BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

---

# BÁO CÁO
# NGHIÊN CỨU KHOA HỌC SINH VIÊN 2020

***Đề tài*:**

## PHƯƠNG PHÁP XÂY DỰNG ĐẶC TRƯNG MỘT SỐ TẤN CÔNG MẠNG

**Mã đề tài: 16-SV-2020-TH2**


**Giáo viên hướng dẫn : TS. NGUYỄN HỒNG SƠN**
**Sinh viên thực hiện : VÕ MINH THUẬN**
**Mã số sinh viên : N17DCAT070**
**Lớp : D17CQAT01-N**


**TP. HỒ CHÍ MINH – 2020**

# MỤC LỤC

## DANH MỤC BẢNG



## DANH MỤC HÌNH

## DANH MỤC VIẾT TẮT

| | |
|---|---|
| IDS: Intrusion Detection System | Hệ thống phát hiện xâm nhập |
| DDoS: Distributed Denial of Service | Tấn công từ chối dịch vụ phân tán |

# MỞ ĐẦU

Với sự phát triển công nghệ thông tin như hiện nay - thời đại 4.0, các doanh nghiệp trên thế giới đang dần chuyển giao từ mô hình kinh doanh thương mại truyền thống sang thương mại điện tử. Do đó, chúng ta sẽ không thể trách khỏi những cuộc tấn công mạng có quy mô lớn đến từ các nhóm tin tặc, đối thủ cạnh tranh không lành mạnh có chủ đích như phá hoại, thu thập thông tin nhạy cảm, đánh cắp thông tin khách hàng từ các doanh nghiệp.

Chính vì lẽ đó, công việc phát hiện và ngăn chặn các hình thức tấn công mạng là rất cần thiết. Hệ thống phát hiện xâm nhập (IDS) được đề xuất, để đảm nhận nhiệm vụ phát hiện, cảnh báo (thậm chí ngăn chặn) tới quản trị viên hệ thống. Ngày nay, IDS được pháp triển và được áp dụng các kỹ thuật máy học (Machine Learing), học sâu (Deep Learning) - phát triển mạnh mẽ trong nhiều lĩnh vực [4] như hệ thống tài chính ngân hàng, y tế, hàng không - hỗ trợ cách thức tiếp cận, nhận dạng, phân loại tấn công mạng.

Nhưng bên cạnh, việc sử dụng toàn bộ dataset cho quá trình tập huấn cũng như là kiểm tra sẽ mất nhiều thời gian, tài nguyên hệ thống, tăng độ phức tạp của thuật toán. Chính vì lẽ đó, bài nghiên cứu này tôi sẽ đề xuất phương pháp trích xuất các thuộc tính có tầm quan trọng và phù hợp từng loại tấn công tối ưu hóa bộ dữ liệu cũng như giảm thời gian tính toán. Nói cách khác, đặc trưng hóa từng loại tấn công, tạo tiền đề cho các bài nghiên cứu có liên quan đến phát triển hệ thống phát hiện xâm nhập mạng (IDS).

# CHƯƠNG 1: TỔNG QUAN

## 1.1 SƠ LƯỢC

Ngày nay việc sử dụng mạng Internet ngày càng tăng dẫn đến nhiều mối nguy hiểm tiềm tàn và các cuộc tấn công mới diễn ra hằng ngày. Để phát hiện những hành động bất thường hoặc lạm dụng internet, Hệ thống phát hiện xâm nhập (IDS) đã được đề xuất như là một thành phần quan trọng trong mạng an toàn[1]. Machine-Learning đã rất hữu ích trong việc phát hiện những hành vi bất thường của người dùng mạng[2]. Việc trích xuất những đặc trưng là hết sức quan trọng cho quá trình nhận dạng, phân loại tấn công. Do đó trong nghiên cứu này, tôi sẽ đề xuất phương pháp trích xuất bộ con gồm các đặc trưng tối ưu cho từng cuộc tấn công phổ biến từ một dataset. Với phạm vi nghiên cứu, tôi sẽ thực nghiệm trên bộ dataset CICIDS 2017 [3], và thuật toán Random Forest.

## 1.2 LỊCH SỬ NGHIÊN CỨU

Hệ thống phát hiện xâm nhập (IDS) đầu tiền từng được ghi nhận được dựa trên bài nghiên cứu được tiến hành bởi Dorothy E.Denning[5] nó cho ta hướng giải quyết được biết đến như là một hệ thống chuyên gia phát hiện xâm nhập. Để có thể phát hiện được những loại xâm nhập từng được biết, nó triển khai phương pháp tiếp cận kép, sử dụng một hệ thống chuyên gia dựa trên các nguyên tắc. Bên cạnh đó, nó tận dụng một thành phần phát hiện bất thường thống kê mà có được dựa trên hệ thống máy chủ, những hồ sơ mô tả người dùng, những hệ thống mục tiêu. Sau này, một phiên bản mới được biết đến như là một hệ thống chuyên gia phát hiện xâm nhập thế hệ tiếp theo được công bố vào năm 1995 [6] với cùng nhóm nghiên cứu trên. Khái niệm của việc tận dụng phát hiện bất thường trong lĩnh vực an toàn thông tin trở nên xu hướng chủ đạo với nhiều bài nghiên cứu

Hossen et al [7] đã khai thác hiệu suất của hệ thống phát hiện xâm nhập mạng, nó có thể phát hiện đa dạng các loại tấn công mạng bằng việc sử dụng thuật toán Deep Reinforcement Learning. Tác giả làm việc trên 85 thuộc tính của CICIDS 2017 mà nó đã hỗ trợ như một phương tiện hữu ích trong việc phát hiện các loại tấn công khác nhau.

Theo Sharafaldin et al[8], thuật toán Random Forest được xem xét có khả năng phân loại tốt nhất cho mỗi tấn công đối với bộ dataset CICIDS 2017. Tác giả đã thực

hiện đánh giá trên nhiều thuật toán học máy khác nhau như cây quyết định (ID3), Naive-Bayes, k-Nearest Neighbor (KNN) rồi đưa ra kết luận.

Kostas [9] đã thực hiện việc phát hiện mạng bất thường bằng việc sử dụng thuật toán Machine Learning, và CICIDS 2017 được sử dụng vì bao gồm những đa dạng các loại tấn công được cập nhật. Chọn lựa thuộc tính được thực hiện bằng thuật toán Random Forest Regressor. Bảy thuật toán Machine Learning cũng được áp dụng để đạt được hiệu suất tốt nhất.

Và trong bài nghiên cứu này tôi, sẽ áp dụng thuật toán Random Forest Regressor để chọn ra bộ dữ liệu con tối ưu, và Random Forest để kiểm tra bộ chính xác của bộ dữ liệu này cho từng tấn công mạng

## 1.3 MỤC TIÊU

- Tìm hiểu các đặc trưng của từng loại tấn công mạng
- Xây dựng phương pháp đặc trưng hóa các loại tấn công mạng điển hình

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## 2.1 Các hình thức tấn công phổ biến

### 2.1.1 DDoS

Thường xảy ra khi nhiều hệ thống, làm ngập băng thông hoặc tài nguyên của máy nạn nhân. Những tấn công như vậy thường là kết quả nhiều hệ thông bị xâm phạm (ví dụ botnet) bằng việc làm ngập máy mục tiêu với việc tao ra một lưu lượng khổng lồ.

### 2.1.2 Tấn công web - Brute Force

Đây là một trong những tấn công phổ biến nhất, nó không chỉ được biết biết đến chỉ để phá vỡ mật khẩu mà còn có thể tìm thấy những trang và nội dung được ẩn trong ứng dụng web. Brute Force cơ bản là tấn công thử sai cho đến khi đạt được quyền xâm nhập vào hệ thống

### 2.1.3 PortScan

Đây là một kỹ thuật mà kẻ tấn công có thể thực hiện kiểm tra các port đang mở trên hệ thống nạn, để mà có những phương pháp tấn công dự tính đến máy chủ.

## 2.2 Tổng quan về bộ dữ Dataset CICIDS 2017

Trong phạm vi nghiên cứu này sẽ sử dụng file MachineLearingCSV, là một phần trong CICIDS-2017[3]

❖ Danh sách những thuộc tính được trích xuất và mô tả tương ứng:

| Tên thuộc tính | Mô tả |
|---|---|
| Flow duration | Thời gian dòng, đơn vị tính Microsecond |
| total Fwd Packet | Tổng các gói gửi |
| total Bwd packets | Tổng các gói được nhận |
| total Length of Fwd Packet | Tổng độ dài các gói gửi đi (byte) |
| total Length of Bwd Packet | Tổng độ dài các gói nhận |
| Fwd Packet Length Min | Kích thước nhỏ nhất của gói gửi đi |
| Fwd Packet Length Max | Kích thước lớn nhất của gói gửi |
| Fwd Packet Length Mean | Kích thước trung bình của gói gửi |
| Fwd Packet Length Std | Độ lệch chuẩn của độ dài gói gửi |
| Bwd Packet Length Min | Kích thước nhỏ nhất của gói được nhận |
| Bwd Packet Length Max | Kích thước lớn nhất của gói được nhận |

| | |
|---|---|
| Bwd Packet Length Mean | Kích thước trung bình của gói được nhận |
| Bwd Packet Length Std | Độ lệch chuẩn của độ dài gói được nhận |
| Flow Bytes/s | Số byte được gửi trên mỗi giây |
| Flow Packets/s | Số gói được gửi trên mỗi giây |
| Flow IAT Mean | Thời gian trung bình giữa 2 gói được chuyển tiếp trong một dòng |
| Flow IAT Std | Độ lệch chuẩn của thời gian giữa 2 gói được chuyển tiếp trong một dòng |
| Flow IAT Max | Thời gian tối đa giữa 2 gói được chuyển tiếp trong một dòng |
| Flow IAT Min | Thời gian nhỏ nhất giữa 2 gói được chuyển tiếp trong một dòng |
| Fwd IAT Min | Thời gian tối thiểu giữa 2 gói được gửi đi |
| Fwd IAT Max | Thời gian tối đa giữa 2 gói được gửi đi |
| Fwd IAT Mean | Thời gian trung bình giữa 2 gói được gửi đi |
| Fwd IAT Std | Độ lệch chuẩn của thời gian giữa 2 gói gửi |
| Fwd IAT Total | Tổng thời gian giữa 2 gói được gửi đi |
| Bwd IAT Min | Thời gian tối thiểu giữa 2 gói được nhận |
| Bwd IAT Max | Thời gian tối đa giữa 2 gói được nhận |
| Bwd IAT Mean | Thời gian trung bình giữa 2 gói được nhận |
| Bwd IAT Std | Độ lệch chuẩn của thời gian giữa 2 gói nhận |
| Bwd IAT Total | Tổng thời gian giữa 2 gói được nhận |
| Fwd PSH flags | Số lần cờ PSH được đặt trong gói gửi đi (0 cho UDP) |
| Bwd PSH Flags | Số lần cờ PSH được đặt trong gói được nhận (0 cho UDP) |
| Fwd URG Flags | Số lần cờ URG được đặt trong gói được gửi (0 for UDP) |
| Bwd URG Flags | Số lần cờ URG được đặt trong gói được nhận (0 cho UDP) |
| Fwd Header Length | Tổng số byte được sử dụng trong các header được gửi đi |

| | |
|---|---|
| Bwd Header Length | Tổng số byte được sử dụng trong các header được nhận |
| FWD Packets/s | Số gói được gửi đi trong 1 giây |
| Bwd Packets/s | Số gói được nhận trong 1 giây |
| Packet Length Min | Độ dài tối thiểu của một gói |
| Packet Length Max | Độ dài tối đa của một gói |
| Packet Length Mean | Trung bình độ dài của một gói |
| Packet Length Std | Độ lệch chuẩn tối đa của một gói |
| Packet Length Variance | Phương sai của một gói |
| FIN Flag Count | Số lượng gói FIN |
| SYN Flag Count | Số lượng gói SYN |
| RST Flag Count | Số lượng gói RST |
| PSH Flag Count | Số lượng gói PUSH |
| ACK Flag Count | Số lượng gói ACK |
| URG Flag Count | Số lượng gói URG |
| CWR Flag Count | Số lượng gói CWR |
| ECE Flag Count | Số lượng gói ECE |
| down/Up Ratio | Tỉ số giữa download và upload |
| Average Packet Size | Trung bình độ dài của một gói |
| Fwd Segment Size Avg | Kích thước trung bình quan sát được gửi đi |
| Bwd Segment Size Avg | Kích thước trung bình quan sát được nhận |
| Fwd Bytes/Bulk Avg | Trung bình tỉ lệ chịu tải số lượng lớn các byte gửi đi |
| Fwd Packet/Bulk Avg | Trung bình tỉ lệ chịu tải số lượng lớn các gói gửi đi |
| Fwd Bulk Rate Avg | Trung bình tỉ lệ Bulk gửi đi |
| Bwd Bytes/Bulk Avg | Trung bình tỉ lệ chịu tải số lượng lớn các byte nhận được |
| Bwd Packet/Bulk Avg | Trung bình tỉ lệ chịu tải số lượng lớn các gói nhận được |
| Bwd Bulk Rate Avg | Trung bình tỉ lệ Bulk nhận được |
| Subflow Fwd Packets | Số gói trung bình trong một luồng con được gửi đi |

| | |
|---|---|
| Subflow Fwd Bytes | Số byte trung bình trong một luồng con được gửi đi |
| Subflow Bwd Packets | Số gói trung bình trong một luồng con được nhận |
| Subflow Bwd Bytes | Số byte trung bình trong một luồng con được nhận |
| Fwd Init Win bytes | Tổng số byte được gửi đi trong cửa sổ ban đầu |
| Bwd Init Win bytes | Tổng số byte được nhận trong cửa sổ ban đầu |
| Fwd Act Data Pkts | Số lượng gói có ít nhất 1 byte tải trọng dữ liệu TCP theo hướng gửi |
| Fwd Seg Size Min | Kích thước phân đoạn tối thiểu được quan sát theo hướng gửi |
| Active Min | Thời gian tối thiểu luồng hoạt động trước khi không hoạt động |
| Active Mean | Thời gian trung bình một luồng hoạt động trước khi không hoạt động |
| Active Max | Thời gian tối đa luồng hoạt động trước khi không hoạt động |
| Active Std | Thời gian độ lệch chuẩn mà một luồng hoạt động trước khi trở nên nhàn rỗi |
| Idle Min | Thời gian tối thiểu luồng không hoạt động trước khi hoạt động |
| Idle Mean | Thời gian trung bình một luồng không hoạt động trước khi hoạt động trở lại |
| Idle Max | Thời gian tối đa luồng không hoạt động trước khi hoạt động |
| Idle Std | Độ lệch chuẩn thời gian luồng không hoạt động trước khi hoạt động |

**Bảng 2.1: Lịch trình thực nghiệm tấn công trên dataset CICIDS 2017**

| Ngày thực hiện | Pcap File size | Thời gian | CSV File Size | Tên tấn công | Số flow |
|---|---|---|---|---|---|
| Thứ 2 | 10 GB | Nguyên ngày | 257 MB | Không tấn công | 529918 |
| Thứ 3 | 10 GB | Nguyên ngày | 166 MB | FT -Patator, SSH-Patator | 445909 |
| Thứ 4 | 12 GB | Nguyên ngày | 272 MB | Dos Hulk, DosGoldenEye, DOSslowloris, DosSlowhttptest,Heartbleed | 692703 |
| Thứ 5 | 7.7 GB | Sáng | 87.7 MB | Web Attacks(Brute Force, XSS, SQL Injection) | 170366 |
| | | Chiều | 103 MB | Infiltration | 288602 |
| Thứ 6 | 8.2 GB | Sáng | 71.8 MB | Bot | 192033 |
| | | Chiều | 92.7 MB | Ddos | 225745 |
| | | Chiều | 97.1 MB | Port Scan | 286467 |

## 2.3 Gini index (Độ đo về độ không tinh khiết của thông tin)

**Gini index[10], cũng được biết với tên Gini impurity, nó tính toán lượng khả năng của một thuộc tính cụ thể mà được phân loại một cách không chính xác khi được chọn một cách ngẫu nhiên. Nếu tất cả các phần tử được liên kết với một lớp duy nhất thì nó có thể gọi là nguyên chất.**

**Chúng ta hãy hiểu tiêu chí của Chỉ số Gini, giống như các thuộc tính của entropy, chỉ số Gini thay đổi giữa các giá trị 0 và 1, trong đó 0 thể hiện độ tinh khiết của phân loại, tức là tất cả các phần tử thuộc về một lớp được chỉ định hoặc chỉ một lớp tồn tại ở đó. Và 1 cho biết sự phân bố ngẫu nhiên của các phần tử**

**trên các lớp khác nhau. Giá trị 0,5 của Chỉ số Gini cho thấy sự phân bố đồng đều của các phần tử trên một số lớp.**

**Gini index được xác định bằng cách trừ đi tổng bình phương của xác suất của mỗi lớp với một, về mặt toán học, Chỉ số Gini có thể được biểu thị như sau:**

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

Gini Index Formula

## 2.4 Thuật toán Random Forest

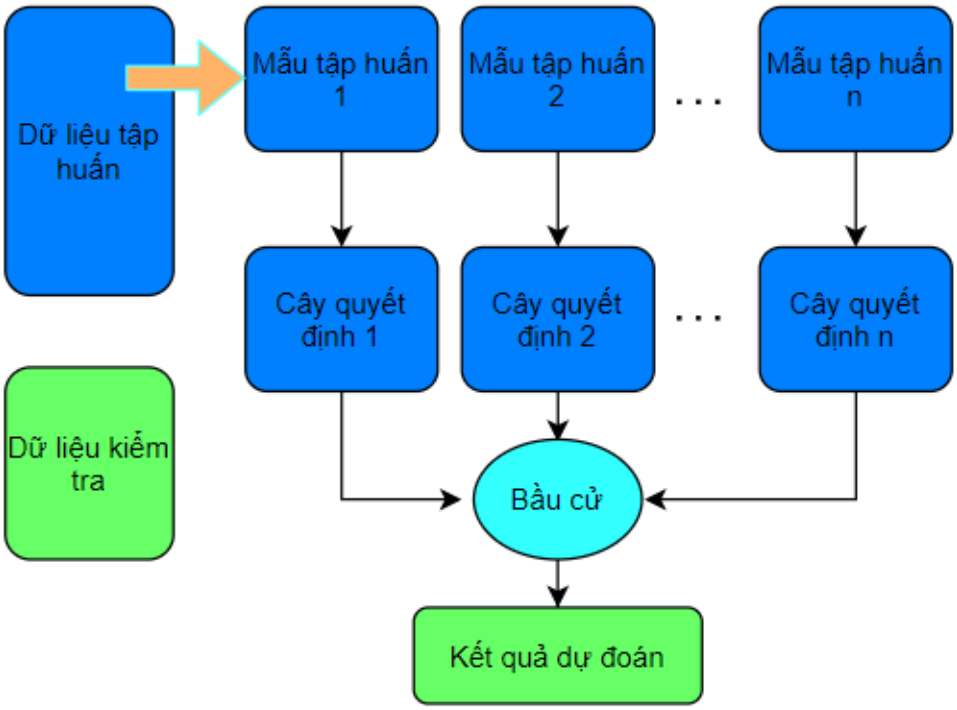
Random Forests [11] là thuật toán học có giám sát (supervised learning).Gồm tập hợp nhiều cây quyết định(Decision Tree), trong đó mỗi Decision Tree được tạo nên ngẫu nhiên từ việc tái chọn mẫu (chọn random 1 phần của data để xây dựng) và random các biến từ toàn bộ các biến trong trong dữ liệu. Với cơ chế như vậy, Random Forest cho ta một kết quả chính xác cao nhưng đánh đổi bằng việc ta không thể hiểu cơ chế hoạt động của thuật toán này do cấu trúc quá phức tạp của mô hình này — do vậy thuật toán này là một trong những phương thức Black Box — tức ta sẽ bỏ tay vào bên trong và rút ra được kết quả chứ không thể giải thích được cơ chế hoạt động của mô hình.

Random Forest là một phương pháp Supervised Learning do vậy có thể xử lý được các bài toán về Classification (phân loại) và Regression (dự báo về các giá trị)

*Thuật toán hoạt động như thế nào?*

Nó hoạt động theo bốn bước:

- Chọn các mẫu ngẫu nhiên từ tập dữ liệu đã cho.
- Thiết lập cây quyết định cho từng mẫu và nhận kết quả dự đoán từ mỗi quyết định cây.
- Hãy bỏ phiếu cho mỗi kết quả dự đoán.
- Chọn kết quả được dự đoán nhiều nhất là dự đoán cuối cùng



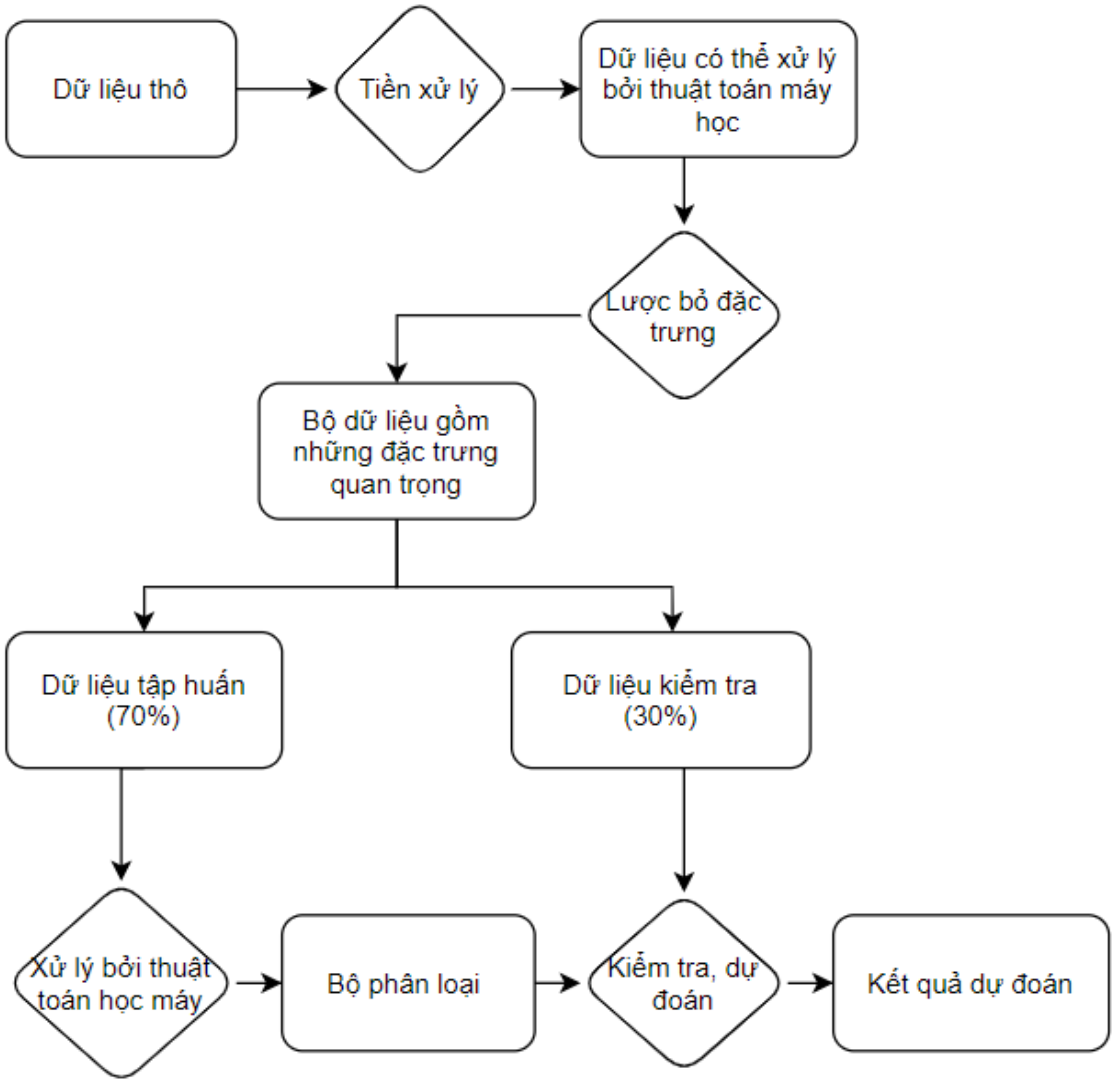

***Hình 2.1: Tổng quan cách hoạt động của RandomForest***

Ưu điểm: Random forests được coi là một phương pháp chính xác và mạnh mẽ hiệu, hiệu suất cao vì số cây quyết định tham gia vào quá trình này. Thuật toán có thể được sử dụng trong cả hai vấn đề phân loại và hồi quy. Random forests cũng có thể xử lý các giá trị còn thiếu. Có hai cách để xử lý các giá trị này: sử dụng các giá trị trung bình để thay thế các biến liên tục và tính toán mức trung bình gần kề của các giá trị bị thiếu. Bạn có thể nhận được tầm quan trọng của tính năng tương đối, giúp chọn các tính năng đóng góp nhiều nhất cho trình phân loại.

Nhược điểm: Random forests chậm tạo dự đoán bởi vì nó có nhiều cây quyết định. Bất cứ khi nào nó đưa ra dự đoán, tất cả các cây trong rừng phải đưa ra dự đoán cho cùng một đầu vào cho trước và sau đó thực hiện bỏ phiếu trên đó. Toàn bộ quá trình này tốn thời gian. Mô hình khó hiểu hơn so với cây quyết định, nơi bạn có thể dễ dàng đưa ra quyết định bằng cách đi theo đường dẫn trong cây.

# CHƯƠNG 3: PHƯƠNG PHÁP XÂY DỰNG ĐẶC TRƯNG

## 3.1 Mô hình thực hiện




***Hình 3.1: Sơ đồ quá trình thực hiện phân loại, đánh giá bộ dữ liệu***

## 3.2 Mô tả chi tiết

Quá trình thực hiện trích xuất bộ dữ liệu cho từng loại tấn công mạng, sẽ trải qua 5 giai, theo sau là mô tả từng giai đoạn:

- Giai đoạn 1: Tiền xử lý bộ dữ liệu

Các bộ dữ liệu được nhận về thường không hoàn thiện hoàn toàn, do có khả năng cột chứa thành phần giá trị null, hoặc bỏ trống, sẽ dẫn đến thuật toán không hoạt động,

hoặc khả năng tính toán không chính xác. Vì vậy bộ dữ liệu cần phải được điều chỉnh lại, để phù hợp với thuật toán cục thể

- Giai đoạn 2: Trích xuất bộ thuộc tính

Với việc dùng toàn bộ các thuộc tính có trong bộ dataset, để nhận biết dạng các hình thức tấn công là chúng ta sẽ gặp nhiều khó khăn trong việc triển khai các thuật toán Machine Learning, về mặt thời gian thực hiện cũng như độ phức tạp của việc tính toán. Theo sau là nhưng bước tiền xử lý đối với bộ dữ liệu để lọc ra những thuộc tính có mức độ quan trọng trong khả năng phát hiện, phân loại tấn công.

- Giai đoạn 3: Thực hiện tạo bộ dữ liêu tập huấn và kiểm tra

Trong suốt quá trình xử lý máy học, dữ liệu cần phải được chia thành 2 phần, phần thứ nhất sẽ dùng để tập huấn với thuật toán (được chia 70% so với dữ liệu gốc, do nếu bộ tập huấn có số bảng ghi nhỏ hơn bộ tập huấn thì sẽ ảnh hưởng đến độ chính xác của thuật toán phân loại), phần thứ 2 sẽ được dùng để kiểm tra, để đánh giá độ chính xác của thuật toán sau khi đã được tập huấn

- Giai đoạn 4: Triển khai thuật toán học máy

Thực hiện sử dụng mô hình máy học để đánh giá độ quan trọng của từng đặc trưng có trong bộ dữ liệu, và chọn ra 4 đặc trưng có độ quan trọng cao nhất[8], để gán bộ đặc trưng này cho tấn công mạng tương ứng

- Giai đoạn 5: Đánh giá bộ dữ liệu

Giai đoạn cuối cùng, chúng ta sẽ tiếp hành đánh giá các chỉ số đo cần thiết đối với bộ dữ liệu con có được từ giai đoạn 4, để đưa ra kết quả cuối cùng. Các thông số đánh giá hiệu suất[12] như sau:

Accuracy: Ti số giữa dữ liệu được phân loại đúng trong toàn bộ dữ liệu

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

Recall: Tỉ số dữ liệu được phân loại là tấn công trên toàn bộ dữ liệu tấn công

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

Precision: Tỉ số của dữ liệu được phân loại là tấn công đúng trên toàn bộ dữ được phân loại là tấn công.

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

F-measure:

$$\text{F- measure} = \frac{2}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}}$$

Với:
- TP: Dương tính thật, loại tấn công được nhận dạng đúng
- FP: Dương tính giả, dữ liệu lành tính được nhận dạng là tấn công
- TN: Âm tính thật, dữ liệu lành tính được nhận dạng là lành tính
- FN: Âm tính giả, dữ liệu tấn công được nhận dạng là lành tính

# CHƯƠNG 4: THỰC NGHIỆM XÂY DỰNG ĐẶC TRƯNG TẤN CÔNG MẠNG

## 4.1 Phần mềm sử dụng

- Ngôn ngữ lập trình: Python 3
- Thư viện: Sklearn, Pandas, Matplotlib, NumPy

## 4.2 Phần cứng sử dụng

- CPU: Intel(R) Core(TM) i7-3630QM CPU @ 2.40GHz (8 CPUs), ~2.4GHz
- RAM: 8GB DDR4
- Hệ điều hành: Windows 10 pro 64-bit
- GPU: NVIDIA GEFORCE GT 650M

## 4.3 Triển khai

Trong phạm vi bài nghiên cứu này, tôi sẽ sử dụng như sau:

- File thực nghiệm: Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv trong CICIDS-2017
- Thuật toán Machine Learning: Random Forest
- Công cụ thực hiện: Spyder
- Ngôn ngữ lập trình: Python 3

### 4.3.1 Tiền xử lý dữ liệu

- Bước 1: Đọc file dataset cần thực hiện

```
import pandas as pd
data  = pd.read_csv("Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv ")
```

- Bước 2: Chuyển đổi tất cả dữ liệu string của cột Label sang số

```
new_Label=[]
for i in data[" Label"]:
    if i =="BENIGN":
        new_Label.append(0)
    else:
        new_Label.append(1)
data[" Label"]=new_Label
```

Bước 3: Loại bỏ giá trị null ( vì thuật toán sẽ không chạy được nếu tồn tại giá trị null, hoặc để trống)

```
def clean_dataset(df):
    assert isinstance(df, pd.DataFrame), "df cần được chuyển sang pd.DataFrame"
    df.dropna(inplace=True)
    indices_to_keep = ~df.isin([np.nan, np.inf, -np.inf]).any(1)
    return df[indices_to_keep].astype(np.float64)
data = clean_dataset(data)
```

### 4.3.2 Trích xuất bộ thuộc tính

Đánh giá mức độ quan trọng của từng thuộc tính bằng chỉ số Gini-index của từng thuộc tính,hàm được áp dụng RandomForestRegressor, ta được kết quả như hình với thuộc tính cao nhất

```
y = data[" Label"].values
X = data.drop(' Label',axis=1).values

forest = RandomForestRegressor(n_estimators=250,random_state=0)
forest.fit(X, y)
importances = forest.feature_importances_
features=list(data.columns.values)
impor = pd.DataFrame({'Features':features[0:20],'importance':importances[0:20]})
impor = impor.sort_values('importance',ascending=False).set_index('Features')
print(impor.head(20))
```

Kết quả:

```
                           importance
Features
 Fwd Packet Length Max       0.573515
Total Length of Fwd Packets  0.238564
 Destination Port            0.001276
 Bwd Packet Length Mean      0.000274
 Bwd Packet Length Min       0.000259
 Flow IAT Min                0.000162
 Fwd Packet Length Std       0.000036
 Total Fwd Packets           0.000033
 Total Backward Packets      0.000024
 Flow IAT Mean               0.000021
 Flow Packets/s              0.000017
 Flow Duration               0.000016
Flow Bytes/s                 0.000015
 Flow IAT Std                0.000014
 Fwd Packet Length Mean      0.000012
 Flow IAT Max                0.000012
Bwd Packet Length Max        0.000007
 Total Length of Bwd Packets 0.000005
 Bwd Packet Length Std       0.000004
 Fwd Packet Length Min       0.000001
```

### 4.3.3 Thực hiện thuật toán Random Forest

Trích ra tập dữ liệu con với 4 đặc trưng có độ quan trọng cao nhất từ tập dữ liệu cha, thực hiện thuật toán học máy Random Forest

```
selected_features = list(impor.index)[:4]
selected_features.append(' Label')
data  = data[selected_features]
y =data[' Label'].values
X = data.drop(' Label',axis = 1).values
X_train, X_test, y_train, y_test = train_test_split(X, y,test_size = 0.30)
rf = RandomForestClassifier(n_estimators=100)
rf.fit(X_train, y_train)
predict =rf.predict(X_test)
```

### 4.3.4 Đánh giá hiệu suất thuật toán

```
from sklearn import metrics
acc = metrics.accuracy_score(y_test, predict)
print("Accuracy: ",acc)
rc=metrics.recall_score(y_test, predict)
print("Recall: ",rc)
pr=metrics.precision_score(y_test, predict)
print("Precision: ",pr)
f_1=metrics.f1_score(y_test, predict)
print("F_measure: ",f_1)
```

Kết quả:

```
Accuracy:  0.9990991523170983
Recall:  0.9992819530876017
Precision:  0.9986331795660345
F_measure:  0.9989574609902412
```

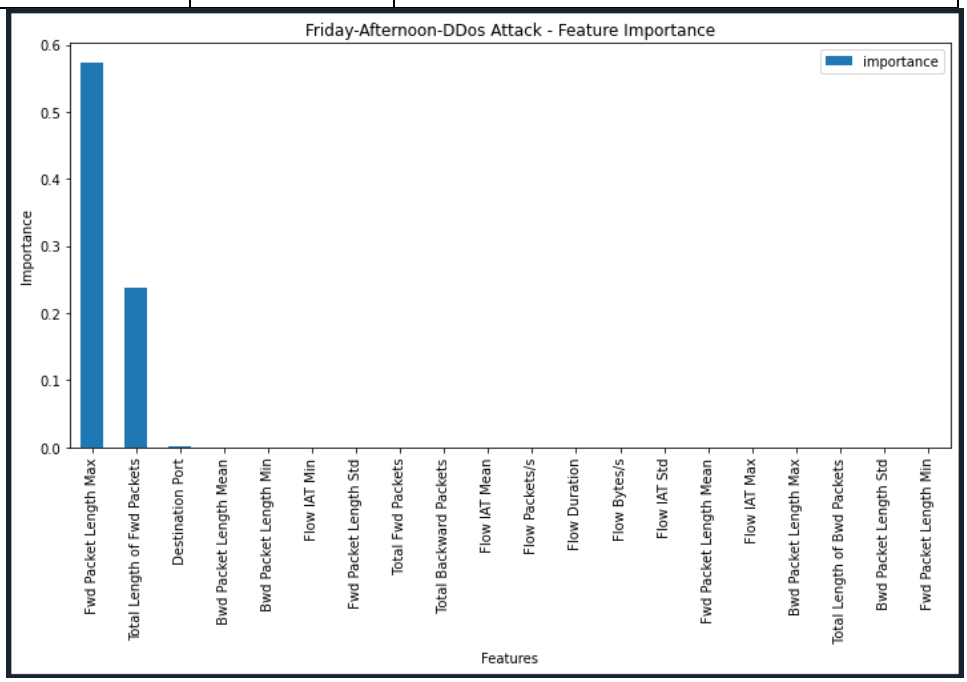
## 4.4 Kết quả đặt được

Sau khi thực nghiệm tương tự với 2 loại tấn công còn lại là web brute-force và Portscan, đã trích dẫn bộ dữ liệu con tối ưu nhất cho từng loại tấn công

### 4.4.1. Tấn công DDoS

**Bảng 4.1: Các đặc trưng tấn công DDoS**

| Đặc trưng | Độ quan trọng | Giải thích |
|---|---|---|
| Fwd Packet Length Max | 0.574 | Độ dài lớn nhất của các gói được gửi đi |
| Total Length of Fwd Packets | 0.239 | Tổng độ dài của các gói được gửi đi |
| Destination Port | 0.001 | Cổng đích |
| Bwd Packet Length Mean | 0.0003 | Độ dài trung bình của gói được nhận |

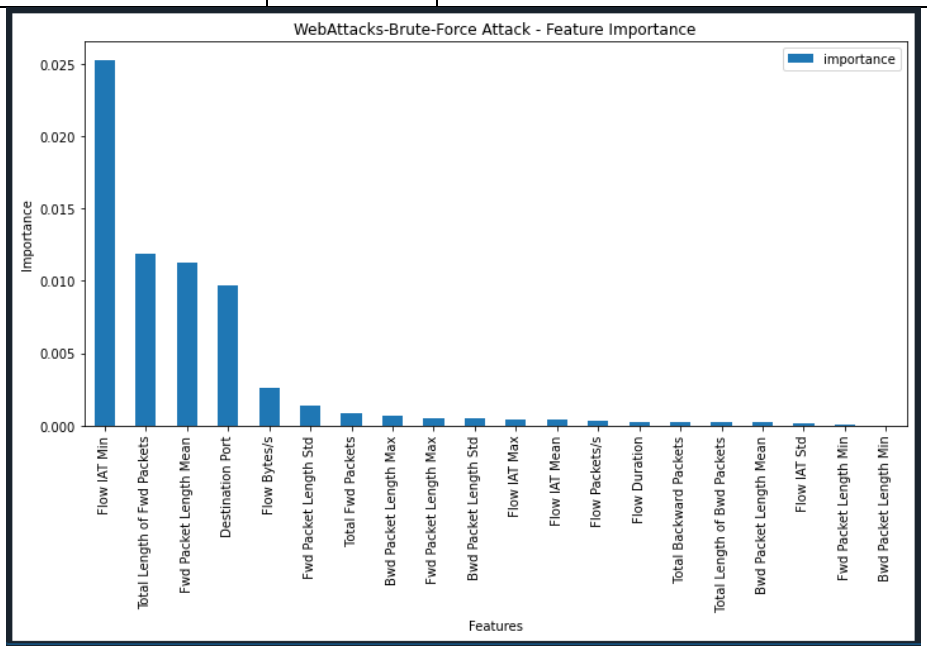



***Hình 4.1: Độ quan trọng của nhóm đặc trưng tấn công Ddos***

### 4.4.2. Tấn công web brute-force

**Bảng 4.2: Các đặc trưng tấn công web brute-force**

| Đặc trưng | Độ quan trọng | Giải thích |
|---|---|---|
| Flow IAT Min | 0.025 | Thời gian nhỏ nhất giữa 2 gói được chuyển tiếp trong một dòng |
| Total Length of Fwd Packets | 0.012 | Tổng độ dài gói được gửi đi |
| Fwd Packet Length Mean | 0.011 | Độ dài trung bình của các gói được gửi đi |
| Destination Port | 0.01 | Cổng đích |

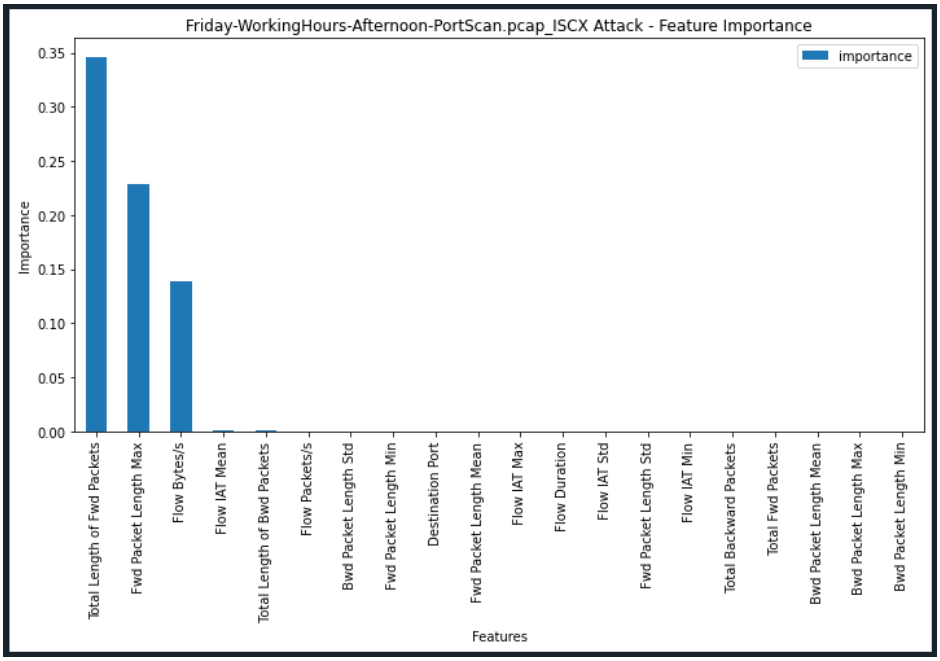



***Hình 4.2: Độ quan trọng của nhóm đặc trưng tấn công web brute-force***

### *4.4.3. Tấn công Portscan*

**Bảng 4.3: Các đặc trưng tấn công Portscan**

| Đặc trưng | Độ quan trọng | Giải thích |
|---|---|---|
| Total Length of Fwd Packets | 0.346 | Tổng độ dài gói được gửi đi |
| Fwd Packet Length Max | 0.228 | Độ dài gói lớn nhất được gửi |
| Flow Bytes/s | 0.138 | Số byte được truyền trên mỗi giây |
| Flow IAT Mean | 0.006 | Thời gian trung bình giữa 2 gói được truyền trong luồng |
| Total Length of Bwd Packets | 0.004 | Tổng độ dài các gói được nhận |




***Hình 4.3: Độ quan trọng của nhóm đặc trưng tấn công portscan***

# KẾT LUẬN

Với kết quả từ việc phân tích các đặc trưng điển hình của các cuộc tấn công, cho thấy là việc trích xuất chọn lọc lập dữ liệu có thể được sử dụng để kiểm tra và mô phỏng cải thiện hiệu suất của hệ thống IDS. Phương pháp được đề xuất giảm kích thước bộ dữ liệu, cải thiện tỉ lệ chính xác và giảm thời gian phát hiện.

Đối với các nghiên cứu trong tương lai, các nhà nghiên cứu nên nghiên cứu khả năng áp dụng các kỹ thuật tối ưu hóa và kết hợp với kết quả nghiên cứu này để đưa ra với mô hình phát hiện xâm nhập có tỷ lệ chính xác tốt hơn.

# TÀI LIỆU THAM KHẢO

[1] Josh Eklow, "The Importance of an Intrusion Detection System (IDS)", onshore.com, 2017

[2] Kunal, Mohit Dua "Machine Learning Approach to IDS: A Comprehensive Review", IEEE,2019

[3] Available from: https://www.unb.ca/cic/datasets/ids-2017.html, UNB_University of New Brunswick est.1785

[4] Côngty TNHH Tự động hóa và Tin học công nghiệp Bách Khoa (BKAII) (2010), Tầm quan trọng của trí tuệ nhân tạo với cuộc sống con người. Truy cập ngày 7/10/2020 từ địa chỉ https://bkaii.com.vn/tin-tuc/239-tam-quan-trong-cua-tri-tue-nhan-tao-voi-cuoc-song-con-nguoi

[5] D.E. Denning, An intruison-detection model, IEEE Symposium on Security and Privacy (1986) 118–131

[6] D. Anderson, T. Frivold, A. Valdes, Next- generation Intrusion Detection Expert System (NIDES): A summary, SRI Int., no. May 1995, p. 47, 1995.

[7] Hossen, S., Janagam A., "Analysis of Network Intrusion Detection System with Machine Learning Algorithms (Deep Reinforcement Learning Algorithm)" [master's thesis]. Karlskrona, Sweden: Faculty of Computing at Blekinge Institute of Technology; 2018.

[8] Sharafaldin I, Lashkari AH, Ghorbani AA., "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization", Proceedings of the 4th International Conference on Information Systems Security and Privacy, 2018, 108-116.

[9] Kostas, K., "Anomaly Detection in Networks Using Machine Learning" [master's thesis]. Colchester, UK: School of Computer Science and Electronic Engineering, University of Essex; 2018.

[10] Neelam Tyagi, "Understanding the Gini Index and Information Gain in Decision Trees", địa chỉ tại https://medium.com/analytics-steps/understanding-the-gini-index-and-information-gain-in-decision-trees-ab4720518ba8, 2020

[11] Nguyen Duy Sim ,"Phân lớp bằng Random Forests trong Python", địa chỉ tại https://viblo.asia/p/phan-lop-bang-random-forests-trong-python-djeZ1D2QKWz, 2018

[12] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: methods, systems and tools," Ieee communications surveys & tutorials, vol. 16, no. 1, pp. 303-336, 2014.

# PHỤ LỤC

## 1. Code chương trình

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import RandomForestRegressor
import matplotlib.pyplot as plt


# ================KHAI BÁO DATASET=============================================
data = pd.read_csv("AFriday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv")

#==================GÁN LẠI NHÃN ================================================
new_Label=[]
for i in data[" Label"]:
    if i =="BENIGN":
        new_Label.append(0)
    else:
        new_Label.append(1)
data[" Label"]=new_Label

#================LOẠI BỎ GIÁ TRỊ NULL=========================================
def clean_dataset(df):
    assert isinstance(df, pd.DataFrame),"df cần được chuyển sang pd.DataFrame"
    df.dropna(inplace=True)
    indices_to_keep = ~df.isin([np.nan, np.inf, -np.inf]).any(1)
    return df[indices_to_keep].astype(np.float64)
data = clean_dataset(data)

#================TÍNH ĐỘ QUAN TRỌNG CỦA TỪNG ĐẶC TRƯNG=========================

y = data[" Label"].values
X = data.drop(' Label',axis=1).values

forest = RandomForestRegressor(n_estimators=250,random_state=0)
forest.fit(X, y)
importances = forest.feature_importances_
features=list(data.columns.values)
impor = pd.DataFrame({'Features':features[0:20],'importance':importances[0:20]})
impor = impor.sort_values('importance',ascending=False).set_index('Features')
print(impor.head(20))
```

```
#====================Hiện thị đồ thị về độ quan trọng của thuộc tính==============================

plt.rcParams['figure.figsize'] = (10, 5)
impor.plot.bar();
plt.title("DDoS Attack - Feature Importance")
plt.ylabel('Importance')
plt.tight_layout()
plt.show()

#==========Training với bộ 4 thuộc tính có trọng lượng gini cao nhất với thuật toán RandomForest==

selected_features = list(impor.index)[:4]
print("Những đặc trưng được chọn")
print(selected_features)
selected_features.append(' Label')
data  = data[selected_features]
y =data[' Label'].values
X = data.drop(' Label',axis = 1).values
X_train, X_test, y_train, y_test = train_test_split(X, y,test_size = 0.30)
rf = RandomForestClassifier(n_estimators=100)
rf.fit(X_train, y_train)
predict =rf.predict(X_test)

#========================Đo các thông số hiệu suất của==================================

from sklearn import metrics
acc = metrics.accuracy_score(y_test, predict)
print("Accuracy: ",acc)
rc=metrics.recall_score(y_test, predict)
print("Recall: ",rc)
pr=metrics.precision_score(y_test, predict)
print("Precision: ",pr)
f_1=metrics.f1_score(y_test, predict)
print("F_measure: ",f_1)

#========================Trích xuất 1 cây quyết định trong Rừng ngẫu nhiên===============
from sklearn import tree
estimator = rf.estimators_[1]
tree.plot_tree(estimator)
```