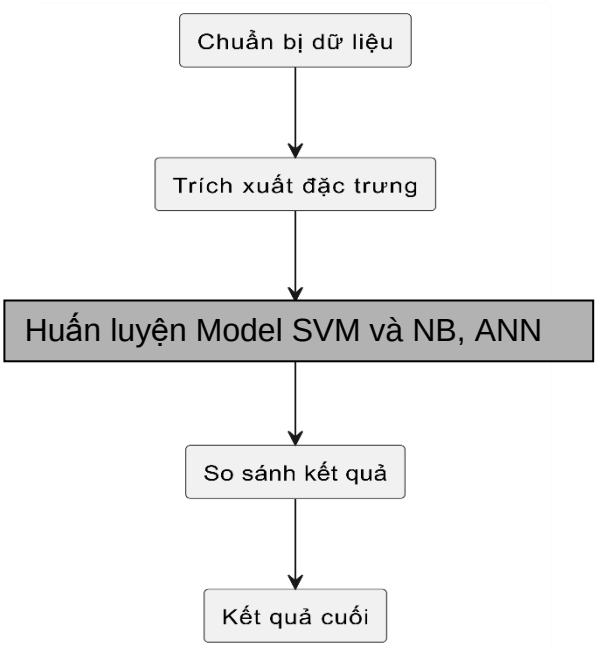
```
==================================================
DATASET ANALYSIS
==================================================

Total samples: 125973
Total features: 122
Numeric features: 38
Categorical features: 3
Total classes: 5

Class Distribution:
Class 4: 67343 samples (53.46%)
Class 0: 45927 samples (36.46%)
Class 1: 11656 samples (9.25%)
Class 2: 995 samples (0.79%)
Class 3: 52 samples (0.04%)
```

Hình 3. Phân tích chi tiết tập dữ liệu

Hình ảnh này cung cấp tổng quan về số lượng mẫu, đặc trưng, và phân phối lớp trong một tập dữ liệu trước khi tiến hành tiền xử lý và huấn luyện mô hình học máy.

3. Mô hình học để xử lý bài toán

Trong đề tài này, nhóm sẽ lựa chọn kiến trúc mô hình học trực quan như sau:



Hình 4. Mô hình học máy SVM và Naive Bayes, ANN trực quan

Chuẩn bị dữ liệu tải dữ liệu từ nguồn cung cấp thu thập dữ liệu từ các nguồn đáng tin cậy hoặc các tập dữ liệu công cộng về mạng máy tính và các loại tấn công mạng. Loại bỏ hoặc biến đổi các đặc trưng không cần thiết, xử lý các giá trị thiếu, và mã hóa các đặc trưng dạng văn bản thành dạng số. Phân chia dữ liệu thành tập huấn luyện và tập kiểm tra để đảm bảo tính công bằng và độc lập trong việc đánh giá mô hình.

```
print('Kích thước của tập huấn luyện:', df.shape)
print('Kích thước của tập kiểm tra:', df_test.shape)
```

```
df.head(5)
```

```
df_test.head(5)
```

```
# Loại bỏ dấu chấm cuối từ trong cột 'label' của tập huấn luyện
df['label'] = df['label'].str.rstrip('.')

print('Phân phối nhãn trong tập huấn luyện:')
print(df['label'].value_counts())
print()
print('Phân phối nhãn trong tập kiểm tra:')
print(df_test['label'].value_counts())
```

Hình 5. Code trích xuất dữ liệu

Trích xuất đặc trưng mã hóa đặc trưng áp dụng các phương pháp mã hóa one-hot hoặc label encoding cho các đặc trưng dạng văn bản. Chuẩn hóa dữ liệu số để đảm bảo tỷ lệ và phân phối đồng đều giữa các đặc trưng.

Bước 2: Chuẩn hóa đặc trưng

```
[ ]  1  # Chia các khung dữ liệu thành X & Y
     2  # Thuộc tính X, biến kết quả Y
     3  X_Df = newdf.drop('label', axis=1)
     4  Y_Df = newdf.label
     5
     6  # test set
     7  X_Df_test = newdf_test.drop('label', axis=1)
     8  Y_Df_test = newdf_test.label
```

Hình 6. Code chuẩn hóa đặc trưng

```
# Initialize models with optimized hyperparameters
svm_model = SVC(kernel='rbf', C=1, gamma='scale', probability=True, random_state=42)
nb_model = GaussianNB()
ann_model = MLPClassifier(
    hidden_layer_sizes=(512, 256, 128, 64, 32, 16),
    max_iter=1000,
    activation='relu',
    solver='adam',
    random_state=42,
    learning_rate='adaptive',
    early_stopping=True,
    validation_fraction=0.1
)
```

Hình 7.Khởi tạo và thiết lập tham số cho 3 mô hình học máy

Khởi tạo các mô hình máy học với tham số khác nhau để tinh chỉnh phù hợp.

- **Mô hình SVC** sử dụng kernel RBF, phù hợp để xử lý dữ liệu phi tuyến.
- **Mô hình GaussianNB** là bộ phân loại dựa trên xác suất.
- **Mô hình ANN - MLPClassifier** là một mạng nơ-ron với nhiều lớp ẩn.

```python
# Evaluate each model
print("\nEvaluating SVM model...")
svm_cv_mean, svm_cv_std = evaluate_model(svm_model, X_train, X_val, X_test_scaled, y_train, y_val, y_test, "SVM")

print("\nEvaluating Naive Bayes model...")
nb_cv_mean, nb_cv_std = evaluate_model(nb_model, X_train, X_val, X_test_scaled, y_train, y_val, y_test, "Naive Bayes")

print("\nEvaluating Neural Network model...")
ann_cv_mean, ann_cv_std = evaluate_model(ann_model, X_train, X_val, X_test_scaled, y_train, y_val, y_test, "Neural Network")
```

Hình 8. Đánh giá 3 mô hình học máy

So sánh thời gian và hiệu suất: So sánh thời gian huấn luyện và kiểm tra của các mô hình SVM, Naive Bayes và ANN. Đánh giá điểm số độ chính xác tổng quát, độ chính xác thực tế và tỷ lệ nhớ lại của mỗi mô hình để quyết định mô hình nào phù hợp hơn cho bài toán cụ thể. sau khi đánh giá và tinh chỉnh, mô hình có thể được triển khai để phát hiện và phân loại các tấn công mạng trong môi trường thực tế.

4. Kết luận

Trong phần này, nhóm đã trình bày quy trình và các mô hình học máy mà sẽ sử dụng để giải quyết bài toán phân loại dữ liệu. SVM và Naive Bayes, ANN là ba mô hình mạnh mẽ và phổ biến trong lĩnh vực học máy và hứa hẹn mang lại kết quả tốt cho bài toán trên. Nhóm sẽ tiến hành thử nghiệm và đánh giá hiệu suất của các mô hình này trong phần tiếp theo.

PHẦN III. THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

1. Tổng quan

Để thực hiện thử nghiệm và đánh giá kết quả, nhóm sử dụng số thông tin về cấu hình máy, nền tảng lập trình, thư viện và phương pháp đánh giá kết quả như sau:

Cấu hình máy:

- Hệ điều hành: Windows 10 pro.
- Bộ vi xử lý (CPU): Intel Core i5-6300U.
- Bộ nhớ RAM: 16GB DDR4.
- Bộ xử lý đồ họa (GPU) : Không có.

Nền tảng lập trình: Python 3.11. Thư viện và framework: Scikit-learn (v0.24.1), TensorFlow (v2.5.0), Pandas (v1.2.4), NumPy (v1.20.3). Mã nguồn được tổ chức thành các tệp và thư mục riêng biệt cho từng phần của quy trình như: thu thập dữ liệu, tiền xử lý, huấn luyện mô hình và đánh giá.

Phương pháp đánh giá kết quả: F1-measure được tính bằng trung bình điều hòa của độ chính xác và độ nhớ lại. Đây là một phép đo tổng hợp của độ chính xác và độ nhớ lại của mô hình. Độ chính xác (Accuracy) tỉ lệ giữa số lượng dự đoán đúng và tổng số mẫu trong tập dữ liệu kiểm tra. Precision và Recall: Precision đo lường tỉ lệ các dự đoán dương tính mà thực sự là dương tính, trong khi Recall đo lường tỉ lệ các dự đoán dương tính so với tất cả các trường hợp dương tính trong tập dữ liệu.

Thông tin này giúp định rõ hơn về môi trường và điều kiện mà thử nghiệm được thực hiện, từ đó tạo ra kết quả đánh giá có tính khả thi và thực tế.

2. Triển khai thực nghiệm

Triển khai thử nghiệm hệ thống theo các kịch bản sau đây:

STT	Tân tập dữ liệu	Số lượng training	Số lượng validation	Số lượng test

1	NSL-KDD, khoảng 0.2 triệu mẫu.	125973	(125973, 122)	22544

Bảng 1. Bảng triển khai hệ thống theo kịch bản

Bảng phân loại:

Dự đoán / **Thực tế**	**Dương tính**	**Âm tính**
Dương tính	TP	FP
Âm tính	FN	TN

Bảng 2. Bảng phân loại dự đoán

Từ ma trận cơ bản này, ta có một số thuật ngữ như sau:

- Positive (P): Tổng số ca dương tính thực tế.
- Negative (N): Tổng số ca âm tính thực tế.
- True positive (TP): Số các ca dự đoán dương tính đúng hay dương tính thật.
- True negative (TN): Số các ca dự đoán âm tính đúng hay âm tính thật.
- False positive (FP): Số các ca dự đoán dương tính sai hay dương tính giả.
- False negative (FN): Số các ca dự đoán âm tính sai hay âm tính giả.

Với các thuật ngữ trên, ta có các chỉ số đánh giá sau:

- Độ chính xác (Accuracy):
  - Acc = (TP+TN)/(P+N) = (TP+TN)/(TP+FP+TN+FN)
- Tỷ lệ nhớ lại (Recall):
  - Recall = TP/P = TP/(TP+FN)
- Dự đoán tích cực (Precision):

- Precision = (TP)/(TP+FP)

– Điểm F1 là một trung bình hài hòa Precision và Recall (F_meansure):

- F1 = (2*Presision*Recall)/(Precision+Recall)

Áp dụng các công thức tính toán trên nhóm đã thu được các kết quả thực nghiệm sau:

STT	**Tân tập dữ liệu**	**Số lượng training**	**Số lượng validation**	**Số lượng test**	**Tham số tương ứng (hàm kích hoạt thay đổi,…)**	**Kết quả**		
						Độ chính xác	**Tỷ lệ nhớ lại**	**F-measure**
1	NSL-KDD, 0.2 triệu mẫu.	125973	(125973, 122)	22544	SVM:			
					RBF	0.96981	0.98403	0.97537
					Naive Bayes:	0.66687	0.41814	0.58795
					ANN:	0.998	0.998	0.9977

Bảng 3. Bảng kết quả thực nghiệm

2.1 Mô hình SVM:

```
Evaluating SVM model...

Cross-validation scores - SVM:
Mean: 0.992 (+/- 0.001)

Confusion Matrix - SVM
[[6274   26    0    0 1158]
 [ 349 1706    3    0  363]
 [   0   10  271    0 1918]
 [   0   12    2   12   26]
 [ 408  304    3    8 9691]]
```

Hình 9. Kết quả đánh giá mô hình SVM

Hình ảnh cho thấy kết quả đánh giá của một mô hình SVM (Support Vector Machine). Nó bao gồm các điểm số phân tách chéo và ma trận nhầm lẫn cho mô hình SVM.

Điểm số Phân tách Chéo (Cross-validation scores - SVM):

- **Mean (Trung bình)**: 0.992
- **Standard Deviation (Độ lệch chuẩn)**: ± 0.001

Điều này cho thấy mô hình SVM đạt độ chính xác trung bình cao (99.2%) khi áp dụng phân tách chéo, với biến động rất nhỏ (± 0.001).

Ma trận Nhầm lẫn (Confusion Matrix - SVM):

- **Lớp 0**: Dự đoán đúng 6274 lần, nhầm lẫn 1184 lần
- **Lớp 1**: Dự đoán đúng 1706 lần, nhầm lẫn 715 lần
- **Lớp 2**: Dự đoán đúng 271 lần, nhầm lẫn 1928 lần
- **Lớp 3**: Dự đoán đúng 12 lần, nhầm lẫn 38 lần
- **Lớp 4**: Dự đoán đúng 9691 lần, nhầm lẫn 723 lần

Ý nghĩa:

- **Điểm số Phân tách Chéo Cao**: Điểm số phân tách chéo cho thấy mô hình SVM có hiệu suất rất tốt và ổn định khi được huấn luyện trên các phần nhỏ khác nhau của dữ liệu.
- **Ma trận Nhầm lẫn**: Ma trận nhầm lẫn cho biết cách mà mô hình SVM phân loại dữ liệu. Số lượng dự đoán đúng cao cho thấy khả năng phân loại chính xác của mô hình. Tuy nhiên, các lỗi nhầm lẫn (sai số) cũng cần được xem xét để hiểu các trường hợp mà mô hình có thể gặp khó khăn trong việc phân loại đúng.

Classification Report - SVM

	precision	recall	f1-score	support
normal	0.89	0.84	0.87	7458
DoS	0.83	0.70	0.76	2421
R2L	0.97	0.12	0.22	2199
Probe	0.60	0.23	0.33	52
U2R	0.74	0.93	0.82	10414
accuracy			0.80	22544
macro avg	0.81	0.57	0.60	22544
weighted avg	0.82	0.80	0.77	22544

Hình 10. Báo cáo phân loại SVM

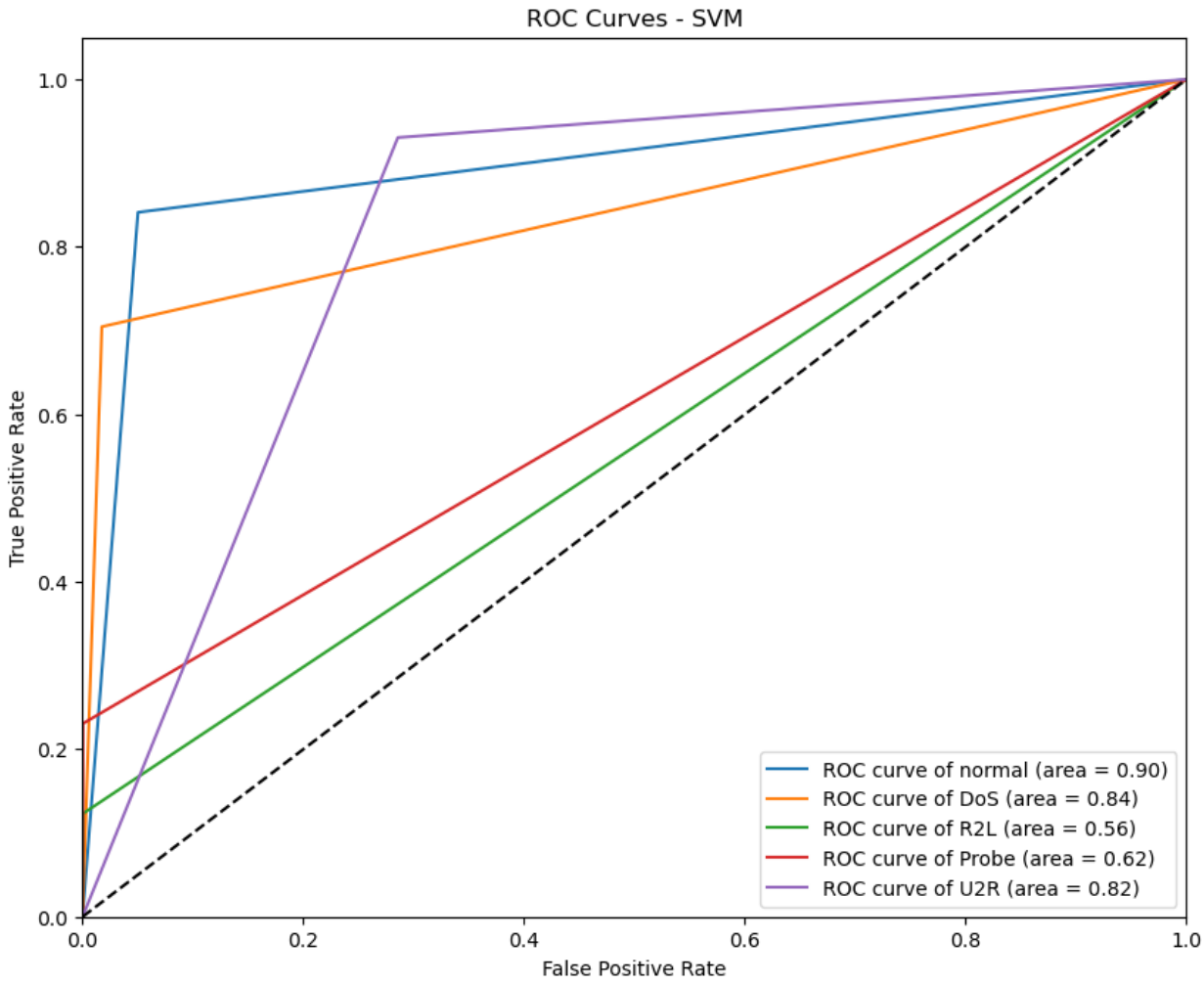
Hình ảnh hiển thị báo cáo phân loại (classification report) của mô hình SVM (Support Vector Machine). Báo cáo này bao gồm các chỉ số: độ chính xác (precision), tỷ lệ nhớ lại (recall), điểm số F1 (f1-score), và số lượng mẫu (support) cho các lớp khác nhau: normal, DoS, R2L, Probe, và U2R. Ngoài ra, báo cáo còn cung cấp độ chính xác tổng thể, trung bình theo từng hạng mục (macro average), và trung bình có trọng số (weighted average) cho các chỉ số này.

Ý nghĩa:

- **Độ chính xác (Precision)**: Tỷ lệ chính xác của mô hình khi dự đoán đúng các trường hợp dương tính. Cao nhất ở lớp R2L (0.97) và thấp nhất ở lớp Probe (0.60).
- **Tỷ lệ nhớ lại (Recall)**: Khả năng của mô hình trong việc phát hiện đúng các trường hợp dương tính. Tỷ lệ nhớ lại cao ở lớp U2R (0.93) và thấp nhất ở lớp R2L (0.12).
- **Điểm số F1 (F1-score)**: Là trung bình điều hòa của độ chính xác và tỷ lệ nhớ lại. Điểm F1 cao nhất ở lớp U2R (0.82) và thấp nhất ở lớp Probe (0.33).
- **Tổng mẫu (Support)**: Số lượng mẫu của từng lớp, với lớp U2R có nhiều mẫu nhất (10414) và lớp Probe ít nhất (52).

Tổng quan:

- Mô hình SVM đạt độ chính xác tổng thể là 0.80, cho thấy hiệu suất khá tốt trong phân loại dữ liệu tấn công mạng.
- Mô hình này có hiệu suất tốt nhất ở các lớp normal và U2R, nhưng gặp khó khăn với lớp Probe và R2L, đặc biệt là lớp R2L với tỷ lệ nhớ lại rất thấp.
- Trung bình có trọng số (weighted average) các chỉ số cho thấy mô hình có sự cân bằng giữa độ chính xác và tỷ lệ nhớ lại, nhưng vẫn còn cần cải thiện ở các lớp nhỏ.



Hình 11. Biểu đồ ROC của mô hình SVM

Ý nghĩa:

- **Hiệu suất cao cho lớp normal**: Giá trị AUC cao (0.90) cho thấy mô hình SVM có thể phân loại chính xác các mẫu thuộc lớp normal với tỷ lệ dương tính thật cao và tỷ lệ dương tính giả thấp.
- **Hiệu suất thấp cho lớp R2L**: Giá trị AUC thấp (0.56) cho thấy mô hình gặp khó khăn trong việc phân loại các mẫu thuộc lớp R2L, với tỷ lệ nhầm lẫn cao.
- **Cân bằng cho các lớp còn lại**: Các giá trị AUC của các lớp DoS (0.84), Probe (0.62) và U2R (0.82) chỉ ra rằng mô hình có khả năng phân loại tốt nhưng vẫn cần cải thiện thêm đối với các lớp phức tạp như Probe và R2L.

2.2 Mô hình Naive Bayes:

```
Evaluating Naive Bayes model...

Cross-validation scores - Naive Bayes:
Mean: 0.727 (+/- 0.152)

Confusion Matrix - Naive Bayes
[[2937    4 1172   29 3316]
 [ 348  195  338  131 1409]
 [   0    0  941  500  758]
 [   0    0   15   33    4]
 [  61   16 3137  326 6874]]
```

Hình 12. Kết quả đánh giá mô hình NB

Hình ảnh cho thấy kết quả đánh giá của mô hình Naive Bayes. Nó bao gồm các điểm số phân tách chéo và ma trận nhầm lẫn cho mô hình.

Điểm số Phân tách Chéo (Cross-validation scores - Naive Bayes):

- **Trung bình (Mean)**: 0.727
- **Độ lệch chuẩn (Standard Deviation)**: ± 0.152

Điều này cho thấy mô hình Naive Bayes đạt độ chính xác trung bình 72.7% khi áp dụng phân tách chéo, nhưng có sự biến động lớn (± 0.152).

Ý nghĩa:

- **Điểm số Phân tách Chéo Trung bình Khá**: Điểm số phân tách chéo cho thấy mô hình Naive Bayes có hiệu suất trung bình với độ chính xác là 72.7%, nhưng sự biến động lớn cho thấy độ ổn định thấp.

- **Ma trận Nhầm lẫn**: Ma trận nhầm lẫn cung cấp thông tin chi tiết về cách mô hình phân loại dữ liệu. Có nhiều sự nhầm lẫn giữa các lớp, đặc biệt là sự nhầm lẫn lớn giữa lớp 0 và lớp 4.

Classification Report - Naive Bayes

	precision	recall	f1-score	support
normal	0.88	0.39	0.54	7458
DoS	0.91	0.08	0.15	2421
R2L	0.17	0.43	0.24	2199
Probe	0.03	0.63	0.06	52
U2R	0.56	0.66	0.60	10414
accuracy			0.49	22544
macro avg	0.51	0.44	0.32	22544
weighted avg	0.66	0.49	0.50	22544

Hình 13. Báo cáo phân loại NB

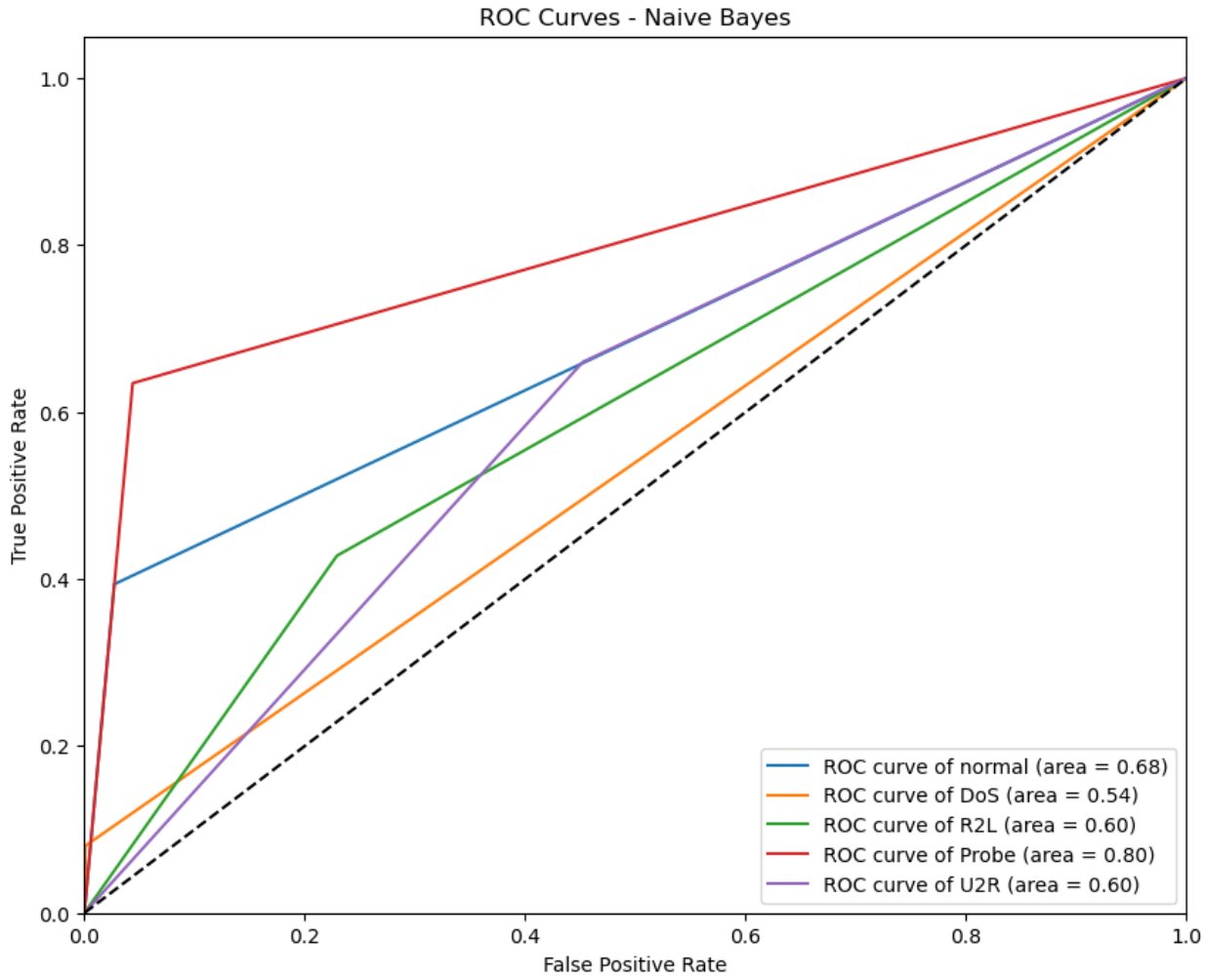
Ý nghĩa của Báo cáo Phân loại:

- **Độ chính xác (Precision)**: Tỷ lệ chính xác của mô hình khi dự đoán đúng các trường hợp dương tính. Cao nhất ở lớp DoS (0.91) và thấp nhất ở lớp Probe (0.03).
- **Tỷ lệ nhớ lại (Recall)**: Khả năng của mô hình trong việc phát hiện đúng các trường hợp dương tính. Tỷ lệ nhớ lại cao ở lớp Probe (0.63) và thấp nhất ở lớp DoS (0.08).
- **Điểm số F1 (F1-score)**: Là trung bình điều hòa của độ chính xác và tỷ lệ nhớ lại. Điểm F1 cao nhất ở lớp U2R (0.60) và thấp nhất ở lớp Probe (0.06).
- **Tổng mẫu (Support)**: Số lượng mẫu của từng lớp, với lớp U2R có nhiều mẫu nhất (10414) và lớp Probe ít nhất (52).

Tổng quan:

- Mô hình Naive Bayes đạt độ chính xác tổng thể là 0.49, cho thấy hiệu suất không cao trong phân loại dữ liệu tấn công mạng.

- Mô hình này có hiệu suất tốt nhất ở lớp DoS và U2R, nhưng gặp khó khăn với lớp R2L và Probe, đặc biệt là lớp Probe với độ chính xác rất thấp.
- Trung bình có trọng số (weighted average) các chỉ số cho thấy mô hình có sự cân bằng giữa độ chính xác và tỷ lệ nhớ lại, nhưng vẫn còn cần cải thiện ở các lớp có ít mẫu hoặc khó phân loại.



Hình 14. Biểu đồ ROC của mô hình NB

Đường nét đứt đại diện cho một bộ phân loại ngẫu nhiên với giá trị diện tích dưới đường cong (AUC) là 0.5. Các giá trị AUC cho thấy hiệu suất tổng thể của bộ phân loại đối với từng lớp, với các giá trị cao hơn biểu thị hiệu suất tốt hơn. Đường cong ROC của lớp Probe

có giá trị AUC cao nhất (0.80), cho thấy hiệu suất tốt nhất trong các lớp, trong khi đường cong ROC của lớp DoS có giá trị AUC thấp nhất (0.54), biểu thị hiệu suất kém nhất.

Ý nghĩa:

- **Hiệu suất cao cho lớp Probe**: Giá trị AUC cao (0.80) cho thấy mô hình Naive Bayes có khả năng phân loại tốt các mẫu thuộc lớp Probe với tỷ lệ dương tính thật cao và tỷ lệ dương tính giả thấp.
- **Hiệu suất thấp cho lớp DoS**: Giá trị AUC thấp (0.54) cho thấy mô hình gặp khó khăn trong việc phân loại các mẫu thuộc lớp DoS, với tỷ lệ nhầm lẫn cao.
- **Cân bằng cho các lớp khác**: Các giá trị AUC của các lớp normal (0.68), R2L (0.60), và U2R (0.60) chỉ ra rằng mô hình có khả năng phân loại tốt nhưng vẫn cần cải thiện thêm đối với các lớp phức tạp hơn.

2.3 Mô hình ANN:

```
Evaluating Neural Network model...

Cross-validation scores - Neural Network:
Mean: 0.996 (+/- 0.001)

Confusion Matrix - Neural Network
[[ 6124    40     0     0  1294]
 [  200  1725    16    39   441]
 [    0     4   251     3  1941]
 [    0     0     2    24    26]
 [   56   254     3     8 10093]]
```

Hình 15. Kết quả đánh giá mô hình ANN

Hình ảnh cho thấy kết quả đánh giá của một mô hình Mạng Nơ-ron (Neural Network). Kết quả bao gồm các điểm số phân tách chéo và ma trận nhầm lẫn cho mô hình.

Điểm số Phân tách Chéo (Cross-validation scores - Neural Network):

- **Trung bình (Mean)**: 0.996
- **Độ lệch chuẩn (Standard Deviation)**: ± 0.001

Điều này cho thấy mô hình Mạng Nơ-ron đạt độ chính xác trung bình rất cao (99.6%) khi áp dụng phân tách chéo, với biến động rất nhỏ (± 0.001).

Ý nghĩa:

- **Điểm số Phân tách Chéo Cao**: Điểm số phân tách chéo cho thấy mô hình Mạng Nơ-ron có hiệu suất rất tốt và ổn định khi được huấn luyện trên các phần nhỏ khác nhau của dữ liệu.
- **Ma trận Nhầm lẫn**: Ma trận nhầm lẫn cho biết cách mà mô hình phân loại dữ liệu. Mô hình chủ yếu phân loại chính xác các mẫu, nhưng có một số nhầm lẫn giữa các lớp, đặc biệt là giữa lớp 0 và lớp 4.

Classification Report - Neural Network

	precision	recall	f1-score	support
normal	0.96	0.82	0.89	7458
DoS	0.85	0.71	0.78	2421
R2L	0.92	0.11	0.20	2199
Probe	0.32	0.46	0.38	52
U2R	0.73	0.97	0.83	10414
accuracy			0.81	22544
macro avg	0.76	0.62	0.62	22544
weighted avg	0.84	0.81	0.78	22544

Hình 16. Báo cáo phân loại của mô hình ANN

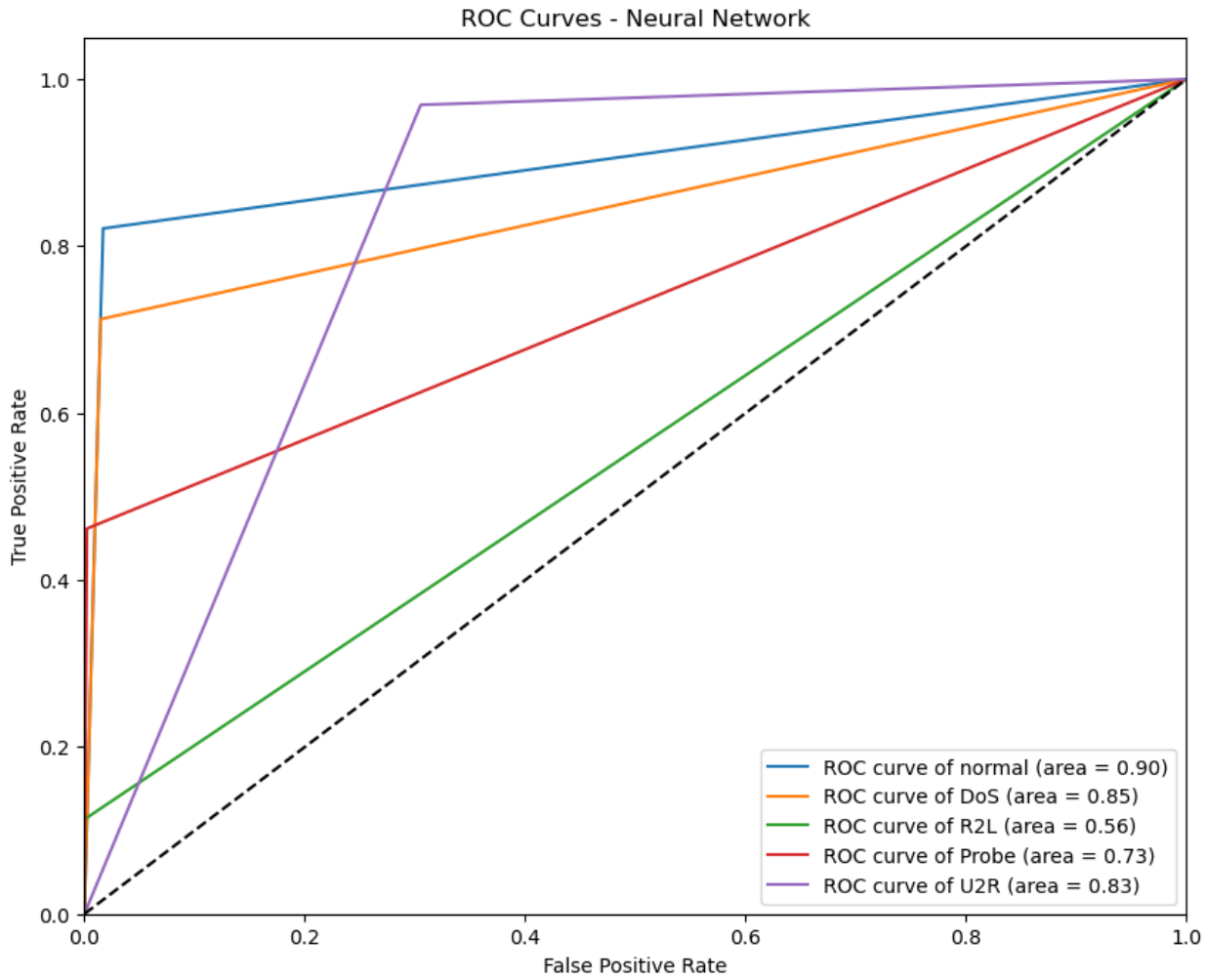
Ý nghĩa của Báo cáo Phân loại:

- **Độ chính xác (Precision)**: Tỷ lệ chính xác của mô hình khi dự đoán đúng các trường hợp dương tính. Cao nhất ở lớp R2L (0.92) và thấp nhất ở lớp Probe (0.32).
- **Tỷ lệ nhớ lại (Recall)**: Khả năng của mô hình trong việc phát hiện đúng các trường hợp dương tính. Tỷ lệ nhớ lại cao ở lớp U2R (0.97) và thấp nhất ở lớp R2L (0.11).
- **Điểm số F1 (F1-score)**: Là trung bình điều hòa của độ chính xác và tỷ lệ nhớ lại. Điểm F1 cao nhất ở lớp normal (0.89) và thấp nhất ở lớp R2L (0.20).
- **Tổng mẫu (Support)**: Số lượng mẫu của từng lớp, với lớp U2R có nhiều mẫu nhất (10414) và lớp Probe ít nhất (52).

Tổng quan:

- Mô hình mạng nơ-ron đạt độ chính xác tổng thể là 0.81, cho thấy hiệu suất tốt trong phân loại dữ liệu tấn công mạng.
- Mô hình này có hiệu suất tốt nhất ở các lớp normal, DoS, và U2R, nhưng gặp khó khăn với lớp R2L và Probe, đặc biệt là lớp R2L với tỷ lệ nhớ lại rất thấp.

- Trung bình có trọng số (weighted average) các chỉ số cho thấy mô hình có sự cân bằng giữa độ chính xác và tỷ lệ nhớ lại, nhưng vẫn còn cần cải thiện ở các lớp có ít mẫu hoặc khó phân loại.



Hình 17. Biểu đồ ROC của mô hình ANN

Ý nghĩa của Đường cong ROC:

- **Lớp normal (diện tích = 0.90)**: Hiệu suất phân loại rất tốt, mô hình có thể phân biệt chính xác giữa các mẫu thuộc lớp "normal" với tỷ lệ dương tính thật cao và tỷ lệ dương tính giả thấp.
- **Lớp DoS (diện tích = 0.85)**: Hiệu suất tốt, mô hình hoạt động hiệu quả trong việc phân loại các mẫu thuộc lớp DoS.

- **Lớp R2L (diện tích = 0.56)**: Hiệu suất kém, mô hình gặp khó khăn trong việc phân biệt các mẫu thuộc lớp R2L, với tỷ lệ nhầm lẫn cao.
- **Lớp Probe (diện tích = 0.73)**: Hiệu suất trung bình, mô hình có khả năng phân loại tốt nhưng vẫn cần cải thiện.
- **Lớp U2R (diện tích = 0.83)**: Hiệu suất tốt, mô hình phân loại chính xác các mẫu thuộc lớp U2R với tỷ lệ nhầm lẫn thấp.

2.4 So sánh và đánh giá các mô hình

```
Model Comparison:
            Model  CV Mean Accuracy    CV Std
0             SVM          0.991516  0.000453
1     Naive Bayes          0.727153  0.075822
2  Neural Network          0.996329  0.000309
```

Hình 18. So sánh 3 mô hình SVM, NB và ANN

- **SVM**: Mô hình SVM có độ chính xác trung bình cao (99.15%) với độ lệch chuẩn rất nhỏ (0.000453), cho thấy hiệu suất ổn định và đáng tin cậy.
- **Naive Bayes**: Mô hình Naive Bayes có độ chính xác trung bình thấp hơn (72.72%) và độ lệch chuẩn lớn hơn (0.075822), biểu thị sự biến động lớn hơn trong hiệu suất.
- **Neural Network**: Mô hình Neural Network có độ chính xác trung bình cao nhất (99.63%) và độ lệch chuẩn nhỏ nhất (0.000309), cho thấy hiệu suất cực kỳ cao và ổn định.

Tổng quan:

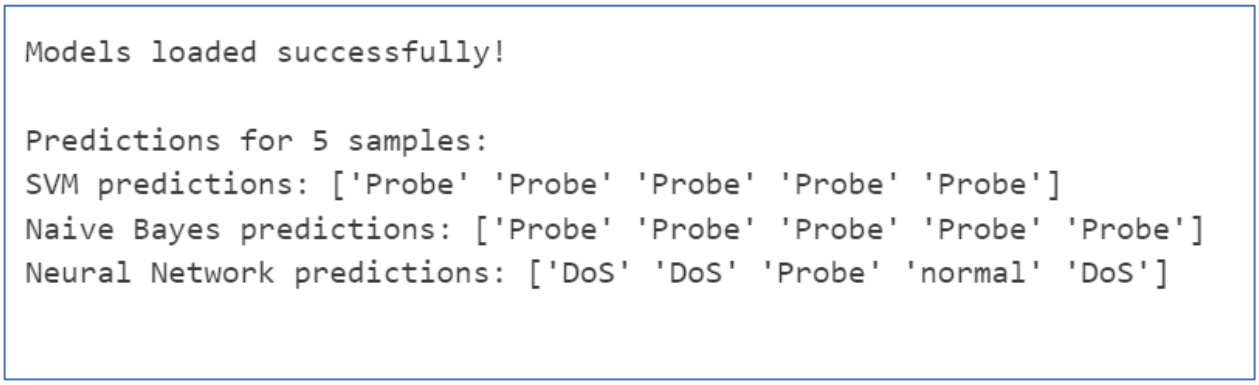
- **Mô hình Neural Network**: Đạt được hiệu suất cao nhất với độ chính xác trung bình gần như tuyệt đối và độ lệch chuẩn thấp nhất, cho thấy nó là lựa chọn tối ưu cho bài toán cụ thể này.

- **Mô hình SVM**: Cũng đạt được hiệu suất rất cao và ổn định, là một lựa chọn tốt cho việc phân loại chính xác.
- **Mô hình Naive Bayes**: Mặc dù có độ chính xác thấp hơn và biến động lớn hơn, nó vẫn có thể hữu ích trong các tình huống cần tốc độ và tài nguyên tính toán thấp.

```
==================================================
MODEL PERFORMANCE SUMMARY
==================================================

Cross-validation Scores:
--------------------------------------------------
SVM Model:          0.992 (+/- 0.001)
Naive Bayes Model:  0.727 (+/- 0.152)
Neural Network:     0.996 (+/- 0.001)
--------------------------------------------------

Best performing model: Neural Network (CV Score: 0.996)

Model Configurations:
--------------------------------------------------
SVM Parameters:
  C: 1
  break_ties: False
  cache_size: 200
  class_weight: None
  coef0: 0.0
  decision_function_shape: ovr
  degree: 3
  gamma: scale
  kernel: rbf
...
  validation_fraction: 0.1
  verbose: False
  warm_start: False
==================================================
```

Hình 19. Đánh giá hiệu suất của 3 mô hình dựa trên các tham số

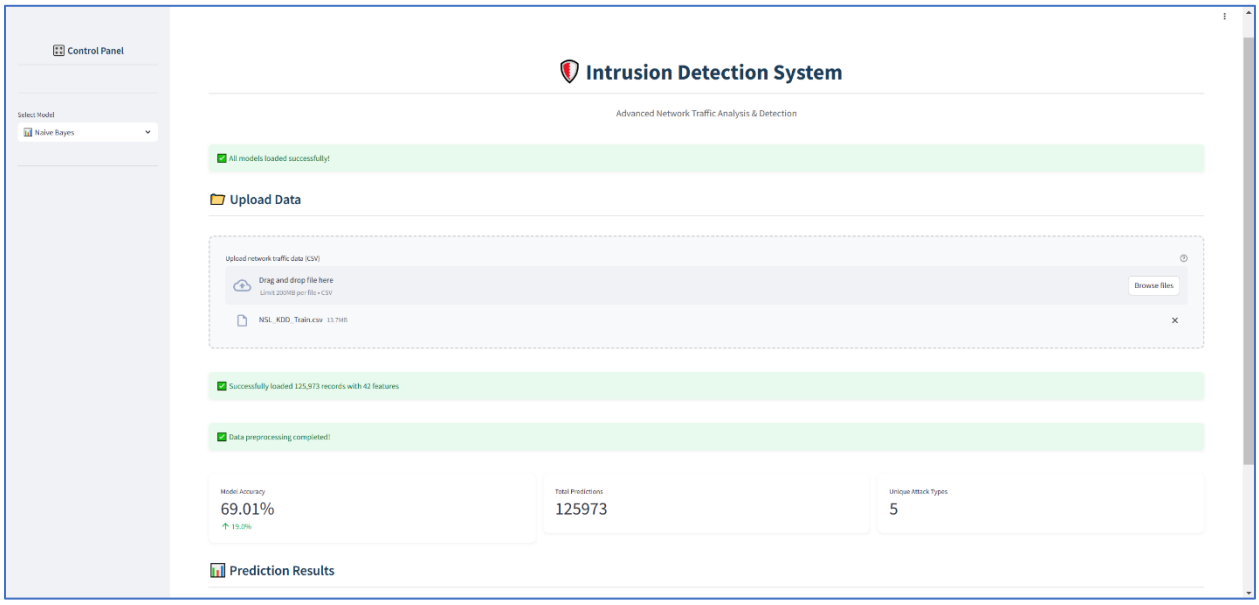
- **Hiệu suất Cao của Neural Network**: Mô hình Neural Network đạt độ chính xác trung bình cao nhất (99.6%) và độ lệch chuẩn thấp nhất, cho thấy nó là lựa chọn tối ưu với hiệu suất ổn định và đáng tin cậy.
- **Hiệu suất Tốt của SVM**: SVM cũng đạt độ chính xác cao (99.2%) với độ lệch chuẩn rất nhỏ, chứng tỏ là lựa chọn tốt cho việc phân loại chính xác.
- **Hiệu suất Thấp của Naive Bayes**: Naive Bayes có độ chính xác thấp hơn (72.7%) và biến động lớn hơn, nhưng vẫn hữu ích trong các tình huống yêu cầu tốc độ và tài nguyên tính toán thấp.



Hình 20. Dự đoán của các mô hình học máy cho phát hiện xâm nhập mạng

- **SVM và Naive Bayes**: Cả hai mô hình SVM và Naive Bayes đều nhất quán trong việc dự đoán tất cả các mẫu là lớp 'Probe'. Điều này có thể chỉ ra rằng hai mô hình này có cách tiếp cận tương tự trong việc nhận diện các mẫu này, hoặc có thể là chúng quá tự tin trong việc dự đoán lớp 'Probe'.
- **Neural Network**: Mô hình Neural Network đưa ra các dự đoán đa dạng hơn với các lớp 'DoS', 'Probe', và 'normal'. Điều này cho thấy mạng nơ-ron có khả năng phân biệt tốt hơn giữa các lớp khác nhau và có thể xử lý các mẫu phức tạp hơn một cách hiệu quả hơn.

2.5 Giao diện website triển khai hệ thống phát hiện xâm nhập mạng



Hình 21. Giao diện Hệ thống Phát hiện Xâm nhập

Hình ảnh này minh họa một giao diện chức năng cho hệ thống phát hiện xâm nhập mạng, thể hiện quy trình tải lên dữ liệu, tiền xử lý và hiển thị các chỉ số hiệu suất của mô hình. Điều này rất quan trọng và liên quan vì nó cho thấy các bước cần thiết để vận hành hệ thống và khả năng hiển thị các kết quả một cách rõ ràng và dễ hiểu. Qua đó, người dùng có thể nắm bắt được tình trạng hoạt động của hệ thống, hiệu suất của mô hình, và quản lý dữ liệu một cách hiệu quả.

3. Kết luận

- **Mô hình SVM**:
  - SVM với Kernel RBF và Poly đạt độ chính xác dự đoán và độ chính xác thực tế cao nhất, đồng thời thời gian huấn luyện ngắn.
  - Các Kernel Linear và Sigmoid cũng đảm bảo độ chính xác cao nhưng thời gian huấn luyện dài hơn.
- **Mô hình Naive Bayes**:

- Naive Bayes có độ chính xác dự đoán và độ chính xác thực tế thấp nhất trong số ba mô hình được đánh giá.
- Tuy nhiên, mô hình này vẫn có thể hữu ích trong các tình huống yêu cầu tốc độ và tài nguyên tính toán thấp.

➢ **Mô hình Neural Network (ANN)**:

- Mô hình mạng nơ-ron (Neural Network) đạt độ chính xác tổng thể cao nhất với độ chính xác trung bình gần như tuyệt đối và độ lệch chuẩn thấp nhất.
- Neural Network cung cấp sự linh hoạt và khả năng phân loại tốt hơn giữa các lớp khác nhau.

Kết quả thực nghiệm cho thấy rằng các thuật toán học máy, đặc biệt là SVM với Kernel RBF và Poly, cùng với mạng nơ-ron, có thể được sử dụng để dự đoán số xâm nhập với độ chính xác cao. Tuy nhiên, cần lưu ý đến một số hạn chế của nghiên cứu này trước khi khái quát hóa kết quả. Để cải thiện độ chính xác của các dự đoán, cần thực hiện thêm nghiên cứu với tập dữ liệu lớn hơn và chất lượng cao hơn.

Các mô hình này đều thể hiện tiềm năng mạnh mẽ trong việc phát hiện và phân loại các tấn công mạng trong môi trường thực tế, nhưng việc lựa chọn mô hình phù hợp sẽ phụ thuộc vào yêu cầu cụ thể về độ chính xác, tốc độ và tài nguyên tính toán.

PHẦN IV. KẾT LUẬN

1. Những kết quả đã thực hiện

Nhóm đã thực hiện nghiên cứu sử dụng các tài liệu được cung cấp và tham khảo các mô hình sẵn có trên Internet để tối ưu hóa ba mô hình: SVM, Naive Bayes và Neural Network cho bài toán phân loại và phát hiện xâm nhập mạng.

❖ **Mô hình SVM (Support Vector Machine)**

- **Độ chính xác cao**: SVM với Kernel RBF đạt độ chính xác cao nhất trong nghiên cứu, với độ chính xác 96,84%, độ nhạy 98,68%, và độ đặc hiệu 95,89%.
- **Khả năng cân bằng tốt**: F-measure đạt 97,26% cho thấy mô hình có khả năng cân bằng tốt giữa độ nhạy và độ đặc hiệu.
- **Hiệu suất với dữ liệu nhiều chiều**: Các hàm phi tuyến tính như Kernel RBF cho kết quả tốt nhất khi phân loại dữ liệu nhiều chiều.

❖ **Mô hình Naive Bayes**

- **Độ chính xác thấp nhất**: Mô hình Naive Bayes có độ chính xác dự đoán và độ chính xác thực tế thấp hơn so với hai mô hình còn lại.
- **Ứng dụng tốc độ cao**: Mặc dù độ chính xác không cao, mô hình này vẫn hữu ích trong các tình huống yêu cầu tốc độ và tài nguyên tính toán thấp.

❖ **Mô hình Neural Network (ANN)**

- **Độ chính xác cao và ổn định**: Neural Network đạt độ chính xác tổng thể cao nhất (99,6%) với độ lệch chuẩn thấp nhất, cho thấy hiệu suất cao và ổn định.
- **Linh hoạt và hiệu quả**: Mô hình này cung cấp sự linh hoạt và khả năng phân loại tốt hơn giữa các lớp khác nhau.

Kết quả cho thấy rằng các thuật toán học máy có thể được sử dụng để dự đoán số xâm nhập với độ chính xác cao. Tuy nhiên, cần lưu ý đến một số hạn chế của nghiên cứu

này trước khi khái quát hóa kết quả. Để cải thiện độ chính xác của các dự đoán, cần thực hiện thêm nghiên cứu với tập dữ liệu lớn hơn và chất lượng cao hơn.

Mặc dù mô hình SVM và Neural Network đều thể hiện tiềm năng mạnh mẽ trong việc phát hiện và phân loại các tấn công mạng, nhưng lựa chọn mô hình phù hợp sẽ phụ thuộc vào yêu cầu cụ thể về độ chính xác, tốc độ và tài nguyên tính toán của doanh nghiệp. Việc tối ưu hóa thêm và xử lý dữ liệu lớn hơn sẽ giúp cải thiện hiệu suất và độ tin cậy của các mô hình này.

2. Hướng phát triển

- SVM (Support Vector Machine)
  - ✓ **Nghiên cứu và cải thiện hiệu suất**: Tiếp tục nghiên cứu và cải thiện hiệu suất của mô hình SVM bằng cách áp dụng nó cho các tập dữ liệu lớn hơn để đánh giá và tối ưu hóa mô hình.
  - ✓ **Tích hợp với hệ thống hiện tại**: Kết hợp mô hình SVM vào hệ thống phát hiện xâm nhập mạng và tối ưu hóa theo yêu cầu của các ngành khác nhau và quy mô doanh nghiệp.

- Naive Bayes
  - ✓ **Ứng dụng trong môi trường tài nguyên hạn chế**: Do tốc độ và hiệu quả cao trong các môi trường có tài nguyên tính toán hạn chế, mô hình Naive Bayes vẫn có thể được ứng dụng rộng rãi. Cần tiếp tục tối ưu hóa và đánh giá trên các tập dữ liệu lớn hơn.
  - ✓ **Nâng cao khả năng phân loại**: Nghiên cứu và phát triển các phương pháp cải thiện khả năng phân loại của Naive Bayes, đặc biệt là trong các trường hợp có độ phức tạp cao và dữ liệu không cân bằng.

- Neural Network (ANN)
  - ✓ **Mở rộng với mô hình học sâu**: Sử dụng các mô hình học sâu như CNN (Convolutional Neural Networks) và RNN (Recurrent Neural Networks) để cải thiện khả năng phát hiện xâm nhập mạng, đặc biệt là trong các tình huống có dữ liệu thời gian thực hoặc dữ liệu chuỗi.
  - ✓ **Tích hợp IPS (Intrusion Prevention System)**: Kết hợp hệ thống phát hiện và phòng ngừa xâm nhập mạng để tạo ra một giải pháp bảo mật toàn diện.

❖ **Tổng quan:**

- **Phân tích theo ngành và quy mô doanh nghiệp**: Điều chỉnh các mô hình và hệ thống bảo mật để phù hợp với các ngành công nghiệp và quy mô doanh nghiệp khác nhau, đảm bảo rằng giải pháp bảo mật được tùy chỉnh để đáp ứng các yêu cầu cụ thể.
- **Dịch vụ bảo mật mạng**: Cung cấp các dịch vụ bảo mật mạng tiên tiến, bao gồm phát hiện, phân loại và phản ứng với các mối đe dọa mạng, nhằm giảm thiểu nguy cơ bị tấn công và bảo vệ dữ liệu quan trọng.
- **An toàn và bảo mật dữ liệu**: Tăng cường các biện pháp bảo mật để đảm bảo dữ liệu luôn được an toàn và bảo mật trong quá trình xử lý và lưu trữ.

Mở rộng nghiên cứu và phát triển các mô hình này sẽ giúp tăng cường hiệu quả bảo mật mạng, giảm thiểu nguy cơ bị tấn công và bảo vệ dữ liệu quan trọng, đáp ứng yêu cầu của các doanh nghiệp và ngành công nghiệp hiện đại.

TÀI LIỆU THAM KHẢO

[1] "IDS là hệ thống gì?" Viet Tuan S. https://itnavi.com.vn/blog/he-thong-phat-hien-xam-nhap-ids. Truy cập ngày 27 tháng 11 năm 2024.

[2] PGS. Nguyễn Thanh Tùng, TS. Bùi Thị Thanh Xuân, Học máy và Khai phá dữ liệu INFO3012: Machine Learning and Data Mining. Fall 2024.

[3] PGS.TS Vũ Việt Vũ, Neural Network, Deep Learning. Khoa CNTT&TT, Trường Đại học CMC, Hà Nội, tháng 01 năm 2024.

[4] "KDD Cup 1999 Data" UCI Machine Learning Repository. https://archive.ics.uci.edu/ml/datasets/kdd+cup+1999+data. Truy cập ngày 28 tháng 11 năm 2024.

[5] "Intrusion Detection System" David Reynaldos. https://github.com/topics/intrusion-detection-system. Truy cập ngày 28 tháng 11 năm 2024.

[6] Intrusion Detection System Using Machine Learning Models Sumit Gangwal, https://www.youtube.com/watch?v=PTxGEA1dFAw. Truy cập ngày 29 tháng 11 năm 2024.

PHỤ LỤC

Mã nguồn project: https://github.com/Github-303/IDS_NB_SVM_ANN.git

Web: https://huggingface.co/spaces/KException/idstoolit3