TRƯỜNG ĐẠI HỌC BÁCH KHOA - ĐẠI HỌC ĐÀ NẴNG

KHOA ĐIỆN TỬ - VIỄN THÔNG

o0o

BÁO CÁO CUỐI KỲ
MẠNG ĐỊNH NGHĨA BẰNG PHẦN MỀM

Đề tài:

Phát Hiện Và Ngăn Chặn Tấn Công DDoS Trong SDN Sử Dụng Mô Hình Deep Learning

Giảng viên hướng dẫn : TS. Tăng Anh Tuấn

Sinh viên thực hiện :	Nguyễn Ngọc Trung	21KTMT2
	Lê Dương Khang	21KTMT2
	Hoàng Bảo Long	21KTMT1

Lớp học phần : 21.44

Nhóm : 2

Đà Nẵng, tháng 6 năm 2025

Mục lục

Lời cảm ơn

Với sự cho phép và hướng dẫn tận tình từ Thầy, nhóm chúng em đã có cơ hội thực hiện đề tài: *"Phát Hiện Và Ngăn Chặn Tấn Công DDoS Trong SDN Sử Dụng Mô Hình Deep Learning"*

Để hoàn thành tốt đề tài này, chúng em xin bày tỏ lòng biết ơn sâu sắc đến *Thầy Tăng Anh Tuấn*, người đã trực tiếp hướng dẫn, hỗ trợ và theo sát chúng em trong suốt quá trình nghiên cứu và thực hiện dự án học phần môn *Mạng Định Nghĩa Bằng Phần Mềm.*

Qua từng buổi trao đổi học thuật và quá trình chỉnh sửa báo cáo, thầy đã không chỉ truyền đạt những kiến thức chuyên môn quý báu mà còn giúp nhóm rèn luyện tư duy logic, khả năng tự học và định hướng nghiên cứu khoa học một cách đúng đắn.

Chúng em cũng chân thành cảm ơn thầy vì sự tận tâm, nghiêm túc trong giảng dạy, luôn sẵn sàng lắng nghe, giải đáp và động viên nhóm vượt qua những khó khăn trong quá trình triển khai mô hình, xây dựng hệ thống và xử lý kết quả. Những kiến thức, kinh nghiệm và định hướng của Thầy chính là nền tảng vững chắc giúp chúng em hoàn thành tốt dự án cũng như làm hành trang quý giá cho quá trình học tập và nghiên cứu sau này.

Mặc dù thời gian thực hiện có giới hạn, nhưng thông qua dự án này, chúng em đã hiểu rõ hơn về cách tiếp cận một vấn đề thực tế trong lĩnh vực mạng, cũng như cách xây dựng và đánh giá một hệ thống hoàn chỉnh. Những gì nhóm thu nhận được không chỉ là kết quả của một học phần, mà còn là một trải nghiệm đáng quý trong hành trình rèn luyện bản thân.

Một lần nữa, chúng em xin gửi lời cảm ơn chân thành nhất đến thầy!

Đà Nẵng, ngày 20 tháng 6 năm 2025

Phụ lục

Thuật ngữ	Ý nghĩa
Ảo hóa mạng	Tạo nhiều mạng ảo trên cùng hạ tầng vật lý.
Control Plane	Quản lý, định tuyến và điều khiển mạng.
Controller	Bộ điều khiển trung tâm trong SDN.
DDoS	Distributed Denial of Service - Tấn công làm quá tải dịch vụ từ nhiều nguồn.
Data Plane	Chuyển tiếp gói tin trong mạng.
Decision Tree	Mô hình phân loại dạng cây.
Địa chỉ MAC	Định danh phần cứng thiết bị mạng.
Địa chỉ IP	Định danh và định vị thiết bị trong mạng.
Flow Entry / Flow Table	Bảng quy tắc xử lý luồng dữ liệu.
HTTP	Giao thức truyền dữ liệu web.
ICMP Flood	Tấn công bằng gói ICMP để gây quá tải.
KNN	K-Nearest Neighbors - Thuật toán phân loại dựa trên hàng xóm gần nhất.
Logistic Regression	Thuật toán phân loại nhị phân.
MLP	Multi-Layer Perceptron - Mô hình mạng nơ-ron nhiều lớp.
Navie Bayes	Phân loại dựa trên Định lý Bayes.
OpenFlow	Giao thức giữa Controller và thiết bị SDN.
Router	Thiết bị định tuyến giữa các mạng.
Random Forest	Tập hợp nhiều cây quyết định.
SVM	Support Vector Machine - Thuật toán phân loại giám sát.
SYN Flood	Tấn công TCP không hoàn tất kết nối.
SMURF Flood	Tấn công ICMP qua địa chỉ broadcast.
SDN	Software Defined Network - Mạng định nghĩa bằng phần mềm.
TCAM	Ternary Content-Addressable Memory - Bộ nhớ tìm kiếm nhanh các luồng.

TCP	Transmission Control Protocol - Giao thức truyền dữ liệu đáng tin cậy.
UDP	User Datagram Protocol - Giao thức truyền dữ liệu không kết nối.
UDP Flood	Tấn công DDoS bằng gói tin UDP.

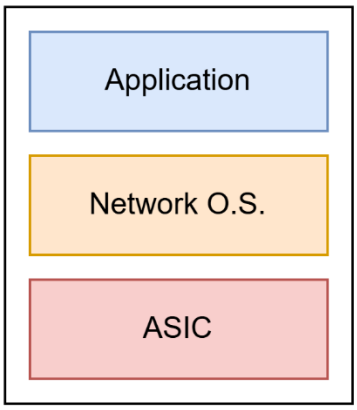
Công cụ	**Ý nghĩa**
Linux	Hệ điều hành mã nguồn mở, ổn định, dùng phổ biến cho mạng và server.
Mininet	Công cụ mô phỏng mạng SDN nhanh chóng và nhẹ.
Python	Ngôn ngữ lập trình phổ biến cho mạng và học máy.
Ryu Controller	Bộ điều khiển SDN mã nguồn mở, dễ triển khai.
Scikit-learn	Thư viện Python cho các thuật toán học máy.
Ubuntu	Hệ điều hành Linux phổ biến.
VirtualBox	Phần mềm tạo máy ảo để chạy hệ điều hành khác.
Wireshark	Công cụ giám sát và phân tích lưu lượng mạng.

Chương 1: Giới thiệu

1.1. Bối cảnh

Trong các hệ thống mạng truyền thống, switch được thiết kế dựa trên kiến trúc phân lớp theo chiều dọc (Current Switch Vertical Stack), như minh họa ở *Hình 1.1*. Mô hình này bao gồm ba lớp chức năng chính:

- **Ứng dụng (Applications):** Đây là các ứng dụng mạng, chúng có thể thực hiện các tác vụ như định tuyến, giám sát, bảo mật, và phân bổ tài nguyên trong mạng.
- **Hệ điều hành mạng (Network OS):** Hệ điều hành này đảm nhiệm việc quản lý và điều phối các phần mềm và phần cứng của thiết bị mạng.
- **ASIC (Application-Specific Integrated Circuit):** Phần cứng tích hợp, thường được sử dụng trong các switch vật lý, chịu trách nhiệm xử lý dữ liệu mạng theo các quy tắc và cài đặt có sẵn.



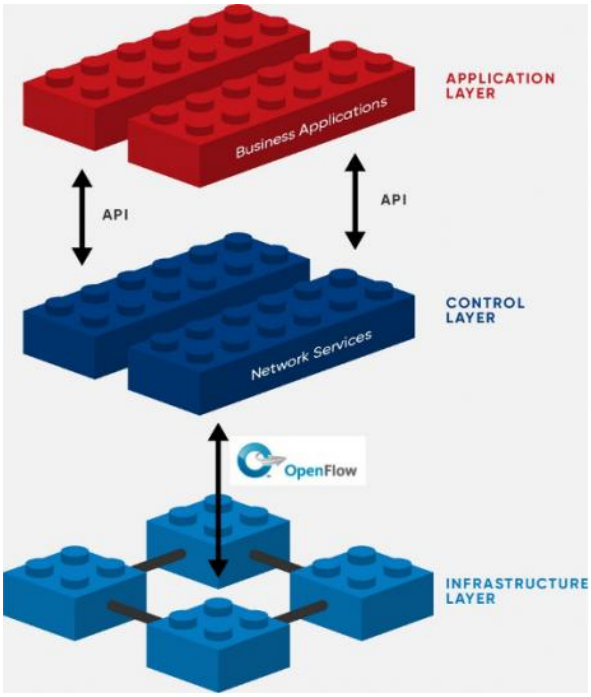
Hình 1.1 *Kiến trúc truyền thống của switch*

Mặc dù kiến trúc này từng là nền tảng của các hệ thống mạng doanh nghiệp và trung tâm dữ liệu, nhưng nó dần bộc lộ nhiều hạn chế khi mạng trở nên phức tạp và quy mô ngày càng lớn:

- **Quản lý phân tán phức tạp:** Mỗi thiết bị phải được cấu hình thủ công hoặc riêng biệt, gây khó khăn trong việc đồng bộ và mở rộng mạng.
- **Thiếu tính linh hoạt:** Khó cập nhật chính sách mạng đồng bộ cho toàn hệ thống; thay đổi chính sách phải thực hiện từng thiết bị.
- **Thiếu khả năng tự động hóa:** Không thể dễ dàng lập trình hoặc kiểm soát hành vi mạng theo thời gian thực.
- **Khó mở rộng và tối ưu:** Các giao thức như STP, OSPF... không dễ dàng mở rộng lên quy mô lớn; việc tối ưu luồng dữ liệu giữa nhiều switch gặp giới hạn.
- **Phản ứng chậm với sự cố:** Do mỗi thiết bị tự xử lý, nên việc phản ứng trước sự cố không được tối ưu hoặc không có cái nhìn toàn cục.

Chính vì những hạn chế trên thúc đẩy sự ra đời của Mạng định nghĩa bằng phần mềm (Software-Defined Networking – SDN) để tách mặt phẳng điều khiển ra khỏi các thiết bị mạng vật lý, đưa toàn bộ điều khiển mạng về một điểm tập trung (controller), từ đó dễ lập trình, quản lý và tự động hóa mạng.

SDN là một phương pháp quản lý mạng cho phép cấu hình mạng động, hiệu quả theo chương trình để cải thiện hiệu suất và giám sát mạng. Đây là một cách mới để quản lý mạng máy tính giúp chúng dễ dàng và linh hoạt hơn khi kiểm soát.



Hình 1.2 *Kiến trúc của Mạng định nghĩa bằng phần mềm (SDN) [1]*

Về mặt kiến trúc, SDN được phân tách rõ ràng thành ba lớp chức năng chính *(Hình 1.2)*, được trình bày chi tiết trong *Bảng 1.1* dưới đây.

Bảng 1.1 *Bảng mô tả các lớp trong kiến trúc của SDN*

Lớp	Định nghĩa	Vai trò
Application Layer (Application plane)	Là lớp cao nhất, nơi các ứng dụng mạng hoạt động (giám sát, bảo mật, tối ưu hóa).	Cung cấp dịch vụ quản lý và tối ưu hóa mạng.
Control Layer (Controller plane)	Là lớp trung gian, trung tâm là SDN controller.	Điều phối, ra quyết định định tuyến, quản lý băng thông.
Infrastructure Layer (Data plane)	Gồm các thiết bị vật lý (switch, router, firewall).	Thực hiện các hành động mạng theo yêu cầu từ controller.

Để ba lớp trong kiến trúc SDN có thể tương tác với nhau hiệu quả, các giao diện chuyên biệt được định nghĩa, đóng vai trò như cầu nối giữa các tầng. *Bảng 1.2* dưới đây trình bày chi tiết về các giao diện kết nối quan trọng trong SDN.

Bảng 1.2 *Bảng mô tả các giao diện để giao tiếp kiết nối giữa các lớp trong SDN*

Giao diện	Định nghĩa	Vai trò
Northbound Interface (NBI)	Là giao diện kết nối giữa Application Layer và Control Layer trong kiến trúc SDN.	Cho phép các ứng dụng giao tiếp với SDN Controller để gửi yêu cầu điều khiển mạng hoặc nhận dữ liệu phản hồi. Sử dụng giao thức A-CPI.
Southbound Interface (SBI)	Là giao diện kết nối giữa Control Layer và Infrastructure Layer.	Cho phép SDN Controller gửi lệnh điều khiển xuống các thiết bị mạng như switch/router. Sử dụng giao thức D-CPI.

Sự khác biệt giữa SDN và mạng truyền thống không chỉ dừng lại ở mặt kiến trúc, mà còn thể hiện rõ trong cách tiếp cận quản lý và vận hành hệ thống mạng. *Bảng 1.3* sau đây sẽ so sánh một số đặc điểm chính giữa hai mô hình này.

Bảng 1.3 *Bảng so sánh giữa Mạng định nghĩa bằng phần mềm và Mạng truyền thống*

Mạng định nghĩa bằng phần mềm	Mạng truyền thống
Mạng được xác định bằng phần mềm là một phương pháp tiếp cận mạng ảo.	Mạng truyền thống là phương pháp tiếp cận mạng thông thường cũ.
Mạng được xác định bằng phần mềm là hệ thống điều khiển tập trung.	Mạng truyền thống là mạng phân tán.
Mạng này có thể lập trình được.	Mạng này không thể lập trình được.
Mạng được xác định bằng phần mềm là giao diện mở.	Mạng truyền thống là một giao diện khép kín.
Trong mặt phẳng dữ liệu và điều khiển Mạng được xác định bằng phần mềm, mặt phẳng được tách biệt bằng phần mềm.	Trong mạng truyền thống, mặt phẳng dữ liệu và mặt phẳng điều khiển được gắn trên cùng một mặt phẳng.

Qua đó ta có thể thấy được mạng định nghĩa bằng phần mềm (SDN) có những ưu điểm phù hợp hơn trong bối cảnh hiện nay để thay thế cho mạng switch truyền thống, tuy nhiên mạng định nghĩa bằng phần mềm cũng có những nhược điểm và *Bảng 1.4* sẽ mô tả tóm tắt ưu nhược điểm của mạng SDN sau đây:

Bảng 1.4 *Bảng tóm tắt ưu nhược điểm của mạng SDN*

Ưu điểm	Nhược điểm
Quản lý và điều khiển mạng tập trung	Phụ thuộc vào bộ điều khiển trung tâm

Khả năng lập trình linh hoạt, dễ cấu hình mạng	Khó khăn trong việc chuyển đổi từ mạng truyền thống
Hỗ trợ ảo hóa mạng, tối ưu tài nguyên	Đòi hỏi nhân lực có chuyên môn cao
Triển khai và cập nhật tính năng mới nhanh chóng	Nguy cơ bảo mật cao do điều khiển tập trung
Hỗ trợ hiệu quả cho các dịch vụ điện toán đám mây	Chi phí triển khai ban đầu cao

Mặc dù có một số nhược điểm như trên, nhưng SDN vẫn được coi là một công nghệ mạng tiên tiến và có tiềm năng lớn trong tương lai. SDN đã mở ra nhiều ứng dụng tiềm năng trong các lĩnh vực khác nhau. Một số ứng dụng quan trọng bao gồm:

- **Mạng di động:** SDN có thể hỗ trợ việc triển khai mạng di động hiệu quả hơn, cho phép quản lý tài nguyên mạng và cấu hình chính sách mạng một cách linh hoạt.
- **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.

1.2. Động lực nghiên cứu

SDN là một cách tiếp cận mang tính cách mạng giúp tách mặt phẳng điều khiển khỏi mặt phẳng dữ liệu, tập trung trí thông minh mạng trong bộ điều khiển SDN. Mặc dù kiến trúc này mang lại tính linh hoạt, khả năng mở rộng và dễ quản lý, nhưng nó cũng đưa ra một số thách thức bảo mật. SDN controller là bộ não của mạng, là mục tiêu chính của các mối đe dọa mạng, nếu bộ controller bị tấn công hoặc bị kiểm soát thì toàn bộ mạng có thể bị sập hoặc thao túng.

Chính vì vậy, bảo mật và phát hiện bất thường giữ vai trò thiết yếu trong SDN, không chỉ nhằm bảo vệ hệ thống khỏi các mối đe dọa tiềm ẩn, mà còn đảm bảo duy trì tính ổn định, tin cậy và hiệu suất vận hành của toàn mạng. Cụ thể, tầm quan trọng của chúng được thể hiện qua các khía cạnh sau:

- **Phát hiện sớm các cuộc tấn công mạng:** như DDoS, ARP spoofing, MiTM, scan port, worm propagation, … SDN cho phép giám sát luồng dữ liệu theo thời gian thực, giúp phát hiện dấu hiệu bất thường ngay từ khi khởi phát.
- **Giúp bảo vệ controller và các thiết bị mạng:** Vì controller là trung tâm điều khiển nên luôn là mục tiêu ưu tiên của tin tặc, hệ thống phát hiện bất thường giúp bảo vệ nó khỏi bị xâm nhập.
- **Đảm bảo tính sẵn sàng và toàn vẹn của mạng:** Phát hiện bất thường giúp duy trì hoạt động ổn định, tránh mất dịch vụ do các hành vi sai lệch gây ra (cấu hình sai, khai thác lỗi,...)

- **Hỗ trợ tự động phản ứng:** Hệ thống SDN có thể tự động cập nhật rule trong switch để cô lập, redirect, hoặc block các lưu lượng đáng ngờ ngay khi phát hiện.

1.3. Vấn đề nghiên cứu

Trong môi trường mạng hiện đại dựa trên kiến trúc SDN, khả năng phát hiện bất thường đóng vai trò then chốt trong việc duy trì an toàn và ổn định hệ thống. Bài toán phát hiện bất thường trong SDN tập trung vào việc xác định sớm các luồng dữ liệu, hành vi thiết bị, hoặc yêu cầu điều khiển không tuân theo đặc trưng hoạt động bình thường của mạng, từ đó chủ động đưa ra phản ứng phù hợp để ngăn chặn hoặc cảnh báo nguy cơ tấn công.

Khác với mạng truyền thống, SDN cung cấp khả năng quan sát mạng tập trung thông qua controller – nơi tập hợp toàn bộ thông tin trạng thái và lưu lượng của hệ thống. Điều này tạo điều kiện thuận lợi để xây dựng hệ thống giám sát và phát hiện bất thường theo thời gian thực dựa trên các dữ liệu đầu vào như:

- **Dữ liệu luồng (flow entries):** Bao gồm các trường đặc trưng như địa chỉ IP nguồn/đích, địa chỉ MAC, cổng giao tiếp, giao thức, thời lượng tồn tại của luồng,...
- **Thống kê lưu lượng (OpenFlow stats):** Số lượng gói tin, byte, tốc độ truyền tải qua từng cổng và từng luồng, giúp đánh giá hành vi của thiết bị mạng.
- **Sự kiện bất thường:** Các hiện tượng bất thường như tỷ lệ rớt gói cao, số lượng kết nối tăng đột biến, hoặc lưu lượng không đều giữa các nút mạng.

Để đảm bảo hiệu quả trong môi trường mạng động và phức tạp, hệ thống cần đáp ứng các mục tiêu cốt lõi:

- Phân biệt được hành vi bình thường và bất thường (có bị tấn công hay không).
- Đảm bảo độ trễ thấp và phản ứng nhanh.
- Tích hợp dễ dàng với hệ thống điều khiển SDN.

1.4. Mục tiêu nghiên cứu

Dự án này tập trung khai thác tiềm năng của học sâu trong việc phát triển mô hình phát hiện và ngăn chặn các cuộc tấn công DDoS (Distributed Denial of Service) trong môi trường mạng SDN. Khác với các phương pháp học máy truyền thống, học sâu có khả năng tự động trích xuất đặc trưng và học các mối quan hệ phi tuyến phức tạp trong dữ liệu lưu lượng mạng, từ đó nâng cao hiệu quả phát hiện các hành vi bất thường.

Mục tiêu chính của dự án là xây dựng một hệ thống SDN hoàn chỉnh, mô phỏng các kịch bản tấn công DDoS phổ biến, đồng thời triển khai và đánh giá hiệu quả của mô hình. Trong quá trình thực nghiệm, các mô hình học sâu được huấn luyện bằng dữ liệu thống kê từ các switch trong mạng SDN, bao gồm các thông tin như lưu lượng, số lượng gói tin, thời gian sống của luồng và các chỉ số bất thường khác.

1.5. Phạm vi và giới hạn

Nhằm đảm bảo tính rõ ràng và tập trung cho dự án này, cần xác định rõ phạm vi và giới hạn của nghiên cứu. Mặc dù mạng SDN mang lại nhiều lợi ích về quản lý và bảo mật, hệ thống

này cũng đối mặt với nhiều thách thức, đặc biệt là trước các cuộc tấn công DDoS. Phạm vi và giới hạn của dự án được xác định trong *Bảng 1.5* và *Bảng 1.6*:

Bảng 1.5 *Phạm vi nghiên cứu của dự án*

Phạm vi	Chi tiết
Loại hình tấn công DDoS	Báo cáo tập trung vào các dạng tấn công DDoS phổ biến như SYN Flood, UDP Flood, ICMP Flood, với mục tiêu phát hiện và ngăn chặn lưu lượng tấn công.
Môi trường thí nghiệm	Nghiên cứu được triển khai trong môi trường mô phỏng sử dụng Mininet và Ryu Controller, áp dụng giao thức OpenFlow để quản lý lưu lượng.
Công nghệ và thuật toán	Sử dụng học sâu để phân tích hành vi lưu lượng và phát hiện bất thường.
Dữ liệu sử dụng	Báo cáo sử dụng tập dữ liệu đã được gán nhãn, bao gồm cả lưu lượng bình thường và lưu lượng tấn công, nhằm đảm bảo độ chính xác khi đánh giá mô hình.
Đánh giá hiệu quả	Sử dụng các chỉ số như Accuracy, Precision, Recall, F1-Score, và AUC để đánh giá mô hình phát hiện tấn công.

Bảng 1.6 *Giới hạn của dự án*

Giới hạn	Chi tiết
Phạm vi dữ liệu	Không bao gồm tất cả các kiểu tấn công DDoS mới như Slow-rate Attack hoặc Low-volume Attack.
Môi trường thử nghiệm	Chỉ thực hiện trên mô phỏng, không kiểm chứng trên mạng thực tế với lưu lượng lớn.
Độ trễ phản hồi	Không tập trung vào tối ưu hóa độ trễ phản hồi của hệ thống khi phát hiện tấn công.
Khả năng mở rộng	Chưa đánh giá khả năng mở rộng khi số lượng thiết bị và lưu lượng mạng tăng đột biến.
Giới hạn về bảo mật vật lý	Không bao gồm các giải pháp bảo mật vật lý như mã hóa hoặc chứng thực thiết bị.

1.6. Cấu trúc báo cáo

Chương 1 – Giới thiệu trình bày tổng quan về công nghệ mạng định nghĩa bằng phần mềm (SDN) như một giải pháp thay thế kiến trúc mạng truyền thống, nhằm cải thiện khả năng quản lý, lập trình và mở rộng mạng thông qua việc tách biệt mặt phẳng điều khiển và dữ liệu. Tuy nhiên, chính sự tập trung hóa điều khiển này cũng kéo theo các rủi ro bảo mật nghiêm trọng, đặc biệt là nguy cơ tấn công vào controller. Từ đó, chương nhấn mạnh tầm quan trọng của việc phát hiện bất thường trong SDN để bảo vệ hệ thống khỏi các cuộc tấn công như DDoS.

Cuối cùng, chương nêu rõ mục tiêu nghiên cứu là xây dựng một hệ thống phát hiện DDoS sử dụng học sâu trong môi trường mô phỏng SDN, đồng thời xác định rõ phạm vi và giới hạn của dự án.

Chương 2 – Tổng quan tài liệu trình bày các thách thức bảo mật trong SDN, đặc biệt là nguy cơ tấn công DDoS do đặc điểm điều khiển tập trung của kiến trúc này. Các hình thức tấn công như làm nghẽn controller, đầy bảng ghi hoặc bão hòa băng thông có thể gây tê liệt toàn mạng. Chương cũng giới thiệu các phương pháp phát hiện DDoS từ truyền thống (dựa trên chữ ký, bất thường) đến hiện đại (học máy, học sâu), cùng các giải pháp như giám sát thời gian thực, phân tích hành vi và cảnh báo sớm. Cuối cùng, chương chỉ ra những hạn chế còn tồn tại như độ trễ cao, khả năng mở rộng kém và độ chính xác chưa tối ưu, làm cơ sở định hướng cho nghiên cứu cải tiến sau này.

Chương 3 – Phương pháp nghiên cứu trình bày quy trình xây dựng hệ thống phát hiện và ngăn chặn tấn công DDoS trong môi trường SDN mô phỏng bằng Mininet và điều khiển bởi Ryu. Dữ liệu lưu lượng bao gồm cả bình thường và tấn công được thu thập, gán nhãn và xử lý để trích xuất các đặc trưng quan trọng. Mô hình học sâu MLP với hai lớp ẩn được sử dụng để phân loại lưu lượng nhờ khả năng học quan hệ phi tuyến và triển khai dễ dàng trong thời gian thực. Sau huấn luyện, mô hình được tích hợp vào controller để tự động nhận diện và xử lý lưu lượng bất thường, đảm bảo tính linh hoạt và hiệu quả trong phát hiện và ngăn chặn tấn công DDoS.

Chương 4 – Kết quả và thảo luận trình bày quá trình đánh giá hiệu quả của mô hình phát hiện tấn công DDoS trong môi trường SDN thông qua các chỉ số đánh giá như Accuracy, Precision, Recall, F1-Score và AUC. Hệ thống được triển khai trên môi trường giả lập sử dụng Mininet và Ryu Controller. Hai mô hình được so sánh là MLP (học sâu) và Random Forest (học máy truyền thống), nhằm phân loại lưu lượng mạng thành tấn công và bình thường. Ngoài ra, chương còn phân tích hiệu quả của hệ thống qua ba kịch bản triển khai: phát hiện không can thiệp, phát hiện và chặn cổng tấn công, phát hiện và giới hạn băng thông.

Chương 5 – Kết luận nhằm thể hiện nội dung chính là tổng hợp những kết quả đạt được từ quá trình nghiên cứu và triển khai mô hình phát hiện tấn công DDoS trong mạng SDN, đồng thời đánh giá hiệu quả, đóng góp, hạn chế và tiềm năng phát triển của hệ thống. Thông qua việc so sánh mô hình học sâu MLP với các phương pháp truyền thống như Random Forest, chương này làm rõ tính khả thi của việc ứng dụng học sâu vào bài toán bảo mật SDN. Bên cạnh việc nhìn nhận những điểm mạnh, chương cũng thẳng thắn chỉ ra các giới hạn thực tế của mô hình, từ đó đề xuất các hướng phát triển cần thiết để nâng cao hiệu quả phát hiện và khả năng ứng phó linh hoạt với các mối đe dọa mạng trong tương lai.

Chương 2: Tổng quan

2.1. Những thách thức về bảo mật trong SDN trước DDoS

Trong mạng SDN, controller đóng vai trò trung tâm, chịu trách nhiệm điều khiển và quản lý toàn bộ lưu lượng mạng, trong khi các switch chỉ đơn thuần chuyển tiếp gói tin theo các quy tắc đã được cài đặt. Mặc dù mô hình này mang lại tính linh hoạt và khả năng quản lý tập trung, nó cũng tạo ra một "điểm yếu" tiềm ẩn. Khi controller hoặc bảng ghi (TCAM - Ternary Content Addressable Memory) trên các switch bị quá tải, toàn bộ hệ thống có thể gặp phải hiện tượng chậm trễ, mất gói hoặc thậm chí ngừng hoạt động.

Một trong những mối đe dọa phổ biến nhất khai thác điểm yếu này là tấn công DDoS. Kẻ tấn công có thể sử dụng botnet để phát tán lưu lượng bất thường, làm tiêu hao băng thông, CPU của controller, hoặc bộ nhớ TCAM của switch, dẫn đến nghẽn mạng. Mặc dù chỉ cần một lưu lượng tấn công vừa phải, nhưng do mỗi luồng mới đều phải xử lý tại controller, kết quả là toàn bộ mạng bị chậm trễ hoặc rớt gói.

DDoS trong SDN thường khai thác theo các hướng sau:

- **Làm nghẽn mặt phẳng điều khiển:** Botnet liên tục gửi các gói "không khớp bảng" đến switch, buộc switch phải gửi Packet-In về controller. Các bản tin này nhanh chóng làm quá tải CPU và băng thông south-bound, dẫn đến nghẽn mạng.
- **Làm đầy bảng ghi trên switch:** Hàng triệu luồng ngắn, với các tiêu đề ngẫu nhiên, chiếm sạch TCAM của switch. Khi TCAM đầy, các gói hợp lệ không thể thêm quy tắc mới và bị loại bỏ.
- **Bão hòa băng thông lõi:** Khai thác các giao thức dễ khuếch đại như DNS (Domain Name System), NTP (Network Time Protocol), hoặc Memcached để tạo ra lưu lượng khổng lồ, vượt quá năng lực đường truyền. Quá trình sinh quy tắc chặn lại phụ thuộc vào controller, dẫn đến phản hồi chậm và nghẽn mạng.

Như vậy, chính ưu điểm "điều khiển tập trung" của SDN lại trở thành mục tiêu lý tưởng cho DDoS. Khả năng tấn công vào một điểm nhưng ảnh hưởng đến toàn bộ mạng khiến việc phát hiện sớm và giảm nhẹ tác động của DDoS trở thành yêu cầu cấp thiết cho mọi hệ thống SDN quy mô lớn.

2.2. Các phương pháp phát hiện DDoS

Để phát hiện các cuộc tấn công DDoS trong mạng SDN, có nhiều phương pháp đã được nghiên cứu và triển khai, từ các kỹ thuật truyền thống như dựa trên chữ ký đến các phương pháp hiện đại dựa trên học máy, học sâu. Mỗi phương pháp đều có những điểm mạnh và hạn chế riêng, phù hợp với các kịch bản và mục tiêu khác nhau. *Bảng 2.1* dưới đây tóm lược các phương pháp phát hiện DDoS phổ biến.

Bảng 2.1 *Bảng tóm tắt các phương pháp phát hiện tấn công DDoS trong SDN*

Phương pháp	Nguyên lý	Điểm mạnh	Hạn chế
Dựa trên chữ ký	So khớp lưu lượng với mẫu đã biết	Triển khai nhanh, hầu như không báo động giả	Không phát hiện biến thể mới, cần cập nhật liên tục
Dựa trên bất thường	Mô hình hóa hành vi bình thường, gắn cờ khi lệch chuẩn	Phát hiện cả tấn công chưa có chữ ký	Dễ báo động giả, phải huấn luyện và đặt ngưỡng phù hợp
Học máy truyền thống	Phân loại dựa trên đặc trưng thủ công (tốc độ gói, thời gian luồng…)	Nhẹ, dễ giải thích, phù hợp tập dữ liệu nhỏ	Hiệu quả kém với mẫu phức tạp, phụ thuộc khâu chọn đặc trưng
Học sâu	Học biểu diễn ẩn của chuỗi gói, NetFlow	Tự trích đặc trưng, bắt quan hệ phi tuyến	Cần dữ liệu lớn và GPU, độ tin cậy tùy chất lượng huấn luyện

2.3. Giải pháp phát hiện DDoS trong SDN

Để đối phó với các cuộc tấn công DDoS trong mạng SDN, nhiều phương pháp đã được phát triển, mỗi phương pháp có những đặc điểm và ưu thế riêng, phù hợp với các kịch bản và mục tiêu khác nhau. *Bảng 2.2* dưới đây tóm lược các giải pháp phổ biến nhất cùng với những lợi thế nổi bật của chúng, giúp tối ưu hóa khả năng phát hiện và phản ứng trước các mối đe dọa an ninh mạng.

Bảng 2.2 *Tổng hợp các phương pháp phát hiện và ngăn chặn tấn công DDoS trong SDN*

Phương pháp	Đặc điểm chính	Ưu điểm
Phân tích lưu lượng	Dùng sFlow, NetFlow hoặc OpenFlow để phát hiện bất thường qua tốc độ, kích thước và biến động gói tin	So khớp lưu lượng với mẫu đã biết
Phân tích hành vi	Áp dụng các mô hình học máy và học sâu để phát hiện các mẫu bất thường	Mô hình hóa hành vi bình thường, gắn cờ khi lệch chuẩn
Giám sát lưu lượng thời gian thực	Đặt ngưỡng cảnh báo khi lưu lượng vượt mức bình thường, tuy nhiên cần tinh chỉnh để giảm cảnh báo sai	Phân loại dựa trên đặc trưng thủ công (tốc độ gói, thời gian luồng…)

Phát hiện dựa trên ngưỡng	Đặt ngưỡng cảnh báo khi lưu lượng vượt mức bình thường, tuy nhiên cần tinh chỉnh để giảm cảnh báo sai	Dễ cấu hình, hiệu quả với các tấn công băng thông cao
Ngăn chặn tấn công	Chặn cổng, giới hạn băng thông, và chuyển hướng lưu lượng khi phát hiện tấn công.	Có thể chặn và ngăn chặn tác động của tấn công ngay khi phát hiện.
Hệ thống cảnh báo sớm	Phân tích dữ liệu trước khi tấn công đạt đỉnh để ngăn chặn thiệt hại.	Phát hiện tấn công từ sớm, ngăn chặn tác động và thời gian phản ứng.

2.4. Những hạn chế trong các nghiên cứu hiện nay

Mặc dù các giải pháp hiện tại đã đạt được nhiều tiến bộ trong việc phát hiện và ngăn chặn tấn công DDoS, vẫn tồn tại những hạn chế cần khắc phục để nâng cao hiệu quả và độ chính xác. Các thách thức này bao gồm khả năng mở rộng, độ trễ xử lý, và tính thích ứng với các kiểu tấn công mới. *Bảng 2.3* dưới đây tóm tắt các khoảng trống chính trong nghiên cứu mạng SDN:

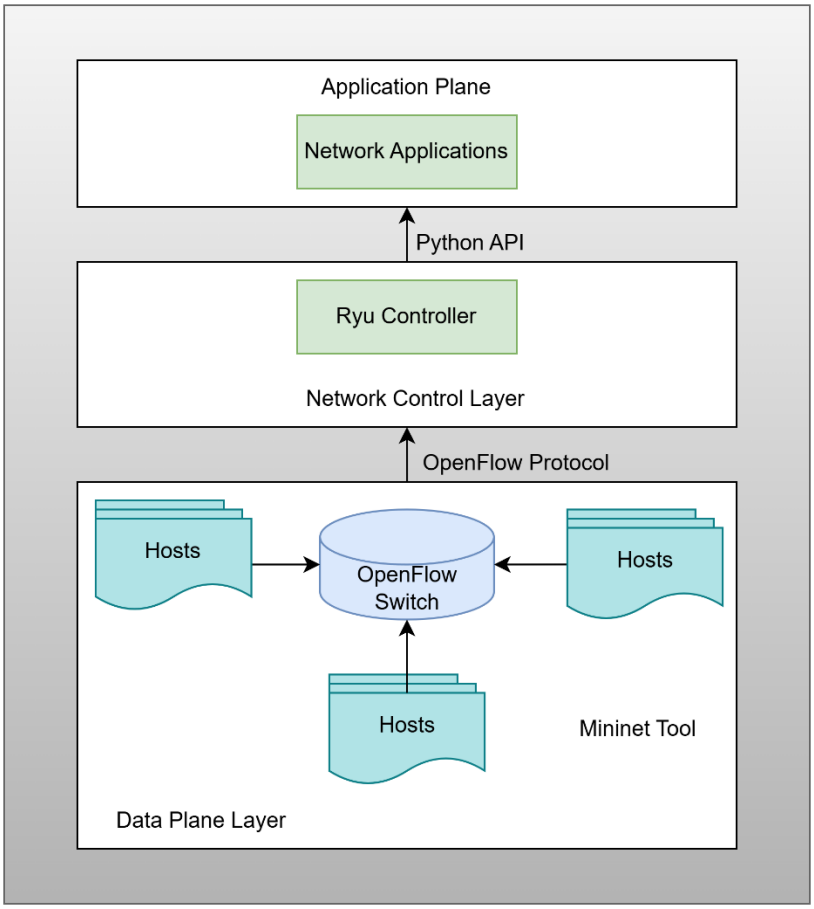
Bảng 2.3 *Bảng tóm tắt những hạn chế trong nghiên cứu*

	Mô tả	Ảnh hưởng
Mở rộng	Khó khăn trong xử lý khi mạng mở rộng nhanh chóng.	Giảm hiệu suất khi số lượng thiết bị và lưu lượng mạng tăng đột biến.
Độ trễ xử lý	Phản hồi chậm có thể làm giảm hiệu quả phòng thủ.	Tạo cơ hội cho kẻ tấn công khai thác điểm yếu.
Độ chính xác	Khó phân biệt giữa lưu lượng bình thường và tấn công, đặc biệt khi kiểu tấn công mới xuất hiện.	Dễ tạo cảnh báo sai (false positives) hoặc bỏ sót tấn công (false negatives).
Khả năng thích ứng	Khó phát hiện các biến thể tấn công mới mà không huấn luyện lại mô hình.	Giảm khả năng phát hiện tấn công trong môi trường thay đổi nhanh.
Bảo mật SDN	Phụ thuộc vào bộ điều khiển trung tâm, tạo điểm yếu tiềm ẩn.	Tăng nguy cơ bị tấn công từ chối dịch vụ và chiếm quyền điều khiển.
Khả năng tích hợp	Cần tích hợp tốt với các hệ thống bảo mật khác để tăng cường hiệu quả.	Giảm khả năng phối hợp giữa các lớp bảo mật, làm giảm hiệu quả bảo vệ.

Chương 3: Phương pháp nghiên cứu

3.1. Kiến trúc hệ thống

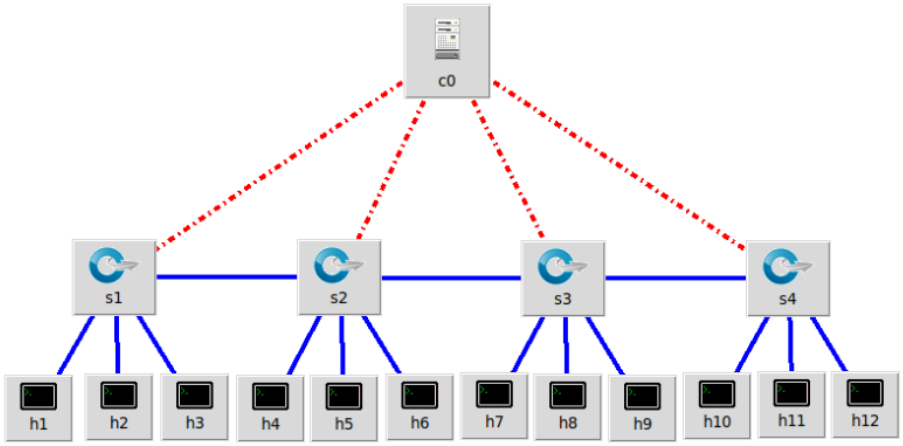
Hình 3.1 thể hiện kiến trúc của hệ thống SDN, trong đó mỗi node/host được tạo ảo bằng Mininet và được kết nối với OpenFlow switch. Switch này định nghĩa các giao thức SDN và OpenFlow, giúp giao tiếp với mặt phẳng điều khiển của SDN. Mặt phẳng điều khiển này kiểm soát mặt phẳng dữ liệu và các switch, thiết lập quy tắc và giám sát luồng dữ liệu mạng. Trong cấu trúc này, Ryu được sử dụng như một controller, mang lại khả năng lập trình và quản lý các hoạt động định tuyến trong mạng.



Hình 3.1 *Sơ đồ kiến trúc của hệ thống*

Cấu trúc hệ thống bao gồm:

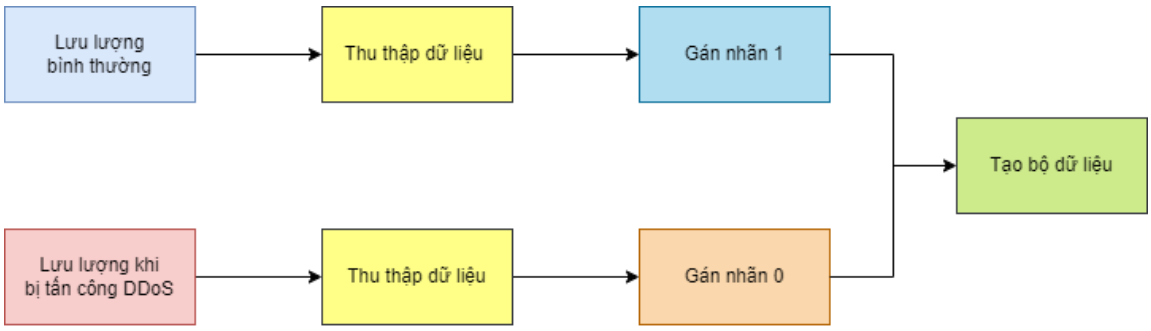
- **Mặt phẳng điều khiển:** Sử dụng Mininet làm môi trường giả lập mạng. Tạo một mạng ảo với cấu trúc mạng tuyến tính (*Hình 3.2*) với 4 switch hỗ trợ với Open vSwitch. Số host sử dụng là 12 host. Mặt phẳng điều khiển giao tiếp với mặt phẳng điều khiển bằng giảo thức OpenFlow.
- **Mặt phẳng điều khiển**: Sử dụng Ryu Controller để điều khiển mạng SDN, Ryu Controller là một bộ điều khiển mạng phần mềm mã nguồn mở được xây dựng trên ngôn ngữ Python và hỗ trợ giao thức OpenFlow.



Hình 3.2 *Sơ đồ cấu trúc mạng*

3.2. Thu thập dữ liệu

Dữ liệu được thu thập từ hai loại lưu lượng chính: bình thường và khi bị tấn công DDoS. Quá trình này bao gồm cả lưu lượng hợp lệ (HTTP, TCP, UDP từ các dịch vụ thông thường) và lưu lượng độc hại (ICMP flood, UDP flood, TCP SYN Flood) mô phỏng tấn công DDoS. Sau khi thu thập, nhãn 1 sẽ được gán cho lưu lượng bình thường, nhãn 0 sẽ được gán cho lưu lượng bị tấn công.



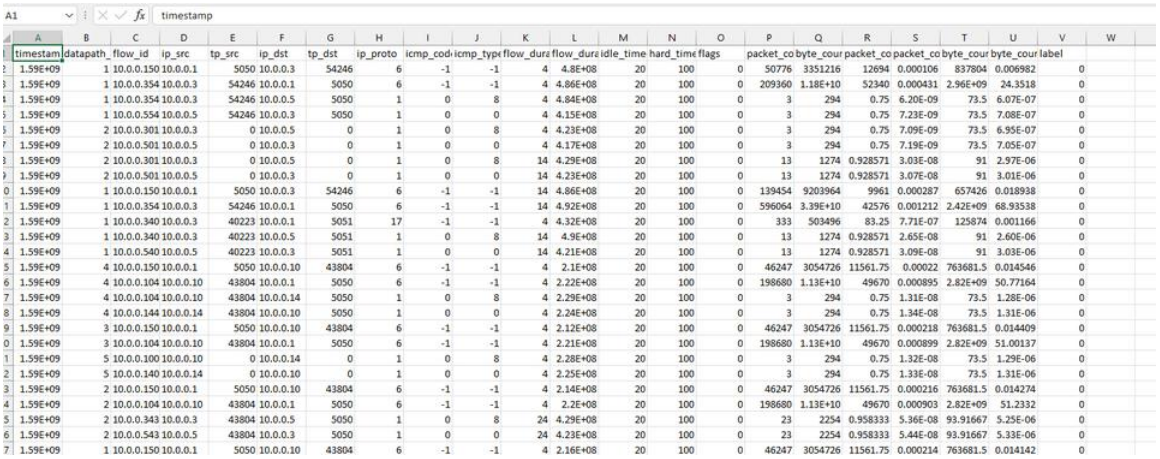
Hình 3.3: Sơ đồ quá trình thu thập dữ liệu

Bộ dữ liệu thu thập được có kích thước 22 cột x 2667523 hàng và chi tiết các đặc trưng được mô tả trong *Bảng 3.1* và *Hình 3.4.*

Bảng 3.1 *Bảng các đặc trưng trong bộ dữ liệu*

Đặc trưng	Mô tả
timestamp	Thời gian thu thập (thời gian Unix)
datapath_id	ID của OpenFlow switch
flow_id	ID tính theo IP nguồn/đích, cổng và giao thức
ip_src	Địa chỉ IP nguồn
tp_src	Cổng nguồn (TCP/UDP)
ip_dst	Địa chỉ IP đích
tp_dst	Cổng đích (TCP/UDP)
ip_proto	Giao thức IP (Ví dụ: 1=ICMP, 6=TCP, 17=UDP…)
icmp_code	Mã ICMP (nếu ip_proto = ICMP)

icmp_type	Loại ICMP (nếu ip_proto = ICMP)
flow_duration_sec	Thời gian tồn tại luồng (giây)
flow_duration_nsec	Thời gian tồn tại luồng (nano giây)
idle_timeout	Thời gian tối đa luồng không hoạt động (idle) trước khi xoá
hard_timeout	Tuổi thọ tối đa của luồng
flags	Các cờ liên quan tới luồng
packet_count	Số gói trong luồng
byte_count	Số byte trong luồng
packet_count_per_second	Tốc độ gói (gói/giây)
packet_count_per_nsec	Tốc độ gói (gói/nano-giây)
byte_count_per_second	Tốc độ dữ liệu (byte/giây)
byte_count_per_nsec	Tốc độ dữ liệu (byte/nano-giây)
label	Nhãn dùng huấn luyện mô hình



timestamp	datapath_	flow_id	ip_src	tp_src	ip_dst	tp_dst	ip_proto	icmp_cod	icmp_type	flow_dura	flow_dura	idle_time	hard_time	flags	packet_co	byte_cour	packet_co	packet_co	byte_cour	byte_cour	label
1.59E+09	1	10.0.0.150	10.0.0.1	5050	10.0.0.3	54246	6	-1	-1	4	4.8E+08	20	100	0	50776	3351216	12694	0.000106	837804	0.006982	0
1.59E+09	1	10.0.0.354	10.0.0.3	54246	10.0.0.1	5050	6	-1	-1	4	4.86E+08	20	100	0	209360	1.18E+10	52340	0.000431	2.96E+09	24.3518	0
1.59E+09	1	10.0.0.354	10.0.0.3	54246	10.0.0.5	5050	1	0	8	4	4.84E+08	20	100	0	3	294	0.75	6.20E-09	73.5	6.07E-07	0
1.59E+09	1	10.0.0.554	10.0.0.5	54246	10.0.0.3	5050	1	0	0	4	4.15E+08	20	100	0	3	294	0.75	7.23E-09	73.5	7.08E-07	0
1.59E+09	2	10.0.0.301	10.0.0.3	0	10.0.0.5	0	1	0	8	4	4.23E+08	20	100	0	3	294	0.75	7.09E-09	73.5	6.95E-07	0
1.59E+09	2	10.0.0.501	10.0.0.5	0	10.0.0.3	0	1	0	0	4	4.17E+08	20	100	0	3	294	0.75	7.19E-09	73.5	7.05E-07	0
1.59E+09	2	10.0.0.301	10.0.0.3	0	10.0.0.5	0	1	0	8	14	4.29E+08	20	100	0	13	1274	0.928571	3.03E-08	91	2.97E-06	0
1.59E+09	2	10.0.0.501	10.0.0.5	0	10.0.0.3	0	1	0	0	14	4.23E+08	20	100	0	13	1274	0.928571	3.07E-08	91	3.01E-06	0
1.59E+09	1	10.0.0.150	10.0.0.1	5050	10.0.0.3	54246	6	-1	-1	14	4.86E+08	20	100	0	139454	9203964	9961	0.000287	657426	0.018938	0
1.59E+09	1	10.0.0.354	10.0.0.3	54246	10.0.0.1	5050	6	-1	-1	14	4.92E+08	20	100	0	596064	3.39E+10	42576	0.001212	2.42E+09	68.93538	0
1.59E+09	1	10.0.0.340	10.0.0.3	40223	10.0.0.1	5051	17	-1	-1	4	4.32E+08	20	100	0	333	503496	83.25	7.71E-07	125874	0.001166	0
1.59E+09	1	10.0.0.340	10.0.0.3	40223	10.0.0.5	5051	1	0	8	14	4.9E+08	20	100	0	13	1274	0.928571	2.65E-08	91	2.60E-06	0
1.59E+09	1	10.0.0.540	10.0.0.5	40223	10.0.0.3	5051	1	0	0	14	4.21E+08	20	100	0	13	1274	0.928571	3.09E-08	91	3.03E-06	0
1.59E+09	4	10.0.0.150	10.0.0.1	5050	10.0.0.10	43804	6	-1	-1	4	2.1E+08	20	100	0	46247	3054726	11561.75	0.00022	763681.5	0.014546	0
1.59E+09	4	10.0.0.104	10.0.0.10	43804	10.0.0.1	5050	6	-1	-1	4	2.22E+08	20	100	0	198680	1.13E+10	49670	0.000895	2.82E+09	50.77164	0
1.59E+09	4	10.0.0.104	10.0.0.10	43804	10.0.0.14	5050	1	0	8	4	2.29E+08	20	100	0	3	294	0.75	1.31E-08	73.5	1.28E-06	0
1.59E+09	4	10.0.0.144	10.0.0.14	43804	10.0.0.10	5050	1	0	0	4	2.24E+08	20	100	0	3	294	0.75	1.34E-08	73.5	1.31E-06	0
1.59E+09	3	10.0.0.150	10.0.0.1	5050	10.0.0.10	43804	6	-1	-1	4	2.12E+08	20	100	0	46247	3054726	11561.75	0.000218	763681.5	0.014409	0
1.59E+09	3	10.0.0.104	10.0.0.10	43804	10.0.0.1	5050	6	-1	-1	4	2.21E+08	20	100	0	198680	1.13E+10	49670	0.000899	2.82E+09	51.00137	0
1.59E+09	5	10.0.0.100	10.0.0.10	0	10.0.0.14	0	1	0	8	4	2.28E+08	20	100	0	3	294	0.75	1.32E-08	73.5	1.29E-06	0
1.59E+09	5	10.0.0.140	10.0.0.14	0	10.0.0.10	0	1	0	0	4	2.25E+08	20	100	0	3	294	0.75	1.33E-08	73.5	1.31E-06	0
1.59E+09	2	10.0.0.150	10.0.0.1	5050	10.0.0.10	43804	6	-1	-1	4	2.14E+08	20	100	0	46247	3054726	11561.75	0.000216	763681.5	0.014274	0
1.59E+09	2	10.0.0.104	10.0.0.10	43804	10.0.0.1	5050	6	-1	-1	4	2.2E+08	20	100	0	198680	1.13E+10	49670	0.000903	2.82E+09	51.2332	0
1.59E+09	2	10.0.0.343	10.0.0.3	43804	10.0.0.5	5050	1	0	8	24	4.29E+08	20	100	0	23	2254	0.958333	5.36E-08	93.91667	5.25E-06	0
1.59E+09	2	10.0.0.543	10.0.0.5	43804	10.0.0.3	5050	1	0	0	24	4.23E+08	20	100	0	23	2254	0.958333	5.44E-08	93.91667	5.33E-06	0
1.59E+09	1	10.0.0.150	10.0.0.1	5050	10.0.0.10	43804	6	-1	-1	4	2.16E+08	20	100	0	46247	3054726	11561.75	0.000214	763681.5	0.014142	0

***Hình 3.4** Bộ dữ liệu thu thập được*

Link dataset: Dataset (Google Driver)

3.3. Trích xuất đặc trưng

Sau khi dữ liệu thô được thu thập, bước trích xuất đặc trưng sẽ loại bỏ các trường không cần thiết và chuẩn hoá các giá trị để chuẩn bị cho mô hình huấn luyện. Cụ thể:

- **Loại bỏ các trường định danh:** *timestamp*, *datapath_id* và *flow_id* được loại bỏ vì chỉ đóng vai trò định danh (metadata), không chứa thông tin phản ánh hành vi của luồng lưu lượng.
- **Mã hóa địa chỉ IP:** *ip_src* và *ip_dst* là địa chỉ IP ở dạng ký tự (chuỗi), cần chuyển đổi thành số nguyên để mô hình học được quan hệ giữa các node mạng. Để làm điều này, dấu "." trong địa chỉ IP được loại bỏ và các trường này được ép kiểu thành int64. Điều này cũng giúp giảm độ phức tạp khi tính toán và tiết kiệm bộ nhớ.
- **Giữ nguyên các trường đặc trưng chính:** Các trường như *tp_src*, *tp_dst*, *ip_proto*, *packet_count*, *byte_count*, *flow_duration_sec*, *flow_duration_nsec* và tốc độ lưu lượng (*byte_count_per_second*, *packet_count_per_second*, v.v.) vẫn được giữ lại, vì chúng phản ánh trực tiếp hành vi lưu lượng và là yếu tố quyết định trong phát hiện DDoS.

Sau bước này, tập đặc trưng sẽ chỉ chứa các trường thực sự có ý nghĩa, giúp mô hình học máy giảm tải tính toán, tránh overfitting và cải thiện độ chính xác khi phân loại lưu lượng.

3.4. Phát hiện DDoS bằng phương pháp học sâu

Để phân biệt luồng bình thường và luồng tấn công, mô hình MLP (Multi-Layer Perceptron) với hai lớp ẩn. Lý do chọn mô hình được chỉ ra ở *Bảng 3.2*.

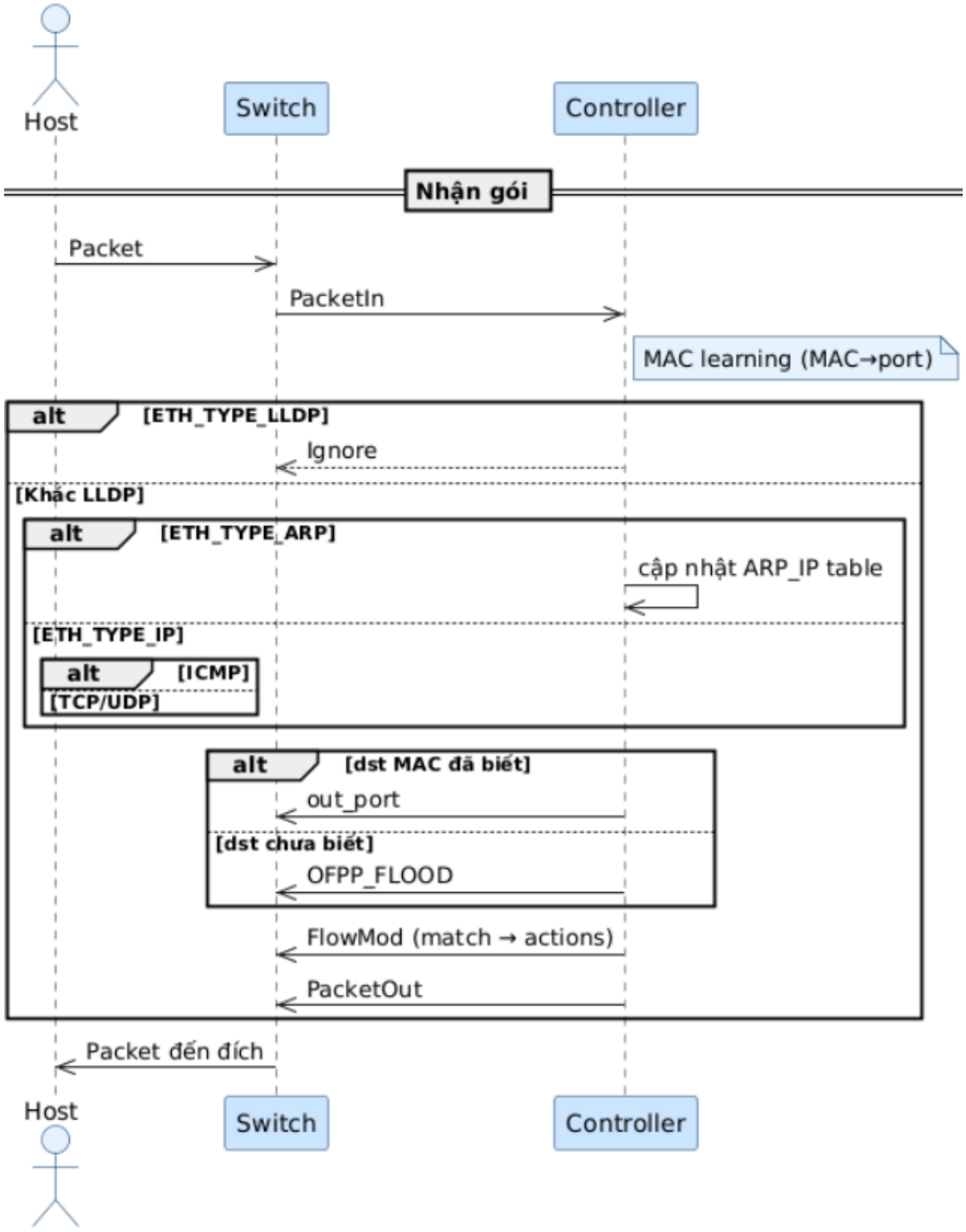
Bảng 3.2 *Bảng các tiêu chí lựa chọn kiến trúc MLP hai lớp ẩn*

Tiêu chí	Đáp ứng
Độ phức tạp vừa phải	Hai lớp ẩn đủ khả năng nắm bắt quan hệ phi tuyến giữa các đặc trưng lưu lượng nhưng không quá sâu để gây chậm trễ.
Thời gian suy luận ngắn	Mô hình chỉ có vài nghìn trọng số nên controller gắn nhãn luồng chỉ trong vài mili-giây, đáp ứng thời gian thực SDN.
Triển khai thuận tiện	Được hỗ trợ sẵn trong thư viện phổ biến (TensorFlow, PyTorch, Scikit-learn,…) và dễ đóng gói để nạp vào Ryu.
Khả năng mở rộng	Khi mạng lớn hơn chỉ cần tăng số nơ-ron hoặc thêm lớp ẩn, không phải thay đổi pipeline đặc trưng.

Với lựa chọn này, phần điều khiển SDN vẫn linh hoạt lập trình, trong khi lớp phát hiện bất thường đủ mạnh để nhận ra các mẫu DDoS đa dạng.

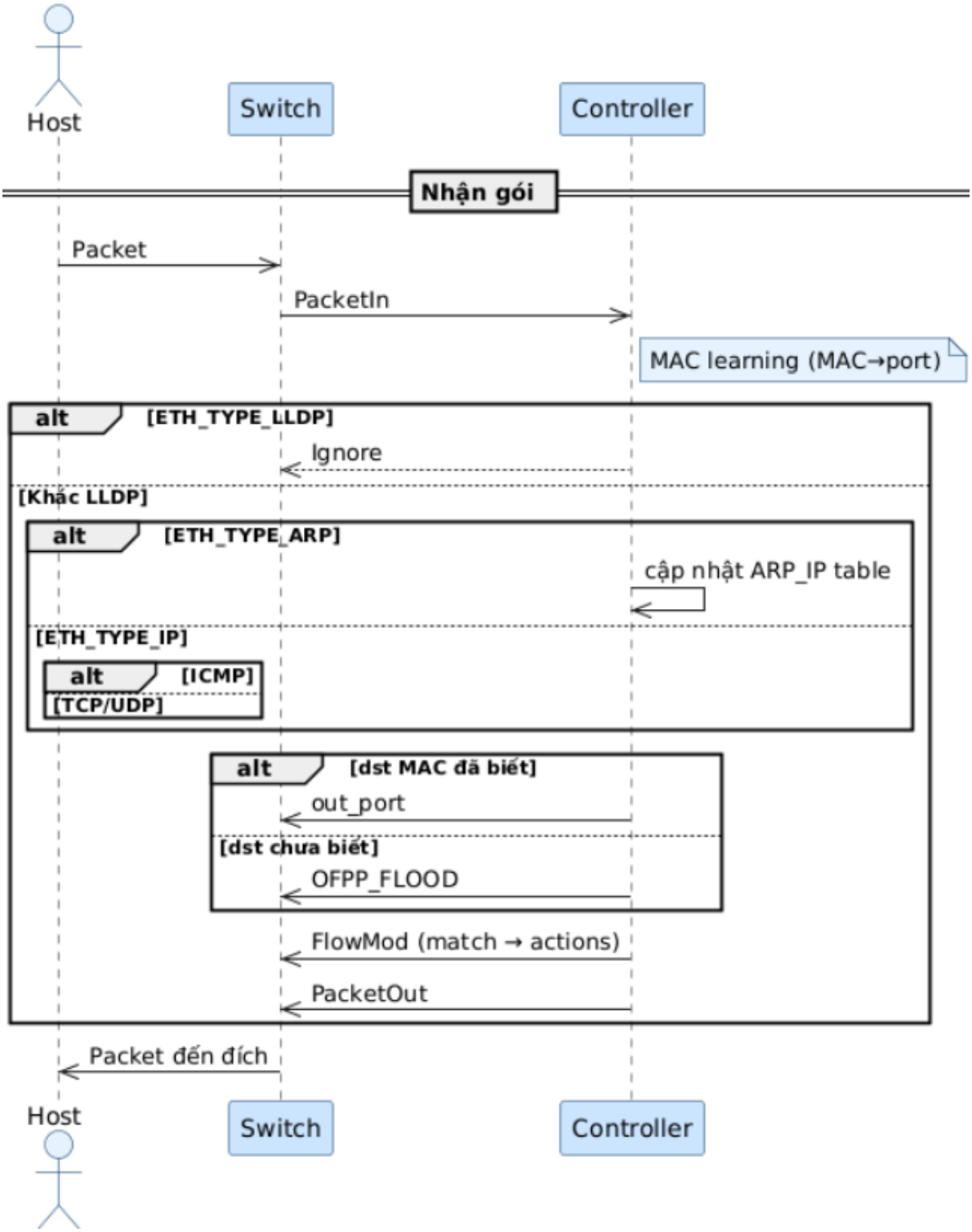
3.5. Ngăn chặn tấn công DDOS

Trong kiến trúc SDN, mọi gói tin chưa khớp bất kỳ flow nào đều được chuyển tới Controller thông qua sự kiện PacketIn. Tận dụng cơ chế này, bộ điều khiển SimpleSwitch13 trước hết học ánh xạ MAC → port và cập nhật bảng ARP (IP ↔ port) bằng cách phân tích các gói ARP hợp lệ. Khi cơ chế phòng thủ (Mitigation) chưa kích hoạt, quá trình xử lý tuân theo *Hình 3.5* sau bước MAC-learning, Controller xác định cổng đích và cài đặt FlowMod forward với độ ưu tiên thấp, rồi trả gói đi qua PacketOut, bảo đảm thông lượng tối đa mà không can thiệp sâu vào datapath.



Hình 3.5 *Luồng chuyển tiếp bình thường trong SDN*

Ngược lại, khi bật Mitigation, Controller so sánh srcIP ở mỗi gói IP với bảng ARP. Nếu cổng gửi gói chứa địa chỉ nguồn không trùng khớp thông tin đã học, cổng đó bị coi là giả mạo (spoofed). Quy trình này được mô tả trong *Hình 3.6*, Controller lập tức sinh FlowMod DROP có độ ưu tiên cao và hard_timeout = 120 s, loại bỏ mọi gói xuất phát từ cổng tấn công ngay tại switch, đồng thời giữ nguyên luồng forward cho lưu lượng hợp lệ.



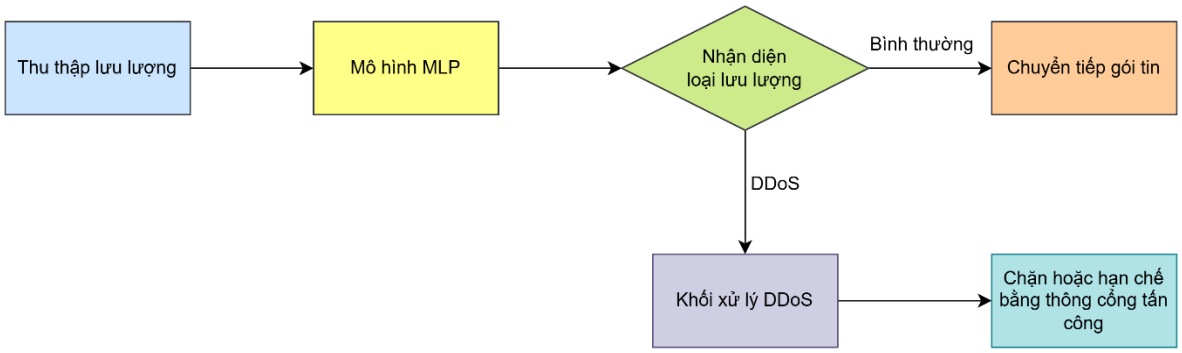
Hình 3.6 *Luồng phát hiện và chặn tấn công DDoS*

Chiến lược "chặn ở mức flow" rút ngắn thời gian phản ứng xuống còn vài mili-giây, cô lập nguồn DDoS tại biên switch, tránh phát sinh packet-loss cho máy bị hại mà không đòi hỏi cấu hình thủ công trên thiết bị vật lý. Hai hình trên do đó làm nổi bật sự khác biệt giữa kênh chuyển tiếp gốc và cơ chế phòng thủ được đề xuất, khẳng định hiệu quả của hướng tiếp cận dựa trên FlowMod trong môi trường SDN.

3.6. Chi tiết triển khai hệ thống

Sau khi mô hình đã được huấn luyện và triển khai trong controller, luồng dữ liệu đi qua mạng SDN được xử lý theo chu trình dưới đây:

- **Thu thập lưu lượng:** Lưu lượng được thu thập tại các switch và gửi về controller dưới dạng bản ghi flow table. Từ mỗi bản ghi, hệ thống trích xuất bộ đặc trưng cần thiết cho khâu phân loại.
- **Phân tích bằng mô hình:** Bộ đặc trưng được nạp vào mô hình MLP hai lớp ẩn - mỗi lớp lần lượt có 128 và 64 nơ-ron, hàm kích hoạt ReLU, tối ưu hoá bằng Adam. MLP học các quan hệ phi tuyến giữa các đặc trưng lưu lượng, từ đó gắn nhãn rõ ràng "bình thường" hoặc "DDoS" cho mỗi luồng.
- **Nhận diện loại lưu lượng:** Nếu lưu lượng được xác định là bình thường, nó sẽ được chuyển tiếp như thông lệ, ngược lại nó sẽ được sang khối xử lý DDoS.
- **Xử lý khi gặp DDoS:** Khối xử lý sẽ cài quy tắc mới trên switch để chặn hoàn toàn hoặc hạn chế băng thông từ cổng tấn công nhờ đó bảo vệ các luồng hợp hợp lệ.



***Hình 3.7** Sơ đồ chu trình xử lý*

Chương 4: Kết quả và thảo luận

4.1. Các chỉ số đánh giá

Để đánh giá hiệu quả của mô hình phát hiện và ngăn chặn tấn công DDoS trong mạng SDN, cần sử dụng các chỉ số đo lường phổ biến như Accuracy, Precision, Recall, F1-Score, và AUC. Mỗi chỉ số phản ánh một khía cạnh khác nhau của hiệu suất mô hình, từ khả năng phân loại chính xác, tỷ lệ phát hiện tấn công, đến độ tin cậy khi dự đoán. *Bảng 4.1* dưới đây tóm tắt ý nghĩa, công thức và đặc điểm của các chỉ số này:

Bảng 4.1 *Bảng các chỉ số đánh giá được sử dụng*

Chỉ số	Ý nghĩa	Công thức
Độ chính xác (Accuracy)	Tỷ lệ mẫu được phân loại đúng trên tổng số mẫu.	$\frac{TP + TN}{TP + TN + FP + FN}$
Độ chính xác (Precision)	Tỷ lệ mẫu được dự đoán là tấn công thực sự là tấn công.	$\frac{TP}{TP + FP}$
Độ nhạy (Recall)	Tỷ lệ mẫu tấn công được phát hiện chính xác.	$\frac{TP}{TP + FN}$
F1-Score	Trung bình điều hòa giữa Precision và Recall, cân bằng giữa bỏ sót và báo động giả.	$\frac{2TP}{2TP + FP + FN}$
AUC(Area Under Curve)	Diện tích dưới đường cong ROC, đo lường khả năng phân biệt giữa lớp tấn công và lớp bình thường.	$\sum_{i=1}^{m} \frac{(FPR_i - FPR_{i-1})(TPR_i - TPR_{i-1})}{2}$

Chú thích:

- TP (True Positive): Số lượng mẫu tấn công được phát hiện đúng.
- TN (True Negative): Số lượng mẫu bình thường được phát hiện đúng.
- FP (False Positive): Số lượng mẫu bình thường bị dự đoán nhầm là tấn công (báo động giả).
- FN (False Negative): Số lượng mẫu tấn công bị bỏ sót, dự đoán nhầm là bình thường
- m: số điểm đánh giá trên ROC (ngưỡng).
- TPR_i (True Positive Rate at threshold i) $= \frac{TP_i}{TP_i + FN_i}$: Tỷ lệ phát hiện đúng tấn công tại ngưỡng i.
- FPR_i (False Positive Rate at threshold i) $= \frac{FP_i}{FP_i + TN_i}$: Tỷ lệ báo động nhầm tại ngưỡng i.

4.2. Triển khai hệ thống

Để đánh giá hiệu quả của hệ thống phát hiện và ngăn chặn tấn công DDoS trong mạng SDN, ta sẽ triển khai nó trong môi trường giả lập với Mininet và Ryu Controller. Mô hình mạng bao gồm các thành phần cơ bản như switch, host, và controller, được cấu hình để tạo ra các lưu lượng tấn công và lưu lượng bình thường. *Bảng 4.2* dưới đây sẽ chỉ ra các công cụ và thư viện để thiết lập hệ thống:

***Bảng 4.2** Bảng các công cụ và thư viện sử dụng*

Thành phần	Công cụ / Thư viện	Vai trò trong hệ thống
Hệ điều hành	Ubuntu 20.04.4 LTS	Máy chủ chạy cả Mininet và Ryu
Ngôn ngữ	Python 3.8.10	Viết controller Ryu, thu thập thống kê, xử lý ML
Mô phỏng mạng	Mininet 2.3.1	Tạo topo ảo, hỗ trợ OpenFlow 1.3
Switch ảo	Open vSwitch 2.13.8	Thực thi quy tắc data-plane và giao tiếp controller
Bộ điều khiển	Ryu 4.34	Điều phối mạng, gọi mô-đun ML để gắn nhãn luồng
Mô hình phân loại	scikit-learn 1.3.2	Cung cấp các mô hình phân loại và công cụ tiền xử lý
	pandas 1.1.5 numpy 1.24.4	Xử lý, chuẩn hoá và chuyển đổi đặc trưng
	joblib 1.4.2	Lưu/đọc mô hình đã huấn luyện (.pkl)

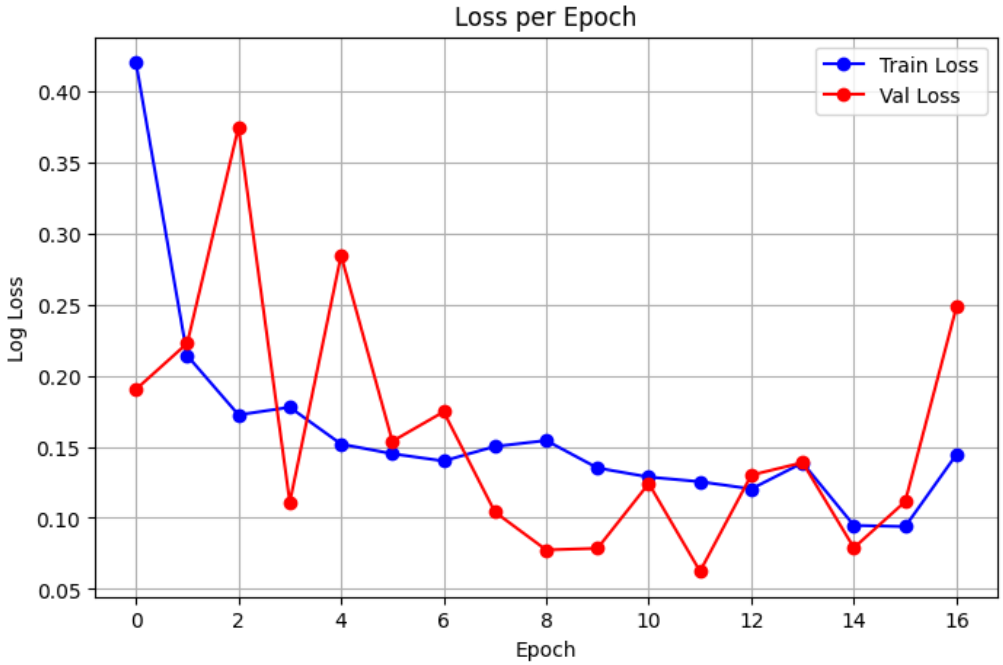
Dataset với tỷ lệ 80 % train – 20 % test (*Bảng 4.3*), bộ dữ liệu đáp ứng đồng thời ba yêu cầu quan trọng:

- Độ phủ huấn luyện cao hơn 2,13 triệu bản ghi được giữ lại, giúp mô hình khai thác đầy đủ sự đa dạng của cả lưu lượng tấn công DDoS và lưu lượng bình thường.
- Tập kiểm tra đại diện khoảng 0,53 triệu bản ghi, duy trì đúng tỷ lệ giữa Class 0 và Class 1, nhờ đó đánh giá khách quan mà không phát sinh mất cân bằng (imbalance).
- Tuân thủ chuẩn 80/20 tỷ lệ này phổ biến trong cộng đồng học máy, cho phép so sánh dễ dàng với các nghiên cứu liên quan và kiểm chứng khả năng tổng quát hóa của mô hình trên dữ liệu chưa từng quan sát.

***Bảng 4.3** Chia bộ dữ liệu thành tập train và test*

	Class 0 (DDoS)	Class 1 (Normal)	Tổng
Tập train	725 482	1 408 536	2 134 018
Tập test	181 371	352 134	533 505

Mô hình MLP được huấn luyện trên tập dữ liệu này bao gồm cả lưu lượng bình thường và lưu lượng DDoS, với cấu trúc hai lớp ẩn (128 và 64 nơ-ron), hàm kích hoạt ReLU và thuật toán tối ưu Adam. Quá trình huấn luyện sử dụng binary cross-entropy làm hàm mất mát. *Hình 4.1* dưới đây mô tả sự thay đổi của Train Loss và Validation Loss qua từng epoch.



Hình 4.1 *Đồ thị sự thay đổi giá trị của hàm mất mát trong quá trình huấn luyện*

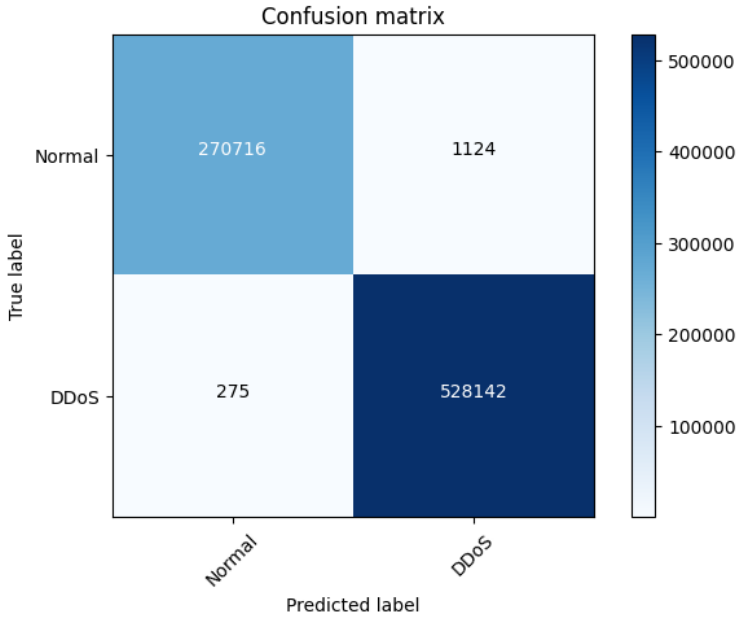
4.3. Kết quả huấn luyện mô hình

a. Mô hình MLP

Sau khi hoàn thành việc xây dựng mô hình phát hiện tấn công DDoS trong môi trường mạng định nghĩa bằng phần mềm (SDN) bằng cách áp dụng kiến trúc học sâu MLP (Multi-Layer Perceptron), ta tiến hành huấn luyện mô hình trên tập dữ liệu đã được thu thập trước đó. Quá trình đánh giá hiệu quả của mô hình được thực hiện thông qua ma trận nhầm lẫn (Confusion Matrix), đồng thời sử dụng các chỉ số đánh giá phổ biến như Accuracy, Precision, Recall và F1-Score để đo lường hiệu suất phát hiện tấn công của hệ thống một cách toàn diện.

Ma trận nhầm lẫn cung cấp cái nhìn chi tiết về số lượng dự đoán đúng và sai, giúp phân tích lỗi cụ thể theo bốn loại: True Positive (TP), True Negative (TN), False Positive (FP) và False Negative (FN).

Kết quả *Hình 4.2* mô tả kết quả ma trận nhầm lẫn của mô hình MLP sau khi huấn luyện xong:



Hình 4.2 *Kết quả ma trận nhầm lẫn của mô hình MLP*

Và kết quả chỉ số đánh giá mô hình như Accuracy, Precision, Recall, F1-Score của mô hình MLP được trình bày ở *Bảng 4.4* bên dưới.

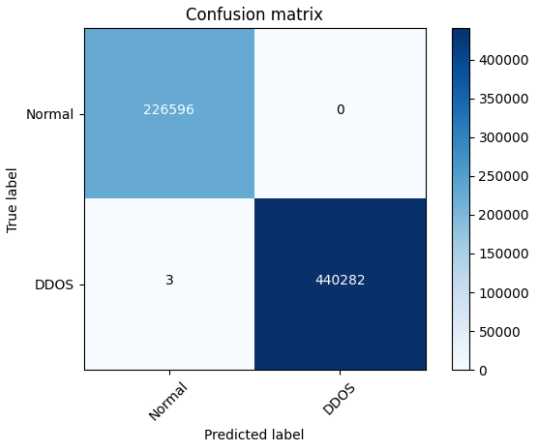
Bảng 4.4 *Kết quả chỉ số đánh giá của mô hình MLP*

Accuracy	Precision	Recall	F1-Score
99.83%	99.79%	99.95%	99.87%

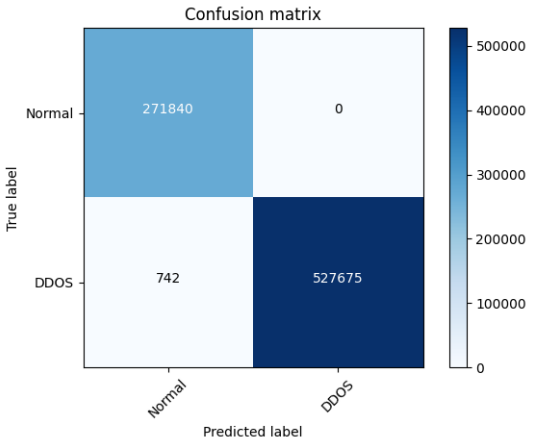
b. Các mô hình học máy

Nhằm kiểm chứng mức độ phù hợp của MLP đối với bài toán phát hiện và ngăn chặn tấn công DDoS, nhóm tiếp tục huấn luyện sáu mô hình học máy kinh điển: Decision Tree, KNN, SVM, Random Forest, Naïve Bayes và Logistic Regression. Sau khi đào tạo, mỗi mô hình được đánh giá bằng ma trận nhầm lẫn. Kết quả cho phép xác định mô hình học máy đạt hiệu năng cao nhất, làm cơ sở so sánh trực tiếp với mô hình học sâu (MLP) đã trình bày ở phần trên.

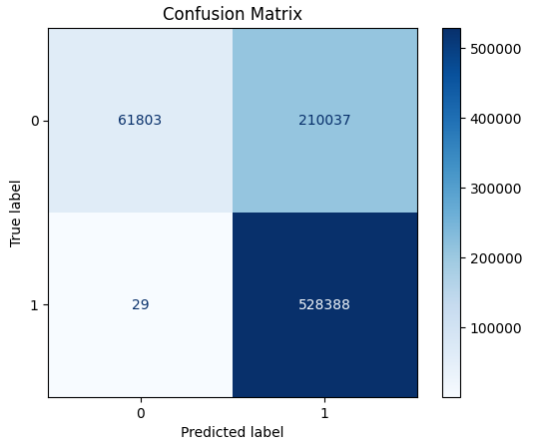
Kết quả ma trận nhầm lẫn của các mô hình học máy Decision Tree, KNN, SVM, RandomForest, Navie Bayes, Logistic Regression sẽ được mô tả lần lượt ở các hình *Hình 4.3, Hình 4.4, Hình 4.5, Hình 4.6, Hình 4.7 và Hình 4.8.*



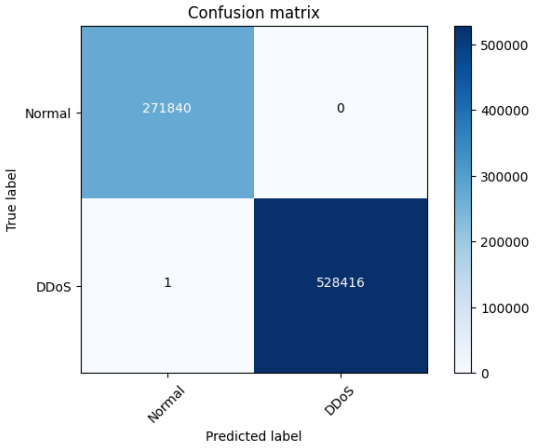
Hình 4.3 *Kết quả ma trận nhầm lẫn của mô hình Decision Tree*



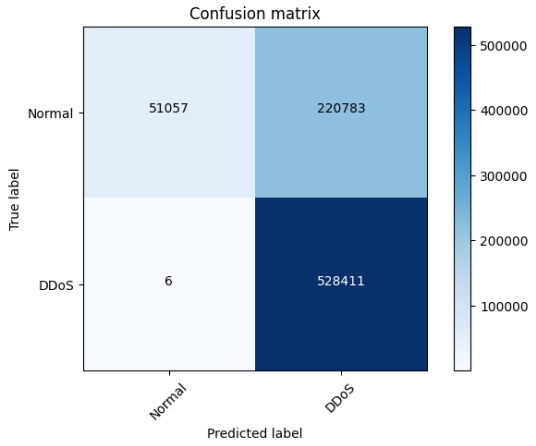
Hình 4.4 *Kết quả ma trận nhầm lẫn của mô hình KNN*



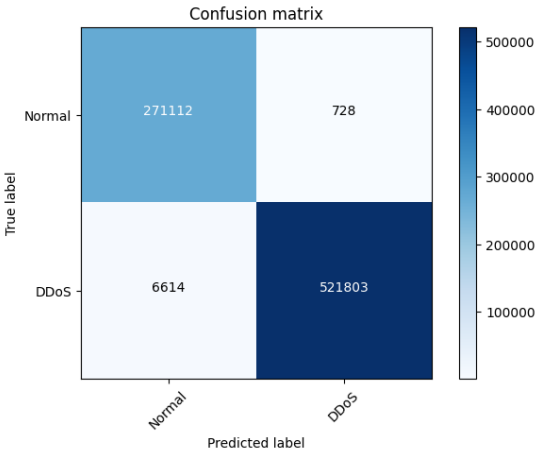
Hình 4.5 *Kết quả ma trận nhầm lẫn của mô hình SVM*



Hình 4.6 *Kết quả ma trận nhầm lẫn của mô hình Random Forest*



Hình 4.7 *Kết quả ma trận nhầm lẫn của mô hình Navie Bayes*



Hình 4.8 *Kết quả ma trận nhầm lẫn của mô hình Logistic Regression*

Dựa vào kết quả ma trận nhầm lẫn trên, ta có *Bảng 4.5* tổng hợp kết quả từ các ma trận nhầm lẫn Confusion Matrix của các mô hình trên để so sánh hiệu suất giữa các mô hình học máy trên.

***Bảng 4.5** Bảng tổng hợp kết quả từ ma trận nhầm lẫn của các mô hình học máy*

Thuật toán	TP	FP	TN	FN
Decision Tree	440282	0	226596	3
KNN	527675	0	271840	742
SVM	528388	210037	61803	29
Random Forest	528416	0	271840	1
Navie Bayes	528411	220783	51057	6
Logistic Regression	521803	728	271112	6614

Ngoài dựa vào ma trận nhầm lẫn, các tiêu chí đánh giá khác như là Accuracy, Precision, Recall và F1-Score cũng được sử dụng để đo lường hiệu suất của các mô hình, cụ thể ở *Bảng 4.6*.

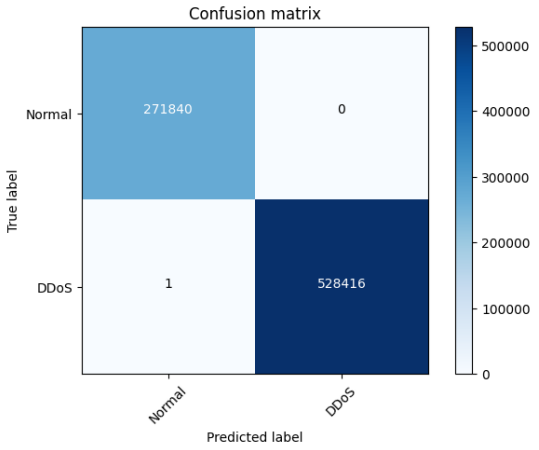
***Bảng 4.6** Bảng tổng hợp kết quả huấn luyện của các mô hình học máy*

Thuật toán	Accuracy	Precision	F1-Score	Recall
Decision Tree	99.99%	100%	99.99%	99.99%
KNN	99.91%	99.86%	99.99%	99.93%
SVM	73.73%	71.56%	83.42%	99.99%
Random Forest	99.99%	100%	99.99%	99.99%
Navie Bayes	72.41%	70.53%	82.72%	100%
Logistic Regression	99.08%	99.86%	99.3%	98.75%

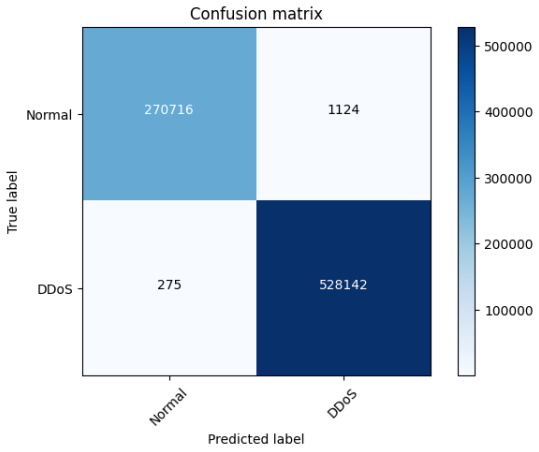
Dựa vào kết quả tổng hợp ở *Bảng 4.5, Bảng 4.6* ta có thể đưa ra kết luận mô hình có kết quả tốt nhất trong tất cả các mô hình học máy trên là Random Forest và mô hình này sẽ được chọn để so sánh với MLP.

c. So sánh và đánh giá hiệu suất của các mô hình

Tiếp theo, nhóm tiến hành đối chiếu hiệu năng giữa hai mô hình đã xây dựng MLP (học sâu) và Random Forest (học máy). *Hình 4.9* và *Hình 4.10* thể hiện ma trận nhầm lẫn của Random Forest và MLP.



Hình 4.9 *Kết quả ma trận nhầm lẫn của mô hình Random Forest*



Hình 4.10 *Kết quả ma trận nhầm lẫn của mô hình MLP*

Kết quả của hai ma trận nhầm lẫn trên được tổng hợp ở *Bảng 4.7* bên dưới:

Bảng 4.7 *Bảng tổng hợp kết quả từ ma trận nhầm lẫn của hai mô hình*

Thuật toán	TP	FP	TN	FN
MLP	528142	1124	270716	275
Random Forest	528416	0	271840	1

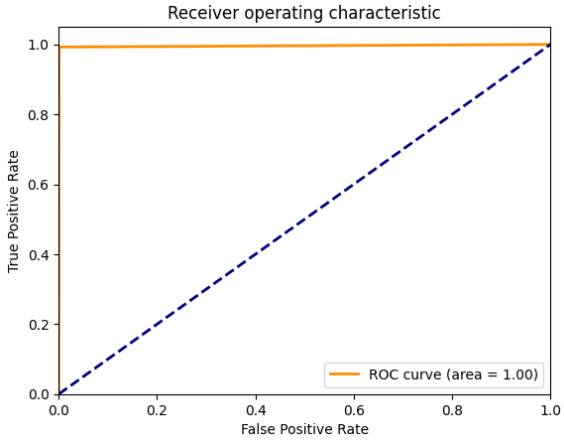
Bảng 4.8 bên dưới tổng hợp kết quả các chỉ số đánh giá như Accuracy, Precision, Recall, F1-Score và AUC, giúp đánh giá ưu nhược điểm của từng mô hình.

Bảng 4.8 *Bảng các chỉ số đánh giá của hai mô hình*

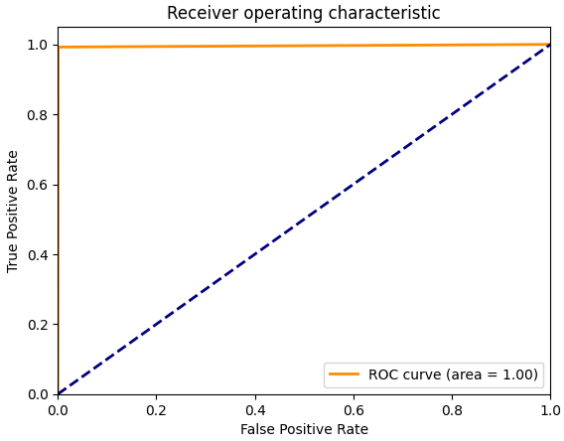
Chỉ số	MLP (Deep Learning)	Random Forest (Machine Learning)
Độ chính xác (Accuracy)	99.83%	99.99%
Độ chính xác (Precision)	99.79%.	100%
Độ nhạy (Recall)	99.95%	99.99%
F1-Score	99.87%	99.99%
AUC (Area Under Curve)	99.77%	100%

Để đánh giá chính xác hơn hiệu quả của các mô hình, đường cong ROC (Receiver Operating Characteristic) cũng được sử dụng.

Đường cong ROC biểu diễn mối quan hệ giữa tỷ lệ phát hiện (TPR) và tỷ lệ báo động giả (FPR) ở các ngưỡng khác nhau, cho thấy khả năng phân biệt giữa lớp tấn công và lớp bình thường của mô hình. AUC (Area Under Curve) đo lường diện tích dưới đường ROC, giá trị càng gần 1 càng tốt.



Hình 4.11 *Đường cong ROC của MLP*



Hình 4.12 *Đường cong ROC của Random Forest*

Thông qua các tiêu chí so sánh trên ta có thể rút ra kết luận rằng mặc dù Random Forest thường nhanh hơn và dễ giải thích hơn, MLP có khả năng học và phân loại các mẫu phức tạp hơn nhờ cấu trúc nhiều lớp, phù hợp với các bài toán phức tạp như phát hiện tấn công DDoS.

4.4. Kết quả triển khai hệ thống

Để đánh giá hiệu quả của hệ thống trong môi trường SDN, nhóm tiến hành triển khai và thử nghiệm hệ thống theo ba kịch bản khác nhau, lần lượt là:

- Kịch bản 1: Phát hiện nhưng không can thiệp
- Kịch bản 2: Phát hiện và chặn cổng tấn công
- Kịch bản 3: Phát hiện và giới hạn băng thông

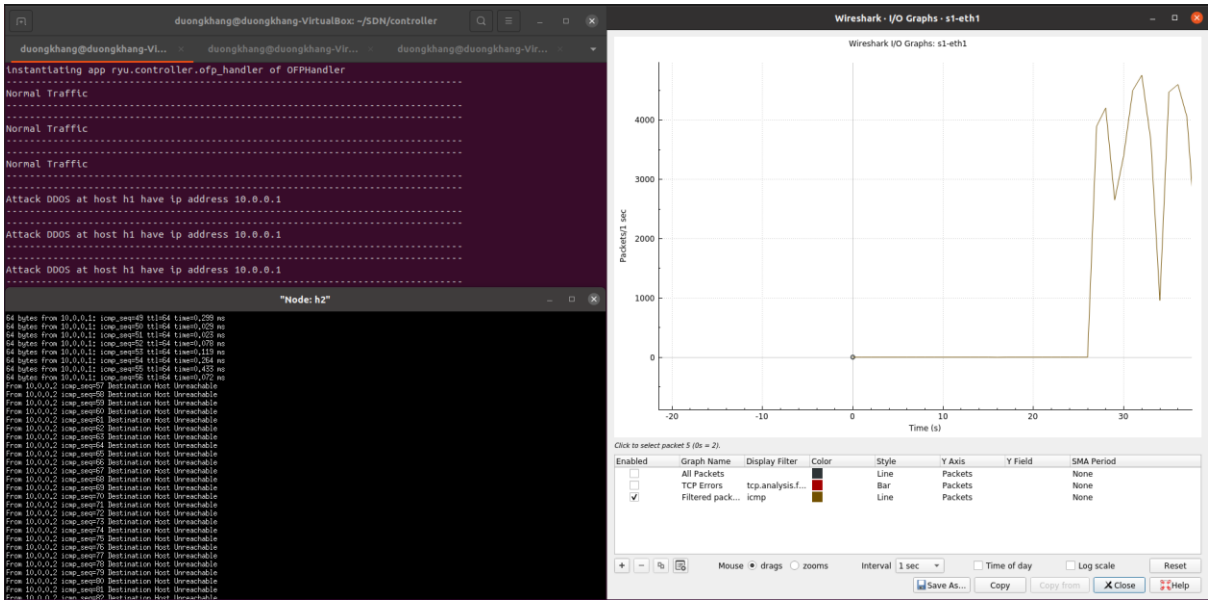
Các kịch bản đều được thực hiện trên cùng một kiến trúc mạng SDN giả lập bao gồm 12 host và 4 switch, kết nối tuyến tính. Mục tiêu là mô phỏng một môi trường mạng với cả lưu lượng hợp lệ và lưu lượng tấn công, cụ thể như sau:

- h2 ping đến h1 liên tục, tạo ra lưu lượng bình thường
- h3 thực hiện tấn công DDoS vào h1 bằng cách sinh ra các dạng lưu lượng bất thường với tần suất cao
- Controller thu thập đặc trưng lưu lượng từ switch, phân tích và gắn nhãn bằng mô hình MLP được huấn luyện sẵn
- Các hành động sau khi phát hiện (nếu có) sẽ khác nhau tùy theo từng kịch bản

a. Kịch bản 1: Phát hiện nhưng không can thiệp

Sau khi controller nhận được bản ghi flow từ switch, hệ thống sử dụng mô hình MLP để phân loại luồng. Không có hành động xử lý nào được thực hiện, tất cả lưu lượng (bao gồm cả tấn công) vẫn được chuyển tiếp bình thường.

Mục tiêu: Đánh giá khả năng phát hiện DDoS của mô hình khi không có sự can thiệp vào dữ liệu mạng.



Hình 4.13 *Kết quả phát hiện nhưng không can thiệp*

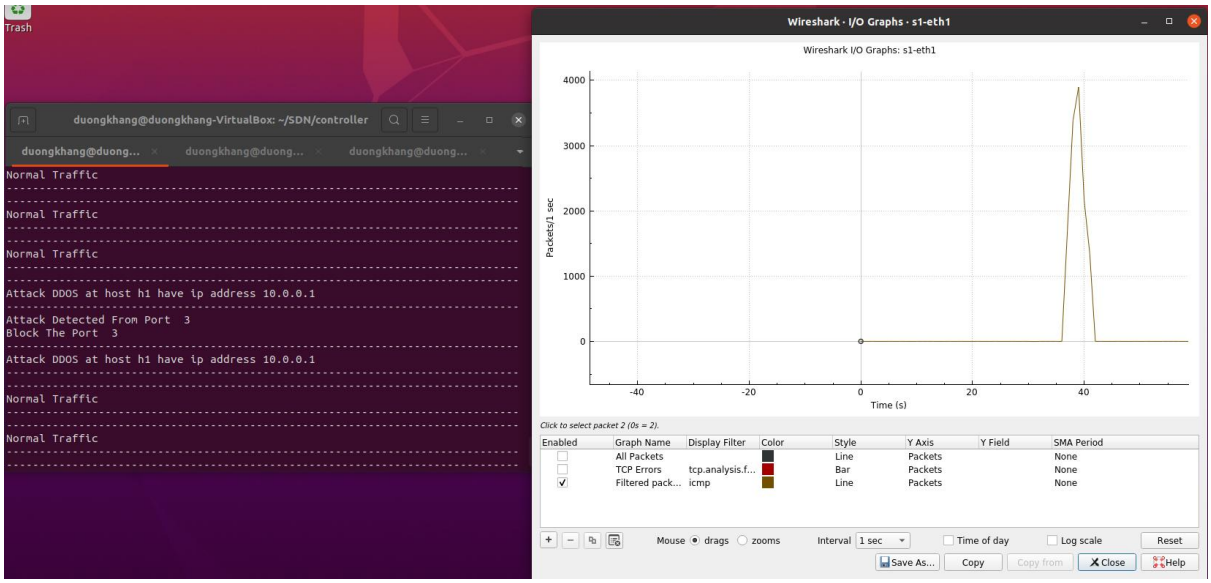
Kết quả (thể hiện ở *Hình 4.13*):

- Hệ thống đã ghi nhận chính xác sự khác biệt giữa lưu lượng bình thường và lưu lượng tấn công. Dòng log *"Normal Traffic"* phản ánh đúng hoạt động ping của host h2 đến h1.
- Các dòng *"Attack DDOS at host h1 have ip address 10.0.0.1"* được ghi liên tục khi host h3 phát sinh tấn công DDoS, cho thấy controller đã phát hiện đúng các luồng tấn công đến đích là h1.
- Do không có cơ chế chặn, lưu lượng DDoS tiếp tục truyền đến h1. Tại cửa sổ của h2, lệnh ping h1 bắt đầu xuất hiện lỗi *"Destination Host Unreachable"* cho thấy h1 bị nghẽn mạng và mất kết nối, không còn phản hồi nữa.
- Biểu đồ Wireshark thể hiện lưu lượng tăng đột ngột về số lượng gói tin trên giây (hơn 4000 packets/sec), cho thấy cường độ tấn công mạnh gây tắc nghẽn hoàn toàn mạng đến h1.

b. Kịch bản 2: Phát hiện và chặn cổng tấn công

Sau khi phân loại luồng, nếu luồng được xác định là tấn công, controller sẽ cài đặt một rule dạng drop lên switch để chặn toàn bộ lưu lượng đến từ cổng tấn công. Các luồng bình thường vẫn được phép truyền qua như thông thường.

Mục tiêu: Đánh giá khả năng phát hiện tấn công DDoS và thực hiện hành động ngăn chặn ngay tại switch nhằm bảo vệ host đích.



Hình 4.14 *Kết quả phát hiện và chặn cổng tấn công*

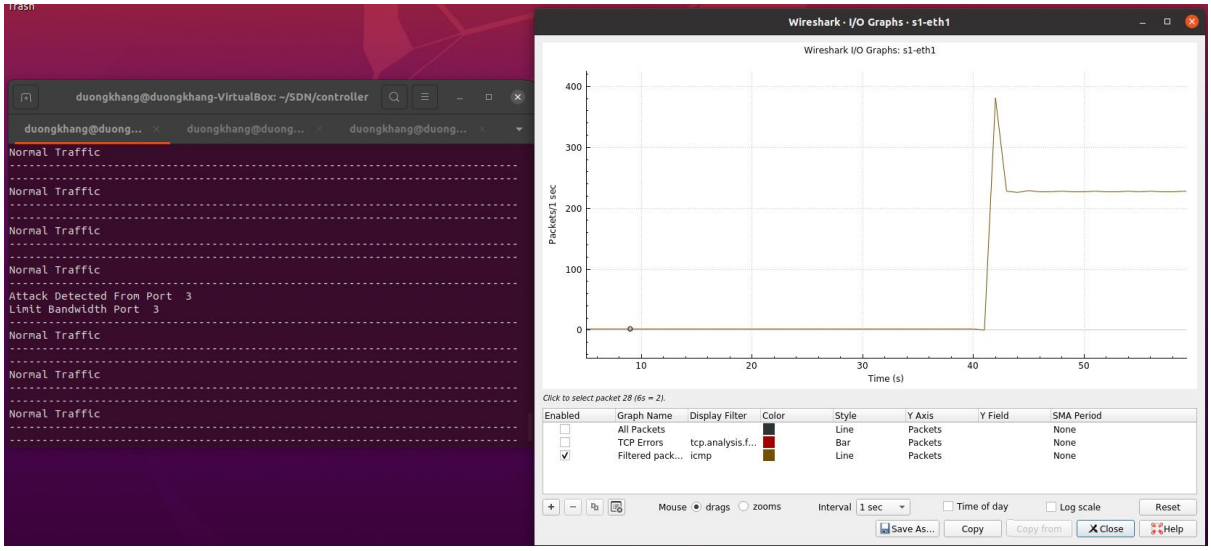
Kết quả (thể hiện ở *Hình 4.14*):

- Hệ thống đã ghi nhận chính xác lưu lượng tấn công từ host h3 đến h1. Dòng log *"Attack Detected From Port 3"* cho thấy controller đã phát hiện đúng nguồn phát sinh lưu lượng tấn công.
- Rule *"Block The Port 3"* được áp dụng ngay sau đó, khiến cho toàn bộ lưu lượng từ h3 bị chặn tại switch, không thể tiếp cận đến h1.
- Biểu đồ Wireshark thể hiện rõ sự thay đổi lưu lượng: khi bị tấn công, số lượng gói tăng vọt; sau khi bị chặn, đường biểu diễn giảm nhanh chóng, chứng minh việc ngăn chặn đã có hiệu lực.
- Dòng *"Normal Traffic"* tiếp tục xuất hiện sau đó, phản ánh lưu lượng hợp lệ từ h2 vẫn được hệ thống xử lý bình thường.

c. Kịch bản 3: Phát hiện và giới hạn băng thông

Sau khi phân loại luồng, nếu một luồng được phát hiện là tấn công, controller sẽ cài đặt rule giới hạn băng thông đối với cổng tấn công. Trong kịch bản này, giới hạn được đặt là 256 Kbps, đủ để làm suy yếu hiệu quả của cuộc tấn công nhưng không cắt đứt hoàn toàn kết nối.

Mục tiêu: Đánh giá khả năng phát hiện tấn công và phản ứng bằng cách hạn chế tốc độ truyền thay vì chặn hoàn toàn, nhằm duy trì tính sẵn sàng của mạng.



Hình 4.15 *Kết quả phát hiện và giới hạn băng thông*

Kết quả (thể hiện ở *Hình 4.15*):

- Hệ thống đã phát hiện đúng lưu lượng tấn công từ host h3, thể hiện qua log *"Attack Detected From Port 3"*.
- Ngay sau đó, dòng *"Limit Bandwidth Port 3"* được ghi ra, xác nhận rằng controller đã cấu hình rule hạn chế tốc độ lưu lượng đến từ cổng của host tấn công.
- Biểu đồ Wireshark cho thấy lưu lượng tăng đột ngột khi tấn công bắt đầu; sau khi hạn chế băng thông được áp dụng, đồ thị cho thấy lưu lượng giảm mạnh và ổn định ở mức ổn định, không còn tăng đột biến, chứng tỏ tấn công đã được kiểm soát.
- Dòng *"Normal Traffic"* xuất hiện đều đặn sau đó, cho thấy hệ thống tiếp tục xử lý các luồng hợp lệ một cách ổn định trong khi đã kiểm soát được mức độ tấn công từ h3.

4.5. Nhận xét

a. Nhận xét kết quả huấn luyện mô hình

Sau khi tiến hành huấn luyện và kiểm tra mô hình phát hiện tấn công DDoS, các kết quả thu được cho thấy sự khác biệt rõ rệt giữa mô hình MLP và Random Forest:

- **MLP**: Mô hình này đạt Accuracy = 99.83%, Precision = 99.79%, Recall = 99.95%, F1-Score = 99.87% và AUC = 99.77%, cho thấy khả năng nhận diện tốt cả lưu lượng bình thường và tấn công. Tuy nhiên, ma trận nhầm lẫn (Hình 4.10) cho thấy vẫn tồn tại False Positives (FP = 1124) và False Negatives (FN = 275), chứng tỏ mô hình đôi khi gặp khó khăn với các mẫu gần ranh giới quyết định. Dù vậy, đường cong ROC (Hình 4.11) của MLP gần đạt mức lý tưởng (AUC ≈ 1.00), khẳng định khả năng phân biệt tốt giữa hai lớp. Nhìn chung, MLP phù hợp với các tình huống có đặc trưng phi tuyến phức tạp, nhưng có thể cần thêm điều chỉnh để giảm FN và FP.
- **Random Forest**: Với Accuracy = 99.99%, Precision = 100%, Recall = 99.99%, F1-Score = 99.99% và AUC = 100%, mô hình Random Forest đạt độ chính xác tuyệt đối trên tập kiểm tra. Ma trận nhầm lẫn (Hình 4.9) cho thấy không có FP và chỉ 3 FN, thể hiện mô hình phân loại hoàn hảo trong ngữ cảnh này. Đường cong ROC (Hình 4.12) đạt mức tối đa, chứng tỏ mô hình có khả năng phân biệt hoàn hảo giữa lưu lượng bình

thường và tấn công. Tuy nhiên, hiệu suất tuyệt đối này có thể là dấu hiệu của overfitting, khi mô hình học quá kỹ vào tập huấn luyện và khó khái quát hóa trên dữ liệu mới, đặc biệt nếu tập kiểm tra không đủ đa dạng.

Mỗi mô hình có những ưu và nhược điểm riêng, phụ thuộc vào kiến trúc và cách học của chúng. *Bảng 4.9* dưới đây so sánh các khía cạnh quan trọng như khả năng học phi tuyến, tốc độ suy luận và khả năng mở rộng của hai phương pháp này.

Bảng 4.9 *So sánh đặc điểm của MLP và Random Forest*

Mô hình	Khả năng học phi tuyến	Giải thích	Tốc độ suy luận	Khả năng mở rộng
MLP	Cao	Khó	Trung bình	Cao
Random Forest	Thấp	Dễ	Nhanh	Trung bình

Để làm rõ hơn về điểm mạnh và điểm yếu của hai phương pháp này, *Bảng 4.10* dưới đây tổng hợp các lợi thế và hạn chế khi triển khai MLP và Random Forest trong phát hiện tấn công DDoS:

Bảng 4.10 *Bảng tổng hợp điểm mạnh và điểm yếu của MLP và Random Forest*

Phương pháp	Điểm mạnh	Điểm yếu
MLP	Khả năng học phi tuyến tốt, phát hiện tấn công phức tạp, độ chính xác cao.	Yêu cầu dữ liệu lớn, khó giải thích, thời gian huấn luyện dài.
Random Forest	Dễ triển khai, nhanh, dễ giải thích.	Dễ overfitting, kém hiệu quả với dữ liệu phi tuyến.

b. Nhận xét kết quả triển khai hệ thống

Sau khi triển khai và thử nghiệm hệ thống trong ba kịch bản khác nhau, kết quả thu được cho chúng ta thấy sự khác biệt rõ rệt về hiệu quả giữa các phương pháp ngăn chặn sau khi phát hiện tấn công DDoS. *Bảng 4.11* sẽ so sánh về mặt hiểu quả giữa ba kịch bản phát hiện và ngăn chặn tấn công DDoS của hệ thống.

Bảng 4.11 *Bảng so sánh hiệu quả giữa ba kịch bản*

Kịch bản triển khai	Mức độ can thiệp	Mức độ bảo vệ mạng	Ảnh hưởng đến lưu lượng hợp lệ	Phù hợp trong tình huống
Phát hiện nhưng không can thiệp	Không có	Không	Ảnh hưởng nghiêm trọng	Đánh giá mô hình phân loại
Phát hiện và chặn cổng tấn công	Mạnh	Cao	Không ảnh hưởng	Khi cần bảo vệ tuyệt đối cho các host

Phát hiện và giới hạn băng thông	Trung bình	Trung bình	Ảnh hưởng nhẹ	Khi cần giữ kết nối nhưng giảm thiểu tác hại

Dựa trên *Bảng 4.11* cùng với các kết quả thực nghiệm thu được trong quá trình phát hiện và xử lý tấn công DDoS, có thể rút ra những nhận định cụ thể về ưu điểm và hạn chế của từng kịch bản triển khai. Các phân tích này được tổng hợp trong *Bảng 4.12* bên dưới, nhằm làm rõ hơn hiệu quả thực tế của từng phương pháp phản ứng.

***Bảng 4.12** Bảng tổng hợp điểm mạnh và điểm yếu của mỗi kịch bản*

Kịch bản triển khai	Điểm mạnh	Điểm yếu
Phát hiện nhưng không can thiệp	Cho phép đánh giá độ chính xác của mô hình trong điều kiện thực tế	Không bảo vệ host, dễ dẫn đến mất kết nối do tấn công
Phát hiện và chặn cổng tấn công	Ngăn chặn triệt để tấn công, duy trì ổn định mạng cho các host hợp lệ	Có thể chặn nhầm nếu phát hiện sai, không linh hoạt nếu host dùng nhiều cổng
Phát hiện và giới hạn băng thông	Duy trì tính sẵn sàng mạng, làm suy yếu tấn công mà không triệt tiêu kết nối	Tấn công vẫn còn tồn tại ở mức độ thấp, ảnh hưởng nhẹ đến host hợp lệ

Ba kịch bản triển khai đã chứng minh tính hiệu quả và linh hoạt của hệ thống trong môi trường SDN:

- Khả năng phát hiện chính xác được đảm bảo nhờ mô hình học sâu MLP, cho phép nhận diện hiệu quả các luồng tấn công trong thời gian thực.
- Controller đóng vai trò trung tâm trong việc đưa ra quyết định và cập nhật quy tắc tại switch, từ đó tạo điều kiện cho phản ứng nhanh chóng và linh hoạt.
- Tùy thuộc vào yêu cầu cụ thể về độ bảo vệ và tính sẵn sàng của hệ thống, có thể lựa chọn chiến lược phản ứng phù hợp: từ cảnh báo giám sát, đến chặn triệt để, hoặc giới hạn lưu lượng.

Việc triển khai này cho thấy tiềm năng lớn của mô hình SDN kết hợp AI trong các giải pháp phòng thủ mạng thông minh, dễ tùy biến và phù hợp với nhiều môi trường khác nhau.

Chương 5: Kết luận

5.1. Tóm tắt kết quả

Về kết quả huấn luyện mô hình, mô hình MLP đã chứng minh được khả năng phân loại tốt các mẫu tấn công DDoS, với Accuracy đạt 99.83%, Precision 99.79%, Recall 99.95%, F1-Score 99.87%, và AUC 99.77%. Mặc dù kết quả thấp hơn một chút so với Random Forest, MLP có ưu thế rõ rệt khi xử lý các mẫu dữ liệu phi tuyến và phức tạp, nhờ khả năng học từ các đặc trưng ẩn và tối ưu hóa thông qua nhiều lớp kết nối.

Về kết quả phát hiện và ngăn chặn tấn công DDoS, sau khi nhóm triển khai ba kịch bản (phát hiện không can thiệp, phát hiện và chặn cổng tấn công, phát hiện và giới hạn băng thông), hệ thống đã chứng minh khả năng nhận diện chính xác sự khác biệt giữa lưu lượng bình thường và lưu lượng tấn công. Đồng thời, cơ chế phòng thủ cũng hoạt động hiệu quả, thông qua việc chặn hoàn toàn hoặc giới hạn băng thông từ cổng tấn công, góp phần bảo vệ tính ổn định và an toàn cho mạng.

5.2. Đóng góp của dự án

Dự án đã đề xuất và triển khai thành công mô hình MLP cho phát hiện và ngăn chặn tấn công DDoS trong mạng SDN, đạt độ chính xác cao và khả năng phân loại tốt. Đồng thời, nghiên cứu đã cung cấp bảng so sánh chi tiết giữa các mô hình học sâu như MLP và mô hình học máy Random Forest, giúp làm rõ điểm mạnh và điểm yếu của từng phương pháp, từ đó hỗ trợ việc lựa chọn mô hình phù hợp với các yêu cầu bảo mật khác nhau.

Ngoài ra, các chỉ số đánh giá quan trọng như Accuracy , Precision, Recall, F1-Score đã được phân tích kỹ lưỡng, tạo cơ sở cho việc đánh giá chính xác hiệu quả của mô hình. Dự án này cũng đặt nền tảng cho các nghiên cứu tiếp theo về phát hiện tấn công mạng sử dụng học sâu và học máy, mở ra hướng đi mới trong việc xây dựng các hệ thống phòng thủ mạng thông minh và tự thích ứng.

Dự án đã xây dựng thành công một hệ thống phát hiện và ngăn chặn tấn công DDoS trong môi trường SDN bằng cách tích hợp mô hình MLP vào Ryu Controller. Thông qua việc triển khai ba kịch bản khác nhau đã cho chúng ta thấy được hệ thống không chỉ nhận diện chính xác lưu lượng tấn công mà còn phản ứng linh hoạt bằng cách chặn hoàn toàn hoặc giới hạn băng thông đối với luồng độc hại. Cơ chế xử lý theo thời gian thực giúp giảm tải cho controller, bảo đảm lưu lượng hợp lệ được duy trì ổn định. Kết quả này khẳng định tiềm năng áp dụng trí tuệ nhân tạo vào bảo mật mạng SDN một cách hiệu quả và khả thi trong thực tiễn.

5.3. Hạn chế của dự án

Mặc dù mô hình MLP đã chứng minh được hiệu quả trong phát hiện tấn công DDoS, nhưng vẫn tồn tại một số hạn chế cần khắc phục.

Đầu tiên, MLP yêu cầu nhiều tài nguyên phần cứng và thời gian huấn luyện dài, khiến việc triển khai trên các hệ thống có tài nguyên hạn chế trở nên khó khăn.

Ngoài ra, mô hình này chưa được đánh giá khả năng mô hình hóa các kiểu tấn công mới như Slow-rate Attack hay Low-volume Attack, những kiểu tấn công này có đặc điểm rất khác biệt và thường khó phát hiện hơn.

Một điểm yếu khác là sự phụ thuộc vào chất lượng và kích thước của tập dữ liệu huấn luyện, dễ bị ảnh hưởng bởi dữ liệu nhiễu và mất cân bằng, làm giảm độ chính xác của mô hình.

Cuối cùng, mô hình chưa được tối ưu hóa để giảm độ trễ trong quá trình phát hiện tấn công, yếu tố quan trọng khi triển khai thực tế, đòi hỏi mô hình phải phản hồi nhanh và chính xác.

5.4. Hướng phát triển trong tương lai

Để nâng cao hiệu quả phát hiện tấn công DDoS và khắc phục những hạn chế đã nêu, một số hướng phát triển tiềm năng có thể được xem xét trong tương lai. Trước hết, cần cải thiện khả năng mở rộng của mô hình để xử lý tốt hơn khi mạng mở rộng, đảm bảo mô hình vẫn duy trì được hiệu suất cao ngay cả khi lưu lượng mạng tăng đột biến. Bên cạnh đó, việc phát triển các mô hình học thích ứng thông qua kỹ thuật học liên tục sẽ giúp mô hình tự động cập nhật kiến thức từ các kiểu tấn công mới, tăng cường khả năng phản ứng với các mối đe dọa chưa biết trước.

Ngoài ra, để giảm độ trễ phản hồi và nâng cao tốc độ phát hiện tấn công, cần nghiên cứu và áp dụng các thuật toán tối ưu hóa thời gian xử lý. Điều này đặc biệt quan trọng trong các hệ thống yêu cầu phản hồi thời gian thực. Hơn nữa, một giải pháp bảo mật toàn diện cần kết hợp với các lớp bảo mật khác như IDS/IPS, Firewall, và Threat Intelligence để tạo ra một hệ thống phòng thủ đa lớp, ngăn chặn nguy cơ bị tấn công.

Cuối cùng, việc triển khai mô hình trên các mạng thực tế với lưu lượng lớn và đa dạng sẽ giúp kiểm chứng tính hiệu quả và độ tin cậy của mô hình, từ đó phát triển các giải pháp bảo mật mạnh mẽ hơn, có thể thích ứng với các môi trường mạng phức tạp và thay đổi liên tục.

Tài liệu tham khảo

[1] T. Sim, "SDN Architecture Explained: 5 Key Features," Telefocal, May 31, 2023. [Online]. Available: **https://www.telefocal.com/sdn-architecture-explained-5-key-features/**. Accessed: Jun. 19, 2025.

[2] A. Sharma, "What Is Software Defined Networking (SDN)?," GeeksforGeeks, Feb. 6, 2025. [Online]. Available: https://www.geeksforgeeks.org/computer-networks/software-defined-networking/. Accessed: Jun. 19, 2025.

[3] S. C. Forbacha, M. K. Kinteh, and M. Hamza, "Enhanced Attacks Detection and Mitigation in Software Defined Networks," American Journal of Computing and Engineering, vol. 7, no. 3, pp. 40–80, 2024. Accessed: Jun. 19, 2025.

[4] C. Biradar, "DDoS Attack Detection and Mitigation," GitHub. [Online]. Available: **https://github.com/chiragbiradar/DDoS-Attack-Detection-and-Mitigation** . Accessed: May 11, 2025.

[5] Mininet Team, "Mininet: An Instant Virtual Network on your Laptop (or other PC)," Mininet. [Online]. Available: **https://mininet.org/**. Accessed: May 11, 2025.

[6] Open vSwitch Project, "Open vSwitch," Open vSwitch. [Online]. Available: **https://www.openvswitch.org/**. Accessed: May 11, 2025.

[7] Ryu SDN Framework Community, "Ryu: A Component-Based Software Defined Networking Framework," Ryu SDN Framework. [Online]. Available: **https://ryu-sdn.org/**. Accessed: May 11, 2025.

[8] Y. Li, B. Liu, S. Zhai, and M. Chen, "DDoS attack detection method based on feature extraction of deep belief networks," *IOP Conf. Ser.: Earth Environ. Sci.*, vol. 252, no. 3, 2019.

[9] P. Xiao, W. Qu, H. Qi, and Z. Li, "Detecting DDoS attacks against data centers with correlation analysis," *Comput. Commun.*, vol. 67, 2015.

[10] F. Khashab, J. Moubarak, A. Feghali, and C. Bassil, "DDoS Attack Detection and Mitigation in SDN using Machine Learning," in *Proc. IEEE 7th Int. Conf. on Network Softwarization (NetSoft)*, 2021.

[11] N. Z. Bawany, J. A. Shamsi, and K. Salah, "DDoS attack detection and mitigation using SDN: methods, practices, and solutions," *Arabian J. Sci. Eng.*, vol. 42, no. 2, pp. 425–441, Feb. 2017.

[12] T. V. Nguyen, "Multi-Layer Perceptron—MLP," *Machine Learning Cơ Bản*, Feb. 24, 2017. [Online]. Available: **https://machinelearningcoban.com/2017/02/24/mlp/**. Accessed: May 11, 2025.

[13] A. Aggarwal, "Multi-Layer Perceptron (MLP) Learning in TensorFlow," GeeksforGeeks, Apr. 3, 2023. [Online]. Available: **https://www.geeksforgeeks.org/multi-layer-perceptron-learning-in-tensorflow/**. Accessed: May 11, 2025.