HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Lưu Bích Hạnh

NGHIÊN CỨU GIẢI PHÁP PHÁT HIỆN XÂM NHẬP (IDS) DỰA TRÊN CÔNG NGHỆ HỌC MÁY CHO IoT GATEWAY

LUẬN VĂN THẠC SĨ KỸ THUẬT

(*Theo định hướng ứng dụng*)

HÀ NỘI - NĂM 2022

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Lưu Bích Hạnh

NGHIÊN CỨU GIẢI PHÁP PHÁT HIỆN XÂM NHẬP (IDS) DỰA TRÊN CÔNG NGHỆ HỌC MÁY CHO THIẾT BỊ IoT GATEWAY

Chuyên ngành: Kỹ thuật viễn thông
Mã số: 8.52.02.08

LUẬN VĂN THẠC SĨ KỸ THUẬT
(*Theo định hướng ứng dụng*)

NGƯỜI HƯỚNG DẪN KHOA HỌC :
PGS.TS. LÊ HẢI CHÂU

HÀ NỘI - NĂM 2022

LỜI CAM ĐOAN

Em xin cam đoan toàn bộ nội dung trong đề tài luận văn "Nghiên cứu giải pháp phát hiện xâm nhập (IDS) dựa trên công nghệ học máy cho thiết bị IoT gateway" là một công trình nghiên cứu độc lập của riêng em dưới sự hướng dẫn của PGS.TS. Lê Hải Châu. Đồng thời, kết quả nghiên cứu có trong đề tài này là hoàn toàn trung thực và không sao chép dưới mọi hình thức. Trong luận văn của em có sử dụng tài liệu tham khảo, em đã trích dẫn và chú thích rõ ràng. Em xin hoàn toàn chịu trách nhiệm nếu phát hiện có sai sót.

Tác giả luận văn

Lưu Bích Hạnh

LỜI CẢM ƠN

Lời đầu tiên em xin chân thành cảm ơn PGS.TS. Lê Hải Châu cùng toàn thể các thầy cô giáo của khoa Viễn Thông I – Học viện công nghệ bưu chính viễn thông đã giúp đỡ em trong toàn bộ quá trình thực hiện luận văn. Đối với em, đây là một hành trình khó khăn và đầy thử thách, cả trong thời gian học tập cũng như nghiên cứu. Nhưng nhờ sự giúp đỡ và tận tình chỉ dạy của các thầy cô trong quá trình học Thạc sĩ tại trường, em đã có thêm những kiến thức, được tạo nền tảng, dạy cách tư duy, định hướng để em có thể hoàn thành quá trình học tập và thực hiện luận văn này.

Sau cùng, em xin cảm ơn các anh và các bạn học viên của lớp M20CQTE02-B đã luôn sát cánh và đồng hành cùng em trong quá trình học tập tại tại trường và hoàn thành luận văn.

Em xin chân thành cảm ơn!

MỤC LỤC

DANH MỤC CÁC THUẬT NGỮ, CHỮ VIẾT TẮT

Thuật ngữ và viết tắt	Tiếng Anh	Nghĩa tiếng Việt
ACL	Access Control List	Danh sách điều khiển truy cập
AMQP	Advanced Message Queuing Protocol	Giao thức xếp hàng thông tin nâng cao
ANN	Artificial Neural Network	Mạng nơ ron nhân tạo
CNN	Convolutional Neural Network	Mạng Nơ ron tích chập
CoAP	Constrained Application Protocol	Giao thức ứng dụng ràng buộc
CSV	Comma Separated Value	Giá trị được phân tách bằng dấu phẩy
DDS	Data Distribution Service	Dữ liệu phân phối dịch vụ
DoS	Denial of Service	Từ chối dịch vụ
EMS	Event Management System	Hệ thống quản lý sự kiện
FTP	File Transfer Protocol	Giao thức truyền tải tập tin
HIDS	Host-based Intrusion Detection System	Hệ thống phát hiện truy nhập dựa trên máy trạm
HTTP	Hypertext Transfer Protocol	Giao thức truyền tải siêu văn bản
IDS	Intrusion Detection System	Hệ thống phát hiện xâm nhập
IMAP	Internet Message Access Protocol	Giao thức truy cập tin nhắn Internet
IoT	Internet of Things	Vạn vật kết nối internet
KNN	K-Nearest Neighbors	K láng giềng gần nhất
LAN	Local Area Network	Mạng nội bộ
LoRa	Long Range Radio	Giao thức không dây để truyền thông tầm xa
LTE	Long Term Evolution	Tiêu chuẩn truyền thông di động 4G

LTE-M	LTE Cat-M1	Giao thức truyền thông di động băng thông thấp
MEMS	Micro Electro Mechanical Systems	Hệ thống vi cơ điện tử
MQTT	Message Queuing Telemetry Transport	Giao thức truyền thông điệp
NIDS	Network-based Intrusion Detection System	Hệ thống phát hiện xâm nhập mạng
POP	Post Office Protocol	Giao thức nhận email từ máy chủ
RF	Random Forest	Thuật toán học có giám sát trong AI
SMTP	Simple Mail Transfer Protocol	Giao thức truyền tải thư tín đơn giản
SVM	Support Vector Machine	Thuật toán học máy có giám sát
WAN	Wide Area Network	Mạng diện rộng
Wifi	Wireless Fidelity	Hệ thống truy cập Internet không dây
WSN	Wireless Sensor Network	Mạng cảm biến không dây

DANH SÁCH BẢNG

DANH SÁCH HÌNH VẼ

MỞ ĐẦU

Hiện nay, IoT đang ngày càng bùng nổ và được coi là một xu hướng mới. Bên cạnh các tiện ích mà IoT mang lại thì IoT cũng hàm chứa những mối nguy hại. Trong những năm vừa qua, IoT cũng đang dần trở thành mục tiêu hàng đầu của giới hacker trên toàn thế giới. Những hacker ngày càng nguy hiểm, tinh vi, trình độ cao và hoạt động phức tạp hơn với các hành động có tổ chức chuyên nghiệp. Trong khi vấn đề của IoT là ngày càng nhiều công nghệ và thiết bị được thử nghiệm, ứng dụng thực tiễn với mục đích đem lại tiện ích cho con người nhưng chưa tập trung nhiều vào các vấn đề an toàn thông tin. Hệ quả là dẫn tới sự kiểm soát trở nên khó khăn đã vô tình tiếp tay kẻ xấu lợi dụng để thực hiện hành vi xâm nhập trái phép, tấn công vào những thiết bị IoT tiềm ẩn nhiều rủi ro của các cá nhân, tổ chức gây ra những thiệt hại nghiêm trọng về tài sản hữu hình lẫn vô hình [1]. Vì vậy, các giải pháp ứng dụng an ninh và bảo mật cho các thiết bị IoT ngày càng được quan tâm hơn. Các sản phẩm công nghệ IoT ngày càng đa dạng về chất lượng và bùng nổ về số lượng nên hệ thống phát hiện xâm nhập (IDS) hiện đang là một trong những giải pháp được quan tâm hàng đầu hiện nay nhằm bảo vệ linh hoạt, hiệu quả trước vô vàn cuộc xâm nhập trái phép trên Internet nhắm tới các thiết bị IoT.

Ở đây, chúng ta có thể phát hiện ngay lập tức những hành vi truy nhập bất thường khi sử dụng kỹ thuật học máy. Thực hiện bằng cách thiết lập mô hình dựa vào những thuật toán học máy, thuật toán thống kê hoặc mạng Nơ ron nhân tạo. Nhưng vốn dĩ các cuộc tấn công bảo mật bây giờ đều khá phức tạp và khó có thể đoán trước được. Do đó, việc phải tạo ra một hệ thống phát hiện xâm nhập tốt, có tính chính xác cao cũng như có tỷ lệ báo động giả thấp trong quá trình phát hiện xâm nhập còn gặp khá nhiều khó khăn.

Do vậy, với mục tiêu nghiên cứu, tìm hiểu và nắm bắt các giải pháp phát hiện xâm nhập hiệu quả cho các thiết bị IoT gateway, nội dung luận văn tập trung nghiên cứu, xây dựng và thử nghiệm giải pháp phát hiện xâm nhập dựa trên công nghệ học máy cho các thiết bị IoT gateway.

Luận văn được trình bày theo 03 chương với nội dung chính như sau:

- ***Chương 1 - Tổng quan về IoT, IoT gateway và các kỹ thuật phát hiện xâm nhập:*** Giới thiệu tổng quan về công nghệ IoT, khái niệm, vai trò và vị trí của thiết bị IoT gateway, đồng thời trình bày kiến trúc, thành phần và chức năng của các thành phần trong hệ thống IDS cùng khả năng ứng dụng, triển khai các hệ thống IDS trên IoT Gateway.
- ***Chương 2 - Giải pháp phát hiện xâm nhập ứng dụng học máy:*** Giới thiệu tổng quan về giải pháp phát hiện xâm nhập cho IoT gateway, đồng thời trình bày các kỹ thuật học máy cơ bản sử dụng trong phát hiện xâm nhập, mô tả chi tiết tập dữ liệu mẫu, phân tích và lựa chọn thuật toán học máy để hỗ trợ cho việc thực hiện đánh giá hiệu quả khi ứng dụng thuật toán học máy trong phát hiện xâm nhập cho IoT gateway.
- ***Chương 3 – Thử nghiệm hệ thống IDS trên IoT gateway:*** Trình bày mô hình phát hiện xâm nhập trên IoT gateway, xây dựng kiến trúc hệ thống phát hiện xâm nhập cho IoT gateway dựa trên học máy, đồng thời thiết lập thử nghiệm hệ thống IDS ứng dụng giải pháp mạng Nơ ron và thuật toán *Random Forest* từ đó đưa ra kết quả đánh giá thử nghiệm.

CHƯƠNG 1: TỔNG QUAN VỀ IoT, IoT GATEWAY VÀ KỸ THUẬT PHÁT HIỆN XÂM NHẬP

1.1 Giới thiệu chung

1.1.1 Công nghệ IoT

IoT được viết tắt bởi cụm từ *Internet of Things* – Công nghệ Internet vạn vật, mang ý nghĩa kết nối mọi thứ với Internet. Trong đó mọi vật đều sẽ được cung cấp các định danh khác nhau, có khả năng tự động truyền tải dữ liệu qua một mạng lưới không cần thông qua tương tác giữa máy tính với con người hay con người với con người. IoT là công nghệ được phát triển từ sự hội tụ của những công nghệ không dây, hệ thống vi cơ điện tử (MEMS) và Internet. Kevin Ashton – Người sáng lập Trung tâm Auto-ID ở đại học MIT chính là người đưa ra cụm từ này vào năm 1999. Có thể khái quát rằng Internet of Things đề cập đến những thiết bị vật lý ở tất cả mọi nơi có khả năng kết nối với nhau, với Internet để biến mọi thứ trở nên chủ động, thông minh hơn.

Hiện nay chúng ta có thể bắt gặp IoT ở khắp mọi nơi, ví dụ như xe tự lái, nhà thông minh, thiết bị đeo theo dõi sức khỏe. Việc biến những thiết bị vật lý thụ động trở nên thông minh, cho phép chúng giao tiếp theo dữ liệu thời gian thực mà không cần sự tham gia của con người đã giúp hợp nhất Thế giới vật lý và kỹ thuật số một cách tối ưu và hiệu quả. Dù vậy, *Internet of Things* sẽ cần đến một nền tảng giúp vận hành, điều này thúc đẩy các doanh nghiệp công nghệ quyết tâm tạo dựng nền tảng dẫn đầu để trở thành người chiến thắng.

1.1.2 Các thiết bị IoT gateway

Thiết bị được sử dụng để kết nối các thiết bị khác với đám mây hay trung tâm dữ liệu sẽ được gọi là IoT gateway. Gateway ở đây được hiểu là thiết bị với chức năng tổng hợp và xử lý dữ liệu được gửi bởi các thiết bị thông minh khác nhau lọc trước khi gửi lên đám mây.

Các thiết bị giao tiếp với IoT Gateway qua kết nối có dây như LAN, RS-232, RS-485/422... hay sử dụng công nghệ không dây tầm ngắn và tầm xa như: Zigbee, Z-

wave, Bluetooth LE, LoRa, WiFi, LTE và LTE-M để giao tiếp với IoT Gateway. Sau đó, IoT Gateway kết nối với đám mây hoặc WAN thông qua cáp quang WAN hoặc Ethernet LAN.

Hơn nữa, trong một hệ sinh thái có thể có hàng trăm, hàng nghìn thiết bị IoT. Một số lượng lớn dữ liệu IoT sẽ được tạo ra mỗi giây, có thể khiến đám mây quá tải. IoT Gateway lọc và tổng hợp dữ liệu thu thập được thành một giao thức tiêu chuẩn duy nhất để dữ liệu dễ dàng được xử lý trên đám mây cũng như chuyển tới biên để tính toán hiệu quả. Một số giao thức phổ biến mà IoT Gateway hay sử dụng là AMQP, DDS, CoAP, MQTT và WebSocket.

Khi dữ liệu đã được tổng hợp, thu gọn và phân tích có tính toán ở vùng biên thì sẽ giảm thiểu khối lượng dữ liệu cần chuyển tiếp lên đám mây, có thể gây tác động lớn đến thời gian hồi đáp cũng như chi phí đường truyền mạng.

Thêm nữa IoT Gateway còn có thể cung cấp cơ chế bảo mật bổ sung cho mạng IoT và dữ liệu được nó vận chuyển. Vì gateway quản lý thông tin di chuyển theo cả hai chiều, do đó có thể bảo vệ dữ liệu khi di chuyển lên đám mây khỏi bị đánh cắp và hạn chế các thiết bị IoT bị xâm phạm bởi các cuộc tấn công bên ngoài. Các tính năng chính: phát hiện giả mạo, mã hóa, tạo số ngẫu nhiên bằng phần cứng và công cụ mã hóa.

1.1.3 Các vấn đề an toàn thông tin trong IoT

Trong kỷ nguyên Internet kết nối vạn vật hiện nay, thiết bị IoT rất dễ bị tấn công mạng. Do đó nếu gặp phải tấn công sẽ có thể hình thành những lỗ hổng khi tiếp xúc nhiều thiết bị, làm hệ sinh thái bị lộ. Nhằm hỗ trợ an ninh mạng, các thiết bị IoT đều thông qua IoT Gateway như thêm một lớp bảo vệ cho hệ sinh thái. Cổng IoT hỗ trợ giảm số lượng thiết bị được kết nối với Internet, đồng thời cung cấp mã hóa đầu cuối nhằm bảo vệ dữ liệu an toàn khỏi các cuộc tấn công trực tuyến và ngoại tuyến.

a. Những rủi ro an ninh trong IoT

- ***Chưa có một giao thức chung***

Như chúng ta đã biết, mạng Internet dùng để kết nối thiết bị này với thiết bị khác. Khiến chúng giao tiếp với nhau thì cần phải có một hoặc nhiều giao thức

(protocols) đây là ngôn ngữ chung để giải quyết các tác vụ liên quan. Điển hình như HTTP là giao thức phổ biến nhất sử dụng để tải web. Bên cạnh đó còn có thêm FTP để trao đổi file hay SMTP, POP cho e-mail. Đối với những giao thức này, máy chủ web, mail và FTP không phải giao tiếp với nhau nhiều nên chúng hoạt động khá tốt. Nếu cần nói chuyện nhiều hơn với nhau thì chỉ cần có một phần mềm phiên dịch cơ bản ở giữa kết nối là hai bên đã có thể hiểu nhau dễ dàng. Tuy nhiên điều này chưa thực sự đúng với các thiết bị IoT, bởi chúng có khá nhiều mối liên kết với đa dạng các thiết bị khác nhau. Hiện nay vẫn chưa thể có giải pháp tối ưu để giúp các giao thức IoT trao đổi dữ liệu, hạn chế tình huống không hiểu nhau giữa các thiết bị.

- ***Vấn đề bảo mật của các thiết bị và Gateway***

Tính bảo mật của các thiết bị IoT và Gateway cũng là một nỗi lo lắng. Trong tầng mạng, các thiết bị cấp thấp bị hạn chế tài nguyên đã gây nên thách thức an ninh khi truyền dữ liệu trong các mạng IoT.

- ***Không mất dữ liệu***

Những cuộc tấn công và thăm dò DoS, DdoS là các cuộc tấn công tùy ý, gây hại cho dịch vụ và ứng dụng IoT. Đây cũng là một thách thức không nhỏ liên quan đến tính toàn vẹn của dữ liệu, điều này xuất hiện khi hệ thống IoT bị ảnh hưởng bởi các cuộc tấn công giả mạo và tiếng ồn.

- ***Tấn công vật lý***

Các mối đe dọa vật lý và tính xác thực là những điều gây ảnh hưởng không nhỏ đến hệ thống IoT. Những thiết bị IoT ở tầng nhận thức ví dụ là cảm biến, chúng phụ thuộc vào hệ thống bảo mật nên dễ dàng bị tấn công vật lý.

- ***Quyền riêng tư thông tin***

Rủi ro tiếp theo trong an ninh IoT chính là quyền riêng tư. Mọi thiết bị đều có thẻ nhận dạng riêng biệt gồm thông tin, hoạt động cũng như vị trí. Chúng sử dụng loại công nghệ nhận dạng khác nhau do thành phần IoT khác nhau. Do đó, việc quản lý, giám sát những dịch vụ thuộc hệ thống IoT có thể đang vi phạm quyền riêng tư. Việc hệ thống quản lý bị xâm nhập khi không được phép đều sẽ đe dọa đến những thông tin riêng của người dùng.

b. Những mối đe dọa an toàn thông tin

Hiện nay, ngày càng nhiều các thiết bị IoT được sử dụng rộng rãi tại các tổ chức, doanh nghiệp ở các quốc gia trên Thế giới nói chung và ở Việt Nam nói riêng. Tuy nhiên họ mới chỉ nhận thấy mặt tích cực cũng như lợi nhuận khổng lồ của IoT mà chưa có những giải pháp tối ưu để đối đầu với những cuộc tấn công an ninh từ hacker. Tuy bên trong những hệ thống IoT cũng tồn tại những giải pháp bảo mật nhưng nhìn chung chúng đều đơn giản và thô sơ, chưa thực sự phù hợp để giải quyết các cuộc tấn công diện rộng nhằm vào tất cả các lĩnh vực như ngân hàng, hàng không,… của hacker.

Những lỗ hổng bảo mật khiến các hacker xâm nhập vào thiết bị IoT:

- Các thiết bị chỉ được bảo vệ bởi mã hóa cứng hoặc mật khẩu yếu khi kết nối với Internet.
- Các thiết bị thông minh hiện nay rất dễ gặp phải lỗ hổng bảo mật Zero-day, cơ hội để các hacker sinh sôi, nảy nở.
- Việc công khai các CVE của thiết bị IoT như Router cũng gây ảnh hưởng đến uy tín của nhà cung cấp và cả doanh nghiệp.
- Khi lập trình IoT trên Linux việc không biết những thư viện tải về dùng để code cũng sẽ là một vấn đề hết sức nguy hiểm.

Các hacker hiện nay thường chọn tấn công vào các thiết bị IoT như smart TV hay CCTV camera giám sát hoặc các hệ thống thông minh, tự động hóa trong nhà thay vì lựa chọn cài mã độc, tấn công máy tính. Đây cũng là một thách thức vì sẽ rất khó để phát hiện mã độc trong các thiết bị này. Đồng thời, việc một thiết bị bị nhiễm mã độc sẽ rất dễ để phát tán mã độc đến những thiết bị khác, gây nên mạng botnet rộng lớn và mở rộng nhanh chóng.

Có thể thấy việc bảo mật cho các thiết bị IoT không hề dễ dàng vì các yếu tố như công nghệ, kỹ thuật, sự hiểu biết của người dùng,… Việc cập nhật những bản vá mới trên máy tính hay điện thoại thông minh cũng còn khá khó khăn khi thuyết phục người dùng cập nhật.

1.2 Hệ thống phát hiện xâm nhập (IDS)

1.2.1 Giới thiệu chung

Hành vi xâm nhập trái phép là khi tính toàn vẹn, tin cậy của hệ thống thông tin bị xâm nhập, phá bỏ các hàng rào bảo vệ của hệ thống. Điều này có thể xuất phát từ chính bên trong hệ thống mạng nội bộ hoặc mạng internet bên ngoài.

Phát hiện xâm nhập là những giải pháp và kỹ thuật công nghệ được sử dụng để phát hiện ra những hành vi bất thường, đáng nghi nhằm tìm kiếm các mối nguy hại ở hệ thống thông tin.

IDS là một hệ thống có tác dụng giám sát, theo dõi để tìm ra các hành vi đáng ngờ, việc này giúp ngăn chặn hệ thống thông tin không bị xâm nhập trái phép. Mục đích của IDS là tìm ra và ngăn chặn các tấn công gây ảnh hưởng đến tính bảo mật, sự toàn vẹn thông tin của hệ thống. Hơn thế nữa, IDS còn có khả năng phân biệt giữa các cuộc tấn công từ bên ngoài với những cuộc tấn công nội bộ. Hệ thống IDS sẽ thu thập thông từ các nguồn trong hệ thống an ninh sau đó phân tích nhằm phát hiện sự xâm nhập trái phép.

IDS được xem là một công cụ bảo mật hết sức quan trọng, một trong những giải pháp được lựa chọn để bổ sung cho Firewall. Có thể nhận biết hành động khả nghi, xâm nhập trái phép vào hệ thống mạng cùng chức năng theo dõi lưu lượng mạng trong khi tấn công IDS sẽ đưa ra cảnh báo và cung cấp thông tin nhận biết cho hệ thống. Bên cạnh đó, IDS còn có khả năng kết hợp cùng Firewall hoặc giải pháp thứ 3 để phát hiện ra những mã độc hoạt động ở hệ thống máy chủ, hệ thống mạng, từ đó có thể đưa ra giải pháp loại bỏ các mã độc đó. Tại một hệ thống mạng, thường sẽ có các kiểu tấn công như các loại virus, worm độc hại, từ chối dịch vụ, đăng nhập bất hợp pháp, phá hủy thông tin dữ liệu,….

1.2.2 Kiến trúc IDS

Hai hướng triển khai hệ thống IDS là tập trung và phân tán. IDS tích hợp cùng Firewall là hướng tập trung còn khi nhiều hệ thống IDS trong cùng 1 mạng lớn được kết nối với nhau là hướng phân tán.

Sensor và các mẫu (*signatures*) hoạt động theo cơ chế so sánh với mẫu. Đầu tiên, *Sensor* sẽ tiến hành bắt các gói tin và rồi sau đó *Sensor* sẽ đọc nội dung cũng như tiến hành so sánh các xâu có trong đó với các mẫu tín hiệu nhận biết nhằm mục đích tìm ra những cuộc tấn công cho hệ thống. Trong trường hợp nhận thấy nội dung trong gói tin có xâu trùng với mẫu thì lập tức *Sensor* sẽ đánh dấu đây là sự kiện bình thường hoặc có dấu hiệu tấn công để từ đó phát ra cảnh báo. Những tín hiệu phát hiện các cuộc tấn công được tổng hợp lại thành một bộ gọi là *signatures* hay "mẫu". Mẫu ở đây được sinh ra dựa trên kinh nghiệm phòng ngừa và chống lại những cuộc tấn công, các trung tâm nghiên cứu được thành lập để đưa "mẫu" đến cho hệ thống IDS trên toàn Thế giới.

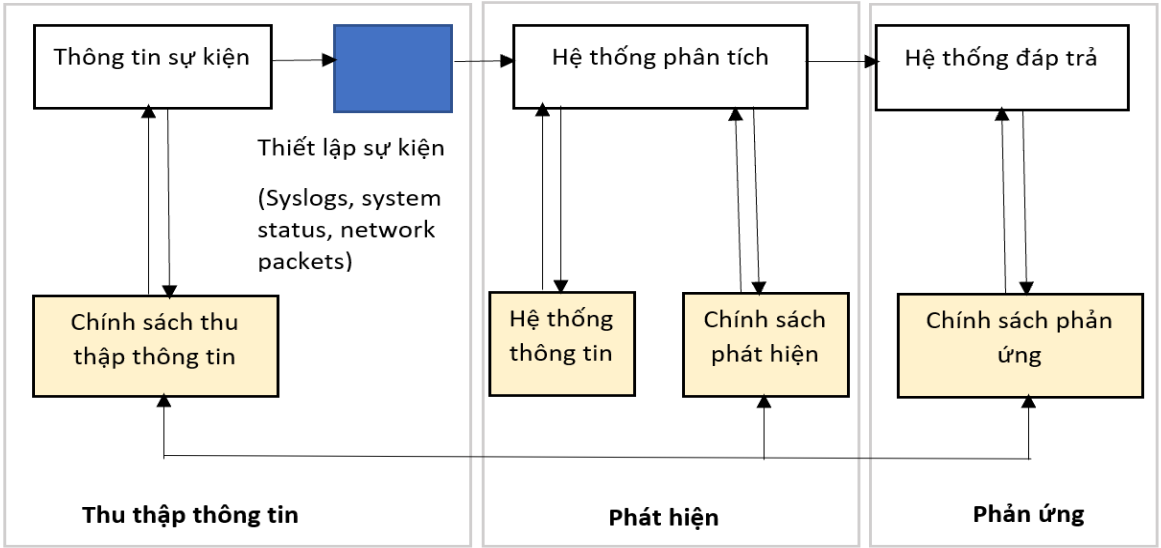
Có 2 loại hệ thống IDS cơ bản như sau:

- HIDS: Phát hiện xâm nhập dựa vào việc sử dụng dữ liệu kiểm tra từ một máy trạm đơn.
- NIDS: Phát hiện xâm nhập dựa vào việc sử dụng dữ liệu tại toàn bộ các lưu lượng mạng và dữ liệu kiểm tra tại 1 hoặc nhiều máy trạm.

1.2.3 Thành phần chính của hệ thống phát hiện xâm nhập

IDS bao gồm 3 thành phần chính sau:

- Thành phần thu thập gói tin
- Thành phần phân tích gói tin
- Thành phần phản hồi: khi gói tin được phát hiện ra là một nguy cơ



Hình 1.1: Các thành phần chính của IDS

Thành phần ở giữa phát hiện, phân tích gói tin là quan trọng nhất và ở thành phần này bộ cảm biến đóng vai trò quyết định. Bộ cảm biến kết nối các thành phần là một bộ tạo sự kiện, sưu tập dữ liệu. Đây là cách sưu tập được xác nhận bằng chính sách tạo sự kiện nhằm nêu lên khái niệm về chế độ lọc thông tin sự kiện. Bộ cảm biến được sử dụng trong việc lọc thông tin, bỏ qua dữ liệu không phù hợp có được từ những sự kiện liên quan đến hệ thống bảo vệ, do đó có thể phát hiện được các hành động đáng ngờ. Bộ phân tích sẽ dùng cơ sở dữ liệu chính sách tìm thấy cho mục này. Đồng thời còn có thêm các thành phần như: dấu hiệu tấn công, profile hành vi thông thường, các tham số cần thiết. Bộ cảm biến cũng có sơ sở dữ liệu riêng.

Chức năng các thành phần trong hệ thống IDS:

- *Sensor/ Agent*: Chức năng phân tích và giám sát hành động. Trong đó *Sensor* thường được dùng cho NIDS còn *Agent* thì được dùng cho HIDS.
- Server quản lý: Đây là một thiết bị trung tâm để thu nhận các thông tin từ *Sensor/ Agent* cũng như quản lý chung.
- Cơ sở dữ liệu: Chức năng để lưu trữ dữ liệu từ Server quản lý và *Sensor/ Agent*.
- Giao diện điều khiển: Chương trình cung cấp giao diện cho IDS, với chức năng giám sát, phân tích và quản trị.
- Hệ thống luật lệ của IDS: Có chức năng định ra các mẫu để so sánh, đối chiếu với dữ liệu ở đầu vào. Và cơ bản tập luật sẽ bao gồm rất nhiều luật, trong đó, mỗi luật gồm 2 thành phần là quy tắc tiêu đều và quy tắc tùy chọn.

Trong đó quy tắc tiêu đề sẽ bao gồm những thông tin:

- Quy tắc hoạt động: Sẽ cho biết các hoạt động được thực hiện khi khớp luật (alert, log, pass, active, dynamic, drop…).
- Giao thức: Cho biết được rằng giao thức sẽ kiểm tra (TCP, UDP, ICMP, IP…)
- Địa chỉ IP: Cung cấp thông tin về địa chỉ IP.
- Số cổng: Cung cấp thông tin về cổng.
- Hướng: Cho biết hướng của dữ liệu thông tin mà đã được khớp.

Và còn quy tắc tùy chọn thì được chia làm những hạng mục:

- Chung: Sẽ mang đến những thông tin chung về luật (msg, reference, rev, classtype…).
- *Payload*: Hỗ trợ tìm kiếm dữ liệu *payload* của gói tin (nội dung, khoảng cách, chiều sâu, *offset,*…).
- *Non-payload*: Hỗ trợ tìm kiếm nội dung *non-payload* của gói tin (*ttl, ack, tos, id, dsize*…).
- Phát hiện sau: Cho biết các phương pháp thực thi tiếp theo (*logto, session, tag*…).

Trong những thành phần của hệ thống IDS thì *Sensor* được coi trọng nhất. Với trách nhiệm tìm ra các xâm nhập nhờ có các cơ cấu ra quyết định cho sự xâm nhập. Kiểm duyệt, kiến thức cơ bản của IDS, thống kê chính là 3 nguồn chính để *Sensor* nhận dữ liệu thô. Đây cũng là những thông tin ảnh hưởng tới quá trình ra quyết định sau này. *Sensor* sẽ được tích hợp với những thành phần chịu trách nhiệm thu gom dữ liệu là máy tạo sự kiện (hệ điều hành, mạng, ứng dụng). Những máy tạo sự kiện này tạo ra chính sách chung cho các sự kiện có thể là *log* hoặc *audit* của các sự kiện trong gói tin hoặc hệ thống. Nó thiết lập cùng những thông tin chính sách được lưu trữ ở bên ngoài hoặc bên trong hệ thống bảo vệ. Hơn nữa tại một số trường hợp khác, dữ liệu sẽ không được lưu trữ mà sẽ được chuyển trực tiếp đến các thành phần phân tích (thường sẽ được áp dụng đối với gói).

1.2.4 Chức năng của IDS

Có thể xác định được mục tiêu của phát hiện xâm nhập là phát hiện được các hành động trái phép đối với IoT. Điều này có thể được gây ra bởi cả người dùng của hệ thống và cả các hacker bên ngoài hệ thống.

Những điều kiện cần phải thỏa mãn của một hệ thống phát hiện xâm nhập:

- ***Hiệu năng***

Đầu tiên phải thỏa mãn về hiệu năng, phải đủ để phát hiện xâm nhập trong thời gian thực. Điều này đồng nghĩa với việc hành động xâm nhập không cho phép phải được phát hiện ra trước khi những tổn hại nghiêm trọng xảy ra.

- ***Tính chính xác***

Phải hạn chế những hành động thông thường nhưng bị coi là bất thường. IDS cần phân biệt được hành động bất thường và các hành động thông thường của hệ thống.

- ***Tính trọn vẹn***

Phải chắc chắn rằng IDS sẽ không bỏ qua một xâm nhập trái phép nào trong hệ thống. Việc bỏ qua những vụ xâm nhập trái phép gọi là âm tính giả.

- ***Khả năng mở rộng***

IDS cần có khả năng không bỏ sót thông tin dù ở trạng thái xấu nhất. Yêu cầu này thường sẽ liên quan trực tiếp đến hệ thống khi có những sự kiện tương quan đến từ những nguồn tài nguyên có số lượng host nhỏ. Còn trong sự phát triển mạnh mẽ của IoT thì hệ thống sẽ gặp phải quá tải do sự gia tăng số lượng lớn những sự kiện.

- ***Chịu lỗi***

Và một yêu cầu hết sức cần thiết chính là bản thân IDS phải có khả năng chống lại được tấn công.

Những chức năng chính của IDS:

- IDS có chức năng giám sát thành phần cần bảo vệ trong hệ thống trước các hoạt động bất thường.
- IDS cần phân tích hành vi truy cập, hoạt động, sự kiện quan trọng liên quan đến thành phần được giám sát dựa vào những hành vi bất thường, tập luật, baseline.
- IDS mang đến những cảnh báo hiểm họa an toàn thông tin. Thay vì dùng những thiết lập mặc định thì cần nâng cao hơn để chống lại kẻ xâm nhập.
- IDS cũng phải thống kê và trích xuất báo cáo.

Những chức năng cơ bản của IDS:

- IDS sẽ cung cấp một cách nhìn tổng thể về lưu lượng mạng.
- IDS cũng giúp nhận diện những hoạt động thâm nhập hay tấn công hệ thống.
- IDS hỗ trợ kiểm tra những sự cố xảy ra trong hệ thống mạng.
- IDS còn được sử dụng để thu gom bằng chứng như *log, event, flow*… cho quá trình điều tra cũng như đối đầu với các sự cố bảo mật.
- IDS sẽ nhận diện được nguy cơ mất an toàn thông tin sẽ xảy ra.

- IDS cũng nhanh chóng phát hiện ra điểm yếu của hệ thống, các lỗ hổng trong chính sách bảo mật.

1.3 Phát hiện xâm nhập trong hệ thống IoT

1.3.1 Kiến trúc chung

Kiến trúc chung của IoT được chia thành năm lớp bao gồm ba miền, đó là miền ứng dụng, miền mạng và miền vật lý. Do đó, IoT có thể được tùy chỉnh để phù hợp với nhu cầu của các môi trường thông minh khác nhau. Miền ứng dụng bao gồm quản lý và sử dụng. Miền mạng chịu trách nhiệm truyền dữ liệu. Miền vật lý chịu trách nhiệm thu thập thông tin.

Tầng nhận thức là tầng phần cứng bao gồm các cảm biến và các đối tượng vật lý ở các dạng khác nhau. Các yếu tố phần cứng này cung cấp nhận dạng, lưu trữ thông tin, thu thập thông tin và xử lý thông tin. Đầu ra thông tin từ lớp này được gửi đến tầng tiếp theo (tầng mạng) để được truyền đến hệ thống xử lý.

Tầng mạng là tầng giúp truyền thông tin từ các đối tượng vật lý hoặc cảm biến đến hệ thống xử lý qua các đường dây an toàn bằng hệ thống truyền thông. Hệ thống liên lạc này có thể là có dây hoặc không dây và có thể dựa trên các công nghệ khác nhau, tùy thuộc vào đối tượng vật lý hoặc các thành phần cảm biến. Đầu ra thông tin từ lớp này được gửi đến tầng tiếp theo (lớp trung gian).

Tầng phần mềm trung gian chịu trách nhiệm quản lý dịch vụ trên các thiết bị IoT để tạo kết nối giữa các thiết bị IoT cung cấp cùng một dịch vụ. Hơn nữa, tầng phần mềm trung gian lưu trữ thông tin đến từ tầng mạng trong cơ sở dữ liệu để tạo điều kiện cho việc ra quyết định trên cơ sở các hoạt động xử lý thông tin.

Tầng ứng dụng chịu trách nhiệm quản lý toàn cầu các ứng dụng IoT. Tầng ứng dụng phụ thuộc vào thông tin được xử lý trong tầng phần mềm trung gian. Hơn nữa, tầng ứng dụng phụ thuộc vào chi tiết cụ thể của các ứng dụng IoT được triển khai khác nhau, chẳng hạn như ngành công nghiệp thông minh, tòa nhà, thành phố và ứng dụng y tế.

Tầng nghiệp vụ cũng chịu trách nhiệm quản lý toàn cầu các ứng dụng IoT cũng như quản lý dịch vụ trên các thiết bị IoT. Tầng nghiệp vụ tạo ra một mô hình phụ thuộc vào thông tin được xử lý trong tầng ứng dụng và phân tích kết quả của các hoạt động xử lý thông tin này.

1.3.2 Môi trường thông minh

Thuật ngữ thông minh dùng để chỉ khả năng thu nhận và áp dụng kiến thức một cách tự chủ, thuật ngữ môi trường dùng để chỉ môi trường xung quanh. Một thành phố thông minh là một loại môi trường thông minh. Yếu tố cốt lõi của một thành phố thông minh là một trung tâm thông tin tích hợp được vận hành bởi nhà cung cấp dịch vụ IoT, nơi cung cấp thông tin về các dịch vụ như điện, nước và gas.

Sức khỏe thông minh, công nghiệp thông minh, tòa nhà thông minh và nhà thông minh là những loại môi trường thông minh khác. Mục tiêu của các môi trường thông minh như vậy là cung cấp dịch vụ thông qua các phương thức thông minh dựa trên thông tin được thu thập bởi các cảm biến hỗ trợ IoT. Kiến trúc của các môi trường thông minh dựa trên IoT.

Các môi trường thông minh dựa trên mô hình IoT có một số đặc điểm nhất định và do đó, các nhu cầu đặc biệt nảy sinh trong việc triển khai các môi trường như vậy. Ví dụ, cần có khả năng giám sát và điều khiển từ xa để cho phép các đối tượng thông minh thu thập và xử lý dữ liệu và thực hiện các hoạt động từ xa. Hơn nữa, khả năng đưa ra quyết định là một đặc điểm quan trọng trong một hệ thống như vậy. Một đối tượng thông minh sẽ có thể đưa ra quyết định thông minh mà không cần sự can thiệp của con người bằng cách sử dụng khai thác dữ liệu và các kỹ thuật khác để trích xuất dữ liệu hữu ích.

Nhờ những đặc điểm này, môi trường thông minh cung cấp một số tính năng nhất định có thể được sử dụng để nâng cao chất lượng dịch vụ (QoS) của các ứng dụng người dùng. Thông tin thời gian thực là một trong những tính năng này. Các đối tượng thông minh có thể thu thập và phân tích dữ liệu và đưa ra quyết định thông minh trong thời gian thực. Hơn nữa, hiệu quả chi phí của các ứng dụng đám mây có

thể được sử dụng để tăng QoS của các ứng dụng môi trường thông minh. Việc tích hợp môi trường thông minh và IoT mang đến những cơ hội mới liên quan đến QoS của các dịch vụ và ứng dụng.

Mục tiêu của môi trường thông minh là làm cho cuộc sống của con người thoải mái và hiệu quả hơn bằng cách sử dụng các cảm biến. Các môi trường thông minh dựa trên IoT cho phép thực hiện hiệu quả các đối tượng thông minh. Bằng mạng IoT, các cảm biến có thể được theo dõi và điều khiển từ xa. Mô hình Internet of Things (IoT) gần đây đã phát triển thành một công nghệ để xây dựng môi trường thông minh. Môi trường thông minh bao gồm các cảm biến làm việc cùng nhau để thực hiện các hoạt động được chỉ định. Cảm biến không dây, công nghệ giao tiếp không dây và IPv6 hỗ trợ mở rộng môi trường thông minh. Những môi trường như vậy rất đa dạng, từ thành phố thông minh và nhà thông minh đến dịch vụ chăm sóc sức khỏe thông minh... Việc tích hợp hệ thống IoT và môi trường thông minh giúp các tiện ích thông minh hoạt động hiệu quả hơn. Tuy nhiên, các hệ thống IoT dễ bị tấn công bảo mật khác nhau, chẳng hạn như các cuộc tấn công từ chối dịch vụ (DoS) và tấn công từ chối dịch vụ phân tán (DDoS). Các cuộc tấn công như vậy có thể gây ra thiệt hại đáng kể cho các dịch vụ IoT và các ứng dụng môi trường thông minh trong mạng IoT. Do đó, việc bảo mật các hệ thống IoT đã trở thành mối quan tâm chính.

Vấn đề bảo mật thông tin và quyền riêng tư được coi là vấn đề chính trong bất kỳ môi trường thông minh nào trong thời đại IoT. Các lỗ hổng bảo mật trong hệ thống IoT tạo ra các mối đe dọa bảo mật ảnh hưởng đến các ứng dụng thông minh. Do đó, rất cần các hệ thống phát hiện xâm nhập (IDS) được thiết kế cho môi trường IoT để giảm thiểu các cuộc tấn công khai thác lỗ hổng bảo mật liên quan. Tuy nhiên, khả năng tính toán và lưu trữ hạn chế của các thiết bị IoT và các giao thức cụ thể được sử dụng, IDS thông thường có thể không phải là một lựa chọn cho môi trường IoT. Hệ thống phát hiện xâm nhập (IDS) là một cơ chế bảo mật hoạt động chủ yếu trong lớp mạng của hệ thống IoT. Phần này trình bày khảo sát toàn diện về các IDS mới nhất được thiết kế cho mô hình IoT, với trọng tâm là các phương pháp, tính năng và cơ

chế tương ứng. Mặc dù các nghiên cứu trước đây về thiết kế và triển khai IDS cho mô hình IoT, phát triển IDS hiệu quả, đáng tin cậy cho môi trường thông minh dựa trên IoT vẫn là một nhiệm vụ quan trọng.

IDS được triển khai cho hệ thống IoT sẽ có thể phân tích các gói dữ liệu và tạo phản hồi trong thời gian thực, phân tích các gói dữ liệu trong các lớp khác nhau của mạng IoT với các ngăn xếp giao thức khác nhau và thích ứng với các công nghệ khác nhau trong môi trường IoT. IDS được thiết kế cho môi trường thông minh dựa trên IoT sẽ hoạt động trong các điều kiện nghiêm ngặt về khả năng xử lý thấp, phản hồi nhanh và xử lý dữ liệu khối lượng lớn. Do đó, IDS thông thường có thể không hoàn toàn phù hợp với môi trường IoT. Bảo mật IoT là một vấn đề liên tục và nghiêm trọng. Do đó, cần phải có sự hiểu biết cập nhật về các lỗ hổng bảo mật của các hệ thống IoT và phát triển các phương pháp giảm thiểu tương ứng. Các yếu tố quan trọng ảnh hưởng đến hiệu suất IDS trong môi trường thông minh, như độ chính xác phát hiện, tỷ lệ dương tính giả, tiêu thụ năng lượng, thời gian xử lý và chi phí hiệu năng. IDS truyền thống không thể đáp ứng các yêu cầu bảo mật IoT do sự đa dạng lớn của các mạng và giao thức IoT. Chẳng hạn, IPv6 trên các mạng khu vực cá nhân không dây công suất thấp (6LoWPAN) không phải là một giao thức được sử dụng trong các mạng viễn thông truyền thống. Thứ ba, các tính năng phổ biến có thể được chuyển từ IDS truyền thống sang IDS dựa trên IoT được nhấn mạnh.

1.4 Ứng dụng giải pháp phát hiện xâm nhập trên IoT gateway

1.4.1 Các kỹ thuật phát hiện xâm nhập

Có hai kỹ thuật phát hiện xâm nhập cơ bản là: Hệ thống phát hiện xâm nhập dựa trên dấu hiệu và hệ thống phát hiện xâm nhập dựa trên bất thường. Nhìn chung, cả hai kỹ thuật trên đều hoạt động theo một quy trình chung sau:

- Một máy chủ tạo ra một gói tin mạng
- Trước khi gói tin được gửi đi, các cảm biến trong mạng sẽ phải đọc trước. Lưu ý cần đặt cảm biến sao cho nó phải đọc được tất cả gói tin.

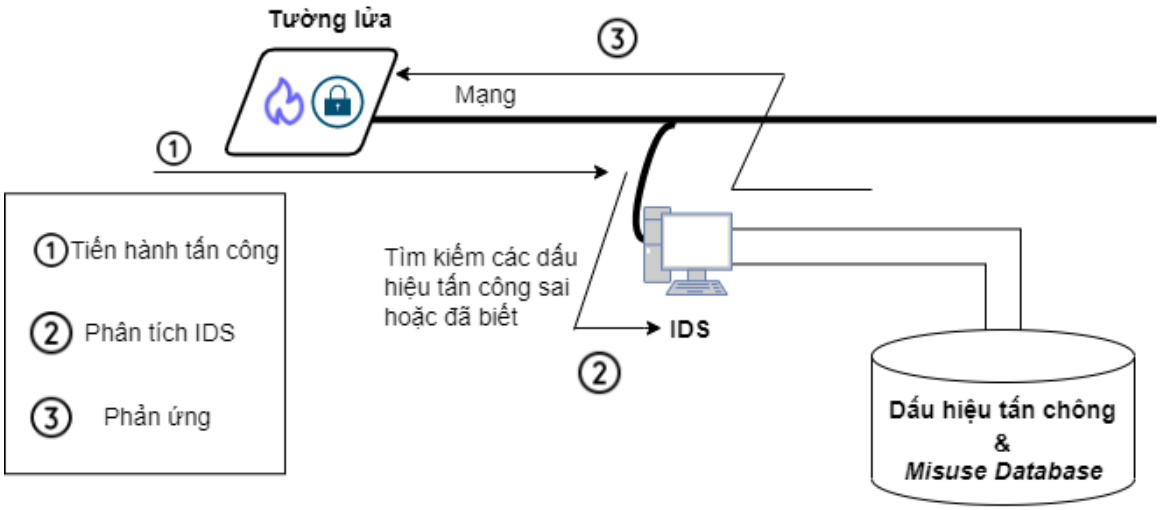
- Chương trình phát hiện sẽ nằm bên trong bộ cảm biến, nó kiểm tra các gói tin để phát hiện vi phạm. Nếu nhận ra có dấu hiệu vi phạm thì ngay lập tức cảnh báo sẽ được phát đi đến giao diện điều khiển.
- Tại giao diện điều khiển, khi nhận được cảnh báo sẽ gửi ngay thông báo cho người hoặc một nhóm có sẵn từ trước (qua pop-up, email,…).
- Thông báo phản hồi sẽ được tạo theo quy định ứng với các dấu hiệu xâm nhập.
- Những cảnh báo cũng sẽ được lưu lại trong cơ sở dữ liệu để tham khảo.
- Đồng thời những cảnh báo này được đối chiếu với các dữ liệu khác đế xác định xem rằng liệu có xảy ra tấn công hay không.

a. Kỹ thuật phát hiện xâm nhập dựa vào dấu hiệu

Phát hiện xâm nhập dựa trên dấu hiệu sử dụng một cơ sở dữ liệu gồm: chữ kí, mô hình mã độc và xâm nhập đã biết để phát hiện ra các cuộc tấn công nổi tiếng. Ba nhược điểm phổ biến của IDS dựa trên dấu hiệu là: quá tải gói mạng, chi phí khớp chữ ký cao và số lượng báo động sai lớn. Bên cạnh đó, việc bị hạn chế bộ nhớ nghiêm trọng trong một vài mạng như WSN, có thể dẫn đến hiệu suất thấp của IDS (dựa trên nhu cầu cần lưu trữ cơ sở dữ liệu lớn về chữ ký tấn công).

Đồng thời, cần phải cập nhật liên tục IDS phù hợp với mẫu, cơ sở dữ liệu chữ ký và mẫu trong IDS dựa vào chữ ký. IDS dựa trên dấu hiệu này được thiết kế nhằm phát hiện các cuộc tấn công, xâm nhập độc hại dựa vào kiến thức trước đó.

IDS là một hệ thống tự giám sát trong thời gian thực. Việc xem lại các thiết lập giám sát sẽ giúp nhận ra các lỗi bảo mật và những cuộc tấn công trực tiếp đang ảnh hưởng đến hệ thống mạng cũng như là một máy chủ nhất định. Để IDS phát hiện ra các tấn công hoặc xâm nhập trái phép thì có hai phương thức cơ bản là: Phát hiện dựa vào dấu hiệu và phát hiện dựa vào sự bất thường. Trong Hình 1.4 phát hiện dựa vào dấu hiệu sẽ so sánh những tình huống thực tế với các dấu hiệu tấn công (*signatures*) đã biết (*signatures* này được lưu trữ trong cơ sở dữ liệu của IDS).



Hình 1.2: Quy trình hoạt động chung của kỹ thuật phát hiện xâm nhập dựa vào dấu hiệu

b. Kỹ thuật phát hiện xâm nhập dựa vào sự bất thường

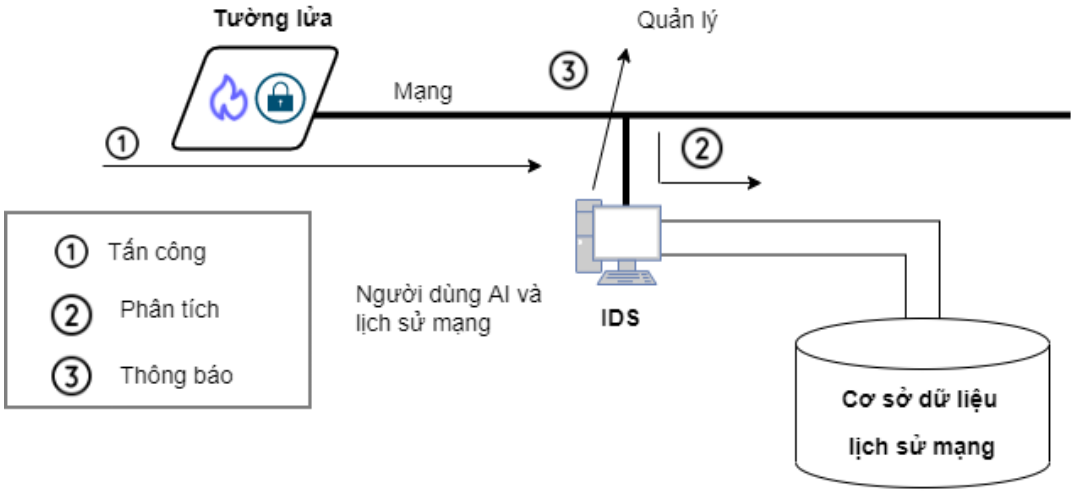
Đối với kỹ thuật phát hiện xâm nhập dựa vào sự bất thường, mẫu dữ liệu sẽ được tạo ra dựa trên dữ liệu của người dùng bình thường và sau đó đem so sánh với các mẫu dữ liệu đang có để phát hiện ra điều bất thường nếu có. Các hành động bất thường sẽ được phát hiện cho việc xác định những cuộc tấn công, đặc biệt là với các cuộc tấn công chưa biết.

IDS dựa vào sự bất thường sẽ hoạt động bằng cách tạo ra một mô hình hành vi bình thường trong môi trường máy tính đã được cập nhật liên tục và dựa trên dữ liệu của người dùng bình thường, sau đó sử dụng mô hình này để phát hiện ra bất cứ sai sót nào so với các hành vi bình thường.

Phát hiện xâm nhập dựa vào sự bất thường sẽ phân tích dựa trên các thuật toán cao cấp và cũng sẽ hoạt động tùy vào môi trường, hỗ trợ phát hiện những biến cố bất thường. Đồng thời, dựa vào các hoạt động thường nhật của hệ thống để tự động phát hiện ra những điều bất thường và dò tìm xem nó sẽ thuộc vào dạng tấn công nào.

Phương pháp khai thác dữ liệu là phương tiện có thể trích xuất thông tin từ một lượng lớn dữ liệu. Mô hình như thế dùng để mô tả hoạt động của dữ liệu từ người dùng hoặc từ mạng máy tính. Đồng thời cách tiếp cận này còn sử dụng được trong IDS tổng quan cũng như trong các môi trường điện toán khác. Phương pháp này hoạt

động rất tốt cho luồng dữ liệu trực tuyến không phụ thuộc, hoạt động liên tục và có sự phát triển về khối lượng.



Hình 1.3: Quy trình hoạt động chung của kỹ thuật phát hiện xâm nhập dựa vào sự bất thường

Giai đoạn đào tạo và giai đoạn phát hiện là hai giai đoạn quan trọng của kỹ thuật học máy. Trong đó, giai đoạn đào tạo sẽ bị phụ thuộc bởi những hàm toán học hoặc thuật toán sử dụng dữ liệu thường làm đầu vào tham chiếu để tìm ra các đặc tính của môi trường IoT. Tiếp theo đó ở giai đoạn phát hiện, các đặc tính này sẽ được dùng để phát hiện và phân loại bất thường. Học có giám sát là kỹ thuật học máy mà ở đó các đặc tính của tập dữ liệu huấn luyện được dùng trong giai đoạn học nhằm tạo thành mô hình phân loại, cái này sẽ dùng để phân loại các trường hợp khó phát hiện mới. Còn học không giám sát là kỹ thuật học máy chỉ cần có các tính năng của dữ liệu mà không cần dùng đến dữ liệu đào tạo phân cụm.

Một phương pháp phân loại đơn chỉ cần phụ thuộc vào một thuật toán học máy đơn thuần. Còn đối với phương pháp phân loại mẫu trong học máy thì sẽ phụ thuộc vào nhận dạng mẫu.

Phương pháp mô hình thống kê sẽ phụ thuộc vào những phép toán thống kê. Và việc thống kê hành vi lịch sử người dùng thường được dùng tạo mô hình bình thường, từ đó mọi sai lệch so với mô hình này sẽ được phát hiện là dữ liệu bất thường. Phương pháp mô hình thống kê sử dụng các phép toán thống kê và áp dụng cho tập

dữ liệu huấn luyện sử dụng để phát hiện lưu lượng truy cập bất thường từ các mẫu lưu lượng được quan sát trước đó.

Phương pháp mô hình quy tắc sẽ bị phụ thuộc bởi việc tạo ra các quy tắc cho môi trường điện toán. Các quy tắc này cũng được trích xuất từ những mẫu lưu lượng dữ liệu. Theo đó, IDS dựa trên mô hình quy tắc phát hiện các lưu lượng dữ liệu bất thường sẽ phá vỡ những quy tắc này, sau đó sẽ coi bất kỳ sự bất thường nào đó cũng như một cuộc xâm nhập. Quá trình tạo quy tắc phụ thuộc nhiều vào lịch sử hành vi trên hệ thống. Vì vậy, hệ thống cần phải được theo dõi trong một thời gian dài để tránh phần trăm dương tính giả mạo quá cao.

Cách tiếp cận mô hình tải trọng dựa vào lưu lượng gói của một cổng hoặc người dùng cụ thể cho một ứng dụng. Trong IDS dựa trên chữ ký, mô hình tải trọng sẽ phụ thuộc vào việc khớp mẫu để xác định các gói tấn công có những đặc điểm cụ thể. Nhưng ngược lại, IDS dựa trên sự bất thường sẽ phải sử dụng phương pháp mô hình tải trọng tạo ra một mô hình phụ thuộc vào byte hoặc tính toán từ những byte mô tả các đặc điểm bình thường của *payload*.

Cách tiếp cận mô hình giao thức dựa vào những giao thức giám sát trong các lớp khác nhau ở môi trường máy tính. IDS dựa vào phương pháp này sẽ phát hiện sự bất thường liên quan đến một giao thức cụ thể hay một giao thức không có trong mô hình bình thường. Để phân tích các giao thức trong môi trường điện toán có thể sử dụng cách tiếp cận dựa trên đặc tả, dựa trên trình phân tích cú pháp hoặc dựa trên các từ khóa giao thức ứng dụng.

Ưu và nhược điểm của các kỹ thuật phát hiện xâm nhập dựa trên sự bất thường ứng dụng trong IDS.

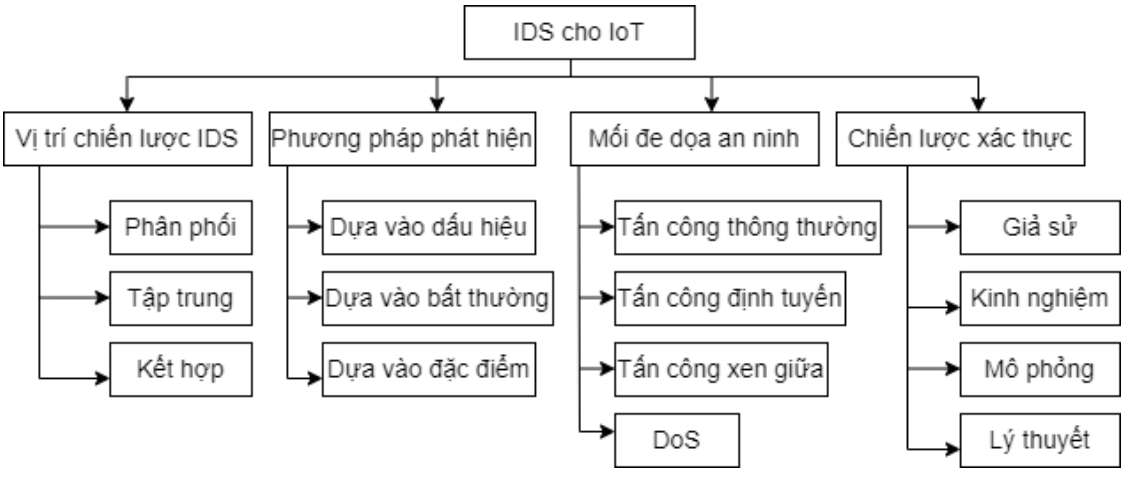
Bảng 1.1: Các kỹ thuật được ứng dụng trong IDS

Kỹ thuật	Ưu điểm	Nhược điểm
Khai thác dữ liệu	Mô hình được tạo tự động và ứng dụng trong các môi trường khác nhau	Phụ thuộc vào những thuật toán phức tạp và dựa trên dữ liệu lịch sử
Học máy	Phát hiện có độ chính xác cao, phù hợp với khối lượng dữ liệu lớn và đơn giản hệ thống	Cần dữ liệu đào tạo, thời gian đào tạo lâu, việc phát hiện chính xác phụ thuộc vào toán học và hoạt động thống kê
Mô hình tải trọng	Các cuộc tấn công đã biết có độ phát hiện chính xác cao	Thời gian xử lý lâu và vấn đề về quyền riêng tư
Mô hình giao thức	Phát hiện có độ chính xác cao đối với cuộc tấn công cụ thể	Thiết kế cho một loại giao thức cụ thể

1.4.2 Ứng dụng trong IoT gateway

Vì những thách thức bảo mật mà các hệ thống IoT phải đối mặt mà các phương pháp có thể chủ động xác định được các cuộc tấn công sẽ là lựa chọn phù hợp nhất để bảo vệ mạng IoT nói chung và IoT Gateway nói riêng. Vì vậy cần phải có hệ thống IDS phù hợp, có thể phát hiện ra các cuộc tấn công mới trong IoT.

So sánh IDS được thiết kế cho các hệ thống IoT và tập trung vào những loại kỹ thuật cũng như tính năng của các hệ thống này liên quan đến khả năng phù hợp của hệ thống IDS trong môi trường thông minh dựa trên IoT.



Hình 1.4: Giải pháp phát hiện xâm nhập ứng dụng cho hệ thống IoT

Trong khuôn khổ luận văn, hai hướng tiếp cận hệ thống IDS bao gồm: phân loại IDS dựa trên vị trí chiến lược và phương thức phát hiện sẽ được tập trung mô tả và làm rõ.

a. Vị trí chiến lược

- ***Mô hình tập trung***

Vị trí IDS tập trung sẽ được đặt trong một thành phần tập trung sử dụng để phân tích các gói tin đi qua bộ định tuyến biên nằm ở giữa miền vật lý và miền mạng. Công việc này đơn thuần chỉ là giám sát lưu lượng bộ định tuyến biên. Triển khai các cảm biến IDS trong mạng năng lượng thấp và tổn thất, chịu trách nhiệm truy tìm lưu lượng mạng và gửi dữ liệu này đến cho công cụ phân tích IDS. Máy chủ chuyên dụng IDS là sợi dây kết nối với các cảm biến IDS để tránh cho việc truyền dữ liệu IDS và dữ liệu thông thường của mạng trong cùng một mạng không dây. Vì vậy, nếu một cuộc tấn công DoS làm giảm đi chất lượng truyền không dây, việc truyền dữ liệu IDS cũng sẽ không bị ảnh hưởng.

- ***Mô hình phân tán***

Vị trí IDS phân tán đặt mọi đối tượng vật lý. ở đây sẽ đề xuất hai kỹ thuật như chuyển dịch phụ trợ và quyết định sớm. Mục đích chính của IDS này là để giảm thiểu các tài nguyên tính toán cần thiết sử dụng trong phát hiện xâm nhập. IDS này cũng quản lý năng lượng nút và kiểm soát lưu lượng vào/ra. Nếu IDS có phát hiện ra bất kỳ cuộc tấn công nào thì nó sẽ phát một thông báo nhằm cảnh báo tất cả các nút.

- ***Mô hình lai***

Vị trí IDS lai kết hợp các khái niệm về vị trí tập trung và phân tán vào để tận dụng điểm mạnh và tránh đi phần nhược điểm. Cách tiếp cận đầu tiên là cho vị trí kết hợp tổ chức mạng thành các cụm hoặc vùng, chỉ nút chính của mỗi cụm lưu trữ một thể hiện IDS. Từ đây, nút này trở thành trách nhiệm giám sát các nút khác của cụm. IDS vị trí kết hợp có thể được thiết kế nhằm tiêu thụ nhiều tài nguyên hơn IDS vị trí phân tán. Tại mô hình này, các nút được chọn trong mạng lưu trữ IDS. Các nút được chọn xác định cuộc xâm nhập bằng cách nghe lén các gói trao đổi trong vùng lân cận của chúng. Thành phần giám sát sẽ quyết định liệu một nút có thể bị xâm phạm theo

một bộ quy tắc hay không. Đồng thời, mỗi thành phần giám sát sẽ có một bộ quy tắc riêng vì mỗi thành phần trong mạng có thể có một hành động khác nhau. Ưu điểm của phương pháp này nằm ở việc cho phép xây dựng một bộ quy tắc khác nhau cho từng khu vực trong mạng.

b. Phương thức phát hiện

- ***Dựa vào dấu hiệu***

Trong các cách tiếp cận dựa vào dấu hiệu thì IDS phát hiện các cuộc tấn công dựa vào hành vi của hệ thống hoặc mạng, khớp với chữ ký tấn công đã được lưu ở trong cơ sở dữ liệu của IDS. Nếu bất kỳ hoạt động mạng hoặc hệ thống nào khớp với các mẫu hoặc dấu hiệu được lưu trữ thì cảnh báo sẽ được kích hoạt. Thành phần phát hiện ra có dấu hiệu tấn công sẽ được mô hình hóa giống như các tế bào miễn dịch có khả năng phân loại datagram độc hại (yếu tố không tự) hoặc bình thường (yếu tố tự). Đồng thời, các thành phần phát hiện có thể tiến hóa để thích nghi trong những điều kiện mới và môi trường được theo dõi.

- ***IDS dựa trên sự bất thường***

IDS dựa trên sự bất thường còn được gọi là phát hiện dựa trên sự kiện. Kỹ thuật này sẽ giúp xác định các hoạt động độc hại bằng cách phân tích sự kiện. Đầu tiên cần xác định hành vi bình thường của mạng, nếu có bất kỳ hoạt động nào khác với hành vi bình thường thì dấu hiệu đó là một sự xâm nhập. Đối với phương pháp này, một nút độc hại có khả năng được phát hiện bằng cách khớp với đặc tả giao thức hiện tại và trạng thái giao thức được xác định trước đó. Đồng thời, cách tiếp cận này cũng sẽ phát hiện các cuộc tấn công hiệu quả hơn IDS dựa vào chữ ký bên trên. Nguyên tắc của IDS được đề xuất là tìm kiếm tất cả sự khác biệt có trong mạng bằng cách giám sát các đặc điểm của các nút lân cận như: kích thước gói và tốc độ dữ liệu.

- ***Dựa vào thông số kỹ thuật***

Kỹ thuật dựa vào thông số kỹ thuật sẽ hơi giống với kỹ thuật phát hiện bất thường. Nhưng trong kỹ thuật này, hành vi bình thường của mạng được xác định bằng tay vì thế nó cho tỷ lệ dương ít chính xác hơn. Tuy nhiên kỹ thuật này cũng sẽ cố gắng chọn lọc tốt nhất giữa các phương pháp: phát hiện dựa trên chữ ký và bất thường,

bằng cách cố gắng làm rõ những sai lệch so với các mẫu hành động thông thường được tạo ra bởi dữ liệu đào tạo và phương pháp học máy. Việc phát triển đặc tả tấn công hoặc giao thức cũng được thực hiện bằng tay nên cần mất nhiều thời gian hơn. Đây là một bất lợi của phương pháp này. IDS dựa trên đặc tả sẽ cho phép người quản trị mạng thiết lập quy tắc để phát hiện tấn công. Nếu một trong những quy tắc này bị vi phạm, IDS sẽ gửi cảnh báo lên hệ thống quản lý sự kiện (EMS). Từ đây, EMS chạy trên một nút mà không có bất kì ràng buộc tài nguyên nào để tương quan cảnh báo cho các nút khác nhau có trong mạng.

- ***Phương pháp lai***

Phương pháp này là sự kết hợp của các khái niệm: phát hiện dựa trên dấu hiệu, dựa trên đặc điểm kỹ thuật và bất thường để tối ưu hóa ưu điểm và giảm thiểu đi tác động của nhược điểm.

1.5 Kết luận Chương 1

Chương 1 giới thiệu tổng quan về công nghệ IoT, các thiết bị IoT gateway cũng như các vấn đề an toàn thông tin trong IoT. Đồng thời, nội dung chương cũng tập trung trình bày về những yếu tố cản trở và hiểm họa của IoT cũng như khái niệm, kiến trúc và chức năng các thành phần của hệ thống phát hiện xâm nhập, các kỹ thuật nhằm phát hiện xâm nhập trái phép, đưa ra quy trình chung, thành phần, mô hình của các giải pháp.

CHƯƠNG 2: GIẢI PHÁP PHÁT HIỆN XÂM NHẬP ỨNG DỤNG HỌC MÁY

2.1 Giới thiệu chung

Quá trình phát hiện xâm nhập trong IoT sẽ giám sát mọi sự kiện xuất hiện trong hệ thống nhằm phân tích các dấu hiệu để phát hiện ra đâu là bình thường đâu là xâm nhập bất thường. Nhằm ngăn chặn hiệu quả các cuộc tấn công mạng thì hệ thống phát hiện xâm nhập sẽ được triển khai ngay tại các thiết bị IoT gateway. Điều này cũng giúp ích cho việc giảm đi thời gian hệ thống bị đình trệ cũng như các chi phí gây ra bởi những cuộc tấn công mạng. Đối với hệ thống IDS sẽ có hai phương pháp nhằm nhận ra các xâm nhập đó là đối sánh mẫu và phát hiện dựa trên những hành vi bất thường. Trong đó, phương pháp đối sánh mẫu sẽ phát hiện ra các cuộc tấn công dựa trên cơ sở dữ liệu chứa các dấu hiệu tấn công, điều này đã được định nghĩa vì thế sẽ có độ chính xác cao cũng như khả năng cảnh báo nhầm ít xảy ra hơn [8]. Mặc dù vậy, phương pháp đối sánh mẫu lại không thể phát hiện ra được những tấn công mới. Bên cạnh đó, phương pháp phát hiện dựa trên dấu hiệu bất thường thì ngược lại, nó có thể phát hiện cuộc tấn công mới nhưng lại cho ra tỷ lệ cảnh báo nhầm khá cao.

Dưới sự phát triển của công nghệ học máy, những giải pháp phát hiện xâm nhập hiện đã và đang được chú trọng nghiên cứu, phát triển cũng như ứng dụng vào thực tế nhằm đóng góp và cải thiện tỷ lệ phát hiện, tính chính xác cũng như giảm đi tối đa số lượng cảnh báo nhầm. Có thể thấy rằng bài toán phân biệt các hành vi truy nhập là bình thường hay bất thường và dùng đến những tài nguyên của hệ thống là một bài toán điển hình trong kỹ thuật học máy. Dựa vào những đặc điểm của từng hành vi thu thập được, hệ thống học máy sẽ dựng lên mô hình tự động phân loại để cho ra kết quả chính xác nhất.

Tại Việt Nam, đã có những nghiên cứu để nâng cao khả năng phát hiện xâm nhập mạng, điển hình là nghiên cứu của tác giả Nguyễn Ngọc Điệp và cộng sự phát hiện xâm nhập dựa trên kỹ thuật học sâu. Nghiên cứu này sử dụng phương pháp phát hiện xâm nhập mạng bằng mạng nơ-ron tích chập CNN và tiền xử lý dữ liệu, thử

nghiệm trên tập dữ liệu NSL-KDD. Nghiên cứu cũng chỉ ra được đặc tính ưu việt trong quá trình học đặc trưng của CNN, giúp mô hình có thể học được các đặc trưng tốt nhất để phân loại các tấn công [10]. Nhằm cung cấp thêm góc độ khác để phát hiện xâm nhập mạng, luận văn sẽ nghiên cứu giải pháp phát hiện xâm nhập (IDS) dựa trên công nghệ học máy cho các thiết bị IoT gateway. Ứng dụng thuật toán học máy Random Forest và mạng Nơ ron để phát hiện xâm nhập dựa trên tập dữ liệu UNSW-NB15. Nghiên cứu phương pháp kết hợp dựa vào dấu hiệu và hành vi bất thường để nâng cao hiệu quả phát hiện xâm nhập nhằm đảm bảo an toàn cho các thiết bị IoT gateway.

IDS được triển khai cho hệ thống IoT sẽ có thể phân tích các gói dữ liệu và tạo phản hồi trong thời gian thực, phân tích các gói dữ liệu trong các lớp khác nhau của mạng IoT với các ngăn xếp giao thức khác nhau và thích ứng với các công nghệ khác nhau trong môi trường IoT. IDS được thiết kế cho môi trường thông minh dựa trên IoT sẽ hoạt động trong các điều kiện nghiêm ngặt về khả năng xử lý thấp, phản hồi nhanh và xử lý dữ liệu khối lượng lớn. Do đó, IDS thông thường có thể không hoàn toàn phù hợp với môi trường IoT.

Bảo mật IoT là một vấn đề liên tục và nghiêm trọng. Do đó, cần phải có sự hiểu biết cập nhật về các lỗ hổng bảo mật của các hệ thống IoT và phát triển các phương pháp giảm thiểu tương ứng. Các yếu tố quan trọng ảnh hưởng đến hiệu suất IDS trong môi trường thông minh, như độ chính xác phát hiện, tỷ lệ dương tính giả, tiêu thụ năng lượng, thời gian xử lý và chi phí hiệu năng. IDS truyền thống không thể đáp ứng các yêu cầu bảo mật IoT do sự đa dạng lớn của các mạng và giao thức IoT.

Một số các biện pháp ngăn chặn xâm nhập được sử dụng khá phổ biến như: tường lửa, mã hóa, xác thực, quyền truy cập,… Mặc dù, bản thân mỗi hệ thống máy tính đều có những cơ chế bảo mật riêng nhằm chống lại và ngăn chặn những xâm nhập trái phép nhưng những giải pháp bảo mật nêu trên chưa đủ mạnh để có thể phát hiện, cảnh báo, ngăn chặn được những cuộc tấn công mới, ngày càng tinh vi hơn. Lợi thế của hệ thống IDS học máy nằm ở độ chính xác phát hiện cao, thích hợp cho khối lượng dữ liệu lớn, đơn giản hệ thống. Nhưng bên cạnh đó cũng có bất lợi như yêu cầu

dữ liệu đào tạo, thời gian đào tạo lâu, độ chính xác phát hiện phụ thuộc vào hoạt động thống kê và toán học.

2.2 Một số kỹ thuật học máy sử dụng trong phát hiện xâm nhập

Có một thứ gọi là định luật "No Free Lunch" trong học máy, cách nói này mang ý nghĩa không có một thuật toán nào là tốt nhất trong mọi vấn đề, do đó việc học dưới sự giám sát (*supervised learning*) sẽ là lựa chọn phù hợp hơn cả. Đồng thời, nên thử nhiều thuật toán khác nhau trong khi dùng một tập kiểm tra còn lại nhằm đánh giá hiệu suất cũng như chọn ra giải pháp tối ưu nhất. Bởi sẽ có rất nhiều yếu tố gây ảnh hưởng, ví dụ như kích thước và cấu trúc của bộ dữ liệu.

Có nhiều vấn đề trong sự phát triển học máy như xử lý lượng dữ liệu lớn, thiếu bộ nhớ, mất cân bằng các kích cỡ giữa các lớp, điều này gây ra khá nhiều trở ngại. Trong đó, dữ liệu không cân bằng sẽ làm cho việc khái quát các lớp dữ liệu trở nên khó khăn hơn.

Hiện nay, ngày càng có nhiều dữ liệu được tạo ra bởi con người, hệ thống thông minh. Đã có môt số phương pháp khai thác dữ liệu được tạo ra để dễ dàng tìm kiếm thông tin từ kho dữ liệu khổng lồ đó. Với kỹ thuật học máy, quy trình tìm kiếm các mẫu kết quả từ việc khai thác dữ liệu là hoàn toàn tự động, đây là phương pháp hữu ích có thể đưa ra dự đoán về dữ liệu mới. Đồng thời, có thể sử dụng trong việc tìm ra dấu hiệu bất thường trong các dữ liệu thông thường.

Đối với dữ liệu mạng Internet có thể là trường hợp tìm ra sự bất thường đe dọa đến quyền riêng tư, bảo mật dữ liệu và cả trong những hoạt động thông thường của hệ thống ngân hàng, nhà máy, hành chính. Đối với dữ liệu mạng nội bộ thì những bất thường này có thể tấn công vào lỗ hổng hệ thống, chứa mã độc hoặc từ chối dịch vụ. Do đó, việc phát hiện sớm và ngăn chặn bất thường cần được ưu tiên cao, đặc biệt là đối với những lỗ hổng không được phát hiện trước khi bị tấn công.

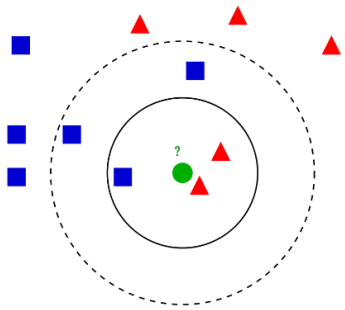
Ứng dụng học máy vào việc phát hiện xâm nhập hệ thống IoT cũng được coi như một vấn đề thuộc bài toán phân lớp: từ tập dữ liệu huấn luyện được dán nhãn tấn công và bình thường, phân loại các thông tin để phát hiện xâm nhập. Sau đây sẽ là

những khái niệm về các thuật toán học máy được nhiều tổ chức, chuyên gia bảo mật nghiên cứu và ứng dụng cho phát hiện xâm nhập trong hệ thống IoT.

2.2.1 K-Nearest Neighbors

K-Nearest Neighbors algorithm (K-NN) còn được gọi là thuật toán *lazy learning* do khi huấn luyện nó không học gì từ dữ liệu huấn luyện. K-NN cũng được xem như một phương pháp để phân lớp các đối tượng theo khoảng cách gần nhất giữa đối tượng cần xếp lớp (*Query point*) với tất cả những đối tượng khác trong dữ liệu đào tạo. Ở đây K-NN còn được xem như một thuật toán phi tham số bởi K-NN sẽ không mang lại những dự đoán về cấu trúc của dữ liệu [2].

KNN được sử dụng để áp dụng vào hai loại của bài toán học có giám sát, đó là phân lớp, hồi quy. Có thể thấy, kết quả dự đoán của một điểm dữ liệu mới sẽ được chỉ ra trực tiếp từ k điểm dữ liệu gần nhất tại tập dữ liệu huấn luyện.



Hình 2.1: Thuật toán K-NN

Mô tả thuật toán K-NN:

- Xác định giá trị tham số K (số *neighbors* gần nhất).

- Tính khoảng cách giữa đối tượng cần phân lớp (*Query Point*) với tất cả những đối tượng trong training data (thường sử dụng khoảng cách *Euclidean*).

- Xếp thứ tự khoảng cách tăng dần và xác định K láng giềng gần nhất với *Query Point*.

- Lấy toàn bộ các lớp của K láng giềng gần nhất đã xác định.

- Việc xác định lớp cho *Query Point* sẽ phụ thuộc vào các lớp của láng giềng gần nhất.

- Trong các bài toán phân lớp, kết quả đầu ra sẽ là lớp mà dữ liệu thuộc về và phụ thuộc vào việc bình chọn của k điểm gần nhất. Đồng thời, trong bài toán hồi quy thì đầu ra của một điểm dữ liệu sẽ bằng trung bình đầu ra của k điểm gần nhất.

Những phương pháp đo khoảng cách giữa các điểm để tìm ra điểm gần nhất phổ biến bao gồm khoảng cách *Hamming*, khoảng cách *Manhattan*, khoảng cách *Minkowski*.

Trong những bài toán phân lớp có thể thấy đầu ra được biểu diễn dưới hình thức tập các xác suất mà mỗi điểm đó thuộc về lớp nào đó. Ví dụ ta có bài toán nhị phân, ở đây xác suất sẽ tính theo công thức

$$P(0) = \frac{N_0}{N_0 + N_1} \quad (2.1)$$

Với P(0) là xác suất của một điểm thuộc lớp 0 và N_0, N_1 là số các điểm lân cận thuộc lớp 0 hoặc 1.

Tính chính xác của thuật toán dự đoán phụ thuộc khá lớn vào giá trị k. Nhưng việc chọn k không phải là một nhiệm vụ đơn giản. Nếu k quá nhỏ thì độ chính xác sẽ giảm, điều này đúng hơn ở những tập dữ liệu chứa nhiều giá trị nhiễu. Khi k quá lớn lại sẽ bị giảm hiệu năng của thuật toán. Không những vậy, khi giá trị k quá lớn còn khiến cho mô hình bị quá tải đồng nghĩa với việc phân cách giữa các lớp cũng trở nên ít khác biệt hơn làm cho độ chính xác cũng bị giảm đáng kể, k thường được chọn theo công thức (2.2) với n là số lớp:

$$k = \sqrt{n} \quad (2.2)$$

Đặc biệt, cần phải chọn k lẻ khi bài toán có số lớp chẵn để tránh xảy ra kết quả hòa.

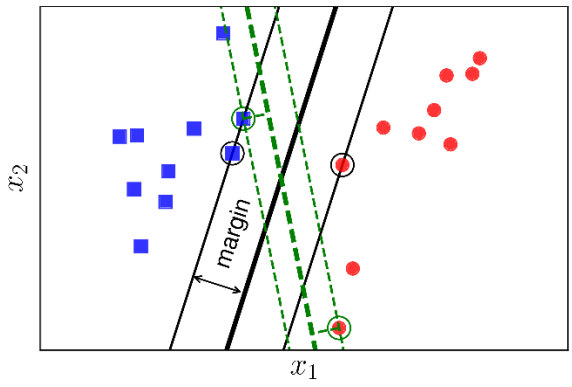
Thuật toán KNN có nhược điểm là hiệu suất kém trên các bộ dữ liệu được phân phối không đồng đều. Nếu một lớp có nhiều điểm dữ liệu hơn các lớp khác thì một điểm dữ liệu mới sẽ có nhiều điểm lân cận thuộc lớp đó và từ đây sẽ dẫn đến kết quả dự đoán không chính xác [1].

2.2.2 SVM

SVM (*Support Vector Machine*) là thuật được sử dụng nhiều trong các bài toán phân lớp. Ý tưởng nằm ở việc tìm ra một siêu mặt phẳng phân chia các lớp tối ưu

nhất. Cụm từ '*support vector*' nhằm chỉ ra rằng các điểm nằm gần siêu mặt phẳng nhất nếu bị xóa đi có thể khiến vị trí của siêu mặt phẳng bị ảnh hưởng. Đồng thời ta cũng có giá trị biên (*margin*) là khoảng cách giữa *support vector* và siêu mặt phẳng.

Siêu mặt phẳng càng nằm xa các lớp chứng tỏ dự đoán càng chính xác. Do đó mặc dù có thể tìm được rất nhiều siêu mặt phẳng cho mỗi bài toán nhưng việc tìm kiếm được một siêu mặt phẳng để biên lớn nhất vẫn là mục tiêu của SVM.



Hình2.2: Thuật toán SVM

Hình 2.2 cho thấy tập dữ liệu được chia thành hai lớp và bài toán nằm trong không gian hai chiều. Ở đây siêu mặt phẳng được biểu diễn dưới dạng đường thẳng. Thuật toán SVM được mô tả là:

- Gọi hai tập X và Y lần lượt là tập đầu vào và nhãn tương ứng.

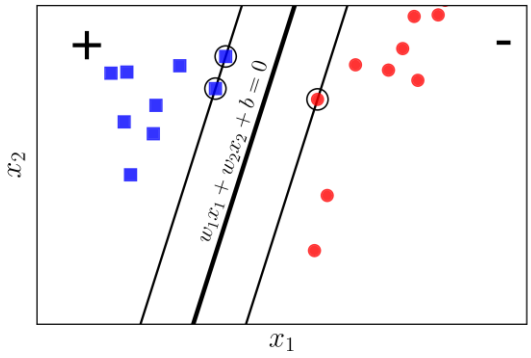
(x_1, y_1), (x_N, y_N) là tập huấn luyện.

Với vector $x_i \in R^d$, trong đó:

d: số chiều của dữ liệu

N: số điểm dữ liệu.

Ví dụ: Đối với trường hợp không gian hai chiều ở hình 2.3, các phép toán sẽ được tổng quát lên không gian nhiều chiều:



Hình 2.3: Thuật toán SVM trong không gian 2 chiều

- Giả sử nhãn của mỗi điểm được xác định bởi $y_i = 1$ hoặc $y_i = -1$.

Các điểm vuông thuộc lớp 1 (+), điểm tròn thuộc lớp -1 (-) và mặt: $w^Tx + b = w_1x_1 + w_2x_2 + b = 0$

Là mặt phẳng phân chia giữa 2 lớp.

- Khoảng cách từ một cặp dữ liệu (x_n, y_n) bất kỳ tới mặt phẳng phân chia là:

$$d = \frac{y_n(w^Tx_n+b)}{||w||_2} \tag{2.3}$$

- Biên sẽ là khoảng cách gần nhất từ một điểm tới mặt phân chia và được tính theo công thức:

$$margin = \min_n \frac{y_n(w^Tx_n+b)}{||w||_2} \tag{2.4}$$

- Mục tiêu của thuật toán SVM là tìm w và b sao cho giá trị biên (*margin*) là lớn nhất, ta có công thức:

$$(w, b) = \arg\max_{w,b}\left\{\min_n \frac{y_n(w^Tx_n + b)}{||w||_2}\right\}$$

$$= \arg\max_{w,b}\left\{\frac{1}{||w||_2}\min_n[y_n(w^Tx_n + b)]\right\} \tag{2.5}$$

Thuật toán SVM thường sẽ cho ra những kết quả khá chính xác. Đặc biệt là đối với các tập dữ liệu "sạch". Và thuật toán SVM cũng thích hợp cho những tập dữ liệu nhiều chiều khác kể cả khi số chiều nhiều hơn số lượng mẫu. Thuật toán SVM cũng có hiệu quả cao đối với các tập dữ liệu chứa nhiều nhiễu hoặc chồng chéo lên nhau. Nhưng bên canh đó sẽ mất thời gian lâu để huấn luyện [8].

2.2.3 Naive Bayes

Naive Bayes là thuật toán phân lớp dựa vào định lý *Bayes* về lý thuyết xác suất. Đây là thuật toán có thể được sử dụng cho các bài toán nhị phân hoặc phân lớp nhiều lớp. Thuật toán *Naive Bayes* sẽ xử lý từng đặc trưng một cách độc lập, nó sẽ tính xác suất của mỗi đặc trưng trước và đưa ra dự đoán dựa vào định lý *Bayes*.

Xác suất tiên nghiệm (xác suất lớp) là xác suất của một lớp thuộc tập dữ liệu và không bị phụ thuộc vào những yếu tố khác. Mặt khác, nếu chọn một điểm bất kỳ trong tập dữ liệu này thì xác suất lớp sẽ là xác suất mà điểm thuộc về một lớp nhất định. Xác suất có điều kiện chính là xác suất mà giá trị đặc trưng của một điểm dữ liệu thuộc một lớp nhất định khi biết xác suất của lớp đó. Nếu gọi xác suất tiên nghiệm của lớp là P(C), xác suất có điều kiện là P(V|C) thì ta có công thức:

$$\mathrm{P(C)} = \frac{\text{Số điểm dữ liệu có trong lớp C}}{\text{Tổng số điểm dữ liệu}} \tag{2.6}$$

$$\mathrm{P(V|C)} = \frac{\text{Số điểm dữ liệu có giá trị V thuộc lớp C}}{\text{Tổng số điểm dữ liệu có giá trị V}} \tag{2.7}$$

Xác suất để một điểm thuộc từng lớp sẽ được tính toán, so sánh và lớp nào có xác suất cao nhất thì sẽ được lựa chọn làm kết quả.

Thuật toán Naive Bayes có ưu điểm đơn giản và dễ hiểu. Thuật toán sẽ thích hợp với những tập dữ liệu có nhiều đặc trưng khác nhau do dự đoán cần phụ thuộc vào xác suất của các đặc trưng đó. Bên cạnh đó, Naive Bayes còn sử dụng ít tài nguyên nhưng lại có hiệu năng cao cũng như không cần tính đến các hệ số phụ như các thuật toán khác. Nhưng do các đặc trưng được xử lý độc lập do đó trong một số trường hợp có thể cho ra kết quả không chính xác [4].

2.2.4 J48 Decision Tree

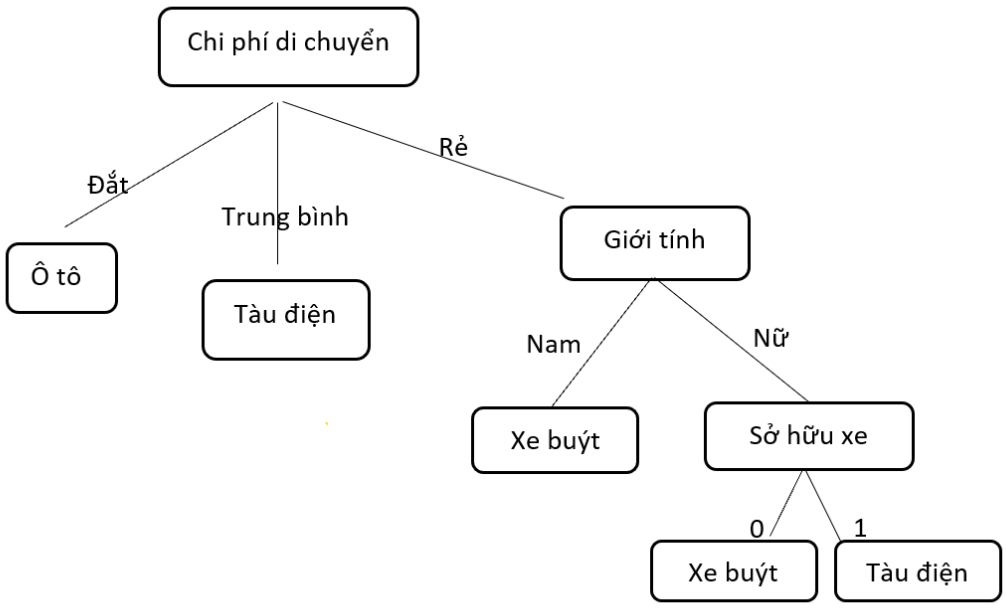
Thuật toán cây quyết định J48 (*J48 Decision Tree*): Đây là một cây phân cấp có cấu trúc được sử dụng để phân lớp các đối tượng dựa vào dãy các luật. Nếu đưa dữ liệu cho các đối tượng bao gồm những thuộc tính cùng với lớp của nó, lúc này *J48 Decision Tree* sẽ sinh ra các luật nhằm dự đoán lớp của các đối tượng chưa biết. Điều

mà thuật toán *J48 Decision Tree* muốn đạt được là kết quả chính xác nhất với số lần lựa chọn ít nhất. Ví dụ:

Bảng 2.1: Dữ liệu huấn luyện của J48 Decision Tree

Thuộc Tính				**Phân loại**
Giới tính	Sở hữu xe	Chi phí di chuyển ($/km)	Mức thu nhập	Phương thức di chuyển
Nam	0	Rẻ	Thấp	Xe buýt
Nam	1	Rẻ	Trung bình	Xe buýt
Nữ	1	Rẻ	Trung bình	Tàu điện
Nữ	0	Rẻ	Thấp	Xe buýt
Nam	1	Rẻ	Trung bình	Xe buýt
Nam	0	Trung bình	Trung bình	Tàu điện
Nữ	1	Trung bình	Trung bình	Tàu điện
Nữ	1	Đắt	Cao	Ô tô
Nam	2	Đắt	Trung bình	Ô tô
Nữ	2	Đắt	Cao	Ô tô

Dựa vào bảng trên ta có thể tạo ra cây quyết định:



Hình 2.4: Cây quyết định minh họa

Nhận thấy trong cây quyết định trên, thuộc tính "Mức thu nhập" đang không có trong cây do khi sử dụng dữ liệu huấn luyện đã cho, thuộc tính "Chi phí di chuyển" sẽ sinh ra cây quyết định tốt dùng để phân loại tốt hơn "Mức thu nhập".

Thuật toán thường dùng trong cây quyết định là *Iterative Dichotomiser 3* (ID3). Thuật toán này sẽ dựa trên các khái niệm *entropy*, độ lợi thông tin. Ở đây *entropy* sử dụng để chỉ mức độ không chắc chắn của dữ liệu. Ví dụ, *entropy* của việc tung đồng xu sẽ là không xác định do không có cách nào có thể xác định chính xác kết quả của điều này. Mặt khác, nếu cả hai mặt của đồng xu đều là hình thì kết quả sẽ là 0 *entropy*, vì kết quả chính xác trước mỗi lần tung đồng xu đã được biết trước [6].

Thuật toán ID3 được mô tả như sau: Bắt đầu từ nút gốc, tại mỗi bước, thuộc tính tốt nhất sẽ được chọn lọc ra sao cho độ lợi thông tin là lớn nhất. Sau đó, với mỗi thuộc tính được chọn ra ta chia dữ liệu vào các nút con tương ứng với mỗi giá trị của thuộc tính đó và tiếp tục áp dụng phương pháp này cho những nút con [9]. Các bước thực hiện:

- Giả sử tập dữ liệu S có N phần tử, C lớp khác nhau. Trong đó Nc điểm thuộc lớp c. Entropy ban đầu H(S) tính theo công thức:

$$H(S) = -\sum_{c=1}^{C} \frac{N_c}{N} \log\left(\frac{N_c}{N}\right) \quad (2.8)$$

- Giả sử thuộc tính được chọn là x, các điểm dữ liệu trong S được phân thành K nút con, trong đó số điểm ở mỗi nút con tương ứng là m1, m2,…, mK. Entropy của mỗi nhánh H(x, S) được tính như sau:

$$H(x, S) = \sum_{k=1}^{K} \frac{m_k}{N} H(S_k) \quad (2.9)$$

- Độ lợi thông tin G(x,S) cũng được tính dựa trên sự chênh lệch giữa *entropy* ban đầu và *entropy* của mỗi nhánh như sau:

$$G(x, S) = H(S) - H(x, S) \quad (2.10)$$

- Theo đó, thuộc tính nào có độ lợi thông tin lớn nhất sẽ được chọn làm nút quyết định.

- Đồng thời, nếu một trong các nhánh của nút quyết định được chọn mà có *entropy* bằng 0 thì nó sẽ trở thành nút lá. Khi đó các nhánh khác sẽ tiếp tục được phân chia.

- Thuật toán sẽ chạy đến khi nào không thể phân chia được nữa thì mới dừng lại. *J48 Decision Tree* là thư viện cài đặt của thuật toán ID3 có trong gói ngôn ngữ R.

Các ưu điểm của cây quyết định có thể thấy như: Xử lý tốt các tập dữ liệu lớn cũng như có nhiều dữ liệu nhiễu, theo dõi quá trình lựa chọn một cách tường minh, do đó cây quyết định trở thành thuật toán phổ biến bởi sự đơn giản cả trong các bài toán như chuẩn đoán y tế, lọc thư rác, sàng lọc an ninh …

Có thể thấy, chỉ trong một khoảng thời gian không dài mà cây quyết định đã xử lý tốt một lượng dữ liệu lớn. Đồng thời sử dụng máy tính cá nhân để phân tích các lượng dữ liệu lớn trong một thời gian không dài khiến dễ dàng đưa ra quyết định dựa trên những phân tích của cây quyết định đưa ra.

Cây quyết định cũng có nhược điểm bên cạnh các ưu điểm trên như: Khó giải quyết được các vấn đề có dữ liệu phụ thuộc thời gian liên tục dễ xảy ra lỗi khi có quá nhiều lớp chi phí tính toán nhằm xây dựng mô hình cây quyết định cao.

2.2.5 Random Forest

Thuật toán *Random Forest* khá thông dụng và được dùng nhiều trong học máy. Điểm đặc biệt là *Random Forest* hầu như không cần xử lý dữ liệu hay lập mô hình trước đó nhưng vẫn mang lại kết quả tương đối chính xác. Thuật toán xây dựng trên tính ngẫu nhiên và được tạo ra bởi nhiều cây quyết định.

Thuật toán *Random Forest* sẽ coi mỗi cây quyết định giống như một cử tri bỏ phiếu độc lập. Khi đó cuối cuộc bầu cử, câu trả lời nhận được nhiều bình chọn nhất từ các cây quyết định sẽ là câu trả lời được lựa chọn.

Nhưng vẫn còn một vấn đề ở đây là nếu như tất cả các cây được dựng theo cùng một cách, vậy có khả năng chúng sẽ cho ra câu trả lời giống nhau. Như vậy cũng không khác với việc chúng ta chỉ sử dụng một cây quyết định duy nhất. Do đó, *Random Forest* có một cách khác để chắc chắn rằng tất cả các cây quyết định sẽ không cho cùng câu trả lời, đó là chọn ngẫu nhiên các quan sát. Nói rõ hơn thì *Random Forest* sẽ xoá một số quan sát và lặp lại một số khác ngẫu nhiên. Về tổng thể, những quan sát này vẫn rất gần với tập các quan sát ban đầu nhưng các thay đổi nhỏ sẽ đảm

bảo từng cây quyết định sẽ cho ra một chút kết quả khác biệt, quá trình này được gọi là *bootstrapping*.

Đồng thời để thực sự chắc chắn các cây quyết định là khác nhau thì *Random Forest* cũng ngẫu nhiên bỏ qua một số câu hỏi trong lúc xây dựng cây quyết định. Tại đây nếu câu hỏi tốt nhất không được chọn, một câu hỏi kế tiếp sẽ được lựa chọn thay thế để dựng cây, quá trình này được gọi là *attribute sampling*.

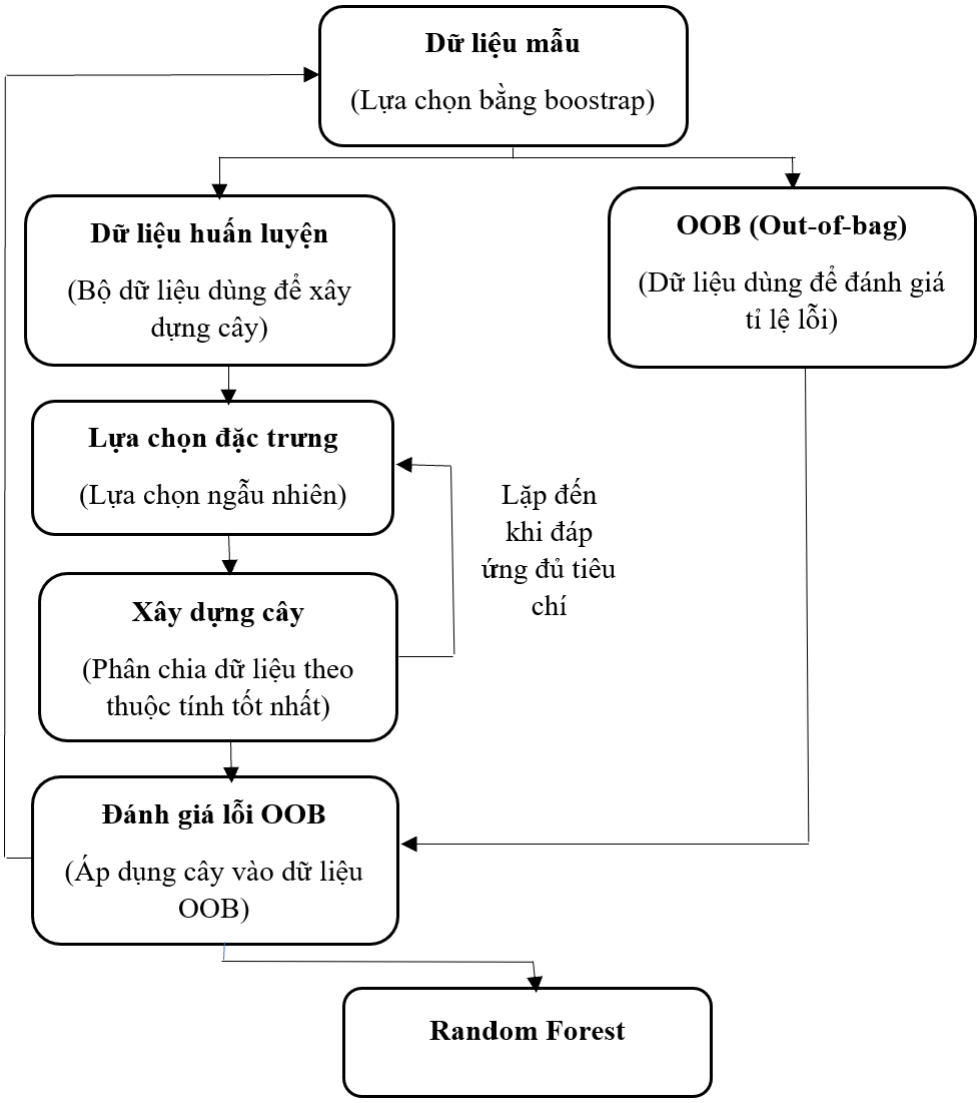
Trong đó, về cơ bản *Random Forest* sẽ xây dựng các cây quyết định dựa trên những tập con độc lập nhau trong tập dữ liệu cho trước. Một vài giá trị đặc trưng cũng sẽ được chọn ra ngẫu nhiên đến khi tìm được cách phân chia tốt nhất, tại mỗi nút. *Random Forest* được mô tả như sau:

- Các cây sẽ được xây dựng dựa vào 2/3 dữ liệu của tập dữ liệu đào tạo (62.3%). Dữ liệu cũng sẽ được lựa chọn ngẫu nhiên.

- Tại đây sẽ có một số biến dự đoán được chọn ngẫu nhiên từ tổng số các biến dự đoán. Do đó cách phân chia tốt nhất của các biến được lựa chọn được dùng để phân chia nút. Về cơ bản thì số lượng biến được chọn chính là căn bậc hai của tổng số những thuộc tính sử dụng để dự đoán và không đổi đối với các cây.

- Từ đây tỉ lệ dự đoán sai sẽ được tính toán phụ thuộc vào phần dữ liệu còn lại (dữ liệu *out-of-bag*).

- Mỗi cây huấn luyện cũng đưa ra một kết quả phân loại. Điều này được gọi là "bỏ phiếu". Lớp nào nhận được nhiều "phiếu" nhất sẽ là lớp được chọn là kết quả cuối cùng.



Hình 2.5: Sơ đồ mô tả thuật toán Random Forest

2.2.6 Mạng Nơ Ron

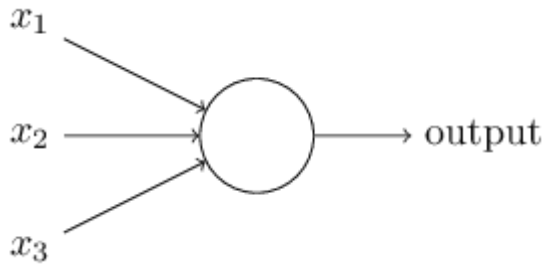
Trong học máy, các mạng nơ ron nhân tạo (ANN) được sử dụng thể hiện mô hình để đưa ra dự đoán cũng như phân loại dữ liệu chưa từng xuất hiện trong tập mẫu. Trong luận văn, ANN sẽ được sử dụng để phân loại dữ liệu các lớp tấn công.

a. Khái niệm

Mạng nơ ron nhân tạo (ANN) được tạo nên từ một số lượng lớn các phần tử nơ ron liên kết với nhau. Tại đây, mỗi nơ ron sẽ tính tổng những giá trị đầu vào với mỗi trọng số đã học được, sau đó chuyển kết quả cho hàm trả về một giá trị. Chức

năng logistic sẽ là một lựa chọn phổ biến cho chức năng kích hoạt. Trong đó, bố cục của các nơ ron cũng phụ thuộc vào kiến trúc mạng.

- *Perceptrons*: Được xây dựng và phát triển vào những năm 1950 - 1960 bởi nhà khoa học Frank Rosenblatt. Hiện nay *perceptrons* phổ biến trong nhiều mô hình mạng nơ ron khác nhau. Một *perceptron* có một số giá trị nhị phân làm đầu vào, tạo nên một đầu ra (output) nhị phân duy nhất [5]:

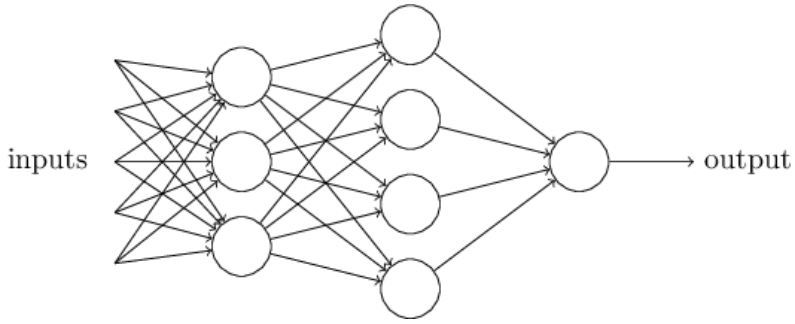


Hình 2.6: Perceptrons

- Hình trên là ví dụ về *perceptron*: Ta thấy có ba đầu vào x1, x2, x3. Ta cũng có thể thêm hoặc bớt số đầu vào. Rosenblatt đề xuất quy tắc đơn giản để tính toán đầu ra. Ta có những trọng số w1, w2 là các giá trị thể hiện sự quan trọng của từng yếu tố đầu vào tương ứng với đầu ra. Các nơ ron đầu ra 0 hoặc 1 sẽ được xác định bởi tổng $\sum_i w_i x_i$, nhỏ hơn hoặc lớn hơn so với một giá trị ngưỡng (*threshold*). Ngưỡng ở đây là số thực và cũng là tham số của nơ ron. Đầu ra sẽ được xác định bằng công thức sau [5]:

$$\text{output} = \begin{cases} 0, \text{ nếu } \sum_i w_i x_i \leq \text{threshold} \\ 1, \text{ nếu } \sum_i w_i x_i > \text{threshold} \end{cases} \quad (2.11)$$

Từ đây chúng ta có được những mô hình khác nhau nhờ việc thay đổi trọng số và ngưỡng. Nhưng một *perceptron* không phải là một mô hình ra quyết định tối ưu nhất vì vậy ta có một mạng lưới kết hợp nhiều *perceptron* để đưa ra quyết định chính xác.



Hình 2.7: Mạng nơ ron chứa nhiều perceptrons

Hình trên ta có lớp đầu tiên của *perceptron* có thể đưa ra ba quyết định từ đầu vào. Tại lớp *perceptron* thứ hai, mỗi *perceptron* được quyết định bởi trọng số lên các đầu ra từ lớp đầu tiên và có thể tự đưa ra quyết định ở mức độ khó hơn *perceptron* trong lớp đầu. Những quyết định khó hơn này có thể được thực hiện từ những *perceptron* trong lớp thứ ba, thứ tư…. Một mạng lưới nhiều lớp của *perceptron* sẽ có thể tham gia vào việc ra quyết định phức tạp bằng cách này. Khi xác định *perceptron* ta phải biết rằng một *perceptron* chỉ có đúng một đầu ra duy nhất mà thôi. Do đó, việc có nhiều mũi tên đầu ra chỉ là cách hữu hiệu cho thấy đầu ra từ một *perceptron* đang được dùng như là đầu vào cho *perceptron* khác. Đồng thời, *perceptron* cho thấy ta có thể đưa ra các thuật toán học tự động điều chỉnh trọng số cũng như định hướng của một mạng ANN. Việc này sẽ xảy ra khi phản ứng với những kích thích bên ngoài, mà không cần đến sự can thiệp của lập trình viên. Các thuật toán được cài đặt để cho phép sử dụng ANN theo cách khác biệt với các cổng logic thông thường. Mạng nơ ron có thể học để giải quyết những vấn đề một cách đơn giản. Trong khi vấn đề đó lại vô cùng phức tạp đối với những mô hình truyền thống.

- *Sigmoid*: Khái niệm này ra đời để giải quyết hạn chế chính của *perceptron*. Đó là có rất nhiều giá trị cần thiết kế điều chỉnh, do những thay đổi nhỏ về trọng số của bất kỳ giá trị *perceptron* nào cũng sẽ mang đến nguy cơ làm thay đổi giá trị toàn mạng đáng kể. Hàm chuyển đổi *sigmoid* tương tự như *perceptron* nhưng sẽ xuất hiện sự sửa đổi để nếu có thay đổi nhỏ trong trọng số cũng như định hướng thì cũng chỉ gây ra một sự thay đổi nhỏ ở đầu ra. Giống như *perceptron*, các nơ ron *sigmoid* có đầu vào x1, x2… Nhưng thay vì chỉ mang giá trị 0 hoặc 1, các đầu vào ở đây có thể đưa vào bất cứ giá trị nào giữa 0 và 1. Các nơ ron *sigmoid* cũng chứa trọng số cho mỗi

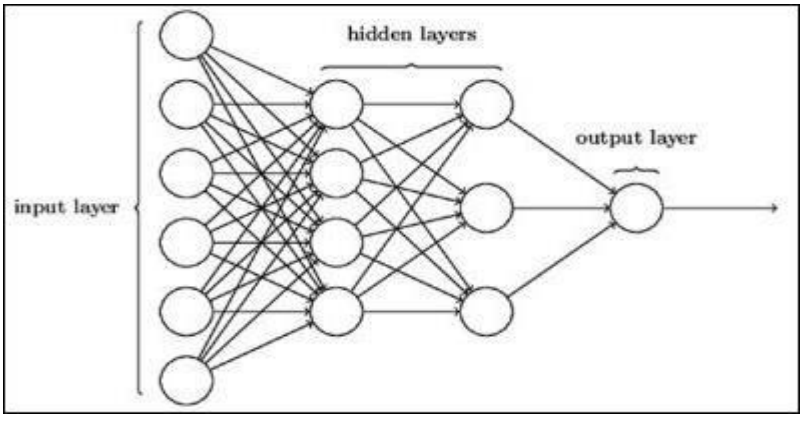
đầu vào là w1, w2 … nhưng đầu ra khi áp dụng hàm chuyển đổi *sigmoid* có thể là giá trị số thực.

b. Kiến trúc

Các mạng nơ ron truyền thống sẽ được chia thành ba loại kiến trúc mạng khác nhau là mạng cấp dữ liệu một lớp, mạng cấp dữ liệu đa lớp và mạng hồi quy.

ANN chuyển tiếp sẽ được xây dựng dựa vào một hoặc nhiều lớp tế bào nơ ron kết nối cùng với các lớp tế bào nơ ron sau, không có kết nối với các lớp trước. Đồng thời đối với mạng chuyển tiếp nguồn cấp một lớp thì chỉ có lớp đầu ra của các nút thực hiện toàn bộ các tính toán. Đối với trường hợp những mạng chuyển tiếp nguồn cấp đa lớp thì các lớp nằm giữa lớp đầu ra và nguồn được gọi là các lớp ẩn.

Mạng nơ ron đơn giản với một lớp ẩn duy nhất. Lớp ngoài cùng bên trái tại mạng này sẽ được gọi là lớp đầu vào đồng thời các nơ ron trong lớp này cũng được gọi là nơ ron đầu vào. Đầu ra hoặc lớp ngoài cùng bên phải sẽ chứa các nơ ron đầu ra. Đối với trường hợp này thì chỉ có một nơ ron đầu ra duy nhất. Trong đó, lớp giữa sẽ được gọi là lớp ẩn, các nơ ron trong lớp này không phải đầu vào nhưng cũng không phải đầu ra. Mạng này đang có một lớp ẩn duy nhất, nhưng một số mạng thì sẽ có nhiều lớp ẩn hơn. Ta có ví dụ về mạng bốn lớp trong hình dưới đây có đến hai lớp ẩn. Trong khi đó, việc thiết lập lớp đầu vào cũng như đầu ra của một mạng nơ ron thường khá phức tạp nên việc tạo các lớp ẩn cũng sẽ mất nhiều thời gian và công sức hơn. Vì vậy các nhà nghiên cứu mạng nơ ron đã phát triển thêm nhiều công nghệ tự động thiết kế cho các lớp ẩn, từ đây giúp mọi người có được những đầu ra theo những mong muốn của họ. Điều này được dùng để cân bằng số lớp ẩn với thời gian cần thiết cho việc đào tạo mạng [5].



Hình 2.8: Mạng nơ ron bốn lớn với hai lớp ẩn

c. Quá trình xử lý

Kết quả đầu ra của một ANN là giải pháp cho một vấn đề cụ thể nào đó. Trọng số liên kết thể hiện độ cần thiết của dữ liệu đầu vào với tiến trình xử lý thông tin. Việc thay đổi những trọng số của dữ liệu đầu vào để có được kết quả mong muốn chính là quá trình học của ANN [5].

Hàm tổng có chức năng tính tổng trọng số của tất cả các phần tử xử lý. Hàm tổng của một nơ ron sẽ cho biết được khả năng kích hoạt của nơ ron đó, đây còn gọi là nội kích hoạt. Các nơ ron này có thể sẽ sinh ra được một đầu ra hoặc sẽ không thể sinh ra đầu ra nào trong ANN. Từ đây, mối quan hệ giữa nội kích hoạt và kết quả sẽ được mô tả bằng hàm chuyển đổi. Vì kết quả xử lý tại các nơ ron đầu ra có thể sẽ rất lớn, do đó trước lúc chuyển qua lớp tiếp thì hàm chuyển đổi sẽ được dùng trong việc xử lý đầu ra này. Kết quả của ANN bị ảnh hưởng rất nhiều bởi việc lựa chọn hàm chuyển đổi. Trong đó hàm chuyển đổi phi tuyến là sigmoid được dùng nhiều nhất trong ANN. Hàm sigmoid còn được gọi là hàm chuẩn hóa. Vì vậy thỉnh thoảng thay vì sử dụng hàm chuyển đổi người ta sẽ sử dụng giá trị ngưỡng (*threshold*) nhằm kiểm soát các đầu ra của những nơ ron tại một lớp đó trước khi chuyển các đầu ra này đi đến các lớp tiếp theo.

Mạng nơ ron nhân tạo được huấn luyện theo hai kỹ thuật cơ bản đó là học có giám sát và học không giám sát [5].

- Kỹ thuật học có giám sát: Đây là quá trình học được lặp lại cho đến khi kết quả của ANN đạt được giá trị mong muốn mới thôi. Đặc trưng cho kỹ thuật này chính là mạng nơ ron lan truyền ngược (*Backpropagation*).
- Kỹ thuật học không giám sát: Quá trình học này không sử dụng kiến thức bên ngoài trong quá trình học. Mạng nơ ron đặc trưng được huấn luyện theo kiểu không giám sát này chính là SOM (*Sefl- Organizing Map*).

d. Phương thức huấn luyện

Mạng nơ ron sẽ có ba cách huấn luyện chính là: Huấn luyện theo gói, huấn luyện ngẫu nhiên và huấn luyện trực tuyến. Riêng đối với huấn luyện trực tuyến thì các trọng số của mạng sẽ được cập nhật ngay lập tức sau khi một mẫu đầu vào được

đưa vào mạng. Còn huấn luyện ngẫu nhiên cũng khá giống với huấn luyện trực tuyến nhưng ở đây việc chọn các mẫu đầu vào để đưa vào mạng từ tập huấn luyện sẽ được thực hiện ngẫu nhiên. Đối với huấn luyện theo gói thì tất cả các mẫu đầu vào sẽ được đưa vào mạng cùng lúc, sau đó cập nhật các trọng số mạng đồng thời. Từ đây có thể rút ra được ưu điểm của huấn luyện trực tuyến là tiết kiệm bộ nhớ do không cần lưu lại số lượng lớn các mẫu đầu vào trong bộ nhớ.

Trong quá trình huấn luyện mạng, thuật ngữ "*epoch*" được dùng để thể hiện quá trình: Tất cả các mẫu đầu vào của tập huấn luyện được đưa vào để huấn luyện mạng. Có thể nói rằng một *epoch* sẽ được hoàn thành khi toàn bộ các dữ liệu trong tập huấn luyện sẽ được đưa vào huấn luyện mạng. Do đó số lượng epoch sẽ xác định số lần mạng được huấn luyện.

2.3 Thuật toán học máy trên IoT gateway

2.3.1 Phân tích và lựa chọn mạng Nơ ron

Mạng Nơ ron là một mô hình học máy phổ biến, nét đặc trưng của mạng Nơ ron là khả năng học. Mạng Nơ ron có thể gần đúng mối quan hệ tương quan phức tạp giữa các yếu tố đầu vào và đầu ra của các quá trình cần nghiên cứu, sau khi đã học được thì việc kiểm tra độc lập thường sẽ cho ra kết quả tốt. Đồng thời, khi đã học xong, mạng Nơ ron nhân tạo có thể tính toán kết quả đầu ra tương ứng với bộ số liệu đầu vào mới. Về mặt cấu trúc, mạng Nơ ron nhân tạo là một hệ thống gồm nhiều phần tử xử lý đơn giản cùng hoạt động song song. Tính năng này của ANN cho phép nó có thể được áp dụng để giải các bài toán lớn.

Về khía cạnh toán học, theo định lý Kolmogorov, một hàm liên tục bất kỳ f(x_1, x_2,…, x_n) xác định trên khoảng I_n(với I =[0,1]) có thể được biểu diễn dưới dạng:

$$f(x) = \sum_{j=1}^{2n+1} \chi_j\left(\sum_{i=1}^{n} \psi_{ij}(x_i)\right) \qquad (2.12)$$

Trong đó: $\chi_j, \psi_{ij}(x_i)$ là các hàm liên tục một biến, $\psi_{ij}(x_i)$ là hàm đơn điệu, không phụ thuộc vào hàm f.

Mặt khác, mô hình mạng Nơ ron cho phép liên kết có trọng số các phần tử phi tuyến (các Nơ ron đơn lẻ) tạo nên dạng hàm tổng hợp từ các hàm thành phần. Do vậy, sau một quá trình học, các phần tử phi tuyến đó sẽ tạo nên một hàm phi tuyến phức tạp có khả năng xấp xỉ hàm biểu diễn quá trình cần nghiên cứu. Kết quả là đầu ra của

nó sẽ tương tự với kết quả đầu ra của tập dữ liệu dùng để huấn luyện mạng. Khi đó ta nói mạng Nơ ron nhân tạo đã học được mối quan hệ tương quan đầu vào - đầu ra của quá trình và lưu lại mối quan hệ tương quan này thông qua bộ trọng số liên kết giữa các Nơ ron. Do đó, mạng Nơ ron có thể tính toán trên bộ số liệu đầu vào mới để đưa ra kết quả đầu ra tương ứng.

Với những đặc điểm đó, mạng Nơ ron đã được sử dụng để giải quyết nhiều bài toán thuộc nhiều lĩnh vực của các ngành khác nhau. Điển hình nhóm ứng dụng mà mạng Nơ ron đã được áp dụng rất có hiệu quả là bài toán phân lớp: Loại bài toán này đòi hỏi giải quyết vấn đề phân loại các đối tượng quan sát được thành các nhóm dựa trên các đặc điểm của các nhóm đối tượng đó. Đây là dạng bài toán cơ sở của rất nhiều bài toán trong thực tế: nhận dạng chữ viết, tiếng nói, phân loại gen, phân loại chất lượng sản phẩm … Do đó luận văn sẽ triển khai thử nghiệm mạng Nơ ron trên bộ dữ liệu UNSW-NB15.

2.3.2 Phân tích và lựa chọn thuật toán Random Forest

Khi so sánh với các thuật toán học có giám sát hiện giờ như *Boosting*, *Baging*, *Nearest neighbors*, SVM, Mạng nơ ron, C45... [3].

Có thể thấy thuật toán *Random Forest* (RF) cho độ chính xác phân lớp cao hơn.

Random Forest bao gồm một tổ hợp các cây quyết định không cắt nhánh. Trong đó, mỗi cây quyết định ở đây sẽ được dựng lên bởi thuật toán CART trên tập mẫu *bootstrap* (lấy mẫu ngẫu nhiên có hoàn lại) thuộc tập dữ liệu ban đầu. Trong mỗi nút, một phân hoạch tốt nhất sẽ được thực hiện dựa vào thông tin trong một không gian con, các thuộc tính ở đây được chọn ngẫu nhiên từ không gian thuộc tính ban đầu. Thuật toán *Random Forest* sẽ tổng hợp kết quả dự đoán của các cây quyết định để làm kết quả cuối cùng. Có thể nhận thấy ưu điểm nổi bật của RF là xây dựng cây mà không cần thực hiện việc cắt nhánh từ các tập dữ liệu con khác nhau và dùng kỹ thuật *boostrap* có hoàn lại vì vậy thu được những cây với lỗi bias thấp. Không những thế, mối quan hệ tương quan giữa các cây quyết định cũng được giảm thiểu nhờ vào việc xây dựng các không gian con thuộc tính một cách ngẫu nhiên. Vì vậy việc kết hợp kết quả của một số lượng lớn những cây quyết định độc lập có bias thấp, phương

sai cao cũng sẽ giúp thuật toán *Random Forest* đạt được cả độ lệch thấp cũng như phương sai thấp. Có thể thấy chất lượng của việc dự đoán cũng như mức độ tương quan giữa các cây quyết định ảnh hưởng không nhỏ đến sự chính xác của RF.

Random Forest phân loại hiệu quả các cuộc tấn công vì là một bộ phân loại đồng bộ và hoạt động tốt so với các phân loại truyền thống khác. Việc đánh giá hiệu suất của mô hình thông qua cách tiến hành thử nghiệm trên bộ dữ liệu NSL-KDD. Kết quả thực nghiệm sẽ cho thấy mô hình đề xuất có tác dụng với tỷ lệ cảnh báo sai thấp và tỷ lệ phát hiện cao.

Vì *Random Forest* là một thành viên trong họ thuật toán *Decision Tree* vậy nên tư tưởng chính của RF là tạo ra nhiều cây quyết định từ dataset, mỗi cây quyết định sẽ dự đoán một kết quả và kết quả nào được nhiều cây quyết định dự đoán nhất thì đó sẽ trở thành kết quả cuối cùng. Cũng để chắc chắn rằng không phải tất cả các cây quyết định đều cho ra cùng một câu trả lời (nếu như các cây quyết định được tạo ra theo cùng 1 cách thì chúng có thể sẽ cho ra cùng một câu trả lời) thì trong quá trình xây dựng cây, RF sẽ chọn ngẫu nhiên các quan sát, đây là quá trình *bootstrapping* và chọn ngẫu nhiên các thuộc tính, đây là quá trình *attribute sampling*. RF được đánh giá cao chính bởi vì tính chính xác của mô hình. Tuy nhiên RF cũng có nhược điểm: khối lượng tính toán lớn. Tuy nhiên hiện nay điều này có thể khắc phúc được nhờ vào sự phát triển của IoT gateway.

Một nhóm nghiên cứu khác đã chỉ ra kết quả của thuật toán *Random Forest* hiệu quả hơn SVM, *Naive Bayes*, *Decision Tree* trên bộ dữ liệu UNSW-NB15 với những chỉ số vượt trội như sau:

Bảng 2.2: Kết quả thử nghiệm của các thuật toán

Methods	Accuracy	Sensitivity	Specificity	Training Time	Prediction Time
Random Forest	**97.49**	**93.53**	**97.75**	**5.69**	**0.08**
SVM	92.28	92.13	91.15	38.91	0.20
Naive Bayes	74.19	92.16	67.82	2.25	0.18
Decision Tree	95.82	92.52	97.10	4.80	0.13

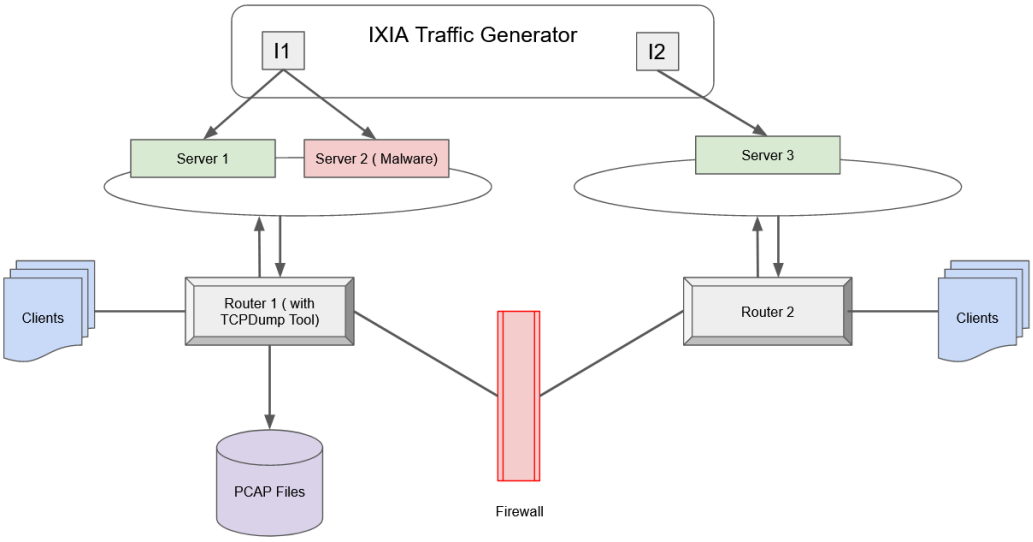
Trong quá trình triển khai thực nghiệm, luận văn này đưa ra kết quả sử dụng thuật toán *Random Forest* cùng với bộ dữ liệu UNSW-NB15 chỉ phân loại tốt dữ liệu

mạng thông thường và dữ liệu mạng tấn công. Kiến trúc tổng thể có những phương thức được dùng đó là: tiền xử lý dữ liệu, đào tạo huấn luyện qua thuật toán RF, đo lường hiệu suất phát hiện mối nguy hại.

2.4 Tập dữ liệu mẫu UNSW-NB15

Hiện nay, một trong những thách thức không nhỏ trong việc nghiên cứu trong lĩnh vực phát hiện xâm nhập IoT chính là không có bộ dữ liệu dựa trên mạng toàn diện, một bộ dữ liệu có thể phản ánh các kịch bản lưu lượng tấn công mạng hiện đại. Trong nghiên cứu hệ thống phát hiện xâm nhập mạng, bộ dữ liệu chuẩn như KDD98, KDDCUP99 và NSLKDD đã được tạo ra từ một thập kỷ trước. Tuy nhiên có rất nhiều nghiên cứu hiện tại đã cho thấy đối với môi trường tấn công thực tế thì những bộ dữ liệu này không thể phản ánh đầy đủ lưu lượng truy cập mạng cũng như các cuộc tấn công hiện đại. Do đó, luận văn sẽ ứng dụng tập dữ liệu UNSW-NB15, đây là tập dữ liệu có được sự kết hợp của dữ liệu mạng bình thường và các phương thức tấn công hiện đại.

Các gói tin mạng thô trong bộ dữ liệu UNSW-NB15 sẽ được xây dựng nhờ công cụ *IXIA PerfectStorm* ở trong Phòng thí nghiệm *Cyber Range* của Trung tâm An ninh mạng (ACCS) - Australia để tạo thành hỗn hợp các hoạt động bình thường trong thực tế cũng như tổng hợp các hành vi tấn công mới.



Hình 2.9: Mô hình mô phỏng phòng thí nghiệm tạo tập dữ liệu UNSW-NB15

Công cụ *Tcpdump* được dùng để thu 100 GB lưu lượng truy cập thô (gói tin Pcap). Trong đó, bộ dữ liệu này có chín loại tấn công bao gồm: *Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode* và *Worms*. Các công cụ *Argus*, Bro-IDS được sử dụng ở đây và mười hai thuật toán được phát triển để tạo thành 49 tính năng với các nhãn lớp.

Cùng tìm hiểu thông tin của tập dữ liệu UNSW-NB15:

Bảng 2.3: Bảng mô tả thông tin của tập dữ liệu UNSW-NB15

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	srcip	nominal	Địa chỉ IP nguồn
2	sport	integer	Số hiệu của cổng ở nguồn
3	dstip	nominal	Địa chỉ IP đích
4	dsport	integer	Số hiệu của cổng ở đích
5	proto	nominal	Giao thức kết nối
6	state	nominal	Chỉ ra trạng thái và giao thức phụ thuộc của nó, ví dụ như: ACC, CLO, CON, ECO, ECR, FIN, INT, MAS, PAR, REQ, RST, TST, TXD, URH, URN, and (-) (if not used state)
7	dur	Float	Ghi lại tổng thời gian
8	sbytes	Integer	Số byte truyền từ nguồn
9	dbytes	Integer	Số byte truyền từ đích
10	sttl	Integer	Giá trị TTL nguồn
11	dttl	Integer	Giá trị TTL đích
12	sloss	Integer	Các gói nguồn được truyền lại hoặc bị hủy
13	dloss	Integer	Các gói đích được truyền lại hoặc bị hủy
14	service	nominal	http, ftp, smtp, ssh, dns, ftp-data ,irc, (-) nếu không sử dụng dịch vụ.
15	Sload	Float	Số bit nguồn mỗi giây
16	Dload	Float	Số bit đích mỗi giây
17	Spkts	integer	Số packet từ nguồn
18	Dpkts	integer	Số packet từ đích
19	swin	integer	Giá trị window advertisement của TCP nguồn
20	dwin	integer	Giá trị window advertisement của TCP đích

21	stcpb	integer	Số sequence number của gói tin TCP nguồn
22	dtcpb	integer	Số sequence number của gói tin TCP đích
23	smeansz	integer	Giá trị trung bình của kích thước luồng gói tin được truyền bởi nguồn
24	dmeansz	integer	Giá trị trung bình của kích thước luồng gói tin được truyền bởi đích
25	trans_depth	integer	Thể hiện độ sâu pipe vào kết nối giao dịch yêu cầu / phản hồi http
26	res_bdy_len	integer	Kích thước nội dung không nén thực tế của dữ liệu được truyền từ dịch vụ http của máy chủ.
27	Sjit	Float	Nguồn jitter (mSec)
28	Djit	Float	Đích jitter (mSec)
29	Stime	Timestamp	Thời gian bắt đầu
30	Ltime	Timestamp	Thời gian kết thúc
31	Sintpkt	Float	Thời gian gói tin bắt đầu chuyển đi (mSec)
32	Dintpkt	Float	Thời gian gói tin đến đích (mSec)
33	tcprtt	Float	Thiết lập kết nối TCP thời gian round-trip, tổng của ’synack’ và ’ackdat’.
34	synack	Float	Thời gian thiết lập kết nối TCP, thời gian giữa các gói SYN và các gói SYN_ACK.
35	ackdat	Float	Thời gian thiết lập kết nối TCP, thời gian giữa các gói SYN_ACK và các gói ACK.
36	is_sm_ips_ports	Binary	Nếu nguồn (1) và đích (3) địa chỉ IP bằng nhau và số cổng (2) (4) bằng nhau thì biến này nhận giá trị 1 khác 0
37	ct_state_ttl	Integer	Số cho mỗi trạng thái (6) theo phạm vi giá trị cụ thể cho nguồn / đích TTL (10) (11).
38	ct_flw_http_mthd	Integer	Số luồng có các phương thức như GET và POST trong dịch vụ http.
39	is_ftp_login	Binary	Nếu phiên ftp được truy cập bởi người dùng và mật khẩu thì 1 khác 0.
40	ct_ftp_cmd	integer	Không có luồng nào có lệnh trong phiên ftp.

41	ct_srv_src	integer	Số kết nối có cùng dịch vụ (14) và địa chỉ nguồn (1) trong 100 kết nối theo lần cuối (26).
42	ct_srv_dst	integer	Số kết nối có cùng dịch vụ (14) và địa chỉ đích (3) trong 100 kết nối theo lần cuối (26).
43	ct_dst_ltm	integer	Số kết nối của cùng một địa chỉ đích (3) trong 100 kết nối theo lần cuối (26).
44	ct_src_ ltm	integer	Số kết nối của cùng một địa chỉ nguồn (1) trong 100 kết nối theo lần trước (26).
45	ct_src_dport_ltm	integer	Không có kết nối nào có cùng địa chỉ nguồn (1) và cổng đích (4) trong 100 kết nối theo lần cuối (26).
46	ct_dst_sport_ltm	integer	Không có kết nối nào có cùng địa chỉ đích (3) và cổng nguồn (2) trong 100 kết nối theo lần cuối (26).
47	ct_dst_src_ltm	integer	Không có kết nối nào của cùng một nguồn (1) và địa chỉ đích (3) trong 100 kết nối theo lần cuối (26).
48	attack_cat	nominal	Tên của từng loại tấn công. Trong bộ dữ liệu này, chín loại, ví dụ: Fuzzers, Analysis, Backdoors, DoS Exploits, Generic, Reconnaissance, Shellcode and Worms
49	Label	binary	0: bình thường và 1: tấn công

Với tổng số bản ghi dữ liệu là 2 triệu, và 540.044 bản ghi được lưu trữ trong bốn tệp CSV. Cụ thể ở đây là UNSW-NB15_1.csv, UNSW-NB15_2.csv, UNSW-NB15_3.csv và UNSW-NB15_4.csv.

Mọi tính năng được mô tả trong tập tin NUSW-NB15_features.csv. Ở đây các tính năng sẽ có nhiều loại khác nhau: *Integer, Float, Binary, Nominal và Timestamp*. Có thể lấy một ví dụ về tính năng số nguyên là kiểu *dloss*, trong đó sẽ đề cập đến các gói đích được truyền lại hoặc hủy bỏ. Và một số tính năng kiểu *float* là *Sload*, điều này có nghĩa là số bit đích trên mỗi giây. Còn lại một số tính năng khác có giá trị *binary* 1 hoặc 0. Trong đó, số *integer, float* và *timestamp* là các giá trị dữ liệu liên tục

không giới hạn. Còn tính năng danh tính thì đề cập đến việc phân loại ra danh mục theo đặc điểm của dữ liệu.

Bảng 2.4: Những danh mục trong tập dữ liệu UNSW-NB15

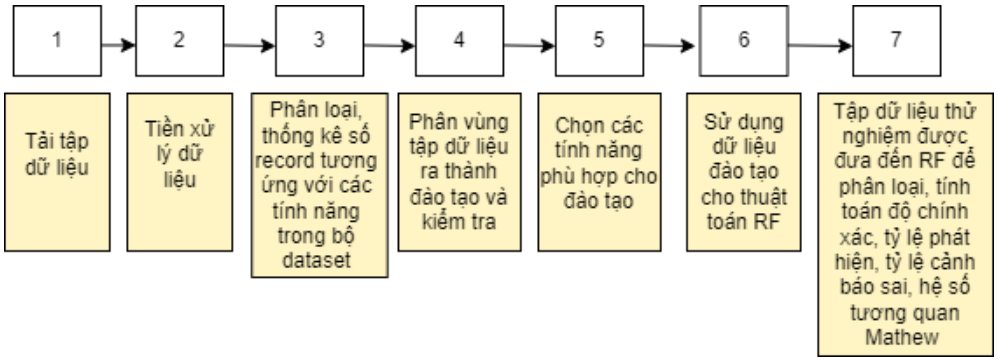
Danh mục	Số lượng
Normal	2218 761
Fuzzers	24 246
Analysis	2 677
Backdoors	2 329
DoS	16 353
Exploits	44 525
Generic	215 481
Reconnaissance	13 987
Shellcode	1 511
Worms	174

- Có một số loại tấn công xuất hiện rất ít điểm dữ liệu ví dụ như worms ở bảng trên.

- Dữ liệu có thể được phân chia để có được 10% cho thử nghiệm và 90% còn lại cho đào tạo, tương ứng với mỗi danh mục trên. Vì các danh mục này vẫn sẽ rất mất cân bằng.

- Giải pháp ở đây là lấy mẫu dữ liệu bằng cách chọn một phần dữ liệu trên mỗi danh mục trên. Việc loại bỏ bớt dữ liệu của một số lớp sẽ chiếm tỉ lệ dữ liệu cao để mang lại ít ảnh hưởng cho thuật toán.

Các bước mà thuật toán Random Forest xử lý tập dữ liệu mẫu như sau:



Hình 2.10: Các bước RF xử lý tập dữ liệu mẫu

2.5 Kết luận Chương 2

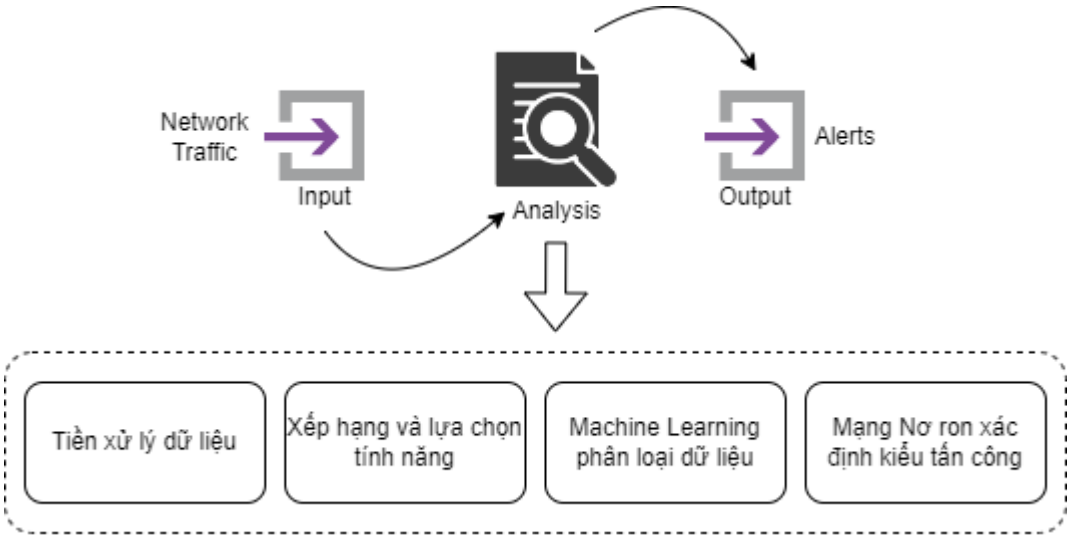
Nội dung Chương 2 tập trung trình bày các kỹ thuật học máy cơ bản sử dụng trong các hệ thống phát hiện xâm nhập. Đồng thời, trong Chương 2 các thông tin, phân tích và lựa chọn thuật toán học máy, tập dữ liệu mẫu cũng được mô tả chi tiết để hỗ trợ cho việc thực hiện thử nghiệm và đánh giá hiệu quả khi ứng dụng thuật toán học máy trong phát hiện xâm nhập trên IoT gateway.

CHƯƠNG 3: THỬ NGHIỆM HỆ THỐNG IDS TRÊN CÁC THIẾT BỊ IoT GATEWAY

3.1 Mô hình phát hiện xâm nhập trên IoT gateway

Mô hình phát hiện xâm nhập cho IoT gateway dựa trên học máy phải thực hiện phân tích chuyên sâu về lưu lượng mạng, gồm một số thành phần như tiền xử lý dữ liệu, xếp hạng, lựa chọn tính năng, phân lớp học máy và nhận dạng tấn công.

Ở đây thành phần tiền xử lý dữ liệu sẽ chịu trách nhiệm xử lý trước những dữ liệu liên quan đến các hoạt động chuyển đổi cũng như chuẩn hóa để mang đến dữ liệu cho thành phần xếp hạng, lựa chọn tính năng.



Hình 3.1: Mô hình các bước thực hiện cho hệ thống phát hiện xâm nhập

Bước 1: Tiền xử lý dữ liệu.

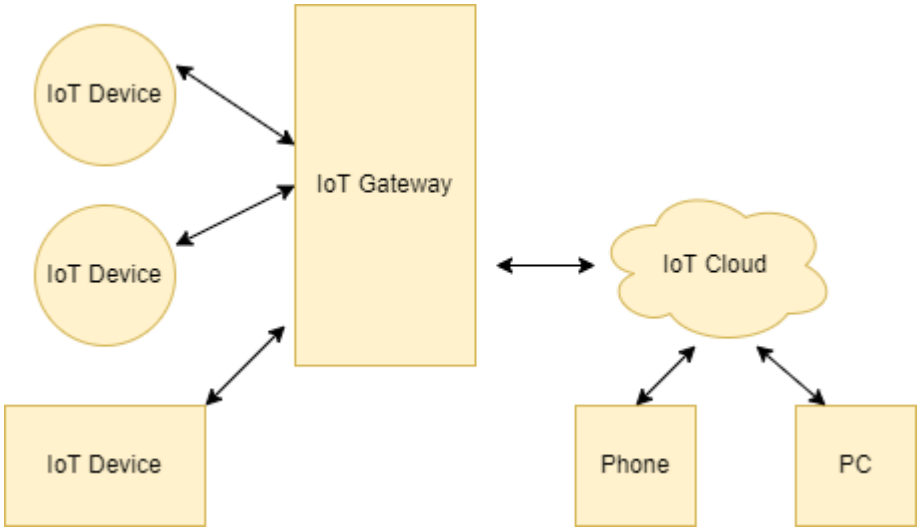
Bước 2: Xếp hạng và lựa chọn tính năng khắc phục dữ liệu đầu vào để tránh sự chênh lệch dẫn tới sự sai lệch kết quả.

Bước 3: Áp dụng học máy phân loại dữ liệu tấn công hay dữ liệu bình thường.

Bước 4: Xác định kiểu tấn công dựa vào mạng Nơ ron.

Mô hình để phân loại các mẫu tập dữ liệu UNSW-NB15 được phát triển bằng phương pháp sử dụng một mạng nơ ron ngẫu nhiên và chuyển tiếp nguồn cấp dữ liệu. Tại đây, hệ thống sẽ sử dụng thuật toán *Random Forest* để phân loại đâu là dữ liệu bình thường và đâu là dữ liệu độc hại. Và từ đây dữ liệu tấn công sẽ tiếp tục được sử

dụng để huấn luyện mạng nơ ron để phân loại thành những loại tấn công khác nhau. Ở đây luận văn đưa ra giải pháp ứng dụng hệ thống phát hiện xâm nhập ở cả hai điểm là IoT Gateway và IoT Cloud.



Hình 3.2: Mô hình mô tả các thành phần trong hệ thống IoT

Cách thành phần chính trong hệ thống IoT bao gồm:

- ***IoT Gateway (IID):*** Thiết bị đóng vai trò giống với một thiết bị trung gian để trước khi các *IoT device* kết nối với Internet thì phải qua IoT Gateway, nó giống như thiết bị Hub. Trên Thế giới hiện nay đã sản xuất những thiết bị IoT Gateway cũng như IoT Hub có tích hợp rất nhiều các module kết nối để có thể hỗ trợ nhiều chủng loại thiết bị IoT có thể điều hướng, định tuyến cũng như chuyển mạch lưu lượng.

- ***IoT device:*** Là cách gọi chung của thiết bị IoT như tủ lạnh thông minh, điện thoại, máy tính cá nhân hay các thiết bị cảm biến chống cháy…

- ***IoT Cloud:*** Đây là một nền tảng cũng như là một hệ thống ứng dụng cung cấp chức năng cho người dùng để quản trị, điều khiển và theo dõi thiết bị IoT từ xa bằng cách sử dụng các thiết bị như smart phone có cài app quản trị.

- ***Triển khai IDS cho IoT Gateway:*** Những thiết bị IoT trong mạng có thể di động cũng như phân phối theo địa lý trong một phạm vi được mạng xác định và bộ định tuyến biên sẽ hoạt động định tuyến. Ở đây các thiết bị giao tiếp bằng các giao thức truyền thông không dây có thể lấy ví dụ như *Wi-Fi, Bluetooth BLE, ZigBee* hoặc các giao thức truyền thông độc quyền như CoAP hoặc *Thread*. IDS sẽ được đề xuất

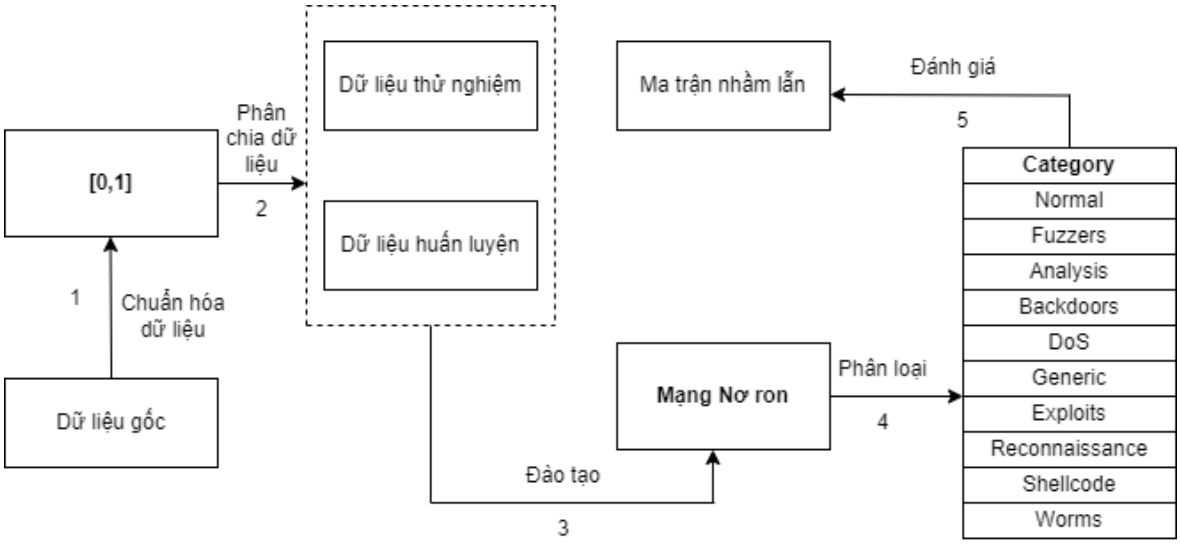
giống như một thiết bị độc lập có thể dễ dàng tích hợp vào mạng IoT. Trong đó, IID hoạt động ở chế độ quảng bá, giám sát và phân tích lưu lượng mạng cũng như sử dụng ảo hóa mạng nhằm kết nối với bộ định tuyến và các thiết bị IoT khác.

- ***Triển khai IDS cho IoT Cloud:*** Những thiết bị khi truy cập *IoT Core* sẽ được IDPS trích xuất dữ liệu *Spanport* qua thiết bị *DMZ Switch*. IDPS triển khai nhiều kỹ thuật học máy phát hiện ra sự tấn công của hacker để chiếm quyền điều khiển của hệ thống.

3.2 Kiến trúc hệ thống phát hiện xâm nhập cho IoT gateway dựa trên học máy

3.2.1 Kiến trúc giải pháp IDS sử dụng mạng Nơ ron

Kiến trúc tổng thể sẽ có những phương thức được sử dụng trong đó bao gồm tiền xử lý dữ liệu, đào tạo và huấn luyện qua mạng Nơ ron cũng như đánh giá qua ma trận nhầm lẫn.



Hình 3.3: Kiến trúc tổng thể khi dùng mạng Nơ ron

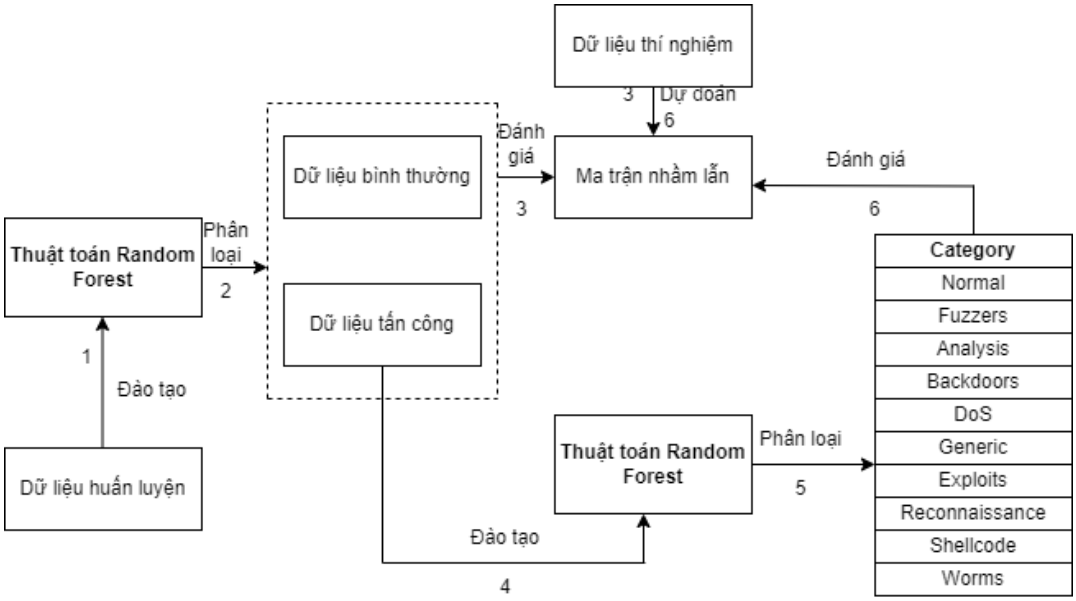
Các bước thực hiện như sau:

- Bước 1: Dữ liệu gốc làm input đầu vào cho thuật toán RF, chuẩn hóa dữ liệu về dạng [0, 1].
- Bước 2: Phân chia dữ liệu thành dữ liệu thử nghiệm và dữ liệu huấn luyện.
- Bước 3: Mạng Nơ ron tiến hành đào tạo.

- Bước 4: Mạng Nơ ron phân loại dữ liệu tấn công thành 10 danh mục bao gồm: *Normal, Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, Worms*.

- Bước 5: Cuối cùng ma trận nhầm lẫn sẽ đánh giá khả năng phân loại của mạng Nơ ron.

3.2.2 Kiến trúc giải pháp IDS sử dụng Random Forest



Hình 3.4: Kiến trúc tổng thể khi dùng Random Forest

Phân tích các bước thực hiện:

- Bước 1: Dữ liệu huấn luyện chính là input đầu vào cho thuật toán RF.

- Bước 2: Thuật toán RF phân loại dữ liệu thành 2 nhóm là: dữ liệu bình thường và dữ liệu tấn công.

- Bước 3: Dữ liệu thử nghiệm và kết quả phân loại ở bước 2 được Ma trận nhầm lẫn đánh giá dự đoán của thuật toán *Random Forest*.

- Bước 4: Tại bước này RF sử dụng dữ liệu tấn công để làm dữ liệu huấn luyện.

- Bước 5: RF sẽ tiếp tục phân loại dữ liệu tấn công thành 10 danh mục bao gồm: *Normal, Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode* và *Worms*.

- Bước 6: Ma trận nhầm lẫn tiếp tục đánh giá khả năng phân loại của RF ở bước 5 dựa vào bộ dữ liệu thử nghiệm.

3.3 Thiết lập thử nghiệm phát hiện xâm nhập dựa trên thuật toán Random Forest và mạng Nơ ron

Quá trình thực hiện khi sử dụng mạng Nơ ron như sau:

- Bước 1: Tiền xử lý dữ liệu: Phân loại các thuộc tính → DictVectorizer các thuộc tính nominal → Gộp với các thuộc tính còn lại → Normalize tạo thành 294 tính năng.

- Bước 2: Thực hiện chia dữ liệu để đào tạo và kiểm tra bộ dữ liệu.

- Bước 3: Tiếp tục huấn luyện mạng Nơ ron với dữ liệu huấn luyện nhằm dự đoán, gán nhãn cho dữ liệu là dữ liệu tấn công hay dữ liệu bình thường.

- Bước 4: Dự đoán danh mục tấn công cùng với dữ liệu thử nghiệm.

- Bước 5: Áp dụng các số liệu hiệu suất để đo lường mức độ hệ thống tổng quát hóa dữ liệu.

Quá trình thực hiện khi sử dụng thuật toán Random Forest như sau:

- Bước 1: Tiền xử lý dữ liệu: Phân loại các thuộc tính → DictVectorizer các thuộc tính nominal → Gộp với các thuộc tính còn lại → Normalize tạo thành 294 tính năng.

- Bước 2: Thực hiện chia dữ liệu để đào tạo và kiểm tra bộ dữ liệu.

- Bước 3: Tiếp tục huấn luyện thuật toán Random Forest với dữ liệu huấn luyện nhằm dự đoán, gán nhãn cho dữ liệu là dữ liệu tấn công hay dữ liệu bình thường mà không cần kết hợp sử dụng tính năng.

- Bước 4: Sử dụng thuật toán Random Forest để tạo các tính năng cho dữ liệu thử nghiệm.

- Bước 5: Dự đoán danh mục tấn công cùng với dữ liệu thử nghiệm (với các tính năng được tạo ra ở bước 4).

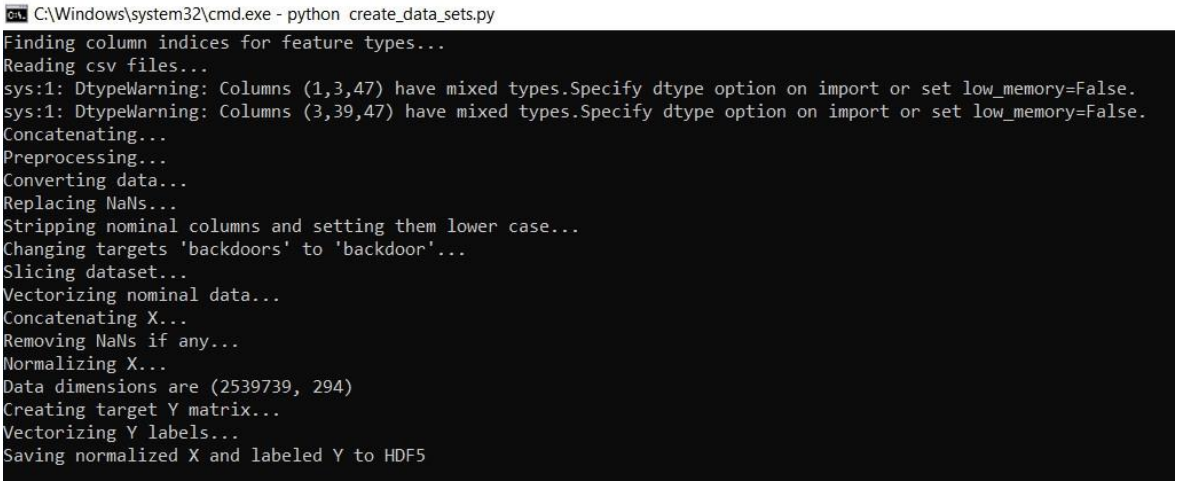
- Bước 6: Tiếp theo sẽ áp dụng các số liệu hiệu suất để đo lường mức độ hệ thống tổng quát hóa dữ liệu.

Dữ liệu sẽ được đọc từ các tệp CSV với Pandas. Số lượng lớn dữ liệu mạng cũng là một thách thức không nhỏ trong mỗi bước xử lý.

3.3.1 Tiền xử lí dữ liệu

Tiền xử lý được chia thành hai giai đoạn. Trong đó giai đoạn đầu tiên được mô tả bên dưới (tệp pre_process.py) như sau:

- Đọc dữ liệu.
- Chia dữ liệu theo các loại tính năng.
- Chuyển đổi dữ liệu thành từng loại cho phù hợp.
  - Thay thế NaN, bằng 0 xóa điểm dữ liệu.
  - Sau đó cắt dữ liệu danh mục từ các khoảng trắng thừa, đặt chữ thường, vector hóa.
- Bình thường hóa dữ liệu giữa [0, 1].
- Lưu vào HDF5.



Hình 3.5: Tiền xử lý tập dữ liệu

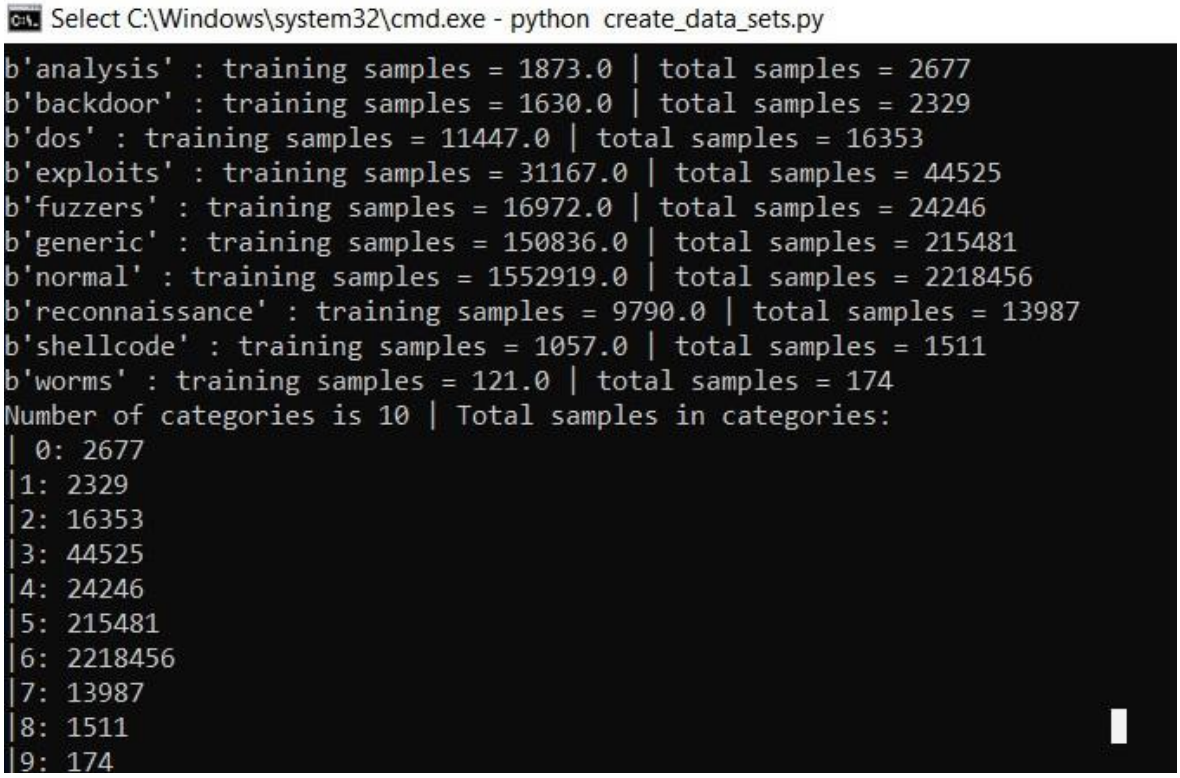
Các tính năng timestamp đã bị xóa, bởi có thể được sử dụng trong mạng nơ ron chuyển tiếp. Đồng thời việc thiết kế một mạng lưới nơ ron hồi qui sẽ nằm ngoài phạm vi của luận văn.

3.3.2 Phân tách dữ liệu

Giai đoạn thứ hai (tệp create_data_sets.py) tiến hành phân tách dữ liệu thành các phần huấn luyện và thử nghiệm. Tổng số mẫu sử dụng: 2.539.739 mẫu. Dữ liệu

có thể được chia để có được 70% cho huấn luyện (1.777.812 mẫu) và 30% (761.927 mẫu) cho thử nghiệm với mỗi danh mục, vì các danh mục cũng sẽ rất mất cân bằng. Được mô tả như sau:

- Đọc dữ liệu đã xử lý trước từ HDF5.
- Chia dữ liệu thành các tập dữ liệu thử nghiệm và huấn luyện không chồng chéo.

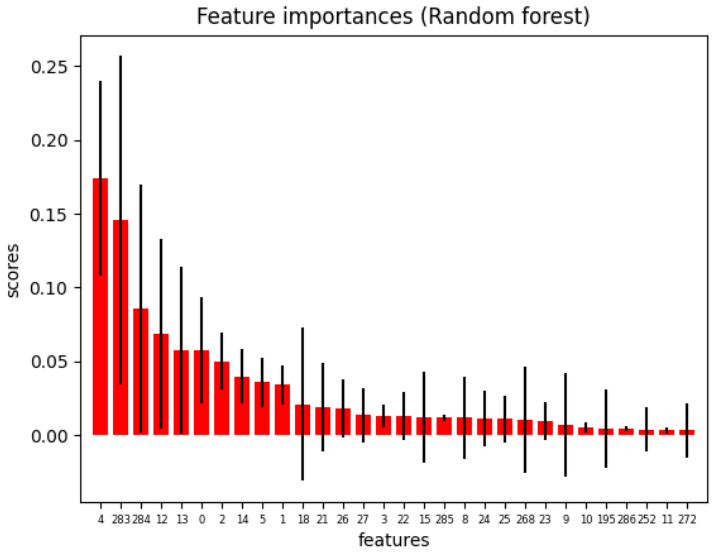


Hình 3.6: Khởi chạy ứng dụng chia dữ liệu

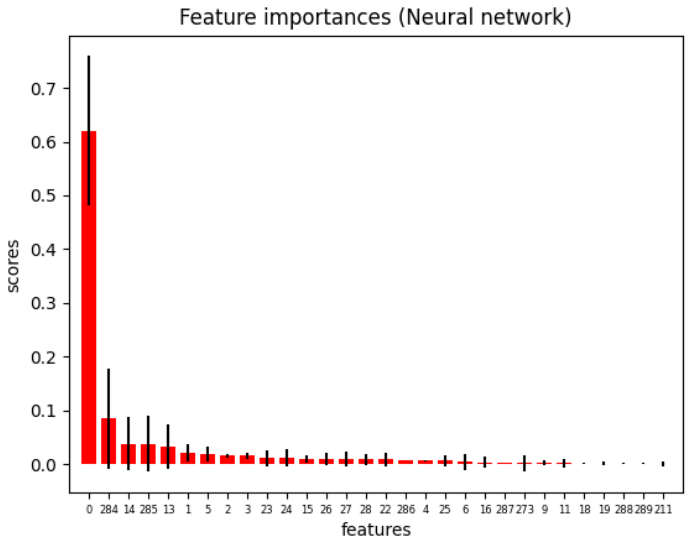
- Tìm các tính năng quan trọng cho cả hai mô hình.
- Chọn các tính năng quan trọng nhất.
- Lưu tập dữ liệu vào HDF5.

Tiếp theo đến bước giảm tính năng. Dữ liệu có điểm này 294 tính năng sau khi vector hóa. ExtraTreesClassifier đã được sử dụng để chọn 10 tính năng quan trọng nhất cho phân loại tấn công hay bình thường (Random Forest) và 25 tính năng quan trọng nhất cho mạng Nơ ron. Hình 3.7 và 3.8 sẽ minh họa cho các tính năng nào có tầm quan trọng trong cả hai nhiệm vụ phân loại.

Chia từng loại dữ liệu thành dữ liệu huấn luyện và thử nghiệm cho cả thuật toán *Random Forest* và mạng Nơ ron (NN). Giá trị điểm dữ liệu chỉ nhằm mục đích minh họa.



Hình 3.7: Mức độ quan trọng của tính năng phân loại tấn công (Random Forest)



Hình 3.8: Mức độ quan trọng của tính năng phân loại tấn công (mạng Nơ ron)

Trong đó, giá trị trục features là các chỉ số tính năng, giá trị trục scores là chỉ mức độ quan trọng của các tính năng.

Đối với trường hợp phát hiện tấn công, Scikit-Learn’s Random Forest Classifier sử dụng 31 công cụ ước tính (cây quyết định) và chất lượng của việc phân

tách cây quyết định được đo bằng mức tăng thông tin ("entropy"). Các cài đặt khác được để mặc định. Chỉ có số lượng cây (estimators) cài đặt giá trị là 31 và chất lượng của việc phân tách cây quyết định (criterion) cài đặt tham số là "entropy".

3.4 Đánh giá kết quả thử nghiệm

Chương 3 này đã mô tả các kết quả cho các trường hợp phân loại tấn công và phát hiện tấn công. Luận văn chia dữ liệu thành tập huấn luyện và kiểm tra. Sử dụng mạng Nơ ron và thuật toán Random Forest để phân loại dữ liệu thành dữ liệu mạng bình thường hoặc dữ liệu tấn công, từ đó đánh giá hiệu quả.

Giải thích ý nghĩa của các giá trị trong những kết quả phân loại dưới đây:

- Lớp 0 có nghĩa là bình thường không phải là một cuộc tấn công. Lớp 1 là một cuộc tấn công.

- TP (True Positive): Chính là số lượng các mẫu thuộc lớp tấn công được phân loại chính xác vào lớp tấn công.

- FP (False Positive): Là số lượng các mẫu không thuộc lớp tấn công nhưng đã bị phân loại nhầm vào lớp tấn công.

- TN (True Negative): Là số lượng các mẫu không thuộc lớp tấn công và đã được phân loại đúng.

- FN (False Negative): Là số lượng các mẫu thuộc lớp tấn công nhưng bị phân loại nhầm vào các lớp không phải là lớp tấn công.

- Đầu tiên là về độ chính xác của mô hình phát hiện được tính theo công thức:

$$\text{Acc} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \tag{3.1}$$

- Precision: Là giá trị thể hiện việc trong số các mẫu được mô hình phân loại vào lớp tấn công sẽ có bao nhiêu mẫu thực sự thuộc về lớp tấn công.

$$\text{Precision} = \frac{TP}{TP+FP} \tag{3.2}$$

- Recall: Là giá trị giúp nhận biết có bao nhiêu mẫu thực sự ở lớp tấn công được mô hình phân lớp đúng trong mọi mẫu thực sự ở lớp tấn công.

$$\text{Recall} = \frac{TP}{TP+FN} \tag{3.3}$$

Precision và Recall có giá trị trong khoảng [0,1], trong đó nếu hai giá trị này càng gần với 1 thì mô hình sẽ càng chính xác. Precision càng cao sẽ đồng nghĩa với việc các mẫu được phân loại càng chính xác. Đồng thời, Recall càng cao càng thể hiện cho việc ít bỏ sót các dữ liệu đúng.

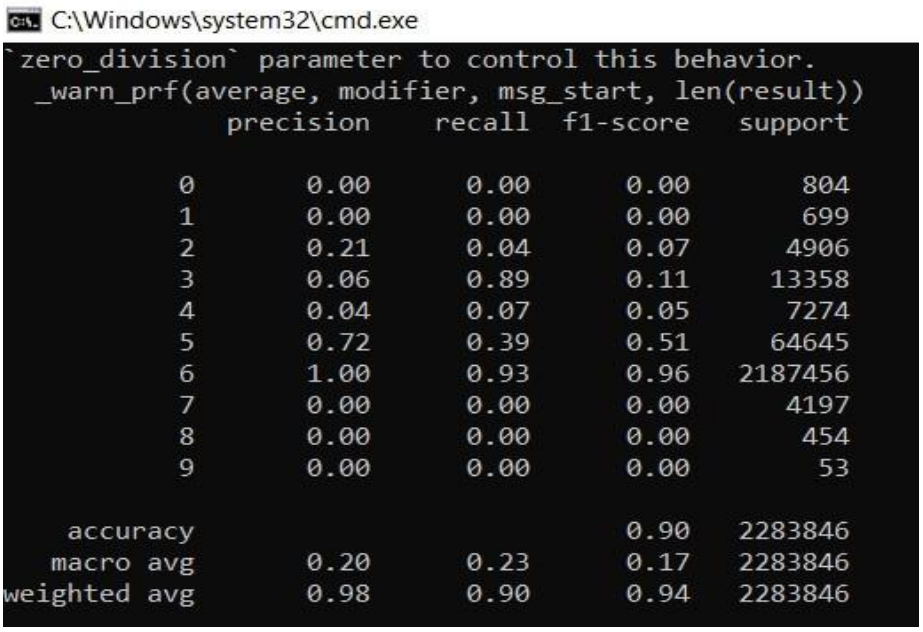
- F1-Score: Là giá trị sử dụng để đánh giá cùng lúc cả Precision và Recall. Được tính theo công thức:

$$F_\beta = (1 + \beta^2)\frac{\text{precision.recall}}{\beta^2.\text{precision} + \text{recall}} \quad (3.4)$$

β là giá trị mô tả độ mạnh của Recall so với độ chính xác của Precision.

- Support: Là số lần xuất hiện của mỗi lớp trong các mẫu được mô hình phân loại vào lớp tấn công hoặc không tấn công.

3.4.1 Kết quả khi sử dụng mạng Nơ ron

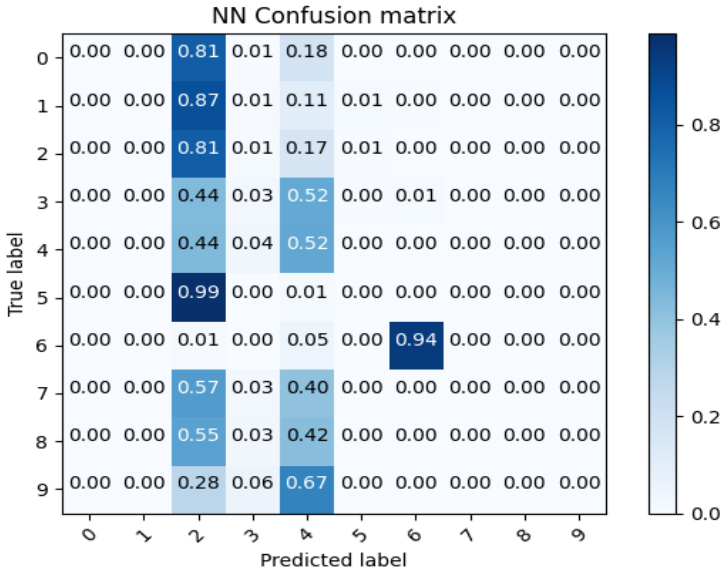


Hình 3.9: Kết quả phân loại tấn công (mạng Nơ ron)

Kết quả cho thấy mạng Nơ ron hoạt động khá ổn với dữ liệu này, độ chính xác 0, 90 cho dữ liệu cuộc tấn công. Các lớp tấn công được thể hiện bằng các số 0 hay 9. Số 6 là một điểm dữ liệu bình thường của điểm số và phần còn lại của các số là các loại tấn công khác nhau.

Lớp 6 hầu như luôn được dự đoán chính xác. Precision là 1.0 và Recall là 0,93. Điều này cho thấy rằng lớp 6 (dữ liệu bình thường) có các tính năng riêng biệt trong tập dữ liệu.

Ma trận nhầm lẫn đánh giá hiệu quả thuật toán phát hiện tấn công như sau



Hình 3.10: Ma trận nhầm lẫn để phân loại tấn công (NN)

Ma trận nhầm lẫn cho thấy các dự đoán chủ yếu thuộc về lớp 2 và 4. Do đó, các trường hợp của lớp 2 và 4 có điểm Recall khá cao (0,81 cho lớp 2 và 0,52 cho lớp 4). Hầu hết các điểm dữ liệu được dự đoán là trường hợp của lớp 2, vì vậy độ chính xác thấp.

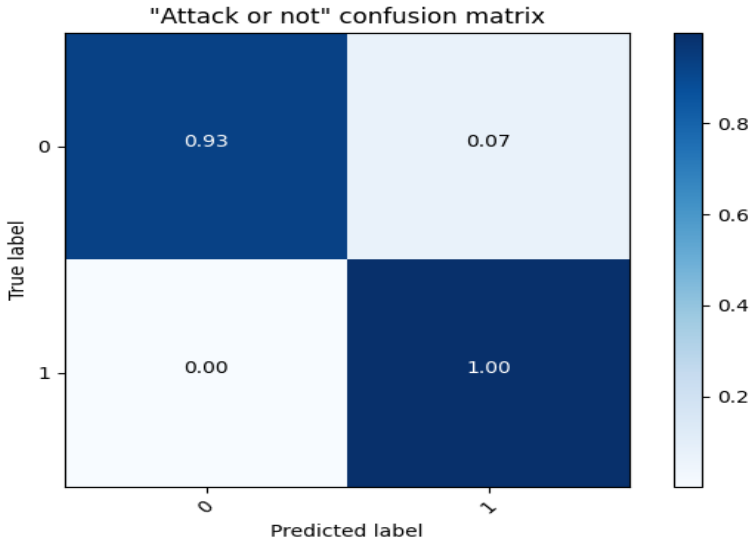
3.4.2 Kết quả khi dùng thuật toán Random Forest

```
C:\Windows\system32\cmd.exe - python  fit_neural.py
Training RandomForestClassifier for "attack or not" labels...
Testing accuracy...
0.9297181158449388
              precision    recall  f1-score   support

         0.0       1.00      0.93      0.96   2187456
         1.0       0.38      1.00      0.55     96390

    accuracy                           0.93   2283846
   macro avg       0.69      0.96      0.75   2283846
weighted avg       0.97      0.93      0.94   2283846
```

Hình 3. 11: Kết quả phân loại tấn công (Random Forest)

Qua kết quả ở trên ta có thể thấy Random Forest Classifier hoạt động khá tốt với dữ liệu này. Điểm số cũng sẽ được cải thiện hơn nữa sau khi giảm tính năng ở cùng một phân loại. Điểm Recall là 0.93 cho lớp 0 và 1.00 cho lớp 1. Đồng thời, Precision cho lớp 1 là 0,38 thấp hơn lớp 0 là 1.00.

Ma trận nhầm lẫn đánh giá hiệu quả thuật toán phát hiện tấn công:



Hình 3. 12: Ma trận nhầm lẫn biểu thị kết quả (Random Forest)

- TP (True Positive): 96261
- FP (False Positive): 129
- TN (True Negative): 2027072
- FN (False Negative): 160384

Sau quy trình xử lý tập dữ liệu mẫu để thực hiện đánh giá hiệu quả khi sử dụng thuật toán học máy Random Forest có thể thấy rằng thuật toán Random Forest hoạt động hiệu quả khá tốt trong phân loại tấn công.

3.5 Kết luận chương 3

Chương 3 trình bày ứng dụng triển khai giải pháp phát hiện xâm nhập dựa trên mạng Nơ ron và thuật toán Random Forest. Trong chương 3 đã nêu ra mô hình phát hiện xâm nhập trong IoT gateway, các kiến trúc hệ thống phát hiện xâm nhập, tiến hành thiết lập thử nghiệm và đưa ra kết quả đánh giá cuối cùng. Các kết quả thử nghiệm cho thấy hệ thống IDS ứng dụng giải pháp mạng Nơ ron có thể đạt độ chính xác 90% và ứng dụng thuật toán *Random Forest* thì có được độ chính xác lên đến 93%.

KẾT LUẬN

Hệ thống phát hiện xâm nhập hiện đang là một trong những giải pháp được quan tâm hàng đầu hiện nay nhằm bảo vệ linh hoạt, hiệu quả trước vô vàn cuộc xâm nhập trái phép trên Internet nhắm tới các thiết bị IoT. Mặc dù còn gặp nhiều thách thức do các nguy cơ tấn công bảo mật đều khá phức tạp và khó có thể đoán trước được, các hệ thống phát hiện xâm nhập ứng dụng kỹ thuật học máy đã và đang cho thấy nhiều tiềm năng và thu hút được nhiều sự quan tâm, đầu tư và nghiên cứu. Trong tình hình đó, việc nghiên cứu, tìm hiểu và nắm bắt các giải pháp phát hiện xâm nhập hiệu quả cho các thiết bị IoT gateway dựa trên công nghệ học máy là rất cấp thiết.

Trong khuôn khổ luận văn này, học viên tập trung nghiên cứu vấn đề an toàn thông tin cho các thiết bị IoT gateway và các kỹ thuật phát hiện xâm nhập ứng dụng kỹ thuật học máy trong IoT, trên cơ sở đó, xây dựng và thử nghiệm giải pháp phát hiện xâm nhập sử dụng công nghệ học máy trong kịch bản ứng dụng cho các thiết bị IoT gateway. Các nội dung chính đạt được trong luận văn bao gồm:

- Nghiên cứu tổng quan về Internet of things, các thiết bị IoT Gateway, các kỹ thuật mà một hệ thống IDS truyền thống sử dụng để phát hiện xâm nhập cũng như lý thuyết về các thuật toán học máy ứng dụng trong phát hiện xâm nhập: KNN, SVM, *Naive Bayes*, *J48 Decision Tree*. Đặc biệt là thuật toán *Random Forest* và mạng Nơ ron.
- Nghiên cứu về thuật toán học máy ứng dụng tiếp cận trong phát hiện xâm nhập, đưa ra những giải pháp phát hiện xâm nhập ứng dụng cho IoT Gateway cũng như đề xuất mô hình ứng dụng học máy trong phát hiện xâm nhập. Đồng thời nghiên cứu về mô hình, kiến trúc của *Random Forest* và mạng Nơ ron nhằm áp dụng vào hệ thống phát hiện xâm nhập giúp cải thiện tỷ lệ phát hiện chính xác cũng như giảm thiểu tỷ lệ cảnh báo nhầm của một hệ thống IDS thông thường.
- Ứng dụng triển khai thử nghiệm giải pháp phát hiện xâm nhập sử dụng mạng Nơ ron và thuật toán *Random Forest* trên tập dữ liệu UNSW-NB15. Thông qua

các bước xây dựng mô hình kiến trúc tổng thể và thiết lập thử nghiệm đã cho thấy hệ thống IDS ứng dụng giải pháp mạng Nơ ron có thể đạt độ chính xác tương đối khoảng 90% và nếu ứng dụng thuật toán *Random Forest* thì có thể đạt độ chính xác lên đến 93%.

Tuy nhiên, vì thời gian nghiên cứu ngắn cũng như phạm vi của lĩnh vực nghiên cứu rộng, nên luận văn này vẫn còn nhiều vấn đề chưa đề cập và chưa giải quyết triệt để. Trong thời gian tới, em sẽ tiếp tục nghiên cứu thử nghiệm kết hợp với các giải pháp phù hợp vào hệ thống phát hiện xâm nhập và cố gắng đưa ứng dụng vào thực tiễn.

DANH MỤC TÀI LIỆU THAM KHẢO

[1] A. Hameed and A. Alomary, "Security Issues in IoT: A Survey," 2019 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), pp. 1-5, 2019.

[2] Jorma Laaksonen, Erkki Oja, (1996). Classification with learning k-Nearest Neighbors.

[3] Maheshkumar Sabhnani Gursel Serpen, (2015), “Application of Machine Learning Algorithms to KDD Intrusion Detection Dataset within Misuse Detection Context”.

[4] Markus Goldstein, Seiichi Uchida, (2013). “Behavior Analysis Using Unsupervised Anomaly Detection”.

[5] Michael Nielsen (2018), Neural Networks and Deep Learning.

[6] Mitchell, T. (1997). Machine Learning.

[7] Ranzhe Jing, Yong Zhang. A View of Support Vector Machines Algorithm on Classification Problems. International Conference on Multimedia Communications.

[8] Syed Ali Raza Shah, Biju Issac, “Performance comparison of intrusion detection systems and application of machine learning to Snort system,” Future Generation Computer Systems, Vol. 80, pp. 157-170, 2018.

[9] Swain, Philip H., Hans Hauska. (1977). The Decision Tree classifier: design and potential. IEEE Transactions on Geoscience Electronics.

[10] Nguyễn Ngọc Điệp, Nguyễn Thị Thanh Thuỷ, “Nâng cao khả năng phát hiện xâm nhập mạng sử dụng mạng CNN”, Tạp chí khoa học công nghệ thông tin và truyền thông, số 4B (CS.01), pp. 61-68, 2020.