

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



**MẠC GIA HUY - 52100694
HUỖNH MINH PHƯỚC - 52100465**

SOLUTION TO DETECT DDOS ATTACKS IN SDN NETWORK

DỰ ÁN CÔNG NGHỆ THÔNG TIN

**MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG
DỮ LIỆU**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2025

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



**MẠC GIA HUY - 52100694
HUỖNH MINH PHƯỚC - 52100465**

SOLUTION TO DETECT DDOS ATTACKS IN SDN NETWORK

DỰ ÁN CÔNG NGHỆ THÔNG TIN

MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG DỮ LIỆU

**Người hướng dẫn
TS. Trương Đình Tú**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2025

LỜI CẢM ƠN

Trước hết, chúng em muốn bày tỏ lòng biết ơn sâu sắc đến thầy Trương Đình Tú, người đã luôn tận tình hướng dẫn và hỗ trợ chúng em trong suốt quá trình nghiên cứu cũng như hoàn thành bài báo cáo.

Chúng em xin được dành lời cảm ơn chân thành cho khoa Công Nghệ Thông Tin đã cho chúng em được tiếp cận môn dự án công nghệ thông tin mà theo chúng em là vô cùng cần thiết.

Chúng em đã nỗ lực và cố gắng tích lũy những kiến thức mà thầy đã truyền đạt trong suốt thời gian vừa qua để hoàn thành được bài báo cáo một cách hoàn thiện nhất trong khả năng của chúng em. Nếu có xảy ra sai sót gì, chúng em rất mong nhận được sự góp ý của thầy và khoa để chúng em có thể hoàn thiện bản thân hơn.

Chúng em xin chân thành cảm ơn, xin chúc những điều tốt đẹp nhất sẽ luôn đồng hành cùng mọi người.

TP. Hồ Chí Minh, ngày 10 tháng 2 năm 2025.

Tác giả

(Ký tên và ghi rõ họ tên)

Phước Huy

Huỳnh Minh Phước Mạc Gia Huy

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 10 tháng 2 năm 2025

Tác giả

(Ký tên và ghi rõ họ tên)

Phước

Huy

Huỳnh Minh Phước Mạc Gia Huy

SOLUTION TO DETECT DDoS ATTACKS IN SDN NETWORK

TÓM TẮT

Phát hiện tấn công DDoS trong SDN là một vấn đề quan trọng để đảm bảo an toàn mạng.

Nghiên cứu này áp dụng các thuật toán học máy để phân loại lưu lượng mạng, giúp nhận diện các cuộc tấn công DDoS. Dữ liệu được thu thập, tiền xử lý và huấn luyện trên nhiều mô hình khác nhau.

Kết quả cho thấy mô hình đề xuất đạt độ chính xác cao, cải thiện đáng kể so với các phương pháp truyền thống.

Việc so sánh với các phương pháp hiện có khẳng định tính hiệu quả và tiềm năng triển khai thực tế của mô hình.

SOLUTION TO DETECT DDOS ATTACKS IN SDN NETWORK

ABSTRACT

In recent years, Software-Defined Networking (SDN) has emerged as an effective approach to managing and optimizing network operations. However, its centralized architecture makes it vulnerable to Distributed Denial-of-Service (DDoS) attacks, which can severely impact network performance. This study presents a machine learning-based solution for detecting DDoS attacks in SDN environments.

The proposed approach involves collecting network traffic data, preprocessing it, and applying the Decision Tree algorithm to classify normal and malicious traffic. Several evaluation metrics, including accuracy, precision, recall, and F1-score, are used to assess the model's effectiveness. Experimental results demonstrate that the proposed method achieves high accuracy and can effectively detect attacks in real time.

This research contributes to enhancing security in SDN networks by leveraging machine learning techniques to improve threat detection capabilities. Future work may explore the integration of more advanced models and adaptive security mechanisms to further strengthen network defenses against evolving cyber threats.

MỤC LỤC

DANH MỤC HÌNH VẼ	ix
DANH MỤC BẢNG BIỂU	xi
DANH MỤC CÁC CHỮ VIẾT TẮT.....	xii
CHƯƠNG 1. MỞ ĐẦU VÀ TỔNG QUAN ĐỀ TÀI.....	1
1.1 Lý do chọn đề tài.....	1
1.2 Mục tiêu thực hiện đề tài.....	1
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT.....	3
2.1 Mạng điều khiển bằng phần mềm (SDN)	3
2.1.1 Giới thiệu về mạng SDN	3
2.1.2 Kiến trúc của mạng SDN	3
2.1.2.1 Sự khác biệt giữa mạng SDN với mạng truyền thống...	3
2.1.2.2 Lớp điều khiển (Control Plane)	4
2.1.2.3 Lớp Cơ sở hạ tầng (Data Plane).....	5
2.1.2.4 Lớp Ứng dụng (Application Plane)	6
2.1.3 Các thành phần của flow table	6
2.1.4 Giao thức OpenFlow	7
2.1.5 Quy trình xử lý gói tin trong mạng SDN.....	8
2.1.6 Mối đe dọa và lỗ hổng bảo mật trong mạng SDN	8
2.1.7 Các hình thức tấn công và giải pháp tại từng tầng trong mạng SDN	9
2.1.8 Ưu nhược điểm của mạng SDN	11
2.2 Tấn công từ chối dịch vụ phân tán (DDoS) trong SDN.....	12
2.2.1 Giới thiệu về DDOS.....	12

2.2.2 Các loại tấn công DDOS	12
2.2.2.1 Low rate DDOS	13
2.2.2.2 High rate DDOS.....	16
2.2.3 Các phương pháp giảm thiểu DDOS trong mạng SDN.....	19
2.3 Phương pháp phát hiện DDOS sử dụng Machine Learning	21
2.3.1 Logistic Regression.....	22
2.3.1.1 Giới thiệu Logistic Regression	22
2.3.1.2 Cách hoạt động của Logistic Regression.....	23
2.3.2 Decision Tree	23
2.3.2.1 Giới thiệu Decision Tree.....	23
2.3.2.2 Cách hoạt động của Decision Tree	24
2.3.3 KNN (<i>K-nearest neighbors</i>).....	24
2.3.3.1 Giới thiệu KNN.....	24
2.3.3.2 Cách hoạt động của KNN	25
2.3.4 SVM (<i>Support Vector Machine</i>).....	25
2.3.4.1 Giới thiệu SVM.....	25
2.3.4.2 Cách hoạt động của SVM	25
CHƯƠNG 3. MÔ HÌNH ĐỀ XUẤT.....	27
3.1 Giới thiệu và đề xuất mô hình.....	27
3.2 Mô phỏng mô hình mạng	27
3.2.1 Mô hình mạng SDN.....	28
3.2.2 Mạng của Attacker.....	29
3.3 Mô hình phát hiện DDOS	29

3.3.1 Cấu trúc tổng thể	29
3.3.2 Thành phần chính	30
CHƯƠNG 4. THỰC NGHIỆM	31
4.1 Cài đặt môi trường thực nghiệm	31
4.1.1 Khởi tạo mô hình mạng SDN	31
4.1.1.1 Khởi tạo Base Ryu controller	31
4.1.1.2 Khởi tạo mô hình mạng	31
4.1.2 Thu thập dữ liệu thực nghiệm	32
4.1.2.1 Mô tả dữ liệu thực nghiệm.....	32
4.1.2.2 Khởi tạo lưu lượng bình thường	32
4.1.2.3 Khởi tạo lưu lượng DDOS	33
4.1.2.4 Lưu trữ kết quả dữ liệu	34
4.1.3 Tiền xử lý dữ liệu	Error! Bookmark not defined.
4.2 Huấn luyện và triển khai mô hình phát hiện DDOS	37
4.2.1 Chọn mô hình và phương pháp huấn luyện.....	37
4.2.1.1 Huấn luyện Logistic Regression	37
4.2.1.2 Huấn luyện Decision Tree	38
4.2.1.3 Huấn luyện KNN	39
4.2.1.4 Huấn luyện SVM	39
4.2.2 Phát hiện tấn công trong thời gian thực.....	40
4.2.2.1 Kiểm tra lưu lượng bình thường	40
4.2.2.2 Kiểm tra lưu lượng DDOS.....	41
4.2.2.3 Áp dụng mô hình Decion tree vào Ryu Controller	43

4.3 Thử nghiệm biện pháp phòng chống DDOS.....	44
4.4 Đánh giá và phân tích kết quả	45
4.4.1 Tiêu chí đánh giá	45
4.4.2 So sánh kết quả huấn luyện giữa các mô hình học máy	45
CHƯƠNG 5. KẾT LUẬN.....	48
5.1 Kết luận	48
5.2 Hướng phát triển	48
TÀI LIỆU THAM KHẢO	49

DANH MỤC HÌNH VẼ

Hình 2.1 Sự khác nhau giữa mạng SDN và mạng truyền thống	3
Hình 2.2 Kiến trúc mạng SDN	4
Hình 2.3 Các loại tấn công DDOS	13
Hình 2.4 Packet-In Saturation	14
Hình 2.5 IP Spoofing Attack	15
Hình 2.6 ARP Poisoning / Spoofing	16
Hình 2.7 HTTP Flood	17
Hình 2.8 UDP/TCP SYN Flood	18
Hình 2.9 BGP Hijacking	19
Hình 2.10 Các phương pháp giảm thiểu DDOS trong mạng SDN	19
Hình 2.11 Các phương pháp Machine Learning phát hiện DDOS	22
Hình 3.1 Mô hình mạng tổng quát	27
Hình 4.1 Khởi tạo mininet	32
Hình 4.2 Khởi tạo và thu thập lưu lượng bình thường	33
Hình 4.3 Khởi tạo và thu thập lưu lượng DDOS	34
Hình 4.4 Kết quả thu thập dữ liệu bình thường và DDOS	34
Hình 4.5 Tiền xử lý dữ liệu	37
Hình 4.6 Huấn luyện Logistic Regression	38
Hình 4.7 Huấn luyện Decision Tree	38
Hình 4.8 Huấn luyện KNN	39
Hình 4.9 Huấn luyện SVM	40
Hình 4.10 Lưu lượng bình thường	40

Hình 4.11 Biểu đồ phân tích lưu lượng bình thường	41
Hình 4.12 Lưu lượng DDOS	42
Hình 4.13 Biểu đồ phân tích lưu lượng DDOS	43
Hình 4.14 Controller phát hiện DDOS	44
Hình 4.15 Block port chống DDOS	44
Hình 4.16 Chỉ số đánh giá giữa các mô hình	46
Hình 4.17 Accuracy giữa các mô hình	47

DANH MỤC BẢNG BIỂU

Bảng 4.1 Mô tả các đặc trưng của dữ liệu thu thập.....	36
--	----

DANH MỤC CÁC CHỮ VIẾT TẮT

DDOS	Distributed Denial of Service
SDN	Software Defined Networking
ODL	Open Day Light
ONOS	Open Network Operating System
API	Application Programming Language
KNN	K-nearest neighbors
SVM	Support Vector Machine

CHƯƠNG 1. MỞ ĐẦU VÀ TỔNG QUAN ĐỀ TÀI

1.1 Lý do chọn đề tài

Mạng điều khiển bằng phần mềm (SDN) mang lại nhiều lợi ích trong việc quản lý linh hoạt và tối ưu hóa tài nguyên mạng. Tuy nhiên, do kiến trúc tập trung của SDN, hệ thống dễ trở thành mục tiêu của các cuộc tấn công từ chối dịch vụ phân tán (DDoS), gây ảnh hưởng nghiêm trọng đến hiệu suất hoạt động. Các phương pháp phát hiện tấn công truyền thống, chẳng hạn như dựa trên quy tắc hoặc chữ ký, thường không đủ hiệu quả trước sự phát triển phức tạp và liên tục thay đổi của các kiểu tấn công mới.

Vì vậy, nghiên cứu và đề xuất giải pháp phát hiện DDoS trong SDN là một yêu cầu cấp thiết nhằm bảo đảm an toàn mạng. Việc ứng dụng các phương pháp tiên tiến, đặc biệt là học máy, giúp nâng cao khả năng nhận diện các dấu hiệu bất thường trong lưu lượng mạng, đồng thời thích ứng tốt với những biến thể mới của tấn công. Đề tài này không chỉ có giá trị nghiên cứu mà còn mang tính ứng dụng cao, góp phần cải thiện khả năng bảo vệ hệ thống SDN trước các mối đe dọa an ninh mạng ngày càng gia tăng.

1.2 Mục tiêu thực hiện đề tài

Đề tài hướng đến việc nghiên cứu và đề xuất giải pháp nhằm phát hiện hiệu quả các cuộc tấn công từ chối dịch vụ phân tán (DDoS) trong mạng điều khiển bằng phần mềm (SDN). Cụ thể, các mục tiêu chính bao gồm:

- Tìm hiểu tổng quan về SDN và các lỗ hổng bảo mật liên quan, đặc biệt là các hình thức tấn công DDoS có thể khai thác kiến trúc tập trung của SDN.
- Phân tích các phương pháp phát hiện tấn công DDoS hiện có, bao gồm các cách tiếp cận truyền thống (dựa trên chữ ký, quy tắc) và hiện đại (học máy, học sâu).
- Đề xuất mô hình phát hiện tấn công DDoS phù hợp với môi trường SDN, đảm bảo khả năng nhận diện chính xác các cuộc tấn công với độ trễ thấp.

- Thử nghiệm và đánh giá hiệu suất của mô hình đề xuất, thông qua việc áp dụng trên tập dữ liệu mạng thực tế hoặc mô phỏng.
- So sánh mô hình với các phương pháp hiện có, từ đó rút ra những ưu điểm, hạn chế và đề xuất hướng cải thiện trong tương lai.

Mục tiêu chính là xây dựng một giải pháp hiệu quả, ứng dụng cao, góp phần nâng cao khả năng bảo vệ mạng SDN trước các mối đe dọa từ tấn công DDoS.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

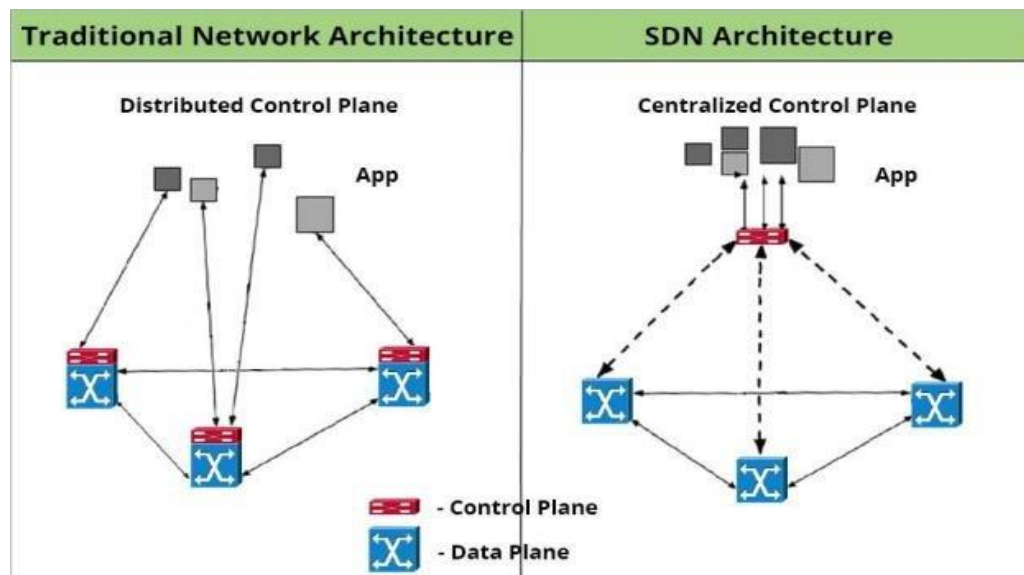
2.1 Mạng điều khiển bằng phần mềm (SDN)

2.1.1 Giới thiệu về mạng SDN

Mạng điều khiển bằng phần mềm (Software-Defined Networking - SDN) là một kiến trúc mạng tiên tiến, giúp tối ưu hóa quản lý, giảm chi phí và tăng khả năng mở rộng. Điểm đặc trưng của SDN là sự tách biệt các chức năng quản lý mạng ra khỏi phần cứng, cho phép điều khiển thông qua phần mềm. Kiến trúc này chia hệ thống thành hai phần chính: lớp điều khiển (Control Plane), chịu trách nhiệm đưa ra quyết định về định tuyến và quản lý mạng, và lớp dữ liệu (Data Plane), thực hiện chức năng chuyển tiếp gói tin. Sự phân tách này giúp SDN trở nên linh hoạt hơn, dễ dàng thích ứng trong bối cảnh nhu cầu ngày càng gia tăng.

2.1.2 Kiến trúc của mạng SDN

2.1.2.1 Sự khác biệt giữa mạng SDN với mạng truyền thống



Hình 2.1 Sự khác nhau giữa mạng SDN và mạng truyền thống

(Nguồn: Rahim, M. K. A., Habaebi, M. H., & et al., 2019. *Performance Analysis of SDN Controllers: POX, Floodlight, and OpenDaylight*. ResearchGate.)

Mạng SDN phân tách việc lập trình khỏi phần cứng, điều này mang lại sự linh hoạt cao và khả năng quản lý mạng hiệu quả. Việc truyền tải thông tin trong SDN được thực hiện theo cách tập trung hóa, qua đó, quá trình điều khiển mạng có thể được lập trình trên lớp điều khiển. Kiến trúc của SDN tuân theo mô hình ba lớp, bao gồm lớp ứng dụng, lớp điều khiển và lớp dữ liệu, giúp tách biệt rõ ràng các chức năng mạng. Trong mô hình này, các quyết định về định tuyến không được thực hiện trực tiếp trên mặt phẳng dữ liệu mà được chuyển giao sang mặt phẳng điều khiển. Mặt phẳng điều khiển có trách nhiệm đưa ra các quyết định về đường dẫn cho dữ liệu, trong khi mặt phẳng dữ liệu chỉ thực hiện việc chuyển tiếp dữ liệu một cách đơn giản qua các thiết bị như bộ định tuyến và bộ chuyển mạch, theo các quyết định đã được đưa ra.

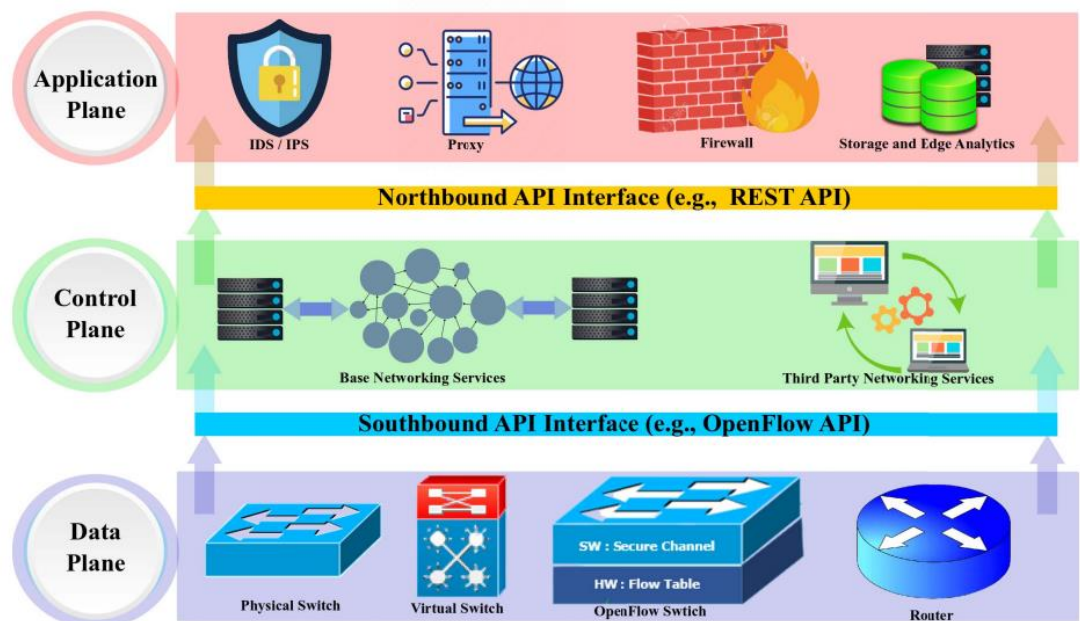


FIGURE 3. The layers of SDN infrastructure and its components.

Hình 2.2 Kiến trúc mạng SDN

(Nguồn: Rahouti, M., Xiong, K., Xin, Y., Jagatheesaperumal, S. K., Ayyash, M., & Shaheed, M. (2022). *SDN security review: Threat taxonomy, implications, and open challenges*. IEEE Access, 10, 43176-43203.)

2.1.2.2 Lớp điều khiển (Control Plane)

Lớp Điều khiển đóng vai trò là "bộ não" của mạng SDN, trong đó Bộ điều khiển SDN chịu trách nhiệm đưa ra các quyết định định tuyến. Các nhà cung cấp khác nhau đã phát triển và cung cấp các bộ điều khiển SDN, với những đại diện nổi bật như Open Daylight (ODL), Hệ điều hành mạng mở (ONOS), Hệ điều hành mạng (NOX), Hệ thống điều hành Python (POX), Beacon và Ryu. Trong số đó, ODL và ONOS là hai bộ điều khiển phổ biến và được ưa chuộng nhất trên thị trường hiện nay.

Bộ điều khiển ODL sử dụng giao thức OpenFlow để thiết lập giao tiếp giữa Mặt phẳng Dữ liệu và Mặt phẳng Điều khiển của SDN. Các quyết định định tuyến chính được đưa ra bởi lớp Điều khiển, trong khi lớp Dữ liệu chỉ đảm nhận vai trò chuyển tiếp dữ liệu đến các cổng cụ thể, theo các quy tắc đã được xác định. Bộ điều khiển SDN còn cung cấp các thông tin quan trọng về tình trạng và hiệu suất của mạng, chẳng hạn như số lượng gói tin, lưu lượng mạng, các luồng dữ liệu (flow), thiết bị trong mạng, và trạng thái các kết nối.

Điều đặc biệt của SDN là khả năng thay đổi chính sách mạng mà không cần lo lắng về phần cứng bên dưới, nhờ vào giao diện lập trình có thể cấu hình. Quản trị viên mạng có thể điều chỉnh các chính sách mạng thông qua giao diện lập trình này một cách dễ dàng. Giao diện Lập trình Ứng dụng (API) SouthBound đóng vai trò như một cầu nối giữa Bộ điều khiển SDN và Mặt phẳng Dữ liệu, cho phép giao tiếp giữa các phần này. Giao tiếp này được thực hiện thông qua Giao thức OpenFlow, qua đó, Bộ điều khiển có thể trực tiếp cập nhật các quy tắc bảng luồng trên bộ chuyển mạch OpenFlow của Mặt phẳng Dữ liệu.

2.1.2.3 Lớp Cơ sở hạ tầng (Data Plane)

Lớp Dữ liệu, lớp thấp nhất trong kiến trúc SDN, bao gồm các bộ chuyển mạch, được gọi chung là Bộ chuyển mạch. Các bộ chuyển mạch OpenFlow chứa bảng luồng (flow table), nơi lưu trữ các quy tắc định tuyến dữ liệu. Bộ chuyển mạch OpenFlow chỉ gửi dữ liệu đến cổng đích theo các quy tắc đã được xác định trong bảng luồng, đảm bảo độ chính xác và hiệu quả trong việc truyền thông tin.

Điều quan trọng là, bộ chuyển mạch OpenFlow không tự quyết định các quy tắc này mà chỉ nhận hướng dẫn từ Bộ điều khiển SDN. Giao tiếp giữa Bộ điều khiển

SDN và bộ chuyển mạch OpenFlow được thực hiện một cách an toàn thông qua Giao thức OpenFlow, giúp đảm bảo tính bảo mật và tính toàn vẹn của dữ liệu. Đặc biệt, Giao thức OpenFlow sử dụng Lớp Ổ cắm Bảo mật (SSL) để bảo vệ dữ liệu trong quá trình giao tiếp, ngăn ngừa sự xâm nhập và bảo vệ các thông tin quan trọng trong mạng.

2.1.2.4 Lớp Ứng dụng (Application Plane)

Lớp Ứng dụng SDN, lớp trên cùng trong kiến trúc SDN, bao gồm các ứng dụng mạng quan trọng như Hệ thống Phát hiện Xâm nhập (IDS), Hệ thống Ngăn chặn Xâm nhập (IPS), tường lửa, và các ứng dụng quản lý đám mây. Ngoài ra, lớp này còn chứa các ứng dụng như cân bằng tải, thực thi chính sách, trình xem topo, tự động hóa mạng và các công cụ quản lý mạng.

Giao tiếp giữa Lớp Ứng dụng và Lớp Điều khiển được thực hiện thông qua Giao diện Hướng Bắc (NorthBound Interface). Thông qua lớp ứng dụng, quản trị viên mạng có thể lập trình Bộ điều khiển SDN để thực hiện các chính sách quản lý mới, đồng thời điều khiển các bộ chuyển mạch một cách tập trung. Các nhà cung cấp mạng cũng có thể bổ sung các ứng dụng mới vào lớp ứng dụng, sử dụng các số liệu thống kê mạng, topo mạng và thông tin về trạng thái mạng.

Giao diện Lập trình Ứng dụng NorthBound hoạt động như cầu nối giữa các ứng dụng SDN và Lớp Điều khiển, cho phép các ứng dụng gửi yêu cầu đến bộ điều khiển. Thông qua Giao diện Người dùng Đồ họa (GUI), quản trị viên mạng có thể truy cập API NorthBound, yêu cầu bộ điều khiển thay đổi cấu hình mạng, áp dụng các chính sách bảo mật, hoặc cung cấp thông tin về hiệu suất và trạng thái mạng.

2.1.3 Các thành phần của flow table

Flow table là một bảng trong bộ chuyển mạch (switch) SDN, nơi lưu trữ các mục nhập (flow entries) để xử lý lưu lượng mạng. Mỗi mục nhập trong flow table bao gồm các thành phần như sau:

Trường khớp (Match Fields): Đây là các tiêu chí dùng để so sánh gói tin với các mục nhập trong bảng lưu lượng. Các trường này có thể bao gồm:

- Địa chỉ MAC nguồn và đích.
- Địa chỉ IP nguồn và đích.
- Số cổng giao thức TCP/UDP.
- Loại giao thức (ví dụ: TCP, UDP, ICMP).
- VLAN ID (nếu có sử dụng VLAN).
- Độ ưu tiên của luồng.

Khi một gói tin đi qua bộ chuyển mạch, nó sẽ được so sánh với các trường khớp để xác định hành động cần thực hiện.

Hành động (Action): Nếu một gói tin khớp với mục nhập trong bảng lưu lượng, bộ chuyển mạch sẽ thực hiện các hành động đã được xác định. Các hành động có thể bao gồm:

- Chuyển tiếp (Forward) gói tin đến một cổng cụ thể.
- Drop (loại bỏ) gói tin.
- Modify (sửa đổi) các trường trong gói tin, ví dụ như thay đổi địa chỉ MAC hoặc IP.
- Redirect (chuyển hướng) gói tin đến bộ điều khiển.

Bộ đếm (Counters): Các bộ đếm lưu trữ thông tin về số lượng gói tin hoặc byte đã khớp với mục nhập trong bảng lưu lượng. Dữ liệu này rất hữu ích cho việc giám sát và thu thập số liệu thống kê mạng.

Độ ưu tiên (Priority): Mỗi mục nhập trong bảng lưu lượng có một mức độ ưu tiên riêng. Nếu một gói tin khớp với nhiều mục nhập, mục nhập có độ ưu tiên cao hơn sẽ được áp dụng để xử lý gói tin.

2.1.4 Giao thức *OpenFlow*

OpenFlow là một giao thức điều khiển trong hệ thống mạng SDN, có chức năng kết nối bộ điều khiển với các bộ chuyển mạch hỗ trợ OpenFlow. Giao thức này cho phép bộ điều khiển điều chỉnh và quản lý các bảng lưu lượng (flow table) của các thiết bị chuyển mạch. Thông qua OpenFlow, controller có thể thay đổi các mục nhập trong bảng lưu lượng, thiết lập các quy tắc xử lý gói tin hoặc thay đổi các hành động đối với các gói tin. Đồng thời, OpenFlow cũng hỗ trợ bộ điều khiển theo dõi lưu

lượng mạng và thực hiện các điều chỉnh trong mạng mà không cần can thiệp vào phần cứng của thiết bị chuyển mạch. Cấu trúc của giao thức này hoạt động theo các bước sau:

- Packet-In: Khi bộ chuyển mạch nhận một gói tin mà không có quy tắc phù hợp trong bảng luồng (flow table), nó sẽ gửi gói tin đó đến bộ điều khiển để yêu cầu quyết định xử lý.
- Flow-Mod: Bộ điều khiển nhận thông tin từ bộ chuyển mạch và đưa ra quy tắc xử lý gói tin, như quyết định chuyển tiếp hoặc bỏ qua gói tin.
- Packet-Out: Sau khi nhận được chỉ thị từ bộ điều khiển, bộ chuyển mạch thực hiện việc chuyển tiếp gói tin đến thiết bị đích hoặc tiếp tục gửi nó về bộ điều khiển.

2.1.5 Quy trình xử lý gói tin trong mạng SDN

Khi một gói tin bắt nguồn từ lớp ứng dụng, quá trình xử lý diễn ra như sau:

1. Khi một gói tin yêu cầu từ địa chỉ IP bên ngoài đến bộ chuyển mạch OpenFlow, bộ chuyển mạch sẽ kiểm tra bảng luồng của nó để tìm quy tắc (flow rule) phù hợp với gói tin.
2. Nếu tìm thấy quy tắc phù hợp, gói tin sẽ được chuyển đến cổng đích đã chỉ định.
3. Nếu không tìm thấy quy tắc phù hợp, bộ chuyển mạch sẽ tạo ra một thông báo Packet-In (chứa thông tin về gói tin) và gửi yêu cầu bộ điều khiển xử lý thêm thông qua giao diện Southbound.
4. Bộ điều khiển sau đó sẽ quyết định cách thức xử lý và gửi lại quy tắc bảng luồng mới cho bộ chuyển mạch OpenFlow.
5. Bộ chuyển mạch OpenFlow sau đó sẽ chuyển gói tin đến cổng đích theo chỉ dẫn từ bộ điều khiển.

2.1.6 Môi đe dọa và lỗ hổng bảo mật trong mạng SDN

SDN mang lại nhiều lợi ích cho người quản trị mạng, nhưng cũng tạo ra những cơ hội cho các cuộc tấn công vào mạng. Việc tập trung vào lớp điều khiển (Control

Plane) của SDN, mặc dù đem lại sự linh hoạt, nhưng cũng tạo ra điểm yếu duy nhất là nếu lớp điều khiển bị xâm nhập, toàn bộ mạng có thể bị gián đoạn. Trái ngược với mạng truyền thống, nơi các bộ định tuyến và bộ chuyển mạch được phân tán và bảo vệ riêng biệt, SDN tập trung các chức năng quan trọng tại một điểm, tạo ra một mục tiêu hấp dẫn cho các cuộc tấn công mạng.

Trong mạng truyền thống, các lớp dữ liệu và điều khiển được tích hợp, điều này giúp tăng cường bảo mật và hiệu suất mạng. Đặc biệt, khi thay đổi chính sách mạng, các thiết bị sẽ được cấu hình lại riêng lẻ, điều này tạo ra một rào cản đối với các cuộc tấn công từ bên ngoài. Ngược lại, trong SDN, khi lớp điều khiển bị xâm phạm, toàn bộ mạng có thể bị ảnh hưởng, làm giảm tính bảo mật và làm tăng khả năng bị tấn công. Mạng truyền thống, nhờ vào cấu trúc phân tán, khó bị tấn công hơn, vì các bộ định tuyến, bộ chuyển mạch và cổng phải bị tấn công riêng biệt, làm giảm nguy cơ tổn hại cho toàn bộ hệ thống.

2.1.7 Các hình thức tấn công và giải pháp tại từng tầng trong mạng SDN

- **Tầng Điều Khiển (Control Plane)**
- **Các Hình Thức Tấn Công:**
 - Chiếm Quyền Vị Trí Máy Chủ (Host Location Hijacking): Kẻ xâm nhập có thể giả mạo địa chỉ MAC để kiểm soát vị trí của bộ điều khiển.
 - Giả Mạo Liên Kết (Link Fabrication Attack): Tạo các liên kết giả để thao túng việc truyền tải dữ liệu.
 - Tấn Công Man-in-the-Middle (MITM): Chặn và sửa đổi lưu lượng truyền giữa bộ điều khiển và các thiết bị trong mạng.
 - Tấn Công Từ Chối Dịch Vụ (DoS/DDoS): Tấn công khiến bộ điều khiển bị quá tải bởi lượng lớn lưu lượng giả.
- **Giải Pháp:**
 - Giám Sát Học Máy: Áp dụng các phương pháp học máy giám sát để nhận diện và ngăn chặn chiếm quyền máy chủ.

- Phân Tích và Phương Pháp Chính Thức: Sử dụng các công cụ phân tích độ trễ và phương pháp kiểm tra chính thức để phát hiện liên kết giả.
- Bảo Mật Qua Mã Hóa và Xác Thực: Thực thi các cơ chế mã hóa như TLS và sử dụng chứng chỉ kỹ thuật số để bảo vệ giao tiếp.
- Phát Hiện DDoS: Triển khai các công cụ phát hiện DDoS như SYNGuard và sử dụng bộ điều khiển phân tán để giảm thiểu thiệt hại của tấn công.
- **Tầng Dữ liệu (Data Plane)**
- **Các Hình Thức Tấn Công:**
 - Quy Tắc Luồng Gian Lận (Fraudulent Flow Rules): Kẻ tấn công có thể chen các quy tắc sai vào các thiết bị trong mạng để chi phối hoặc ngừng truyền dữ liệu.
 - Chuyển Hướng Lưu Lượng (Traffic Diversion): Lợi dụng các lỗ hổng mạng để chuyển hướng dữ liệu tới những điểm không mong muốn.
- **Giải pháp:**
 - Xác Thực Quy Tắc Luồng: Đảm bảo mọi quy tắc luồng đều được xác thực đúng đắn giữa các tầng ứng dụng và điều khiển để tránh rủi ro từ quy tắc giả.
 - Chuyển Hướng Lưu Lượng và Phát Hiện Bất Thường: Sử dụng các thuật toán động để điều chỉnh lưu lượng và áp dụng học máy để nhận diện các chuyển hướng dữ liệu bất thường.
- **Tầng Ứng dụng (Application Plane)**
- **Các Hình Thức Tấn Công:**
 - Lỗ Hổng trong Kiểm Soát Truy Cập và Trách Nhiệm Giải Trình: Các ứng dụng của bên thứ ba có thể thiếu các chính sách bảo mật chặt chẽ.
 - Lỗ Hổng trong Xác Thực: Thiếu cơ chế xác thực vững chắc khi có nhiều ứng dụng được triển khai.
 - Chèn Giả Mạo Quy Tắc Luồng/Giao Thông: Các ứng dụng bị tấn công có thể thêm quy tắc giả, đe dọa tính toàn vẹn hệ thống.

- **Giải pháp:**

- Phát Hiện Kênh Ẩn (CCD): Kiểm tra và ngăn ngừa các cuộc tấn công kênh ẩn, đồng thời xử lý các xung đột quy tắc.
- Kiểm Soát Truy Cập Theo Vai Trò: Cấp quyền cho các ứng dụng dựa trên vai trò thông qua các phân mở rộng của bộ điều khiển SDN.
- Kiểm Tra Quy Tắc Luồng và Chính Sách: Sử dụng các công cụ như FlowChecker và VeriFlow để phân tách các quy tắc độc hại và kiểm tra cấu hình trong thời gian thực.

2.1.8 Ưu nhược điểm của mạng SDN

- **Ưu Điểm:**

- Quản Lý Linh Hoạt và Tập Trung: SDN giúp tách biệt hoàn toàn giữa phần điều khiển và dữ liệu, cho phép các bộ điều khiển trung tâm quản lý toàn bộ mạng một cách hiệu quả hơn. Điều này giúp dễ dàng điều chỉnh cấu hình, tối ưu hóa lưu lượng và nâng cao hiệu suất tổng thể.
- Tối Ưu Hóa Tài Nguyên Mạng: Nhờ khả năng lập trình động, SDN cho phép tự động điều chỉnh tài nguyên mạng theo nhu cầu thực tế, cải thiện việc phân bổ băng thông và giảm tắc nghẽn.
- Triển Khai Dịch Vụ Nhanh Chóng: Các ứng dụng và dịch vụ mới có thể được triển khai linh hoạt thông qua lập trình, giúp các doanh nghiệp và nhà cung cấp dịch vụ nhanh chóng thích ứng với nhu cầu thay đổi.
- Hỗ Trợ Ảo Hóa và Điện Toán Đám Mây: SDN tích hợp tốt với các môi trường ảo hóa và nền tảng đám mây, cho phép quản lý mạng động và hiệu quả hơn trong các hệ thống quy mô lớn.

- **Nhược Điểm:**

- Rủi Ro Tập Trung Điều Khiển: Do bộ điều khiển SDN đóng vai trò cốt lõi trong toàn bộ hệ thống, nếu nó bị lỗi hoặc bị tấn công, toàn bộ mạng có thể bị gián đoạn hoặc gặp sự cố nghiêm trọng.

- Khả Năng Chịu Lỗi Hạn Chế: So với các mô hình mạng truyền thống, SDN phụ thuộc nhiều vào phần mềm, do đó dễ bị ảnh hưởng nếu gặp lỗi hệ thống, lỗi lập trình hoặc các cuộc tấn công bảo mật.
- Đòi Hỏi Hạ Tầng Phù Hợp: Việc triển khai SDN yêu cầu các thiết bị phần cứng hỗ trợ giao thức OpenFlow hoặc các tiêu chuẩn SDN khác, gây ra chi phí đầu tư ban đầu cao hơn cho các tổ chức.
- Phức Tạp trong Quản Trị và Vận Hành: SDN mang lại sự linh hoạt nhưng cũng đòi hỏi đội ngũ quản trị viên có chuyên môn cao để lập trình, triển khai và duy trì hệ thống một cách hiệu quả.
- Tương Thích Ngược với Mạng Truyền Thống: Việc chuyển đổi từ mạng truyền thống sang SDN có thể gặp nhiều thách thức do sự khác biệt về kiến trúc, yêu cầu thiết bị và cách thức hoạt động.

2.2 Tấn công từ chối dịch vụ phân tán (DDoS) trong SDN

2.2.1 Giới thiệu về DDOS

Tấn công DDoS (Distributed Denial-of-Service) là một hình thức tấn công mạng phổ biến, trong đó kẻ tấn công lợi dụng nhiều thiết bị bị kiểm soát để tạo ra lượng lớn lưu lượng truy cập đến mục tiêu. Mục đích chính là làm quá tải tài nguyên hệ thống, gây gián đoạn hoặc ngừng hoạt động các dịch vụ hợp pháp.

Hình thức tấn công này thường được thực hiện thông qua mạng botnet, bao gồm nhiều thiết bị bị nhiễm mã độc và bị điều khiển từ xa. Khi DDoS diễn ra, máy chủ hoặc hệ thống mục tiêu có thể gặp tình trạng quá tải băng thông, cạn kiệt tài nguyên xử lý, dẫn đến tốc độ phản hồi chậm hoặc mất khả năng phục vụ người dùng thực sự.

2.2.2 Các loại tấn công DDOS

Trong mô hình SDN, các cuộc tấn công DDoS có thể tác động lên nhiều tầng khác nhau với các mục đích riêng biệt và nguồn tài nguyên sẽ khai thác. Tấn công DDoS trong SDN thường được phân thành hai nhóm chính: Low-Rate DDoS và High-Rate DDoS, trong đó mỗi loại lại bao gồm nhiều hình thức khác nhau.

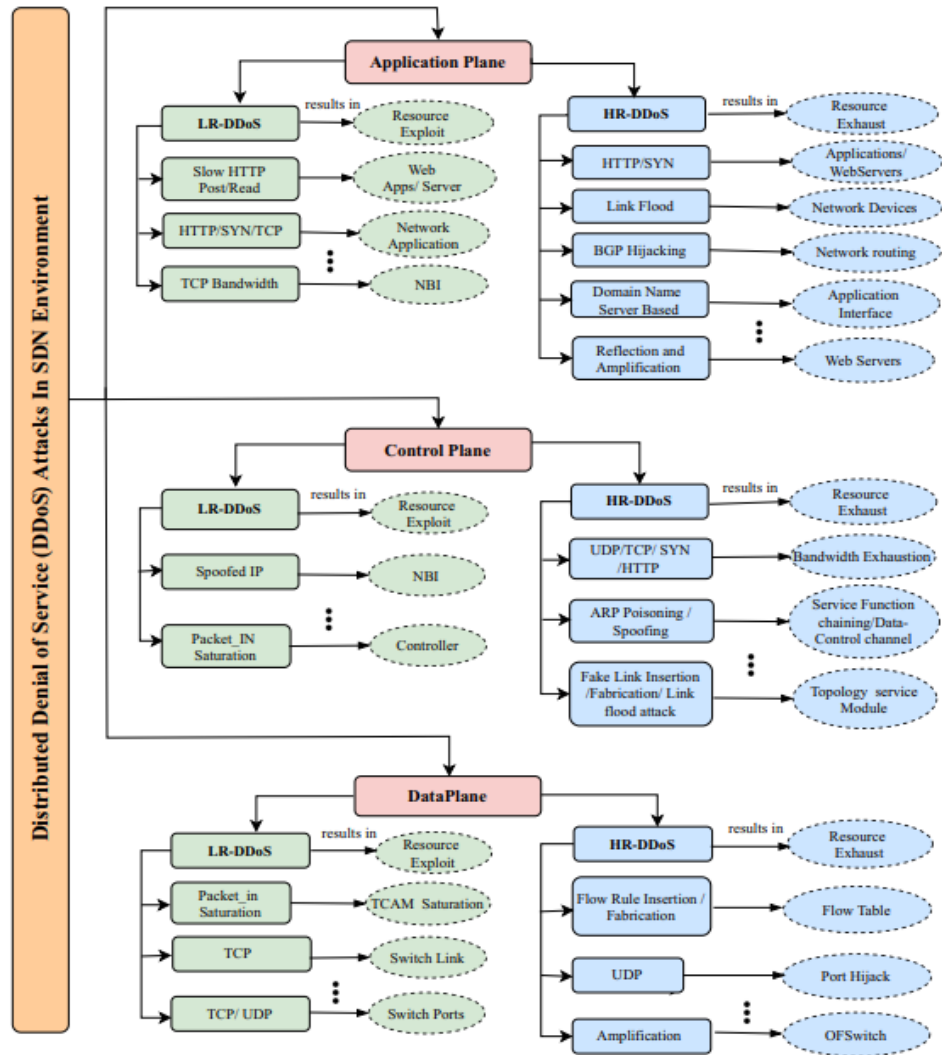


Fig. 5 DDoS Attack Taxonomy

Hình 2.3 Các loại tấn công DDOS

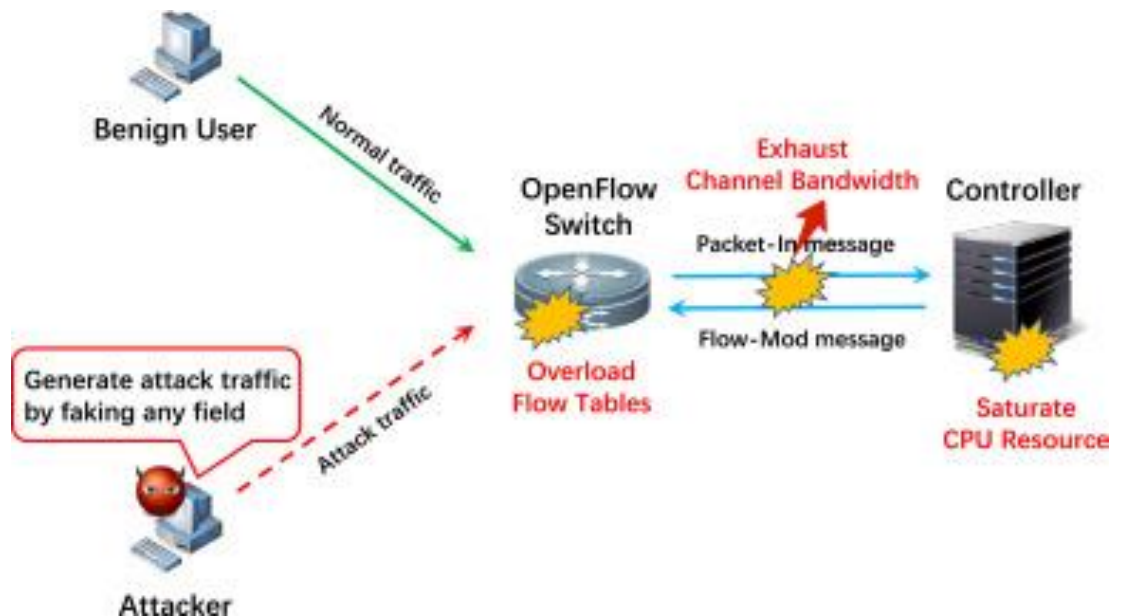
(Nguồn: Tariq, R., Raza, S. H., Shafiq, M., Alhussein, M., Alanazi, E., & Ghouzali, S. (2023). *A comprehensive survey on low-rate and high-rate DDoS defense approaches in SDN: Taxonomy, research challenges, and opportunities*. ResearchGate.)

2.2.2.1 Low rate DDOS

Đây là nhóm các kiểu tấn công với tốc độ thấp, thường tận dụng các kỹ thuật tinh vi để làm cạn kiệt tài nguyên mà không tạo ra lưu lượng quá lớn. Một số hình thức tấn công tiêu biểu của nhóm low rate ddos như sau:

- **Packet-In Saturation**

Tấn công Packet-In Saturation lợi dụng cơ chế xử lý gói tin trong SDN để làm suy giảm hiệu suất của bộ điều khiển. Kẻ tấn công phát sinh một lượng lớn gói tin với tiêu đề ngẫu nhiên và gửi đến thiết bị chuyển mạch OpenFlow. Do không có quy tắc xử lý phù hợp, switch sẽ chuyển tiếp những gói tin này lên bộ điều khiển thông qua thông điệp Packet-In. Khi lượng yêu cầu tăng quá mức, bộ điều khiển phải phản hồi liên tục bằng Flow-Mod, làm tiêu tốn tài nguyên xử lý và gây tình trạng quá tải. Ngoài ra, việc trao đổi dữ liệu giữa switch và bộ điều khiển cũng làm nghẽn kênh truyền thông, dẫn đến suy giảm hiệu suất hệ thống. Nếu cuộc tấn công kéo dài, bảng dòng chảy của switch có thể bị đầy, ảnh hưởng đến hoạt động của mạng SDN.



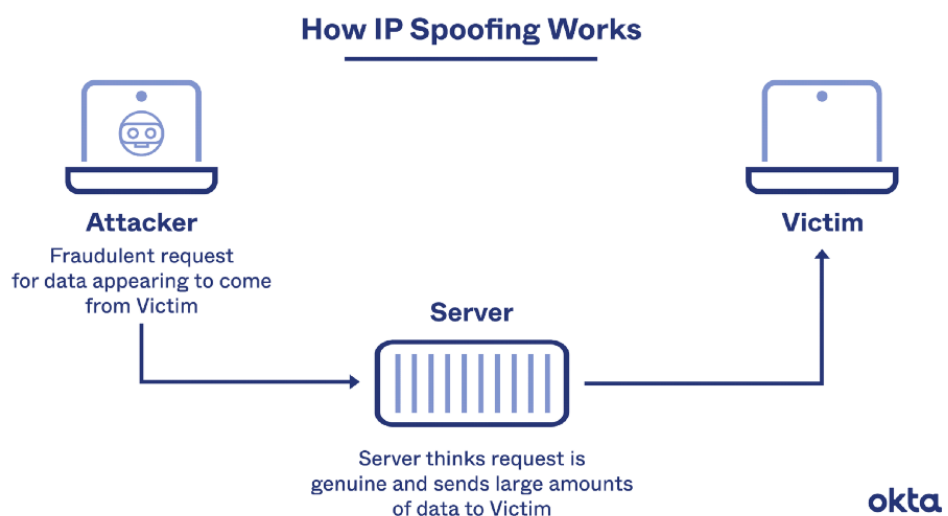
Hình 2.4 Packet-In Saturation

(Nguồn: Gharaibeh, A., Salahuddin, M. A., Hussain, S. A., Khreishah, A., & Guizani, M. (2020). *Smart detection of distributed denial of service attacks using support vector machine in software-defined networking*. Computer Networks, 168, 107035)

- **IP Spoofing Attack**

Tấn công IP Spoofing là phương thức giả mạo địa chỉ IP nguồn nhằm đánh lừa hệ thống mạng, khiến nó nhận diện sai nguồn gốc thực sự của gói tin. Trong mạng

SDN, kỹ thuật này có thể được khai thác để làm rối loạn quá trình định tuyến hoặc che giấu danh tính kẻ tấn công. Khi bộ điều khiển tiếp nhận các gói tin từ địa chỉ IP giả mạo, nó có thể đưa ra những quyết định xử lý không chính xác, dẫn đến việc chuyển tiếp dữ liệu sai lệch hoặc cập nhật bảng điều khiển với các quy tắc không mong muốn. Điều này không chỉ làm giảm hiệu suất hệ thống mà còn có thể dẫn đến các cuộc tấn công quy mô lớn hơn như tấn công từ chối dịch vụ (DoS) hoặc tấn công xen giữa (MitM), gây ảnh hưởng đến tính bảo mật và ổn định của mạng.



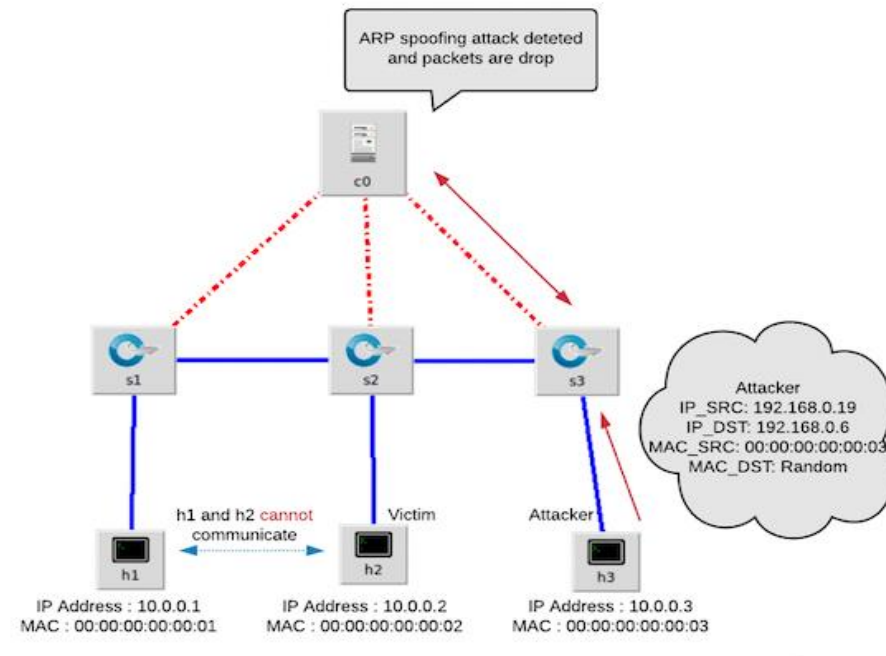
Hình 2.5 IP Spoofing Attack

(Nguồn: Okta. (n.d.). *IP spoofing*. Okta. Truy cập ngày 7 tháng 2 năm 2025)

- **ARP Poisoning / Spoofing**

Tấn công ARP Spoofing khai thác cơ chế phân giải địa chỉ ARP (Address Resolution Protocol) để giả mạo địa chỉ MAC, khiến các thiết bị trong mạng cập nhật sai ánh xạ giữa địa chỉ IP và MAC. Trong sơ đồ trên, kẻ tấn công (h3) gửi các gói tin giả mạo, làm cho switch và bộ điều khiển SDN hiểu sai về vị trí thực sự của thiết bị trong mạng. Kết quả là lưu lượng từ h1 đến h2 bị gián đoạn, khiến chúng không thể giao tiếp. Nếu cuộc tấn công không bị ngăn chặn, dữ liệu có thể bị chuyển hướng đến kẻ tấn công (h3), tạo điều kiện cho các cuộc tấn công Man-in-the-Middle (MITM) hoặc gây gián đoạn dịch vụ. Tuy nhiên, trong tình huống này, bộ điều khiển (c0) đã

phát hiện hoạt động giả mạo ARP và thực hiện chặn gói tin bất thường, giúp bảo vệ hệ thống khỏi cuộc tấn công.



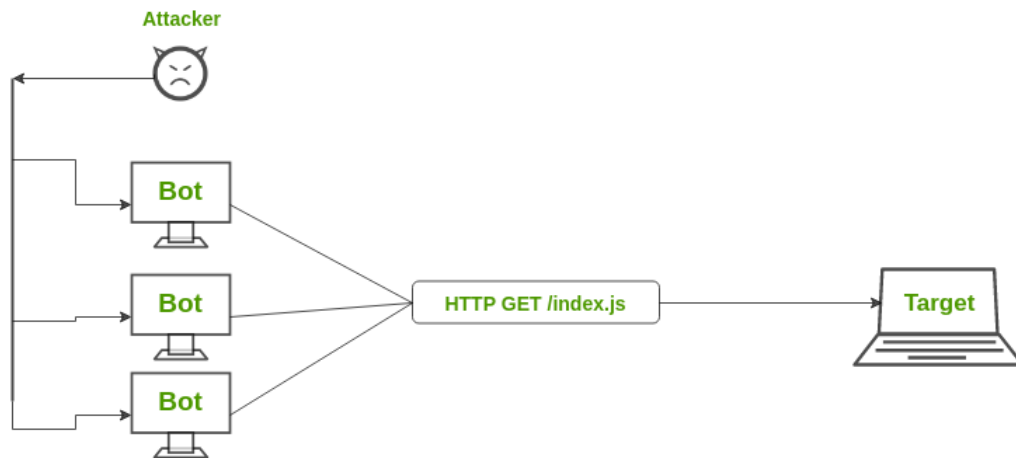
Hình 2.6 ARP Poisoning / Spoofing

(Nguồn: Rahman, M. A., Islam, S. R., Kabir, S., Alrajeh, N., & Rashed, A. N. (2019). Hybrid controller for securing SDN from switched DDoS and ARP poisoning attacks. *ResearchGate*)

2.2.2.2 High rate DDOS

Đây là nhóm các hình thức gây ra lượng lớn lưu lượng đột ngột, làm quá tải băng thông và ảnh hưởng nghiêm trọng đến hiệu suất hệ thống. Một số hình thức tấn công tiêu biểu của nhóm high rate ddos như sau:

- **HTTP Flood**



Hình 2.7 HTTP Flood

(Nguồn: GeeksforGeeks. (n.d.). What is distributed reflection denial of service? *GeeksforGeeks*.)

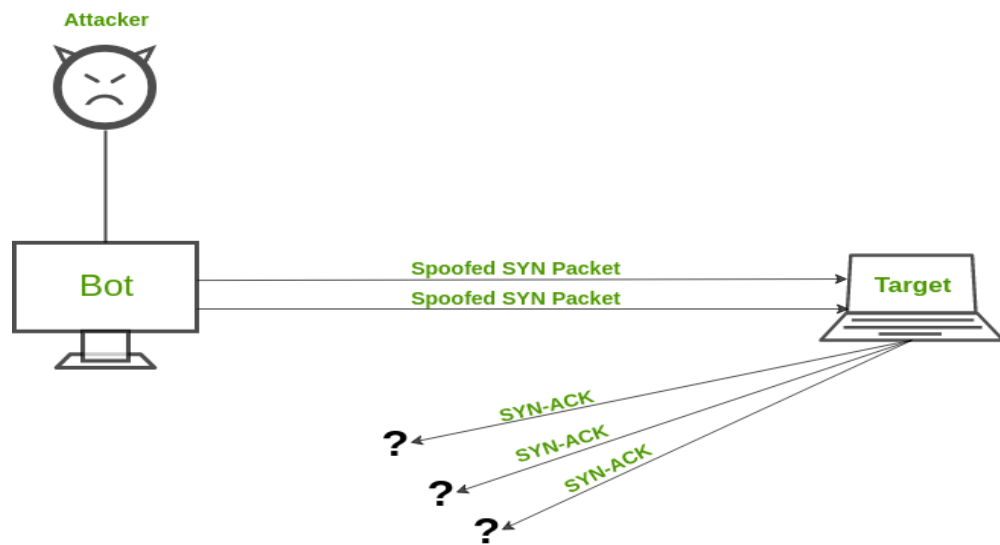
Một trong những hình thức tấn công HR-DDoS phổ biến là HTTP Flood. Trong kiểu tấn công này, kẻ tấn công gửi liên tục một số lượng lớn yêu cầu HTTP đến máy chủ hoặc bộ điều khiển SDN với tần suất rất cao. Khi server nhận quá nhiều yêu cầu cùng lúc, hệ thống không thể xử lý kịp, dẫn đến việc cạn kiệt tài nguyên và không thể phục vụ các yêu cầu hợp lệ.

Tấn công HTTP Flood đặc biệt nguy hiểm vì không cần khai thác các lỗ hổng trong giao thức, mà chỉ đơn giản là gửi các yêu cầu HTTP hợp lệ, làm tắc nghẽn băng thông và tài nguyên của máy chủ. Hệ thống bị chiếm dụng tài nguyên mà không thể phục vụ được các kết nối hợp pháp.

- **UDP/TCP SYN Flood**

Tấn công kết hợp TCP SYN Flood và UDP Flood là một dạng tấn công từ chối dịch vụ (DDoS) phức tạp, khai thác những điểm yếu đặc thù của cả hai giao thức. Với TCP SYN Flood, kẻ tấn công gửi đi lượng lớn gói SYN có địa chỉ IP giả, buộc máy chủ phản hồi bằng gói SYN-ACK và chờ gói ACK để hoàn tất quy trình bắt tay. Tuy nhiên, do địa chỉ IP giả mạo, gói ACK không bao giờ đến, khiến máy chủ phải duy trì các kết nối "nửa mở", làm cạn kiệt tài nguyên. Trong khi đó, UDP Flood gửi đi số lượng lớn gói tin UDP tới các cổng ngẫu nhiên. Máy chủ phải kiểm tra các cổng này

và nếu không có dịch vụ lắng nghe, nó sẽ gửi lại gói ICMP báo lỗi "Destination Unreachable", làm tăng áp lực xử lý. Sự kết hợp giữa hai hình thức tấn công này tạo ra sức ép lớn lên hệ thống, nhanh chóng làm gián đoạn hoạt động của máy chủ. Để giảm thiểu tác động, cần áp dụng các biện pháp như giới hạn số lượng kết nối, sử dụng tường lửa để lọc lưu lượng, bật tính năng SYN Cookies và triển khai công cụ giám sát để phát hiện các hoạt động bất thường.

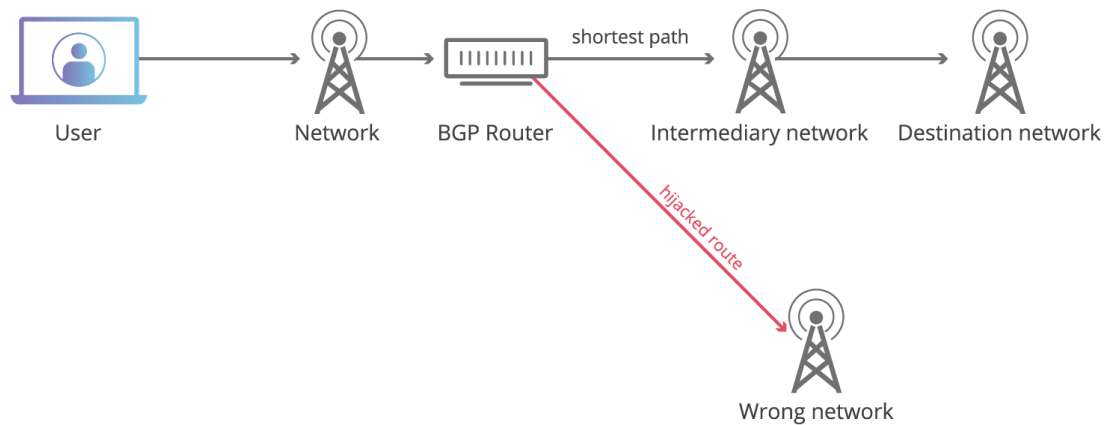


Hình 2.8 UDP/TCP SYN Flood

(Nguồn: GeeksforGeeks. (n.d.). What is distributed reflection denial of service? *GeeksforGeeks*.)

- **BGP Hijacking**

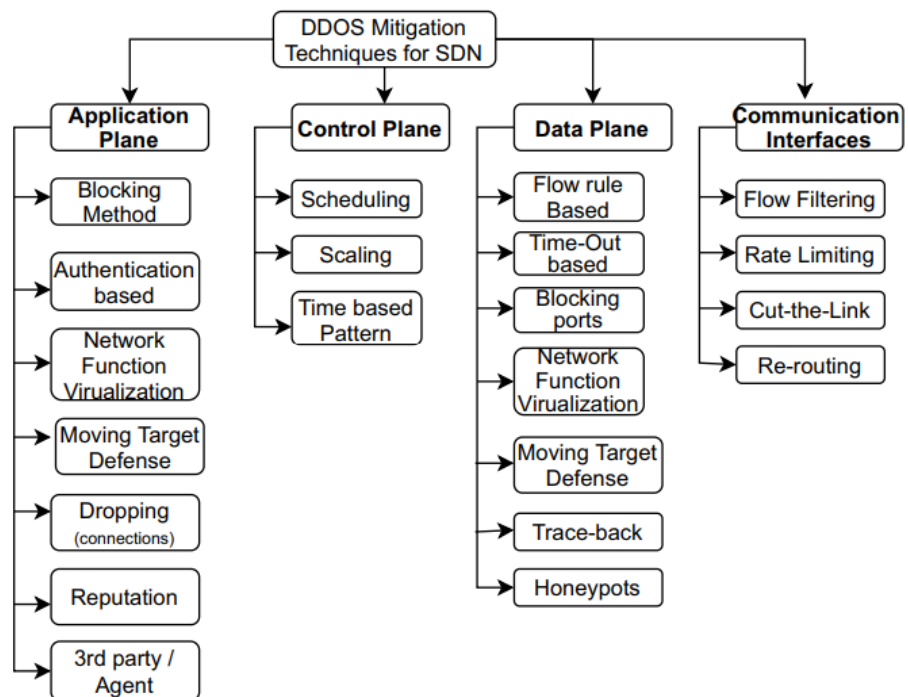
BGP Hijacking là một tấn công nguy hiểm, trong đó kẻ tấn công giả mạo thông tin định tuyến BGP để chuyển hướng lưu lượng mạng qua các tuyến không hợp lệ, như minh họa trong hình. Thay vì lưu lượng đi theo tuyến hợp pháp đến đích, nó bị chuyển qua một mạng độc hại do kẻ tấn công kiểm soát. Trong môi trường SDN, nơi định tuyến được quản lý tập trung, tấn công này đặc biệt nghiêm trọng. Kẻ tấn công có thể lợi dụng cấu trúc định tuyến động để gây rò rỉ dữ liệu, đánh cắp thông tin, hoặc thực hiện tấn công Man-in-the-Middle quy mô lớn. Các giải pháp như xác thực định tuyến, RPKI và giám sát lưu lượng bất thường là cần thiết để bảo vệ hệ thống SDN khỏi loại tấn công này.



Hình 2.9 BGP Hijacking

(Nguồn: Kurbatov, D. (2023, [ngày/tháng nếu có]). Cybersecurity, BGP, Network Security. *LinkedIn*)

2.2.3 Các phương pháp giảm thiểu DDOS trong mạng SDN



Hình 2.10 Các phương pháp giảm thiểu DDOS trong mạng SDN

(Nguồn:Karnani, S., & Shakya, H. K. (2022). Hình ảnh từ *Mitigation strategies for distributed denial of service (DDoS) in SDN: A survey and taxonomy. Information Security Journal: A Global Perspective*, 32(4), 1-25.)

Có rất nhiều phương pháp được dùng để phòng chống DDOS, mỗi tầng trong mạng SDN sẽ có các phương pháp khác nhau. Sau đây là một số phương pháp tại từng tầng:

- **Lớp Ứng Dụng (Application Plane)**

- Chặn kết nối độc hại: Các địa chỉ IP có lịch sử thực hiện hành vi bất thường hoặc gây ra lưu lượng đáng ngờ sẽ bị ngăn chặn, giúp giảm thiểu rủi ro từ các nguồn tấn công đã biết.
- Tích hợp xác thực nhiều lớp: Việc yêu cầu người dùng trải qua nhiều bước xác thực sẽ giúp nâng cao độ an toàn, đặc biệt đối với các hệ thống quan trọng.
- Ảo hóa chức năng mạng (NFV - Network Function Virtualization): Việc triển khai linh hoạt các tính năng bảo mật như tường lửa và cân bằng tải giúp cải thiện khả năng chống chịu trước các cuộc tấn công quy mô lớn.
- Chiến lược phòng thủ mục tiêu di động (MTD - Moving Target Defense): Bằng cách thay đổi liên tục cấu hình mạng như địa chỉ IP và cổng kết nối, hệ thống sẽ khiến kẻ tấn công khó xác định mục tiêu cụ thể.
- Hệ thống danh tiếng (Reputation System): Theo dõi và đánh giá danh tiếng của các nguồn lưu lượng để xác định những nguồn tiềm ẩn nguy cơ.
- Hợp tác với bên thứ ba (Third-Party Mitigation Services): Sử dụng dịch vụ bảo vệ từ các nhà cung cấp chuyên nghiệp giúp tăng cường khả năng phát hiện và xử lý các cuộc tấn công phức tạp.

- **Lớp Điều Khiển (Control Plane)**

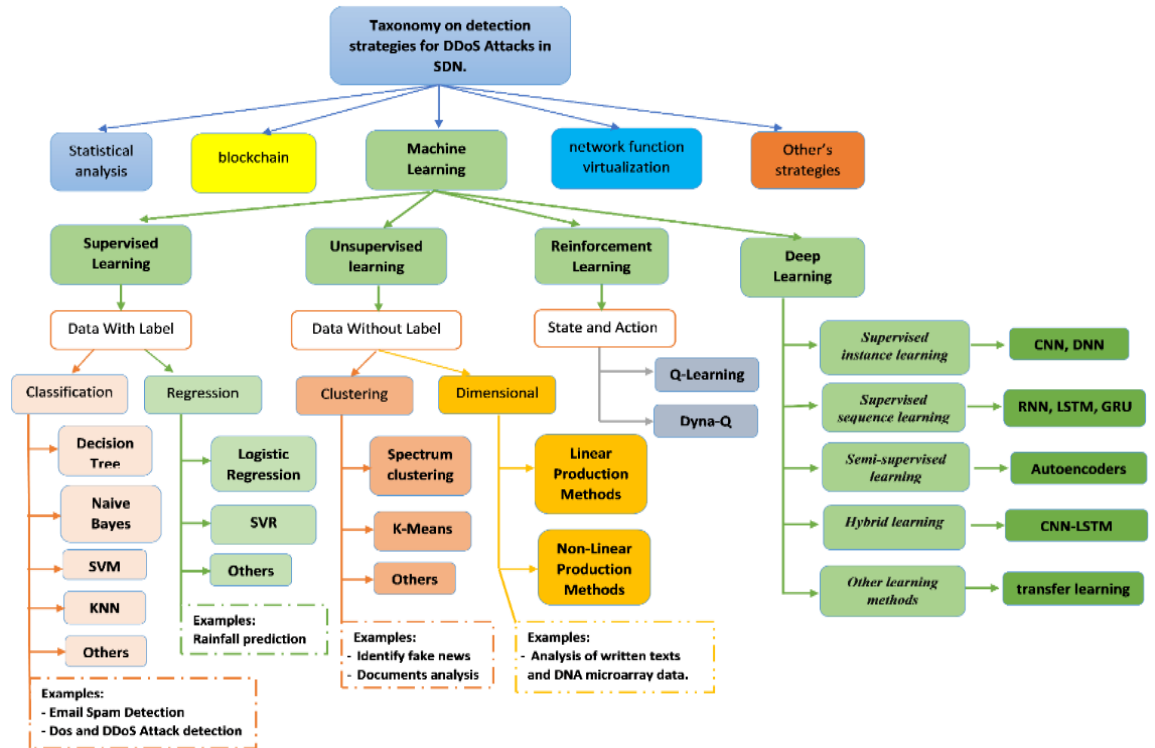
- Quản lý bộ điều khiển hiệu quả: Áp dụng các thuật toán lập lịch giúp tối ưu hóa việc xử lý yêu cầu, tránh tình trạng quá tải bộ điều khiển.
- Mở rộng tài nguyên động: Khi lưu lượng tấn công tăng cao, hệ thống có thể mở rộng băng thông hoặc bổ sung máy chủ để duy trì hiệu suất.

- Phân tích mẫu dựa trên thời gian: Xác định các hành vi bất thường bằng cách theo dõi sự thay đổi của lưu lượng mạng theo từng giai đoạn.
- Ứng dụng học máy (Machine Learning): Các thuật toán như SVM, KNN, K-Means Clustering giúp phát hiện các đặc điểm bất thường trong lưu lượng mạng, từ đó phân loại và ngăn chặn tấn công hiệu quả hơn.
- **Lớp Dữ Liệu (Data Plane)**
 - Áp dụng quy tắc lọc luồng (Flow-Rule Based Mitigation): Thiết lập quy tắc nhằm xác định, chặn hoặc chuyển hướng lưu lượng có dấu hiệu tấn công.
 - Cơ chế giới hạn thời gian (Timeout Mechanisms): Đặt thời gian chờ cho các kết nối không hợp lệ để tránh tình trạng tiêu tốn tài nguyên không cần thiết.
 - Chặn truy cập vào các cổng dễ bị tấn công: Hạn chế hoặc đóng các cổng không cần thiết để giảm thiểu nguy cơ khai thác lỗ hổng.
 - Hệ thống honeypot: Dùng các hệ thống giả lập để thu hút và nghiên cứu hành vi của kẻ tấn công, từ đó nâng cao khả năng phòng vệ.
- **Giao Diện Truyền Thông (Communication Interfaces)**
 - Lọc luồng dữ liệu và gói tin: Xác định và loại bỏ lưu lượng không mong muốn dựa trên các quy tắc bảo mật đã thiết lập.
 - Giới hạn tốc độ (Rate Limiting): Điều chỉnh mức độ lưu lượng từ một nguồn cụ thể nhằm ngăn chặn tình trạng tắc nghẽn do tấn công.
 - Chuyển hướng lưu lượng (Re-routing/Directing): Điều hướng dữ liệu ra khỏi khu vực bị tấn công để giảm áp lực lên hệ thống mục tiêu.

2.3 Phương pháp phát hiện DDOS sử dụng Machine Learning

Trong lĩnh vực phát hiện tấn công DDoS, học máy là một trong các phương pháp hiệu quả để phân tích và xác định các mẫu lưu lượng bất thường. Có nhiều cách tiếp cận khác nhau, bao gồm học có giám sát (Supervised Learning), học không giám

sát (Unsupervised Learning), học tăng cường (Reinforcement Learning) và học sâu (Deep Learning), mỗi phương pháp có ưu điểm riêng trong việc xử lý dữ liệu mạng.



Hình 2.11 Các phương pháp Machine Learning phát hiện DDOS

(Nguồn: Ali, T. E., Chong, Y. W., & Manickam, S. (2023). Machine learning techniques to detect a DDoS attack in SDN: A systematic review. *Applied Sciences*, 13(5), 3183.)

Trong bài báo cáo này, lựa chọn học có giám sát (Supervised Learning) và áp dụng các thuật toán Logistic Regression, Decision Tree, K-Nearest Neighbors (KNN) và Support Vector Machine (SVM). Những thuật toán này có khả năng phân loại lưu lượng mạng thành hai nhóm: hợp lệ và độc hại, từ đó hỗ trợ phát hiện và ngăn chặn tấn công DDoS một cách hiệu quả.

2.3.1 Logistic Regression

2.3.1.1 Giới thiệu Logistic Regression

Logistic Regression là một mô hình phân loại nhị phân phổ biến trong học máy, được sử dụng để phân loại dữ liệu thành hai lớp, ví dụ như "tấn công" và "hợp lệ" (không tấn công). Mô hình này sử dụng một hàm hồi quy tuyến tính kết hợp với hàm sigmoid để chuyển đầu ra thành xác suất, sau đó dự đoán xem mẫu có thuộc lớp tấn công hay không. Sự phân loại được xác định dựa trên một ngưỡng xác suất.

2.3.1.2 Cách hoạt động của Logistic Regression

- Xây dựng mô hình:

Mô hình nhận vào một vector các đặc trưng $\mathbf{x} = [x_1, x_2, \dots, x_n]$ và tính toán tổng trọng số của chúng cùng với độ lệch b để tạo ra một giá trị z . Công thức tính toán này như sau:

$$z = \mathbf{w}^T \mathbf{x} + b$$

$\mathbf{w}^T \mathbf{x}$: Tính toán tích vô hướng giữa vector trọng số \mathbf{w} và vector đặc trưng \mathbf{x} .

b : Là độ lệch, giúp dịch chuyển siêu phẳng phân chia giữa các lớp.

- Hàm sigmoid:

Để chuyển giá trị z thành xác suất, ta sử dụng hàm sigmoid:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Xác suất $\sigma(z)$ sẽ cho biết khả năng mẫu thuộc vào lớp tấn công (1) hoặc không tấn công (0).

- Phân loại:

Khi xác suất $\sigma(z)$ lớn hơn hoặc bằng 0.5, mẫu được phân loại là "tấn công". Ngược lại, nếu xác suất nhỏ hơn 0.5, mẫu được phân loại là "hợp lệ".

2.3.2 Decision Tree

2.3.2.1 Giới thiệu Decision Tree

Decision Tree (Cây Quyết Định) là một thuật toán phân loại mạnh mẽ sử dụng cấu trúc cây để phân chia dữ liệu thành các nhóm nhỏ hơn. Mỗi nút trong cây đại diện cho một đặc trưng, và mỗi nhánh thể hiện các giá trị của đặc trưng đó. Mục tiêu là tạo ra một cây mà tại mỗi nút, các dữ liệu trong mỗi nhóm con đều đồng nhất về lớp.

2.3.2.2 Cách hoạt động của Decision Tree

- Quyết định phân chia:

Thuật toán quyết định việc phân chia dữ liệu dựa trên tiêu chí như Entropy hoặc Gini Index. Những tiêu chí này đo độ thuần nhất của các lớp trong các nhóm con và tìm kiếm phân chia tối ưu.

Entropy đo lường mức độ hỗn loạn của một tập dữ liệu:

$$H(D) = - \sum_{i=1}^k p_i \log_2(p_i)$$

Trong đó:

p_i : là xác suất mẫu thuộc lớp i .

k : là số lớp.

Information Gain đo sự thay đổi của entropy khi chia dữ liệu theo một đặc trưng A :

$$IG(D, A) = H(D) - \sum_{v \in A} \frac{|D_v|}{|D|} H(D_v)$$

- Phân chia dữ liệu:

Thuật toán chọn đặc trưng sao cho Information Gain là lớn nhất, tức là sự giảm entropy là lớn nhất. Tiếp theo, dữ liệu được chia thành các nhóm con dựa trên các giá trị của đặc trưng này.

- Tạo cây:

Quá trình này tiếp tục cho đến khi mỗi nhóm con trở nên thuần nhất hoặc đạt được một điều kiện dừng nhất định, ví dụ như độ sâu của cây.

2.3.3 KNN (*K-nearest neighbors*)

2.3.3.1 Giới thiệu KNN

K-Nearest Neighbors (KNN) là một thuật toán phân loại không tham số, dựa vào nguyên lý rằng các mẫu gần nhau trong không gian đặc trưng thường có cùng

lớp. Thuật toán không cần huấn luyện mà chỉ tính toán khoảng cách giữa mẫu cần phân loại và các mẫu trong tập huấn luyện để xác định lớp của mẫu mới.

2.3.3.2 Cách hoạt động của KNN

- Tính toán khoảng cách:

Để tìm kkk mẫu gần nhất, thuật toán tính toán khoảng cách giữa mẫu cần phân loại và tất cả các mẫu trong tập huấn luyện. Khoảng cách này thường được tính bằng Euclidean Distance:

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{k=1}^n (x_{ik} - x_{jk})^2}$$

Trong đó:

\mathbf{x}_i và \mathbf{x}_j : là các mẫu cần so sánh.

n : là số đặc trưng của mỗi mẫu.

- Chọn lớp:

Sau khi tính toán khoảng cách, kkk mẫu gần nhất được chọn. Mẫu cần phân loại sẽ được gán vào lớp có số lần xuất hiện nhiều nhất trong nhóm kkk mẫu gần nhất.

- Phân loại:

Mẫu được phân loại vào lớp mà có số mẫu nhiều nhất trong số kkk mẫu gần nhất. Nếu $k = 3$, ví dụ, ta sẽ chọn lớp xuất hiện nhiều nhất trong 3 mẫu gần nhất.

2.3.4 SVM (*Support Vector Machine*)

2.3.4.1 Giới thiệu SVM

Support Vector Machine (SVM) là một thuật toán phân loại mạnh mẽ, tìm kiếm một siêu phẳng (hyperplane) phân chia các lớp sao cho khoảng cách giữa siêu phẳng và các điểm gần nhất (được gọi là Support Vectors) là lớn nhất. Mục tiêu của SVM là tối đa hóa margin (khoảng cách) giữa các lớp.

2.3.4.2 Cách hoạt động của SVM

- Tìm siêu phẳng tối ưu:

SVM tìm một siêu phẳng có dạng $\mathbf{w}^T \mathbf{x} + b = 0$, sao cho khoảng cách từ siêu phẳng đến các điểm dữ liệu của hai lớp là lớn nhất.

- Tối đa hóa margin:

Khoảng cách này được gọi là margin, và SVM tối đa hóa nó bằng cách tối thiểu hóa giá trị của trọng số $\|\mathbf{w}\|$, vì khoảng cách giữa siêu phẳng và các điểm gần nhất được tính bằng:

$$\text{Margin} = \frac{1}{\|\mathbf{w}\|}$$

- Phân loại:

Sau khi tìm được siêu phẳng tối ưu, mẫu mới sẽ được phân loại dựa trên dấu của hàm $\mathbf{w}^T \mathbf{x} + b$. Nếu kết quả dương, mẫu thuộc lớp 1, ngược lại thuộc lớp 0.

- Xử lý với dữ liệu không phân tách được:

Khi dữ liệu không thể phân chia hoàn hảo, SVM sử dụng một hàm mất mát có tên hinge loss để xử lý các điểm dữ liệu bị lỗi.

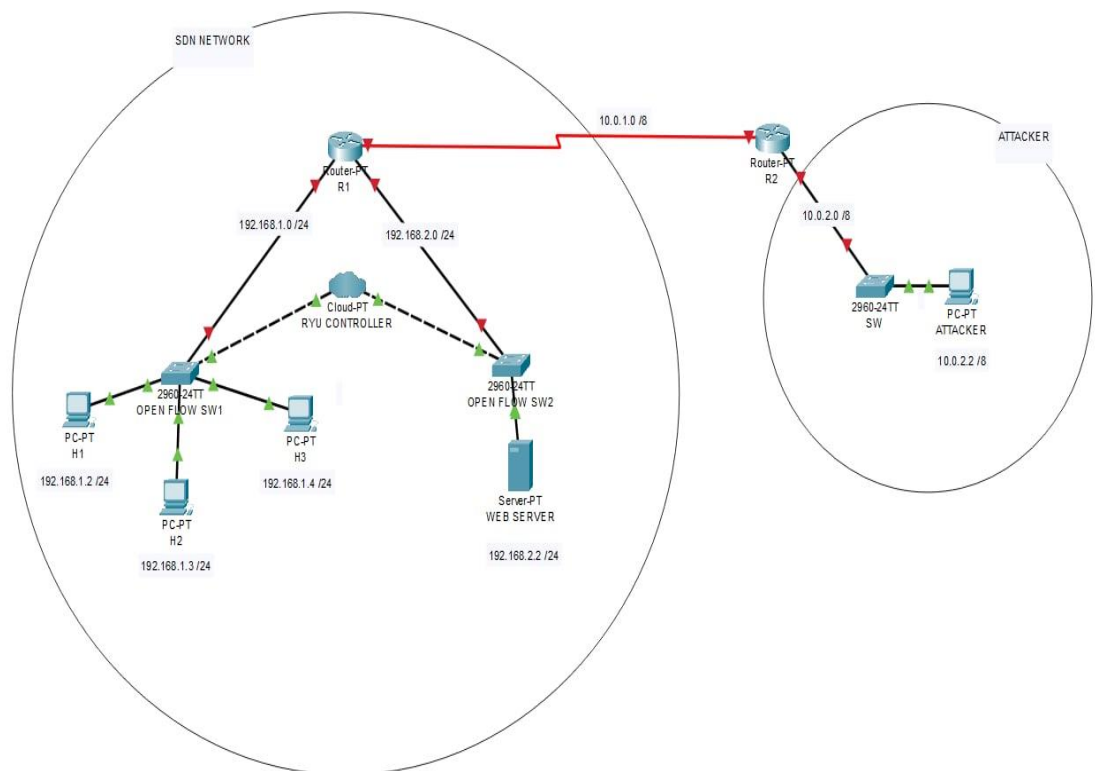
CHƯƠNG 3. MÔ HÌNH ĐỀ XUẤT

3.1 Giới thiệu và đề xuất mô hình

Mạng SDN (Software-Defined Networking) mang lại sự linh hoạt cao trong việc quản lý và kiểm soát lưu lượng mạng, nhưng cũng đối mặt với nguy cơ các cuộc tấn công DDoS có thể làm gián đoạn hoạt động của hệ thống. Để giảm thiểu rủi ro này, việc ứng dụng học máy vào phát hiện DDoS đã chứng minh được hiệu quả, nhờ vào khả năng phân tích và nhận diện các mẫu bất thường trong lưu lượng mạng.

Mô hình học máy được đề xuất trong bài báo cáo này nhằm phát hiện tấn công DDoS trong môi trường mạng SDN, sử dụng các thuật toán như Logistic Regression, Decision Tree, KNN và SVM. Những thuật toán này giúp nhận diện các hành vi đáng ngờ và nâng cao tính bảo mật của mạng SDN, từ đó tối ưu hóa việc phát hiện và ngăn chặn các tấn công mạng.

3.2 Mô phỏng mô hình mạng



Hình 3.1 Mô hình mạng tổng quát

Mô hình mạng này được chia thành hai khu vực riêng biệt: hệ thống SDN và mạng Attacker, với một liên kết trung gian giúp kết nối chúng lại. Hệ thống SDN hoạt động dựa trên nguyên tắc điều khiển tập trung, cho phép quản lý luồng dữ liệu và kiểm soát các thiết bị trong mạng một cách linh hoạt. Ngược lại, mạng Attacker được thiết kế để giả lập các kịch bản tấn công từ bên ngoài nhằm kiểm tra khả năng bảo vệ của hệ thống. Hai phần này được kết nối với nhau thông qua một tuyến định tuyến, tạo ra môi trường thử nghiệm phục vụ nghiên cứu và đánh giá bảo mật.

3.2.1 Mô hình mạng SDN

Mạng SDN trong hệ thống được xây dựng với kiến trúc điều khiển tập trung, trong đó bộ điều khiển SDN quản lý toàn bộ lưu lượng mạng. Mô hình bao gồm các thành phần chính sau:

- **Router R1 (SDN Router):** Đây là bộ định tuyến trung tâm của mạng SDN, đóng vai trò cổng chính kết nối giữa các subnet nội bộ và bên ngoài. Router này đảm bảo truyền tải dữ liệu giữa các thiết bị và quản lý việc giao tiếp với mạng Attacker.
- **Bộ điều khiển SDN (RYU Controller):** Điều khiển các OpenFlow Switch và quản lý chính sách mạng, giúp linh hoạt trong việc điều phối dữ liệu.
- **Hai Switch OpenFlow (SW1, SW2):** Là các switch SDN, chịu trách nhiệm chuyển tiếp gói tin dựa trên quy tắc mà bộ điều khiển đặt ra. Chúng giúp phân phối lưu lượng giữa các thiết bị trong mạng.
- **Các máy tính (PC1, PC2, PC3) và máy chủ (Web Server):** Các thiết bị đầu cuối này được kết nối vào mạng thông qua switch SDN. Web Server có nhiệm vụ cung cấp dịch vụ cho các máy tính và được dùng để mô phỏng tấn công từ mạng bên ngoài vào server của hệ thống.
- **Địa chỉ IP trong mạng:** phân chia mạng SDN thành hai subnet:
 - **192.168.1.0/24:** Kết nối các máy tính PC1, PC2, PC3 với SW1.
 - **192.168.2.0/24:** Kết nối Web Server với SW2.

- **Kết nối và quản lý mạng:** Router R1 giúp kết nối các subnet với nhau và đảm bảo giao tiếp giữa mạng SDN với các hệ thống bên ngoài.

3.2.2 Mạng của Attacker

Bên cạnh mạng SDN sẽ có một mạng cho Attacker để mô phỏng các kịch bản tấn công và đánh giá kết quả DDOS. Mạng này bao gồm:

- **Router R2 (Attacker Router):** Bộ định tuyến này giúp kết nối mạng Attacker với mạng SDN, đồng thời chịu trách nhiệm định tuyến lưu lượng giữa hai hệ thống.
- **Switch Attacker (SW):** Là thiết bị trung gian giúp kết nối các thiết bị trong mạng Attacker.
- **Máy Attacker (PC Attacker):** Đóng vai trò thực hiện các cuộc thử nghiệm xâm nhập vào mạng SDN.
- **Địa chỉ IP trong mạng:** Thiết lập mạng Attacker với dải địa chỉ **10.0.2.0/8**, tách biệt hoàn toàn với mạng SDN.
- **Kết nối giữa mạng Attacker và mạng SDN:**
 - Router R2 kết nối với Router R1 qua tuyến **10.0.1.0/8**.
 - PC Attacker kết nối với mạng thông qua switch SW.

3.3 Mô hình phát hiện DDOS

Hệ thống phát hiện tấn công DDoS trong mạng SDN được thiết kế nhằm xác định và phân loại lưu lượng mạng, từ đó đưa ra biện pháp ngăn chặn kịp thời. Mô hình bao gồm ba giai đoạn chính: sinh dữ liệu và thu thập dữ liệu, huấn luyện mô hình học máy và triển khai phát hiện tấn công dựa trên mô hình đã được huấn luyện.

3.3.1 Cấu trúc tổng thể

Quá trình phát hiện DDoS diễn ra theo các bước chính sau:

- **Thu thập dữ liệu:** Ghi nhận lưu lượng mạng từ hệ thống SDN, bao gồm cả truy cập hợp lệ và lưu lượng DDoS để tạo tập dữ liệu phục vụ huấn luyện.

- **Huấn luyện mô hình:** Xử lý dữ liệu thu thập được, trích xuất các đặc trưng quan trọng và áp dụng thuật toán học máy để xây dựng mô hình phát hiện tấn công.
- **Dự đoán và phản hồi:** Sử dụng mô hình đã huấn luyện để giám sát lưu lượng mạng, từ đó phát hiện và phản ứng khi có dấu hiệu tấn công.

3.3.2 Thành phần chính

Mô hình phát hiện DDoS được triển khai với các thành phần quan trọng sau:

- **Bộ mô phỏng mạng SDN:**
 - Mô phỏng mạng bằng Mininet, giúp tạo môi trường thực nghiệm gần với thực tế.
 - Sử dụng Ryu controller để quản lý lưu lượng mạng và điều phối các luồng dữ liệu.
- **Bộ thu thập dữ liệu:**
 - Ghi nhận thông tin từ mạng, bao gồm cả lưu lượng hợp lệ và lưu lượng từ các cuộc tấn công DDoS.
 - Trích xuất các đặc trưng quan trọng từ gói tin để làm đầu vào cho mô hình học máy.
- **Bộ sinh dữ liệu tấn công và hợp lệ:**
 - Tạo ra các mẫu lưu lượng tấn công DDoS với các phương thức phổ biến như SYN Flood, UDP Flood hoặc HTTP Flood.
 - Sinh lưu lượng mạng hợp lệ để đảm bảo mô hình học máy có dữ liệu cân bằng giữa hai nhóm.
- **Bộ huấn luyện mô hình học máy:**
 - Xử lý dữ liệu thu thập được, làm sạch và chuẩn hóa để phù hợp với thuật toán học máy.
 - Huấn luyện các mô hình như Logistic Regression, Decision Tree, KNN và SVM nhằm tối ưu khả năng phân loại lưu lượng mạng.
 - Lưu mô hình đã huấn luyện để sử dụng trong quá trình giám sát.

CHƯƠNG 4. THỰC NGHIỆM

4.1 Cài đặt môi trường thực nghiệm

4.1.1 Khởi tạo mô hình mạng SDN

4.1.1.1 Khởi tạo Base Ryu controller

Bộ điều khiển Ryu được xây dựng để quản lý các switch hỗ trợ OpenFlow 1.3, cho phép kiểm soát luồng dữ liệu trong mạng SDN. Mục đích của controller này sẽ tạo các chức năng cơ bản cần có của một controller và sau đó sẽ cho các controller mới kế thừa các chức năng của controller này mà không phải tạo lại.

Controller này sẽ thực hiện nhiệm vụ khi một switch kết nối, controller sẽ cài đặt một quy tắc mặc định (table-miss flow entry) để chuyển các gói tin không khớp về controller xử lý. Khi nhận được gói tin, nó kiểm tra thông tin địa chỉ MAC, lưu lại vị trí của các thiết bị, và quyết định chuyển tiếp dựa trên bảng học địa chỉ. Nếu gói tin thuộc một giao thức mạng như ICMP, TCP hoặc UDP, controller sẽ tạo quy tắc cụ thể để xử lý, giúp giảm tải các sự kiện PacketIn.

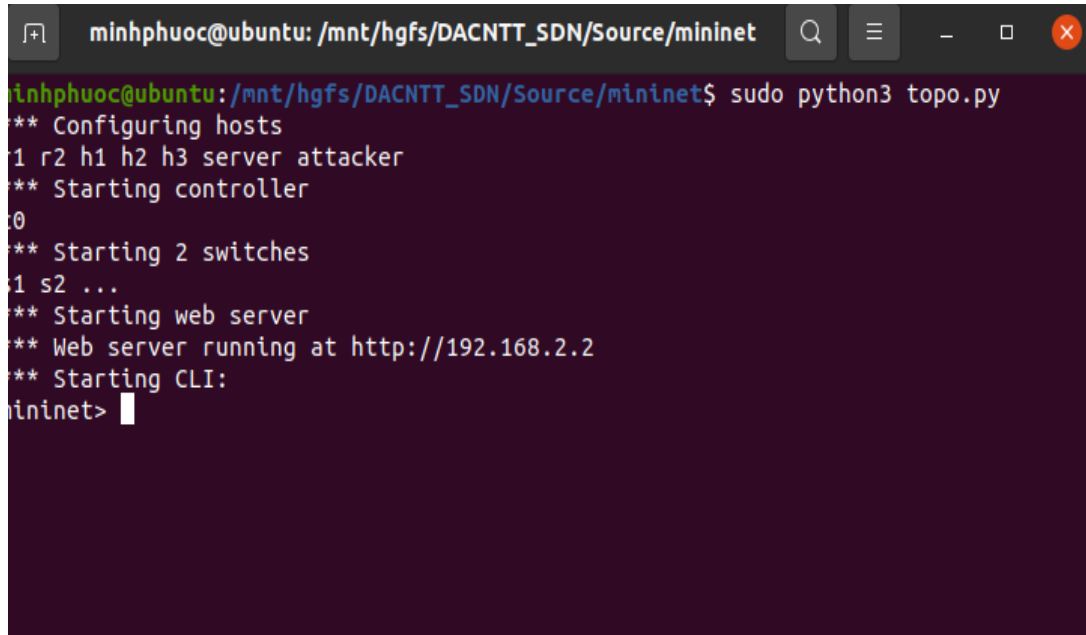
4.1.1.2 Khởi tạo mô hình mạng

Mạng SDN mô phỏng bằng Mininet, bao gồm một bộ điều khiển từ xa, các bộ định tuyến, switch OpenFlow, các máy chủ, và một tác nhân tấn công. Bộ điều khiển từ xa hoạt động trên địa chỉ 127.0.0.1 và cổng 6633, chịu trách nhiệm quản lý luồng dữ liệu trong hệ thống.

Mạng được chia thành ba phần chính: mạng nội bộ SDN, mạng máy chủ, và mạng bên ngoài có tác nhân tấn công. Bộ định tuyến SDN (r1) kết nối với các switch (s1 và s2), đảm bảo việc giao tiếp giữa các thiết bị trong mạng. Trong khi đó, bộ định tuyến ISP (r2) duy trì kết nối với tác nhân tấn công, mô phỏng một mạng bên ngoài.

Sau khi các thiết bị được thêm vào, chương trình tiến hành cấu hình địa chỉ IP cho từng cổng mạng của bộ định tuyến và thiết lập tính năng chuyển tiếp gói tin (IP forwarding). Các tuyến đường (routes) cũng được cấu hình thủ công để đảm bảo rằng tất cả các thiết bị có thể liên lạc với nhau đúng cách.

Bên cạnh đó, máy chủ web trên server được khởi chạy bằng cách sử dụng `python3 -m http.server` để mô phỏng một dịch vụ có thể bị tấn công. Cuối cùng, Mininet CLI được khởi chạy để người dùng có thể kiểm tra và tương tác với hệ thống trước khi mạng bị tắt.



```

minhphuoc@ubuntu: /mnt/hgfs/DACNTT_SDN/Source/mininet$ sudo python3 topo.py
** Configuring hosts
1 r2 h1 h2 h3 server attacker
** Starting controller
0
** Starting 2 switches
1 s2 ...
** Starting web server
** Web server running at http://192.168.2.2
** Starting CLI:
mininet>

```

Hình 4.1 Khởi tạo mininet

4.1.2 Thu thập dữ liệu thực nghiệm

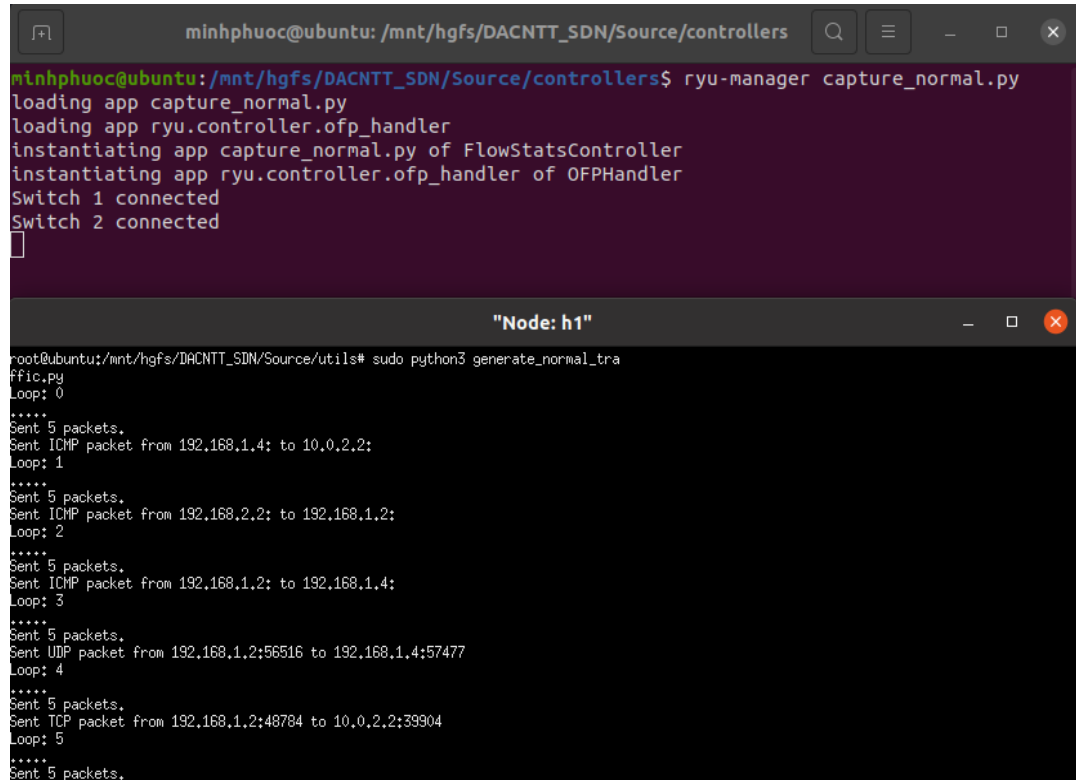
4.1.2.1 Mô tả dữ liệu thực nghiệm

Trong bài báo cáo này, dữ liệu thực nghiệm được thu thập thông qua một môi trường mô phỏng mạng được thiết lập bằng Mininet và Ryu. Hai tập dữ liệu riêng biệt đã được tạo ra, gồm lưu lượng bình thường (normal traffic) và lưu lượng tấn công từ chối dịch vụ phân tán (DDoS).

4.1.2.2 Khởi tạo lưu lượng bình thường

Dữ liệu mạng thông thường được thu thập trong điều kiện không có tấn công bằng cách giám sát và ghi nhận các đặc trưng của luồng dữ liệu. Lưu lượng hợp lệ trong quá trình thử nghiệm được tạo ra bằng cách sử dụng các giao thức TCP, UDP và ICMP, nhằm mô phỏng hoạt động mạng thông thường. Giao thức TCP và UDP giúp thiết lập kết nối giữa các địa chỉ IP được chọn ngẫu nhiên, với các cổng giao tiếp

thay đổi liên tục để đảm bảo sự đa dạng của dữ liệu truyền tải. Trong khi đó, ICMP được dùng để thực hiện các truy vấn ping, phản ánh hoạt động kiểm tra kết nối phổ biến trên mạng. Các gói tin này được gửi qua hệ thống thử nghiệm một cách ngẫu nhiên, giúp tạo ra môi trường mạng có tính thực tế cao.



```

minhphuoc@ubuntu: /mnt/hgfs/DACNTT_SDN/Source/controllers
minhphuoc@ubuntu:/mnt/hgfs/DACNTT_SDN/Source/controllers$ ryu-manager capture_normal.py
loading app capture_normal.py
loading app ryu.controller.ofp_handler
instantiating app capture_normal.py of FlowStatsController
instantiating app ryu.controller.ofp_handler of OFPHandler
Switch 1 connected
Switch 2 connected
[]

"Node: h1"
root@ubuntu:/mnt/hgfs/DACNTT_SDN/Source/utlis# sudo python3 generate_normal_traffic.py
ffic.py
Loop: 0
*****
Sent 5 packets.
Sent ICMP packet from 192.168.1.4: to 10.0.2.2:
Loop: 1
*****
Sent 5 packets.
Sent ICMP packet from 192.168.2.2: to 192.168.1.2:
Loop: 2
*****
Sent 5 packets.
Sent ICMP packet from 192.168.1.2: to 192.168.1.4:
Loop: 3
*****
Sent 5 packets.
Sent UDP packet from 192.168.1.2:56516 to 192.168.1.4:57477
Loop: 4
*****
Sent 5 packets.
Sent TCP packet from 192.168.1.2:48784 to 10.0.2.2:39904
Loop: 5
*****
Sent 5 packets.

```

Hình 4.2 Khởi tạo và thu thập lưu lượng bình thường

4.1.2.3 Khởi tạo lưu lượng DDOS

Trong quá trình thử nghiệm, lưu lượng tấn công DDoS được tạo bằng cách sử dụng công cụ hping3, áp dụng các giao thức TCP và UDP nhằm mô phỏng kịch bản tấn công thực tế. Các gói tin được gửi với địa chỉ IP nguồn ngẫu nhiên, liên tục nhắm vào máy chủ đích với tốc độ cao, gây áp lực lên tài nguyên hệ thống. Hình thức TCP flood được sử dụng để gửi các gói tin SYN, làm đầy bảng kết nối của máy chủ, trong khi UDP flood tập trung vào việc truyền tải dữ liệu với cường độ lớn, làm tăng tải xử lý. Để đảm bảo thử nghiệm diễn ra an toàn, số lượng gói tin được giới hạn theo từng giao thức, giúp kiểm soát tác động trong môi trường mô phỏng.

```

minhphuoc@ubuntu: /mnt/hgfs/DACNTT_SDN/Source/controllers$ ryu-manager capture_ddos.py
loading app capture_ddos.py
loading app ryu.controller.ofp_handler
instantiating app capture_ddos.py of FlowStatsController
instantiating app ryu.controller.ofp_handler of OFPHandler
Switch 1 connected
Switch 2 connected
[]

"Node: h1"

root@ubuntu:/mnt/hgfs/DACNTT_SDN/Source/utlis# sudo python3 generate_ddos_traffic.py
Starting DDOS Attack on 192.168.2.2 with packet limits:
TCP: 1000, UDP: 1000
DDOS traffic is running. Use CTRL+C to stop.
root@ubuntu:/mnt/hgfs/DACNTT_SDN/Source/utlis# HPING 192.168.2.2 (h1-eth0 192.168.2.2): S set, 40 headers + 0 data bytes
HPING 192.168.2.2 (h1-eth0 192.168.2.2): udp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
hping in flood mode, no replies will be shown

```

Hình 4.3 Khởi tạo và thu thập lưu lượng DDOS

4.1.2.4 Lưu trữ kết quả dữ liệu

```

network_traffic.csv
dataset > network_traffic.csv

1 timestamp,datapath_id,flow_id,ip_src,tp_src,ip_dst,tp_dst,ip_proto,icmp_code,icmp_type,flow_duration_sec,flow_duration_nsec,idle_timeout,hard_timeout,flags,packet_count
2 1736824844.9656348,2,192.168.1.20,192.168.2.20,192.168.1.2,0,192.168.2.2,0,1,0,0,1,741000000,20,100,0,3,138,3,0,4,048582995951417e-09,138,0,1,8623481781376517e-07,0
3 1736824844.9656348,2,192.168.1.30,192.168.2.20,192.168.1.3,0,192.168.2.2,0,1,0,0,3,878000000,20,100,0,0,0,0,0,0,0,0,0,0,0
4 1736824844.9656348,2,192.168.2.20,192.168.1.20,192.168.2.2,0,192.168.1.2,0,1,0,0,1,735000000,20,100,0,4,184,4,0,5,4421768707483e-09,184,0,2,583481368544218e-07,0
5 1736824844.9698493,1,10.0.2.20,192.168.1.30,10.0.2.2,0,192.168.1.3,0,1,0,0,2,888000000,20,100,0,4,184,2,0,4,950495049504951e-09,92,0,2,2772277227722772e-07,0
6 1736824844.9698493,1,10.0.2.255566,192.168.1.39798,10.0.2.2,55566,192.168.1.3,9798,6,-1,-1,5,5000000,20,100,0,0,0,0,0,0,0,0,0,0,0
7 1736824844.9698493,1,192.168.1.20,192.168.2.20,192.168.1.2,0,192.168.2.2,0,1,0,0,1,745000000,20,100,0,4,184,4,0,5,369127516778523e-09,184,0,2,4697986577181287e-07,0
8 1736824844.9698493,1,192.168.1.30,10.0.2.20,192.168.1.3,0,10.0.2.2,0,1,0,0,2,883000000,20,100,0,0,0,0,0,0,0,0,0,0,0
9 1736824844.9698493,1,192.168.1.39798,10.0.2.255566,192.168.1.3,9798,10.0.2.2,55566,6,-1,-1,5,12000000,20,100,0,0,0,0,0,0,0,0,0,0,0
10 1736824844.9698493,1,192.168.1.30,192.168.1.40,192.168.1.3,0,192.168.1.4,0,1,3,3,0,616000000,20,100,0,0,0,0,0,0,0,0,0,0,0
11 1736824844.9698493,1,192.168.1.30,192.168.2.20,192.168.1.3,0,192.168.2.2,0,1,0,0,3,887000000,20,100,0,0,0,0,0,0,0,0,0,0,0
12 1736824844.9698493,1,192.168.1.465062,192.168.1.342296,192.168.1.4,65862,192.168.1.3,42296,17,-1,-1,0,626000000,20,100,0,0,0,0,0,0,0,0,0,0,0
13 1736824844.9698493,1,192.168.2.20,192.168.1.20,192.168.2.2,0,192.168.1.2,0,1,0,0,1,738000000,20,100,0,4,184,4,0,5,420054200542006e-09,184,0,2,4932249322493225e-07,0
14 1736824844.9698493,1,192.168.2.20,192.168.1.30,192.168.2.2,0,192.168.1.3,0,1,0,0,3,891000000,20,100,0,4,184,1,3333333333333333,4,489337822671156e-09,61,3333333333333336,2
15 1736824854.9683943,2,10.0.2.20,192.168.2.20,10.0.2.2,0,192.168.2.2,0,1,0,0,3,816000000,20,100,0,0,0,0,0,0,0,0,0,0,0
16 1736824854.9683943,2,192.168.1.20,192.168.2.20,192.168.1.2,0,192.168.2.2,0,1,0,0,11,742000000,20,100,0,3,138,0,2727272727272727,4,043126684636119e-09,12,545454545454545,5
17 1736824854.9683943,2,192.168.1.30,192.168.2.20,192.168.1.3,0,192.168.2.2,0,1,0,0,13,879000000,20,100,0,5,238,0,3846153846153846,5,688282138794884e-09,17,692387692387693
18 1736824854.9683943,2,192.168.1.40,192.168.2.20,192.168.1.4,0,192.168.2.2,0,1,0,0,2,707000000,20,100,0,4,184,2,0,5,657708628005658e-09,92,0,2,682545968882683e-07,0
19 1736824854.9683943,2,192.168.1.440502,192.168.2.222673,192.168.1.4,48502,192.168.2.2,222673,6,-1,-1,7,169000000,20,100,0,4,216,0,6714285714285714,2,3668633985325444e-08,38
20 1736824854.9683943,2,192.168.2.20,10.0.2.20,192.168.2.2,0,10.0.2.2,0,1,0,0,3,809000000,20,100,0,4,184,1,3333333333333333,4,944375772558714e-09,61,3333333333333336,2,274411

```

Hình 4.4 Kết quả thu thập dữ liệu bình thường và DDOS

Dữ liệu thu thập từ lưu lượng mạng được lưu trữ và xử lý thông qua việc sử dụng Ryu SDN Controller, trong đó các luồng dữ liệu được ghi nhận từ các switch OpenFlow. Hệ thống theo dõi trạng thái của các switch, gửi yêu cầu truy vấn thông tin thống kê về các luồng mạng và trích xuất các đặc trưng quan trọng sau:

Tên Feature	Mô tả
timestamp	Dấu thời gian ghi nhận khi luồng dữ liệu được phát hiện hoặc xử lý.
datapath_id	ID của switch OpenFlow chịu trách nhiệm xử lý luồng dữ liệu.
flow_id	Định danh duy nhất của luồng, dựa trên địa chỉ IP nguồn, IP đích và giao thức sử dụng.
ip_src	Địa chỉ IP nguồn của gói tin.
tp_src	Cổng nguồn (TCP/UDP) mà gói tin sử dụng.
ip_dst	Địa chỉ IP đích của gói tin.
tp_dst	Cổng đích (TCP/UDP) mà gói tin hướng đến.
ip_proto	Giao thức mạng được sử dụng, ví dụ: TCP, UDP, ICMP.
icmp_code	Mã ICMP được sử dụng để xác định loại lỗi hoặc thông điệp ICMP.
icmp_type	Loại ICMP, thể hiện hành vi hoặc loại thông điệp của gói tin ICMP.
flow_duration_sec	Thời gian tồn tại của luồng tính theo giây.

flow_duration_nsec	Thời gian tồn tại của luồng tính theo nanosecond.
idle_timeout	Khoảng thời gian chờ trước khi luồng bị xóa do không có bất kỳ hoạt động nào.
hard_timeout	Thời gian tối đa trước khi luồng bị xóa, ngay cả khi vẫn còn hoạt động.
flags	Cờ điều khiển của gói tin (ví dụ: SYN, ACK, FIN, RST), đặc biệt quan trọng trong giao thức TCP.
packet_count	Tổng số lượng gói tin trong luồng dữ liệu.
byte_count	Tổng số byte được truyền trong toàn bộ luồng dữ liệu.
packet_count_per_second	Số lượng gói tin truyền trong một giây.
packet_count_per_nsecond	Số lượng gói tin truyền trong một nanosecond.
byte_count_per_second	Lưu lượng truyền dữ liệu tính bằng byte mỗi giây.
byte_count_per_nsecond	Lưu lượng truyền dữ liệu tính bằng byte mỗi nanosecond.
label	Nhãn của luồng: "0" cho lưu lượng hợp lệ và "1" cho lưu lượng DDoS.

Bảng 4.1 Mô tả các đặc trưng của dữ liệu thu thập

4.1.3 Tiền xử lý dữ liệu

```
# Preprocess dataset
dataset['ip_src'] = dataset['ip_src'].str.replace('.', '')
dataset['ip_dst'] = dataset['ip_dst'].str.replace('.', '')
dataset['flow_id'] = dataset['flow_id'].str.replace('.', '')

features = ['ip_src', 'ip_dst', 'tp_src', 'tp_dst', 'flow_duration_nsec', 'flags',
            'packet_count', 'flow_duration_sec', 'byte_count',
            'packet_count_per_second', 'byte_count_per_second']

X = dataset[features].values.astype("float")
y = dataset['label'].values

# Chia dữ liệu thành tập huấn luyện và kiểm tra
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42)
```

Hình 4.5 Tiền xử lý dữ liệu

Trước khi đưa vào mô hình học máy, tập dữ liệu cần trải qua quá trình tiền xử lý nhằm đảm bảo định dạng phù hợp. Đầu tiên, các trường địa chỉ IP nguồn (ip_src), địa chỉ IP đích (ip_dst) và ID luồng (flow_id) được chuyển đổi sang dạng số bằng cách loại bỏ ký tự dấu chấm (.). Việc này giúp giảm độ phức tạp khi xử lý dữ liệu chuỗi. Tiếp theo, một nhóm các đặc trưng quan trọng được chọn, bao gồm thông tin về địa chỉ, cổng, thời gian, số lượng gói tin, lượng dữ liệu truyền tải và một số chỉ số liên quan đến tốc độ. Toàn bộ các giá trị đặc trưng được ép kiểu về dạng số (float) trước khi chia dữ liệu thành hai phần: tập huấn luyện (chiếm 75%) và tập kiểm tra (chiếm 25%). Việc tách dữ liệu này giúp đảm bảo rằng mô hình được đào tạo trên một phần dữ liệu và được đánh giá dựa trên phần còn lại, từ đó phản ánh khả năng tổng quát hóa tốt hơn.

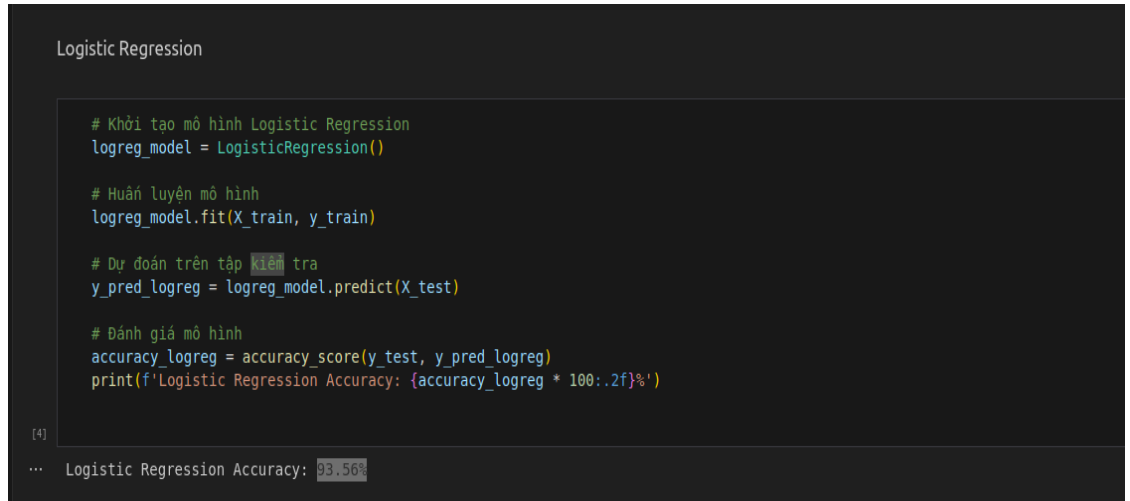
4.2 Huấn luyện và triển khai mô hình phát hiện DDOS

4.2.1 Chọn mô hình và phương pháp huấn luyện

4.2.1.1 Huấn luyện Logistic Regression

Mô hình Logistic Regression được sử dụng đầu tiên trong bài toán phân loại lưu lượng mạng, với mục đích phân loại lưu lượng mạng vào hai nhóm: bình thường và tấn công. Quy trình huấn luyện bao gồm việc khởi tạo mô hình, huấn luyện trên

tập dữ liệu huấn luyện và sau đó dự đoán kết quả trên tập kiểm tra. Logistic Regression sẽ phân tích mối quan hệ giữa các đặc trưng mạng và nhãn của lưu lượng để đưa ra quyết định phân loại.



```

Logistic Regression

# Khởi tạo mô hình Logistic Regression
logreg_model = LogisticRegression()

# Huấn luyện mô hình
logreg_model.fit(X_train, y_train)

# Dự đoán trên tập kiểm tra
y_pred_logreg = logreg_model.predict(X_test)

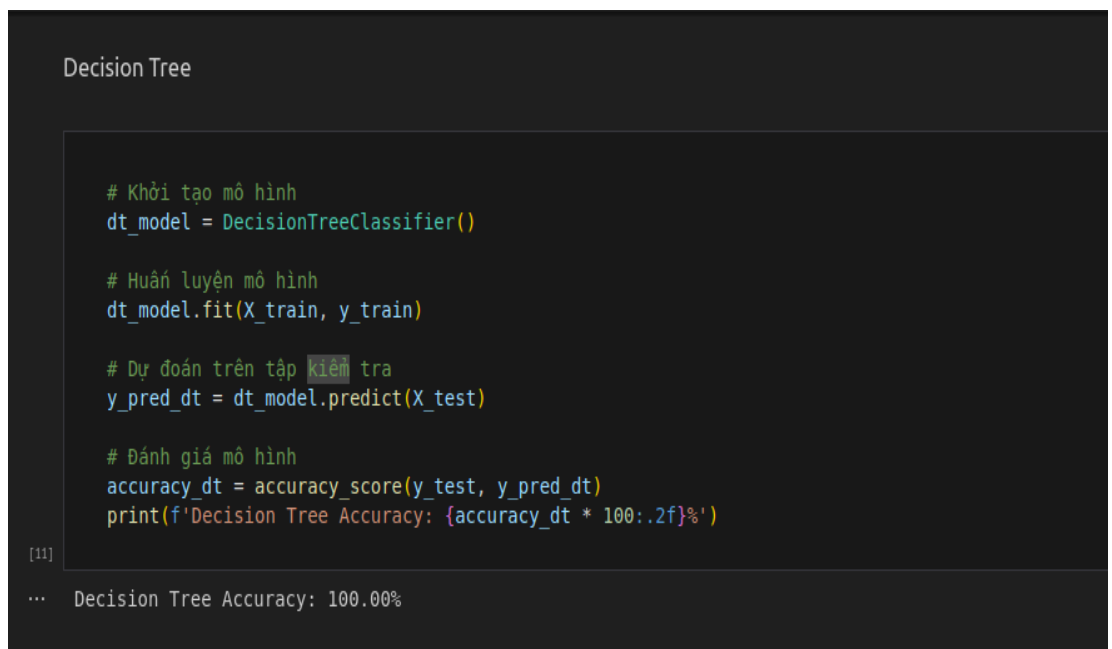
# Đánh giá mô hình
accuracy_logreg = accuracy_score(y_test, y_pred_logreg)
print(f'Logistic Regression Accuracy: {accuracy_logreg * 100:.2f}%')

[4]
... Logistic Regression Accuracy: 93.56%

```

Hình 4.6 Huấn luyện Logistic Regression

4.2.1.2 Huấn luyện Decision Tree



```

Decision Tree

# Khởi tạo mô hình
dt_model = DecisionTreeClassifier()

# Huấn luyện mô hình
dt_model.fit(X_train, y_train)

# Dự đoán trên tập kiểm tra
y_pred_dt = dt_model.predict(X_test)

# Đánh giá mô hình
accuracy_dt = accuracy_score(y_test, y_pred_dt)
print(f'Decision Tree Accuracy: {accuracy_dt * 100:.2f}%')

[11]
... Decision Tree Accuracy: 100.00%

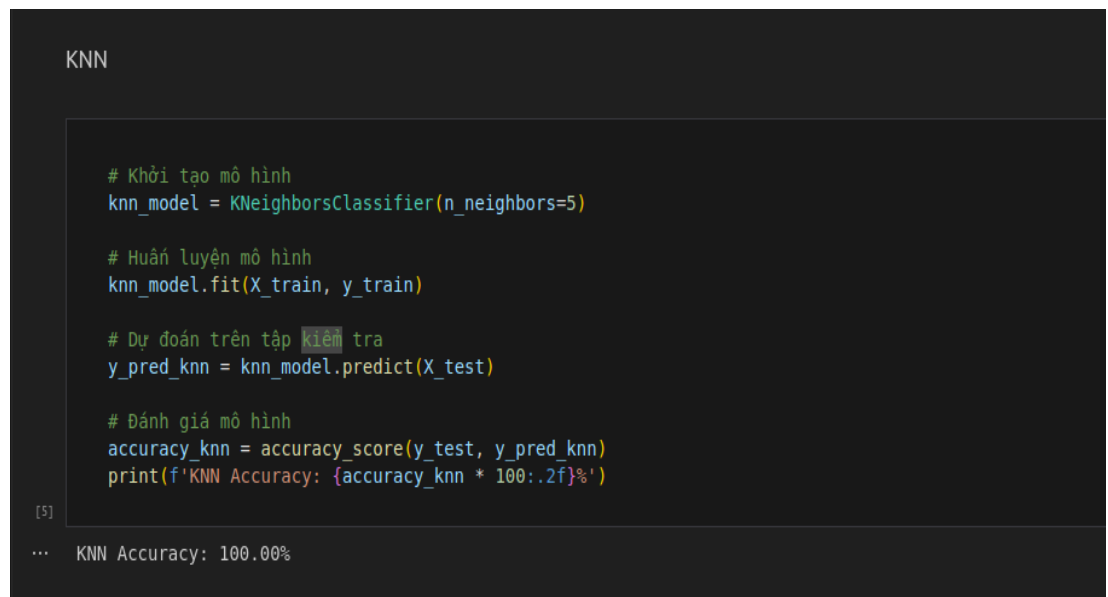
```

Hình 4.7 Huấn luyện Decision Tree

Mô hình Decision Tree cũng được áp dụng trong việc phát hiện tấn công DDoS. Quy trình huấn luyện của Decision Tree tương tự như Logistic Regression, nhưng với thuật toán phân loại dựa trên việc xây dựng một cây quyết định để phân

chia dữ liệu vào các nhánh dựa trên các đặc trưng đầu vào. Decision Tree dễ hiểu và trực quan, giúp mô hình dễ dàng giải thích các quyết định phân loại, đồng thời cũng có thể xử lý tốt dữ liệu phi tuyến tính.

4.2.1.3 Huấn luyện KNN



The screenshot shows a Jupyter Notebook cell titled 'KNN'. It contains Python code for training and testing a K-Nearest Neighbors classifier. The code sets the number of neighbors to 5, fits the model on training data, predicts on test data, and prints the accuracy. The output shows 100% accuracy.

```

KNN

# Khởi tạo mô hình
knn_model = KNeighborsClassifier(n_neighbors=5)

# Huấn luyện mô hình
knn_model.fit(X_train, y_train)

# Dự đoán trên tập kiểm tra
y_pred_knn = knn_model.predict(X_test)

# Đánh giá mô hình
accuracy_knn = accuracy_score(y_test, y_pred_knn)
print(f'KNN Accuracy: {accuracy_knn * 100:.2f}%')

[5]
... KNN Accuracy: 100.00%

```

Hình 4.8 Huấn luyện KNN

K-Nearest Neighbors (KNN) là một mô hình học máy đơn giản nhưng hiệu quả, đặc biệt là trong các bài toán phân loại dựa trên sự tương đồng giữa các điểm dữ liệu. Với KNN, mỗi điểm dữ liệu trong tập kiểm tra sẽ được phân loại dựa trên các điểm dữ liệu gần nhất trong không gian đặc trưng. Mô hình KNN được huấn luyện và sử dụng để dự đoán loại tấn công DDoS bằng cách tính toán khoảng cách giữa các điểm và xác định nhãn của các điểm láng giềng gần nhất.

4.2.1.4 Huấn luyện SVM

Support Vector Machine (SVM) là một trong những mô hình mạnh mẽ nhất trong học máy, được sử dụng để phân loại dữ liệu trong không gian cao chiều. SVM tìm kiếm một siêu phẳng tối ưu để phân chia các lớp dữ liệu sao cho khoảng cách giữa các điểm của các lớp khác nhau là lớn nhất. Quá trình huấn luyện của SVM bao gồm việc tối ưu hóa hàm mất mát để xác định siêu phẳng phân chia tốt nhất. Với

SVM, mô hình có khả năng phát hiện tấn công DDoS hiệu quả, đặc biệt là khi dữ liệu có sự phân tách rõ ràng.

```

SVM

# Khởi tạo mô hình
svm_model = SVC(kernel='rbf')

# Huấn luyện mô hình
svm_model.fit(X_train, y_train)

# Dự đoán trên tập kiểm tra
y_pred_svm = svm_model.predict(X_test)

# Đánh giá mô hình
accuracy_svm = accuracy_score(y_test, y_pred_svm)
print(f'SVM Accuracy: {accuracy_svm * 100:.2f}%')

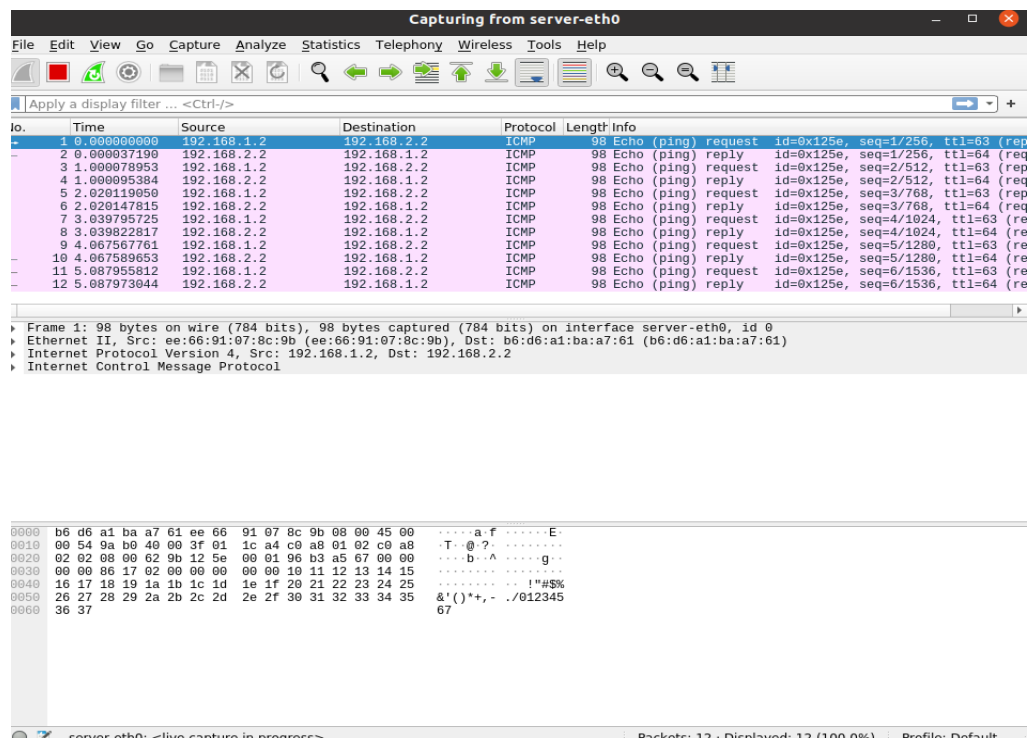
[14]
... SVM Accuracy: 92.58%

```

Hình 4.9 Huấn luyện SVM

4.2.2 Phát hiện tấn công trong thời gian thực

4.2.2.1 Kiểm tra lưu lượng bình thường



Hình 4.10 Lưu lượng bình thường

Sau khi thực hiện lệnh ping từ h1 tới server thì sẽ dùng wireshark để kiểm tra kết nối thông qua giao thức ICMP, giúp xác minh khả năng liên lạc giữa hai thiết bị trong mạng. Kết quả từ quá trình này cho thấy các gói tin phản hồi đầy đủ, không có hiện tượng mất mát hoặc trễ bất thường, chứng tỏ hệ thống mạng đang vận hành ổn định.



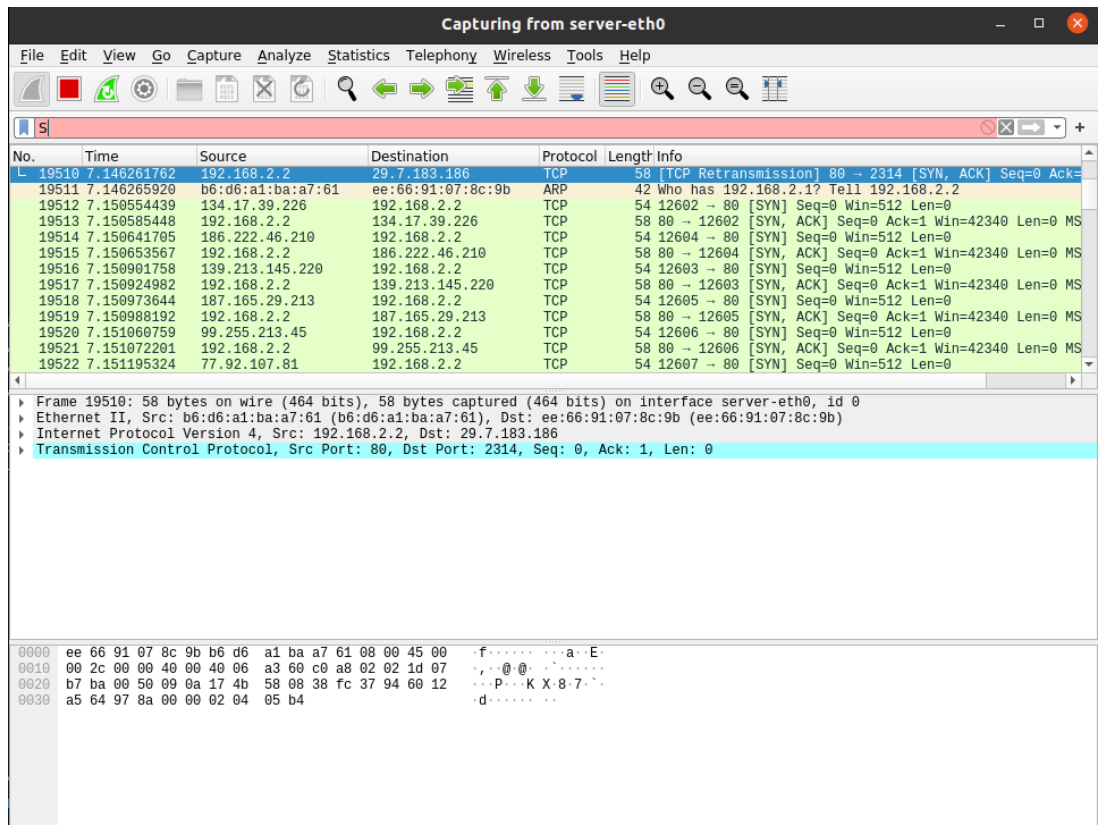
Hình 4.11 Biểu đồ phân tích lưu lượng bình thường

Biểu đồ hiển thị số lượng gói tin được truyền qua giao diện mạng theo thời gian. Dữ liệu thu thập cho thấy lưu lượng dao động trong khoảng 2 đến 4 gói/giây, phản ánh trạng thái bình thường của hệ thống. Việc kiểm tra và phân tích lưu lượng hợp lệ giúp cung cấp một mô hình tham chiếu, hỗ trợ quá trình phát hiện các luồng dữ liệu bất thường trong các tình huống có tấn công.

4.2.2.2 Kiểm tra lưu lượng DDOS

Sau khi thực hiện lệnh `hping3 -S -p 80 --rand-source --flood` từ attacker tới server thì sẽ dùng wireshark để kiểm tra Dữ liệu thu thập từ Wireshark cho thấy một

cuộc tấn công DDoS vào máy chủ server với số lượng lớn gói tin TCP SYN gửi đến cổng 80. Hầu hết các gói tin này không hoàn tất quá trình bắt tay ba bước, dẫn đến hiện tượng TCP Retransmission do máy chủ liên tục phải xử lý các yêu cầu kết nối giả mạo. Điều này khiến tài nguyên hệ thống bị tiêu tốn quá mức, làm giảm hiệu suất và có thể dẫn đến tình trạng từ chối dịch vụ đối với các kết nối hợp lệ.



No.	Time	Source	Destination	Protocol	Length	Info
19510	7.146261762	192.168.2.2	29.7.183.186	TCP	58	[TCP Retransmission] 80 → 2314 [SYN, ACK] Seq=0 Ack=
19511	7.146265920	b6:d6:a1:ba:a7:61	ee:66:91:07:8c:9b	ARP	42	Who has 192.168.2.1? Tell 192.168.2.2
19512	7.150554439	134.17.39.226	192.168.2.2	TCP	54	12602 → 80 [SYN] Seq=0 Win=512 Len=0
19513	7.150585448	192.168.2.2	134.17.39.226	TCP	58	80 → 12602 [SYN, ACK] Seq=0 Ack=1 Win=42340 Len=0 MS
19514	7.150641705	186.222.46.210	192.168.2.2	TCP	54	12604 → 80 [SYN] Seq=0 Win=512 Len=0
19515	7.150653567	192.168.2.2	186.222.46.210	TCP	58	80 → 12604 [SYN, ACK] Seq=0 Ack=1 Win=42340 Len=0 MS
19516	7.150901758	139.213.145.220	192.168.2.2	TCP	54	12603 → 80 [SYN] Seq=0 Win=512 Len=0
19517	7.150924982	192.168.2.2	139.213.145.220	TCP	58	80 → 12603 [SYN, ACK] Seq=0 Ack=1 Win=42340 Len=0 MS
19518	7.150973644	187.165.29.213	192.168.2.2	TCP	54	12605 → 80 [SYN] Seq=0 Win=512 Len=0
19519	7.150988192	192.168.2.2	187.165.29.213	TCP	58	80 → 12605 [SYN, ACK] Seq=0 Ack=1 Win=42340 Len=0 MS
19520	7.151060759	99.255.213.45	192.168.2.2	TCP	54	12606 → 80 [SYN] Seq=0 Win=512 Len=0
19521	7.151072201	192.168.2.2	99.255.213.45	TCP	58	80 → 12606 [SYN, ACK] Seq=0 Ack=1 Win=42340 Len=0 MS
19522	7.151195324	77.92.107.81	192.168.2.2	TCP	54	12607 → 80 [SYN] Seq=0 Win=512 Len=0

Frame 19510: 58 bytes on wire (464 bits), 58 bytes captured (464 bits) on interface server-eth0, id 0
 Ethernet II, Src: b6:d6:a1:ba:a7:61 (b6:d6:a1:ba:a7:61), Dst: ee:66:91:07:8c:9b (ee:66:91:07:8c:9b)
 Internet Protocol Version 4, Src: 192.168.2.2, Dst: 29.7.183.186
 Transmission Control Protocol, Src Port: 80, Dst Port: 2314, Seq: 0, Ack: 1, Len: 0

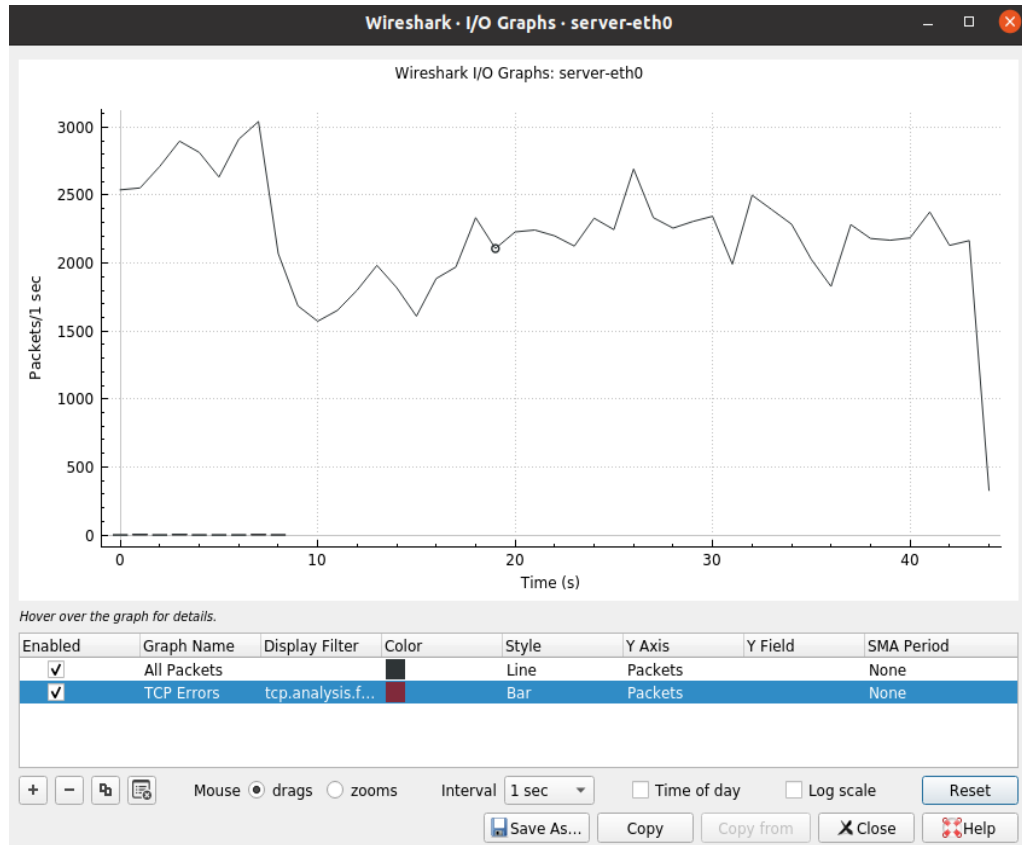
```

0000  ee 66 91 07 8c 9b b6 d6  a1 ba a7 61 08 00 45 00  .f.....a..E-
0010  00 2c 00 00 40 00 00 06  a3 60 c0 a8 02 02 1d 07  .,....@.....
0020  b7 ba 00 50 09 0a 17 4b  58 08 38 fc 37 94 60 12  .P...KX8-7...
0030  a5 64 97 8a 00 00 02 04  05 b4                                     .d.....
  
```

Hình 4.12 Lưu lượng DDOS

Biểu đồ bên dưới đây hiển thị lưu lượng mạng với sự gia tăng đáng kể, có thời điểm vượt quá 3000 packets mỗi giây. Cùng với sự tăng vọt này, số lượng lỗi TCP cũng xuất hiện với tần suất cao, điều này cho thấy có thể đang diễn ra một sự bất thường trong quá trình truyền tải dữ liệu. Đặc biệt, khi lưu lượng biến động mạnh kèm theo tỷ lệ lỗi lớn, đây là dấu hiệu điển hình của một cuộc tấn công từ chối dịch vụ phân tán (DDoS). Cụ thể hơn, mô hình này là một cuộc tấn công SYN Flood, trong đó kẻ tấn công gửi một lượng lớn gói tin SYN nhằm lợi dụng cơ chế bắt tay ba bước của giao thức TCP để làm cạn kiệt tài nguyên của máy chủ mục tiêu. Việc quan sát

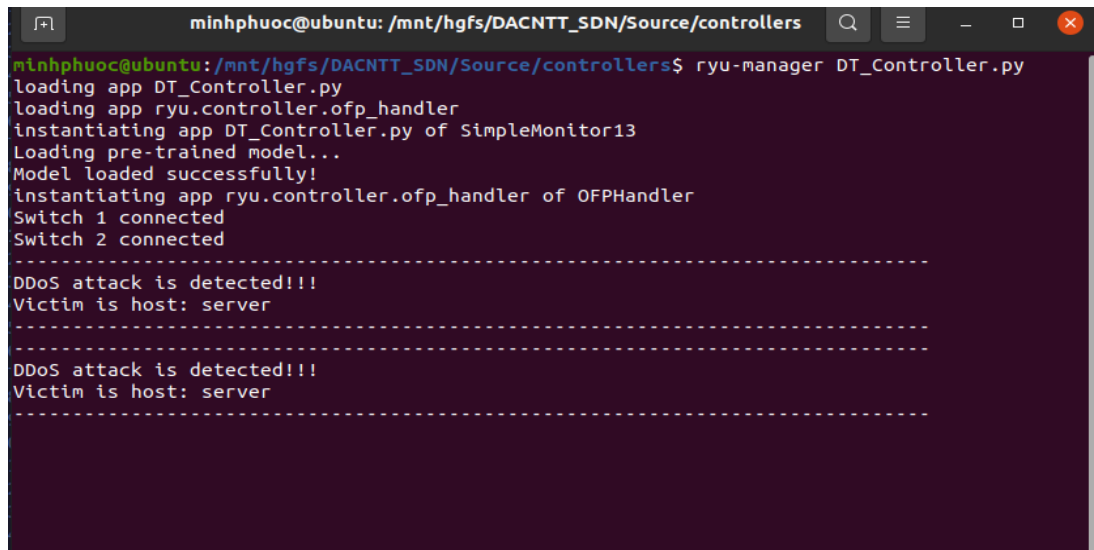
kỹ xu hướng này có thể giúp phát hiện sớm và triển khai các biện pháp phòng thủ thích hợp để giảm thiểu tác động tiêu cực đến hệ thống mạng.



Hình 4.13 Biểu đồ phân tích lưu lượng DDOS

4.2.2.3 Áp dụng mô hình Decion tree vào Ryu Controller

Sau khi hoàn tất quá trình huấn luyện các mô hình thì chúng em chọn Decision Tree để thực nghiệm, tệp mô hình đã được lưu dưới định dạng .pkl để dễ dàng tích hợp vào hệ thống. Tiếp theo, mô hình này được triển khai trong Ryu Controller nhằm hỗ trợ việc giám sát và phát hiện tấn công DDoS theo thời gian thực. Khi controller khởi chạy, nó tải mô hình đã huấn luyện và sử dụng để phân tích các luồng dữ liệu mạng đang diễn ra. Nếu phát hiện có dấu hiệu bất thường phù hợp với hành vi của một cuộc tấn công DDoS, hệ thống sẽ đưa ra cảnh báo, đồng thời xác định mục tiêu bị tấn công.



```

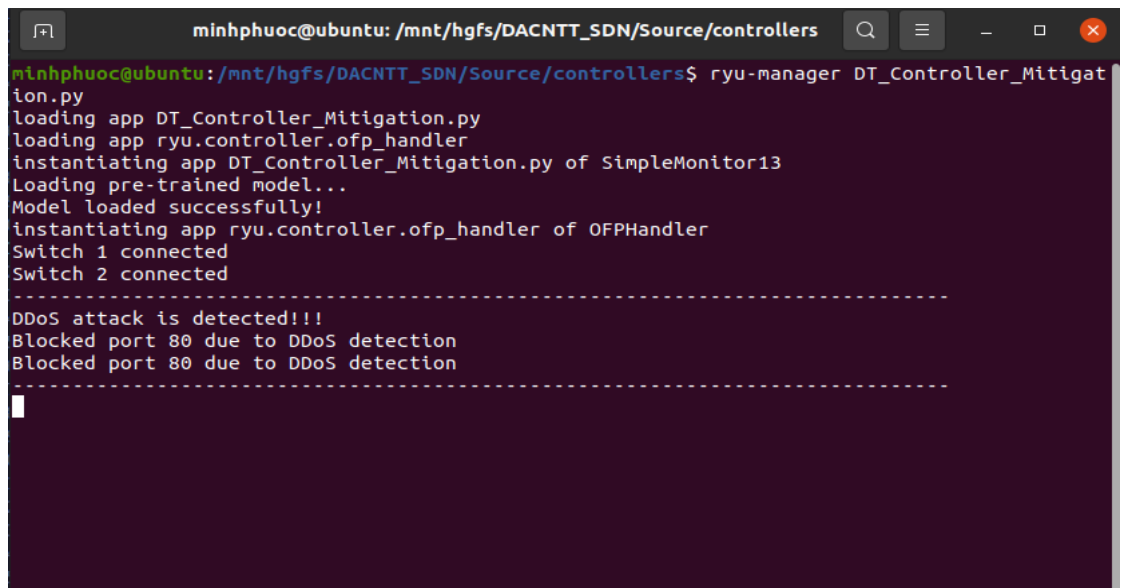
minhphuoc@ubuntu: /mnt/hgfs/DACNTT_SDN/Source/controllers
minhphuoc@ubuntu:/mnt/hgfs/DACNTT_SDN/Source/controllers$ ryu-manager DT_Controller.py
loading app DT_Controller.py
loading app ryu.controller.ofp_handler
instantiating app DT_Controller.py of SimpleMonitor13
Loading pre-trained model...
Model loaded successfully!
instantiating app ryu.controller.ofp_handler of OFPHandler
Switch 1 connected
Switch 2 connected
-----
DDoS attack is detected!!!
Victim is host: server
-----
DDoS attack is detected!!!
Victim is host: server
-----

```

Hình 4.14 Controller phát hiện DDOS

4.3 Thử nghiệm biện pháp phòng chống DDOS

Sau khi phát hiện tấn công DDoS bằng mô hình Decision Tree tích hợp trong Ryu Controller, hệ thống đã thực hiện biện pháp giảm thiểu bằng cách chặn lưu lượng trên cổng 80. Khi controller xác nhận có lưu lượng bất thường trùng khớp với các đặc trưng của cuộc tấn công, nó sẽ áp dụng chính sách chặn, ngăn chặn các gói tin đi qua cổng này. Điều này giúp hạn chế tác động của cuộc tấn công, bảo vệ hệ thống trước tình trạng tài nguyên bị quá tải do lưu lượng độc hại.



```

minhphuoc@ubuntu: /mnt/hgfs/DACNTT_SDN/Source/controllers
minhphuoc@ubuntu:/mnt/hgfs/DACNTT_SDN/Source/controllers$ ryu-manager DT_Controller_Mitigation.py
loading app DT_Controller_Mitigation.py
loading app ryu.controller.ofp_handler
instantiating app DT_Controller_Mitigation.py of SimpleMonitor13
Loading pre-trained model...
Model loaded successfully!
instantiating app ryu.controller.ofp_handler of OFPHandler
Switch 1 connected
Switch 2 connected
-----
DDoS attack is detected!!!
Blocked port 80 due to DDoS detection
Blocked port 80 due to DDoS detection
-----

```

Hình 4.15 Block port chống DDOS

4.4 Đánh giá và phân tích kết quả

4.4.1 Tiêu chí đánh giá

Việc đánh giá hiệu suất mô hình phát hiện tấn công DDoS trong môi trường SDN được thực hiện dựa trên các thước đo quan trọng, bao gồm:

- **Accuracy:** Chỉ số này thể hiện tỷ lệ mẫu được dự đoán đúng so với tổng số mẫu. Tuy nhiên, nếu dữ liệu mất cân bằng, độ chính xác có thể không phản ánh đầy đủ hiệu quả của mô hình.
- **Precision:** Đánh giá xem trong số các mẫu bị nhận diện là tấn công, có bao nhiêu mẫu thực sự thuộc về nhóm này. Precision cao giúp giảm thiểu các cảnh báo sai.
- **Recall:** Cho biết tỷ lệ các cuộc tấn công được phát hiện chính xác so với tổng số cuộc tấn công thực tế. Giá trị recall cao đồng nghĩa với việc mô hình ít bỏ sót các trường hợp nguy hiểm.
- **F1-score:** Là giá trị trung bình điều hòa giữa precision và recall, giúp cân bằng giữa khả năng phát hiện tấn công và hạn chế cảnh báo sai.

4.4.2 So sánh kết quả huấn luyện giữa các mô hình học máy

Mặc dù Decision Tree và KNN đạt độ chính xác tuyệt đối 100%, nhưng điều này có thể là dấu hiệu của tình trạng overfitting. Cả hai mô hình này đều có precision, recall và F1-score đạt mức tối đa (1.00 cho cả hai lớp), cho thấy chúng đã ghi nhớ hoàn toàn dữ liệu huấn luyện thay vì học được các đặc trưng tổng quát. Trong môi trường thực tế, nơi dữ liệu có thể có nhiều biến động và chứa các mẫu chưa từng thấy trước đó, mô hình overfitting thường hoạt động kém do thiếu khả năng tổng quát hóa. Vì vậy, cần tiến hành kiểm tra trên các tập dữ liệu thực tế để đánh giá hiệu suất thực sự, thay vì chỉ dựa vào kết quả trên tập huấn luyện.

Trong khi đó, Logistic Regression đạt độ chính xác 93.56%, với precision của lớp bình thường là 0.95 và lớp tấn công là 0.93, recall tương ứng là 0.92 và 0.95, dẫn đến F1-score lần lượt là 0.93 và 0.94. Điều này cho thấy mô hình có hiệu suất khá ổn định, với sự cân bằng tốt giữa precision và recall, giúp hạn chế cả việc phân loại sai

các gói tin hợp lệ lẫn bỏ sót các cuộc tấn công. Tuy nhiên, tỷ lệ recall của lớp bình thường chỉ đạt 0.92, có nghĩa là vẫn có một số gói tin hợp lệ bị nhận diện nhầm là tấn công, điều này có thể gây ảnh hưởng nếu hệ thống phòng thủ sử dụng mô hình này để chặn lưu lượng mạng.

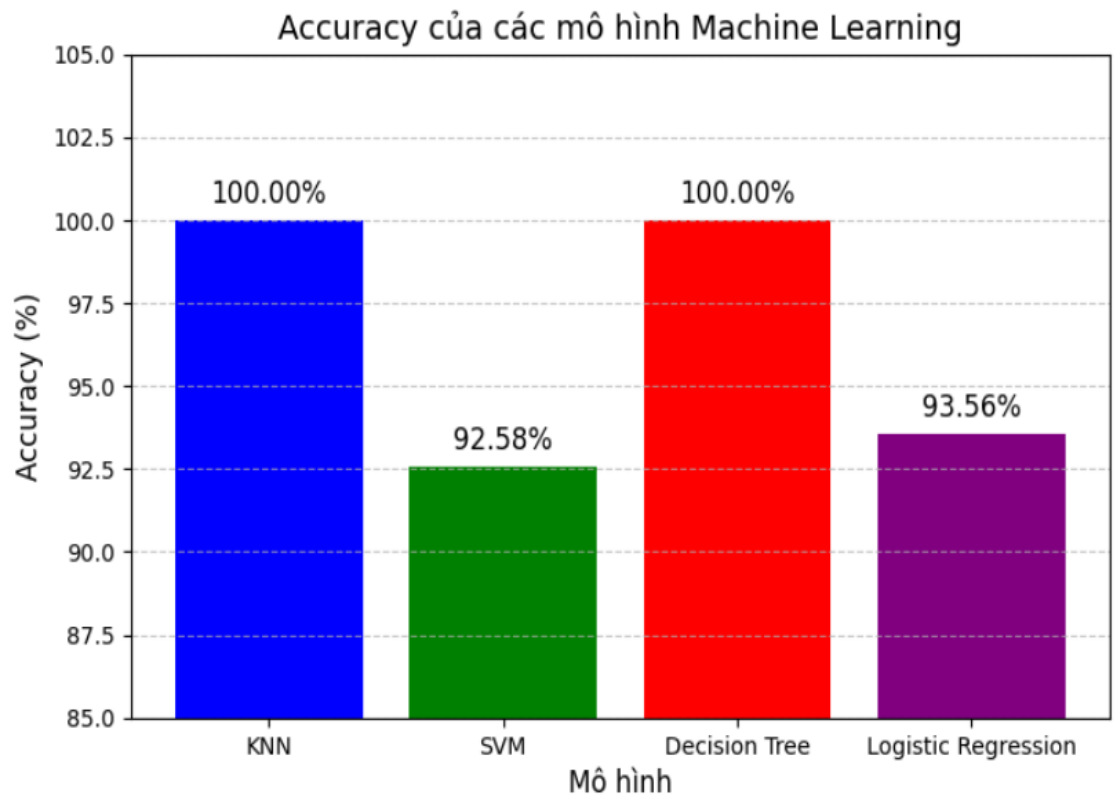
SVM có độ chính xác thấp nhất trong số các mô hình, chỉ đạt 92.58%. Điều đáng chú ý là precision của lớp bình thường là 0.86, recall đạt 1.00, trong khi lớp tấn công có precision đạt 1.00 nhưng recall chỉ là 0.86. Điều này cho thấy mô hình có xu hướng bỏ sót các cuộc tấn công thực tế nhiều hơn so với Logistic Regression. Nguyên nhân có thể đến từ việc lựa chọn kernel chưa phù hợp hoặc SVM chưa được tối ưu hóa tốt cho đặc điểm của tập dữ liệu. Điều này đặc biệt quan trọng trong bối cảnh phát hiện tấn công mạng, khi recall thấp có thể dẫn đến nguy cơ bỏ sót nhiều cuộc tấn công nguy hiểm.

Logistic Regression Accuracy: 93.56%					Decision Tree Accuracy: 100.00%				
Classification Report:					Classification Report:				
	precision	recall	f1-score	support		precision	recall	f1-score	support
0	0.95	0.92	0.93	70001	0	1.00	1.00	1.00	70001
1	0.93	0.95	0.94	78693	1	1.00	1.00	1.00	78693
accuracy			0.94	148694	accuracy			1.00	148694
macro avg	0.94	0.93	0.94	148694	macro avg	1.00	1.00	1.00	148694
weighted avg	0.94	0.94	0.94	148694	weighted avg	1.00	1.00	1.00	148694

KNN Accuracy: 100.00%					SVM Accuracy: 92.58%				
Classification Report:					Classification Report:				
	precision	recall	f1-score	support		precision	recall	f1-score	support
0	1.00	1.00	1.00	70001	0	0.86	1.00	0.93	70001
1	1.00	1.00	1.00	78693	1	1.00	0.86	0.92	78693
accuracy			1.00	148694	accuracy			0.93	148694
macro avg	1.00	1.00	1.00	148694	macro avg	0.93	0.93	0.93	148694
weighted avg	1.00	1.00	1.00	148694	weighted avg	0.94	0.93	0.93	148694

Hình 4.16 Chỉ số đánh giá giữa các mô hình

Nhìn chung, mặc dù Decision Tree và KNN đạt kết quả rất cao trên tập dữ liệu huấn luyện, nhưng cần thử nghiệm thêm trên dữ liệu thực tế để trực quan hơn. Trong khi đó, Logistic Regression và SVM có độ chính xác thấp hơn nhưng có thể ổn định hơn trong môi trường thực tế. Một cách tiếp cận hiệu quả hơn là kết hợp nhiều mô hình để cải thiện khả năng phát hiện tấn công, hoặc tinh chỉnh tham số và thử nghiệm thêm các kỹ thuật giảm overfitting để đảm bảo hệ thống phát hiện hoạt động tốt trong điều kiện thực tế.



Hình 4.17 Accuracy giữa các mô hình

CHƯƠNG 5. KẾT LUẬN

5.1 Kết luận

Trong nghiên cứu này, một phương pháp phát hiện tấn công từ chối dịch vụ phân tán (DDoS) trong mạng điều khiển bằng phần mềm (SDN) đã được đề xuất và triển khai. Bằng cách ứng dụng các thuật toán học máy, hệ thống có khả năng nhận diện các mẫu lưu lượng bất thường, giúp phát hiện sớm các cuộc tấn công và hạn chế tác động tiêu cực đến hệ thống mạng. Kết quả thực nghiệm cho thấy mô hình đề xuất đạt hiệu suất cao, với độ chính xác tốt trong việc phân loại lưu lượng hợp lệ và lưu lượng tấn công.

Ngoài ra, việc tích hợp phương pháp này vào bộ điều khiển SDN giúp tăng cường tính bảo mật mà không làm ảnh hưởng đáng kể đến hiệu suất hoạt động của mạng. Sự khác biệt giữa các mô hình học máy cũng đã được phân tích, từ đó lựa chọn được phương pháp tối ưu phù hợp với bối cảnh triển khai thực tế.

5.2 Hướng phát triển

Mặc dù mô hình đề xuất đã đạt được kết quả khả quan, vẫn tồn tại một số hạn chế cần khắc phục để nâng cao hiệu quả trong thực tế. Trước tiên, việc mở rộng tập dữ liệu huấn luyện với nhiều dạng tấn công mới sẽ giúp mô hình thích ứng tốt hơn với các mối đe dọa chưa từng gặp. Bên cạnh đó, tối ưu hóa thuật toán nhằm cải thiện tốc độ xử lý, đặc biệt trong môi trường có lưu lượng mạng lớn, là một hướng quan trọng. Việc triển khai trên hệ thống phân tán hoặc điện toán biên có thể giúp giảm tải và nâng cao khả năng ứng dụng thực tế. Ngoài ra, kết hợp cơ chế phản ứng linh hoạt, tự động điều chỉnh chính sách bảo mật theo thời gian thực sẽ giúp hệ thống không chỉ phát hiện mà còn chủ động ngăn chặn các cuộc tấn công một cách hiệu quả hơn.

TÀI LIỆU THAM KHẢO

Tiếng Việt

...

Tiếng Anh

Kumar, S., & Kush, A. (2023). Machine Learning-Based DDoS Attack Detection in Software-Defined Networking. *Applied Sciences*, 13(5), 3183. <https://www.mdpi.com/2076-3417/13/5/3183>

Rahouti, M., Xiong, K., Xin, Y., Jagatheesaperumal, S. K., Ayyash, M., & Shaheed, M. (2022). SDN Security Review: Threat Taxonomy, Implications, and Open Challenges. *IEEE Access*, 10, 43176–43203. <https://ieeexplore.ieee.org/abstract/document/9760465>

Karnani, S., & Shakya, H. K. (2022). Mitigation Strategies for Distributed Denial of Service (DDoS) in SDN: A Survey and Taxonomy. *Information Security Journal: A Global Perspective*, 32(4), 1–25. https://www.researchgate.net/publication/364259319_Mitigation_strategies_for_distributed_denial_of_service_DDoS_in_SDN_A_survey_and_taxonomy

Al-Hawawreh, M., Jararweh, Y., & Gupta, B. (2023). Deep Learning-Based DDoS Attack Detection in SDN Environments. *IEEE Transactions on Network and Service Management*. <https://ieeexplore.ieee.org/abstract/document/10746482>

Ahmed, M., Tariq, M., Mahmood, A. N., & Wahab, A. (2023). A Comprehensive Survey on Low-Rate and High-Rate DDoS Defense Approaches in SDN: Taxonomy, Research Challenges, and Opportunities. *ResearchGate*. https://www.researchgate.net/publication/374292562_A_comprehensive_survey_on_low-rate_and_high-rate_DDoS_defense_approaches_in_SDN_taxonomy_research_challenges_and_opportunities