ĐẠI HỌC QUỐC GIA HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG

NGUYỄN ĐỨC TRUNG
NGUYỄN VIỆT HOÀNG
LÊ QUANG MINH

ĐỒ ÁN CHUYÊN NGÀNH

NGHIÊN CỨU VỀ KHẢ NĂNG CHỐNG LẠI CÁC MẪU ĐỐI KHÁNG VÀ MÔ HÌNH KHẢ DIỄN GIẢI CHO HỆ THỐNG PHÁT HIỆN XÂM NHẬP

A STUDY ON THE RESISTANCE TO ADVERSARIAL SAMPLES AND EXPLAINABLE MODEL FOR INTRUSION DETECTION SYSTEMS

TP. Hồ Chí Minh, 2023

ĐẠI HỌC QUỐC GIA HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG

NGUYỄN ĐỨC TRUNG - 20520956
NGUYỄN VIỆT HOÀNG - 20520189
LÊ QUANG MINH - 20520245

ĐỒ ÁN CHUYÊN NGÀNH

NGHIÊN CỨU VỀ KHẢ NĂNG CHỐNG LẠI CÁC MẪU ĐỐI KHÁNG VÀ MÔ HÌNH KHẢ DIỄN GIẢI CHO HỆ THỐNG PHÁT HIỆN XÂM NHẬP

A STUDY ON THE RESISTANCE TO ADVERSARIAL SAMPLES AND EXPLAINABLE MODEL FOR INTRUSION DETECTION SYSTEMS

GIẢNG VIÊN HƯỚNG DẪN:
ThS. Phan Thế Duy

TP.Hồ Chí Minh - 2023

LỜI CẢM ƠN

Trong quá trình nghiên cứu và hoàn thành đồ án chuyên ngành, nhóm đã nhận được sự định hướng, giúp đỡ, các ý kiến đóng góp quý báu và những lời động viên của các giáo viên hướng dẫn và giáo viên bộ môn. Nhóm xin bày tỏ lời cảm ơn tới thầy Phan Thế Duy đã tận tình trực tiếp hướng dẫn, giúp đỡ trong quá trình nghiên cứu.

Nhóm xin gửi lời cảm ơn đến gia đình và bạn bè đã động viên, đóng góp ý kiến trong quá trình làm đồ án

Nguyễn Việt Hoàng **Nguyễn Đức Trung**

Lê Quang Minh

MỤC LỤC

DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT

IDS	Intrusion Detection System
XAI	Explainable Artificial Intellegence
GAN	Generative Adversarial Network
CNN	Convolutional Neural Network
SVM	Support Vector Machine
RNN	Recurrent Neural Network
DBN	Deep Belief Network
DNN	Deep Neural Network
MLP	Multilayer Perceptron
U2R	User to Root
ART	Adversarial Robustness Toolbox
SHAP	SHapley Additive exPlanations

DANH MỤC CÁC HÌNH VẼ

DANH MỤC CÁC BẢNG BIỂU

TÓM TẮT KHÓA LUẬN

Tính cấp thiết của đề tài nghiên cứu:

Khi thế giới không ngừng thay đổi và ngày càng tiên tiến, sự bùng nổ của những thiết bị, hệ thống là miếng mồi ngon cho những kẻ tấn công. Vì thế sự ra đời của hệ thống phát hiện xâm nhập (IDS) là dấu mốc quan trọng trong việc đảm bảo an ninh hệ thống, an ninh mạng. Có rất nhiều phương pháp để triển khai IDS, một trong những phương pháp phổ biến hiện nay đó là sử dụng học máy. Bằng cách áp dụng học máy, IDSs có thể xử lý các dữ liệu lớn và cho hiệu năng tốt hơn. Tuy nhiên, các nghiên cứu gần đây cho thấy các mô hình học máy phân loại (classification) dễ bị tấn công đối kháng. Do đó, nhóm chúng tôi đề xuất một phương pháp phòng chống tấn công đối kháng trong học máy dựa trên hệ thống phát hiện xâm nhập tích hợp trí tuệ nhân tạo khả diễn giải (XAI). Phương pháp đề xuất của chúng tôi chia làm ba giai đoạn: tạo mẫu đối kháng, tạo whitelist và phát hiện tấn công đối kháng. Để tạo mẫu đối kháng, chúng tôi sử dụng các kĩ thuật tạo mẫu đối kháng để đánh giá độ bền của các mô hình học sâu. Sau đó, huấn luyện mô hình IDS và trích xuất những thuộc tính được xác định là Normal để làm whitelist bằng cách sử dụng SHAP (SHapley Additive exPlanations). Bằng cách dựa vào whitelist này, những tấn công đối kháng qua mặt IDS sẽ bị phát hiện nếu những thuộc tính của chúng không nằm trong whitelist và sẽ bị phân loại thành tấn công đối kháng. Nhóm chúng tôi sử dụng tập dữ liệu InSDN để đánh giá phương pháp đề xuất này.

CHƯƠNG 1. TỔNG QUAN

Chương này giới thiệu về ngữ cảnh và các nghiên cứu liên quan. Đồng thời, trong chương này chúng tôi cũng trình bày phạm vi và cấu trúc của Đồ án chuyên ngành.

1.1. Đặt vấn đề

Trong bối cảnh các thiết bị kết nối Internet gia tăng một cách nhanh chóng, các nhà nghiên cứu cũng như các tổ chức đã và đang tìm hiểu, triển khai những biện pháp bảo vệ nhằm phát hiện các lỗ hổng bảo mật trong các hệ thống, thiết bị. Hệ thống phát hiện xâm nhập (Intrusion Detection System - IDS) là một biện pháp hiệu quả trong việc tìm kiếm, phát hiện luồng dữ liệu độc hại, không được cấp quyền nhằm cung cấp một môi trường an toàn.

Có rất nhiều phương pháp để triển khai IDS, đặc biệt trong những năm gần đây thì IDS còn được triển khai bằng các mô hình học máy nhằm tăng hiệu suất phát hiện của IDS. Tuy nhiên, những nghiên cứu gần đây cho thấy các mô hình học máy phân loại dễ bị tấn công đối kháng. Những cuộc tấn công đối kháng thì hoạt động bằng cách thay đổi không đáng kể những dữ liệu gốc làm cho các mô hình phân loại nhầm và giảm hiệu năng[6]. Những cuộc tấn công đối kháng đã được thử nghiệm trên mô hình học máy IDS và thực sự làm giảm hiệu năng của chúng. Một trong những phương pháp để phát hiện được tấn công đối kháng đó là trí tuệ nhân tạo khả diễn giải (XAI - explainable AI). XAI là một loại AI cung cấp cho chúng ta tính minh bạch, nguyên nhân và hệ quả, sự công bằng và an toàn liên quan tới quyết định của AI. XAI đã thành công trong việc phát hiện những tấn công đối kháng trong phân loại hình ảnh nhưng chưa có một ứng dụng cụ thể nào cho các IDS.

Do đó chúng tôi đề xuất một khung phát hiện các tấn công đối kháng trong học máy dựa trên hệ thống phát hiện xâm nhập kết hợp XAI. Phương pháp của chúng tôi có 3 giai đoạn là tạo mẫu đối kháng, tạo whitelist và phát hiện mẫu đối kháng. Đầu tiên, tạo các mẫu tấn công đối kháng để đánh giá khả năng của các mô hình học sâu, chúng tôi huấn luyện hệ thống phát hiện xâm nhập bằng mô hình học sâu CNN và MLP bằng tập dữ liệu InSDN. Sau đó, tiến hành trích xuất các thuộc tính được phân loại là Normal từ tập dữ liệu huấn luyện. Tiếp tục đưa vào mô hình XAI để trích xuất ra các thuộc tính giải thích vì sao phân loại là Normal. Dựa vào các giải thích của XAI, chúng tôi tổng hợp thành một whitelist cho giai đoạn phát hiện. Trong giai đoạn phát hiện tấn công, hệ thống phát hiện xâm nhập đã được huấn luyện trước đó sẽ có đầu vào là các tấn công đối kháng đã tạo, nếu được phân loại là tấn công thì IDS sẽ gửi cảnh báo. Nếu như IDS không thể phân loại được thì ta sẽ tới lớp phòng thủ tiếp theo, đó là whitelist đã tạo trong giai đoạn trước. Đối với phần giải thích của XAI, chúng tôi sử dụng mô hình SHAP để giải thích dữ liệu.

1.2. Giới thiệu những nghiên cứu liên quan

1.2.1. Tấn công đối kháng vào hệ thống phát hiện xâm nhập

Đã có nhiều nghiên cứu được thực hiện thành công về tấn công đối kháng vào hệ thống phát hiện xâm nhập dựa trên học máy. Tấn công sử dụng những mẫu đối kháng đã làm giảm hiệu năng của các mô hình học máy một cách tinh vi. Theo nghiên cứu của [8], một nghiên cứu tiêu biểu gần đây, nhóm tác giả đã đề xuất TIKI-TAKA, một khung công cụ toàn diện được thiết kế để (i) đánh giá tính ổn định của các hệ thống phát hiện xâm nhập hàng đầu dựa trên học sâu đối với các biện pháp can thiệp đối kháng, và (ii) tích hợp các cơ chế phòng thủ được đề xuất bởi chính họ nhằm tăng cường khả năng chống lại các chiến lược tấn công né tránh như vậy.

1.2.2. Phát hiện các cuộc tấn công đối kháng trong hệ thống phát hiện xâm nhập bằng mô hình khả diễn giải

Theo xu hướng nghiên cứu gần đây, đã xuất hiện khá nhiều nghiên cứu ứng dụng mô hình khả diễn giải để phát hiện được các cuộc tấn công đối kháng nhưng lại có ít nghiên cứu ứng dụng trong việc hỗ trợ cho các hệ thống phát hiện xâm nhập. Một trong những nghiên cứu đầu tiên và tiêu biểu nhất là nghiên cứu của [6]. Nhóm tác giả đã đề xuất một khung công cụ để nhận biết các mẫu đối kháng trong các hệ thống phát hiện xâm nhập dựa trên máy học, sử dụng XAI (Explaining Artificial Intelligence). Khung công cụ bao gồm hai giai đoạn: khởi tạo và phát hiện. Trong giai đoạn khởi tạo, chúng ta huấn luyện một hệ thống phát hiện xâm nhập (IDS) dựa trên mô hình phân loại SVM và trích xuất các giải thích của dữ liệu bình thường từ tập dữ liệu bằng cách sử dụng LIME (giải thích mô hình cục bộ không phụ thuộc vào mô hình).

1.3. Tính ứng dụng

Đề tài này tích hợp hệ thống phát hiện xâm nhập với mô hình khả diễn giải. Các mẫu đối kháng được tạo ra sẽ qua một mô hình hai lớp để kiểm tra. Vì vậy kết quả sau khi thực nghiệm sẽ nêu lên tính ứng dụng của hệ thống.

1.4. Những thách thức

Khác với các mô hình trước, sử dụng mô hình khả diễn giải là LIME chứ không phải SHAP. Khi sử dụng SHAP, việc trích xuất diễn giải sẽ phụ thuộc vào mô hình và tốn rất nhiều thời gian.

1.5. Mục tiêu và cấu trúc đồ án chuyên ngành

1.5.1. Mục tiêu nghiên cứu

Ứng dụng các loại tấn công đối kháng để tạo mẫu đối kháng, đồng thời thử nghiệm khả năng phát hiện mẫu đối kháng với mô hình phát hiện xâm nhập tích hợp mô hình khả diễn giải.

1.5.2. Cấu trúc đồ án chuyên ngành

Chúng tôi xin trình bày nội dung của đồ án chuyên ngành theo cấu trúc như sau:

- Chương 1: Giới thiệu tổng quan về đề tài của Đồ án và những nghiên cứu liên quan.
- Chương 2: Trình bày cơ sở lý thuyết và kiến thức nền tảng liên quan đến đề tài.
- Chương 3: Trình bày mô hình nghiên cứu đề xuất.
- Chương 4: Trình bày thực nghiệm và đánh giá.
- Chương 5: Kết luận và hướng phát triển của đề tài.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

Chương này trình bày cơ sở lý thuyết của nghiên cứu: Bao gồm hệ thống phát hiện xâm nhập, các kỹ thuật sinh đối kháng và mô hình khả diễn giải cho hệ thống tìm kiếm, phát hiện xâm nhập.

2.1. Hệ thống phát hiện xâm nhập (IDS)

Hệ thống phát hiện xâm nhập (IDS) là hệ thống phần mềm hoặc phần cứng tự động thực hiện quy trình phát hiện xâm nhập bao gồm theo dõi các sự kiện diễn ra trong một hệ thống máy tính hoặc mạng máy tính và phân tích để nhận biết các dấu hiệu của sự bất thường (hành vi xâm nhập - intrusion). IDS chủ yếu được phân loại dựa theo nguồn dữ liệu và kỹ thuật phân tích:

- NIDS (Network-based IDS): thường được triển khai ở biên mạng như gần firewall hoặc router biên, server VPN, server remote access và mạng không dây. NIDS có chức năng theo dõi lưu lượng mạng cho một phần mạng (network segment) hoặc các thiết bị, phân tích các hoạt động mạng và các giao thức ứng dụng để xác định hành vi bất thường.

- HIDS (Host-based IDS): thường được triển khai ở các host quan trọng như các server có thể truy cập internet và các server chứa thông tin quan trọng. HIDS có chức năng theo dõi các đặc điểm của một host và các sự kiện xảy ra trong host đó (trong mạng LAN) để nhận biết các hành vi đáng ngờ.

- Signature-based IDS: hay còn được gọi là knowledge-based, đây là IDS hoạt động giống các phần mềm diệt virus, dựa vào các chữ ký (signature) và các dấu hiệu của nguy cơ cũng như tấn công đã biết từ trước. Loại IDS này

không thể phát hiện các bất thường chưa biết trước hoặc biến đổi nhỏ trong những tấn công đã biết.

- Anomaly-based IDS: hay còn được gọi là profile-based, IDS này hoạt động bằng cách dựa trên các profile về các hành vi bình thường, dự kiến được thiết lập trước (giống như một whitelist). Bất kỳ hành vi nào khác nằm ngoài profile đều được xem là bất thường nên có thể phát hiện các tấn công đã biết và chưa biết. Tuy nhiên có tỉ lệ false positive cao vì nhầm hành vi bình thường thành tấn công (do profile chưa được xây dựng chặt chẽ).

2.2. Giới thiệu về học máy

2.2.1. Khái niệm học máy

Học máy là một phương pháp sử dụng các kỹ thuật, thuật toán nhằm tự động hóa việc đưa ra các dự đoán dựa vào các quan sát đã xuất hiện. Có 2 loại học máy là phân loại (classification) và dự đoán (prediction). Những bài toán phân loại như nhận diện hình ảnh, chữ viết,... Những bài toán dự đoán điển hình như dự đoán giá trị cổ phiếu, giá bất động sản, xu hướng thị trường trong tương lai...

2.2.2. Các loại học máy

- Học không giám sát (Unsupervised learning): thuật toán không dự đoán đầu ra hoặc nhãn mà phụ thuộc vào dữ liệu đầu vào mà thuật toán sẽ sử dụng cấu trúc của dữ liệu để thực hiện các tác vụ như phân nhóm hoặc giảm số chiều dữ liệu để thuận tiện cho việc lưu trữ và tính toán. Vì tập dữ liệu huấn luyện không cần nhãn, việc thiết lập phương pháp này rất dễ dàng. Tuy nhiên, vì không có nhãn nên các thuật toán không giám sát không thể đưa ra dự đoán trực tiếp mà cần trải qua bước tiền xử lý dữ liệu trước khi đưa vào huấn luyện. Các thuật toán phổ biến về học không giám sát có

thể kể đến như K-Means và đối với học sâu có Recurrent Neural Network (RNN), Deep Belief Network (DBN)...

- Học giám sát (Supervised learning): dự đoán kết quả của một dữ liệu mới dựa trên các cặp dữ liệu đã biết trước. Trong quá trình huấn luyện, dữ liệu đã biết có vai trò như một người giám sát trong việc hướng dẫn thuật toán học. Học có giám sát có lợi thế về tính đơn giản và dễ thiết kế. Tuy nhiên, một thách thức đối với học có giám sát là việc gắn nhãn dữ liệu, đặc biệt khi không có sẵn nhãn cho dữ liệu. Các thuật toán phổ biến trong học có giám sát bao gồm Linear Regression, Logistic Regression, Random Forest, Decision Tree và các mô hình học sâu như Deep Neural Network (DNN) và Convolutional Neural Network (CNN)...

- Học bán giám sát (Semi-supervised learning): ta thường gặp trường hợp chỉ có một phần dữ liệu trong tập dữ liệu được gán nhãn. Khi đó ta sẽ kết hợp hai phương pháp học máy giám sát và không giám sát. Đầu tiên, ta sử dụng dữ liệu đã được gán nhãn để huấn luyện một phần thuật toán học máy. Sau đó, phần thuật toán đã được huấn luyện sẽ tự động gán nhãn cho phần dữ liệu chưa được gán nhãn thông qua một quá trình được gọi là giả gắn nhãn. Phương pháp này có tính thực tế vì việc thu thập dữ liệu gán nhãn thường tốn nhiều thời gian và tài nguyên.

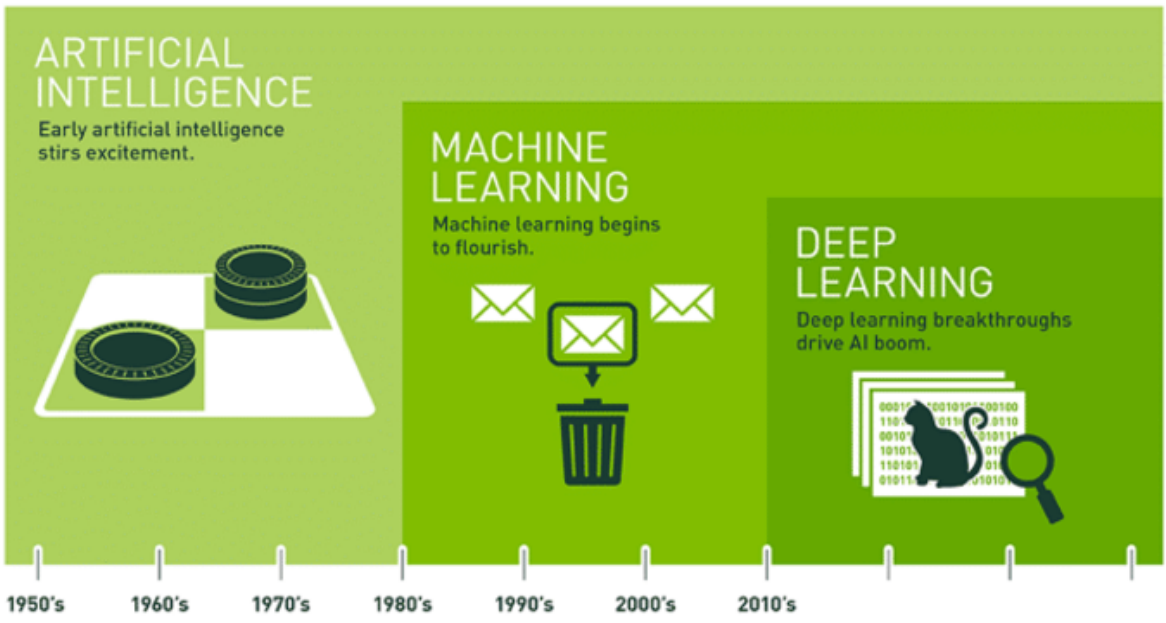
- Học tăng cường (Reinforcement learning): đưa ra các dự đoán dựa trên việc thử và sai nhằm đạt được kết quả tốt nhất, dạy cho các máy (agent) thực hiện tốt 1 nhiệm vụ (task) bằng tương tác với môi trường (environment) thông qua hành động (action) và nhận được phần thưởng (reward)

2.3. Giới thiệu về học sâu

2.3.1. Khái niệm học sâu

Học sâu là một lĩnh vực của học máy, nơi máy tính được đào tạo để học một cách tự nhiên giống như con người. Học sâu được áp dụng chủ yếu trong các ứng dụng như xe tự lái, cho phép chúng tham gia giao thông mà không cần sự can thiệp của con người. Ngoài ra, học sâu cũng được áp dụng trong các thiết bị thông minh như trợ lý ảo trên loa thông minh, máy tính bảng và điện thoại thông minh. Vì những lợi ích này, học sâu đang trở thành một xu hướng quan trọng, thu hút sự quan tâm rất lớn và đạt được những thành tựu đáng kể, với tiềm năng phát triển tiếp theo.

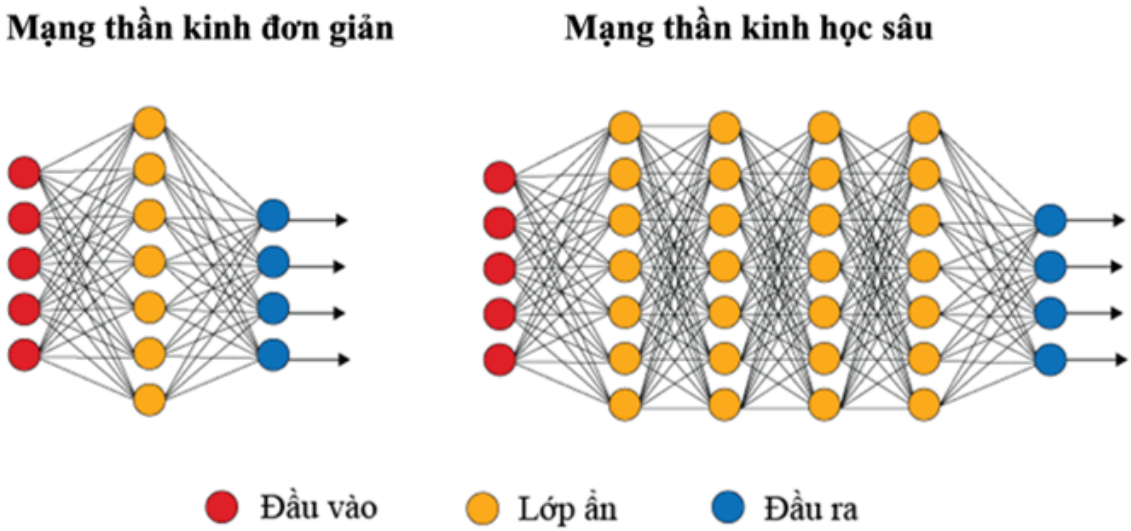
Kiến trúc của học sâu bao gồm nhiều lớp dữ liệu được gán nhãn và sử dụng nhiều kiến trúc mạng nơ-ron nhân tạo. Dữ liệu được đưa qua các lớp mạng từ lớp đầu vào, đi qua các lớp ẩn và kết thúc tại lớp đầu ra. Các lớp mạng ẩn trong kiến trúc học sâu cung cấp khả năng học mạnh mẽ, giúp thuật toán học sâu đạt được kết quả tốt hơn so với các mô hình học máy truyền thống.



Hình 2.1: *Mối quan hệ giữa AI, học máy và học sâu*

2.3.2. Một số khái niệm trong học sâu

- Mạng nơ-ron: Mạng nơ-ron trong học sâu mô phỏng lại cấu trúc mạng lưới nơ-ron trong não người, trong đó các nơ-ron được kết nối với nhau. Các nơ-ron trong mạng nơ-ron được chia thành ba loại là lớp đầu vào, lớp ẩn và lớp đầu ra.



Hình 2.2: *Cấu trúc mạng thần kinh và các lớp*

- Tế bào thần kinh (perceptron): Một tế bào thần kinh có thể được hiểu đơn giản như một hàm toán học, nơi nó nhận đầu vào từ một hoặc nhiều số và thực hiện các phép toán để tính toán kết quả đầu ra. Trọng số của tế bào thần kinh là các giá trị mà chúng ta cần tìm và được xác định thông qua quá trình huấn luyện.

- Hàm kích hoạt (activation functions): Trong một mô hình, các nơ-ron trong lớp ẩn sử dụng các hàm phi tuyến tính để tính toán đầu ra của chúng và chuyển tiếp nó cho lớp tiếp theo. Có một số hàm kích hoạt phổ biến được sử dụng, bao gồm Sigmoid, Tanh và Rectified Linear Unit (ReLU). Các hàm kích hoạt này giúp tạo ra tính phi tuyến tính và khả năng học linh hoạt cho mô hình.

– Sigmoid: Hàm sigmoid hay còn được gọi là đường cong sigmoid, là một hàm liên tục mà ánh xạ đầu vào từ các số thực vào các giá trị trong khoảng từ 0 đến 1. Hàm này được sử dụng trong học máy để chuyển đổi đầu vào thành xác suất hoặc những giá trị có ý nghĩa xác suất. Giá trị trả về được biểu diễn dưới dạng một hàm số.

– Tanh: Hàm tanh là một hàm kích hoạt được sử dụng trong học máy, với đặc điểm là đầu ra của nó nằm trong khoảng (-1, 1). Điều này làm cho hàm tanh phù hợp cho các mô hình có đầu ra với ba giá trị: âm, trung tính (0) và dương. Hàm tanh giúp biểu diễn các mức độ khác nhau của đầu vào và tạo ra một phản ứng tương tự như hàm sigmoid, nhưng với khoảng giá trị mở rộng hơn. Hàm tanh cũng là một hàm liên tục và có thể biểu diễn dưới dạng một hàm số.

– ReLU: Hàm ReLU (Rectified Linear Unit) được xây dựng dựa trên ý tưởng loại bỏ các tham số không quan trọng trong quá trình huấn luyện, nhằm tạo ra một mô hình mạng nhẹ, nhanh chóng và hiệu quả hơn. Hàm ReLU thực hiện việc giữ nguyên các giá trị đầu vào lớn hơn 0, trong khi đối với các giá trị nhỏ hơn 0, chúng được coi như là 0. Điều này giúp hàm ReLU đơn giản hóa tính toán và giảm độ phức tạp của mạng. Hàm ReLU không có đạo hàm tại 0, nhưng trong thực tế, điều này ít ảnh hưởng đến quá trình huấn luyện và đã được chứng minh là rất hiệu quả trong nhiều mô hình mạng nơ-ron.

– Softmax: Hàm softmax, còn được gọi là hàm trung bình mũ, được sử dụng để tính toán xác suất của một sự kiện, thường được áp dụng trong bài toán phân loại đa lớp. Hàm softmax tính toán khả năng xuất hiện của mỗi lớp trong tổng số các lớp có thể xuất hiện, sau đó sử dụng xác suất này để xác định lớp mục tiêu cho đầu vào. Hàm softmax giúp chúng ta hiểu mức độ đáng tin cậy của các lớp và thường được sử dụng để tạo ra phân phối xác suất đa lớp.

– Dropout: Dropout là một kỹ thuật được sử dụng để ngăn chặn hiện

tượng overfitting (quá khớp) trong mô hình học máy. Kỹ thuật này hoạt động bằng cách ngẫu nhiên loại bỏ một số đơn vị (neuron) trong quá trình huấn luyện. Khi loại bỏ một đơn vị, nó sẽ không được sử dụng trong quá trình tính toán và cập nhật các trọng số trong mạng. Dropout giúp giảm sự phụ thuộc quá mức giữa các đơn vị trong mạng nơ-ron kết nối đầy đủ (fully-connected) trong mô hình học sâu. Điều này có tác dụng giúp mô hình trở nên chống lại overfitting và tổng quát hóa tốt hơn trên dữ liệu mới.

- One-hot Coding: One-hot encoding là một phương pháp được sử dụng để biểu diễn các biến hoặc lớp đầu ra trong các bài toán phân loại. Phương pháp này chuyển đổi các giá trị thành các đặc trưng nhị phân chỉ có giá trị 1 hoặc 0. Mỗi mẫu trong tập dữ liệu sẽ được chuyển thành một vector có kích thước n, trong đó giá trị 1 chỉ ra trạng thái "active" và giá trị 0 cho trạng thái "inactive" của đặc trưng tương ứng. One-hot encoding giúp đưa thông tin về sự hiện diện hoặc vắng mặt của một đặc trưng trong một mẫu cụ thể.
- Max pooling: Max pooling là một lớp được áp dụng giữa các lớp tích chập trong mô hình học sâu nhằm giảm kích thước của dữ liệu thông qua quá trình lấy mẫu. Quá trình này thực hiện bằng cách chia dữ liệu thành các ô nhỏ và chọn giá trị lớn nhất (max) trong mỗi ô làm giá trị đại diện. Kỹ thuật max pooling giúp giảm kích thước dữ liệu, giữ lại các đặc trưng quan trọng và giảm hiện tượng overfitting (quá khớp) trong mô hình học sâu.

2.4. Các mô hình học sâu sử dụng trong đề tài

2.4.1. Mô hình Convolutional neuron network (CNN)

Convolutional Neural Network là một loại mạng nơ-ron, thường được áp dụng cho các bài toán phân loại và thị giác máy tính. Nó cung cấp một phương pháp

tiếp cận tốt và có khả năng mở rộng bằng cách sử dụng các nguyên tắc từ đại số tuyến tính, đặc biệt là phép nhân ma trận để xác định các mẫu nằm trong dữ liệu. Đồng thời, so với các mạng nơ-ron khác, CNN có hiệu suất vượt trội khi xử lý các đầu vào là tín hiệu hình ảnh, giọng nói hoặc âm thanh. CNN sử dụng ba lớp chính để xử lý dữ liệu:

- Lớp tích chập (convolutional): Là thành phần chính và nơi quan trọng trong quá trình học và tính toán của mạng nơ-ron. Nó sử dụng các bộ lọc, còn được gọi là bộ phát hiện đặc trưng, để quét qua từng vùng của dữ liệu đầu vào và xác định sự xuất hiện của các đặc trưng. Ta cũng phải xem xét cẩn thận các siêu tham số (hyperparameters) của bộ lọc, vì chúng ảnh hưởng đến kích thước của dữ liệu đầu ra. Bên cạnh đó, việc chia sẻ các trọng số giữa các vùng của đầu vào giúp bộ lọc không bị thay đổi khi di chuyển qua từng vùng khác nhau của dữ liệu.

- Lớp pooling: Lớp này được sử dụng để giảm kích thước không gian của dữ liệu và giảm số lượng tham số đầu vào. Điều này giúp làm giảm độ phức tạp của mô hình, nâng cao hiệu quả tính toán và hạn chế rủi ro overfitting. Giống như lớp tích chập, lớp pooling cũng sử dụng một bộ lọc để quét qua từng vùng của đầu vào. Tuy nhiên, bộ lọc này không có trọng số như lớp tích chập. Thay vào đó, nó sử dụng một hàm tổng hợp trên từng vùng tiếp nhận của đầu vào và đưa ra một giá trị duy nhất cho mỗi vùng, sau đó ghi kết quả này vào một mảng đầu ra. Điều này giúp giảm kích thước của dữ liệu mà không ảnh hưởng quá nhiều đến thông tin quan trọng trong dữ liệu.

- Lớp fully-connected: Đây là lớp có nhiệm vụ biến đầu ra của lớp trước đó thành một vector và thực hiện phân loại dựa trên các đặc trưng đã được trích xuất qua các lớp trước đó và bộ lọc tương ứng. Mỗi nút trong lớp fully-connected được kết nối trực tiếp với tất cả các nút trong lớp trước đó và sử dụng các hàm kích hoạt như sigmoid hoặc softmax để tính toán đầu ra và phân loại. Các hàm sigmoid được sử dụng trong trường hợp phân loại

nhị phân, trong khi hàm softmax thường được sử dụng trong bài toán phân loại đa lớp, để xác định xác suất của mỗi lớp đầu ra.

Convolutional Neural Network bắt đầu với lớp convolutional làm lớp đầu tiên. Các lớp sau đó có thể bao gồm các lớp convolutional bổ sung, lớp pooling hoặc lớp fully-connected. Các lớp đầu tiên trong mạng này giúp xác định các tính năng đơn giản trong dữ liệu. Khi qua mỗi lớp, Convolutional Neural Network tăng độ phức tạp của nó để xác định các tính năng lớn hơn và phức tạp hơn.Việc sử dụng Convolutional Neural Network mang đến một số lợi ích:

- Không cần giám sát của con người trong việc xác định các tính năng quan trọng: Mạng nơ-ron convolutional có khả năng tự động học và trích xuất các đặc trưng quan trọng từ dữ liệu, không yêu cầu sự can thiệp của con người trong việc xác định các đặc trưng cụ thể.

- Giảm thiểu số lượng tính toán so với các mạng thần kinh thông thường: Việc sử dụng các lớp convolutional và pooling giúp giảm kích thước không gian dữ liệu, từ đó giảm số lượng tính toán cần thiết, làm cho mạng nơ-ron convolutional có hiệu quả tính toán cao hơn so với các mạng thần kinh thông thường.

- Chia sẻ các trọng số trên các vùng tiếp nhận của một lớp: Một trong những đặc điểm đáng chú ý của mạng nơ-ron convolutional là khả năng chia sẻ các trọng số giữa các vùng tiếp nhận của một lớp. Điều này giúp giảm số lượng tham số trong mô hình, từ đó giúp mô hình trở nên hiệu quả và dễ huấn luyện hơn.

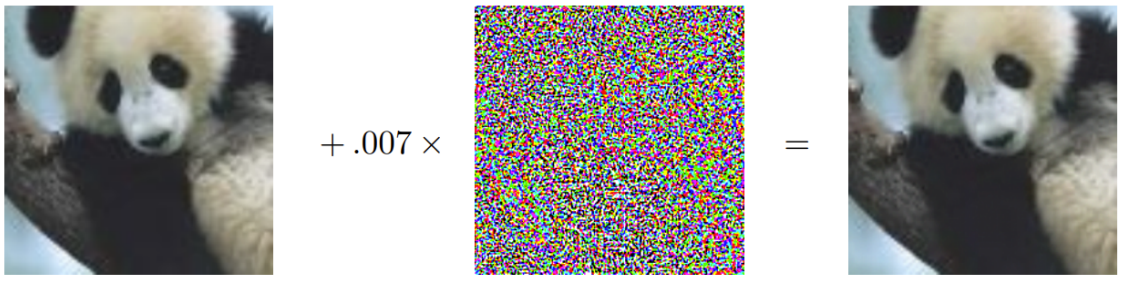
2.4.2. Mô hình Multilayer Perceptron (MLP)

MLP (Multi-Layer Perceptron) là một mạng nơ-ron đa tầng trong học sâu được sử dụng trong các bài toán có giám sát. Mô hình này bao gồm nhiều lớp liên tiếp được kết nối với nhau, bao gồm lớp đầu vào, lớp ẩn và lớp đầu ra. Khi

hoạt động, MLP chuyển tiếp đầu vào từ lớp đầu vào thông qua các lớp ẩn, mỗi nơ-ron tính tổng trọng số của đầu vào và áp dụng một hàm truyền để đưa ra kết quả. Kết quả này tiếp tục truyền qua các lớp ẩn cho đến khi các nơ-ron tại lớp đầu ra đưa ra kết quả dự đoán cuối cùng. MLP được ứng dụng trong các bài toán xử lý dữ liệu dạng bảng, vì dữ liệu đầu vào có thể biểu diễn dưới dạng các vector, giúp mô hình hiểu và phân tích thông tin từ dữ liệu này. Tuy nhiên, một hạn chế của MLP là số lượng thông số cần để định nghĩa mô hình có thể rất lớn, yêu cầu dữ liệu huấn luyện lớn và tính toán phức tạp để đạt được hiệu quả cao. Điều này đòi hỏi sự điều chỉnh cẩn thận của các thông số mô hình để đảm bảo hiệu suất tốt nhất cho từng bài toán cụ thể.

2.5. Tấn công đối kháng (Adversarial attacks)

Tấn công đối kháng là loại tấn cống bằng cách thay đổi không đáng kể các giá trị đầu vào của tập dữ liệu nhằm đánh lừa khả năng phân loại của IDS nhằm làm giảm hiệu năng của mô hình. Hiểu một cách đơn giản, tấn công đối kháng là việc tạo ra các mẫu dữ liệu đưa vào các mô hình học máy và khiến cho mô hình dự đoán sai khác so với thực tế. Ví dụ như hình ảnh bên bên dưới, ta có thể thấy sau khi bị thêm vào một số nhiễu, mô hình đã phân loại sai.



Hình 2.3: *Kết quả mô hình đưa ra kết quả phân loại sai sau khi thêm nhiễu vào mẫu ban đầu*

2.6. Mô hình khả diễn giải (XAI)

Khả năng diễn giải một mô hình có thể được chia thành hai loại: khả năng diễn giả toàn cục (global interpretability) và khả năng diễn giải cục bộ (local interpretability). Khả năng diễn giải toàn cục có nghĩa là người dùng có thể hiểu mô hình trực tiếp từ cấu trúc tổng thể của nó còn khả năng diễn giải cục bộ chỉ kiểm tra một đầu vào và tìm hiểu tại sao mô hình lại đưa ra quyết định cụ thể cho đầu vào đó.

CHƯƠNG 3. THIẾT KẾ HỆ THỐNG

Ở chương này chúng tôi sẽ trình bày mô hình phát hiện tấn công đối kháng bằng mô hình học máy tích hợp mô hình khả diễn giải.

3.1. Phát sinh dữ liệu đối kháng bằng tấn công đối kháng

Theo hướng tiếp cận của tấn công đối kháng, nhóm thực hiện dựa trên 2 mô hình tấn công hộp trắng là *Fast Gradient Method* [3] và *Projected Gradient Descent* [5] và 1 mô hình tấn công hộp đen là *HopSkipJump* [1]. Trước khi thực hiện đánh giá hiệu năng mô hình trước các cuộc tấn công đối kháng, nhóm thực hiện đưa ra lí thuyết tổng quan cho mỗi mô hình tấn công này

1. **Fast Gradient Method** Phương pháp thực hiện các bước cập nhật trọng số theo độ dốc của mô hình để kết quả trả về của mô hình trở nên xa khỏi nhãn đúng đối với mỗi mẫu dữ liệu

2. **Projected Gradient Descent** Là một biến thể của FGM tuy nhiên không bắt đầu tại ảnh gốc mà tại một điểm ngẫu nhiên trong khoảng ϵ xung quanh mẫu dữ liệu gốc

3. **HopSkipJump** Là một phương pháp tấn công dựa trên truy vấn, không sử dụng siêu tham số, bao gồm ba bước chính: (i) ước tính hướng gradient, (ii) tìm kiếm step size thông qua cấp số nhân và (iii) tìm kiếm ranh giới thông qua phương pháp tìm kiếm nhị phân.

Đối với việc tạo các mẫu đối kháng tấn công vào các IDPS áp dụng học máy cần đảm bảo một số yêu cầu như đảm bảo được chức năng và tính nguyên vẹn của mẫu dữ liệu khi được thêm nhiễu. Một cách cụ thể hơn, chỉ một số thuộc

tính trong tập dữ liệu ban đầu được phép thay đổi và các thuộc tính này sau khi được thay đổi này phải giữ nguyên được các đặc trưng vốn ban đầu của nó Ở đây nhóm thực hiện thay đổi 24 thuộc tính được đề cập trong [7]. Từ đó các mẫu dữ liệu được tạo ra vẫn giữ nguyên được chức năng của nó và giữ nguyên được nhãn ban đầu.

3.2. Xây dựng các hệ thống phát hiện xâm nhập dựa trên các mô hình học máy

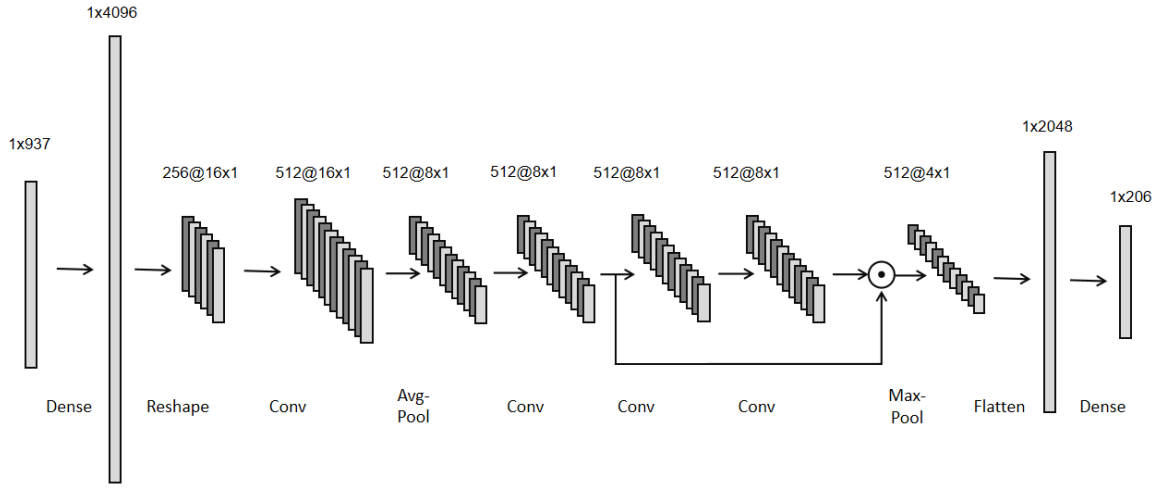
Nhóm xây dựng 2 mô hình học sâu cho việc phân loại tập dữ liệu InSDN, bao gồm Multilayer Perceptron (MLP) và Convolutional Neural Network (CNN).

1. **Multilayer Perceptron** Trong bài toán này nhóm thực hiện xây dựng kiến trúc MLP được mô tả ở bảng 3.1. Mô hình MLP được huấn luyện với tổng cộng 10 epochs, $batch\_size = 10$, hàm Adam được dùng làm hàm tối ưu với $learningrate = 0.001$

Layer	Input	Output	Activation
Dense	input_shape	200	ReLU
Dense	300	300	ReLU
Batch Normalization			
Dropout			
Dense	200	200	Relu
Batch Normalization			
Dense	100	100	Relu
Dense	100	2	Sigmoid

Bảng 3.1: *Kiến trúc mạng MLP*

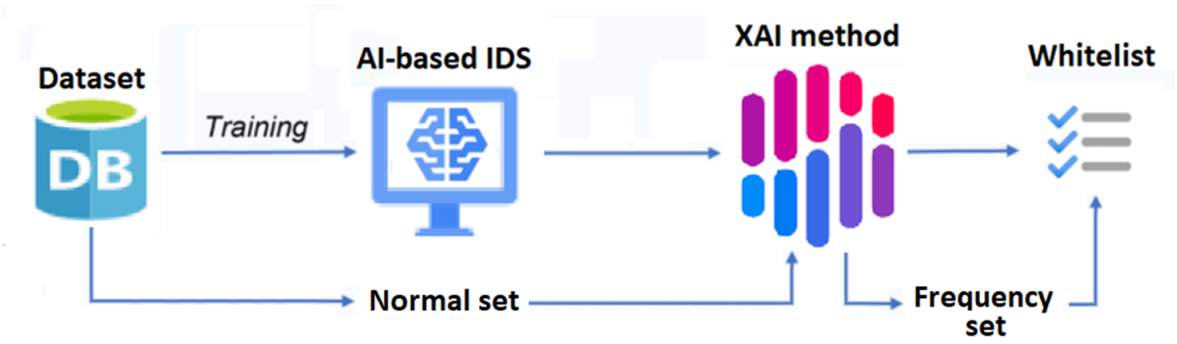
2. **Convolutional Neural Network** Đối với mô hình CNN, nhóm thực hiện huấn luyện mô hình với những tham số như tương tự như mô hình MLP với tổng cộng 10 epochs, $batch\_size = 10$, hàm Adam được dùng làm hàm tối ưu với $learningrate = 0.001$. Với kiến trúc CNN được minh họa như hình 3.1



Hình 3.1: *Kiến trúc mạng CNN*

3.3. Trích xuất danh sách đặc trưng (whitelist) bằng mô hình khả diễn giải

Luồng hoạt động chính của giai đoạn này được thể hiện ở hình 3.2.



Hình 3.2: *Luồng hoạt động của phase 2*

- Đầu tiên, tách những mẫu dữ liệu có nhãn là 'normal' trong tập train ra thành một tập dữ liệu riêng, thu được normal set.

- Xét các mẫu dữ liệu (sample) trong normal set:

- Đối với mỗi sample, sử dụng SHAP để trích xuất top 10 đặc trưng khiến mô hình nhận diện được sample đó là normal.

- Lặp lại bước trên cho đến hết tập dữ liệu.

- Thu được một tập dữ liệu mới chứa top 10 đặc trưng cho tất cả các mẫu dữ liệu trong normal set. Một vài ví dụ cho các mẫu dữ liệu được được thể hiện ở hình 3.3.

```
'Normal sample 0': ['FIN Flag Cnt', 'Idle Max', 'Bwd IAT Max', 'Pkt Len Max', 'Bwd Pkts/s', 'URG Flag Cnt', 'Fwd IAT Max', 'ACK Flag Cnt', 'SYN Flag Cnt', 'Bwd Seg Size Avg'],
'Normal sample 1': ['SYN Flag Cnt', 'ACK Flag Cnt', 'FIN Flag Cnt', 'Bwd Pkts/s', 'Bwd PSH Flags', 'PSH Flag Cnt', 'Init Bwd Win Byts', 'Down/Up Ratio', 'Flow Pkts/s', 'URG Flag Cnt'],
'Normal sample 2': ['SYN Flag Cnt', 'ACK Flag Cnt', 'FIN Flag Cnt', 'Bwd Pkts/s', 'Bwd PSH Flags', 'PSH Flag Cnt', 'Down/Up Ratio', 'Init Bwd Win Byts', 'URG Flag Cnt', 'Flow Pkts/s'],
'Normal sample 3': ['SYN Flag Cnt', 'ACK Flag Cnt', 'FIN Flag Cnt', 'Bwd Pkts/s', 'Bwd PSH Flags', 'PSH Flag Cnt', 'Init Bwd Win Byts', 'Down/Up Ratio', 'URG Flag Cnt', 'Flow Pkts/s'],
'Normal sample 4': ['FIN Flag Cnt', 'Bwd Pkts/s', 'ACK Flag Cnt', 'Bwd PSH Flags', 'PSH Flag Cnt', 'URG Flag Cnt', 'Bwd IAT Tot', 'Bwd URG Flags', 'Flow IAT Max', 'Flow Duration'],
'Normal sample 5': ['SYN Flag Cnt', 'ACK Flag Cnt', 'FIN Flag Cnt', 'Bwd Pkts/s', 'Bwd PSH Flags', 'Init Bwd Win Byts', 'PSH Flag Cnt', 'Down/Up Ratio', 'URG Flag Cnt', 'Flow Pkts/s']
```

Hình 3.3: *Top 10 đặc trưng cho 6 mẫu dữ liệu trích xuất bằng SHAP*

- Tính tần số xuất hiện của từng đặc trưng trên tập dữ liệu vừa thu được, sau đó sắp xếp theo thứ tự tần số giảm dần, thu được frequency set.

- Cuối cùng, nhóm trích xuất ra whitelist từ top N đặc trưng trong frequency set.

3.4. Phát hiện các dữ liệu đối kháng mà mô hình học máy không thể phát hiện bằng whitelist đã được trích xuất

Xét tập dữ liệu chứa các mẫu đối kháng (được tạo ở giai đoạn 1):

- Nếu mô hình phát hiện xâm nhập có thể phát hiện mẫu đối kháng là tấn công, mô hình sẽ phát ra cảnh bảo.

- Nếu mô hình phát hiện xâm nhập đưa ra nhận định mẫu đối kháng là 'normal':

- Mẫu đối kháng đó sẽ được đưa vào SHAP để trích xuất top 10 đặc trưng mà mô hình dựa vào để xác định nó là là normal. Ví dụ top 10 đặc trưng được thể hiện ở hình 3.4

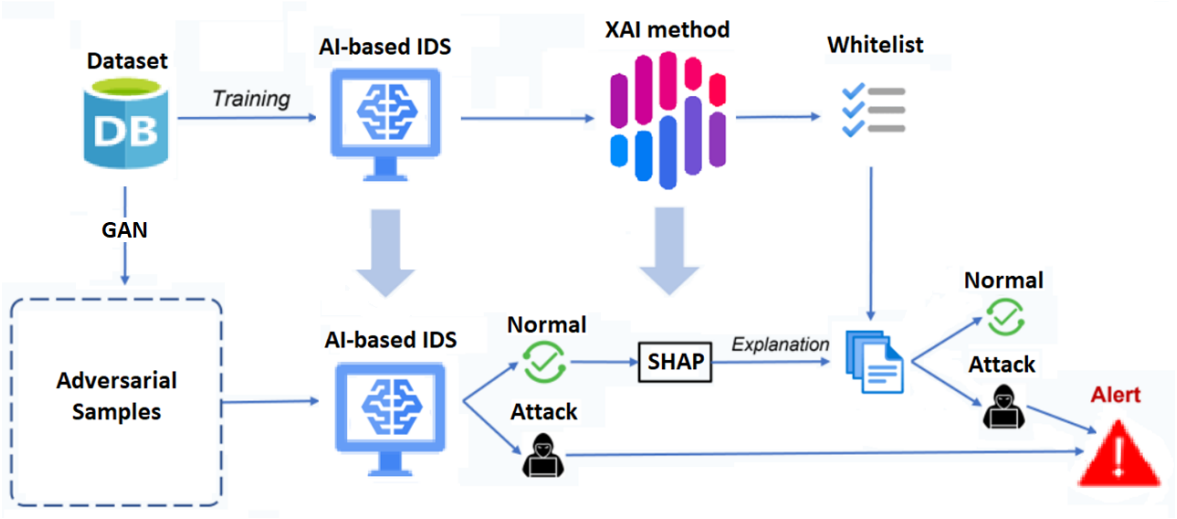
- Nếu có bất kì 1 đặc trưng nào trong top 10 đặc trưng vừa được trích xuất không nằm trong whitelist, mô hình sẽ đưa ra cảnh báo.

```
'Adv sample 0': ['ACK Flag Cnt', 'Bwd IAT Max', 'SYN Flag Cnt', 'Bwd Pkts/s', 'Fwd Pkts/b Avg', 'Active Min', 'Bwd PSH Flags', 'FIN Flag Cnt', 'Flow Pkts/s', 'Bwd IAT Min']
```

Hình 3.4: *Top 10 đặc trưng cho 1 mẫu dữ liệu đối kháng được trích xuất bằng SHAP*

3.5. Luồng hoạt động mô hình đề xuất

Hình 3.5 là sơ đồ của luồng hoạt động của mô hình. Tuần tự các bước như sau:



Hình 3.5: *Mô hình nghiên cứu đề xuất*

- Giai đoạn 1: Tạo tập dữ liệu chứa các mẫu đối kháng bằng tấn công đối kháng.
- Giai đoạn 2: Huấn luyện các mô hình phát hiện xâm nhập, kết hợp với mô hình khả diễn giải (XAI) để trích xuất danh sách đặc trưng (whitelist).
- Giai đoạn 3: Phát hiện các dữ liệu đối kháng mà mô hình học máy không thể phát hiện bằng whitelist đã được trích xuất.

CHƯƠNG 4. THÍ NGHIỆM VÀ ĐÁNH GIÁ

Ở chương này chúng tôi tiến tạo môi trường, cài đặt và đưa ra các tiêu chí đánh giá về mức độ hiệu quả của mô hình.

4.1. Thiết lập thí nghiệm

Hệ thống phát hiện xâm nhập tích hợp mô hình khả diễn giải của chúng tôi được thực hiện trên môi trường Google Colab server với cấu hình RAM là 12GB và dung lượng 100GB. Ngoài ra, ngôn ngữ chính được sử dụng để xây dựng hệ thống là Python.

4.1.1. Tập dữ liệu InSDN

- Một tập dữ liệu SDN [2] toàn diện để xác minh hiệu suất của các hệ thống phát hiện xâm nhập.
- Bao gồm các loại tấn công: DDoS, Probe, Normal, DoS, BFA, Web-Attack, BOTNET, U2R.
- Chứa tổng cộng 343889 mẫu dữ liệu, mỗi dữ liệu có 84 đặc trưng.

4.1.2. Tiền xử lí dữ liệu

Nhóm tiến hành thí nghiệm trên tập dữ liệu InSDN với mô hình phân loại nhị phân, trong đó cột Label bao gồm các loại tấn công sẽ được gộp lại với nhau thành nhãn 1 đại diện cho mẫu độc hại (Malicious) và giữ nguyên các mẫu lành tính với nhãn 0 (Benign). Ngoài ra, toàn bộ các missing values, infinite values cũng sẽ được triệt tiêu nhằm tránh việc giảm khả năng phân loại của mô hình.

Thêm vào đó, nhóm cũng thực hiện việc loại bỏ các cột dữ liệu như Source IP, Destination IP, flow ID,... để tránh gây ra hiện tượng overfitting cho mô hình bởi các đặc trưng này có thể thay đổi từ mạng này sang mạng khác. Tập dữ liệu cuối cùng sẽ có 77 thuộc tính, bao gồm cả thuộc tính Label. Các giá trị giữa các đối tượng vectơ đầu vào có các đại lượng khác nhau sẽ được chuẩn hóa và chuyển đổi về một phạm vi cụ thể. Ở trường hợp này, nhóm thực hiện chuẩn hóa các đối tượng vectơ về khoảng [0,1] bằng phương pháp MinMaxScaler với công thức như sau:

$$x_{\text{scaled}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{4.1}$$

Trong đó:

- x: Giá trị của thuộc tính trước khi chuẩn hóa
- $x_{\min}$: Giá trị nhỏ nhất của thuộc tính trước khi chuẩn hóa
- $x_{\max}$: Giá trị lớn nhất của thuộc tính trước khi chuẩn hóa
- x_{scaled}: Giá trị của thuộc tính sau khi được chuẩn hóa

4.1.3. Adversarial Robustness Toolbox

Adversarial Robustness Toolbox (ART) là một thư viện Python mã nguồn mở cho phục vụ cho mục đích bảo mật trong học máy. ART cung cấp một bộ công cụ toàn diện để đánh giá, bảo vệ và xác minh các mô hình máy học chống lại các cuộc tấn công đối kháng. Các mô-đun tấn công của ART có thể được sử dụng để tạo các mẫu đối kháng khai thác các lỗ hổng của các mô hình máy học. Các mô-đun này bao gồm một loạt các kỹ thuật tấn công, bao gồm tấn công hộp trắng, tấn công hộp đen và tấn công chuyển giao. Ngoài ra ART cũng cung cấp các mô-đun phòng thủ có thể được sử dụng để làm cho các mô hình học máy trở nên mạnh mẽ hơn trước các cuộc tấn công đối kháng. Các mô-đun này

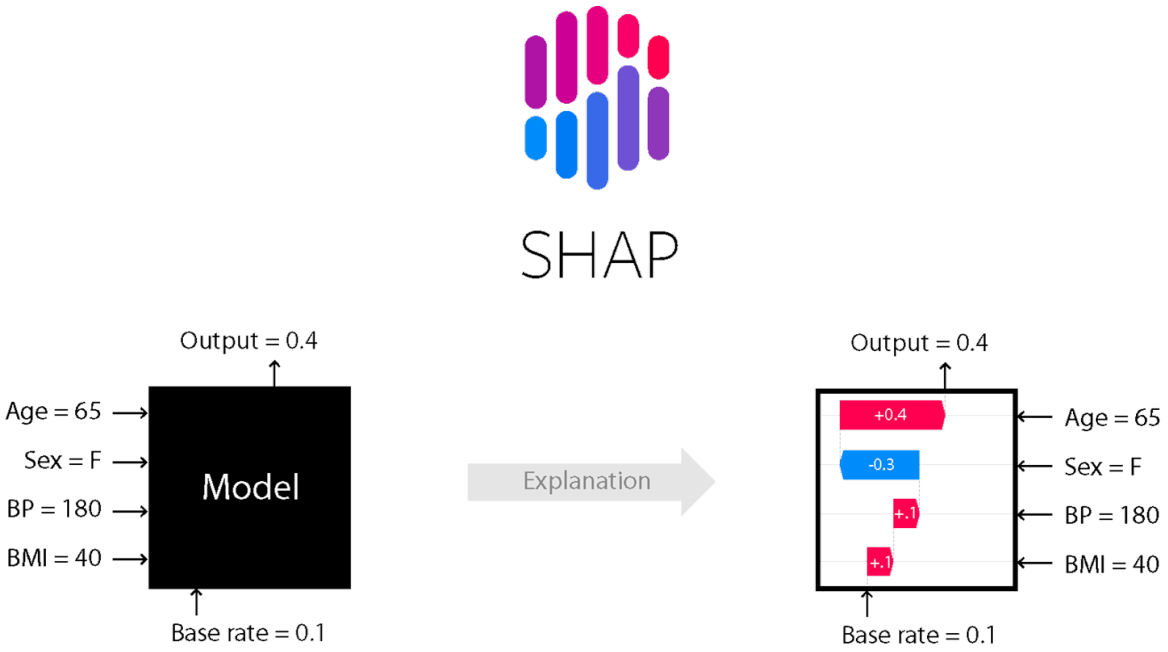
bao gồm một loạt các kỹ thuật phòng thủ, bao gồm huấn luyện đối kháng, tăng cường dữ liệu và chính quy hóa. Các công cụ đánh giá của ART có thể được sử dụng để đánh giá tính dễ bị tổn thương của các mô hình máy học trước các cuộc tấn công đối kháng. Tạo ra các ví dụ về đối kháng, đánh giá độ chính xác của các mô hình và so sánh độ hiệu quả của các kỹ thuật phòng thủ khác nhau. Trong thí nghiệm này nhóm sử dụng bộ công cụ ART để tạo ra những mẫu đối kháng mới từ tập dữ liệu InSDN, nhằm mục đích đánh giá khả năng chống lại trước các cuộc tấn công đối kháng. Ở đây nhóm sử dụng bộ công cụ *Evasion Attack* trong đó bao gồm *Fast Gradient Method*, *Projected Gradient Descent* và *Hop Skip Jump* để thực hiện biến đổi từ tập dữ liệu huấn luyện. Trong đó bộ dữ liệu sẽ được thay đổi chỉ dựa trên một số thuộc tính đã được quy định sẵn và cũng như cần đảm bảo mẫu đối kháng mới được tạo ra vẫn có khả năng qua mặt được các mô hình học sâu.

4.1.4. SHAP

SHAP (SHapley Additive exPlanations) là một phương pháp trong lý thuyết trò chơi được sử dụng để giải thích kết quả của mọi mô hình học máy. Nó liên kết việc phân bổ tín dụng tối ưu với việc cung cấp các diễn giải cục bộ bằng cách sử dụng giá trị Shapley cổ điển từ lý thuyết trò chơi và các phần mở rộng liên quan của nó. Cơ chế hoạt động chung của SHAP có thể được quan sát qua hình 4.1.

Trong đề tài nghiên cứu này, nhóm sử dụng Kernal SHAP Explainer [4] được cung cấp bởi thư viện SHAP [1] để trích xuất các diễn giải cục bộ của mô hình. Kernel SHAP sử dụng một kỹ thuật hồi quy tuyến tính cục bộ có trọng số đặc biệt để tính toán các giá trị SHAP cho các loại mô hình.

[1]https://github.com/slundberg/shap



Hình 4.1: *Cơ chế hoạt động chung của SHAP*

4.2. Kết quả thí nghiệm

Ở mục này, nhóm sẽ trình bày các kết quả thực nghiệm và đưa ra đánh giá.

Nhóm tập trung trả lời câu hỏi sau:

Sự khác biệt về khả năng phát hiện mẫu đối kháng của các mô hình phát hiện xâm nhập trước và sau khi tích hợp mô hình khả diễn giải?

4.2.1. Kết quả xây dựng các mô hình học máy phát hiện tấn công

Nhóm thực hiện đánh giá hiệu năng của mô hình học máy và học sâu khi thực hiện phân loại các mẫu từ tập dữ liệu ban đầu trong bộ dữ liệu InSDN. Nhóm thực hiện chia bộ dữ liệu InSDN thành các thành phần như sau:

- 70% dữ liệu dành cho tập Train
- 30% dữ liệu dành cho tập Test, phục vụ cho mục đích đánh giá hiệu năng của mô hình và sinh mẫu đối kháng

Đánh giá kết quả của mô hình dựa trên 4 tiêu chí: *Accuracy*, *Precision*, *Recall*

và *F1-Score*. Các mô hình học máy được sử dụng để đánh giá bao gồm MLP, CNN và Random Forest

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1-Score**
RF	0.9996	0.9998	0.9996	0.9997
MLP	0.9993	0.9993	0.9998	0.9996
CNN	0.9995	0.9995	0.9999	0.9997

Bảng 4.1: *Kết quả đánh giá của mô hình học máy cho tập dữ liệu InSDN*

Quan sát ta có thể thấy tỉ lệ phát hiện của các mô hình học máy đều cao khi tất cả các chỉ số đánh giá đều trung bình trên 0.999, Ngoài ra, ba mô hình được coi là hoạt động tương tự nhau, vì sự khác biệt giữa điểm số F1-Score đạt được của mỗi mô hình không bao giờ vượt quá 0,001. Điều này phù hợp với yêu cầu đối với các giải pháp IDPS dựa trên các mô hình học máy, từ đó cho thấy sự đáng tin cậy ở mô hình

4.2.2. Tỉ lệ phát hiện mẫu đối kháng thành công không sử dụng mô hình khả diến giải

Trong ngữ cảnh tấn công đối kháng vào các mô hình học máy, đối với các mô hình tấn công hộp trắng, attacker sẽ dựa vào độ dốc của mô hình đi theo hướng ngược lại từ đó tìm được mức độ thêm nhiễu phù hợp. Còn đối các mô hình tấn công hộp đen, attacker sau khi gửi luồng dữ liệu đến mô hình, sẽ dựa vào kết quả trả về của mô hình từ đó đưa ra sự thay đổi và thêm nhiễu phù hợp. Hiệu năng của mô hình cũng được đánh giá dựa trên 4 tiêu chí: *Accuracy*, *Precision*, *Recall* và *F1-Score*

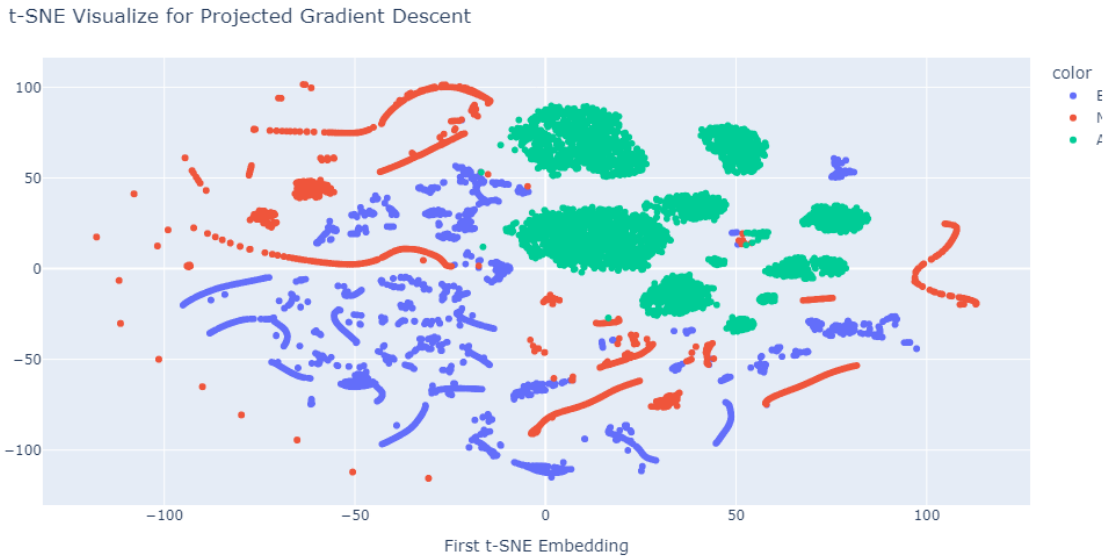
Bảng 4.2: *Kết quả đánh giá của mô hình học máy cho tập dữ liệu InSDN*

Algorithm	**Dataset**	**Accuracy**	**Precision**	**Recall**	**F1-Score**
MLP	Original	0.9993	0.9993	0.9998	0.9996
	FGM	0.5863	1	0.5863	0.7392
	PGD	0.5189	1	0.5189	0.6863
	HSJ	0.4357	1	0.4357	0.6128
CNN	Original	0.9995	0.9995	0.9999	0.9997
	FGM	0	0	0	0
	PGD	0	0	0	0
	HSJ	0	0	0	0

Từ bảng kết quả, ta thấy rằng khi áp dụng tấn công đối kháng vào các mô hình học sâu, độ chính xác của mô hình MLP giảm còn khoảng một nửa so với ban đầu cụ thể là 58% cho FGM, 51% cho PGD và % cho HSJ, đặc biệt ở mô hình CNN khi mà độ chính xác của mô hình gần xấp xỉ bằng 0 ở cả 3 loại tấn công, cho thấy khả năng chống lại trước các cuộc tấn công đối kháng của mô hình gần như không có.

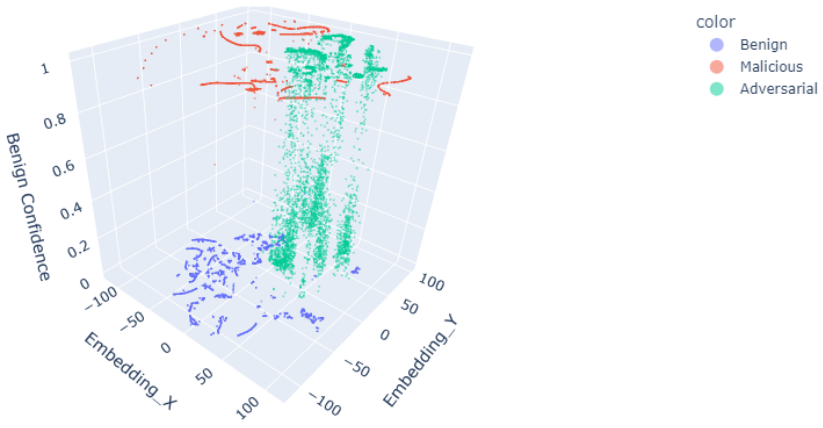
4.2.3. Trực quan hóa dữ liệu

Để thực hiện đánh giá sự khác biệt của các các mẫu dữ liệu trước và sau khi thực hiện tấn công đối kháng, nhóm sử dụng kĩ thuật t-distributed Stochastic Neighbor Embedding giảm số chiều của dữ liệu. Nhóm thực hiện chọn ra 5000 mẫu dữ liệu Thông thường (màu xanh biển), 5000 mẫu dữ liệu Tấn công (màu đỏ) và 5000 mẫu dữ liệu tấn công đối kháng (màu xanh lá) được sinh ra từ tập tấn công với mục tiêu đánh lừa mô hình học sâu phân loại thành tập dữ liệu thông thường. Ở đây nhóm thực hiện chọn ra 1 hình thức tấn công đối kháng là Projected Gradient Descent để trực quan dữ liệu:



Hình 4.2: *Trực quan hóa dữ liệu trong 2 chiều t-SNE với 3 lớp Benign, Malicious và Adversarial*

Dựa vào hình 4.2 ta có thể thấy rằng các tập dữ liệu được phân thành các cụm khác nhau Ngoài ra, nhóm cũng bổ sung mô hình hóa dữ liệu dưới dạng 3 chiều trong đó ở cột z là mức độ tự tin trong phân loại các mẫu lành tính. Ở đây khi xây dựng mô hình phân lớp, ở lớp cuối cùng nhóm sử dụng activation function là sigmoid với đầu ra là phân phối xác xuất của các lớp. Độ tự tin càng cao thì mô hình càng dự đoán đó là mẫu lành tính (Benign). Một mẫu được phân loại là lành tính nếu độ tự tin trên 0.5 và ngược lại



Hình 4.3: *Trực quan hóa dữ liệu trong 3 chiều t-SNE với 3 lớp Benign, Malicious và Adversarial với cột z là độ tự tin ở phân lớp cuối của mô hình*

4.2.4. Tỉ lệ phát hiện mẫu đối kháng thành công khi tích hợp mô hình khả diễn giải

Kết quả phát hiện mẫu đối kháng bằng cách so sánh top 10 đặc trưng với các đặc trưng trong whitelist được thể hiện ở hình 4.4

Bên cạnh đó, nhóm kiểm tra thêm về tỉ lệ dương tính giả (False Positive Rate) của các whitelist được trích xuất với số lượng đặc trưng khác nhau, kết quả được thể hiện ở hình 4.5

Từ 2 kết quả trên nhóm thấy được, nếu số lượng đặc trưng trong whitelist giảm thì khả năng phát hiện mẫu đối kháng tăng nhưng tỉ lệ dương tính giả sẽ tăng cao. Như vậy sẽ có trade off giữa tỉ lệ dương tính giả và khả năng phát hiện mẫu đối kháng.

Base model	Number of alerts	Detection rate (%)	Algorithms	Number of adv samples	Number of features in whitelist
CNN	100	100	Hop Skip Jump	100	24
	100	100	PGD		
	98	98	FGM		
	99	99	Hop Skip Jump		38
	99	99	PGD		
	97	97	FGM		
MLP	956	95.6	Hop Skip Jump	1000	51
	991	99.1	PGD		
	997	99.7	FGM		
	345	34.5	Hop Skip Jump		59
	729	72.9	PGD		
	902	90.2	FGM		

Hình 4.4: *Kết quả phát hiện mẫu đối kháng bằng whitelist*

Base model	X	N	WHITELIST (top N most important features selected from explanation of X 'Normal' data in InSDN Trainset)	False Positive	FP rate (%)
CNN	150	24	URG Flag Cnt', 'Bwd Pkts/s', 'FIN Flag Cnt', 'Bwd PSH Flags', 'SYN Flag Cnt', 'PSH Flag Cnt', 'ACK Flag Cnt', 'Down/Up Ratio', 'Flow Pkts/s', 'Init Bwd Win Byts', 'Bwd IAT Max', 'Bwd IAT Tot', 'Bwd URG Flags', 'Flow Duration', 'Idle Min', 'Bwd Pkt Len Max', 'Fwd IAT Max', 'Flow IAT Std', 'Fwd IAT Mean', 'Flow IAT Max', 'Tot Fwd Pkts', 'Bwd Pkt Len Mean', 'Bwd IAT Std', 'Idle Max'	24	16
		38	URG Flag Cnt', 'Bwd Pkts/s', 'FIN Flag Cnt', 'Bwd PSH Flags', 'SYN Flag Cnt', 'PSH Flag Cnt', 'ACK Flag Cnt', 'Down/Up Ratio', 'Flow Pkts/s', 'Init Bwd Win Byts', 'Bwd IAT Max', 'Bwd IAT Tot', 'Bwd URG Flags', 'Flow Duration', 'Idle Min', 'Bwd Pkt Len Max', 'Fwd IAT Max', 'Flow IAT Std', 'Fwd IAT Mean', 'Flow IAT Max', 'Tot Fwd Pkts', 'Bwd Pkt Len Mean', 'Bwd IAT Std', 'Idle Max', 'Tot Bwd Pkts', 'TotLen Fwd Pkts', 'Idle Mean', 'Bwd Seg Size Avg', 'Pkt Len Mean', 'Flow IAT Mean', 'Pkt Len Std', 'TotLen Bwd Pkts', 'Bwd Pkt Len Std', 'Bwd Header Len', 'Subflow Bwd Pkts', 'Fwd Act Data Pkts', 'Fwd Pkt Len Min', 'Pkt Len Min'	6	4
MLP	1500	51	Init Bwd Win Byts', 'Idle Max', 'Bwd Pkt Len Max', 'Idle Mean', 'Tot Fwd Pkts', 'Idle Min', 'Bwd Pkt Len Std', 'Tot Bwd Pkts', 'Bwd Pkt Len Mean', 'TotLen Fwd Pkts', 'TotLen Bwd Pkts', 'Bwd IAT Max', 'Bwd IAT Tot', 'Fwd IAT Max', 'Bwd Seg Size Avg', 'Pkt Len Min', 'SYN Flag Cnt', 'Flow Duration', 'Fwd Pkt Len Std', 'Fwd IAT Tot', 'Bwd IAT Std', 'Flow Pkts/s', 'Down/Up Ratio', 'Fwd IAT Std', 'Fwd Pkt Len Min', 'Fwd Pkt Len Max', 'Subflow Bwd Pkts', 'Bwd Pkt Len Min', 'Flow IAT Mean', 'Flow IAT Min', 'Fwd IAT Min', 'Bwd IAT Mean', 'Bwd IAT Min', 'Bwd Header Len', 'Subflow Bwd Byts', 'Subflow Fwd Byts', 'Fwd Header Len', 'Bwd PSH Flags', 'Idle Std', 'Fwd Pkts/s', 'Fwd Act Data Pkts', 'Pkt Len Max', 'Fwd IAT Mean', 'Active Max', 'Active Std', 'Pkt Len Std', 'Pkt Len Var', 'Subflow Fwd Pkts', 'Flow Byts/s', 'Pkt Size Avg', 'Active Mean'	752	50.13
		59	Init Bwd Win Byts', 'Idle Max', 'Bwd Pkt Len Max', 'Idle Mean', 'Tot Fwd Pkts', 'Idle Min', 'Bwd Pkt Len Std', 'Tot Bwd Pkts', 'Bwd Pkt Len Mean', 'TotLen Fwd Pkts', 'TotLen Bwd Pkts', 'Bwd IAT Max', 'Bwd IAT Tot', 'Fwd IAT Max', 'Bwd Seg Size Avg', 'Pkt Len Min', 'SYN Flag Cnt', 'Flow Duration', 'Fwd Pkt Len Std', 'Fwd IAT Tot', 'Bwd IAT Std', 'Flow Pkts/s', 'Down/Up Ratio', 'Fwd IAT Std', 'Fwd Pkt Len Min', 'Fwd Pkt Len Max', 'Subflow Bwd Pkts', 'Bwd Pkt Len Min', 'Flow IAT Mean', 'Flow IAT Min', 'Fwd IAT Min', 'Bwd IAT Mean', 'Bwd IAT Min', 'Bwd Header Len', 'Subflow Bwd Byts', 'Subflow Fwd Byts', 'Fwd Header Len', 'Bwd PSH Flags', 'Idle Std', 'Fwd Pkts/s', 'Fwd Act Data Pkts', 'Pkt Len Max', 'Fwd IAT Mean', 'Active Max', 'Active Std', 'Pkt Len Std', 'Pkt Len Var', 'Subflow Fwd Pkts', 'Flow Byts/s', 'Pkt Size Avg', 'Active Mean', 'ACK Flag Cnt', 'Flow IAT Max', 'Active Min', 'Pkt Len Mean', 'Fwd Pkt Len Mean', 'Flow IAT Std', 'Fwd Seg Size Avg', 'PSH Flag Cnt'	86	5.73

Hình 4.5: *Tỉ lệ dương tính giả khi sử dụng whitelist*

CHƯƠNG 5. KẾT LUẬN

Ở chương này, chúng tôi đưa ra những kết luận về nghiên cứu và đồng thời đưa ra hướng cải thiện và phát triển.

5.1. Kết luận

Nghiên cứu về chống tấn công đối kháng, đặc biệt là hướng áp dụng IDS dựa trên học máy có nhiều tiềm năng. Tuy vậy, các phương pháp hiện nay chưa có cơ chế giải thích cho những quyết định của IDS, vì vậy còn nhiều khó khăn để hiểu trong thực tế. Để giải quyết vấn đề này, chúng tôi đề ra một hệ thống cải tiến từ hệ thống trước đó với bước thêm mô hình khả diễn giải để thêm một cơ chế phòng thủ cho IDS.

Qua việc xây dựng hệ thống phát hiện xâm nhập kết hợp XAI này, nhóm chúng tôi đã hiểu sâu hơn về các hướng nghiên cứu liên quan, hiểu được các hạn chế để góp phần cải thiện dần. Đồ án này đã đạt được những kết quả sau:

- Phát sinh các mẫu đối kháng bằng tấn công đối kháng thành công.
- Tìm hiểu về mô hình XAI và tích hợp nó vào mô hình phát hiện xâm nhập.
- Trích xuất và sử dụng whitelist để phát hiện thành công các mẫu đối kháng.

Kết quả mà nhóm thu được qua thực nghiệm cho thấy hệ thống tạo mẫu đối kháng hoàn toàn có khả năng tạo ra những mẫu vừa né tránh được mô hình phát hiện. Đồng thời, chúng tôi cũng xây dựng được một hệ thống IDS có hai lớp bảo vệ, có khả năng chống lại tấn công đối kháng nhờ tích hợp mô hình khả diễn giải.

5.2. Hướng phát triển

- Dùng GAN thay cho phương pháp adversarial attack để sinh những mẫu đối kháng tốt hơn.
- Huấn luyện mô hình học sâu với các tập dataset khác để kiểm tra khả năng cũng như cải thiện phương pháp XAI.
- Áp dụng thêm các phương pháp bảo vệ trước các sự tấn công đối kháng như Ensemble Learning, Adversarial Trainning,...

TÀI LIỆU THAM KHẢO

Tiếng Anh:

[1] Jianbo Chen, Michael I. Jordan, and Martin J. Wainwright, *HopSkipJumpAttack: A Query-Efficient Decision-Based Attack*, 2020, arXiv: 1904.02144 [cs.LG].

[2] Mahmoud Said Elsayed, Nhien-An Le-Khac, and Anca D. Jurcut (2020), "InSDN: A Novel SDN Intrusion Dataset", *IEEE Access*, 8, pp. 165263–165284, DOI: 10.1109/ACCESS.2020.3022633.

[3] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy (2014), "Explaining and harnessing adversarial examples", *arXiv preprint arXiv:1412.6572*.

[4] Scott M Lundberg and Su-In Lee, "A Unified Approach to Interpreting Model Predictions", in: *Advances in Neural Information Processing Systems 30*, ed. by I. Guyon et al., Curran Associates, Inc., 2017, pp. 4765–4774, URL: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

[5] Aleksander Madry et al., *Towards Deep Learning Models Resistant to Adversarial Attacks*, 2019, arXiv: 1706.06083 [stat.ML].

[6] Erzhena Tcydenova et al. (2021), "Detection of adversarial attacks in AI-based intrusion detection systems using explainable AI", *Human-Centric Comput Inform Sci*, 11.

[7] Chaoyun Zhang, Xavier Costa-Pérez, and Paul Patras, "Tiki-taka: Attacking and defending deep learning-based intrusion detection systems", in: *Proceedings of the 2020 ACM SIGSAC Conference on Cloud Computing Security Workshop*, 2020, pp. 27–39.

[8] Chaoyun Zhang, Xavier Costa-Perez, and Paul Patras (2022), "Adversarial attacks against deep learning-based network intrusion detection systems and defense mechanisms", *IEEE/ACM Transactions on Networking*, 30 (3), pp. 1294–1311.