HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

LÊ ANH TUẤN

NGHIÊN CỨU, SO SÁNH MỘT SỐ THUẬT TOÁN CÂY QUYẾT ĐỊNH TRONG PHÁT HIỆN CÁC CUỘC TẤN CÔNG MẠNG DỰA TRÊN BỘ DỮ LIỆU KDD99 VÀ UNSW-NB15

Chuyên ngành: Khoa học máy tính

Mã số : 8.48.01.01

TÓM TẮT LUẬN VĂN THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

HÀ NỘI – NĂM 2020

Luận văn được hoàn thành tại:

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Người hướng dẫn khoa học: **TS. NGÔ QUỐC DŨNG**

Phản biện 1: ……

Phản biện 2: ……..

Luận văn sẽ được bảo vệ trước Hội đồng chấm luận văn thạc sĩ tại Học viện Công nghệ Bưu chính Viễn thông

Vào lúc: ..... giờ ..... ngày ..... tháng .... năm 2020

LỜI MỞ ĐẦU

1. Lý do chọn đề tài.

Kể từ nhưng năm 90 của thế kỷ XX, chính phủ tại một số quốc gia cũng như nhiều chuyên gia đã bắt đầu nghiên cứu về "thành phố thông minh", đó là việc xây dựng thành phố sử dụng các thành tựu công nghệ thông tin để thu thập và xử lý dữ liệu để quản lý tài sản và tài nguyên một cách hiệu quả. Trong những năm gần đây, các quốc gia đã có sự quan tâm đặc biệt tới vấn đề xây dựng thành phố thông minh do sự thay đổi về công nghệ, kinh tế và môi trường, ví dụ về các chương trình xây dựng thành phố thông minh đã được triển khai tại Singapore, Dubai, Milton Keynes, Southampton, Barcelona, và Việt Nam.

Để xây dựng một thành phố thông minh cần có sự thu thập, kết nối và xử lý một lượng thông tin khổng lồ. Các thông tin thường được thu thập bằng các cảm biến nhỏ từ người dân, thiết bị và tài sản, sau đó sẽ được tổng hợp và xử lý. Do thông tin cần thu thập là rất lớn nên vấn đề bảo mật và quyền riêng tư cá nhân là một vấn đề cần quan tâm. Các hệ thống lớn luôn có một hệ thống phòng thủ đủ mạnh để chống lại hầu hết các hành vi tấn công và xâm nhập trái phép, song đối với các hệ thống nhỏ như các sensor thì thường không có hệ thống phòng thủ nào hoặc không đủ để đảm bảo an toàn.

Đầu năm 2018, IBM X-Force Red và Threatcare đã phát hiện ra 17 lỗ hổng "zero-day" trong các hệ thống cảm biến và điều khiển thành phố thông minh được sử dụng tại các thành phố trên khắp thế giới. Các lỗ hổng này cho phép hacker truy cập vào và điều khiển thao tác dữ liệu, và chỉ cần một cảnh báo sai của hệ thống cảm biến có thể gây ra tổn hại lớn. Từ đó, IBM có đưa ra một số hướng dẫn để đảm bảo an toàn cho hệ thống như sau:

+ Thực hiện các hạn chế địa chỉ IP cho những máy có thể kết nối với các thiết bị, đặc biệt với các thiết bị sử dụng mạng internet công cộng.
+ Tận dụng các công cụ quét ứng dụng cơ bản để xác định các lỗ hổng của thiết bị.
+ Sử dụng các quy tắc bảo mật mạng để ngăn chặn truy cập vào các hệ thống nhạy cảm và thường xuyên thay đổi mật khẩu.
+ Vô hiệu hóa các tính năng quản trị từ xa và những cổng không cần thiết.
+ Sử dụng các công cụ quản lý sự kiện để quét lưu lượng mạng và xác định lưu lượng truy cập đáng ngờ.
+ Sử dụng hacker mũ trắng để thử nghiệm độ an toàn của hệ thống.

Trong đó, phương pháp sử dụng các công cụ quản lý sự kiện để quét lưu lượng mạng và xác định lưu lượng truy cập đáng ngờ được coi là biện pháp đơn giản, dễ thực hiện với các hệ thống nhỏ do có chi phí rẻ, dễ triển khai và cài đặt.

Thực tế đã có nhiều nghiên cứu về phân tích lưu lượng mạng để đưa ra cảnh báo. Tuy nhiên các phương pháp trên đều có các hạn chế riêng và dễ bị hacker lợi dụng để nó tránh bị phát hiện.

Với những lý do trên, việc nghiên cứu đề tài "*Nghiên cứu, so sánh một số thuật toán cây quyết định trong phát hiện các cuộc tấn công mạng trên bộ dữ liệu kdd99 và unsw-nb15*" sẽ mang lại ý nghĩa khoa học và thực tế trong vấn đề bảo mật và an toàn.

2. Mục tiêu, nhiệm vụ nghiên cứu

Mục tiêu nghiên cứu: Nghiên cứu về xây dựng một hệ thống phân tích, phát hiện hành vi tấn công bằng phương pháp sử dụng thuật toán học máy.

+ Tìm hiểu về việc thu thập và xử lý dữ liệu.
+ Tìm hiểu về các thuật toán cây quyết định (Decision Tree) trong học máy.

+ Sử dụng các thuật toán để xây dựng hệ thống phát hiện các cuộc tấn công mạng dựa trên dữ liệu về lưu lượng mạng.

Nhiệm vụ nghiên cứu: Để đạt được mục tiêu nghiên cứu, cần thực hiện lần lượt các nhiệm vụ sau:

+ Nghiên cứu về hệ thống phát hiện hành vi tấn công dựa trên phân tích lưu lượng mạng.
+ Nghiên cứu, xây dựng và so sánh nhóm thuật toán học máy Decision Tree trong việc phân tích dữ liệu mạng.
+ Nghiên cứu và sử dụng bộ dữ liệu hành vi mạng kdd99 và unsw-nb15.
+ Tiến hành áp dụng với dữ liệu thực tế và đánh giá hiệu quả.

3. Đối tượng và phạm vi nghiên cứu của đề tài

+ Vấn đề xây dựng hệ thống phát hiện hành vi đối với thiết bị vừa và nhỏ.
+ Sử dụng bộ dữ liệu hành vi mạng kdd99 và unsw-nb15.
+ Quy trình xây dựng mô hình học máy, nhóm các thuật toán Decision Tree.

4. Phương pháp nghiên cứu

Để hoàn thành mục tiêu, luận văn đã kết hợp sử dụng phương pháp nghiên cứu tài liệu và nghiên cứu thực tiễn.

4.1. Phương pháp nghiên cứu tài liệu

- *Phương pháp phân tích và tổng hợp lý thuyết:* Luận văn đã thực hiện phân tích, tổng hợp một số bài báo khoa học có liên quan đến vấn đề cần nghiên cứu được đăng trên các tạp chí, hội nghị uy tín trên thế giới được cộng đồng nghiên cứu sử dụng.

- *Phương pháp phân loại và hệ thống hóa lý thuyết:* Từ những kiến thức thu được bằng phân tích và tổng hợp lý thuyết, luận văn đã hệ thống và sắp xếp lại các thông tin thu được một cách khoa học, đồng thời sử dụng chúng để nhận định, đánh giá các phương pháp đã có, từ đó có những đề xuất tìm ra các phương pháp mới tối ưu hơn cho bài toán đặt ra.

4.2 Phương pháp nghiên cứu thực tiễn

- *Phương pháp thực nghiệm khoa học*: Sử dụng các phương pháp đã có để áp dụng cho bài toán đặt ra, phương pháp này giúp kiểm chứng tính chính xác và tính khả thi của những giải pháp, thuật toán được đề xuất của đề tài và cũng là cơ sở để đánh giá tính hiệu quả so với các phương pháp đã có về mặt thực nghiệm.

- *Phương pháp thống kê*: Từ những kết quả, số liệu từ phương pháp thực nghiệm khoa học, luận văn tiến hành tổng hợp, thống kê, xử lý và mô tả bằng các biểu đồ thích hợp, phục vụ quá trình phân tích đánh giá.

5. Kết cấu đề tài

Ngoài phần mở đầu, kết luận, danh mục tài liệu tham khảo và phụ lục, đề tài của tôi gồm 3 chương:

Chương 1: Tổng quan về tấn công qua mạng và các nghiên cứu liên quan.

Chương 2: Phương pháp đề xuất.

Chương 3: Thực nghiệm và kết quả.

CHƯƠNG 1. TỔNG QUAN VỀ TẤN CÔNG MẠNG VÀ CÁC NGHIÊN CỨU LIÊN QUAN

1.1. Thực trạng về vấn đề tấn công mạng.

1.1.1. Xu thế phát triển và các vấn đề về àn toàn thông tin.

Do ảnh hưởng của cuộc cách mạng 4.0, hướng tới sự kết nối và chia sẻ thông tin. Biểu hiện ở việc xây dựng thành phố thông minh, phổ cập Internet, ứng dụng chia sẻ, sử dụng trí tuệ nhân tạo,... Đặc biệt gần đây là sự kiện thương mại hóa mạng 5G để giúp đáp ứng các nhu cầu của cách mạng 4.0.

Do nhu cầu quá lớn của các thiết bị kết nối mạng, cảm biến, và các thiết bị IoT, khiến các nhà sản xuất thiết bị trên bắt đầu chạy đua lợi nhuận, tăng mạnh về số sản lượng sản xuất nhưng không chú trọng nghiên cứu, cập nhật các vấn đề về mức an toàn của thiết bị. Từ đó dẫn tới hacker lợi dụng được các lỗ hổng bảo mật, "backdoor" tồn tại trên thiết bị.

Ngoài ra, các công trình nghiên cứu về bảo mật trên các thiết bị mạng nhỏ và vừa chỉ bắt đầu xuất hiện nhiều trong vòng vài năm gần đây, và chưa có sự phổ biến cao hoặc thương mại hóa để các nhà sản xuất có thể sử dụng dễ dàng. Các hệ thống kết nối mạng của các thiết bị nhỏ và vừa hiện tại không có một chuẩn chung về bảo mật để đánh giá khiến chúng dễ bị tấn công và lợi dụng bởi các hacker.

1.1.2. Sự phát triển của xu hướng tấn công các thiết bị mạng

Tại Việt Nam, chỉ riêng 6 tháng đầu năm 2018 đã phát hiện hơn 4.500 cuộc tấn công mạng nhằm vào các cơ quan Chính phủ, bộ, ngành với nhiều hình thức khác nhau. Việt Nam xếp thứ 4 trong tốp 10 quốc gia bị kiểm soát bởi mạng máy tính ma [13]. Tại Việt Nam đã xuất hiện một số vụ tấn công lớn như việc lộ lọt dữ liệu 5,4 triệu người dùng của Thế giới di động và được tung lên tại Raidforums dưới danh tính của một hacker ẩn danh, hoặc cuộc tấn công làm tê liệt hệ thống của VietNam Airlines và lấy đi dữ liệu cá nhân của 411.000 người dùng, trong đó có nhiều người dùng là hội viên "Bông sen vàng" đã gây ảnh hưởng nghiêm trọng và gây thiệt hại lớn.

Hình 1.5. Vụ tấn công làm thay đổi giao diện của trang chủ VietNam AirLines vào năm 2016.

Ngoài ra, trên thế giới nói chung và Việt Nam nói riêng đã có xu hướng chuyển dịch các hệ thống quan trọng như hệ thống khai thác dầu mỏ, hệ thống thủy điện, hệ thống tín hiệu giao

thông sang tự động hóa bằng máy móc. Và nếu những hệ thống trên bị xâm nhập và kiểm soát có thể dẫn tới nguy cơ ảnh hưởng tới an ninh cấp quốc gia.

1.2. Tấn công mạng và các nghiên cứu liên quan.

1.2.1. Tấn công mạng là gì.

Theo luật an ninh mạng ban hành năm 2018, hành vi tấn công mạng được định nghĩa: *"Tấn công mạng là hành vi sử dụng không gian mạng, công nghệ thông tin hoặc phương tiện điện tử để phá hoại, gây gián đoạn hoạt động của mạng viễn thông, mạng Internet, mạng máy tính, hệ thống thông tin, hệ thống xử lý và điều khiển thông tin, cơ sở dữ liệu, phương tiện điện tử".*

Quy trình tấn công gồm 5 bước lần lượt là:

1. Xác định mục tiêu.
2. Thu thập thông tin mục tiêu, tìm kiếm lỗ hổng.
3. Lựa chọn mô hình tấn công.
4. Thực hiện tấn công.
5. Xóa dấu vết (nếu cần thiết).

Có rất nhiều các phương pháp tấn công mạng khác nhau nhưng được quy về 3 phương pháp tấn công chính.

1. *Tấn công thăm dò*: Là phương pháp sử dụng các công cụ bắt gói tin tự động, quét cổng, và kiểm tra các dịch vụ đang chạy với mục đích là thu thập thông tin về hệ thống. Các công cụ để thăm dò rất phổ biến và dễ sử dụng, ví dụ như Nmap, Wireshark,...
2. *Tấn công truy cập*: Là phương pháp khai thác lỗ hổng trên các thiết bị của nạn nhân, ví dụ như các lỗ hổng trên dịch vụ, thiết bị, hoặc chính sách bảo mật. Phương pháp tấn công này đòi hỏi người tấn công phải có trình độ cao, thường không có các công cụ hỗ trợ hoặc một quy trình chung nào. Đây là hình thức tấn công ít gặp nhất nhưng cũng là hình thức gây thiệt hại nhiều nhất và khó phát hiện nhất.
3. *Tấn công từ chối dịch vụ*: Tấn công từ chối dịch vụ là phương thức tấn công làm cho một hệ thống nào đó bị quá tải và không thể cung cấp dịch vụ cho người dùng bình thường, làm gián đoạn hoạt động của hệ thống hoặc làm hệ thống phải ngừng hoạt động. Đây là hình thức tấn công phổ biến nhất. Việt Nam là một nước nằm trong nhóm bị ảnh hưởng nhiều do tấn công từ chối dịch vụ trên thế giới.

Hình 1.7. Lưu lượng tấn công DDoS trên toàn thế giới trong năm 2018 (Nguồn: https://www.blackmoreops.com)

1.2.2. Các nghiên cứu liên quan về tấn công mạng.

Việc nghiên cứu các vấn đề liên quan đến tấn công mạng và ngăn chặn tấn công mạng đã có từ những năm 90 của thế kỷ trước với rất nhiều đề xuất, phương pháp có tính khả thi khi áp dụng thực tế. Đặc biệt với các phương pháp phát hiện, chủ động phòng ngừa các hành vi tấn công mạng dựa trên phân tích hành vi người dùng hoặc phân tích các thông tin về lưu lượng mạng để đưa ra cảnh báo hoặc ngăn chặn trực tiếp. Các phương pháp đề xuất thường được chia làm 2 loại:

+ Tạo các tập mẫu có sẵn về thông tin, hành vi của người dùng và hành vi nào vượt quá ngưỡng của tập mẫu sẽ bị coi là hành vi bất thường.

+ Xây dựng hệ thông phát hiện xâm nhập dựa trên các hành vi khác thường của kẻ tấn công (tập luật). Dựa trên tập luật đó để quyết định một hành vi của người dùng có được coi là bất thường hay không.

Cả hai phương pháp đều có ưu điểm là dễ cấu hình, có tỷ lệ ngăn chặn tốt nếu chọn được tập mẫu hoặc cấu hình tập luật đủ tốt. Xong nhược điểm của các phương pháp trên là thiếu tính linh động, có thể đưa ra quyết định sai lầm khi có các thông tin mang tính ngẫu nhiên xuất hiện hoặc dễ dàng bị hacker nếu không cập nhật thường xuyên. Do đó, trong thời gian gần đây đã có các nghiên cứu thử nghiệm các mô hình tích hợp các thuật toán vào trong hệ thống trong phân tích và phát hiện các hành vi bất thường, đặc biệt là các mô hình sử dụng thuật toán học máy, và đem lại các kết quả rất khả quan về tính khả thi.

Lý do việc tích hợp các thuật toán học máy vào việc dự đoán và phát hiện tấn công là do đặc điểm của các thuật toán học máy có tính tự động học hỏi dựa trên dữ liệu đầu vào. Một mô hình học máy có thể tạo ra các bộ luật khác nhau đối với các hệ thống có dữ liệu khác nhau nhưng vẫn đảm bảo được hiệu quả khi kết hợp với các hệ thống bảo vệ sẵn có. Các mô hình học máy này thường được tích hợp trong hệ thống IDS và ứng dụng chúng để dự đoán các hành vi bất thường, phát hiện các cuộc tấn công mạng hoặc phân tích các gói tin mạng, tuy chưa có khả năng thay thế được một kỹ sư an ninh mạng nhưng mô hình này có thể hỗ trợ trong việc đưa ra phán đoán của người quản trị, đặc biệt là khi khối lượng dữ liệu quá lớn và vượt khỏi khả năng xử lý của con người.

Dưới đây là một số nghiên cứu nổi tiếng về ứng dụng học máy trong phát hiện và ngăn chặn hành vi bất thường có thể tham khảo:

1. Machine Learning Techniques for Intrusion Detection.
2. Long Short Term Memory Networks for Anomaly Detection in Time Series.
3. Anomaly Detection Framework Using Rule Extraction for Efficient Intrusion Detection.
4. A survey of network anomaly detection techniques.
5. Shallow and Deep Networks Intrusion Detection System: A Taxonomy and Survey.
6. Deep Packet: A Novel Approach For Encrypted Traffic Classification Using Deep Learning.
7. Performance Comparison of Intrusion Detection Systems and Application of Machine Learning to Snort System.
8. Evaluation of Machine Learning Algorithms for Intrusion Detection System.
9. One Class collective Anomaly Detection based on LSTM.
10. Network Traffic Anomaly Detection Using Recurrent Neural Networks.
11. Sequence Aggregation Rules for Anomaly Detection in Computer Network Traffic.
12. Big collection of all approaches for IDS.

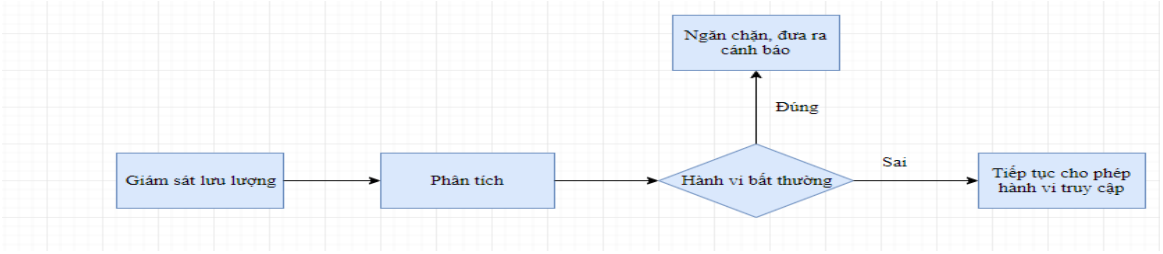
Có thể nhận thấy cách tiếp cận và phương pháp xây dựng mô hình rất đa dạng với việc ứng dụng rất nhiều thuật toán học máy, thậm chí cả thuật toán học sâu. Đối tượng dữ liệu để

phân tích cũng rất đa dạng như luồng dữ liệu mạng, bộ nhớ mạng, phân biệt hành vi người dùng.

1.3. Hệ thống phát hiện xâm nhập IDS

1.3.1. Giới thiệu về hệ thống phát hiện xâm nhập IDS

Hệ thống phát hiện xâm nhập (IDS) là một hệ thống bằng phần cứng hoặc phần mềm giám sát mạng nhằm phát hiện các hành vi bất thường vào hệ thống. Một IDS có nhiệm vụ phân tích các gói tin mà tường lửa cho phép đi qua, những hành vi bất thường sẽ được báo cáo cho người quản trị viên để có được hành động xử lý hoặc xử lý tự động.



Hình 1.8. Mô hình IDS

Các tính năng của hệ thống IDS bao gồm:

+ Giám sát lưu lượng mạng và các hành vi bất thường.
+ Cảnh báo về tình trạng mạng của hệ thống cho người quản trị.
+ Kết hợp với các hệ thống giám sát, tường lửa, diệt virus tạo thành một hệ thống bảo mật.

Một hệ thống IDS phải có đạt được những yêu cầu sau:

+ Tính chính xác: IDS không được nhầm các hành vi thông thường của người dùng là hành vi bất thường.
+ Tính trọn vẹn: IDS phải phát hiện được mọi xâm nhập trái phép hoặc hành vi tấn công vào hệ thống mạng. Đây cũng là điều rất khó khăn đạt, vì không hệ thống nào trên thế giới dám đảm bảo phát hiện được mà phải thường xuyên cập nhật, thay đổi.
+ Chịu lỗi: Bản thân hệ thống IDS cũng phải có khả năng ngăn chặn tấn công.
+ Khả năng mở rộng: Như đã nói, hệ thống IDS phải có khả năng cập nhật để duy trì và không bị lạc hậu.

1.3.2. Các kỹ thuật phát hiện của IDS

Có rất nhiều phương pháp được sử dụng để phát hiện xâm nhập được sử dụng để cấu hình cho một hệ thống IDS, nhưng các phương pháp được sử dụng nhiều nhất gồm:

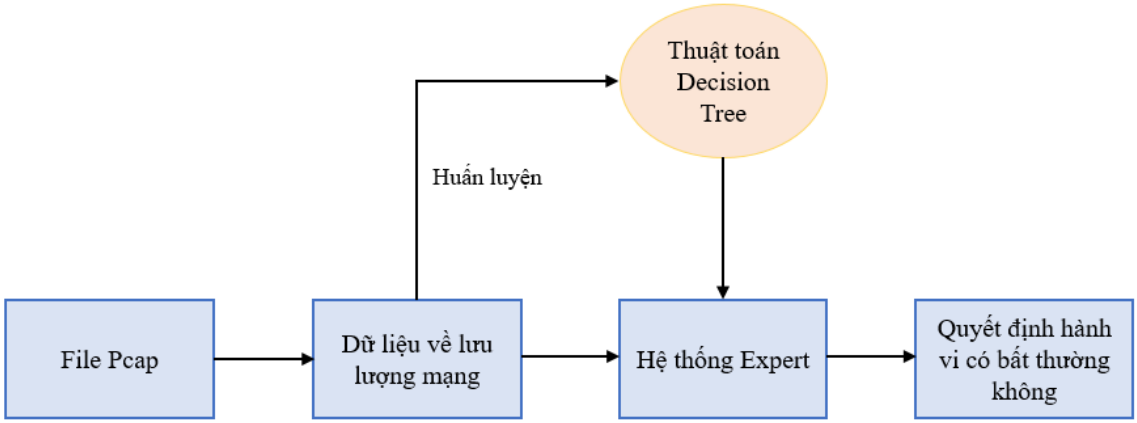
1. **Hệ thống Expert**: Hệ thống xây dựng một tập nguyên tắc đã được định nghĩa trước để miêu tả tấn công. Tất cả các sự kiện đều được kết hợp kiểm tra dưới dạng quy tắc if – then – else.
2. **Phân tích trạng thái phiên**: Một hành vi bất thường được miêu tả bằng một tập các mục tiêu và phiên cần được thực hiện để gây tổn hại hệ thống. Do đó nếu phát hiện hành vi trùng với phiên thì hệ thống sẽ coi đó là hành vi bất thường.
3. **Phân biệt ý định người dùng**: Kỹ thuật này sẽ mô hình hóa hành vi người dùng bằng một tập các mức cao nhất và người dùng bình thường có thể thực hiện trên hệ thống. Nếu có hành vi nào vượt quá thì sẽ coi là hành vi bất thường.
4. **Sử dụng Machine Learning**: Đây là kỹ thuật mới, trong đó hệ thống sẽ liên tục lưu trữ cả hành vi bình thường và bất thường mà thu thập được. Sau đó dựa vào thuật toán học máy để tạo thành bộ luật và dùng nó để tham chiếu dự đoán hành vi của người dùng.

CHƯƠNG 2. PHƯƠNG PHÁP ĐỀ XUẤT

2.1. Phương pháp đề xuất.

Dựa trên tìm hiểu về và phân tích về các mô hình phát hiện tấn công mạng đã được nghiên cứu. Luận văn cũng tiền hành đề xuất một mô hình IDS để phát hiện các cuộc tấn công mạng dựa trên phân tích lưu lượng mạng, đó là sử dụng các thuật toán cây quyết định để tiến hành phát hiện lưu lượng mạng bắt được có phải là hành vi của người dùng bình thường hay là hành vi tấn công vào hệ thống, từ đó quyết định ngăn chặn hay không.

Mô hình IDS đề xuất hoạt động như sau:



Hình 2.1. Mô hình IDS đề xuất

Lý do luận văn đề xuất mô hình này với việc thay đổi quan trọng nhất là sử dụng thuật toán học máy vào để sử dụng do đây là kỹ thuật mới, có độ chính xác, độ linh động cao, tự động cập nhật dựa trên quá trình tự học của hệ thống. Nhờ đó người quản trị không cần phải có kiến thức quá cao để sử dụng và cập nhật hệ thống, đặc biệt hiệu quả trong thời điểm thiếu nhân lực trong ngành an toàn thông tin. Trong phần sau, luận văn cũng giới thiệu lý do việc sử dụng nhóm thuật toán cây quyết định trong mô hình.

Các thuật toán học máy được sử dụng trong mô hình mà luận văn đề xuất sẽ sử dụng dữ liệu thư viện đã được xây dựng vì 2 lý do sau:

+ Tính đúng đắn của thư viện được đảm bảo.
+ Tính hiệu quả: các thuật toán đã được tối ưu hóa về các tổ chức, lưu trữ dữ liệu nên có tốc độ tốt hơn so với thuật toán tự xây dựng.

2.2. Thuật toán Cây quyết định

2.2.1. Giới thiệu về học máy và xây dựng mô hình học máy

2.2.1.1. Sơ lược về trí tuệ nhân tạo và học máy

Trí tuệ nhân tạo (AI) là một thuật ngữ miêu tả những trí tuệ được biểu diễn bởi bất cứ hệ thống nhân tạo nào. Thuật ngữ này thường dùng để nói tới các máy tính và các ngành khoa học nghiên cứu về các lý thuyết và ứng dụng của trí tuệ nhân tạo. Luận văn sẽ chỉ đề cập trong phạm vi của khoa học máy tính, trong đó "trí tuệ nhân tạo" được hiểu là trí tuệ do con người lập trình tạo nên với mục tiêu giúp máy tính có thể tự động hóa các hành vi thông minh như con người. Trí tuệ nhân tạo khác với việc lập trình logic trong các ngôn ngữ lập trình là ở việc

ứng dụng các hệ thống học máy (machine learning) để mô phỏng trí tuệ của con người trong các xử lý mà con người làm tốt hơn máy tính.

Trong lĩnh vực AI có một nhánh nghiên cứu về khả năng tự học của máy tính được gọi là học máy (machine learning). Hiện nay không có 1 định nghĩa chính thức nào về học máy cả nhưng có thể hiểu rằng nó là các kỹ thuật giúp cho máy tính có thể tự học mà không cần phải cài đặt các luật quyết định. Thường một chương trình máy tính cần các quy tắc, luật lệ để có thể thực thi được một tác vụ nào đó như dán nhãn cho các email là thư rác nếu nội dung email có chứ từ khoá "quảng cáo". Nhưng với học máy, các máy tính có thể tự động phân lại các thư rác thành mà không cần chỉ trước bất kỳ quy tắc nào cả. Đã có rất nhiều công trình nghiên cứu về học máy và cho phép bất cứ ai tham khảo, cùng với sự hỗ trợ từ các thư viện học máy phổ biến như scikit-learn, tensorflow, openAI,... nên việc tự nghiên cứu và đưa ra sản phẩm cũng đã bớt khó khăn đi rất nhiều.

2.2.1.2 Phân loại kỹ thuật học máy

Các thuật toán toàn học máy thường được chia làm 4 nhóm.

+ Học có giám sát (*Supervised Learning*): Là phương pháp sử dụng những dữ liệu đã được gán nhãn từ trước để suy luận ra quan hệ giữa đầu vào và đầu ra. Các dữ liệu này được gọi là dữ liệu huấn luyện và chúng là cặp các đầu vào - đầu ra. Học có giám sát sẽ xem xét các tập huấn luyện này để từ đó có thể đưa ra dự đoán đầu ra cho 1 đầu vào mới chưa gặp bao giờ. Biểu diễn theo toán học, là khi chúng ta có đầu vào là các biến $X=\{x_1,x_2,\ldots,x_2\}$ ứng với các nhãn $Y=\{y_1,y_2,\ldots,y_n\}$ trong đó x_i, y_i là các vector. Từ các dữ liệu này thuật toán sẽ đưa ra một hàm số.

$$yi \approx f(xi), \ \forall\, i = 1,2,\ldots,N$$

Khi đó với đầu vào là biến x_m thì sẽ cho ra biến y_m tương ứng.

+ Học không giám sát (*Unsupervised Learning*): Khác với học có giám sát, học phi giám sát sử dụng những dữ liệu chưa được gán nhãn từ trước để suy luận. Phương pháp này thường được sử dụng để tìm cấu trúc của tập dữ liệu. Tuy nhiên không có phương pháp đánh giá được cấu trúc tìm ra được là đúng hay sai. Theo biểu diễn toán học, là ta chỉ có tập các biến X mà không biết nhãn Y tương ứng của nó.

+ Học bán giám sát (*Semi Supervised Learning*): Là phương thức học ở giữa hai loại trên, tức là ta chỉ có một phần trong dữ liệu có gán nhãn. Thực tế cho thấy rất nhiều các bài toán Machine Learning thuộc vào nhóm này vì việc thu thập dữ liệu có nhãn tốn rất nhiều thời gian và có chi phí cao. Rất nhiều loại dữ liệu thậm chí cần phải có chuyên gia mới gán nhãn được (ảnh y học). Ngược lại, dữ liệu chưa có nhãn có thể được thu thập với chi phí thấp.

+ Học củng cố (*Reinforcement Learning***)**: Là phương thức học giúp cho hệ thống tự thích ứng và đạt được lợi ích cao nhất trong các hoàn cảnh khác nhau. Để đạt được điều này, cần có một hệ thống tự động sinh ra các hoàn cảnh khác nhau để hệ thống tự học và xây dựng các hành động hợp lý nhất. Hiện tại, học củng cố thường được áp dụng vào các bài toàn Lý thuyết trò chơi và xe tự lái.

2.2.1.3. Quy trình xây dựng một mô hình học máy

Học máy là một quá trình phức tạp do vậy cần có một quy trình thực hiện để đảm bảo hiệu quả. Một quy trình xây dựng hệ thống học máy thường có 3 bước: Thu thập, xử lý dữ liệu; lựa chọn thuật toán và tiến hành huấn luyện cho mô hình; kiểm nghiệm thực tế và đánh giá.

a. Thu thập, xử lý dữ liệu.

Trong khi xây dựng học máy, dữ liệu được coi là quan trọng nhất để quyết định khả năng dự đoán của hệ thống là tốt hay không. Dữ liệu trong học máy là rất quan trọng, xong không phải cứ nhiều dữ liệu là thuật toán sẽ chạy tốt, mà còn cần sự đa dạng, chính xác và khái quát từ thực tế. Ví dụ nếu dữ liệu đầu vào không có dữ liệu về tấn công DDoS thì máy tính sẽ không thể phán đoán khi nào hệ thống đang bị DDoS. Xử lý dữ liệu gồm có làm sạch dữ liệu và trích xuất đặc trưng để cung cấp cho mô hình.

b. Lựa chọn thuật toán và tiến hành huấn luyện cho mô hình.

Sau khi có dữ liệu, ta tiến hành chọn thuật toán và tiến hành huấn luyện cho hệ thống học máy (training). Và có rất nhiều các thuật toán học máy và người xây dựng sẽ phải lựa chọn thuật toán phù hợp với bài toán cần giải quyết, có thể kết hợp nhiều thuật toán và phương pháp tạo ra mô hình thích hợp.

Sau đó cần chia dữ liệu làm 2 phần: Phần để huấn luyện (training data) và phần để kiểm tra (testing data), tùy theo mô hình để chia tỷ lệ thích hợp. Tiến hành thử nghiệm và đánh giá mô hình để có sự điều chỉnh phù hợp.

c. Kiểm nghiệm thực tế và đánh giá.

Sau khi thử nghiệm, cần đưa mô hình ra sử dụng trong kiểm nghiệm thực tiễn. Từ đó phát hiện các thiếu sót như: dữ liệu thực tế khác biệt, mô hình hoạt động không phù hợp, thời gian chạy quá lâu, ... Và từ đó để tiến hành bổ sung, chỉnh sửa và hoàn thiện mô hình.

Trong luận văn, dữ liệu kiểm nghiệm và dữ liệu huấn luyện sẽ lấy từ bộ dữ liệu mạng unsw-nb15, do đó là bộ dữ liệu có tính thực tế cao, đầy đủ đã được sử dụng trong nhiều nghiên cứu khoa học khác.

2.2.2. Nhóm thuật toán cây quyết định

Trong các thuật toán của học máy, có một nhóm thuật toán đưa ra quyết định dựa trên các câu hỏi, nhóm thuật toán ấy được gọi là cây quyết định (Decision Tree). Các thuật toán xây dựng một cây quyết định với các nốt là các câu hỏi ứng một thuộc tính của dữ liệu, mỗi một nhánh của nốt sẽ biểu thị một kết quả khác nhau của câu hỏi tại nốt đó. Và đường dẫn từ gốc đến lá là đại diện cho một quy tắc phân loại.

Một ví dụ đơn giản về Decision Tree: Một sinh viên sẽ quyết định đi học dựa trên thông tin về thời tiết, nếu trời mưa thì sinh viên đó sẽ ở nhà, và nếu trời nắng thì đi học.

Cây quyết định là một trong những phương pháp học có giám sát tốt nhất và được sử dụng nhiều nhất. Các phương pháp tạo ra một mô hình cây có độ chính xác cao, ổn định và dễ theo dõi, loại bỏ các thuộc tính không cần thiết. Không giống các thuật toán có mô hinh tuyến tính, cây quyết định giải quyết các bài toàn có dữ liệu nhiễu rất tốt. Đây là lý do luận văn sử dụng thuật toán Decision Tree để xây dựng mô hình IDS.

Các ưu điểm của các thuật toán cây quyết định gồm:

1. Dễ dàng theo dõi khi nhìn vào cây.
2. Xử lý tốt với dữ liệu có dán nhãn đầy đủ, cho kết quả tốt.
3. Dữ liệu đầu vào càng lớn độ chính xác càng cao, ít bị ảnh hưởng bởi dữ liệu gây nhiễu.
4. Tốc độ đưa ra kết quả nhanh.

Các nhược điểm của nhóm thuật toán cây quyết định:

1. Xây dựng cây quyết định tốn thời gian.
2. Là thuật toán học có giám sát nên cần dữ liệu có dán nhãn rõ ràng.

3. Cây quyết định dễ bị hiện tượng "overfitting", là hiện tượng xảy ra khi tập dữ liệu huấn luyện quá phù hợp với mô hình dẫn tới việc dự đoán các kết quả không có trong tập dữ liệu huấn luyện thường sai.

Có rất nhiều thuật toán trong Decison Tree, nhưng do thời gian có hạn luận văn nên sẽ chỉ đề cập và sử dụng các thuật toán phổ biến và có hiểu quả cao. Đó là 4 thuật toán, trong đó có 3 thuật toán dựa trên tư tưởng của Hunt (ID3, C4.5, CART) và thuật toán Random Forest.

2.2.3. Các thuật toán dựa trên tư tưởng của Hunt

Tư tưởng về thuật toán này được Hunt và các đồng sự công bố vào năm 1966 được mô tả như sau:

+ Tại mỗi bước, mỗi thuộc tính tốt nhất sẽ được chọn ra dựa trên một tiêu chuẩn nào đó. Thuộc tính tốt nhất ở đây được hiểu là thuộc tính có ảnh hưởng cao nhất tới phán đoán kết quả.

+ Với mỗi thuộc tính được chọn, ta phân chia ra các nhánh của nốt dựa sự phân chia của dữ liệu trong thuộc tính đó.

+ Liên tục đệ quy với các thuộc tính còn lại đến khi chạm tới khi tất cả các thuộc tính đều được chọn. Khi đó ta có được cây quyết định.

+ Khi gặp dữ liệu để đưa ra dự đoán, thuật toán sẽ chọn đường đi từ nốt đầu tiền cho tới lá (kết quả) với mỗi ngã rẽ là các câu hỏi của từng nốt.

2.2.3.1. Entropy

Entropy sử dụng trong luận văn là entropy thông tin, nó một khái niệm mở rộng của entropy trong nhiệt động học và cơ học của vật lý. Entropy thông tin mô được sử dụng để mô tả mức độ hỗn loạn trong một tín hiệu lấy từ sự kiện ngẫu nhiên, nó giúp miêu tả độ thuần khiết của thông tin trong một tín hiệu, với thông tin là các thành phần không có sự ngẫu nhiên trong tín hiệu. Đây là khái niệm được Claude E. Shannon đưa ra vào năm 1948.

Entropy có phải thỏa mãn các điều kiện sau:

1. Entropy tỷ lệ thuận với xác suất xuất hiện các phần tử ngẫu nhiên trong tín hiệu. Thay đổi nhỏ trong xác suất cũng làm thay đổi nhỏ trong entropy.
2. Nếu các phần tử ngẫu nhiên đều có xác suất xuất hiện bằng nhau, việc tăng số lượng phần tử ngẫu nhiên phải làm tăng entropy.
3. Nếu có thể tạo các chuỗi tín hiệu theo nhiều bước thì entropy của cả tín hiệu phải bằng tổng entropy của từng bước.

Do phạm vi nghiên cứu của luận văn nên luận văn chỉ đề cập đến entropy rời rạc.

Công thức tính entropy rời rạc được Shannon định nghĩa như sau: Cho một hàm phân phối xác suất với biến giá trị rời rạc x, với tập giá trị của $x = \{x_1, x_2, ..., x_n\}$ với xác xuất tương ứng là $p_i = p(x = x_i)$ với $0 \leq pi \leq 1$, $\sum_{i=1}^{n} p_i = 1$. Khi đó entropy của phân phối này được là H_p với

$$H_p = \sum_{i=1}^{n} p_i \log_{10} p_i$$

2.2.3.2. Thuật toán ID3

Thuật toán ID3 được J. Ross Quinlan trình bày vào năm 1996 và được phân loại là một thuật toán học có giám sát. Thuật toán ID3 coi hàm mất mát khi xây dựng một cây quyết định là tổng entropy của các trọng số tại các lá. Các trọng số ở đây tỉ lệ với số điểm dữ liệu được

phân vào mỗi nốt, và mục tiêu của thuật toán là phải chọn cách xây dựng nào sao cho hàm mất mát phải là bé nhất. Để đạt được điều này thì tại mỗi bước phân chia, entropy phải giảm đi một lượng lớn nhất. ID3 sử dụng infomation gain để đánh giá mức độ mất của entropy tại mỗi bước và lựa chọn thuộc tính làm nốt tại mỗi bước.

Thuật toán ID3 được mô tả như sau:

Giả sử bài toàn có F thuộc tính khác nhau, tại một nốt không phải lá có các điểm dữ liệu tạo thành một tập S với số phần tử của tập |S|=N. Và trong N điểm dữ liệu này, có N_c (c = 0,1,2,....C) điểm thuộc lớp f ∈ F. Xác suất để điểm dữ liệu này rơi vào thuộc tính f là $\frac{N_c}{N}$. Và entropy tại điểm này sẽ được tính bằng.

$$H(S) = \sum_{c=1}^{C} \frac{N_c}{N} \log_{10}(\frac{N_c}{N})$$

Tiếp theo, giả sử thuộc tính được chọn là f ∈ F, dựa trên f ta phân các điểm dữ liệu trên tập S thành M nốt con $S_1, S_2,....,S_K$ với số điểm trong mỗi nốt con lần lượt là $m_1, m_2,...,m_K$. Ta gọi tổng trọng số entropy trong từng nốt là:

$$H(S, x) = \sum_{k=1}^{K} \frac{m_K}{N} H(S_k)$$

Ta định nghĩa *information gain* dựa trên thuộc tính f:

$$\text{Gain}(x, S) = \text{H}(S) - \text{H}(x, S)$$

Và trong ID3, tại mỗi nốt sẽ chọn thuộc tính có argmax(G(*x*,*S*)).

2.2.3.3. Thuật toán C4.5

Thuật toán C4.5 được đề xuất vào năm 1993 bởi Ross Quinlan để khắc phục những hạn chế của thuật toán ID3 trước đó. Nhược điểm của ID3 là dễ bị phụ thuộc vào các thuộc tính có số lượng dữ liệu lớn và bỏ qua các thuộc tính có số lượng dữ liệu bé nhưng ảnh hưởng lớn tới kết quả. Ngoài ra, ID3 còn dễ bị hiện tượng "*overfitting*", một hiện tượng khi mô hình huấn luyện quá khớp với trainning data, nhưng khi thử với testing data thì không phù hợp dẫn tới kết quả đúng không cao. Do đó, C4.5 có sử dụng một thước đo dữ liệu khác đó là "*gain ratio*" cộng thêm sử dụng một số kỹ thuật "cắt tỉa" để tránh "*overfitting*".

Gain ratio được định nghĩa như sau:

$$\text{GainRatio}(x, S) = \frac{Gain(x,S)}{SplitInfo(x,S)}$$

Trong đó, $SplitInfo(x, S)$ được tính như sau:

$$SplitInfo(x, S) = -\sum_{i=1}^{n} x'(\frac{i}{x}) \times \log_{10}(x'\left(\frac{i}{x}\right))$$

$x'\left(\frac{i}{x}\right)$ là tỷ lệ các phần tử xuất hiện ở lớp x.

Ngoài ra C4.5 cũng áp dụng thêm kỹ thuật cắt tỉa được gọi là "pruning", phương pháp này có thể áp dụng có bất cứ cây quyết định nào. Kỹ thuật này được diễn tả như sau: Sau khi xây dựng mọi điểm trong trainning data đều có phân lớp. Một số nốt con có chung một nốt sẽ được

cắt tỉa và nốt đó sẽ thành leaf-node, với phân lớp là lớp chiếm đa số sẽ được phân vào nốt đó. Trong luận văn này, kỹ thuật pruning sẽ được đưa vào mã nguồn của tất cả các thuật toán để đảm bảo tối ưu kết quả.

2.2.3.4. Thuật toán CART

Thuật toán CART (Classification and Regression Trees) là một thuật toán cho phép việc giải quyết bài toán cây kết quả phân loại dạng nhị phân rất hiệu quả. Thuật toán được Breiman và các đồng sự công bố vào năm 1984 cùng với một thước đo mới là "*Gini-index*" thước đo độ thuần khiết của thông tin. "Trong một quần thể, nếu chúng ta chọn ngẫu nhiên hai cá thể và chúng xác suất để chúng cùng lớp là 1 thì quần thể này được coi là thuần khiết"[4].

Gini-index được tính bằng công thức như sau:

$$Gini = 1 - \sum_{i=1}^{n}((P_i)^2)$$

Trong đó P_i là xác suất của kết quả nhị phân xuất hiện trong lớp. Sau đó ta tính tổng trong số Gini-index trong từng thuộc tính là lấy chọn thuộc tính có tổng trọng số bé nhất.

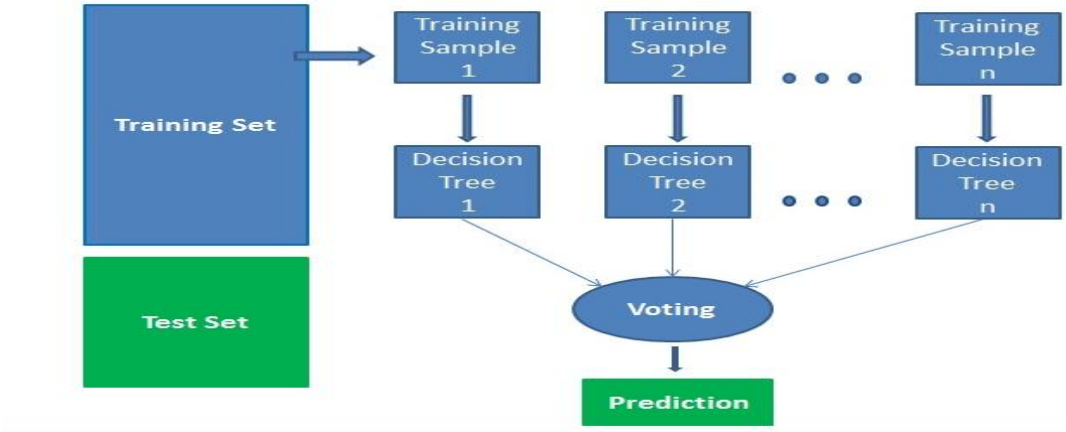
2.2.4. Thuật toán Random Forest

Trong các thuật toán trên, Random Forest là một thuật toán đặc biệt hơn với các thuật toán Decision Tree khác, do nó sử dụng một phương thức gọi "hộp đen", tức là ta đưa dữ liệu vào và đưa ra kết quả nhưng không thể giải thích được cơ chế hoạt động của mô hình. Random Forest được đề xuất bởi Tin Kam vào năm 1995.

Ý tưởng được mô tả như sau: thuật toán Random Forest sẽ sinh ra hàng trăm cây quyết định, trong đó mỗi cây sẽ được tạo ngẫu nhiên với các nốt là các câu hỏi về thuộc tính của dữ liệu, câu trả lời cuối sẽ ở nốt lá.

Để tạo ra một cây quyết định, thuật toán Random Forest làm như sau:

+ Chọn ra k thuộc tính ngẫu nhiên từ tập có m thuộc tính.
+ Từ tập k thuộc tính đó, xây dựng cây quyết định như các thuật toán trên. Thước đo thường sử dụng là "*information gain*".
+ Lập lại các bước 1-2 để tạo ra đủ số cây cần thiết.
+ Tiến hành bình chọn giữa hàng trăm cây mới sinh. Câu trả lời mà nhiều cây trả cùng đáp án thì được coi là câu trả lời đúng.



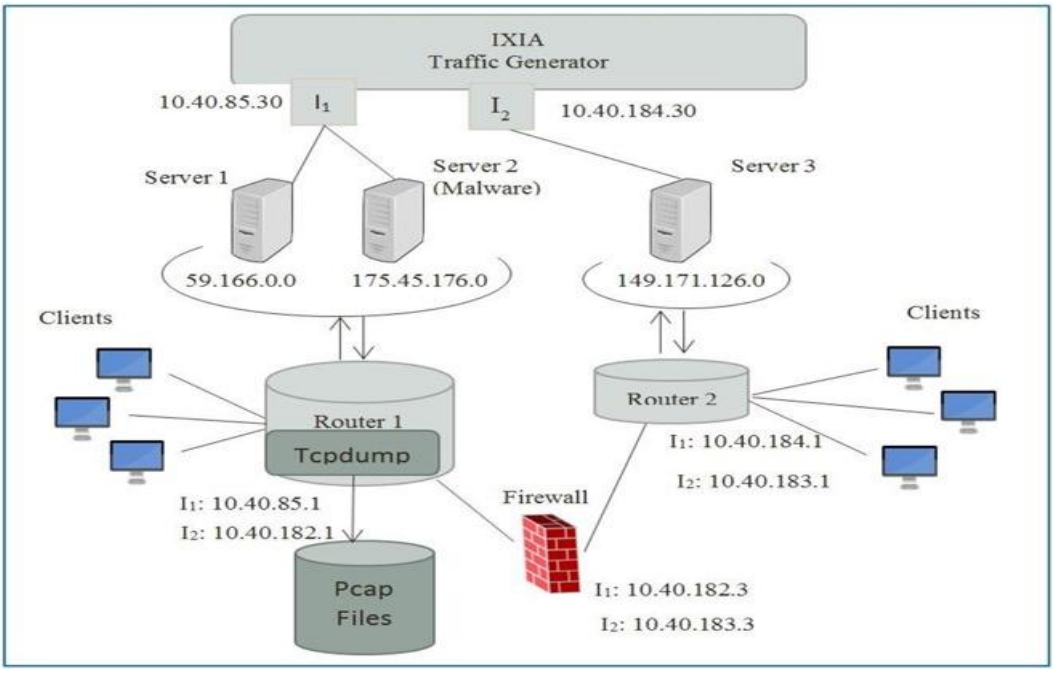
Hình 2.5. Mô hình thuật toán Random Forest

Để đảm bảo các mẫu thử không bị hiện tượng "overfiting", Random Forest sẽ ngẫu nhiên bỏ qua một số thuộc tính khi xây dựng cây. Nếu thuộc tính có *information gain* cao thứ m bị bỏ qua, thuộc tính có *information gain* cao thứ (m-1) sẽ chắc chắn được chọn. Đây gọi là kỹ thuật *"attribute sampling"*.

2.3. Giới thiệu về bộ dữ liệu UNSW-NB15

Bộ dữ liệu UNSW-NB15, được tạo vào năm 2015 và lần cập nhật cuối là 2018 bởi tiến sĩ Nour Moustafa và giáo sư Jill Slay thuộc đại học New South Wale tại Úc.

Phương pháp thu thập dữ liệu của bộ dữ liệu UNSW-NB15 là sử dụng trình tạo lưu lượng gồm 3 server ảo, trong đó 2 server phân tán lưu lượng truy cập bình thường và 1 server hình thành hoạt động bất thường/tấn công trong lưu lượng mạng. Tất cả lưu lượng mạng đều tới 1 router và được lưu lại bằng các file pcap.



Hình 2.6. Mô hình mô phỏng lưu lượng mạng của bộ dữ liệu unsw-nb15

Toàn bộ file pcap được thu và xử lý, phân loại ra và cuối cùng là các file csv với 49 thuộc tính. Dữ liệu sử dụng trong luận văn lấy từ tập dữ liệu unsw-nb15 gồm:

+ UNSW_NB15_training: chứa 175.341 bản ghi.
+ UNSW_NB15_testing: chứa 82.332 bản ghi.

Lý do luận văn chọn bộ dữ liệu này vì dữ liệu là dữ liệu về lưu lượng mạng mới nhất và có số lượng bản ghi lớn. Ngoài ra vì các bản ghi là các file csv nên sẽ dễ dàng hơn trong việc xử lý thông tin để huấn luyện thuật toán và đưa ra kết quả tốt.

2.4. Giới thiệu về bộ dữ liệu KDDCup99

Năm 1999, Stolfo đề xuất bộ dữ liệu KDD'99 (UCI KDD Archive, 1999) dựa trên các dữ liệu bắt được bởi chương trình đánh giá hệ thống phát hiện xâm nhập DARPA'98. Bộ dữ liệu này gồm gần 5 triệu bản ghi, mỗi bản ghi có 41 thuộc tính và được gán nhãn là bình thường hay các dạng tấn công đặc trưng. KDD'99 đã được sử dụng rộng rãi để đánh giá các kỹ thuật phát hiện bất thường. Các dạng tấn công được phân thành các nhóm như sau:

✓ *Tấn công từ chối dịch vụ (DoS)*

- ✓ *User to Root Attack (U2R)*
- ✓ *Remote to Local Attack (R2L)*
- ✓ *Probing Attack*

Một số chuyên gia cho rằng hầu hết các tấn công mới đều là biến thể của các tấn công đã biết và các dấu hiệu của các tấn công đã biết có thể đủ để nhận dạng các biến thể mới. Bộ dữ liệu huấn luyện KDD'99 bao gồm 24 loại tấn công khác nhau và có thêm 14 loại tấn công mới được thêm vào trong bộ dữ liệu kiểm tra. Dựa vào các đặc trưng tấn công có thể phân loại KDD'99 thành các nhóm chính như sau:

- *Đặc trưng cơ bản:* Gồm tất cả các thuộc tính có thể có từ các kết nối TCP/IP.
- *Đặc trưng lưu lượng:* Gồm các đặc trưng được tính toán với mối liên hệ với khoảng thời gian.
- *Đặc trưng same host:* Chỉ kiểm tra các kết nối trong khoảng thời gian dưới 2 giây có cùng host đích như kết nối hiện hành và thống kê liên quan đến các hành vi giao thức, dịch vụ, …
- *Đặc trưng same service:* Chỉ kiểm tra những kết nối trong khoảng thời gian dưới 2 giây có cùng dịch vụ như kết nối hiện hành.
- *Đặc trưng nội dung:* Khác với hầu hết tấn công DoS, Probing, R2L và U2R không có bất cứ một mẫu tấn công nào. Bởi vì DoS và Probing liên quan đến nhiều kết nối với một số host trong một khoảng thời gian rất ngắn, tuy nhiên tấn công R2L và U2R được nhúng trong đoạn gói dữ liệu và thường xuyên chỉ bao gồm một kết nối. Để phát hiện những loại tấn công này, cần một số đặc trưng để có thể tìm kiếm những hành vi nghi ngờ trong phần dữ liệu, chẳng hạn số lần cố gắng đăng nhập thất bại. Đây được gọi là đặc trưng nội dung.

Hai loại kể trên của đặc trưng lưu lượng được gọi dựa trên thời gian. Tuy nhiên, có một số tấn công thăm dò quét host (cổng) sử dụng khoảng thời gian lớn hơn 2 giây, có thể trong 1 phút. Kết quả là tấn công này không tạo ra các mẫu tấn công trong khoảng thời gian 2 giây.

- Bảng phân loại 24 loại tấn công trong KDDCup 99

Loại	Các tấn công trong bộ dữ liệu KDDCup 99
Probe	Ipsweep, Nmap, Portsweep, Satan
DoS	Back, Land, Neptune, Pod, Smurf, Teardrop
U2R	Buffer_overflow, Loadmodule, Perl, Rootkit
R2L	Ftp_write, Guess_passwd, Imap, Multihop, Phf, Spy, Warezclient, Warezmaster

CHƯƠNG 3. THỰC NGHIỆM VÀ KẾT QUẢ

3.1. Công nghệ áp dụng

Python là một ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng, do Guido van Rossum tạo ra và lần đầu ra mắt vào năm 1991, nó được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu, có hình thức, cấu trúc dễ hiểu cho người mới học lập trình.

Phiên bản luận văn sử dụng là là Python 3, do đây là phiên bản mới nhất của Python, có hỗ trợ các thư viện mà luận văn có sử dụng như: scikit-learn, numpy, pandas và matplotlib.

Cấu hình chi tiếp của hệ thống phục vụ thu thử nghiệm kết quả của luận văn gồm:

+ Phần mềm: hệ điều hành Windows 10, sử dụng Python bản 3.5.
+ Phần cứng: CPU i3 2328M, 4GB RAM, 120GB SSD.

3.2. Tiến hành xử lý dữ liệu

3.2.1. Các thuộc tính của bộ dữ liệu UNSW-NB15

Luận văn sẽ sử dụng bộ dữ liệu UNSW-NB15, đã được công bố và cho phép sử dụng miễn phí. Bộ dữ liệu gốc gồm 47 thuộc tính để đầu vào và 2 thuộc tính kết quả với các đầu vào. Các thuộc tính chi tiết được ghi tại bảng sau. Tuy nhiên, luận văn sử dụng một phần bộ dữ liệu đã được xử lý riêng cho phân tích, huấn luyện và kiểm thử và cũng được tải trực tiếp từ nguồn. Bộ dữ liệu này sử dụng 42 thuộc tính và 2 thuộc tính kết quả.

Tên thuộc tính	Kiểu dữ liệu	Miêu tả
srcip	nominal	Địa chỉ IP nguồn
sport	integer	Port nguồn
dstip	nominal	Địa chỉ IP đích
dsport	integer	Port đích
proto	nominal	Giao thức
state	nominal	Trạng thái và giao thức phụ thuộc
dur	Float	Thời gian
sbytes	Integer	Số bytes trao đổi từ nguồn tới đích
dbytes	Integer	Số bytes trả về từ đích tới nguồn
sttl	Integer	Thời gian tồn tại của byte dữ liệu từ nguồn tới đích
dttl	Integer	Thời gian tồn tại của byte dữ liệu từ đích tời nguồn
sloss	Integer	Số gói tin từ nguồn bị truyền lại hoặc bị mất
dloss	Integer	Số gói tin từ đích bị truyền lại hoặc bị mất
service	nominal	Tên dịch vụ sử dụng
sload	float	Số bit nguồn truyền mỗi giây
dload	float	Số bit đích trả mỗi giây
spkts	integer	Số gói tin từ nguồn
dpkts	integer	Số gói tin từ đích
swin	integer	Số gói tối đa được gửi từ nguồn
dwin	integer	Số gói tối đa được gửi từ đích
stcpb	integer	Sequence number của nguồn

dtcpb	integer	Sequence number của đích
smean	integer	Kích thước gói tin được truyền bởi scr
dmean	integer	Kích thước gói tin được truyền bởi dst
trans_depth	integer	Sử dụng giao thức yêu cầu/phản hồi của http hay không
res_bdy_len	integer	Kích thước thực của dữ liệu không nén truyền từ dịch vụ http của máy chủ
sjit	float	Giá trị Jitter của nguồn (ms)
djit	float	Giá trị Jitter của đích (ms)
stime	timestamp	Thời gian bắt đầu
ltime	timestamp	Thời gian kết thúc
sintpkt	float	Thòi gian đến từ nguồn
dintpkt	float	Thời gian đến từ đích
tcprtt	float	Thời gian khứ hồi được thiết lập
synack	float	Thời gian kết nối giữa gói SYN và gói SYN_ACK
ackdat	float	Thời gian kết nối giữa gói SYN_ACK và gói ACK
is_sm_ips_ports	binary	Nếu số cổng giống nhau thì giá trị ghi (1), nếu khác ghi (0)
ct_state_ttl	integer	Giá trị cụ thể cho thời gian tồn tại của gói tin
ct_flw_http_mthd	integer	Các phương thức GET và POST của giao thức http
is_ftp_login	binary	Phiên truy cập fpt được xác thực bởi tên đăng nhập và mật khẩu thì ghi (1), nếu khác ghi (0)
ct_ftp_cmd	integer	Số lệnh trong phiên truy cập fpt
ct_srv_src	integer	Số kết nối có cùng dịch vụ và địa chỉ nguồn trong 100 kết nối.
ct_srv_dst	integer	Số kết nối có cùng dịch vụ và địa chỉ đích trong 100 kết nối.
ct_dst_ltm	integer	Số kết nối của cùng một địa chỉ đích trong 100 kết nối.
ct_src_ ltm	integer	Số kết nối của cùng một địa chỉ nguồn trong 100 kết nối.
ct_src_dport_ltm	integer	Số kết nối của cùng một địa chỉ nguồn và cổng đích trong 100 kết nối.
ct_dst_sport_ltm	integer	Số kết nối của cùng một địa chỉ đích và cổng nguồn trong 100 kết nối.
ct_dst_src_ltm	integer	Số kết nối của cùng một nguồn và địa chỉ đích (3) trong 100 kết nối.
attack_cat	nominal	Tên loại giao thức tấn công, nếu không có ghi (Normal)
label	binary	Nếu bị tấn công ghi (1), còn lại ghi (0).

3.2.2. Các thuộc tính của bộ dữ liệu KDD99

Dưới sự bảo trợ của Cơ quan Quản lý Nghiên cứu Dự Án Phòng Thủ Tiên tiến thuộc Bộ Quốc phòng Mỹ (DARPA) và phòng thí nghiệm nghiên cứu không quân (AFRL), năm 1998 phòng thí nghiệm MIT Lincoln đã thu thập và phân phối bộ dữ liệu được coi là bộ dữ liệu tiêu chuẩn cho việc đánh giá các nghiên cứu trong hệ thống phát hiện xâm nhập mạng máy tính. Dữ liệu được sử dụng trong cuộc thi KDD cup 99 là một phiên bản của bộ dữ liệu DARPA 98. Tập dữ liệu đầy đủ của bộ KDD cup 99 chứa 4.898.431 dòng dữ liệu, đây là một khối lượng dữ liệu lớn. Trong nghiên cứu và thử nghiệm, tập dữ liệu 10% của bộ KDD cup 99 thường được lựa chọn. Tập 10% của bộ KDD 99 tuy là tập con nhưng nó mang đầy đủ dữ liệu cho các loại hình tấn công khác nhau, đầy đủ thông tin quan trọng để thử nghiệm.

Bảng sau đây cho thấy số mẫu của các kiểu tấn công xuất hiện trong 10% bộ dữ liệu KDD cup 99 và nhãn lớp của chúng.

Kiểu tấn công	**Số mẫu ban đầu**	**Nhãn lớp**
Back	2,203	DOS
land	21	DOS
Neptune	107,201	DOS
pod	264	DOS
smurf	280,790	DOS
teardrop	979	DOS
satan	1,589	PROBE
ipsweep	1,247	PROBE
nmap	231	PROBE
portsweep	1,040	PROBE
normal	97,277	NORMAL
Guess_passwd	53	R2L
ftp_write	8	R2L
imap	12	R2L
phf	4	R2L
multihop	7	R2L
warzemaster	20	R2L
warzclient	1,020	R2L
spy	2	R2L
Buffer_overflow	30	U2R
Loadmodule	9	U2R
perl	3	U2R
rootkit	10	U2R

Từ bảng trên, các kiểu tấn công khác nhau trong bộ dữ liệu được nhóm thành 5 loại (gán nhãn lớp) của bộ dữ liệu KDD cup'99 bao gồm:

1. Normal: dữ liệu thể hiện loại kết nối TCP/IP bình thường;

2. DoS (Denial of Service): dữ liệu thể hiện loại tấn công từ chối dịch vụ;

3. Probe: dữ liệu thể hiện loại tấn công thăm dò;

4. R2L (Remote to Local): dữ liệu thể hiện loại tấn công từ xa khi hacker cố gắng xâm nhập vào mạng hoặc các máy tính trong mạng;

5. U2R **(**User to Root): dữ liệu thể hiện loại tấn công chiếm quyền Root (quyền cao nhất) bằng việc leo thang đặc quyền từ quyền người dùng bình thường lên quyền Root.

Trong bộ dữ liệu KDD cup 99, với mỗi kết nối TCP/IP có 41 thuộc tính số và phi số được trích xuất. Đồng thời, mỗi kết nối được gán nhãn (thuộc tính 42) giúp phân biệt kết nối bình thường (Normal) và các tấn công. Các thuộc tính của bộ dữ liệu KDD cup 99 được mô tả chi tiết trong bảng dưới đây. Bảng thông tin chi tiết 41 thuộc tính của tập dữ liệu huấn luyện và kiểm tra trong KDD99.

TT	**Tên thuộc tính**	**Mô tả**	**Tính chất**	**Ví dụ**
1	Duration	Chiều dài (số giây) của kết nối.	Liên tục	0
2	Protocol_type	Loại giao thức, ví dụtcp, udp, vv..	Rời rạc	tcp
3	Service	Dịch vụ mạng trên các điểm đến ví dụ http,telnet, vv..	Rời rạc	http
4	Src_bytes	Số byte dữ liệu từ nguồn đến đích	Liên tục	SF
5	DTt_bytes	Số byte dữ liệu từ đích đến nguồn	Liên tục	181
6	Flag	Trạng thái bình thường hoặc lỗi của kết nối	Rời rạc	5450
7	Land	1 nếu kết nối là from/to cùng máy chủ/cổng; 0 nếu ngược lại	Rời rạc	0
8	Wrong_fragment	Số lượng đoạn "sai"	Liên tục	0
9	Urgent	Số gói tin khẩn cấp	Liên tục	0
10	Hot	Chỉ số "hot"	Liên tục	0
11	Num_failed_logins	Số lần đăng nhập không thành công	Liên tục	0
12	Logged_in	1 nếu đăng nhập thành công; 0 nếu ngược lại	Rời rạc	1
13	Num_compromised	Số lượng điều kiện thỏa hiệp	Liên tục	0
14	Root_shell	Bằng 1 nếu thu được root shell; 0 nếu ngược lại	Rời rạc	0
15	Su_attempted	Bằng 1nếu cố gắng thực hiện lệnh "su root"; 0 nếu ngược lại	Rời rạc	0
16	Num_root	Số lần truy cập quyền "root"	Liên tục	0
17	Num_file_creations	Số hoạt động tạo tập tin	Liên tục	0
18	Num_shells	Số lượng shell prompts	Liên tục	0
19	Num_access_files	Kiểm soát số lần truy cập file	Liên tục	0

TT	Tên thuộc tính	Mô tả	Tính chất	Ví dụ
20	Num_outbound_cmDT	Số lượng lệnh outbound trong 1 phiên ftp	Liên tục	0
21	Is_host_login	Bằng 1nếu đăng nhập thuộc về danh sách "máy chủ" đã biết, 0 nếu ngược lại	Rời rạc	0
22	Is_guest_login	Bằng 1 nếu đăng nhập là một tài khoản khách, 0 nếu ngược lại	Rời rạc	0
23	Count	Số lượng kết nối đến các máy chủ tương tự giống như các kết nối hiện hành trong 2 giây đã qua.	Liên tục	8
24	Serror_rate	Số % kết nối có lỗi "SYN"	Liên tục	8
25	Rerror_rate	Số % kết nối có lỗi"REJ"	Liên tục	0.00
26	Same_srv_rate	Số % các kết nối đến những dịch vụ tương tự	Liên tục	0.00
27	Diff_srv_rate	% kết nối với các dịch vụ khác nhau.	Liên tục	0.00
28	Srv_count	số kết nối đến cùng dịch vụ với kết nối hiện hành trong hai giây qua	Liên tục	0.00
29	Srv_serror_rate	% kết nối có lỗi "SYN" từ các dịch vụ	Liên tục	1.00
30	Srv_rerror_rate	% kết nối có lỗi "REJ" từ các dịch vụ.	Liên tục	0.00
31	Srv_diff_host_rate	Tỉ lệ % kết nối đến máy chủ khác nhau từ dịch vụ	Liên tục	0.00
32	DTt_host_count	Đếm các kết nối có cùng một đích đến.	Liên tục	9
33	DTt_host_srv_count	Đếm các kết nối có cùng 1host đích và sử dụng các dịch vụ tương tự.	Liên tục	9
34	DTt_host_same_srv_rate	% các kết nối có cùng 1host đích và sử dụng cácdịch vụ tương tự	Liên tục	1.00
35	DTt_host_diff_srv_rate	% các dịch vụ khác nhau trên các host hiện hành	Liên tục	0.00
36	DTt_host_same_src_ port_rate	% các kết nối đến các host hiện thời có cùng cổng src	Liên tục	0.11
37	DTt_host_srv_diff_host_rate	% các kết nối đến các dịch vụ tương tự đến từ các host khác nhau	Liên tục	0.00
38	DTt_host_serror_rate	% các kết nối đến các host hiện thời có một lỗi SO	Liên tục	0.00

TT	Tên thuộc tính	Mô tả	Tính chất	Ví dụ
39	DTt_host_srv_serror_rate	% các kết nối đến các host hiện hành và dịch vụ quy định rằng có một lỗi SO	Liên tục	0.00
40	DTt_host_rerror_rate	% các kết nối đến các host hiện thời có một lỗi RST	Liên tục	0.00
41	DTt_host_srv_rerror_rate	% các kết nối đến các máy chủ hiện hành và dịch vụ quy định rằng có một lỗi RST	Liên tục	0.00
42	Nhãn	Kết nối bình thường/tấn công	Tượng trưng	Normal

Ví dụ về một vài dòng dữ liệu trong bộ KDD cup 99:

0,tcp,http,SF,181,5450,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0.00,0.00,0.00,0.00,1.00,0.00,0.00,
9,9,1.00,0.00,0.11,0.00,0.00,0.00,0.00,0.00,normal.

0,icmp,ecr_i,SF,1032,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,511,511,0.00,0.00,0.00,0.00,1.00,0.00
,0.00,255,255,1.00,0.00,1.00,0.00,0.00,0.00,0.00,0.00,smurf.

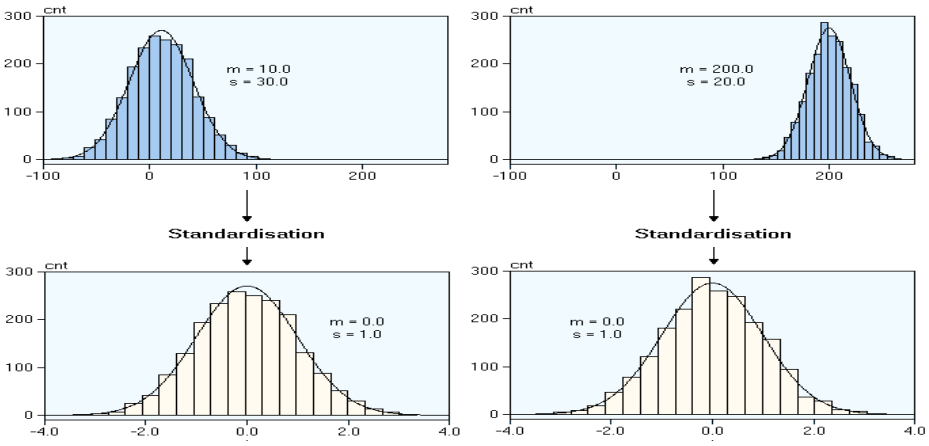
Một số chuyên gia phát hiện xâm nhập mạng cho rằng, hầu hết các loại tấn công mới là các biến thể của các loại tấn công đã biết và dấu hiệu của các loại tấn công đã biết có thể đủ để nắm bắt được các biến thể mới lạ.

Trong thực nghiệm, tôi chia tập dữ liệu thành 2 bộ training set và testing set theo tỷ lệ 7:3

3.2.3. Chuẩn hóa dữ liệu

Do phạm vi giá trị của dữ liệu đầu vào rất khác nhau, trong một số thuật toán học máy chẳng hạn như Decision Tree, các hàm mục tiêu sẽ không hoạt động đúng nếu không chuẩn hóa. Ví dụ, nhiều bộ phân loại tính toán khoảng cách giữa hai điểm dựa trên khoảng cách Euclide. Nếu một trong các đặc trưng có phạm vi giá trị rộng, khoảng cách mà bộ phân loại tính toán sẽ bị chi phối lớn hơn bởi đặc trưng này. Do đó, phạm vi của tất cả các đặc trưng nên được chuẩn hóa để mỗi đặc trưng đóng góp một vai trò tương đương nhau trong quá trình xây dựng bộ phân loại.

Một lý do khác khiến chuẩn hóa dữ liệu được áp dụng là việc giảm độ dốc của đạo hàm trong thuật toán gradient descent giúp việc hàm mất mát hội tụ nhanh hơn nhiều so với khi không áp dụng.



Hình 3.1. Minh họa chuẩn hóa dữ liệu

Do vậy, tôi tiến hành chuẩn hóa dữ liệu huấn luyện của các bộ dữ liệu KDD99 và UNSW-NB15 bằng thuật toán Standardization với công thức chuẩn hóa như sau:

$$x' = \frac{x - \mu}{\sigma}$$

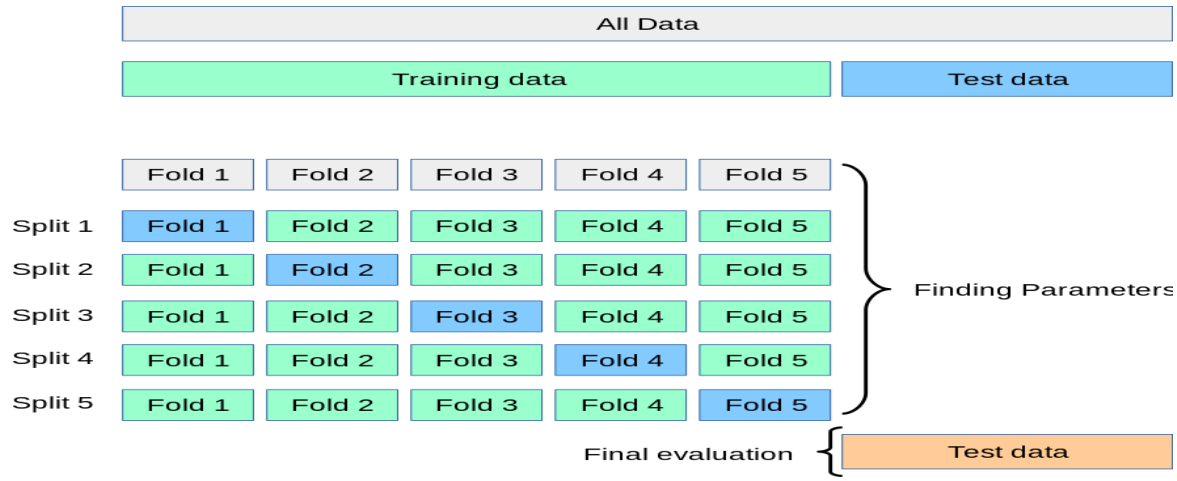
Trong đó, μ và σ lần lượt là kỳ vọng và phương sai (standard deviation) của thành phần đó trên toàn bộ training data.

3.2.4. Hyperparameter tuning và Cross-validation

Trong phương pháp đề xuất trong luận văn này, tôi tiến hành tìm siêu tham số cho thuật toán Decision Tree bằng bộ các siêu tham số như sau:

Siêu tham số	Tập giá trị
Splitter (chiến thuật để chia một đỉnh trong cây)	- best: chọn cách chia tốt nhất - random: chọn cách chia ngẫu nhiên tốt nhất
Max features (Số lượng đặc trưng được xem xét trong mỗi lần chia đỉnh)	- sqrt: căn bậc hai tổng số đặc trưng - log2: logarit cơ số 2 tổng số đặc trưng - None: sử dụng toàn bộ đặc trưng

Tất nhiên, để giữ cho thuật toán luôn luôn không nhìn thấy dữ liệu kiểm thử, một phần của bộ dữ liệu training sẽ được lấy ra không train mà để đánh giá các siêu tham số. Tôi sử dụng cross-validation. Theo đó, phương pháp này sẽ chia training set thành k phần. Sau đó, ta lần lượt sử dụng một phần làm validation set và k - 1 phần còn lại làm training set. Độ tốt của mô hình (lúc hyperparameter tuning) sẽ bằng trung bình cộng độ tốt trên validation set set qua k lần huấn luyện đó.



Hình 3.2. Minh họa phương pháp cross-validation

3.3. Tiêu chí đánh giá

Các tiêu chí sau được sử dụng cho việc đánh giá độ hiệu quả-chính xác của phương pháp đề xuất:

- Condition positive (P): số mẫu tấn công trong bộ dữ liệu.
- Condition negative (N): số mẫu bình thường trong bộ dữ liệu.
- True positive (TP): số mẫu tấn công được phân loại đúng là tấn công.
- True negative (TN): số mẫu bình thường được phân loại đúng là bình thường.

- False positive (FP): số mẫu bình thường bị gắn nhầm nhãn thành tấn công.
- False negative (FN): số mẫu tấn công bị gắn nhầm nhãn thành bình thường.

Các tiêu chí được sử dụng để đánh giá độ chính xác-hiệu quả của mô hình được xây dựng như sau:

- True positive rate (TPR) hay Sensitivity, Recall, Hit rate: Tỷ lệ số mẫu tấn công được dự đoán đúng trên tổng số các mẫu thực sự là tấn công. Tiêu chí cho thấy xác suất phát hiện tấn công của mô hình. Một mô hình có TPR cao đồng nghĩa với việc mô hình bỏ sót ít các mẫu thực sự là tấn công.

$$TPR = \frac{TP}{P} = \frac{TP}{TP + FN}$$

- False positive rate (FPR) hay Fall-out: Tỷ lệ số mẫu bình thường được dự đoán nhầm thành tấn công trên tổng số các mẫu bình thường. Tiêu chí cho thấy xác suất báo động sai của mô hình. Một mô hình có FPR thấp đồng nghĩa với việc mô hình ít khi báo động nhầm tấn công.

$$FPR = \frac{FP}{N} = \frac{FP}{FP + TN}$$

- Accuracy (ACC): Tỷ lệ các mẫu dự đoán đúng trên tổng các mẫu được dự đoán. ACC thể hiện độ hiệu quả của mô hình nói chung, tuy nhiên không đáng tin cậy trong các bộ dữ liệu không cân bằng.

$$ACC = \frac{TP + TN}{P + N} = \frac{TP + TN}{TP + TN + FP + FN}$$

- Precision hay positive predictive rate (PPV): Tỷ lệ số mẫu tấn công được dự đoán đúng trên tổng số các điểm được dự đoán là tấn công. PPV thể hiện độ chính xác của mô hình.
- F1-score: là trung bình cộng điều hòa (harmonic mean) của Precision và Recall. F1-score càng cao thể hiện bộ phân lớp càng tốt.

$$F_1 = 2\frac{Precision.Recall}{Precision + Recall}$$

- Area Under the Curve (AUC): Tiêu chí dựa trên đường ROC để đánh giá độ hiệu quả của mô hình. Đặc biệt AUC thường được dùng trong các bài toán phân lớp nhị phân với bộ dữ liệu không cân bằng.

3.4. Kết quả thực nghiệm và đánh giá

Thực nghiệm sử dụng nền tảng scikit-learn và tiến hành đánh giá phương pháp với 4 thuật toán dựa trên cây quyết định bao gồm:

- Decision Tree với Entropy
- Decision Tree với Gini
- Random Forest với Entropy
- Random Forest với Gini

3.4.1. Đối với bộ dữ liệu KDD99

- ***Bảng kết quả các thuật toán cây quyết định với bộ dữ liệu KDD99***

	Decision Tree (Entropy)	Decision Tree (Gini)	Random Forest (Entropy)	Random Forest (Gini)
Accuracy (%)	99.98	99.98	99.99	99.99
Precision (%)	99.98	99.98	99.99	99.99
Recall (%)	99.99	99.99	99.99	99.99
F1-score (%)	99.99	99.99	99.99	99.99
AUC (%)	99.98	99.96	99.98	99.98
TPR (%)	99.93	99.93	99.97	99.98
FPR (%)	0.01	0.01	0.01	0.01

Nhìn vào bảng kết quả trên, chúng ta có thể thấy thuật toán cây quyết định cho kết quả phân loại rất tốt, gần như tuyệt đối ở mọi tiêu chí. Trong đó thuật toán random forest với gini cho kết quả tốt nhất. Như đã được đề cập ở trên, các thuật toán cây quyết định luôn có nguy cơ overfitting. Tuy nhiên, phương pháp đề xuất đã sử dụng các phương pháp validation giúp hạn chế overfitting đến tối đa, do vậy có thể khẳng định thuật toán cây quyết định có hiệu quả cao trên tập dữ liệu KDD99.

Về thời gian thực hiện, các thuật toán huấn luyện khá nhanh. Kết quả được mô tả trong bảng sau:

- ***Bảng kết quả thời gian thực hiện với bộ dữ liệu KDD99***

	Decision Tree (Entropy)	Decision Tree (Gini)	Random Forest (Entropy)	Random Forest (Gini)
Thời gian chạy (giây)	12	16	8	15

3.4.2. Đối với bộ dữ liệu UNSW-NB15

- ***Bảng kết quả các thuật toán cây quyết định với bộ dữ liệu UNSW-NB15***

	Decision Tree (Entropy)	Decision Tree (Gini)	Random Forest (Entropy)	Random Forest (Gini)
Accuracy (%)	85.74	85.37	87.04	87.60
Precision (%)	92.21	93.14	96.62	95.48
Recall (%)	74.55	72.80	73.74	76.00
F1-score (%)	82.45	81.72	83.64	84.64
AUC (%)	84.71	84.21	85.82	86.53
TPR (%)	94.86	95.62	97.89	97.07
FPR (%)	25.45	27.20	26.26	24.0

Nhìn vào bảng kết quả trên, chúng ta có thể thấy thuật toán cây quyết định cho kết quả phân loại ở mức khá. Mặc dù có khả năng phát hiện tấn công tốt, nhưng tỷ lệ báo động giả khá lớn. Thuật toán cho kết quả tốt nhất vẫn là random forest với gini. Đánh giá về độ hiệu quả của thuật toán trên bộ dữ liệu UNSW-NB15 sẽ được trình bày ở phần tiếp theo.

Về thời gian thực hiện, các thuật toán huấn luyện nhanh hơn so với bộ dữ liệu KDD99 do có kích thước dữ liệu đầu vào nhỏ hơn. Kết quả được mô tả trong bảng sau:

- ***Bảng kết quả thời gian thực hiện với bộ dữ liệu UNSW-NB15***

	Decision Tree (Entropy)	Decision Tree (Gini)	Random Forest (Entropy)	Random Forest (Gini)

Thời gian chạy (giây)	13	11	8	6

3.4.3. Đánh giá

Như vậy, bằng thực nghiệm cho thấy thuật toán cây quyết định có hiệu quả tốt trên bộ dữ liệu KDD99. Tuy nhiên, độ chính xác cũng như hiệu quả chỉ ở mức khá tốt đối với bộ dữ liệu UNSW-NB15. Điều này có thể được giải thích như sau:

- Bộ dữ liệu KDD99 đã lỗi thời và không còn được khuyến nghị phân tích trong việc phát hiện các cuộc tấn công mạng. Thực tế, ngày nay bộ dữ liệu này đã không còn được ứng dụng rộng rãi vào thực tiễn an ninh mạng và được khuyến cáo thay thế bằng các bộ dữ liệu mới hơn. Tuy nhiên, KDD99 vẫn còn giá trị trong công tác nghiên cứu và giáo dục, do vậy luận văn vẫn tiến hành thực nghiệm trên bộ dữ liệu này.
- UNSW-NB15 được xây dựng từ năm 2015 nên đã được bổ sung nhiều loại tấn công mới so với bộ dữ liệu KDD99, do đó việc ứng dụng học máy trong phân tích, phát hiện tấn công là cần thiết, và đòi hỏi những thuật toán mạnh hơn nữa. Với một thuật toán cổ điển như cây quyết định, kết quả thu được là khá tích cực.
- Tỷ lệ số mẫu giữa training set và testing set trong bộ dữ liệu UNSW-NB15 khá nhỏ. Thông thường, tỷ lệ này nằm ở mức 7:3 với các bộ dữ liệu nhỏ và lớn hơn đối với các bộ dữ liệu lớn. Ngay nay, testing set thông thường chỉ nằm ở mức vài nghìn mẫu là đủ để đánh giá một mô hình. Tỷ lệ train:test khá cao trong bộ dữ liệu UNSW-NB15 là một nguyên nhân cho việc độ chính xác không cao. Điều này dễ dàng được khắc phục bằng việc tăng số lượng mẫu dành cho training set. Tuy nhiên, trong phạm vi luận văn, tôi sử dụng nguyên bản cách chia ban đầu của bộ dữ liệu UNSW-NB15

Như vậy, thuật toán cây quyết định nói riêng, hay học máy nói chung có khả năng phát hiện tấn công khá tốt khi thực nghiệm với bộ dữ liệu nổi tiếng KDD99 và UNSW-NB15. Điều này cho thấy tính khả thi và hứa hẹn về việc áp dụng rộng rãi các mô hình IDS dựa trên hành vi và học máy nhằm phát hiện các cuộc tấn công mạng.

Bên cạnh đó, kết quả thực nghiệm còn cho thấy và khẳng định, thuật toán random forest nói riêng hay các thuật toán tập hợp nói chung thường cho kết quả tốt hơn là các mô hình riêng lẻ.

KẾT LUẬN VÀ KIẾN NGHỊ

Cách mạng 4.0 đã kéo theo sự phát triển của các thiết bị mạng, thiết bị cảm biến. Nhưng sự phát triển của công nghệ quá nhanh mà không có sự quan tâm đến vấn đề bảo mật khiến những thiết bị này trở thành mục tiêu dễ dàng cho các hình thức tấn công mạng. Và nhưng hậu quả của việc tấn công có thể trở nên rất lớn nếu thiết bị tấn công có chứa thông tin nhậy cảm. Do đó việc xây dựng một biện pháp bảo vệ các thiết bị mạng là rất cần thiết. Mô hình này phải dể vận dụng kể cả trên các thiết bị có dung lượng nhỏ như thiết bị IOT.

Trong luận văn đã đề xuất mô hình để tiến hành dự đoán các hành vi tấn công mạng dựa trên lưu lượng bằng các thuật toán machine learning, cụ thể là decision tree. Luận văn đã đạt được một số kết quả như sau:

+ Nghiên cứu về bài toán phát hiện hành vi tấn công dựa trên lưu lượng mạng.
+ Đề xuất mô hình dự đoán hành vi tấn công dựa trên thuật toán học máy (decision tree) và xây dựng được mô hình học máy thành công.
+ Tiến hành nghiên cứu bộ dữ liệu về lưu lượng mạng kdd99 và unsw-nb15.

+ So sánh được tỷ lệ phát hiện của các thuật toán cây quyết định.

Phương hướng nghiên cứu tiếp theo của luận văn:

+ Xây dựng mô hình bằng ngôn ngữ nhúng như C.
+ Tích hợp được mô hình vào các thiết bị mạng nhỏ và vừa, đặc biệt là thiết bị IOT.

  Giám sát, thu thập dữ liệu dữ liệu để tiếp tục hoàn thiện mô hình.