TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**TRẦN THANH HUY - 51900800**

**NGUYỄN TRƯỜNG KHẢ - 51900752**

**PHÁT HIỆN TẤN CÔNG DDOS**

**DỰ ÁN CÔNG NGHỆ THÔNG TIN 2**

**MẠNG MÁY TÍNH**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**TRẦN THANH HUY - 51900800**

**NGUYỄN TRƯỜNG KHẢ- 51900752**

**PHÁT HIỆN TẤN CÔNG DDOS**

**DỰ ÁN CÔNG NGHỆ THÔNG TIN 2**

**MẠNG MÁY TÍNH**

Người hướng dẫn

**TS. TRƯƠNG ĐÌNH TÚ**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

**LỜI CẢM ƠN**

Lời đầu tiên em xin chân thành cảm ơn quý thầy, cô khoa Công Nghệ Thông Tin đã tận tình giảng dạy, chỉ bảo em trong những năm học vừa qua, những kiến thức mà em nhận được trên giảng đường đại học là nền tảng cho quá trình nghiên cứu đề tài này. Em xin chân thành cảm ơn Khoa Công nghệ thông tin đã tạo điều kiện để em có thể thực hiện đề tài nghiên cứu.

Em xin chân thành cảm ơn thầy Trương Đình Tú – giảng viên hướng dẫn em môn học Dự án Công nghệ thông tin, đã luôn theo sát, nhiệt tình giúp đỡ em trong suốt quá trình thực hiện đề tài **Phát Hiện tấn công DDoS**.

Mặc dù bản thân em đã nỗ lực hết mức, nhưng bài báo cáo này không thể tránh khỏi còn nhiều thiếu sót. Vì thế, em mong rằng sẽ nhận được những lời nhận xét, góp ý từ quý thầy, cô để em có thể tiếp tục phát triển và hoàn thiện hơn nữa.

Một lần nữa, em xin chân thành cảm ơn!

*TP. Hồ Chí Minh, ngày ... tháng … năm 20..*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*

**CÔNG TRÌNH ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi và được sự hướng dẫn khoa học của TS. Trương Đình Tú. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong Dự án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung Dự án của mình**. Trường Đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày … tháng … năm 20..*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*

**PHÁT HIỆN TẤN CÔNG DDoS**

**TÓM TẮT**

Các cuộc tấn công từ chối dịch vụ phân tán (DDoS) đã trở thành vũ khí ưa thích của tin tặc, tống tiền mạng và khủng bố mạng. Những cuộc tấn công này có thể nhanh chóng vô hiệu hóa nạn nhân, gây thiệt hại lớn về doanh thu. Mặc dù có nhiều giải pháp truyền thống để giảm thiểu, các cuộc tấn công DDoS vẫn tiếp tục gia tăng về tần suất, quy mô và mức độ nghiêm trọng. Điều này đòi hỏi một mô hình mạng mới để đối phó với các mối đe dọa an ninh đầy thách thức ngày nay. Mạng điều khiển bằng phần mềm (SDN) là một mô hình mạng mới nổi, đã thu hút được sự quan tâm đáng kể từ nhiều nhà nghiên cứu để đáp ứng yêu cầu của các trung tâm dữ liệu hiện nay.

Phương pháp học máy đang ngày càng được ứng dụng rộng rãi trong việc phát hiện tấn công từ chối dịch vụ phân tán (DDoS). So với các phương pháp truyền thống, như dựa trên các quy tắc tĩnh hoặc ngưỡng phát hiện cố định, phương pháp học máy có khả năng phân tích và nhận diện các mẫu hành vi phức tạp và bất thường trong lưu lượng mạng. Các mô hình học máy có thể học từ dữ liệu lịch sử, cải thiện độ chính xác và khả năng phát hiện các cuộc tấn công mới hoặc chưa từng gặp trước đây. Điều này làm cho phương pháp học máy trở nên linh hoạt và hiệu quả hơn trong môi trường mạng liên tục biến đổi, giúp bảo vệ hệ thống tốt hơn trước các mối đe dọa DDoS ngày càng tinh vi.

**DDoS ATTACK DETECTION**

**ABSTRACT**

Distributed denial of service (DDoS) attacks have become the preferred weapon of hackers, cyber extortion, and cyber terrorism. These attacks can quickly disable victims, causing major revenue losses. Despite many traditional solutions for mitigation, DDoS attacks continue to increase in frequency, scale, and severity. This requires a new networking paradigm to deal with today's challenging security threats. Software-driven networking (SDN) is an emerging networking paradigm that has attracted significant attention from many researchers to meet the requirements of today's data centers.

Machine learning methods are increasingly being widely applied in detecting distributed denial of service (DDoS) attacks. Compared to traditional methods, such as based on static rules or fixed detection thresholds, machine learning methods are capable of analyzing and identifying complex and unusual behavioral patterns in network traffic. Machine learning models can learn from historical data, improving accuracy and the ability to detect new or never-before-seen attacks. This makes machine learning more flexible and effective in constantly changing network environments, helping to better protect systems against increasingly sophisticated DDoS threats.

**MỤC LỤC**

[DANH MỤC HÌNH VẼ vii](#_Toc171863830)

[DANH MỤC BẢNG BIỂU viii](#_Toc171863831)

[DANH MỤC CÁC CHỮ VIẾT TẮT ix](#_Toc171863832)

[CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI 1](#_Toc171863833)

[1.1 Giới thiệu đề tài 1](#_Toc171863834)

[1.2 Mục tiêu thực hiện đề tài 2](#_Toc171863835)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 3](#_Toc171863836)

[2.1 Tấn cộng từ chối dịch vụ phân tán (DDoS) 3](#_Toc171863837)

[2.2 Công nghệ SDN 4](#_Toc171863838)

[2.2.1 Lịch sử và sự ra đời của SDN 4](#_Toc171863839)

[2.2.2 Định nghĩa về SDN 5](#_Toc171863840)

[2.2.3 Kiến trúc SDN 6](#_Toc171863841)

[2.2.4 Một số mô hình SDN phổ biến 8](#_Toc171863842)

[2.2.5 So sánh SDN và mạng truyền thống 9](#_Toc171863843)

[2.2.6 Ưu điểm và nhược điểm của SDN 9](#_Toc171863844)

[2.3 Mô hình Decision Tree 12](#_Toc171863845)

[2.3.1 Định nghĩa Decision Tree 12](#_Toc171863846)

[2.3.2 Cấu trúc cây quyết định 12](#_Toc171863847)

[2.3.3 Các phiên bản của thuật toán Decision Tree 16](#_Toc171863848)

[2.3.4 Ứng dụng của Decision Tree 17](#_Toc171863849)

[2.3.5 Ưu điểm và nhược điểm của Decision Tree 18](#_Toc171863850)

[CHƯƠNG 3. THỰC NGHIỆM 19](#_Toc171863851)

[3.1 Dữ liệu thực nghiệm 19](#_Toc171863852)

[3.2 Cài đặt thực nghiệm 20](#_Toc171863853)

[CHƯƠNG 4. KẾT LUẬN 21](#_Toc171863854)

[4.1 Kết luận 21](#_Toc171863855)

[4.2 Hướng phát triển 21](#_Toc171863856)

[TÀI LIỆU THAM KHẢO 22](#_Toc171863857)

# DANH MỤC HÌNH VẼ

[Hình 1: Kiến trúc cơ bản của SDN 7](#_Toc171863901)

[Hình 2: Open SDN 8](#_Toc171863902)

[Hình 3: SDN via Hypervisor-based Overlay Network 9](#_Toc171863903)

[Hình 4: Cấu trúc cây quyết định 13](#_Toc171863904)

# DANH MỤC BẢNG BIỂU

[Bảng 4.1: Thống kê kiểu thực thể trong tập VLSP 2016 11](#_Toc171792390)

# DANH MỤC CÁC CHỮ VIẾT TẮT

|  |  |
| --- | --- |
| DDoS | Distributed denial of service |
| SDN | Software defined Networking |
| MLM | Masked Language Model |
| NLP | Natural Language Processing |
| NSP | Next Sentence Prediction |

# TỔNG QUAN ĐỀ TÀI

## Giới thiệu đề tài

Các cuộc tấn công từ chối dịch vụ phân tán (DDoS) đã trở thành mối đe dọa thực sự đối với mạng, hạ tầng kỹ thuật số, và hạ tầng an ninh mạng. Những cuộc tấn công này có khả năng gây ra sự gián đoạn lớn trong bất kỳ cơ sở hạ tầng công nghệ thông tin và truyền thông (ICT) nào. Có nhiều lý do khác nhau để tiến hành các cuộc tấn công DDoS, bao gồm lợi ích tài chính, lợi ích chính trị, và gây gián đoạn. Các cuộc tấn công DDoS có thể làm tê liệt mạng lưới và dịch vụ bằng cách làm ngập các máy chủ, liên kết mạng, và các thiết bị mạng (bộ định tuyến, bộ chuyển mạch, v.v.) với lưu lượng truy cập không hợp lệ. Chúng có thể gây ra sự suy giảm dịch vụ hoặc từ chối dịch vụ hoàn toàn, dẫn đến tổn thất lớn.

Có nhiều giải pháp độc quyền và mã nguồn mở để phát hiện và giảm thiểu các cuộc tấn công DDoS. Tuy nhiên, những cuộc tấn công này tiếp tục gia tăng về tần suất, độ phức tạp, và mức độ nghiêm trọng. Việc phát hiện và giảm thiểu nhanh chóng các cuộc tấn công DDoS đã trở nên vô cùng thách thức khi kẻ tấn công liên tục sử dụng các kỹ thuật mới để thực hiện tấn công DDoS. Số lượng ngày càng tăng của các cuộc tấn công DDoS, cùng với sự đa dạng ngày càng lớn của chúng, gây ra tác động thảm khốc, đã làm cho việc phát hiện, giảm thiểu và phòng chống tấn công DDoS trở thành ưu tiên hàng đầu.

Ví dụ, Arbor Networks Inc., một trong những nhà cung cấp giải pháp bảo vệ khỏi mối đe dọa DDoS hàng đầu, đã báo cáo một cuộc tấn công 334 Gbps nhắm vào một nhà điều hành mạng ở châu Á gần đây. Ngoài ra, công ty này cũng báo cáo nhiều cuộc tấn công lớn hơn 100 Gbps trên toàn cầu vào năm 2015. Nhiều sự cố như vậy cho thấy rõ rằng chúng ta cần có những phương pháp tiếp cận mới để giải quyết vấn đề tấn công DDoS. Những phương pháp mới này phải được thiết kế để đáp ứng các yêu cầu về hiệu suất và khả năng mở rộng của các trung tâm dữ liệu hiện đại và cung cấp mức độ bảo vệ tối đa chống lại các cuộc tấn công mới nổi, phức tạp và khó nắm bắt.

Với những tiến bộ gần đây trong công nghệ mạng điều khiển bằng phần mềm (SDN) và sự chấp nhận nhanh chóng, rộng rãi của cộng đồng mạng, nhiều nhà nghiên cứu đã tích cực tham gia phát triển các giải pháp bảo mật mạng dựa trên SDN. Các giải pháp dựa trên SDN đã thu hút được nhiều sự chú ý hơn kể từ khi chúng được áp dụng trong các mạng diện rộng quy mô lớn. Công nghệ này cho phép các nhà phát triển lập trình, kiểm soát, và quản lý tài nguyên mạng một cách trực tiếp và tập trung thông qua bộ điều khiển SDN.

Bài báo cáo này sẽ tập trung vào việc nghiên cứu và phân tích phương pháp phát hiện tấn công DDoS bằng mô hình học máy trong môi trường SDN. Chúng ta sẽ xem xét các thuật toán học máy, đánh giá hiệu quả và tính khả thi của chúng trong việc bảo vệ hệ thống mạng SDN trước các mối đe dọa DDoS. Qua đó, chúng ta sẽ đề xuất những hướng đi và giải pháp mới nhằm nâng cao khả năng phòng chống tấn công DDoS trong môi trường mạng hiện đại.

## Mục tiêu thực hiện đề tài

Mục tiêu của đồ án là xây dựng được hệ thống SDN, mô phỏng các cuộc tấn công DDOS và thử nghiệm các phương pháp phát hiện và giảm thiểu. Kết quả thực nghiệm cho thấy phương pháp phát hiện và giảm thiểu tấn công DDoS dựa trên Machine Learning kết hợp được đề xuất có tỷ lệ phát hiện tốt đối với tấn công DDoS phổ biến.

# CƠ SỞ LÝ THUYẾT

## Tấn cộng từ chối dịch vụ phân tán (DDoS)

Trước khi định nghĩa về DDoS chúng ta sẽ nói về DoS (Denial of Service). Tấn công DoS là một cuộc tấn công mạng nhằm mục đích làm cho một dịch vụ, hệ thống hoặc mạng trở nên không thể truy cập được bởi người dùng hợp pháp. Điều này được thực hiện bằng cách làm quá tải hệ thống mục tiêu với lưu lượng truy cập hoặc yêu cầu không hợp lệ, khiến hệ thống bị chậm hoặc ngừng hoạt động.

* Phương thức tấn công: Tấn công DoS thường được thực hiện từ một nguồn duy nhất, như một máy tính hoặc một địa chỉ IP.
* Cách thức: Tin tặc gửi một lượng lớn yêu cầu hoặc gói dữ liệu tới mục tiêu, khiến hệ thống không thể xử lý kịp và dẫn đến tình trạng quá tải hoặc lỗi.

Tấn công DDoS là một loại tấn công DoS nhưng với quy mô lớn hơn, được thực hiện từ nhiều nguồn khác nhau. Mục tiêu của DDoS cũng là làm cho hệ thống, dịch vụ hoặc mạng trở nên không thể truy cập được, nhưng với cách tấn công phân tán từ nhiều máy tính hoặc thiết bị. Bảng 1 so sánh một số thông tin cơ bản giữa DoS và DDos.

* Phương thức tấn công: Tấn công DDoS được thực hiện từ nhiều nguồn khác nhau, thường là từ các mạng botnet bao gồm hàng ngàn hoặc hàng triệu thiết bị bị chiếm quyền điều khiển.
* Cách thức: Tin tặc kiểm soát mạng botnet để gửi lượng lớn lưu lượng truy cập tới mục tiêu, làm cho hệ thống bị quá tải và không thể phục vụ người dùng hợp pháp.

|  |  |
| --- | --- |
| **DoS** | **DDoS** |
| DoS viết tắt của tấn công Từ chối  dịch vụ. | DDoS viết tắt của tấn công Từ chdịch vụ phân tán. |
| Trong tấn công DoS, một hệ thống  đơn lẻ tấn công hệ thống mục tiêu. | Trong tấn công DDoS, nhiều hệ  thống tấn công hệ thống mục tiêu. |
| Máy tính của nạn nhân bị quá tải  từ các gói dữ liệu được gửi từ một  vị trí duy nhất. | Máy tính của nạn nhân bị quá tải từ  các gói dữ liệu được gửi từ nhiều vị  trí. |
| Tấn công DoS chậm hơn so với  DDoS. | Tấn công DDoS nhanh hơn so với tấn công DoS |
| Dễ dàng bị chặn do chỉ có một hệ  thống được sử dụng. | Khó chặn vì nhiều thiết bị gửi các  gói dữ liệu và tấn công từ nhiều vị  trí. |
| Trong tấn công DoS, chỉ có một  thiết bị được sử dụng với các công  cụ tấn công DoS. | Trong tấn công DDoS, các volumeBots  được sử dụng để tấn công cùng  lúc. |
| Tấn công DoS dễ dàng bị truy vết. | Tấn công DDoS khó bị truy vết. |
| Các loại tấn công DoS:  1. Tấn công tràn bộ đệm  2. Ping of Death hoặc ICMP flood  3. Tấn công Teardrop  4. Tấn công Flooding | Các loại tấn công DDoS:  1. Tấn công Volumetric  2. Tấn công Phân mảnh  3. Tấn công Tầng ứng dụng  4. Tấn công Giao thức |

Bảng 1: Sự khác nhau giữa DoS và DDoS

## Công nghệ SDN

### Lịch sử và sự ra đời của SDN

Cuối những năm 1990 - đầu 2000:

* Các nghiên cứu ban đầu về SDN bắt đầu từ những năm 1990, khi các nhà nghiên cứu nhận thấy sự hạn chế của mô hình mạng truyền thống trong việc quản lý và điều khiển mạng.
* Dự án "4D" (Data, Discovery, Dissemination, and Decision) tại Đại học Stanford và Berkeley là một trong những nỗ lực nghiên cứu ban đầu để tách biệt chức năng điều khiển khỏi phần cứng mạng.

Năm 2005:

* Dự án Ethane, do Giáo sư Nick McKeown và một nhóm các nhà nghiên cứu tại Đại học Stanford khởi xướng, đã đặt nền móng cho SDN. Ethane tập trung vào việc kiểm soát và quản lý mạng dựa trên các chính sách được định nghĩa phần mềm.

Năm 2008:

* Dự án OpenFlow được khởi xướng, mở đường cho sự phát triển của SDN. OpenFlow là giao thức cho phép các bộ điều khiển mạng tương tác với các switch và router để thiết lập các quy tắc chuyển tiếp dữ liệu. OpenFlow chính là yếu tố then chốt giúp hiện thực hóa khái niệm SDN.

Năm 2011:

* Sự ra đời của Open Networking Foundation (ONF), một tổ chức phi lợi nhuận, nhằm thúc đẩy sự phát triển và tiêu chuẩn hóa của SDN và OpenFlow. ONF được thành lập bởi các công ty công nghệ lớn như Google, Facebook, Microsoft, Verizon, và Yahoo.

### Định nghĩa về SDN

SDN (Software-Defined Networking) là một phương pháp tiếp cận quản lý mạng dựa trên phần mềm. Nó tách biệt các chức năng điều khiển mạng khỏi các thiết bị phần cứng truyền thống như router và switch, thay vào đó sử dụng phần mềm để quản lý và điều khiển mạng.

Về cơ bản, SDN chia tách độc lập hai cơ chế hiện đang tồn tại trong cùng một thiết bị mạng: Cơ chế điều khiển (Control Plane), cơ chế chuyển tiếp dữ liệu (Data Plane) để có thể tối ưu hoạt động của hai cơ chế này.

Control Plane (Mặt phẳng điều khiển)

* Chức năng: Control Plane chịu trách nhiệm ra quyết định về cách các gói dữ liệu sẽ được chuyển tiếp qua mạng. Nó bao gồm các chức năng như định tuyến, định tuyến lại (rerouting), xây dựng bảng định tuyến (routing tables), và các giao thức điều khiển như OSPF, BGP, và RIP.
* Hoạt động: Các thiết bị mạng như router và switch sử dụng Control Plane để trao đổi thông tin về trạng thái mạng, từ đó đưa ra quyết định về đường đi của các gói dữ liệu. Ví dụ, router sử dụng giao thức định tuyến để học biết về các mạng khác và xác định đường đi tốt nhất cho các gói dữ liệu.
* Vị trí: Control Plane thường được tích hợp trong các thiết bị mạng (router, switch). Mỗi thiết bị mạng có Control Plane riêng của nó để quản lý và ra quyết định.

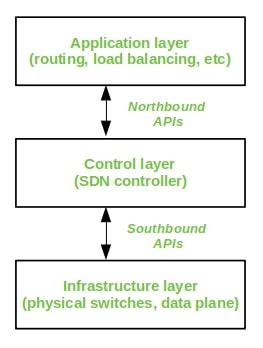
Data Plane (Mặt phẳng dữ liệu)

* Chức năng: Data Plane chịu trách nhiệm chuyển tiếp các gói dữ liệu theo các quyết định được đưa ra bởi Control Plane. Nó thực hiện các hoạt động như truyền tải, định tuyến và lọc gói tin.
* Hoạt động: Khi một gói dữ liệu đến một thiết bị mạng, Data Plane sẽ dựa vào bảng định tuyến (được xây dựng bởi Control Plane) để xác định đường đi của gói tin và chuyển tiếp nó đến đích. Data Plane thực hiện các hoạt động này ở tốc độ cao để đảm bảo hiệu suất mạng.
* Vị trí: Data Plane cũng được tích hợp trong các thiết bị mạng, nơi nó hoạt động song song với Control Plane nhưng với vai trò khác biệt.

Trong các mạng truyền thống, Control Plane và Data Plane thường được tích hợp trong cùng một thiết bị mạng. Tuy nhiên, với sự ra đời của SDN (Software-Defined Networking), hai mặt phẳng này được tách biệt rõ ràng:

* Control Plane: Được chuyển lên các bộ điều khiển trung tâm (controllers), quản lý và ra quyết định cho toàn bộ mạng.
* Data Plane: Vẫn được duy trì trong các thiết bị mạng nhưng hoạt động dưới sự chỉ đạo của Control Plane từ các bộ điều khiển.

### Kiến trúc SDN



Hình 1: Kiến trúc cơ bản của SDN

Kiến trúc SDN bao gồm ba lớp chính:

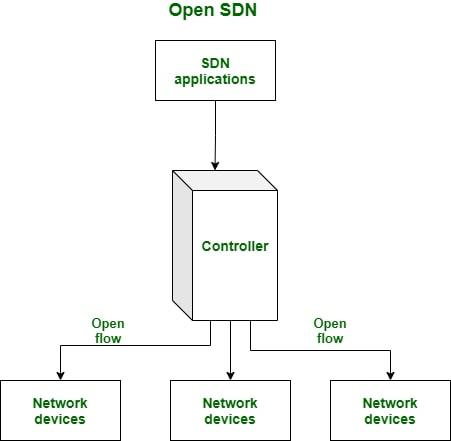
* Lớp hạ tầng mạng (Infrastructure Layer):
  + Bao gồm các thiết bị mạng như switch và router.
  + Chịu trách nhiệm truyền tải dữ liệu.
* Lớp điều khiển (Control Layer):
  + Gồm các bộ điều khiển (controller), đây là nơi diễn ra việc ra quyết định và quản lý toàn bộ mạng.
  + Controller giao tiếp với các thiết bị mạng qua giao diện phía Nam (Southbound Interface), thường sử dụng giao thức OpenFlow.
* Lớp ứng dụng (Application Layer):
  + Chứa các ứng dụng mạng chạy trên đỉnh của controller.
  + Các ứng dụng này cung cấp các chức năng và dịch vụ mạng cụ thể, như quản lý lưu lượng, bảo mật, và ảo hóa mạng.

Các lớp giao tiếp thông qua một tập hợp các giao diện được gọi là northbound APIs và southbound APIs.

### Một số mô hình SDN phổ biến

Open SDN (hình 2): được triển khai bằng cách sử dụng switch OpenFlow. Đây

là một cách triển khai đơn giản của SDN. Trong Open SDN, bộ điều khiển giao tiếp với các switch bằng cách sử dụng southbound APIs thông qua giao thức OpenFlow.



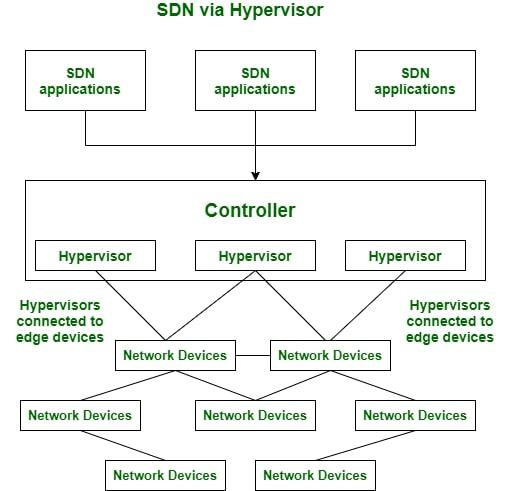
Hình 2: Open SDN

SDN via APIs: các chức năng trong các thiết bị từ xa như switch được gọi

bằng các phương pháp truyền thống như SNMP hoặc CLI hoặc thông qua các phương pháp mới như Rest API. Ở đây, các thiết bị được cung cấp với điểm điều khiển cho phép bộ điều khiển thao tác các thiết bị từ xa bằng cách sử dụng các API.

SDN via Hypervisor-based Overlay Network (hình 3): cấu hình của các thiết

bị vật lý không thay đổi. Thay vào đó, các mạng lớp phủ dựa trên Hypervisor được tạo ra trên mạng vật lý. Chỉ có các thiết bị ở biên của mạng vật lý được kết nối với các mạng ảo hóa, do đó che giấu thông tin của các thiết bị khác trong mạng vật lý.



Hình 3: SDN via Hypervisor-based Overlay Network

Hybrid SDN là sự kết hợp của Mạng truyền thống với Mạng được định nghĩa

bằng phần mềm trong một mạng duy nhất để hỗ trợ các loại chức năng khác nhau trên một mạng.

### So sánh SDN và mạng truyền thống

|  |  |
| --- | --- |
| **Mạng Định Nghĩa Bằng Phần Mềm** | **Mạng Truyền Thống** |
| Mạng Định Nghĩa Bằng Phần Mềm là một cách tiếp cận mạng ảo. | Mạng truyền thống là cách tiếp cận mạng cổ điển. |
| Kiểm soát tập trung. | Kiểm soát phân tán. |
| Có thể lập trình được. | Không thể lập trình được |
| Có giao diện mở. | Có giao diện đóng. |
| Data plane và Control Plane được tách biệt bởi phần mềm. | Data Plane và Control Plane được gắn kết trên cùng một mặt phẳng. |

Bảng 2: So sánh giữa SDN và mạng truyền thống

### Ưu điểm và nhược điểm của SDN

Ưu điểm của SDN:

* Tập trung quản lý và kiểm soát mạng: SDN tách biệt phần điều khiển mạng (control plane) khỏi phần chuyển tiếp dữ liệu (data plane), cho phép quản lý và kiểm soát toàn bộ mạng từ một điểm trung tâm duy nhất.
* Khả năng lập trình cao: Với việc tách biệt phần điều khiển, SDN cung cấp khả năng lập trình mạng, giúp quản lý và cấu hình mạng một cách linh hoạt và hiệu quả hơn.
* Ảo hóa mạng: SDN cho phép ảo hóa tài nguyên mạng vật lý, tạo ra các mạng ảo riêng biệt cho các ứng dụng và dịch vụ khác nhau, tăng cường hiệu quả sử dụng tài nguyên.
* Đổi mới và triển khai nhanh: Nhờ khả năng lập trình, SDN giúp đơn giản hóa việc triển khai và đổi mới các tính năng mạng mới, giảm thời gian và chi phí.
* Hỗ trợ điện toán đám mây: SDN là một công nghệ quan trọng trong việc hỗ trợ và cải thiện các dịch vụ điện toán đám mây, cho phép quản lý tài nguyên mạng một cách hiệu quả hơn.
* An ninh mạng tăng cường: Với khả năng giám sát và kiểm soát tập trung, SDN có thể được sử dụng để triển khai các giải pháp an ninh mạng hiệu quả hơn.

Nhược điểm của SDN:

* Phụ thuộc vào bộ điều khiển trung tâm: SDN phụ thuộc vào bộ điều khiển trung tâm để quản lý và kiểm soát mạng. Nếu bộ điều khiển bị lỗi hoặc tấn công, toàn bộ mạng sẽ bị ảnh hưởng.
* Vấn đề tương thích và di chuyển: Việc chuyển đổi từ mạng truyền thống sang SDN có thể gặp phải vấn đề tương thích và khó khăn trong di chuyển.
* Yêu cầu về nhân lực có kỹ năng chuyên môn: Triển khai và vận hành SDN đòi hỏi nhân lực có kỹ năng chuyên môn cao về lập trình mạng và quản lý hệ thống phức tạp.
* Rủi ro về bảo mật: Tập trung điều khiển mạng tại một điểm duy nhất cũng làm tăng nguy cơ bị tấn công và xâm nhập bảo mật.
* Chi phí triển khai ban đầu cao: Chi phí triển khai SDN ban đầu có thể cao hơn so với mạng truyền thống, đặc biệt là đối với các tổ chức quy mô lớn.

Mặc dù có một số nhược điểm, nhưng SDN vẫn được coi là một công nghệ mạng tiên tiến và có tiềm năng lớn trong tương lai. Mạng định nghĩa bằng phần mềm (Software-Defined Networking - SDN) đã mở ra nhiều ứng dụng tiềm năng trong các lĩnh vực khác nhau. Một số ứng dụng quan trọng của SDN bao gồm:

* Quản lý và kiểm soát mạng tập trung: SDN cho phép quản lý và kiểm soát toàn bộ mạng từ một điểm trung tâm duy nhất, điều này giúp đơn giản hóa việc triển khai và cấu hình mạng, cũng như nâng cao hiệu quả quản lý.
* Ảo hóa mạng: Nhờ khả năng lập trình của SDN, các tài nguyên mạng vật lý có thể được ảo hóa và cấp phát một cách linh hoạt cho các ứng dụng và dịch vụ khác nhau, tăng cường hiệu quả sử dụng tài nguyên.
* Điện toán đám mây: SDN là một công nghệ quan trọng trong việc hỗ trợ và cải thiện các dịch vụ điện toán đám mây, cho phép quản lý tài nguyên mạng một cách hiệu quả và linh hoạt hơn.
* An ninh mạng: Với khả năng giám sát và kiểm soát tập trung, SDN có thể được sử dụng để triển khai các giải pháp an ninh mạng hiệu quả hơn, như phát hiện và ngăn chặn các cuộc tấn công mạng, triển khai tường lửa và chính sách kiểm soát truy cập.
* Internet of Things (IoT): Với khả năng quản lý và cấu hình mạng linh hoạt, SDN có thể đóng một vai trò quan trọng trong việc kết nối và quản lý các thiết bị IoT một cách hiệu quả.
* Mạng doanh nghiệp: SDN cung cấp khả năng quản lý và kiểm soát tập trung, giúp đơn giản hóa việc triển khai và vận hành mạng doanh nghiệp, đồng thời tăng cường an ninh và hiệu quả sử dụng tài nguyên.
* Mạng trung tâm dữ liệu: Trong các trung tâm dữ liệu lớn, SDN giúp tối ưu hóa việc sử dụng tài nguyên mạng, nâng cao hiệu năng và khả năng mở rộng của mạng.

## Mô hình Decision Tree

### Định nghĩa Decision Tree

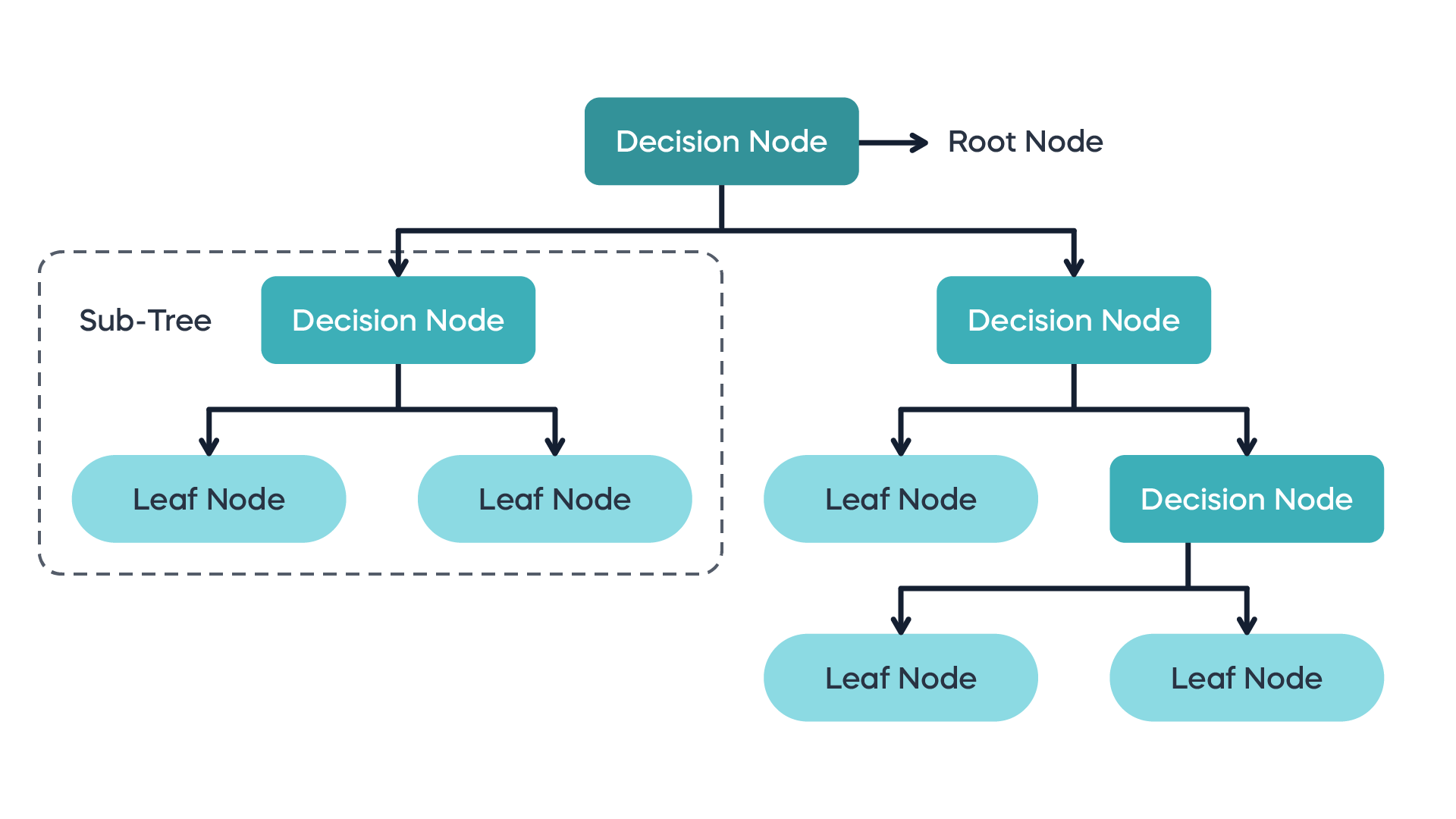
Cây quyết định (Decision Tree) là một trong những công cụ mạnh mẽ nhất của các thuật toán học có giám sát được sử dụng cho cả hai nhiệm vụ phân loại và hồi quy. Nó xây dựng một cấu trúc cây giống như sơ đồ luồng, trong đó mỗi nút bên trong biểu thị một bài kiểm tra trên một thuộc tính, mỗi nhánh đại diện cho một kết quả của bài kiểm tra và mỗi nút lá (nút cuối) chứa một nhãn lớp. Cây được xây dựng bằng cách chia đệ quy dữ liệu huấn luyện thành các tập con dựa trên các giá trị của các thuộc tính cho đến khi một tiêu chí dừng được đáp ứng, chẳng hạn như độ sâu tối đa của cây hoặc số lượng mẫu tối thiểu cần thiết để chia một nút.

Trong quá trình huấn luyện, thuật toán Cây Quyết Định chọn thuộc tính tốt nhất để chia dữ liệu dựa trên một số chỉ số như entropy hoặc độ bất thuần Gini (Gini impurity), mà đo lường mức độ bất thuần hoặc ngẫu nhiên trong các tập con. Mục tiêu là tìm thuộc tính tối đa hóa thu được thông tin hoặc giảm độ bất thuần sau khi chia.

### Cấu trúc cây quyết định

Hình 4 mô tả cấu trúc đơn giản của thuật toán Decision Tree. Các thuật ngữ được giải thích như sau:

* Root Node: Là nút ở trên cùng của cây, đại diện cho toàn bộ tập dữ liệu. Đây là điểm khởi đầu của quá trình ra quyết định.
* Decision/Internal Node: Nút biểu thị một lựa chọn liên quan đến một đặc trưng đầu vào. Việc phân nhánh từ các nút nội bộ kết nối chúng với các nút lá hoặc các nút nội bộ khác.
* Leaf/Terminal Node: Nút không có nút con, chỉ ra một nhãn lớp hoặc một giá trị số.
* Branch/Sub-Tree: Một phần nhỏ của cây quyết định bắt đầu từ một nút nội bộ và kết thúc tại các nút lá.



Hình 4: Cấu trúc cây quyết định

Một số thuật ngữ khác:

* Splitting: Quá trình chia một nút thành hai hoặc nhiều nút con bằng cách sử dụng tiêu chí chia tách và một đặc trưng được chọn.
* Parent Node: Nút chia thành một hoặc nhiều nút con.
* Child Node: Các nút xuất hiện khi một nút cha được chia.
* Impurity: Một phép đo độ đồng nhất của biến mục tiêu trong một tập hợp dữ liệu con. Nó đề cập đến mức độ ngẫu nhiên hoặc không chắc chắn trong một tập hợp các ví dụ. Chỉ số Gini và entropy là hai phép đo độ bất thuần thường được sử dụng trong cây quyết định cho các nhiệm vụ phân loại.
* Variance: Phương sai đo lường mức độ biến đổi giữa các biến dự đoán và biến mục tiêu trong các mẫu khác nhau của một tập dữ liệu. Nó được sử dụng cho các bài toán hồi quy trong cây quyết định. Mean Squared Error, Mean Absolute Error, friedman\_mse, hoặc Half Poisson deviance được sử dụng để đo lường phương sai cho các nhiệm vụ hồi quy trong cây quyết định.
* Information Gain: Thông tin thu được là một phép đo sự giảm độ bất thuần đạt được bằng cách chia một tập dữ liệu trên một đặc trưng cụ thể trong cây quyết định. Tiêu chí chia tách được xác định bởi đặc trưng cung cấp thông tin thu được lớn nhất. Nó được sử dụng để xác định đặc trưng thông tin nhất để chia tại mỗi nút của cây, với mục tiêu tạo ra các tập con thuần nhất.
* Pruning: Quá trình loại bỏ các nhánh khỏi cây không cung cấp thêm thông tin hoặc dẫn đến hiện tượng quá khớp (overfitting).

Quá trình tạo cây quyết định bao gồm:

* Chọn thuộc tính tốt nhất: Sử dụng số liệu như Gini impurity, entropy, hoặc information gain, thuộc tính tốt nhất để phân chia dữ liệu sẽ được chọn.
* Tách tập dữ liệu: Tập dữ liệu được chia thành các tập con dựa trên thuộc tính đã chọn.
* Lặp lại quy trình: Quy trình được lặp lại đệ quy cho từng tập hợp con, tạo một nút bên trong hoặc nút lá mới cho đến khi đáp ứng tiêu chí dừng (ví dụ: tất cả các phiên bản trong một nút thuộc về cùng một lớp hoặc đạt đến độ sâu được xác định trước).

**Entropy** là thước đo mức độ ngẫu nhiên hoặc không chắc chắn trong tập dữ liệu. Trong trường hợp phân loại, nó đo lường sự ngẫu nhiên dựa trên phân bố của

các nhãn lớp trong tập dữ liệu. Công thức tính entropy cho một tập dữ liệu D với n lớp được định nghĩa như sau:

Entropy(D) =

Trong đó:

* là xác suất (tần suất) của lớp thứ i trong tập dữ liệu D.
* n là số lượng lớp khác nhau trong tập dữ liệu D.

Entropy đạt giá trị cao nhất khi các lớp được phân phối đều nhau, tức là mức độ bất định cao nhất. Ngược lại, entropy bằng 0 khi tất cả các phần tử thuộc về một lớp duy nhất, tức là không có bất định nào.

Việc sử dụng entropy trong cây quyết định giúp chọn thuộc tính phân tách tốt nhất bằng cách giảm thiểu mức độ bất định sau khi phân tách. Thuộc tính nào giảm entropy nhiều nhất sẽ được chọn để phân tách tại mỗi nút.

**Information Gain** (IG) là một chỉ số dùng để chọn thuộc tính tốt nhất để phân tách dữ liệu tại mỗi nút trong cây quyết định. IG đo lường sự giảm thiểu bất định (entropy) khi phân tách dữ liệu dựa trên một thuộc tính cụ thể. Công thức tính IG dựa trên entropy như sau:

IG(D,A) =

Trong đó:

* + IG(D,A) là Information Gain khi phân tách tập dữ liệu D dựa trên thuộc tính A.
  + Entropy(D) là entropy của toàn bộ tập dữ liệu D.
  + Values(A) là tập hợp các giá trị có thể có của thuộc tính A.
  + là tập con của D mà thuộc tính A có giá trị bằng v.
  + là số lượng phần tử trong tập con .
  + là số lượng phần tử trong toàn bộ tập dữ liệu D.

Chỉ số **Gini** (Gini impurity) là một chỉ số khác được sử dụng để đo lường mức độ tinh khiết của một tập dữ liệu và giúp chọn thuộc tính tốt nhất để phân tách tại mỗi nút trong cây quyết định. Công thức tính Gini impurity cho một tập dữ liệu D với n lớp được định nghĩa như sau:

Gini(D) = 1

Trong đó:

* ​ là xác suất (tần suất) của lớp thứ i trong tập dữ liệu D.
* n là số lượng lớp khác nhau trong tập dữ liệu D.

Giống như Entropy, Gini impurity đạt giá trị cao nhất khi các lớp được phân phối đều nhau và giá trị thấp nhất khi tất cả các phần tử thuộc về một lớp duy nhất.

### Các phiên bản của thuật toán Decision Tree

**ID3 (Iterative Dichotomiser 3)**

* + Mô tả: ID3 là một trong những thuật toán cây quyết định đầu tiên, phát triển bởi Ross Quinlan vào những năm 1980. ID3 sử dụng lợi ích thông tin (Information Gain) để chọn thuộc tính phân tách.
  + Đặc điểm:
* Dễ hiểu và dễ triển khai.
* Chỉ sử dụng cho dữ liệu phân loại.
* Nhược điểm:
* Dễ bị overfitting.
* Không hỗ trợ dữ liệu liên tục (phải được rời rạc hóa trước khi sử dụng).

**C4.5**

* Mô tả: Cũng được phát triển bởi Ross Quinlan như một cải tiến của ID3. C4.5 khắc phục nhiều nhược điểm của ID3.
* Đặc điểm:
* Hỗ trợ cả dữ liệu phân loại và liên tục.
* Sử dụng tỷ lệ lợi ích (Gain Ratio) để chọn thuộc tính phân tách.
* Xử lý các giá trị thiếu (missing values).
* Pruning cây để tránh overfitting.
* Nhược điểm:
* Có thể khá phức tạp và tốn thời gian tính toán.

**CART (Classification and Regression Trees)**

* Mô tả: Được phát triển bởi Breiman và các cộng sự, CART có thể được sử dụng cho cả bài toán phân loại và hồi quy.
* Đặc điểm:
* Sử dụng chỉ số Gini (Gini Impurity) cho phân loại và độ lệch bình phương trung bình (Mean Squared Error) cho hồi quy.
* Pruning cây để tránh overfitting.
* Nhược điểm:
* Có thể tạo ra các cây phức tạp nếu không pruning đúng cách.

**CHAID (Chi-squared Automatic Interaction Detector)**

* Mô tả: CHAID sử dụng kiểm định Chi-squared để xác định phân tách tốt nhất tại mỗi nút.
* Đặc điểm:
* Thích hợp cho cả bài toán phân loại và hồi quy.
* Tạo ra các cây quyết định không nhị phân (các nút có thể có nhiều nhánh hơn).
* Nhược điểm:
* Có thể tạo ra cây phức tạp và khó hiểu.

### Ứng dụng của Decision Tree

Các thuật toán cây quyết định được sử dụng rộng rãi trong một loạt các ứng dụng học máy, bao gồm:

* Phân loại (Classification):
* Phát hiện gian lận trong giao dịch tài chính.
* Chẩn đoán y khoa, phân loại bệnh dựa trên các triệu chứng.
* Phân loại email vào các thư mục khác nhau, như thư rác và thư quan trọng.
* Hồi quy (Regression):
* Dự đoán giá nhà dựa trên các đặc điểm như diện tích, vị trí, số phòng.
* Dự đoán doanh thu bán hàng dựa trên các yếu tố thị trường, quảng cáo.
* Dự đoán nhu cầu năng lượng hoặc tiêu thụ điện.
* Phân tích khách hàng (Customer Analysis):
* Phân đoạn thị trường, xác định nhóm khách hàng có hành vi mua sắm tương tự.
* Dự đoán khả năng khách hàng rời bỏ dịch vụ (churn prediction).
* Phân tích giá trị khách hàng để tối ưu hóa chiến lược marketing.

### Ưu điểm và nhược điểm của Decision Tree

Ưu điểm:

* Dễ hiểu và dễ triển khai:
* Kết quả của cây quyết định có thể được diễn giải dễ dàng bằng các biểu đồ cây.
* Quyết định tại mỗi nút rất trực quan, dễ dàng giải thích cho người không chuyên.
* Xử lý tốt cả dữ liệu phân loại và liên tục:
* Có thể áp dụng cho cả bài toán phân loại và hồi quy.
* Hỗ trợ dữ liệu với nhiều loại thuộc tính khác nhau.
* Không yêu cầu tiền xử lý dữ liệu nhiều:
* Không cần chuẩn hóa dữ liệu hoặc loại bỏ giá trị thiếu.
* Không bị ảnh hưởng nhiều bởi các thuộc tính không quan trọng.
* Xử lý tốt các tương tác phi tuyến tính giữa các thuộc tính:
* Có thể tìm ra các mối quan hệ phức tạp giữa các thuộc tính mà các mô hình tuyến tính có thể bỏ sót.

Nhược điểm:

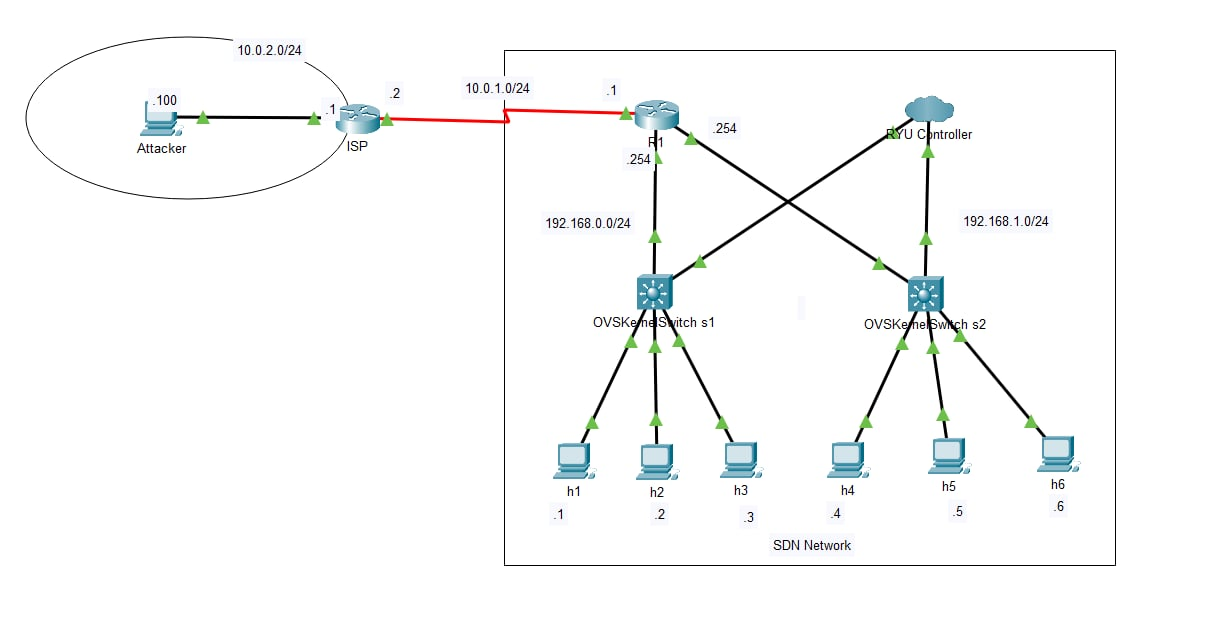
* Dễ bị overfitting:
* Cây quyết định có thể học quá mức các chi tiết của dữ liệu huấn luyện, dẫn đến hiệu suất kém trên dữ liệu mới.
* Cần phải áp dụng kỹ thuật pruning để giảm thiểu vấn đề này.
* Không ổn định:
* Một thay đổi nhỏ trong dữ liệu có thể dẫn đến cấu trúc cây rất khác nhau.
* Các thuộc tính khác nhau có thể gây ra các cây khác nhau khi huấn luyện lại trên cùng một dữ liệu.
* Hiệu suất kém khi xử lý các thuộc tính với giá trị nhỏ:
* Các thuộc tính với giá trị nhỏ có thể không được chọn làm thuộc tính phân tách chính.
* Đặc biệt khi có nhiều thuộc tính với giá trị không đồng đều.
* Khả năng tổng quát hóa kém:
* Khi dữ liệu có quá nhiều lớp hoặc giá trị, cây quyết định có thể không tổng quát hóa tốt.
* Điều này dẫn đến các quyết định sai lầm khi áp dụng trên dữ liệu mới.

# THỰC NGHIỆM

## Yêu cầu đặt ra

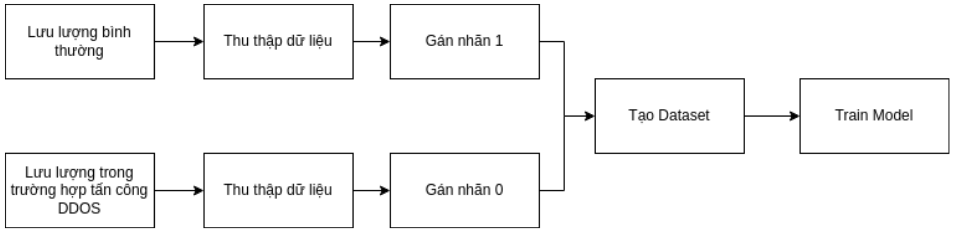
* Cài đặt: Mininet, Ubuntu 20.4.6, Ryu, Python 3, Git
* Yêu cầu về mạng :

## Thiết kế hệ thống



Hình 5: Thiết kế hệ thống SDN

## Thu thập dữ liệu



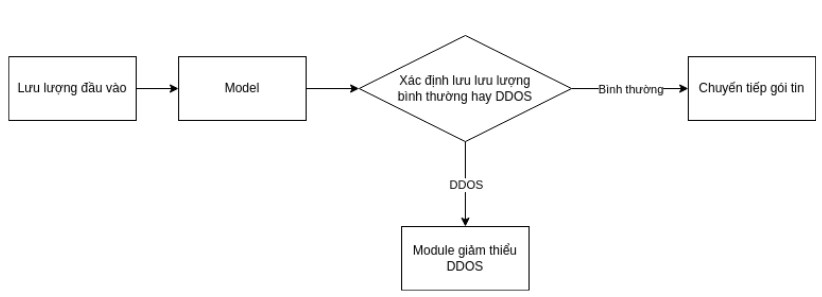
Hình 6: Sơ đồ quá trình thu thập dataset

Hình 5 mô tả quá trình thu thập dữ liệu để xây dựng dataset, dữ liệu được thu thập trong 2 loại lưu lượng là bình thường và lưu lượng trong trường hợp tấn công DDoS. Dữ liệu thu thập từ 2 loại lưu lượng được tiến hành trích xuất đặc trưng, các đặc trưng được mô tả như bảng 3, nhãn 1 sẽ được gán cho lưu lượng bình thường, nhãn 0 sẽ được gán cho trường hợp có DDoS. Dataset này sau đó được sử dụng để huấn luyện mô hình máy học tại khối "Train Model". Mô hình được huấn luyện có thể được sử dụng để phân loại và phát hiện tấn công DDoS dựa trên dữ liệu lưu lượng mạng. Quy trình này thể hiện một hệ thống phát hiện tấn công DDoS bằng cách sử dụng kỹ thuật học máy có giám sát, trong đó dữ liệu bình thường và dữ liệu tấn công được gán nhãn khác nhau để huấn luyện mô hình phân loại.

|  |  |
| --- | --- |
| **Tên trường** | **Mô tả** |
| timestamp | Thời gian thu thập thống kê, ở định dạng timestamp (số giây từ thời điểm bắt đầu của epoch). |
| datapath\_id | ID của datapath (switch OpenFlow) mà thống kê được thu thập từ đó. |
| flow\_id | ID được tính toán dựa trên thông tin của luồng, bao gồm địa chỉ IP nguồn, cổng nguồn, địa chỉ IP đích, cổng đích và giao thức IP. |
| ip\_src | Địa chỉ IP nguồn của luồng. |
| tp\_src | Cổng nguồn của luồng (TCP hoặc UDP). |
| ip\_dst | Địa chỉ IP đích của luồng. |
| tp\_dst | Cổng đích của luồng (TCP hoặc UDP). |
| ip\_proto | Giao thức IP của luồng (ví dụ: 1 cho ICMP, 6 cho TCP, 17 cho UDP). |
| icmp\_code | Mã ICMP (nếu giao thức IP là ICMP). |
| icmp\_type | Loại ICMP (nếu giao thức IP là ICMP). |
| flow\_duration\_sec | Thời gian tồn tại của luồng (tính bằng giây). |
| flow\_duration\_nsec | Thời gian tồn tại của luồng (tính bằng nano giây). |
| idle\_timeout | Giá trị idle timeout của luồng (thời gian tối đa luồng không hoạt động trước khi bị xóa). |
| hard\_timeout | Giá trị hard timeout của luồng (thời gian tối đa luồng tồn tại trước khi bị xóa). |
| flags | Các cờ liên quan đến luồng. |
| packet\_count | Số lượng gói tin trong luồng. |
| byte\_count | Lượng dữ liệu (tính bằng byte) trong luồng. |
| packet\_count\_per\_second | Tốc độ gói tin mỗi giây của luồng. |
| packet\_count\_per\_nsecond | Tốc độ gói tin mỗi nano giây của luồng. |
| byte\_count\_per\_second | Tốc độ dữ liệu mỗi giây của luồng (tính bằng byte/giây). |
| byte\_count\_per\_nsecond | Tốc độ dữ liệu mỗi nano giây của luồng (tính bằng byte/nano giây). |
| label | Được sử dụng để gán nhãn cho các luồng khi sử dụng dữ liệu cho mục đích huấn luyện mô hình ML. |

Bảng 3: Mô tả các trường của dataset

## Mô tả cách hoạt động của quá trình thực nghiệm



Hình 7: Sơ đồ tổng quan cách hoạt động của quá trình thực nghiệm

Sau khi xây dựng Dataset và lựa chọn Model, quá trình xử lý lưu lượng đầu

vào của hệ thống được mô tả như hình 6. Quá trình xử lý diễn ra như sau:

* Lưu lượng đầu vào: Lưu lượng truy cập hệ thống từ các nguồn khác nhau được nhận vào khối "Lưu lượng đầu vào". Thông tin về luồng dữ liệu (flow traffic) từ flow table của từng switch được thu thập liên tục. Các thông tin này sẽ được trích xuất đặc trưng để đưa vào mô hình phân loại dự đoán xác định loại lưu lượng là lưu lượng bình thường hay là lưu lượng DDoS.
* Mô hình phân tích: Lưu lượng đầu vào sau khi trích xuất đặc trưng được đưa vào bộ phân loại dựa trên thuật toán Decision Tree, sử dụng chỉ số ’entropy’ để xác định phương pháp tính độ đo tính tạp (impurity) trong quá trình chia nút của cây quyết định. Giá trị ’entropy’ có nghĩa là sử dụng tiêu chuẩn entropy (entropy thông tin) để tính toán độ đo tính tạp.
* Xác định lưu lượng bình thường hay DDoS: Bộ phân loại sẽ dự đoán lưu lượng đầu vào và xác định xem đó là lưu lượng bình thường hay lưu lượng tấn công DDoS.
* Lưu lượng bình thường: Nếu lưu lượng được xác định là bình thường, nó sẽ được chuyển tiếp đến "Chuyển tiếp gói tin" để xử lý tiếp theo trong hệ thống.
* Lưu lượng DDoS: Nếu lưu lượng được xác định là tấn công DDoS, nó sẽ được chuyển đến "Module giảm thiểu DDoS". Module này có chức năng block\_port là thêm một luồng (flow) mới vào bảng luồng của switch để chặn tất cả lưu lượng qua một cổng nhất định trong một khoảng thời gian được thiết lập sẵn. Hàm này được sử dụng để ngăn chặn tấn công từ chối dịch vụ (DoS) bằng cách chặn cổng đang nhận lưu lượng tấn công.

# KẾT LUẬN

## Kết luận

## Hướng phát triển

# TÀI LIỆU THAM KHẢO

Tiếng Việt

…

Tiếng Anh

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-term Memory. *Neural Computation*, *9*, 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2023). *Attention Is All You Need* (arXiv:1706.03762). arXiv. https://doi.org/10.48550/arXiv.1706.03762