

**UNIVERSITY OF INFORMATION TECHNOLOGY
FACULTY OF SOFTWARE ENGINEERING**

-----[?] [?] [?] [?]-----



UIT

**SEMINAR CÁC VẤN ĐỀ HIỆN ĐẠI CỦA CÔNG
NGHỆ PHẦN MỀM**

**TOPIC: IOT CYBER SECURITY WITH MACHINE
LEARNING**

Giảng viên hướng dẫn: Nguyễn Tấn Toàn

Sinh viên thực hiện:

20521486_Đặng Bá Kiên

19522035_Nguyễn Đặng Hữu Phúc

TPHCM, ngày 17 tháng 12 năm 2023

Mục lục

ABSTRACT5

1 INTRODUCTION6

1.1 OVERVIEW7

1.2 PHÂN TÍCH VỀ VẤN ĐỀ BẢO MẬT CỦA IOT10

1.2.1 VẤN ĐỀ XÁC THỰC10

1.2.2 YÊU CẦU VỀ AN NINH VÀ BẢO MẬT TRONG IOT11

1.3 ĐỘNG LỰC12

1.4 ĐÓNG GÓP CỦA BÀI BÁO CÁO15

1.4.1 ĐẶT VẤN ĐỀ16

1.4.2 GIẢI PHÁP18

2 BACKGROUND20

2.1 IOT LÀ GÌ22

2.1.1 GIÁ TRỊ CỦA IOT23

2.1.2 KIẾN TRÚC CỦA IOT25

2.1.3 CÁCH THỨC HOẠT ĐỘNG CỦA IOT27

2.2 CÁC RỦI RO VÀ THÁCH THỨC CỦA IOT29

2.2.1 BẢO MẬT29

2.2.2 CHÍNH SÁCH30

2.3 DISTRIBUTED DENIAL-OF-SERVICE31

2.3.1 BOT31

2.3.2 BOTNET31

2.3.3 BOTMASTER32

2.3.4 CÁCH THỨC TẤN CÔNG CỦA BOTNET32

2.3.5 TẠI SAO BOTNET ĐƯỢC TẠO RA35

2.4 MACHINE LEARNING35

2.4.1 SUPERVISED LEARNING35

2.4.2 UNSUPERVISED LEARNING36

2.4.3 MACHINE LEARNING APPROACH37

2.4.3.1 LOGISTICS REGRESSION 37

- 2.4.3.2 DECISION TREE 38
- 2.4.3.3 RANDOM FOREST 38
- 2.4.3.4 MULTI LAYER PERCEPTRON 39
- 2.4.3.5 XGBOOST 39
- 2.4.3.6 LIGHTGBM 39
- 2.4.3.7 HISTGRADIENTBOOSTING 40

3 PHƯƠNG PHÁP NGHIÊN CỨU 40

3.1 SYSTEM OVERVIEW 40

- 3.1.1 UNSW-NB15 DATASET 40
 - 3.1.1.1 CẤU HÌNH TESTBED SỬ DỤNG CÔNG CỤ IXIA 41
 - 3.1.1.2 ARCHITECTURE FRAMEWORK 42
- 3.1.2 SYSTEM ARCHITECTURE 47
- 3.1.3 MACHINE LEARNING TECHNIQUES 48
 - 3.1.3.1 LOGISTICS REGRESSION 48
 - 3.1.3.2 DECISION TREE 51
 - 3.1.3.3 RANDOM FOREST 53
 - 3.1.3.4 MULTI LAYER PERCEPTRON 55
 - 3.1.3.5 XGBOOST 57
 - 3.1.3.6 LIGHTGBM 58
 - 3.1.3.7 HISTGRADIENTBOOSTING 60

3.2 CÁC BƯỚC TRIỂN KHAI 60

- 3.2.1 TRÍCH XUẤT ĐẶC TRƯNG 60
- 3.2.2 TIỀN XỬ LÝ DỮ LIỆU 61
- 3.2.3 CHỌN ĐẶC TRƯNG 62
- 3.2.4 XÂY DỰNG MÔ HÌNH 62
- 3.2.5 ĐÁNH GIÁ MÔ HÌNH 62

4 KẾT QUẢ VÀ THẢO LUẬN 63

4.1 PHÂN LOẠI TẤN CÔNG 63

4.2 SO SÁNH CÁC THUẬT TOÁN BẰNG KỸ THUẬT CROSS-VALIDATION 65

4.3 SO SÁNH CÁC THUẬT TOÁN SỬ DỤNG F1-SCORE 66

4.4 SO SÁNH CÁC THUẬT TOÁN SỬ DỤNG ACCURACY 69

4.5 SO SÁNH CÁC THUẬT TOÁN SỬ DỤNG AUC71

5 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 74

5.1 KẾT LUẬN 74

5.2 HƯỚNG PHÁT TRIỂN 75

6 ỨNG DỤNG ML VÀO THỰC TẾ 76

7 TÀI LIỆU THAM KHẢO 77

ABSTRACT

Internet of Things (IoT) là công nghệ đang được phát triển cung cấp sự đơn giản và lợi ích trong việc trao đổi dữ liệu với các thiết bị khác sử dụng đám mây hoặc mạng không dây. Tuy nhiên, những thay đổi và trong môi trường làm việc của IoT đang làm cho các hệ thống IoT dễ bị tấn công mạng từ đó có thể dẫn đến các vụ xâm nhập. Những tác động của những xâm nhập này có thể dẫn đến tổn thất rất lớn về kinh tế. Bài báo cáo này chủ yếu tập trung vào hệ thống của IoT, các phương pháp dựa trên máy học và những khó khăn mà các thiết bị hoặc hệ thống IoT phải đối mặt sau khi xảy ra một cuộc tấn công. Các phương pháp dựa trên học được xem xét bằng các loại tấn công mạng khác nhau, chẳng hạn như tấn công từ chối dịch vụ (DoS), tấn công từ chối dịch vụ phân tán (DDoS), tấn công botnet. Để phát hiện các cuộc tấn công botnet, bài báo cáo này đã sử dụng một loạt các tính toán AI, sau đó đánh giá xem chúng các thuật toán AI đó tác động như thế nào đối với các cuộc tấn công mạng

1 INTRODUCTION

Internet of Things là công nghệ kết nối mọi thứ trên toàn thế giới thông qua internet. Công nghệ IoT đảm bảo cải thiện và hỗ trợ cuộc sống cho cá nhân và xã hội của chúng ta. Tuy nhiên, công nghệ này có thể bị tấn công mạng giống như bất kỳ mạng nào khác. Hệ thống phát hiện xâm nhập (IDS) là một kỹ thuật hiệu quả để phát hiện các cuộc tấn công mạng trong bất kỳ mạng nào. Hầu hết các IDS hiện tại đều dựa trên thuật toán học máy để đào tạo và phát hiện các cuộc tấn công mạng. Mạng IoT bao gồm các kết nối giữa các đối khác nhau từ siêu máy tính đến các thiết bị nhỏ như camera hay ti-vi vì vậy bảo mật mạng là một thách thức. Do đó an ninh mạng là một điểm yếu rất lớn trong việc triển khai mạng IoT. Tấn công DDoS là một trong những cuộc tấn công mạng lớn đã ảnh hưởng đến mạng IoT gây ra thiệt hại lớn[2].

Do cấu hình bảo mật được tích hợp trong các thiết bị IoT ít phức tạp hơn, chúng trở thành mục tiêu dễ bị tấn công cho các kẻ tấn công. Do việc bảo mật kém, các cuộc tấn công mạng vào các thiết bị này đã tăng mạnh. Hầu hết các botnet IoT, chẳng hạn như Mirai, Bashlite, Emotet và Reaper, được thiết kế một cách thông minh để thực hiện nhiều như lây nhiễm, gọi lại C&C và thực thi. Do đó, việc phát

hiện chính xác các botnet này trong mạng IoT đã trở thành điều cần thiết để giảm thiểu các rủi ro liên quan đến chúng.

=>Mục tiêu của nghiên cứu này là sử dụng các phương pháp trí tuệ nhân tạo để phát hiện một cách hiệu quả botnet IoT bằng dữ liệu lưu lượng mạng.

1.1 OVERVIEW

IDS là một ứng dụng được thiết kế để giám sát hoạt động mạng hoặc hệ thống và có khả năng phát hiện các vấn đề . IDS có hai loại trong việc thu thập thông tin khi giám sát mạng, đó là Host-Based IDS (HIDS) và Network-Based IDS. HIDS quét và phân tích sự thay đổi hệ thống tệp, cuộc gọi hệ thống và hoạt động khác trên máy chủ. Trong khi đó, NIDS giám sát lưu lượng mạng bằng cách sử dụng kỹ thuật như sniffing gói tin để thu thập dữ liệu lưu lượng mạng và cố gắng phát hiện các hoạt động độc hại như tấn công từ chối dịch vụ, quét port hoặc thậm chí là cố gắng xâm nhập vào máy tính.[4]

Có 2 loại học máy phổ biến là Supervised learning và Unsupervised learning. Học máy không giám sát (mô tả) và học máy có giám sát (dự đoán) là hai loại học máy. Giả sử dữ liệu chứa một loạt các quan sát, việc huấn luyện mô hình trên dữ liệu đó và dự đoán tự động kết quả mong muốn dựa trên thông tin đã thu được trước đó tạo thành quá trình phát triển một mô hình trí tuệ nhân tạo (mô hình AI). Hai loại kết quả có thể được tạo ra: phân loại và giá trị thực. Khi xử lý dự đoán đầu ra phân loại, loại tác vụ này được gọi là phân loại hoặc nhận dạng mô hình, và khi xử lý dự đoán đầu ra mục tiêu giá trị thực, loại tác vụ này được gọi là hồi quy.[5]

Mục tiêu của học máy không giám sát là nhận diện các mô hình trong dữ liệu mà không có một tập hợp mục tiêu được xác định trước. Có những phương pháp học máy giám sát và học máy không giám sát có thể được sử dụng để phân tích dữ liệu từ các mạng thiết bị IoT trong bài báo cáo này. Nó sẽ được sử dụng để tạo các mô hình học máy giám sát để giải quyết một nhiệm vụ phân loại được xác định trước: một tập hợp đã được quy định trước các thông số và giá trị của chúng được ghi lại tại một thời điểm nhất định nên được xác định là độc hại (Mirai, Gafgyt) hoặc bình thường (vô hại) với độ chính xác cao. [5]

Các phương pháp truyền thống để bảo vệ khỏi cuộc tấn công DDoS tập trung vào phát hiện tấn công và các biện pháp giảm nhẹ tại nguồn. Tuy nhiên việc áp dụng biện pháp giảm nhẹ cho hàng ngàn đến hàng trăm nghìn kẻ tấn công đặt một yêu cầu không thể chấp nhận được đối với cơ sở hạ tầng mạng [1]. Những thách thức này đã khởi xướng một tập hợp các công trình nghiên cứu tập trung vào việc sử dụng trí tuệ nhân tạo để nhận diện hoạt động của botnet tại nguồn tấn công thay vì tại cá nhân bị tấn công.[6]

Hình 1.1 mô tả sự gia tăng tấn công DDoS theo cấp số nhân trong 5 năm qua. Các cuộc tấn công DDoS bắt đầu ở quy mô nhỏ, với lưu lượng tấn công ước tính là 200Mbps . Trong những ngày đầu, một cuộc tấn công nhỏ như vậy sẽ đủ để làm sập mạng của nạn nhân. Một cuộc tấn công DDoS liên tục với tốc độ tối đa là 90 Mbps đã gây thiệt hại cho Internet của Estonia trong khoảng ba tuần vào năm 2007, gây ra sự cố đáng kể cho các trang web của chính phủ, ngân hàng và truyền thông . Một năm sau đó, cơ sở hạ tầng Internet của Georgia, nằm ở giao lộ của

Tây Á và Đông Âu, đã bị tắt khoảng một tháng do một cuộc tấn công DDoS với lưu lượng 200Mbps [6].

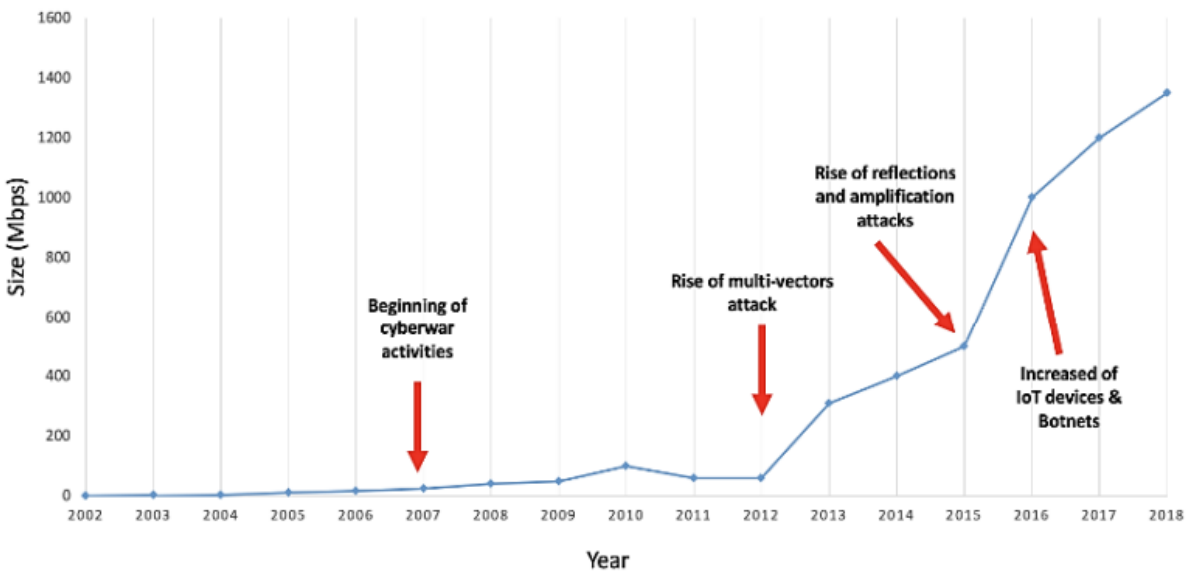


Figure 1.1: DDoS Attack Growth in Terms of Size (Mbps) from 2002-2018[8, 9, 10].

Mặc dù đã được đưa ra nhiều cách tiếp cận để xử lý các cuộc tấn công DDoS, nhưng các cách tiếp cận hiện có thiếu tính hiệu quả, khả năng mở rộng và chia sẻ thông tin để phát hiện sớm và chính xác các cuộc tấn công DDoS lớn. Khả năng nhận diện lưu lượng tấn công là một bước quan trọng trong việc giải quyết các vấn đề phát hiện cuộc tấn công DDoS. Nếu cuộc tấn công DDoS không được phát hiện, sẽ khó thực hiện bất kỳ biện pháp chống DDoS nào để xử lý cuộc tấn công. Do đó, rất cần phải có kiến thức về luồng tấn công DDoS các phương pháp phát hiện hiện tại để cải thiện việc phát hiện cuộc tấn công DDOS.[5]

1.2 PHÂN TÍCH VỀ VẤN ĐỀ BẢO MẬT CỦA IOT

IoT mở rộng Internet đến thế giới vật lý và do đó đặt ra nhiều thách thức về bảo mật khác nhau. Để đảm bảo chống lại những cuộc tấn công đó, cần xem xét các vấn đề bảo mật và các mục đích kiểm soát có hại tiềm tàng. Dưới đây, bài báo cáo này sẽ đề cập đến các vấn đề bảo mật khác nhau:

1.2.1 VẤN ĐỀ XÁC THỰC

Trong môi trường triển khai các thiết bị IoT có quy mô lớn, bảo mật và quản lý chúng trở nên phức tạp hơn, đặc biệt là trong các khu vực công cộng không có bảo vệ. Ví dụ, một cảm biến bất hợp pháp có thể đăng ký với một địa chỉ tuyệt đối tại một vị trí nhất định, trong khi thực tế nó ở một vị trí khác. Việc xác thực các thiết bị IoT, bao gồm việc nhận diện thiết bị và xác minh kết nối của chúng với một địa chỉ chính xác, là một thách thức rất lớn.

Trong lĩnh vực chăm sóc sức khỏe, vấn đề bảo mật là rất quan trọng, và các vấn đề này đã được giải quyết thông qua cơ chế xác thực mạnh. Cơ chế xác thực này được đề xuất chủ yếu dựa trên CoAP với sử dụng tính toán ECC. Khóa ECC giảm yêu cầu tính toán và cung cấp một phương thức mã hóa hiệu quả hơn so với các loại mã hóa khác. Tuy nhiên, thách thức đối với các nhà nghiên cứu không chỉ là đề xuất các công cụ xác thực mới, mà còn là đề xuất một cơ chế xác thực hỗ trợ nhiều thiết bị IoT khác nhau.

Đối với các thiết bị như đồng hồ đeo tay, bộ điều khiển nhiệt độ và các cảm biến khác, các phương pháp xác thực được sử dụng cho điện thoại di động có thể được áp dụng. Hiện nay, đã có hai giải pháp bảo mật xác thực thiết bị chính được đề xuất: giải pháp bảo mật dựa trên vật lý và giải pháp xác thực dựa trên mật mã học. Phương pháp bảo mật dựa trên vật lý thiết kế để bảo vệ thiết bị khỏi hư hỏng hoặc tấn công tầng vật lý bằng cách áp dụng các khái niệm vật lý, nhưng các phương pháp và kỹ thuật xác thực tiêu chuẩn yêu cầu tài nguyên xử lý cao. Vì vậy, cần áp dụng một phương pháp xác thực.

1.2.2 YÊU CẦU VỀ AN NINH VÀ BẢO MẬT TRONG IOT

An ninh và bảo mật đã trở thành một vấn đề cốt yếu trong lĩnh vực chăm sóc sức khỏe. Để đảm bảo sự an toàn cho dữ liệu và thông tin y tế, chúng ta cần xem xét một kiến trúc bảo mật toàn diện trong hệ thống. Đồng thời, việc tạo ra các lĩnh vực chuyên về bảo mật và phân tích trong các hệ thống chăm sóc sức khỏe dựa trên Internet of Things (IoT) là rất quan trọng.[4]

Các thiết bị IoT thông minh thường có giới hạn về khả năng tính toán do sử dụng các bộ xử lý tốc độ thấp. Chúng được thiết kế để thực hiện các nhiệm vụ hiệu quả trong môi trường có hạn chế. Để nâng cao hiệu suất và bảo mật của các thiết bị này, việc sử dụng các giải pháp bảo mật tối ưu hóa tài nguyên và tăng cường tính bảo mật là cần thiết. Ví dụ, các thiết bị IoT sử dụng trong chăm sóc sức khỏe như cảm biến huyết áp, cảm biến nhiệt độ và các thiết bị tương tự có hạn chế về năng lượng pin, và vì vậy chúng tiết kiệm năng lượng bằng cách chuyển sang chế độ ngủ khi không cần thiết để đo đạc. Ngoài ra, cần giải quyết các vấn đề bảo mật khác liên quan đến tính linh hoạt và khả năng mở rộng của hệ thống.[4]

1.3 ĐỘNG LỰC

Với sự gia tăng không ngừng của số lượng thiết bị kết nối internet đã tạo ra những thách thức đáng kể cho các chuyên gia an ninh. Nguyên nhân chính khiến các thiết bị IoT trở thành mục tiêu dễ bị tấn công là do cách chúng được thiết kế. Các thiết bị IoT này có quyền truy cập internet mà không có bất kỳ bộ lọc bằng thông nào. Hơn nữa, do giới hạn của hệ điều hành sử dụng trên các thiết bị này, không có đủ không gian để nâng cao tính bảo mật. Theo một nghiên cứu của Viện Tiêu dùng Mỹ, hơn 8-10 bộ định tuyến trong gia đình và văn phòng có thể bị tấn công. Tài khoản mật khẩu yếu hoặc sử dụng mật khẩu mặc định cùng với các lỗ hổng là những nguyên nhân chính khiến các kẻ xấu lợi dụng thiết bị IoT. Phần lớn phần mềm độc hại sử dụng khả năng tính toán của các thiết bị IoT để thực hiện các cuộc tấn công từ chối dịch vụ (DoS) trên quy mô lớn. Phát hiện sự hiện diện của một botnet IoT có thể đáng kể giảm khả năng xảy ra cuộc tấn công DoS. Sử dụng trí tuệ nhân tạo để phát hiện botnet IoT là một phương pháp hiệu quả và nhanh chóng hơn. Đánh giá này nghiên cứu các mô hình giải pháp khác nhau để phát hiện lưu lượng botnet một cách chính xác và hiệu quả.

Internet of Things (IoT) đã mở rộng khả năng thu thập thông tin từ nhiều lĩnh vực trong cuộc sống, bao gồm thiết bị tiên tiến, nhà máy, ô tô, nông nghiệp và y tế. Tuy nhiên, để thực sự hiểu được tiềm năng của IoT, chúng ta cần khai thác thông tin cơ bản một cách chính xác thông qua mô hình toán học tương đối. Thông tin trong hệ thống IoT có nhiều ứng dụng khác nhau. Để đạt được lợi ích đáng kể,

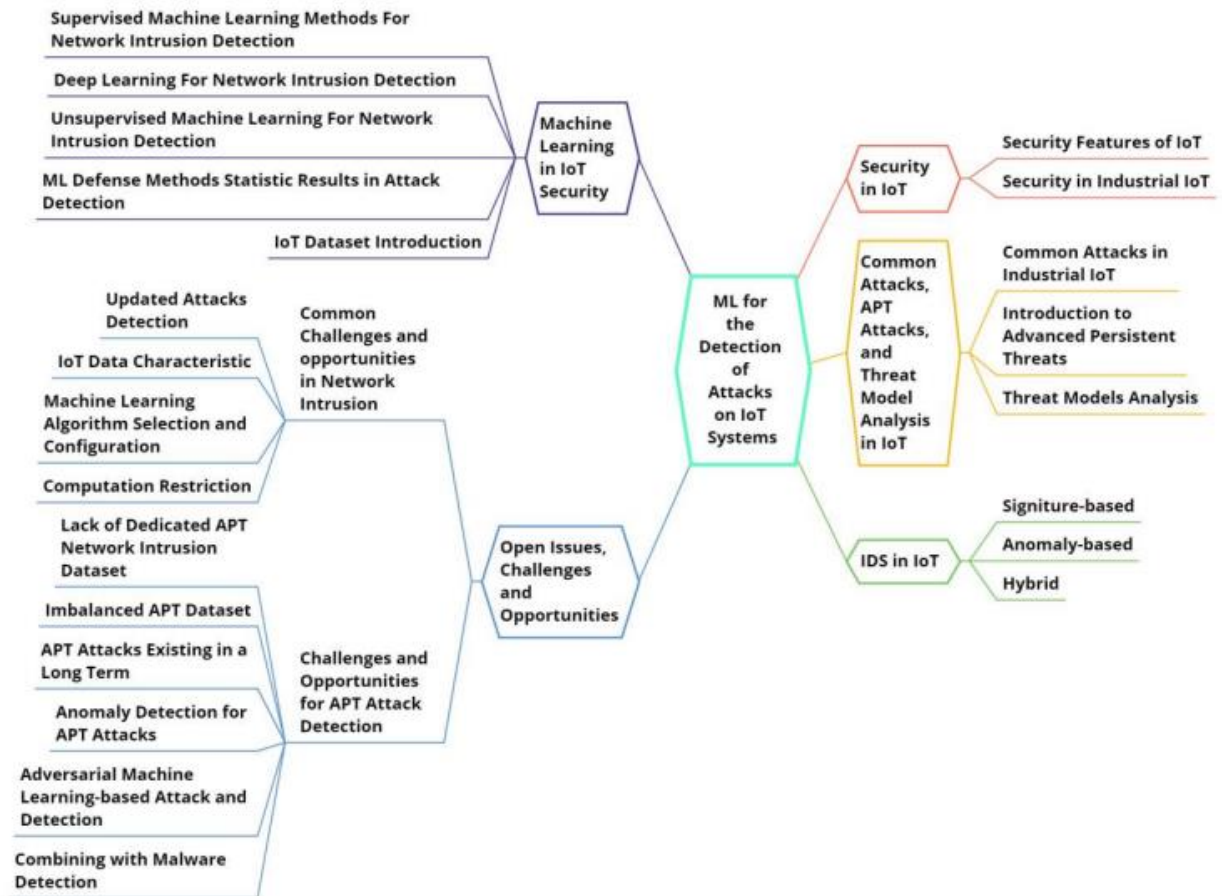
điều quan trọng là hiểu rằng thông tin không đầy đủ hoặc không chính xác sẽ không mang lại giá trị.

Một số thiết bị IoT có thể trở thành thành phần của một mạng botnet lớn hơn, được sử dụng để thực hiện các cuộc tấn công phủ đầu dịch vụ (DDoS) quy mô lớn và đa dạng. Ví dụ, botnet Mirai, được biết đến với tên gọi MalwareMustDie, đã xuất hiện vào tháng 8 năm 2016. Nó tham gia vào một số cuộc tấn công DDoS kinh khủng và đã thu hút sự chú ý toàn cầu. Kể từ khi mã nguồn Mirai được công khai, nó đã bị tận dụng bởi nhiều tên tội phạm, với những chỉnh sửa nhỏ. Nhóm này đang cố gắng tạo ra malware này để lợi dụng game thủ Minecraft và kiếm lợi nhuận nhỏ. Theo thời gian, Mirai đã trải qua sự cải tiến và trở nên đáng kinh ngạc hơn những gì chúng ta có thể tưởng tượng từ khi nó xuất hiện lần đầu. Đối với lĩnh vực bảo mật, đây là một câu chuyện về những nguy cơ có thể xảy ra. Chính vì vậy, chúng ta cần nỗ lực tạo ra và triển khai các phương pháp tính toán để xác định và ngăn chặn các cuộc tấn công đó.

Dựa trên hiệu quả của các phương pháp dựa trên máy học (ML) trong việc phát hiện các loại tấn công phổ biến trong IoT, bài báo cáo này nhằm đánh giá các mô hình tấn công trong hệ thống cơ sở hạ tầng được giám sát bằng IoT và xem xét việc sử dụng thuật toán ML để phát hiện những cuộc tấn công. [5]

Gần đây, đã có nhiều cuộc khảo sát nghiên cứu các phương pháp dựa trên máy học (ML) để phát hiện xâm nhập mạng, như được thể hiện trong Bảng 2. Mặc dù các phương pháp dựa trên ML đã xuất hiện, việc nghiên cứu các phương pháp truyền thống để hiểu rõ các lợi ích và hạn chế của chúng là rất quan trọng. Trong khi đó, một tập dữ liệu cụ thể cho IoT là một phần cần thiết trong nghiên cứu về

bảo mật IoT sử dụng các thuật toán máy học. Tuy nhiên, việc triển khai mạng IoT là khó khăn và tốn kém.[7]



Để thu thập các mẫu dữ liệu làm tập dữ liệu IoT, việc sử dụng một tập dữ liệu công khai đã được hướng tới IoT là một phương pháp tiết kiệm chi phí để thiết lập một dự án và tập trung vào các phương pháp nghiên cứu. Tuy nhiên, hiếm khi có tài liệu tóm tắt thông tin cụ thể về IoT. Với động lực này, bài viết này nhằm mục đích giới thiệu nghiên cứu về các kỹ thuật truyền thống và phương pháp dựa trên ML cho phát hiện tấn công trong một bài báo và tổng hợp các tập dữ liệu công khai hiện có về IoT. [7]

Bảng dưới đây thể hiện sự khác biệt trong các khảo sát có liên quan dựa trên các tiêu chí sau: ML và/hoặc Deep Learning (DL), kỹ thuật IDS truyền thống, mô hình tấn công và đe dọa.

Ref.	ML/ DL	Non-M Techniques	Threat Model	APT	Limitations
[51]	DL	×	×	×	DL-based detection methods without ML
[76]	Both	✓	×	×	No threat analysis specific to the IoT network attack pattern
[41]	Both	×	×	×	No discussion of non-ML-based detecting techniques and no traditional security methods
[32]	Both	✓	✓	×	No APT attack life cycle and attack pattern discussion
[60]	Both	×	×	×	Nonetwork attacks categorized and old datasets (around 2010) discussed
[13]	Both	×	✓	×	Lack analysis of IoT-related datasets
Our	Both	✓	✓	✓	With ML, DL, and non-ML detection schemes, and analysis impact of APT attacks in the IoT context for the first time

1.4. ĐÓNG GÓP CỦA BÀI BÁO CÁO

Các nghiên cứu gần đây đã chỉ ra rằng phương pháp học máy và học sâu có độ chính xác cao trong việc phân loại dữ liệu lưu lượng mạng là độc hại hay vô hại. Tuy nhiên, vẫn còn thiếu các giải pháp áp dụng cho môi trường IoT quy mô lớn hơn, như các mạng lưới doanh nghiệp. Phát hiện bất thường trong lưu lượng mạng dựa trên các phương pháp học máy đang là một lĩnh vực triển vọng, nhờ khả năng học các mẫu lưu lượng mạng phức tạp và phát hiện sự bất thường. Tuy vậy, cần giải quyết một số thách thức trong việc phát hiện các cuộc tấn công botnet dựa trên học máy, bao gồm khả năng mở rộng cho các mạng lưới lớn, giảm tiêu thụ năng lượng tính toán, xử lý các loại cuộc tấn công khác nhau và phân loại lưu lượng giao thông thành nhiều lớp như vô hại hoặc độc hại, đồng thời đảm bảo tính trực quan cao của mô hình học máy bằng cách đơn giản hóa mô hình để không tăng thêm chi phí triển khai trong môi trường thực tế IoT.

Một nghiên cứu gần đây tập trung so sánh hiệu suất của các thuật toán học sâu và học máy sau khi chọn 2, 3 và 10 tính năng hàng đầu dựa trên xếp hạng điểm Fisher (phương pháp lọc). Kết quả nghiên cứu đã chứng minh rằng việc kết hợp các phương pháp lựa chọn tính năng với các thuật toán học máy cổ điển có thể đạt được hiệu suất tốt hơn mà không cần sử dụng các mô hình học sâu phức tạp hơn trong triển khai tương lai. Bài báo cáo hiện tại sẽ nghiên cứu liên quan đến phát hiện cuộc tấn công botnet dựa trên phân tích mẫu lưu lượng mạng tập trung vào việc tăng cường hiệu suất của các mô hình học máy cổ điển bằng cách sử dụng các thuật toán lựa chọn các tính năng khác nhau.

1.4.1 ĐẶT VẤN ĐỀ

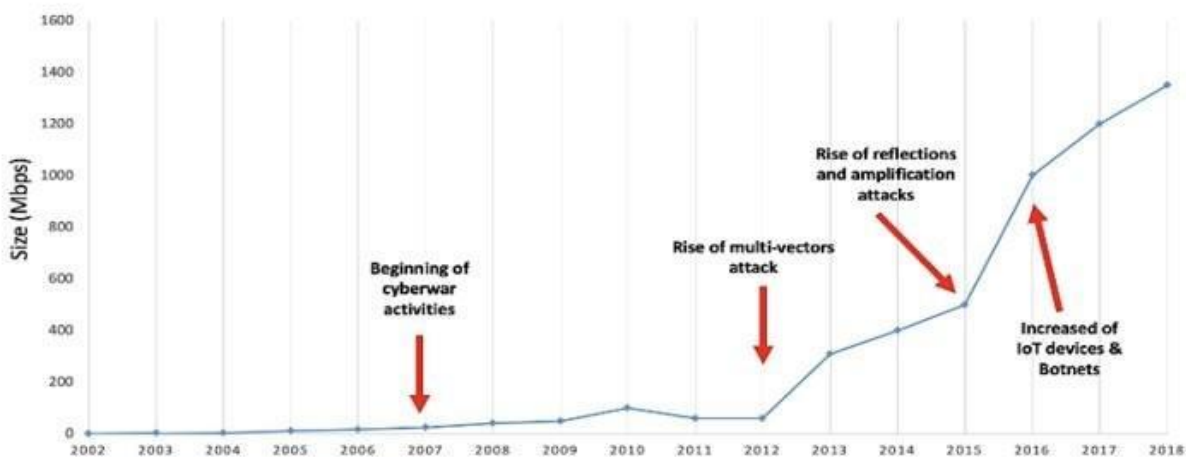
Tấn công từ chối dịch vụ (DoS) đã tồn tại từ năm 1974 khi một học sinh trung học 13 tuổi đã khám phá ra một lệnh có thể thực hiện trên terminal PLATO. Ví dụ, bằng cách sử dụng lệnh 'ext' và thực hiện Ext "Sepaker", 1, "on", một hệ thống không được kết nối với thiết bị bên ngoài có thể bị treo và buộc phải tắt nguồn để khởi động lại. David Dennies đã phát hiện vấn đề này và anh đã viết một chương trình gửi lệnh này đến tất cả người dùng trong hệ thống Plato. Kết quả là 31 người dùng đã bị tắt nguồn cùng một lúc, gây ra một sự hỗn loạn lớn trong trường vào thời điểm đó. Sau đó, kỹ sư hệ thống của Plato đã phát hiện và sửa lỗi chương trình bằng cách thay đổi từ "on" thành "off".[8]

Tấn công DDoS gây ra những hậu quả nghiêm trọng về tài chính và uy tín cho tổ chức, đồng thời rất khó kiểm soát do đi qua các cổng không được bảo mật của tường lửa. Điều này tạo ra một thách thức khó khăn trong việc đối phó với nó. Thực tế là hầu như mọi tổ chức công ty lớn đã từng trải qua ít nhất một cuộc tấn công DDoS. Tác động lớn của những cuộc tấn công này khiến nhân viên bảo mật

mạng luôn gặp khó khăn khi xử lý DDOS. Đối với các doanh nghiệp nhỏ và vừa, mục tiêu của nghiên cứu này là phát triển một công cụ phát hiện DDoS toàn diện, có khả năng xác định một cuộc tấn công DDoS với độ chính xác cao.[9]

Các phương pháp truyền thống để bảo vệ khỏi cuộc tấn công DDoS tập trung vào việc phát hiện tấn công và giảm nhẹ tác động tại nguồn. Tuy nhiên, việc áp dụng các biện pháp giảm nhẹ cho hàng ngàn đến hàng trăm nghìn kẻ tấn công [15] đặt một yêu cầu không thể chấp nhận được đối với cơ sở hạ tầng mạng. Nhận thức về những thách thức này, nhiều nghiên cứu đã tập trung vào việc sử dụng trí tuệ nhân tạo để nhận diện hoạt động của botnet tại nguồn tấn công.

Hình 1.1 mô tả sự gia tăng mạnh mẽ của cuộc tấn công DDoS trong 5 năm qua theo một cấp số nhân. Ban đầu, các cuộc tấn công DDoS chỉ có quy mô nhỏ, với lưu lượng tấn công ước tính là 200Mbps. Tuy nhiên, đã xảy ra một cuộc tấn công DDoS liên tục với tốc độ tối đa là 90 Mbps gây ra sự cố lớn trên Internet của Estonia trong khoảng ba tuần vào năm 2007, gây thiệt hại đáng kể cho các trang web của chính phủ, ngân hàng và truyền thông.[10]



Với sự phát triển của Internet, tấn công DDoS đang trở nên ngày càng phổ biến và tinh vi, nhằm tấn công nhiều mục tiêu khác nhau. Trong quá khứ, các cuộc tấn công DDoS thường nhắm vào một tổ chức cụ thể, nhưng hiện nay, chúng đã tiến xa hơn và có khả năng làm đổ sập nhiều tổ chức và cơ quan trên Internet trong một khu vực cụ thể [8]. Một cuộc tấn công đáng chú ý đã gây tắc nghẽn Internet trên toàn châu Âu và đánh sập máy chủ Spamhaus, làm giảm tốc độ truy cập vào các trang web chính [8]. Năm sau đó, cuộc tấn công DDoS lớn nhất được ghi nhận vào OVH, một công ty lưu trữ web, với tốc độ tấn công đạt 400Gbps [9]. Cuộc tấn công DDoS quan trọng khác đã sử dụng botnet Mirai và đạt đỉnh tấn công với hơn 1Tbps vào năm 2016, gây ra sự không thể truy cập một số trang web nổi tiếng như Twitter, Reddit, GitHub và Airbnb.

Mặc dù đã có nhiều phương pháp tiếp cận để xử lý cuộc tấn công DDoS, nhưng hiện tại các phương pháp này thiếu hiệu quả, khả năng mở rộng và chia sẻ thông tin để phát hiện sớm và chính xác các cuộc tấn công DDoS lớn và nhỏ. Để giải quyết vấn đề phát hiện cuộc tấn công DDoS, khả năng nhận diện luồng tấn công là

rất quan trọng. Nếu không thể phát hiện cuộc tấn công DDoS, thì khó có thể thực hiện bất kỳ biện pháp chống DDoS nào để đối phó với cuộc tấn công đó. Do đó, việc hiểu về các luồng tấn công DDoS và những hạn chế của các phương pháp phát hiện hiện tại là rất cần thiết để cải thiện khả năng phát hiện cuộc tấn công.

1.4.2 GIẢI PHÁP

Bài báo cáo này nhằm đánh giá các thuật toán khác nhau và xác định những hạn chế chính ngăn cản việc áp dụng ngay lập tức các hệ thống phát hiện Máy học trong lĩnh vực an ninh mạng.

Các kết luận của bài báo cáo này dựa trên việc khám phá một số lượng lớn tài liệu cùng với việc thực hiện các thí nghiệm trên các hệ thống doanh nghiệp thực tế và lưu lượng mạng. Bài báo cáo cung cấp một phân loại gốc ban đầu về các phương pháp Máy học được áp dụng trong an ninh mạng. Sau đó áp dụng phân loại này vào ba vấn đề an ninh mạng chính là phát hiện xâm nhập, phân tích phần mềm độc hại. Đồng thời cũng phân tích những hạn chế chính của các phương pháp hiện có.

Nghiên cứu chỉ rõ những ưu và nhược điểm của các phương pháp khác nhau, đặc biệt là trong việc xác định positive hoặc negative. Ngoài ra bài báo cáo cũng chỉ ra rằng việc quản lý kiến trúc Máy học trong lĩnh vực an ninh mạng có tính phức tạp cao, nhưng lại thiếu thông tin công khai và dữ liệu. Đồng thời, trình bày một phân tích dành cho các chuyên gia an ninh về các kỹ thuật Máy học áp dụng cho phát hiện xâm nhập, phân tích phần mềm độc hại

2. Background

Lĩnh vực IoT (Internet of Things) đang phát triển nhanh chóng với số lượng thiết bị và hệ thống kết nối tăng lên. Sự kết nối này mang lại nhiều lợi ích, nhưng đồng thời cũng đặt ra những thách thức bảo mật mạng lớn. Bảo vệ hệ thống IoT chống lại các mối đe dọa mạng là một vấn đề quan trọng đối với cả cá nhân và tổ chức. Các phương pháp bảo mật truyền thống có thể không đủ để đáp ứng các đặc điểm và lỗ hổng độc đáo của các thiết bị IoT. Do đó, ngày càng có sự quan tâm đến việc áp dụng kỹ thuật Máy học trong bảo mật IoT.[10]

Máy học đã cho thấy tiềm năng của mình trong nhiều lĩnh vực, bao gồm thị giác máy tính, xử lý ngôn ngữ tự nhiên và phân tích dữ liệu. Khả năng học các mẫu và dự đoán dựa trên dữ liệu giúp Máy học trở thành một công cụ tiềm năng để phát hiện và giảm thiểu các mối đe dọa mạng trong IoT. Bằng cách huấn luyện các mô hình Máy học trên các tập dữ liệu lớn về lưu lượng mạng và hành vi của thiết bị IoT, ta có thể phát triển các hệ thống thông minh có khả năng nhận biết sự bất thường, phát hiện các hoạt động độc hại và nâng cao tổng thể an ninh môi trường IoT.[11]

Trong những năm gần đây, nhiều bài báo khoa học đã tập trung vào sự giao thoa giữa bảo mật IoT và Máy học. Những bài báo này nhằm khám phá và đánh giá các phương pháp và kỹ thuật khác nhau nhằm nâng cao an ninh của các hệ thống IoT bằng Máy học. Chúng cung cấp những thông tin về những ưu điểm, hạn chế và ứng dụng tiềm năng của Máy học trong bảo mật IoT.[11]

Máy học có những ưu điểm quan trọng khi được áp dụng trong bảo mật IoT. Đầu tiên, Máy học có khả năng học từ dữ liệu và phát hiện các mẫu bất thường. Điều này cho phép nó nhận biết các hoạt động độc hại hoặc không bình thường trong lưu lượng mạng IoT. Thứ hai, Máy học có thể thích ứng với sự thay đổi và tính động của các thiết bị IoT. Với khả năng tự học, các mô hình Máy học có thể điều chỉnh và cập nhật để phù hợp với các giao thức mới và các hình thức tấn công mới. Cuối cùng, Máy học cung cấp khả năng phân tích dữ liệu mạnh mẽ và xây dựng các mô hình dự đoán dựa trên dữ liệu lưu lượng mạng. Điều này giúp cải thiện hiệu suất phát hiện và giảm thiểu số lượng các cảnh báo sai tích cực hoặc tiêu cực.[10]

Tuy nhiên, Máy học cũng gặp một số hạn chế trong bảo mật IoT. Một trong những hạn chế quan trọng là vấn đề về quyền riêng tư và bảo mật dữ liệu. Máy học yêu cầu sử dụng dữ liệu lưu lượng mạng để huấn luyện và xây dựng mô hình, và điều này đặt ra thách thức trong việc bảo vệ thông tin cá nhân và dữ liệu nhạy cảm. Hơn nữa, Máy học có thể gặp khó khăn trong việc nhận diện các tấn công mới và chưa từng được ghi nhận trước đó. Nếu không có đủ dữ liệu đào tạo hoặc không có các mẫu tấn công mạng mới, Máy học có thể không hiệu quả trong việc phát hiện các mối đe dọa mới.[10]

Với những ưu điểm và hạn chế này, Máy học có nhiều ứng dụng tiềm năng trong bảo mật IoT. Một ứng dụng khác của Máy học trong bảo mật IoT là phát hiện và ngăn chặn các cuộc tấn công mạng. Máy học có thể học từ các mô hình tấn công đã biết và dự đoán các mẫu tấn công tương tự trong lưu lượng mạng IoT. Điều này giúp ngăn chặn các cuộc tấn công trước khi chúng gây hại cho hệ thống.[10]

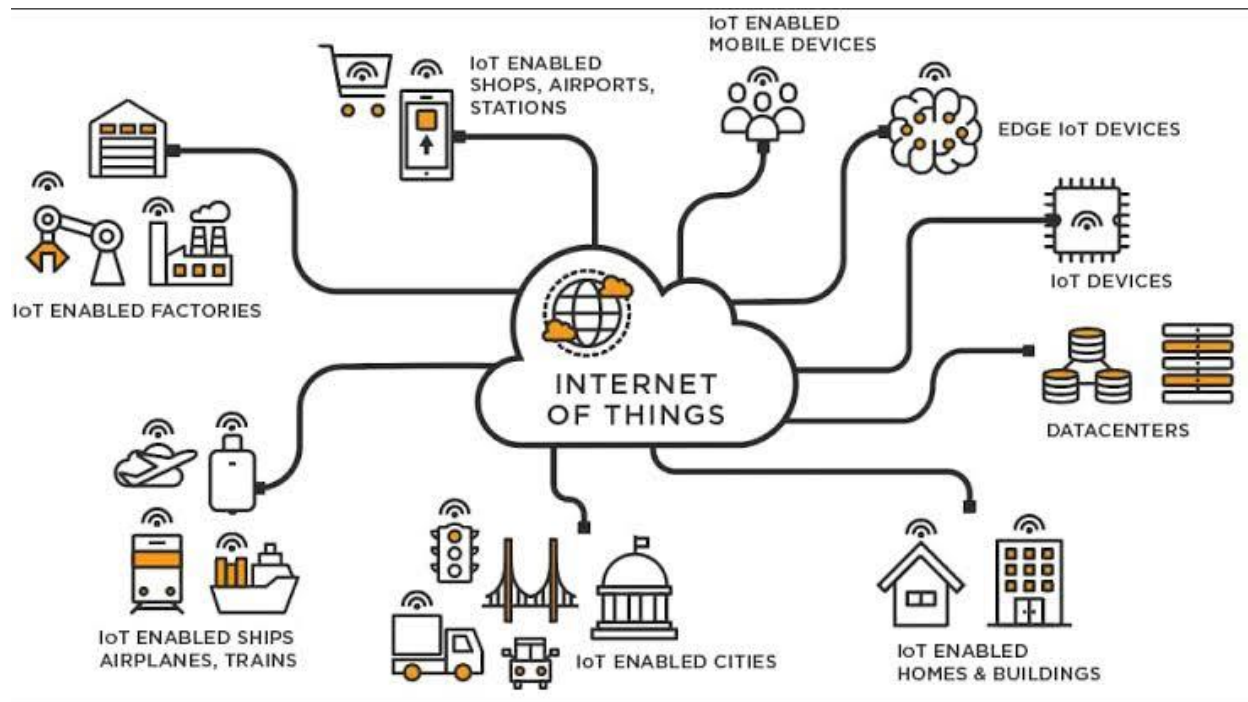
2.1 IOT LÀ GÌ?

Trong khía cạnh tổng quát nhất, thuật ngữ IoT bao gồm tất cả các thiết bị được kết nối với internet. Theo Matthew Evans, người đứng đầu chương trình IoT tại techUK, "Simply, the Internet of Things is made up of devices – from simple sensors to smartphones and wearables – connected together,"[13]

Bằng cách kết hợp các thiết bị kết nối này với các hệ thống tự động, ta có thể thu thập thông tin, phân tích dữ liệu và tạo ra hành động nhằm hỗ trợ người sử dụng trong một nhiệm vụ cụ thể hoặc học hỏi từ quá trình đó. Phạm vi ứng dụng của IoT trải rộng từ hệ thống phát tín hiệu cho đến các thiết bị thông minh có tích hợp Internet.

Theo Caroline Gorski, người đứng đầu IoT tại Digital Catapult, "Gather information, analyse it and create an action". Điều này không chỉ vượt ra khỏi việc giới hạn giao tiếp trong nhóm thiết bị cùng loại mà còn cho phép giao tiếp qua các mạng khác nhau, tạo ra một thế giới kết nối toàn diện hơn.[13]

Tóm lại, IoT không chỉ liên quan đến việc kết nối các thiết bị và mạng lưới, mà còn tập trung vào việc khai thác thông tin từ dữ liệu và tạo ra các hành động thông minh. Sự kết hợp giữa các thiết bị kết nối và hệ thống tự động trong IoT mang lại tiềm năng vô tận để tạo nên một thế giới kết nối, thông minh và hiệu quả.



2.1.1 GIÁ TRỊ CỦA IOT

Internet of Things (IoT) đã trở thành một công nghệ đột phá với giá trị to lớn trong các lĩnh vực khác nhau. Nó cho phép kết nối và giao tiếp giữa các thiết bị, tạo ra lượng lớn dữ liệu có thể được sử dụng để thu thập thông tin và thực hiện các hành động. IoT mang lại những lợi ích đáng kể về tự động hóa, hiệu quả và khả năng ra quyết định. Nó giúp các doanh nghiệp thu thập thông tin thời gian thực, phân tích nó và thực hiện biện pháp cải tiến quy trình và nâng cao trải nghiệm khách hàng.[15]

Giá trị đề xuất của IoT nằm ở khả năng tạo ra các hệ thống và quy trình thông minh. Như được đề cập trong một nghiên cứu của Porter và Heppelmann (2015), IoT cho phép tích hợp các thiết bị vật lý với công nghệ số, tạo ra mạng lưới thông minh có thể tối ưu hóa hoạt động, thúc đẩy sáng tạo và mở ra nguồn thu mới. Tổ

chức có thể tận dụng IoT để theo dõi và điều khiển thiết bị từ xa, dự đoán nhu cầu bảo trì, tối ưu hóa tiêu thụ năng lượng và nâng cao năng suất.[14]

Hơn nữa, IoT tạo điều kiện cho những hiểu biết dựa trên dữ liệu và quyết định có căn cứ. IoT tạo ra lượng lớn dữ liệu từ các thiết bị kết nối, có thể được phân tích để thu được những thông tin quý giá về hành vi khách hàng, xu hướng thị trường và hiệu quả hoạt động. Bằng cách tận dụng sức mạnh của phân tích dữ liệu lớn và các thuật toán học máy, các doanh nghiệp có thể ra quyết định dựa trên dữ liệu, cá nhân hóa trải nghiệm khách hàng và xác định cơ hội kinh doanh mới.[16]

Ngoài ra, IoT còn tiềm năng cách mạng hóa các ngành công nghiệp truyền thống. Nhấn mạnh rằng IoT đang tạo ra những thay đổi đáng kể trong các lĩnh vực như sản xuất, y tế, năng lượng và nông nghiệp. Các thiết bị thông minh kết nối với nhau và với hệ thống phân tán, tạo ra khả năng tự động hóa, quản lý hiệu quả và cải thiện sự đáng tin cậy. Ví dụ, trong lĩnh vực y tế, IoT cho phép theo dõi sức khỏe của bệnh nhân từ xa, cung cấp dịch vụ chăm sóc y tế tốt hơn và giảm thiểu sai sót trong quá trình điều trị.[16]

Tuy nhiên, cũng cần nhận thức về những thách thức và rủi ro liên quan đến IoT. IoT đặt ra những vấn đề về bảo mật thông tin và riêng tư. Với sự kết nối của nhiều thiết bị và dữ liệu nhạy cảm được truyền qua mạng, việc bảo vệ dữ liệu trở thành một vấn đề quan trọng. Ngoài ra, các vấn đề về tiêu thụ năng lượng, chuẩn hóa và tính tương thích cũng cần được xem xét để đảm bảo tính hợp nhất và sự phát triển bền vững của IoT.[15]

2.1.2 KIẾN TRÚC CỦA IOT

IoT có kiến trúc không được thống nhất trên toàn cầu. Thế nhưng kiến trúc IOT phổ biến và rộng rãi nhất là cấu trúc IoT 3 lớp hoặc 4 lớp. Bốn lớp đó bao gồm Sensing layer, Network Layer, Data processing layer và Application layer:

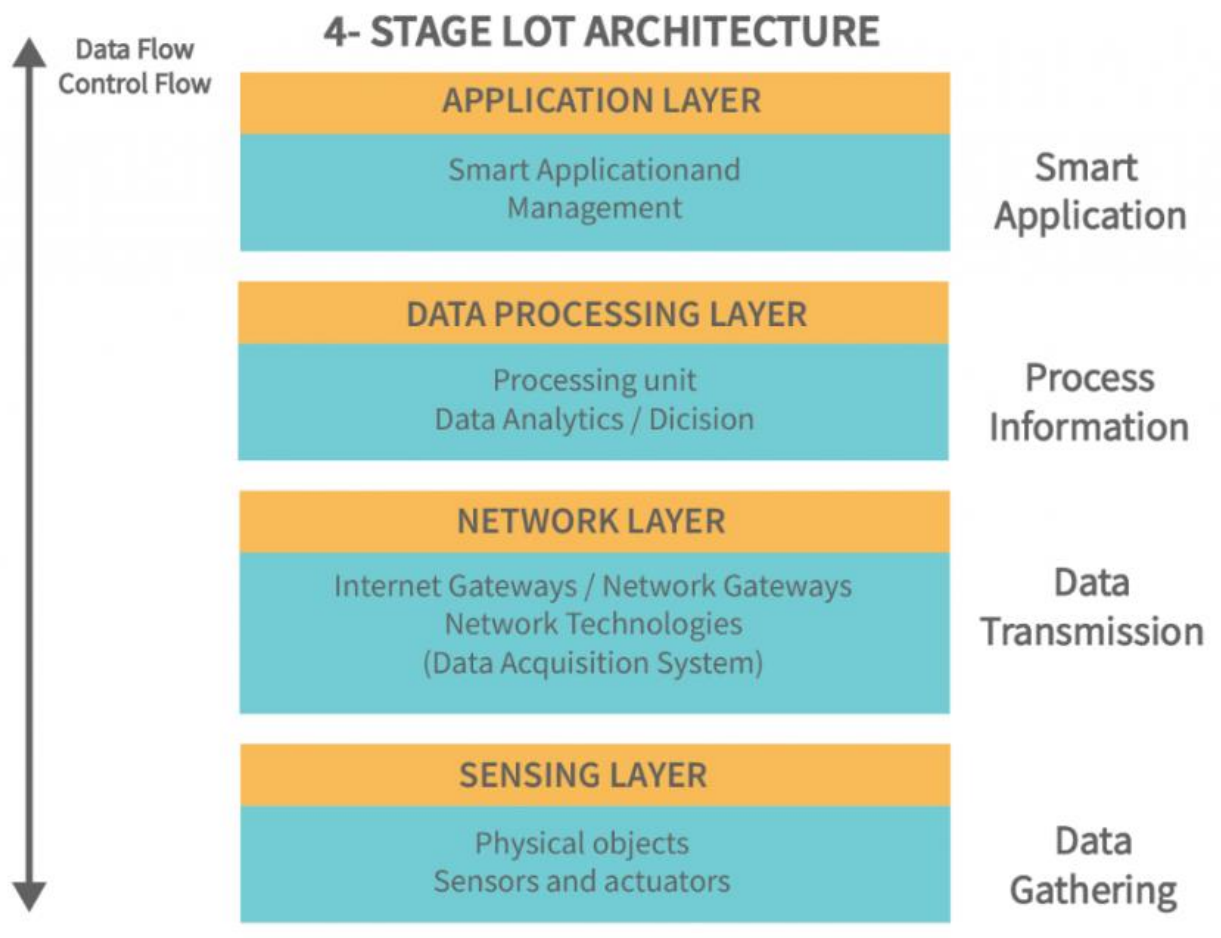
Sensing layer là lớp đầu tiên trong kiến trúc IoT và chịu trách nhiệm thu thập dữ liệu từ các nguồn khác nhau. Nó bao gồm cảm biến và bộ điều khiển được đặt trong môi trường để thu thập thông tin về nhiệt độ, độ ẩm, ánh sáng, âm thanh và các thông số vật lý khác. Các thiết bị này kết nối với lớp mạng thông qua các giao thức truyền thông có dây hoặc không dây.[17]

Network là lớp quan trọng trong kiến trúc IoT, đảm nhận nhiệm vụ cung cấp giao tiếp và kết nối giữa các thiết bị trong hệ thống IoT. Nó sử dụng các giao thức và công nghệ để thiết lập kết nối và truyền thông giữa các thiết bị với nhau và với internet. Các công nghệ mạng phổ biến như WiFi, Bluetooth, Zigbee và mạng di động 4G và 5G được sử dụng trong IoT. Lớp mạng cũng có thể bao gồm các thiết bị như cổng và định tuyến trung gian để hỗ trợ kết nối và tính bảo mật, như mã hóa và xác thực, để ngăn chặn truy cập trái phép.[17]

Data processing layer là nơi các thành phần phần cứng và phần mềm được sử dụng để thu thập, phân tích và hiểu dữ liệu từ các thiết bị IoT. Lớp này nhận dữ liệu thô từ các thiết bị, tiến hành xử lý và chuẩn bị dữ liệu cho phân tích hoặc hành động tiếp theo. Các công nghệ và công cụ như hệ thống quản lý dữ liệu, nền tảng phân tích và thuật toán học máy được sử dụng để trích xuất thông tin ý

nghĩa từ dữ liệu và đưa ra quyết định dựa trên nó. Một ví dụ của công nghệ trong lớp xử lý dữ liệu là hồ dữ liệu, nơi lưu trữ dữ liệu thô từ các thiết bị IoT.[17]

Application Layer - Nằm ở lớp cao nhất của kiến trúc IoT, Application layer tương tác trực tiếp với người dùng cuối. Nhiệm vụ chính của lớp này là cung cấp giao diện và chức năng thân thiện để người dùng có thể truy cập và điều khiển các thiết bị IoT. Lớp này bao gồm một loạt phần mềm và ứng dụng đa dạng như ứng dụng di động, cổng thông tin web và các giao diện người dùng khác, được thiết kế để tương tác với cơ sở hạ tầng IoT cơ bản. Ngoài ra, Application layer cũng bao gồm các dịch vụ trung gian giúp các thiết bị và hệ thống IoT khác nhau có thể giao tiếp và chia sẻ dữ liệu một cách mượt mà. Application layer cũng có khả năng phân tích và xử lý dữ liệu, cho phép dữ liệu được phân tích và chuyển đổi thành những thông tin có ý nghĩa. Điều này có thể bao gồm việc sử dụng các thuật toán học máy, công cụ trực quan hóa dữ liệu và các khả năng phân tích tiên tiến khác.[17]



2.1.3 CÁCH THỨC HOẠT ĐỘNG CỦA IOT

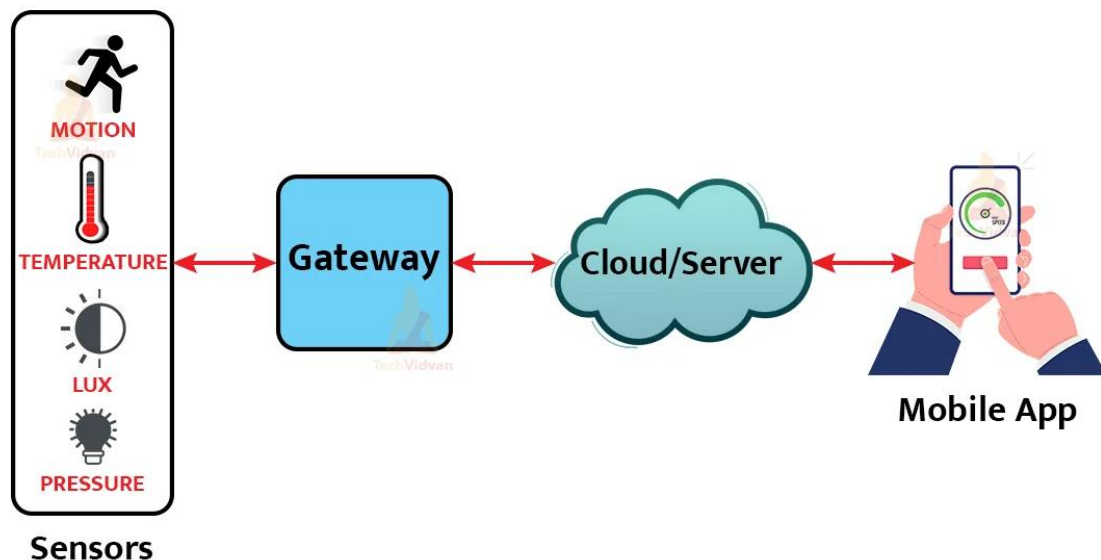
1. Thu thập dữ liệu: các cảm biến như độ ẩm, chất lượng không khí, nhiệt độ, chuyển động, ánh sáng và các cảm biến khác là những ví dụ về cảm biến. Hệ thống cảm biến này cho phép chúng ta thu thập thông tin từ thế giới thực. Mặc dù thông tin không được cấu trúc và thô, nhưng chúng ta có thể rút ra thông tin từ đó, cuối cùng giúp đưa ra các quyết định có cơ sở thông tin.

2. Giao diện người dùng: cho phép sử dụng các cảnh báo để thông báo về tin nhắn, email, cuộc gọi, cảnh báo và nhiều hơn nữa. Điều này cho phép người dùng

truy cập vào một điểm tương tác, cho phép họ theo dõi hệ thống một cách chủ động và đưa ra hành động tương ứng.

3. Xử lý dữ liệu: bao gồm việc chuẩn bị dữ liệu trước khi nó được gửi đến đám mây. Điều này có thể đơn giản như kiểm tra nhiệt độ và lưu ý nếu nó trở nên quá nóng. Hơn nữa, cũng có thể xảy ra các sự kiện bất ngờ.

Working of IoT



4. Kết nối: Là cốt lõi của toàn bộ công nghệ IoT, đóng vai trò là trung gian trong việc truyền tải thông tin từ hệ thống này sang hệ thống khác. Dữ liệu được thu thập bởi các cảm biến sau đó được truyền qua kết nối mạng.



2.2 CÁC RỦI RO VÀ THÁCH THỨC CỦA IOT

Internet of Things (IoT) đã trở thành một công nghệ đổi mới có tiềm năng lớn trong nhiều ngành công nghiệp. Tuy nhiên, vẫn còn tồn tại những khoảng cách và thách thức đáng kể cần được giải quyết để đạt được sự thâm nhập và thành công rộng rãi của IoT. Vì vậy, nhiều nghiên cứu và bài báo khoa học đã thảo luận về thách thức liên quan đến việc triển khai IoT.

2.2.1 BẢO MẬT

Trong thời đại hiện đại, giám sát các mối đe dọa về bảo mật không còn là một khái niệm mới. Tuy nhiên, sự phổ biến của các thiết bị IoT đã đặt ra những thách thức bảo mật mới và phức tạp hơn. Người dùng và nhà sản xuất công nghệ IoT cần nhận thức rằng dữ liệu cá nhân được lưu trữ trong các thiết bị của họ không đảm bảo an toàn do sự tồn tại của các lỗ hổng trong hệ thống nội bộ. Nguyên nhân chính cho tình trạng này là IoT cung cấp một lượng lớn thông tin cá nhân chi tiết

mà không yêu cầu sự tương tác động từ người dùng, và công nghệ này đã trở thành một phần không thể thiếu trong cuộc sống hàng ngày của chúng ta. Với tình trạng quá tải của các thiết bị trong tổ chức hiện tại, việc duy trì hệ thống trở thành một nhiệm vụ khó khăn. Không thể tách rời khái niệm về các thiết bị IoT khỏi Internet theo bất kỳ cách nào sẽ làm giảm tính an toàn và sức mạnh tổng thể của Internet. Hơn nữa, một sự cộng tác có thể tạo ra tác động đáng kể để giải quyết vấn đề này và đồng thời thúc đẩy sự tiến bộ của công nghệ nói chung. Thậm chí, các hệ thống có thể bị tấn công thông qua việc biến các thiết bị như bộ định tuyến và camera không dây thành Bot có thể được kiểm soát bởi Botmaster.[18]Rõ ràng, việc sử dụng mật khẩu và các quy ước bảo mật mặc định đã trở nên quá dễ dàng để tránh được sự tấn công như botnet Mirai. Tuy nhiên, việc sử dụng mật khẩu mạnh hơn, các quy trình ủy quyền và xác thực, cùng với mã hóa, sẽ có tác động tích cực đối với vấn đề bảo mật.

2.2.2 CHÍNH SÁCH

Trong môi trường IoT (Internet of Things), sự kết nối của các thiết bị với phần cứng và phần mềm phức tạp đem đến nhiều khó khăn trong việc giám sát và kiểm soát hệ thống. Tuy nhiên, một vấn đề tiềm ẩn là khả năng rò rỉ dữ liệu nhạy cảm hoặc thông tin cá nhân bởi các lỗ hổng bảo mật mà hacker có thể lợi dụng. Mỗi thiết bị trong hệ thống IoT có thể chia sẻ một lượng lớn thông tin cá nhân về người dùng, bao gồm tên, nguồn gốc và nhiều chi tiết khác. Việc truyền tải, xử lý và lưu trữ thông tin này đòi hỏi ưu tiên cao về an toàn. Như Pollmann (2017) đã đề cập, "rất nhiều thông tin này là thông tin cá nhân và một số trong số đó có thể rất nhạy cảm." Vì vậy, mối lo ngại về an ninh và bảo mật trong môi trường IoT là điều không thể bỏ qua. Tuy nhiên, không thể phủ nhận rằng IoT đã đơn giản hóa các hoạt động hàng ngày của chúng ta thông qua tính đa chức năng và khả năng

thu thập thông tin từ môi trường xung quanh và trình bày nó một cách dễ hiểu. Điều quan trọng là IoT đã mang đến khả năng thực hiện các nhiệm vụ hàng ngày từ xa và đóng vai trò quan trọng trong sự phát triển của các ngành công nghiệp.

2.3 DISTRIBUTED DENIAL-OF-SERVICE (DDOS)

Cuộc tấn công lũy tiến từ chối dịch vụ (DDoS) là một trong những mối quan ngại hàng đầu của các chuyên gia an ninh. Mục tiêu chủ yếu của cuộc tấn công DDoS là làm gián đoạn quyền truy cập hợp pháp của người dùng vào các dịch vụ. Kẻ tấn công thường tận dụng lỗ hổng trong các máy tính để xâm nhập và tạo ra một quân đội tấn công, còn được gọi là Botnets. Sau khi thiết lập quân đội tấn công, kẻ tấn công có thể tiến hành một cuộc tấn công quy mô lớn, có sự phối hợp đối với một hoặc nhiều mục tiêu. Phát triển một cơ chế phòng thủ toàn diện chống lại các cuộc tấn công DDoS đã được xác định và được dự đoán là mục tiêu của cộng đồng nghiên cứu về phát hiện xâm nhập và ngăn chặn. Tuy nhiên, để phát triển một cơ chế như vậy, chúng ta cần hiểu rõ về vấn đề và các kỹ thuật đã được sử dụng để ngăn chặn, phát hiện và phản ứng với các cuộc tấn công DDoS khác nhau.[19]

2.3.1 BOT

Bot là một chương trình phần mềm, còn được gọi là phần mềm độc hại, được cài đặt trên máy chủ bị xâm phạm và thường thực hiện các hoạt động độc hại. Các bot thường được cài đặt thông qua việc truy cập vào các trang web bị nhiễm hoặc các cơ chế lây nhiễm khác. Quan trọng nhất, bot không tận dụng các lỗ hổng trong ứng dụng hoặc hệ điều hành, mà nó được lan truyền bởi worm hoặc được sử dụng để cài đặt cửa sau trên các máy bị xâm phạm. Mỗi khi thiết bị bị nhiễm khởi động, bot sẽ được cấu hình và khởi động, và các hoạt động sẽ được kích hoạt thông qua các lệnh gửi từ botmaster qua kênh điều khiển và điều phối. Sự tồn tại

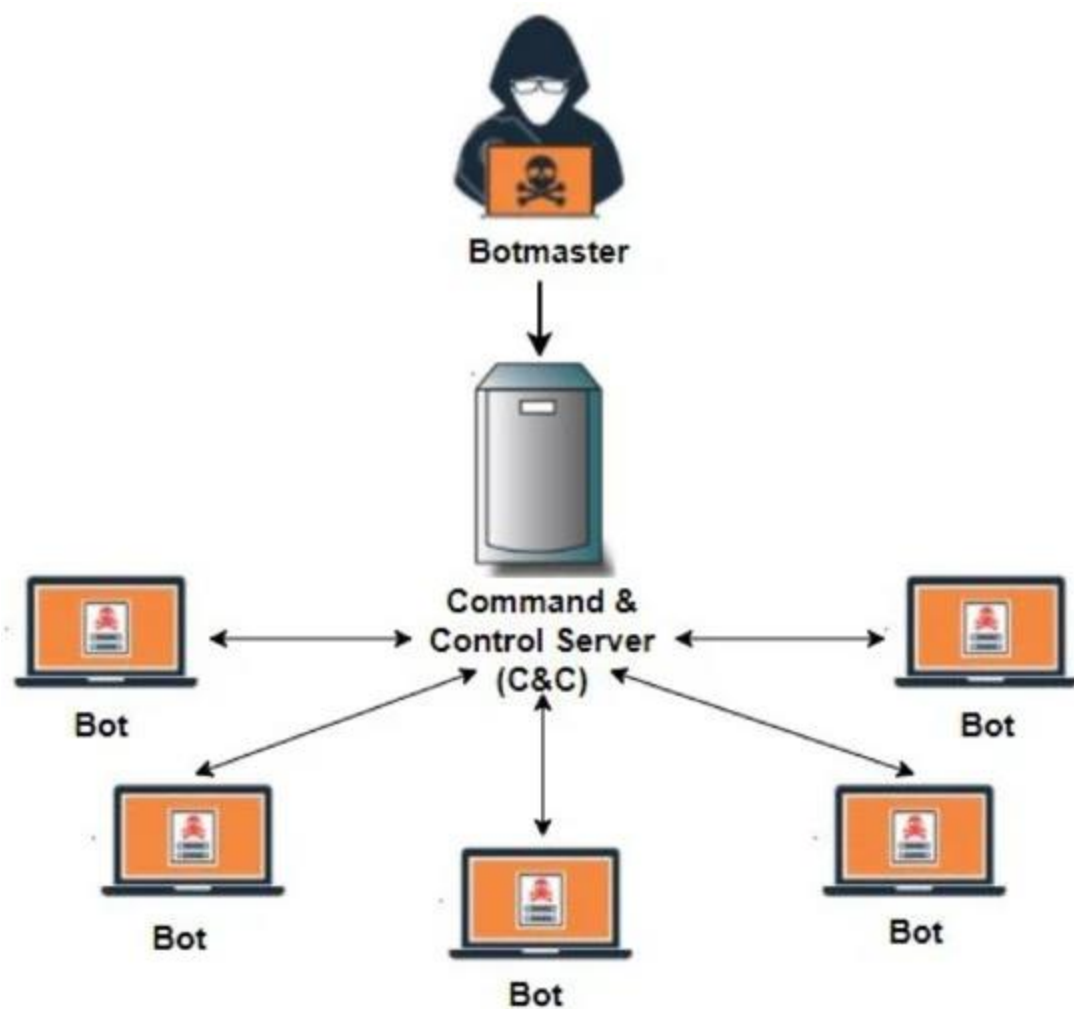
của kênh điều khiển và điều phối là điểm khác biệt chính giữa bot và các loại phần mềm độc hại khác.[20]

2.3.2 BOTNET

Botnet là một nhóm các máy chủ bị nhiễm và chạy các bot, kết nối đến một kênh điều khiển và điều phối (C&C) để chờ lệnh thực hiện các hoạt động độc hại.[20]

2.3.3. BOTMASTER

Botmaster điều khiển botnet từ một vị trí từ xa bằng cách gửi các lệnh đến các bot để thực hiện các hành vi bất hợp pháp, chẳng hạn như lợi dụng tài chính bằng "việc cho thuê mạng để gửi thư rác đến người dùng khác". Botmaster cố gắng xâm nhập vào các máy tính yếu đang có cơ chế phòng thủ kém bằng cách sử dụng cơ chế truyền bá. Sau khi bị nhiễm, những máy tính này trở thành "nô lệ" hoặc "zombie" và được sử dụng để tấn công các máy chủ yếu hoặc tiến hành các cuộc tấn công từ chối dịch vụ (DoS)[21]



2.3.4 CÁCH THỨC TẤN CÔNG CỦA BOTNET

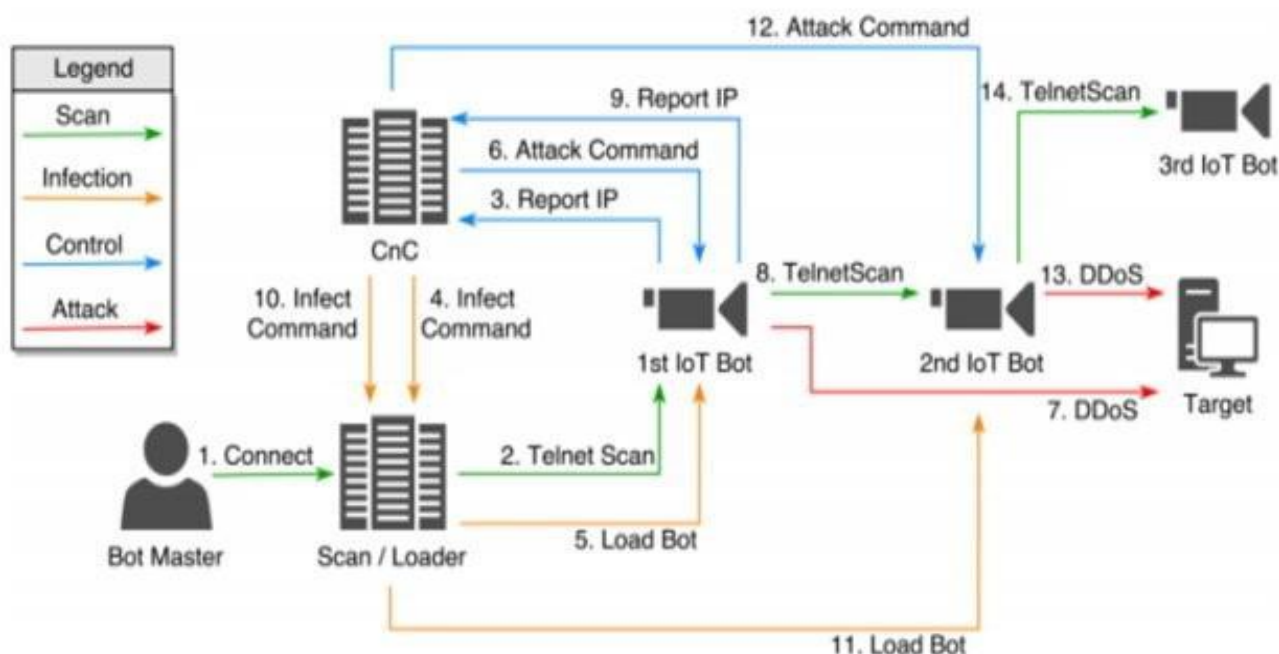


Figure 2.6: Botnet Detection in the Internet of Things [21].

Bước 1: Botmaster bắt đầu quá trình bằng cách kết nối tới máy chủ Scan/Loader

Bước 2: Quét trên Internet để tìm các thiết bị IoT dễ bị tấn công

Bước 3: Khi phát hiện một thiết bị dễ bị tấn công, phần mềm độc hại sẽ thử đăng nhập bằng phương pháp tấn công vét cạn bằng danh sách 62 tên người dùng và mật khẩu mặc định được biết đến.

Bước 4: Lệnh nhiễm độc được gửi từ máy chủ C&C tới máy chủ Scan/Loader chứa tất cả thông tin cần thiết như chi tiết đăng nhập, địa chỉ IP, kiến trúc phần cứng.

Bước 5: Phần mềm Scan/Loader sử dụng thông tin này để đăng nhập và chỉ thị cho thiết bị IoT có lỗ hổng. Sau khi thực thi, thiết bị IoT bị nhiễm độc đưa vào botnet Mirai và có thể liên lạc với máy chủ (C&C).

Bước 6: Botmaster bây giờ có thể đưa ra các lệnh tấn công, chỉ định các tham số như thời gian tấn công và mục tiêu.

Bước 7: Phần mềm độc hại bao gồm 10 loại tấn công DDoS, bao gồm UDP ,SYN ,ACK, GRE (gre ip), có thể được sử dụng để tấn công mục tiêu

Từ bước 8 trở đi thì botnet mới sẽ thực hiện phương thức tấn công như bước 1[]

2.3.5 TẠI SAO BOTNET ĐƯỢC TẠO RA?

Từ những hoạt động có những hành vi phá hoại, thường được tài trợ bởi nhà nước hoặc vì lợi ích tài chính. Sử dụng dịch vụ botnet trực tuyến thường rẻ và mang lại tổn hại lớn. Hơn nữa, việc hình thành các botnet nhỏ càng dễ dàng đến mức nó có thể trở thành một nguồn thu nhập đáng kể cho một số nhà cung cấp dịch vụ, đặc biệt là trong các quốc gia có quy định hạn chế. Điều này đã dẫn đến sự phát triển của nhiều dịch vụ trực tuyến botnet.

2.4 Machine Learning

2.4.1 SUPERVISED LEARNING

Supervised Learning là một phương pháp học máy được xác định bởi việc sử dụng các bộ dữ liệu được gán nhãn. Những bộ dữ liệu này được thiết kế để đào tạo hoặc "giám sát" các thuật toán vào việc phân loại dữ liệu hoặc dự đoán kết quả một cách chính xác. Bằng cách sử dụng các đầu vào và đầu ra được gán nhãn, mô hình có thể đo đặc độ chính xác của mình[22]

Supervised Learning có thể được chia thành hai loại vấn đề trong khai thác dữ liệu: **Classification** và **Regression**:

Classification sử dụng một thuật toán để phân loại chính xác dữ liệu kiểm tra vào các danh mục cụ thể. Hoặc trong thế giới thực, các thuật toán học có giám sát có thể được sử dụng để phân loại thư rác vào một thư mục riêng biệt so với hộp thư đến. Linear classifiers, support vector machines, decision trees và random forest là những loại thuật toán phân loại phổ biến.[22]

Regression là một phương pháp học có giám sát khác sử dụng một thuật toán để hiểu mối quan hệ giữa các biến phụ thuộc và độc lập. Các mô hình hồi quy hữu ích trong việc dự đoán các giá trị số dựa trên các điểm dữ liệu khác nhau, chẳng hạn như dự đoán doanh thu bán hàng cho một doanh nghiệp cụ thể. Một số thuật toán hồi quy phổ biến là linear regression, logistic regression và polynomial regression.[22]

2.4.2 UNSUPERVISED LEARNING

Unsupervised Learning sử dụng các thuật toán học máy để phân tích và gom nhóm các bộ dữ liệu không được gán nhãn. Các thuật toán này khám phá các mẫu ẩn trong dữ liệu mà không cần sự can thiệp của con người

Các mô hình Unsupervised Learning được sử dụng cho ba tác vụ chính: clustering, association và dimensionality reduction

Clustering là một kỹ thuật khai thác dữ liệu để nhóm các dữ liệu không được gán nhãn dựa trên sự tương đồng hoặc khác biệt của chúng. Ví dụ, thuật toán gom nhóm K-means sắp xếp các điểm dữ liệu tương tự vào các nhóm, trong đó giá trị K đại diện cho kích thước và độ chi tiết của nhóm. Kỹ thuật này hữu ích trong việc phân đoạn thị trường, nén ảnh, v.v.

Association là một loại phương pháp học không có giám sát khác sử dụng các quy tắc khác nhau để tìm các mối quan hệ giữa các biến trong một tập dữ liệu cho trước. Các phương pháp này thường được sử dụng cho phân tích giỏ hàng mua sắm và hệ thống đề xuất, theo dạng "Khách hàng đã mua mục này cũng đã mua" các gợi ý.

Dimensionality reduction là một kỹ thuật học được sử dụng khi số lượng đặc trưng (hoặc chiều) trong một tập dữ liệu cho trước quá cao. Nó giảm số lượng đầu vào dữ liệu xuống một kích thước có thể quản lý được trong khi vẫn bảo tồn tính toàn vẹn của dữ liệu. Thường thì kỹ thuật này được sử dụng trong giai đoạn tiền xử lý dữ liệu, chẳng hạn như khi autoencoder loại bỏ nhiễu từ dữ liệu hình ảnh để cải thiện chất lượng hình ảnh.

2.4.3 MACHINE LEARNING APPROACH

2.4.3.1. LOGISTICS REGRESSION

Hồi quy tuyến tính là một phương pháp phân tích thống kê sử dụng phân tích hồi quy trong thống kê toán học để xác định mối quan hệ định lượng giữa hai hoặc nhiều biến số. Lấy hai biến số làm ví dụ, (Y_1, Y_2, \dots, Y_i) là biến phụ thuộc và (X_1, X_2, \dots, X_i) là biến độc lập. Khi biến phụ thuộc và biến độc lập có một mối quan hệ tuyến tính, hàm bình phương tối thiểu trong phương trình hồi quy tuyến tính có thể được sử dụng để xây dựng một mô hình toán học về mối quan hệ giữa biến độc lập và biến phụ thuộc. Một phân tích hồi quy dựa trên mô hình tuyến tính này được gọi là hồi quy tuyến tính. Mục đích của nó là tìm các thông số phù hợp nhất và sử dụng một đường thẳng để khớp các điểm dữ liệu đã được gắn kết.[24]

2.4.3.2. DECISION TREE

Decision Tree là đồ thị giống như cây gồm các nút trong đó các nhánh biểu thị kết quả và các nút lá biểu thị một nhãn lớp. Các quy tắc phân loại được hình thành bằng cách chọn một đường đi từ nút gốc đến nút lá. Để phân chia mỗi dữ liệu đầu vào, trước tiên chọn nút gốc vì nó là thuộc tính nổi bật nhất để tách dữ liệu. Cây được xây dựng bằng cách xác định các thuộc tính và các giá trị liên quan của chúng sẽ được sử dụng để phân tích dữ liệu đầu vào tại mỗi nút trung gian của cây. Sau khi cây được hình thành, nó có thể dự đoán dữ liệu mới bằng cách duyệt từ nút gốc đến nút lá, truy cập tất cả các nút trong đường dẫn phụ thuộc vào điều kiện kiểm tra của các thuộc tính tại mỗi nút.[25]

2.4.3.3. RANDOM FOREST

Random Forest là một phương pháp tập hợp (ensemble method) trong Machine Learning, được xây dựng dựa trên cây quyết định. Nó kết hợp dự đoán của nhiều cây quyết định độc lập với nhau để tạo ra một mô hình dự đoán mạnh mẽ. Mỗi cây quyết định trong Random Forest là một cây giống như cây với các nút biểu thị một bài kiểm tra trên một thuộc tính, các nhánh biểu thị kết quả của bài kiểm tra và các nút lá biểu thị một nhãn lớp. Các quy tắc phân loại được hình thành bằng cách chọn một đường đi từ nút gốc đến nút lá, tương tự như trong cây quyết định. Tuy nhiên, điểm đặc biệt của Random Forest là nó sử dụng một tập hợp các cây quyết định. Mỗi cây được xây dựng trên một phần khác nhau của tập dữ liệu huấn luyện, và các cây được độc lập với nhau. Khi cần dự đoán cho một điểm dữ liệu mới, Random Forest sẽ dùng tất cả các cây để đưa ra dự đoán riêng cho mỗi cây, và cuối cùng, dự đoán cuối cùng sẽ được tính toán dựa trên đa số phiếu bầu của các cây.[26]

2.5.3.4. MULTI LAYER PERCEPTRON

Multi-layer Perceptron (MLP) là một thuật toán học có giám sát, nó học một hàm bằng cách huấn luyện trên một tập dữ liệu: $F(.) : \mathbb{R}^m \rightarrow \mathbb{R}^o$ trong đó m là số chiều của đầu vào và o là số chiều của đầu ra. Với một tập các đặc trưng và mục tiêu $X = x_1, x_2, x_3, \dots, x_m$. Nó có thể học hàm phi tuyến cho phân loại hoặc hồi quy. Nó khác với hồi quy logistic ở chỗ giữa lớp đầu vào và lớp đầu ra, có thể có một hoặc nhiều lớp phi tuyến, gọi là các lớp ẩn.

2.5.3.5. XGBOOSTING

Các mô hình Boosting Tree truyền thống chỉ sử dụng thông tin đạo hàm bậc nhất. Khi huấn luyện cây thứ n , việc triển khai huấn luyện phân tán trở nên khó khăn vì sử dụng các dư thừa của $n-1$ cây trước đó. XGBoost thực hiện một phương trình Taylor bậc hai trên hàm mất mát và có thể tự động sử dụng đa luồng của CPU cho tính toán song song. Bên cạnh đó, XGBoost sử dụng nhiều phương pháp để tránh tình trạng quá khớp.

2.5.3.6. LIGHTGBM

LightGBM là một thuật toán khác được thiết kế bởi Microsoft Research Asia sử dụng khung GBDT. Mục tiêu của nó là cải thiện hiệu suất tính toán, để các vấn đề dự đoán với dữ liệu lớn có thể được giải quyết hiệu quả hơn. Trong thuật toán GBDT, phương pháp sắp xếp trước được sử dụng để chọn và chia các chỉ số. Mặc dù phương pháp này có thể xác định chính xác điểm chia, nhưng nó yêu cầu thời gian và bộ nhớ nhiều hơn. Trong LightGBM, thuật toán dựa trên biểu đồ tần số và chiến lược phát triển theo chiều lá cây với giới hạn độ sâu tối đa được áp dụng để tăng tốc độ huấn luyện và giảm tiêu thụ bộ nhớ. [33]

2.5.3.7. HIST GRADIENT BOOSTING

Hist Gradient Boosting Regression là một phương pháp để huấn luyện cây quyết định nhanh hơn được sử dụng trong gradient boosting. Rời rạc hóa có thể được sử dụng để tăng tốc độ quá trình huấn luyện cây được thêm vào một tập hợp cây. Điều này làm cho phương pháp hist gradient boosting thực hiện thuật toán của mình cho các biến đầu vào. Mỗi cây được thêm vào tập hợp cây cố gắng sửa các sai sót dự báo thông qua các mô hình đã tồn tại trong tập hợp.[35]

2.5.3.8. KNN

KNN là viết tắt của thuật toán "K-Nearest Neighbors" (Người láng giềng gần nhất), một thuật toán máy học được sử dụng chủ yếu trong bài toán phân loại và dự đoán. Thuật toán này hoạt động dựa trên việc so sánh khoảng cách giữa điểm dữ liệu cần dự đoán và các điểm dữ liệu đã biết trong tập huấn luyện. KNN được coi là một thuật toán đơn giản và linh hoạt.

3 Research Methodology

Số lượng phương pháp tiếp cận trong Machine và trí tuệ Nhân tạo (AI) là rất lớn. Em sử dụng các kỹ thuật học máy sau để hoàn thành việc nghiên cứu i: (Logistics Regression, Decision Tree Random Forest, MLP, XGBoost, LightGBM). Thiết kế là một phần quan trọng trong thành công của học máy. Thiết kế thành phần thích hợp sẽ dẫn đến kết quả và độ đo cải thiện hiệu suất cao hơn. Phần này cung cấp sự hỗ trợ cho việc sử dụng tất cả bốn thuật toán, bất kể dữ liệu sử dụng là trực tiếp hay gián tiếp. Phần này đi sâu vào từng thuật toán đó để giải thích ý nghĩa của chúng đối với nghiên cứu của bài báo cáo này

3.1 SYSTEM OVERVIEW

3.1.1 UNSW-NB15 DATASET

Trong phần này, trình bày chi tiết về cấu hình môi trường tổng hợp và tạo ra dữ liệu UNSW-NB15. Phần này chủ yếu bao gồm thông tin về cấu hình testbed và toàn bộ quy trình liên quan đến việc tạo ra UNSW-NB15 từ testbed đã được cấu hình.

A. CẤU HÌNH TESTBED SỬ DỤNG CÔNG CỤ IXIA

Theo hình 1, trình tạo lưu lượng IXIA được cấu hình với ba máy chủ ảo. Các máy chủ 1 và 3 được cấu hình để phân tán lưu lượng bình thường trong khi máy chủ 2 tạo ra các hoạt động bất thường/độc hại trong lưu lượng mạng. Thiết lập giao tiếp giữa các máy chủ, thu thập lưu lượng mạng, có hai giao diện ảo có địa chỉ IP là 10.40.85.30 và 10.40.184.30. Các máy chủ được kết nối với các máy chủ thông qua hai bộ định tuyến. Bộ định tuyến 1 có địa chỉ IP là 10.40.85.1 và 10.40.182.1, trong khi bộ định tuyến 2 được cấu hình với địa chỉ IP là 10.40.184.1 và 10.40.183.1. Các bộ định tuyến này được kết nối với thiết bị tường lửa được cấu hình để chuyển tất cả lưu lượng, bất kể là bình thường hay bất thường. Công cụ tcpdump được cài đặt trên bộ định tuyến 1 để ghi lại các tệp Pcap trong quá trình mô phỏng. Hơn nữa, mục đích chính của toàn bộ testbed này là ghi lại lưu lượng bình thường hoặc bất thường, xuất phát từ công cụ IXIA và phân tán trong các nút mạng (ví dụ: máy chủ và máy khách). Quan trọng, công cụ IXIA được sử dụng như một trình tạo lưu lượng tấn công cùng với lưu lượng bình thường, hành vi tấn công được lấy từ trang CVE để tái hiện một môi trường đe dọa hiện đại.[34]

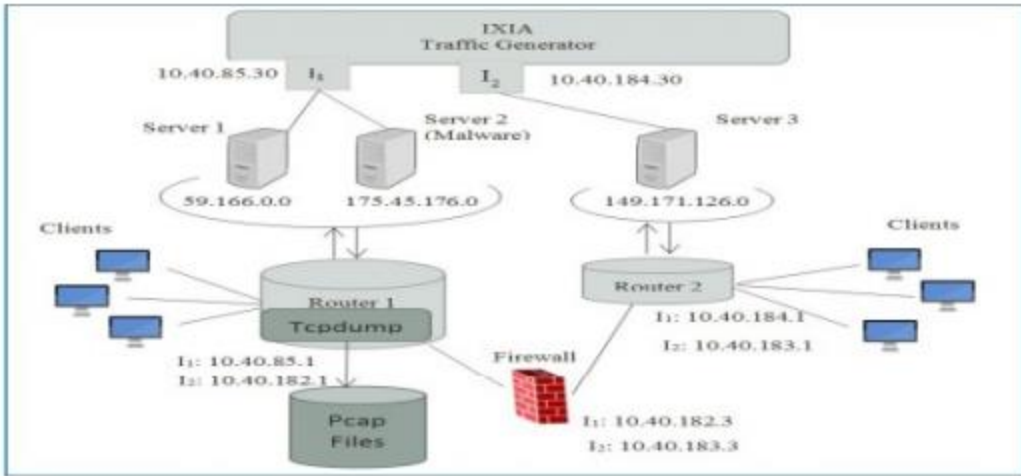


Figure 1. The Testbed Visualization for UNSW-NB15

B. ARCHITECTURE FRAMEWORK

Kiến trúc tổng thể được sử dụng để tạo ra dữ liệu UNSW-NB15 cuối cùng từ các tệp pcap thành các tệp CSV với 49 thuộc tính được trình bày trong Hình 3. Tất cả 49 thuộc tính của bộ dữ liệu UNSW-NB15 được trình bày chi tiết từ Bảng II đến Bảng VII cùng với giải thích về quy trình tạo ra Bảng cách tuân thủ kiến trúc này, các tệp pcap gốc được chuyển đổi thành một tập dữ liệu có cấu trúc với 49 thuộc tính ý nghĩa được biểu diễn dưới định dạng CSV. Tập dữ liệu này có thể được sử dụng để phân tích, xây dựng mô hình và đánh giá trong ngữ cảnh của bộ dữ liệu UNSW-NB15.[34]

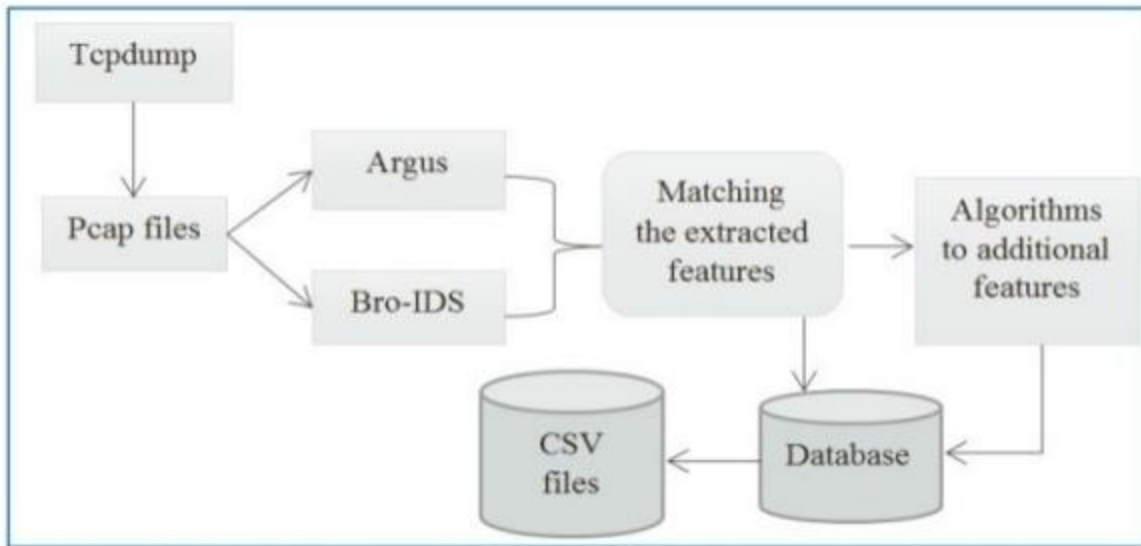


TABLE II. FLOW FEATURES

#	Name	T.	Description
1	<i>srcip</i>	N	Source IP address
2	<i>sport</i>	I	Source port number
3	<i>dstip</i>	N	Destination IP address
4	<i>dsport</i>	I	Destination port number
5	<i>proto</i>	N	Transaction protocol

TABLE III. BASIC FEATURES

#	Name	T	Description
6	<i>state</i>	N	The state and its dependent protocol, e.g. ACC, CLO, else (-)
7	<i>dur</i>	F	Record total duration
8	<i>sbytes</i>	I	Source to destination bytes
9	<i>dbytes</i>	I	Destination to source bytes
10	<i>sttl</i>	I	Source to destination time to live
11	<i>dttl</i>	I	Destination to source time to live
12	<i>sloss</i>	I	Source packets retransmitted or dropped
13	<i>dloss</i>	I	Destination packets retransmitted or dropped
14	<i>service</i>	N	http, ftp, ssh, dns ..,else (-)
15	<i>sload</i>	F	Source bits per second
16	<i>dload</i>	F	Destination bits per second
17	<i>spkts</i>	I	Source to destination packet count
18	<i>dpkts</i>	I	Destination to source packet count

TABLE IV. CONTENT FEATURES

#	Name	T	Description
19	<i>swin</i>	I	Source TCP window advertisement
20	<i>dwin</i>	I	Destination TCP window advertisement
21	<i>stcpb</i>	I	Source TCP sequence number
22	<i>dcpb</i>	I	Destination TCP sequence number
23	<i>smeansz</i>	I	Mean of the flow packet size transmitted by the src
24	<i>dmeansz</i>	I	Mean of the flow packet size transmitted by the dst
25	<i>trans_depth</i>	I	the depth into the connection of http request/response transaction
26	<i>res_bdy_len</i>	I	The content size of the data transferred from the server's http service.

TABLE V. TIME FEATURES

#	Name	T	Description
27	<i>sjit</i>	F	Source jitter (mSec)
28	<i>djit</i>	F	Destination jitter (mSec)
29	<i>stime</i>	T	record start time
30	<i>ltime</i>	T	record last time
31	<i>sintpkt</i>	F	Source inter-packet arrival time (mSec)
32	<i>dintpkt</i>	F	Destination inter-packet arrival time (mSec)
33	<i>tcprtt</i>	F	The sum of 'synack' and 'ackdat' of the TCP.
34	<i>synack</i>	F	The time between the SYN and the SYN_ACK packets of the TCP.
35	<i>ackdat</i>	F	The time between the SYN_ACK and the ACK packets of the TCP.

TABLE VI. ADDITIONAL GENERATED FEATURES

#	Name	T	Description
<i>General purpose features</i>			
36	<i>is_sm_ips_ports</i>	B	If source (1) equals to destination (3)IP addresses and port numbers (2)(4) are equal, this variable takes value 1 else 0

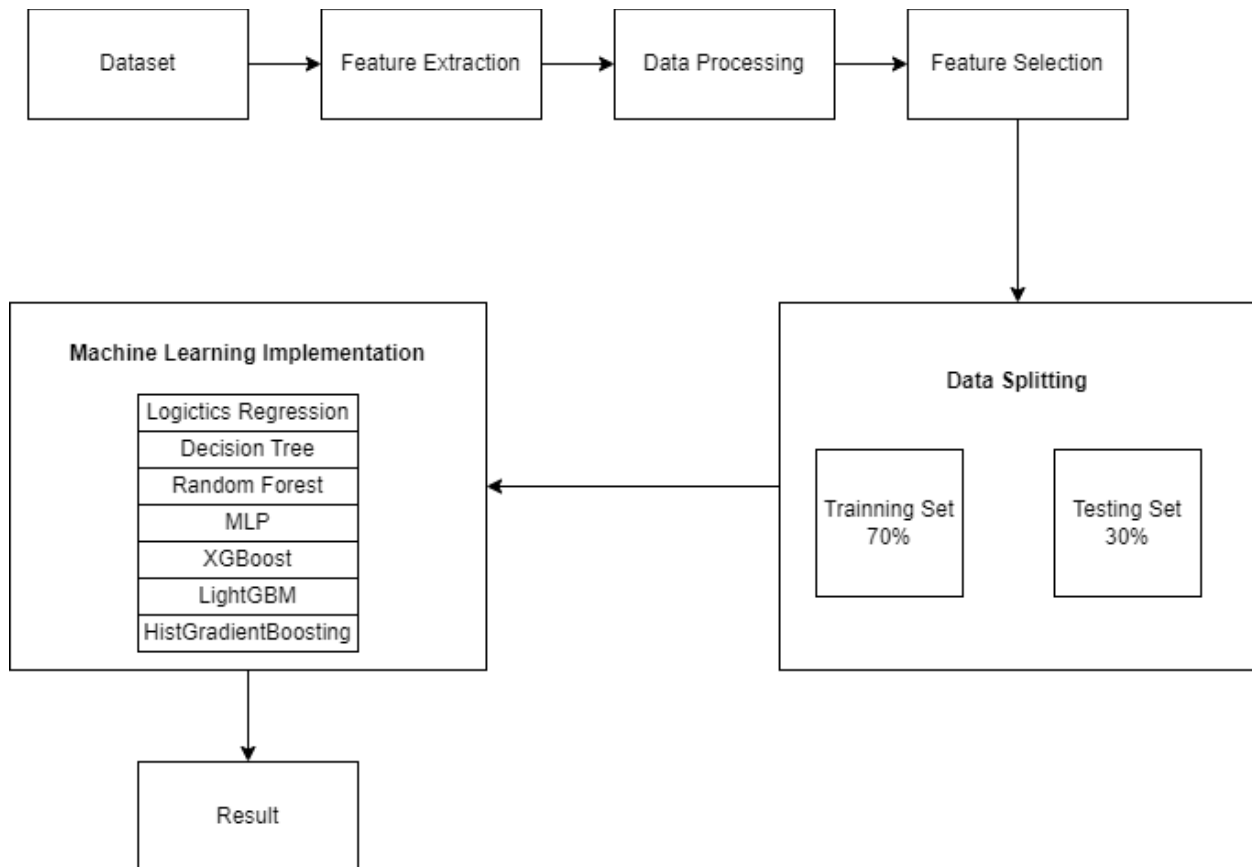
37	<i>ct_state_ttl</i>	I	No. for each state (6) according to specific range of values for source/destination time to live (10) (11).
38	<i>ct_flw_http_mthd</i>	I	No. of flows that has methods such as Get and Post in http service.
39	<i>is_ftp_login</i>	B	If the ftp session is accessed by user and password then 1 else 0.
40	<i>ct_ftp_cmd</i>	I	No of flows that has a command in ftp session.
Connection features			
41	<i>ct_srv_src</i>	I	No. of connections that contain the same service (14) and source address (1) in 100 connections according to the last time (26).
42	<i>ct_srv_dst</i>	I	No. of connections that contain the same service (14) and destination address (3) in 100 connections according to the last time (26).
43	<i>ct_dst_ltm</i>	I	No. of connections of the same destination address (3) in 100 connections according to the last time (26).
44	<i>ct_src_ltm</i>	I	No. of connections of the same source address (1) in 100 connections according to the last time (26).
45	<i>ct_src_dport_ltm</i>	I	No of connections of the same source address (1) and the destination port (4) in 100 connections according to the last time (26).
46	<i>ct_dst_sport_ltm</i>	I	No of connections of the same destination address (3) and the source port (2) in 100 connections according to the last time (26).
47	<i>ct_dst_src_ltm</i>	I	No of connections of the same source (1) and the destination (3) address in in 100 connections according to the last time (26).

TABLE VIII. DATA SET RECORD DISTRIBUTION

Type	No. Records	Description
Normal	2,218,761	Natural transaction data.
Fuzzers	24,246	Attempting to cause a program or network suspended by feeding it the randomly generated data.
Analysis	2,677	It contains different attacks of port scan, spam and html files penetrations.
Backdoors	2,329	A technique in which a system security mechanism is bypassed stealthily to access a computer or its data.
DoS	16,353	A malicious attempt to make a server or a network resource unavailable to users, usually by temporarily interrupting or suspending the services of a host connected to the Internet.
Exploits	44,525	The attacker knows of a security problem within an operating system or a piece of software and leverages that knowledge by exploiting the vulnerability.
Generic	215,481	A technique works against all block-ciphers (with a given block and key size), without consideration about the structure of the block-cipher.
Reconnaissance	13,987	Contains all Strikes that can simulate attacks that gather information.
Shellcode	1,511	A small piece of code used as the payload in the exploitation of software vulnerability.
Worms	174	Attacker replicates itself in order to spread to other computers. Often, it uses a computer network to spread itself, relying on security failures on the target computer to access it.

3.1.2 KIẾN TRÚC HỆ THỐNG

Trong tập dữ liệu UNSW-NB15, các kỹ thuật học máy khác nhau đã được sử dụng để dự đoán botnet IoT, được mô tả trong phần này. Hình dưới đây mô tả phương pháp được áp dụng trong nghiên cứu này.



Đầu tiên, các đặc trưng dựa trên luồng dữ liệu được rút trích từ tập dữ liệu gốc. Sau đó, trong giai đoạn đầu tiên, tập dữ liệu được chia thành hai phần: training và testing. Việc chuẩn bị dữ liệu trước khi xử lý là rất quan trọng để chuyển đổi dữ liệu thành định dạng được sử dụng cho các thuật toán học máy. Các thuộc tính được sử dụng bởi các thuật toán sẽ được xác định trong giai đoạn lựa chọn chức năng sau các hoạt động này. Cuối cùng, thiết kế các thuật toán học máy là giải pháp được sử dụng trong bài báo này.

Thông tin được trình bày tiếp theo được tiếp tục với các quy trình giảm chiều sâu để phát triển tính khả giải thích và loại bỏ các thuộc tính không cần thiết trong giai đoạn tiền xử lý. Giải pháp ở đây là muốn giảm các khía cạnh bằng cách sử dụng bốn phương pháp giảm khía cạnh khác nhau, phân tách các tập dữ liệu khác nhau, sau đó so sánh và kiểm tra kết quả để xác định phương pháp nào phù hợp nhất.

3.1.3 CÔNG NGHỆ MACHINE LEARNING

Bộ dữ liệu Bot-IoT đã được sử dụng để kiểm tra các thuật toán phân loại phổ biến trong học máy: Logistic Regression, Decision Tree, Random Forest, MLP, XGBoost, LightGBM

3.1.3.1 LOGISTICS REGRESSION

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

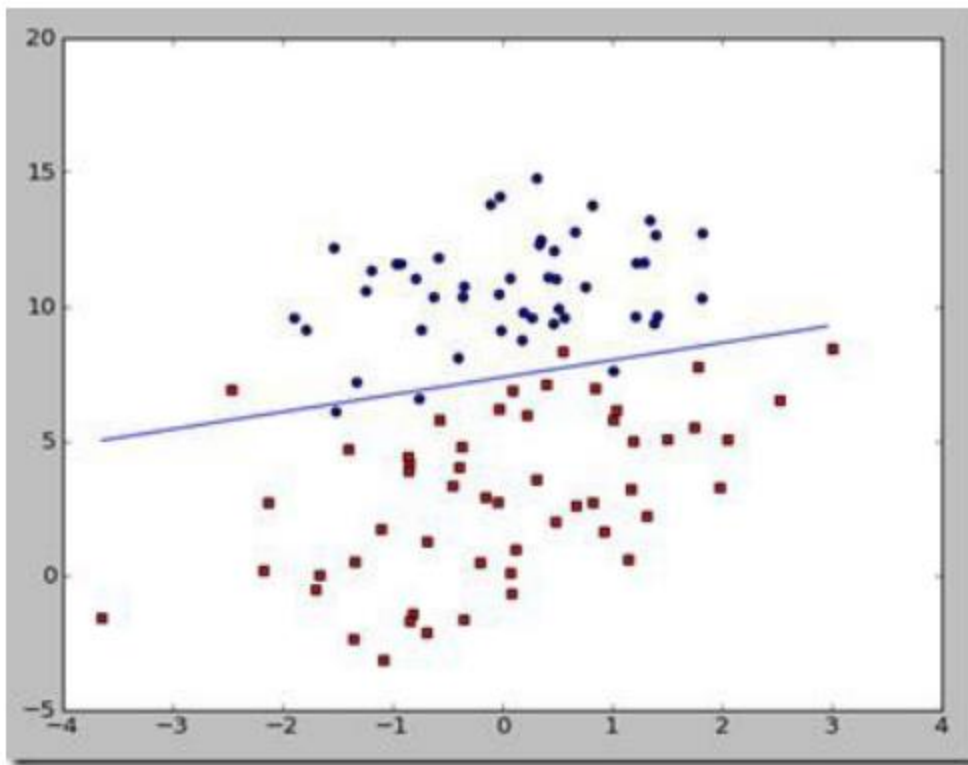


Fig. 2. Logistic regression

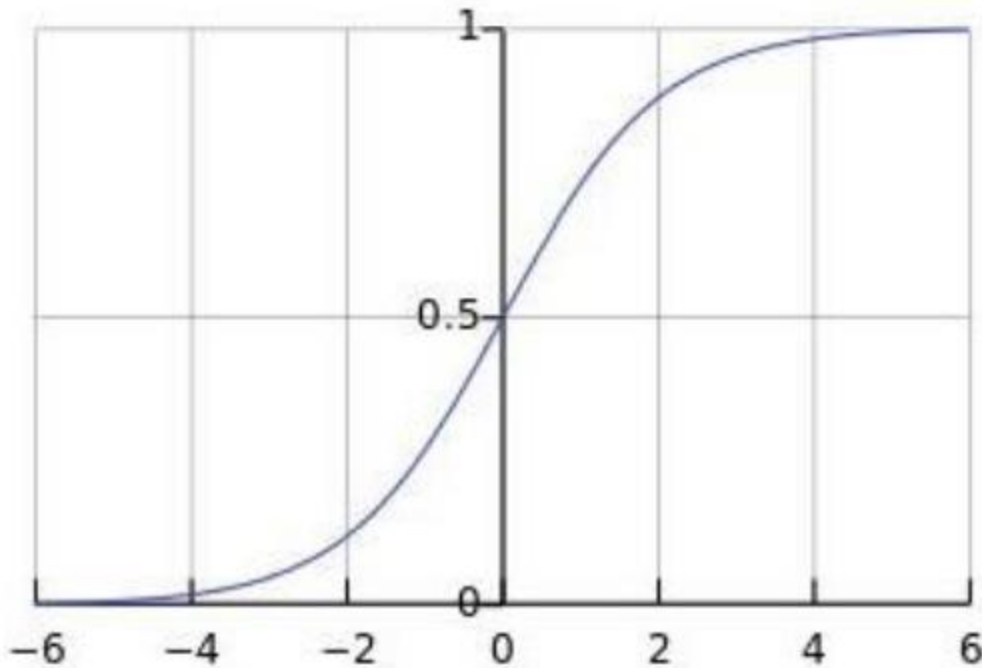


Fig. 3. Sigmoid function

Như có thể thấy từ hình vẽ, $\sigma(0) = 0.5$. Khi $z > 0$, giá trị của hàm tiến tới 1 và trở thành lớp 1 khi z tăng lên. Khi $z < 0$, giá trị của hàm tiến tới 0 và trở thành lớp 0, đáp ứng các yêu cầu của hàm phân loại trên. Quá trình phân loại của Logistic regression có thể được mô tả như sau: Giả định rằng các đặc trưng của dữ liệu đầu vào có thể được biểu diễn bằng $(x_0, x_1, x_2, \dots, x_n)$, và mỗi đặc trưng được nhân với một hệ số hồi quy tương ứng $(w_0, w_1, w_2, \dots, w_n)$, sau đó tổng hợp các đầu vào $z[4]-[6]$ để tính giá trị hàm sigmoid.[27]

$$z = w_0x_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n \quad (2)$$

$$z = w^T x \quad (3)$$

Trong phương trình (3), w là vector hàng, là hệ số hồi quy, x là vector cột, là dữ liệu đầu vào của bộ phân loại.

Đầu ra là một giá trị nằm giữa 0 và 1. Các dữ liệu có đầu ra lớn hơn 0.5 được phân loại vào lớp 1, và các dữ liệu có đầu ra nhỏ hơn 0.5 được phân loại vào lớp 0. Hiện tại, điều quan trọng là xác định hệ số hồi quy tốt nhất ($w_0, w_1, w_2, \dots, w_n$) trong bộ phân loại này.[27]

i Triển khai Logistic Regression trên Transport Layer Dataset

```
73 from sklearn.linear_model import LogisticRegression
74
75 classifier = LogisticRegression (random_state=123, max_iter=5000)
76 classifier.fit(X_train, y_train)
77
78 y_pred=classifier.predict(X_test)
```

3.1.3.2 DECISION TREE

Trong phương pháp decision tree, thông tin đóng vai trò quan trọng trong việc xác định thuộc tính phù hợp cho mỗi nút của cây quyết định. Do đó, chúng ta có thể chọn thuộc tính có thông tin lượng lớn nhất (giảm nhiều thông tin ở mức tối đa) làm thuộc tính kiểm tra của nút hiện tại. Theo cách này, thông tin cần thiết để phân loại tập mẫu đào tạo thu được từ việc phân chia sau này sẽ là nhỏ nhất. Nói cách khác, việc sử dụng thuộc tính này để phân chia tập mẫu chứa trong nút hiện tại sẽ làm giảm độ đa dạng của các loại khác nhau cho tất cả các tập mẫu con được tạo ra. Do đó, việc sử dụng phương pháp lý thuyết thông tin này sẽ hiệu quả trong việc giảm số lượng chia đối tượng cần phân loại.

Tập S bao gồm s mẫu dữ liệu với thuộc tính loại có thể có m định dạng khác nhau tương ứng với m loại khác nhau của C_i (1,2,3, ..., m). Giả sử i s là số mẫu của C_i . Khi đó, lượng thông tin cần thiết để phân loại một dữ liệu đã cho là

$$I(s_1, s_2, \dots, s_m) = - \sum_{i=1}^m p_i \log(p_i) \quad (1)$$

trong đó $P_i = S_{ij} / |S_j|$ là xác suất của bất kỳ tập con của mẫu dữ liệu nào thuộc về tập C_i

Giả sử A là một thuộc tính có v giá trị khác nhau $\{a_1, a_2, a_3, \dots, a_n\}$. Sử dụng thuộc tính A, tập S có thể được chia thành v tập con $\{S_1, S_2, S_3, \dots, S_n\}$, trong đó S_j chứa các mẫu dữ liệu có thuộc tính A bằng với a_j trong tập S. Nếu thuộc tính A được chọn là thuộc tính để kiểm tra, tức là được sử dụng để phân vùng cho tập mẫu hiện tại, giả sử S_{ij} là một tập con dữ liệu của loại C_i trong tập con S_i , thì entropy thông tin cần thiết là [28]

$$E(A) = \sum_{j=1}^v \frac{S_{1j} + S_{2j} + \dots + S_{mj}}{S} I(S_{1j}, \dots, S_{mj}) \quad (2)$$

Việc sử dụng thuộc tính A trên nút nhánh hiện tại tương ứng với phân chia tập dữ liệu mẫu thu được thông tin đạt được (information gain) là.

$$Gain(A) = I(s_1, s_2, \dots, s_m) - E(A) \quad (3)$$

Thuật toán D3 có thể quyết định bằng cách sử dụng chiến lược tìm kiếm tham lam từ trên xuống và không bao giờ quay lại và xem xét lại lựa chọn trước đó.

Information gain chính là độ đo để chọn thuộc tính tốt nhất trong mỗi bước phát triển cây trong thuật toán ID3.[28]

i Triển khai Decision Tree trên Transport Layer Dataset

```
from sklearn.tree import DecisionTreeClassifier

classifier = DecisionTreeClassifier (random_state=123)
classifier.fit(X_train, y_train)

y_pred=classifier.predict(X_test)
```

3.1.3.3 RANDOM FOREST

Random Forest là một bộ phân loại bao gồm một tập hợp các bộ phân loại cây có cấu trúc $\{h(x, \Theta_k), k=1, \dots\}$ trong đó Θ_k là các vector ngẫu nhiên độc lập và phân phối đồng nhất, và mỗi cây đưa ra một phiếu đơn vị cho lớp phổ biến nhất tại đầu vào x .

Định nghĩa này cho thấy Random Forest là sự kết hợp của nhiều bộ phân loại cây. Trong mô hình RF của Breiman, mỗi cây được trồng dựa trên một tập mẫu huấn luyện và một biến ngẫu nhiên, biến ngẫu nhiên tương ứng với cây thứ k được ký hiệu là Θ_k , giữa hai biến ngẫu nhiên này là độc lập và phân phối đồng nhất. Kết quả là một bộ phân loại $h(x, \Theta_k)$ với x là vector đầu vào. Sau khi chạy k lần, chúng ta thu được chuỗi bộ phân loại $\{h_1(x), h_2(x), h_3(x), \dots, h_k(x)\}$ và sử dụng chúng để tạo thành một hệ thống mô hình phân loại, kết quả cuối cùng của hệ thống này được xác định bằng cách bỏ phiếu đa số, hàm quyết định là:[29]

$$H(x) = \arg \max_Y \sum_{i=1}^k I(h_i(x) = Y) \quad (1)$$

Trong đó, $H(x)$ là sự kết hợp của mô hình phân loại, h_i là một mô hình cây quyết định đơn lẻ, Y là biến đầu ra, $I(\cdot)$ là hàm chỉ số. Đối với một biến đầu vào đã cho, mỗi cây có quyền bỏ phiếu để chọn kết quả phân loại tốt nhất. Quá trình cụ thể được thể hiện trong Hình 1.[29]

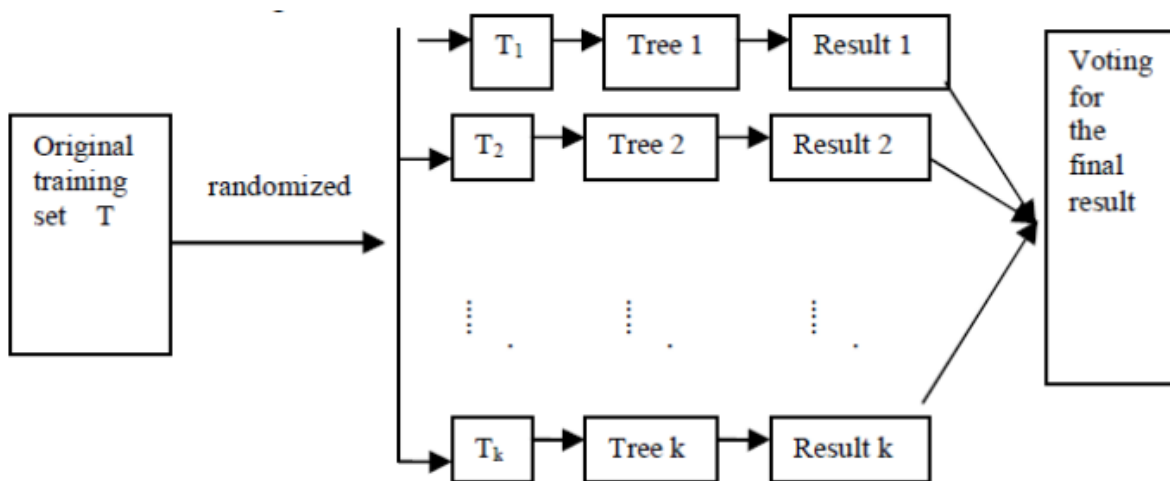


Fig. 1. Random forest schematic

i Triển khai Random Forest trên Transport Layer Dataset

```

from sklearn.ensemble import RandomForestClassifier

classifier = RandomForestClassifier(random_state=123)
classifier.fit(X_train, y_train)

y_pred=classifier.predict(X_test)
  
```

3.1.3.4 MULTI LAYER PERCEPTRON

Hình dưới đây cho thấy một MLP có một lớp ẩn và đầu ra là một số.[30]

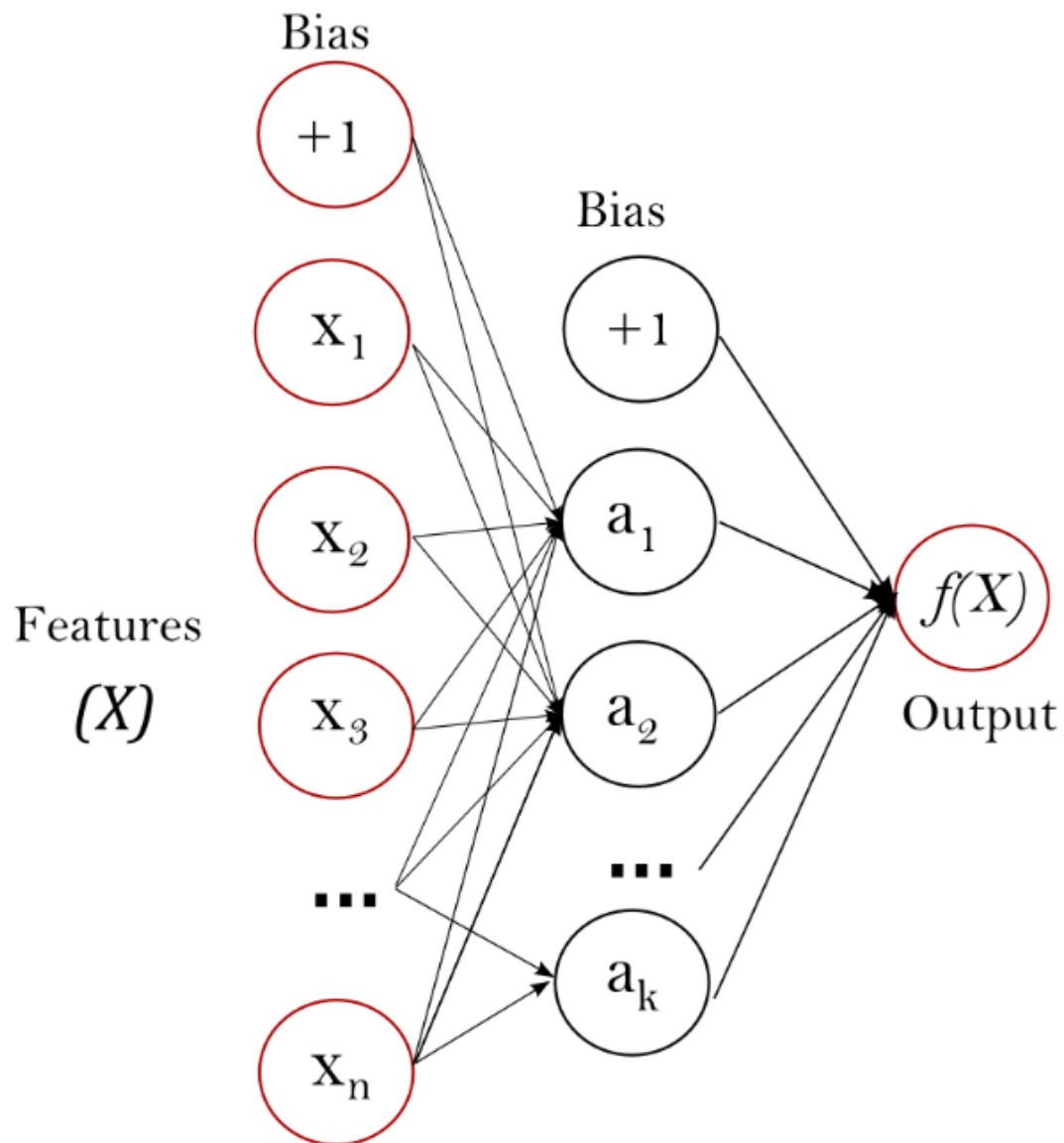


Figure 1 : One hidden layer MLP.

Lớp nằm bên trái cùng, được gọi là lớp đầu vào, bao gồm một tập hợp các neuron đại diện cho các đặc trưng đầu vào $\{x_i \mid x_1, x_2, x_3, \dots, x_m\}$. Mỗi neuron trong lớp ẩn biến đổi giá trị từ lớp trước đó thông qua một tổng tuyến tính được trọng số hóa $\{w_1x_1 + w_2x_2 + w_3x_3 + \dots + w_mx_m\}$, sau đó áp dụng hàm kích hoạt phi tuyến $g(\cdot) : \mathbb{R} \rightarrow \mathbb{R}$ như

hàm tanh hyperbolic. Lớp đầu ra nhận giá trị từ lớp ẩn cuối cùng và biến đổi chúng thành các giá trị đầu ra.[30]

MLP được huấn luyện bằng cách sử dụng Stochastic Gradient Descent (SGD), Adam hoặc L-BFGS. Stochastic Gradient Descent (SGD) cập nhật các tham số bằng cách sử dụng gradient của hàm mất mát đối với một tham số cần thích nghi:

$$w \leftarrow w - \eta \left(\alpha \frac{\partial R(w)}{\partial w} + \frac{\partial Loss}{\partial w} \right)$$

trong đó η là learning rate. Loss là hàm mất mát được sử dụng cho mạng.

i Triển khai MLP trên Transport Layer Dataset

```
classifier = MLPClassifier(random_state=123, solver='adam', max_iter=8000)
classifier.fit(X_train, y_train)

y_pred=classifier.predict(X_test)
```

3.1.3.5 XGBOOST

Giả sử có tổng cộng K cây và sử dụng F để đại diện cho mô hình, sau đó[31]

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i), f_k \in F \quad (1)$$

Hàm mục tiêu là:

$$L = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \quad (2)$$

Trong đó l là hàm mất mát, đại diện cho sai số giữa giá trị dự đoán và giá trị thực tế; Ω là hàm được sử dụng để điều chuẩn và ngăn chặn việc quá khớp (overfitting)[31]

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \|w\|^2 \quad (3)$$

Sau khai triển Taylor bậc hai của hàm mục tiêu và các phép tính khác, chúng ta sẽ có thông tin của hàm mục tiêu sau mỗi phân tách là:

$$Gain = \frac{1}{2} \left[\frac{(\sum_{i \in I_L} g_i)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{(\sum_{i \in I_R} g_i)^2}{\sum_{i \in I_R} h_i + \lambda} + \frac{(\sum_{i \in I} g_i)^2}{\sum_{i \in I} h_i + \lambda} \right] - \gamma \quad (4)$$

Như có thể thấy từ công thức (4), để kiểm soát sự phát triển của cây và ngăn chặn mô hình quá khớp, một ngưỡng chia γ được thêm vào. Các nút lá chỉ được phân tách nếu thông tin lớn hơn γ . Điều này tương đương với việc định giá trước cây trong quá trình tối ưu hàm mục tiêu.[31]

I Triển khai XGBoost trên Transport Layer Dataset

```
from xgboost import XGBClassifier

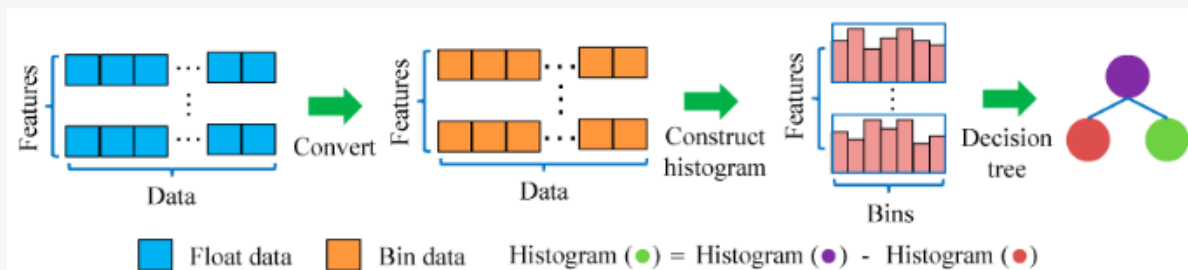
classifier = XGBClassifier(random_state=123)
classifier.fit(X_train, y_train)

y_pred=classifier.predict(X_test)
```

3.1.3.6 LIGHTGBM

Thuật toán dựa trên biểu đồ tần số được thể hiện trong hình dưới. Có thể thấy rằng các giá trị riêng liên tục dạng số thực được rời rạc thành s nhỏ. Sau đó, các khoảng này được sử dụng để xây dựng biểu đồ tần số với độ rộng s. Khi dữ liệu được duyệt lần đầu tiên, các thống kê cần thiết được tích lũy trong biểu đồ tần số. Dựa trên giá trị rời rạc của biểu đồ tần số, điểm chia tối ưu có thể được tìm thấy. Bằng cách sử dụng phương pháp này, chi phí lưu trữ và tính toán có thể được giảm.[32]

Figure 4. Histogram-based decision tree algorithm.



Theo chiến lược phát triển theo cấp độ, các lá cây cùng tầng được chia đồng thời. Điều này thuận lợi cho việc tối ưu hóa với nhiều luồng và kiểm soát độ phức tạp của mô hình. Tuy nhiên, các lá cây cùng tầng được xử lý mà không phân biệt, trong khi chúng có thông tin lợi ích khác nhau. Thông tin lợi ích chỉ ra sự giảm entropi kỳ vọng do chia các nút dựa trên thuộc tính.[33]

$$IG(B, V) = En(B) - \sum_{v \in Values(V)} \frac{|B_v|}{B} En(B_v)$$

trong đó $En(B)$ là thông tin entropy của tập hợp B , p_d là tỷ lệ của B thuộc vào danh mục d , D là số lượng danh mục, v là giá trị của thuộc tính V và B_v là tập con của B mà thuộc tính có giá trị v .

i Triển khai LightGBM trên Transport Layer Dataset

```
from lightgbm import LGBMClassifier

classifier = LGBMClassifier(random_state=123)
classifier.fit(X_train, y_train)

y_pred=classifier.predict(X_test)
```

3.1.3.7 HISTGRADIENTBOOSTING

Phương pháp hGBR được triển khai cùng với các kỹ thuật khác. Ví dụ, nó có thể được triển khai bằng cách sử dụng thư viện máy học scikit-learn, đó là một thư viện cung cấp một cài đặt thử nghiệm của gradient boosting và hỗ trợ phương pháp histogram. Cụ thể, thư viện này cung cấp các lớp

HistGradientBoostingRegressor và HistGradientBoostingClassifier. Theo tài liệu của scikit-learn, việc triển khai hGBR có tốc độ nhanh hơn một số lần so với việc triển khai mặc định của GBR được cung cấp bởi thư viện.

i Triển khai HistGradientBoosting trên Transport Layer Dataset

```
from sklearn.ensemble import HistGradientBoostingClassifier

classifier = HistGradientBoostingClassifier(random_state=123)
classifier.fit(X_train, y_train)

y_pred=classifier.predict(X_test)
```

3.2 CÁC BƯỚC TRIỂN KHAI

3.2.1 TRÍCH XUẤT ĐẶC TRƯNG

Dưới đây là các đặc trưng quan trọng của mạng được trích xuất từ tập dữ liệu bằng cách sử dụng Extra Tree Classifier. Phương pháp này đã tạo ra 10 đặc trưng quan trọng của mạng bao gồm sinpkt, sttl, sbytes, rate, swin, dur, dload, dttl, spkts .

3.2.2 TIỀN XỬ LÝ DỮ LIỆU

Khi giải quyết bài toán phát hiện bất thường không giám sát dưới dạng bài toán phân loại nhị phân, thuật toán phát hiện bất thường cố gắng phân loại mẫu là bình thường hay không bình thường sau khi được huấn luyện trước đó trên dữ liệu chỉ chứa các mẫu bình thường. Để huấn luyện thuật toán, chúng ta sử dụng dữ liệu không chứa dữ liệu bất thường (chỉ chứa các mẫu bình thường) được tạo ra bằng cách lấy mẫu từ tất cả các mẫu bình thường trong tập dữ liệu gốc. Dữ liệu bình thường được lấy ngẫu nhiên từ 80% số lượng mẫu bình thường và sau đó được chuẩn hóa bằng phương pháp IQR.

Sau khi huấn luyện thuật toán trên dữ liệu bình thường, chúng ta cần huấn luyện mẫu kiểm tra chứa cả dữ liệu bình thường và dữ liệu bất thường. Mẫu kiểm tra được trích xuất từ tập dữ liệu cân bằng đã được đề cập trước đó và bao gồm cả ba lớp dữ liệu. Chúng ta lấy một phần nhỏ (2,4%) của mẫu từ tập dữ liệu cân bằng bằng phương pháp lấy mẫu ngẫu nhiên, sau đó áp dụng phương pháp chuẩn hóa IQR. Các nhãn lớp cho các bản ghi thuộc nhóm bình thường (inliers) được đánh dấu là '1', trong khi các nhãn lớp cho các gia đình Mirai và Gafgyt được đánh dấu là '-1' (outliers). Tỷ lệ giữa dữ liệu bình thường và dữ liệu kiểm tra là khoảng 50% và 50% tương ứng.

3.2.3 CHỌN ĐẶC TRƯNG

Trích xuất đặc trưng là một phương pháp được sử dụng để tạo mô hình dự đoán bằng cách chọn lọc các biến đầu vào quan trọng. Việc giảm số chiều dữ liệu trong quá trình phân loại là rất cần thiết để tiết kiệm thời gian tính toán và cải thiện khả năng phân tích dự đoán của mô hình. Trong nghiên cứu về dữ liệu của tổ chức, nhóm em đã áp dụng một phương pháp học máy để phân tích thông tin và sau đó lựa chọn những đặc trưng quan trọng nhất để áp dụng vào mô hình nhận dạng IoT. Tập dữ liệu UNSW_NB15 mà nhóm em sử dụng có tổng cộng 45 đặc trưng. Tuy nhiên, chỉ có 17 đặc trưng quan trọng được chọn lọc để sử dụng trong mô hình phân loại.

3.2.4 XÂY DỰNG MÔ HÌNH

Sau khi lựa chọn thuật toán, em đã xây dựng mô hình bằng cách đưa dữ liệu huấn luyện vào thuật toán đã chọn. Trong quá trình này có thể điều chỉnh các siêu tham số của thuật toán để tối ưu hóa hiệu suất của mô hình.

3.2.5 ĐÁNH GIÁ MÔ HÌNH

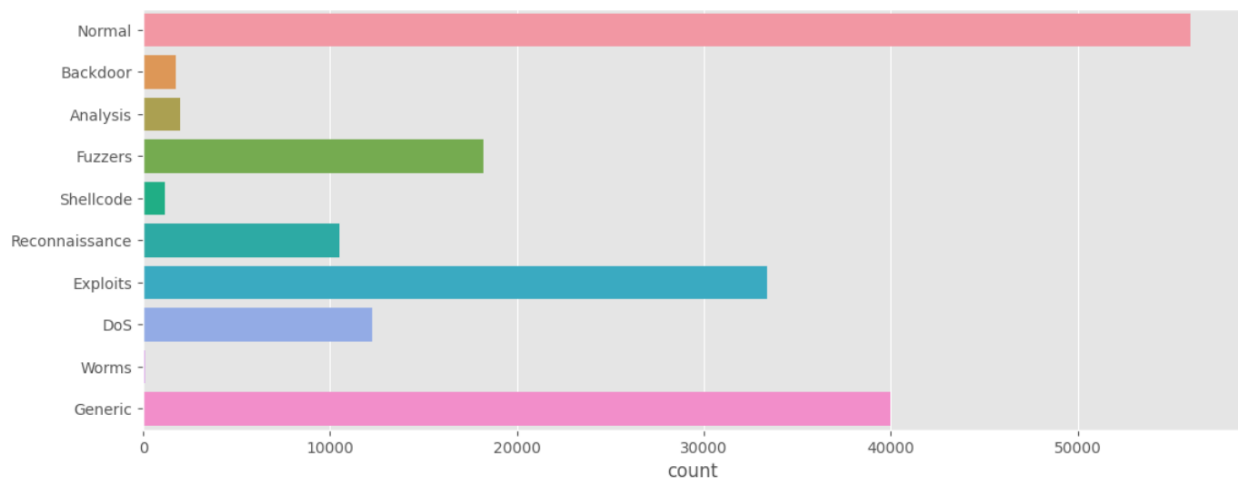
Sau khi xây dựng mô hình, nhóm em đã đánh giá hiệu suất của mô hình bằng cách sử dụng tập dữ liệu kiểm tra. Các phép đo như độ chính xác (accuracy), độ phân loại (precision), độ phủ (recall) và F1-score được sử dụng để đánh giá khả năng dự đoán của mô hình.

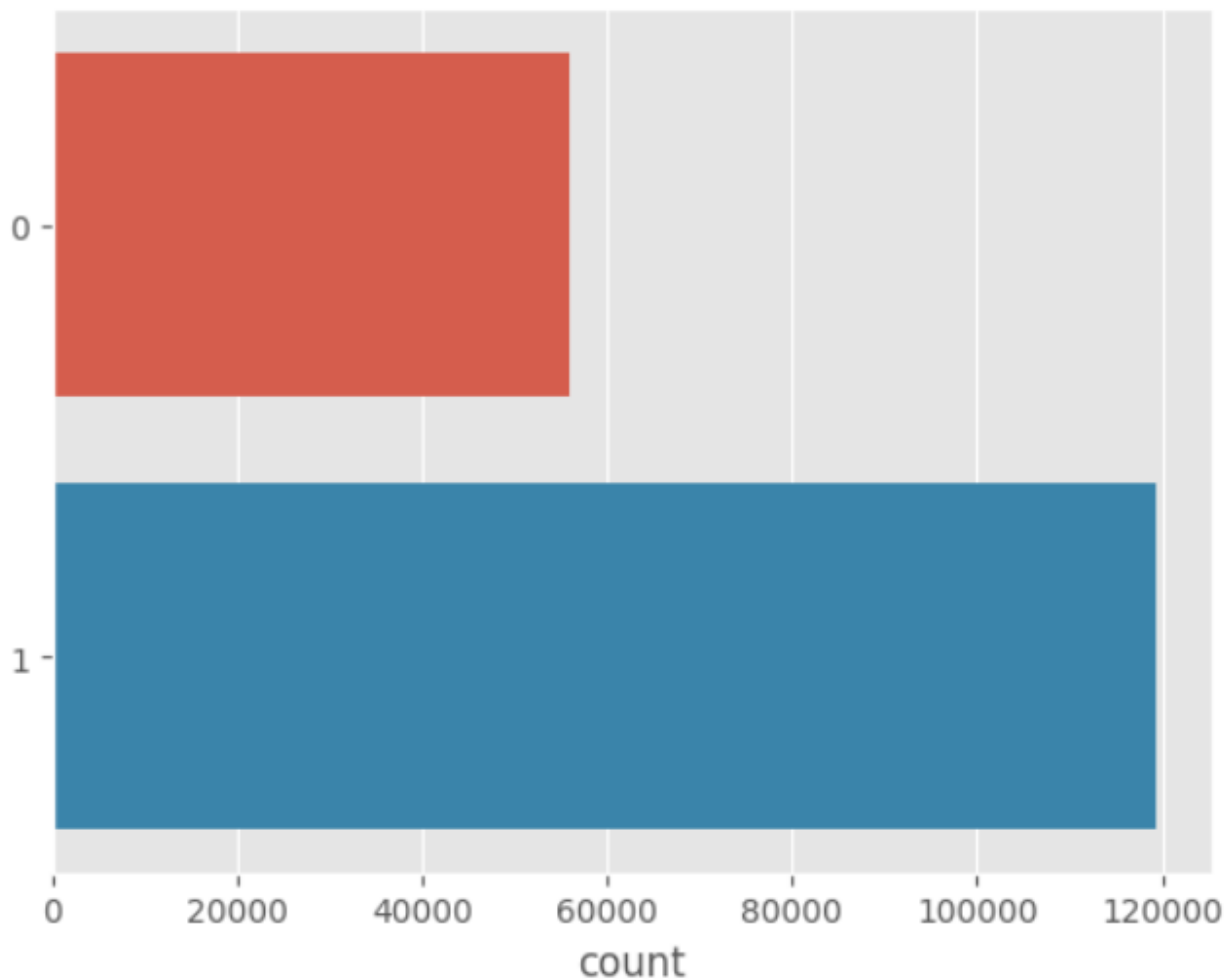
4 KẾT QUẢ VÀ THẢO LUẬN

Bài báo cáo sẽ trình bày kết quả nghiên cứu dựa trên các metric như điểm F1, Độ chính xác (Accuracy), Precision, Recall và AUC (Diện tích dưới đường cong ROC). Những giá trị độ chính xác này được biểu thị dưới dạng phần trăm. Cuối cùng, bài báo cáo sẽ so sánh các thuật toán dựa trên kết quả đạt được. Bài báo cáo này sử dụng Jupyter Notebook trong môi trường Windows với 16GB RAM và chip Intel Core i7 12th.

4.1 PHÂN LOẠI TẤN CÔNG

Dataset UNSW-NB15 bao gồm 9 loại tấn công: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms.





Như đã giới thiệu ở trên hình 1 cho thấy số lượng của từng loại tấn công từ đó cho chúng ta cái nhìn tổng quát tập dữ liệu của chúng ta.

Quá trình thử nghiệm sẽ chia thành 2 hành vi "Bình thường" hoặc "Tấn công". Hình 2 cung cấp chi tiết về số lượng và giá trị của mỗi lớp tấn công trong các tập dữ liệu, trong đó 0 đại diện cho "Bình thường" và 1 đại diện cho "Hành vi Tấn công". Chúng ta có thể thấy rằng tập dữ liệu đã cân bằng đủ cho biến phản hồi nhị phân về hành vi hoạt động.

4.2 SO SÁNH CÁC THUẬT TOÁN SỬ DỤNG KỸ THUẬT CROSS VALIDATION

Cross-validation (CV) là một kỹ thuật phân chia và đánh giá mô hình trên dữ liệu để đánh giá hiệu suất tổng quát hóa của mô hình trên dữ liệu mới. Thay vì chỉ chia dữ liệu thành hai phần (huấn luyện và kiểm tra), CV chia dữ liệu thành nhiều phần nhỏ hơn, huấn luyện và đánh giá mô hình trên các phần này theo một cách xác định. [42]

Có một số kiểu cross-validation phổ biến, bao gồm:

1. **K-fold Cross-Validation:** Đây là kiểu CV phổ biến nhất. Dữ liệu được chia thành K phần bằng nhau (gọi là fold). Mô hình được huấn luyện trên K-1 fold và được đánh giá trên fold còn lại. Quá trình này được lặp lại K lần với mỗi fold được sử dụng lần lượt làm tập kiểm tra. Kết quả cuối cùng được tính trung bình từ K lần đánh giá. [43]
2. **Stratified K-fold Cross-Validation:** Đây là một biến thể của K-fold Cross-Validation, nhưng đảm bảo rằng sự phân chia dữ liệu trong mỗi fold đại diện cho tỷ lệ các lớp trong dữ liệu ban đầu. Điều này hữu ích khi dữ liệu không cân bằng và muốn đảm bảo rằng mô hình được đánh giá đúng tỷ lệ các lớp.[43]
3. **Leave-One-Out Cross-Validation (LOOCV):** Mỗi mẫu dữ liệu trong tập huấn luyện chỉ được sử dụng một lần làm tập kiểm tra và K-1 lần làm tập huấn luyện. Đây là một kiểu CV rất tốn kém tính toán, nhưng nó có thể cung cấp ước lượng chính xác nhất về hiệu suất tổng quát hóa của mô hình. [43]
4. **Hold-out Cross-Validation:** Dữ liệu được chia thành hai phần: một phần được sử dụng để huấn luyện mô hình và phần còn lại được sử dụng để

đánh giá hiệu suất. Thông thường, tỷ lệ phân chia là 70-30 hoặc 80-20. Tuy nhiên, đối với tập dữ liệu nhỏ, có thể sử dụng tỷ lệ 60-40 hoặc 50-50.[43]

Kỹ thuật cross-validation giúp đánh giá hiệu suất mô hình một cách khách quan và đáng tin cậy. Nó cho phép ta kiểm tra khả năng tổng quát hóa của mô hình trên dữ liệu mới mà chưa từng được sử dụng trong quá trình huấn luyện.

Ở bài báo cáo này sẽ sử dụng phương pháp Stratified K-fold Cross-Validation chia tập dữ liệu Training thành 5 phần trong đó 4 phần được sử dụng để Training và 1 phần dùng để Testing và lặp lại 5 lần. Kết quả được hiển thị như bên dưới: [42]

	Model Name	CV Fit Time	CV Accuracy mean	CV Precision mean	CV Recall mean	CV F1 mean	CV AUC mean
5	LightGBM	2.373257	0.956359	0.959820	0.976772	0.968221	0.993038
6	HistGradientBoosting	4.563331	0.955127	0.958778	0.976035	0.967329	0.992793
2	RandomForest	52.158373	0.959736	0.963049	0.978381	0.970654	0.993565
4	XGBoost	7.940769	0.958840	0.962620	0.977485	0.969995	0.993712
1	DecisionTree	3.954414	0.948552	0.962916	0.961438	0.962176	0.942744
3	MultiLayerPerceptron	315.985334	0.947896	0.956921	0.967044	0.961917	0.990449
0	LogisticRegression	6.948703	0.927421	0.912972	0.987498	0.948773	0.969048

Kết quả ở trên cho thấy kỹ thuật cross validation đạt hiệu suất rất tốt khi toàn bộ 7 thuật toán áp dụng kỹ thuật này đều đạt trên 95%. Con số này cho thấy chúng ta đã sẵn sàng cho việc kiểm tra trên tập Testing.

4.3 SO SÁNH CÁC THUẬT TOÁN BẰNG F1-SCORE

Precision (còn được gọi là Positive Predictive Value) là một chỉ số đánh giá trong bài toán phân loại. Nó đo lường tỷ lệ các dữ liệu được dự đoán là positive (tích cực) mà thực sự thuộc lớp positive so với tổng số dữ liệu được dự đoán là positive. [36]

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

Trong đó:

- + True Positives (TP) là số lượng dữ liệu thực tế thuộc lớp positive mà mô hình dự đoán đúng.
- + False Positives (FP) là số lượng dữ liệu thực tế thuộc lớp negative mà mô hình dự đoán nhầm là positive.

Precision cao cho thấy mô hình có khả năng phân loại chính xác các dữ liệu positive. Điều này quan trọng trong các tình huống mà việc xác định chính xác các trường hợp positive là ưu tiên, để tránh việc đưa ra những dự đoán sai và gây nhầm lẫn. Tuy nhiên, precision có thể bị ảnh hưởng bởi tỷ lệ dữ liệu negative (âm tính) lớn hoặc sự thiếu cân bằng giữa các lớp dữ liệu.[37]

Recall (còn được gọi là True Positive Rate hoặc Sensitivity) là một chỉ số đánh giá trong bài toán phân loại. Nó đo lường tỷ lệ các dữ liệu thuộc lớp positive (tích cực) mà mô hình dự đoán đúng so với tổng số dữ liệu thuộc lớp positive trong tập dữ liệu thực tế. [38]

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

Trong đó:

+ True Positives (TP) là số lượng dữ liệu thực tế thuộc lớp positive mà mô hình dự đoán đúng.

+ False Negatives (FN) là số lượng dữ liệu thực tế thuộc lớp positive mà mô hình dự đoán sai.

Recall cao cho thấy mô hình có khả năng phát hiện và bắt lấy đúng nhiều dữ liệu positive. Điều này hữu ích trong các tình huống mà việc bỏ sót các trường hợp positive có thể gây hậu quả nghiêm trọng, ví dụ như phát hiện bệnh hiểm nghèo. Tuy nhiên, recall có thể bị ảnh hưởng bởi tỷ lệ dữ liệu negative (âm tính) lớn hoặc sự thiếu cân bằng giữa các lớp dữ liệu.[36]

F1 score là một độ đo kết hợp giữa precision (độ chính xác) và recall (độ bao phủ) được sử dụng để đánh giá hiệu suất của một mô hình phân loại. [38]

F1 score là trung harmonics của precision và recall, và được tính bằng công thức:

$$\begin{aligned} \text{F1 Score} &= \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} \\ &= \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \end{aligned}$$

F1 score thường được sử dụng trong các bài toán phân loại mà cần cân nhắc cả độ chính xác và độ bao phủ. Nếu một mô hình có F1 score cao, điều đó có nghĩa là mô hình đạt được cân bằng giữa việc dự đoán chính xác các trường hợp positive và đảm bảo bao phủ đầy đủ các trường hợp positive trong tập dữ liệu. [38]

Bảng dưới đây phân tích từng tính toán riêng lẻ dưới quy trình giảm thiểu yếu tố riêng biệt của chúng dựa trên F1 Score

Classification		F1 Score (Percentage)
1	Logistic Regression	0.821391
2	DecisionTree	0.884491
3	RandomForest	0.893433
4	Multi Layer Perceptron	0.876010
5	XGBoost	0.892927
6	LightGBM	0.895905
7	HistGradientBoosting	0.894972
8	KNN	0.872784

Như có thể thấy, thuật toán LightGBM có F1 score cao nhất do thuật toán này áp dụng phương pháp Boosting. Nhưng nhìn chung các F1 score của các thuật toán vẫn khá cao nên kết này là chấp nhận được.

4.4 SO SÁNH CÁC THUẬT TOÁN BẰNG ACCURACY

Accuracy (độ chính xác) là một độ đo được sử dụng để đánh giá hiệu suất của một mô hình phân loại. Nó đo lường tỷ lệ các dự đoán chính xác (đúng) so với tổng số lượng dự đoán. [39]

Accuracy được tính bằng công thức:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy đo lường khả năng của mô hình phân loại trong việc dự đoán chính xác các trường hợp positive và negative. Một accuracy cao cho thấy mô hình có khả năng phân loại chính xác và đáng tin cậy. Tuy nhiên, accuracy không phù hợp trong một số trường hợp khi dữ liệu mất cân bằng, tức là tỷ lệ các lớp positive và negative không đồng đều. Trong những trường hợp này, các độ đo khác như F1 score, precision và recall thường được sử dụng để đánh giá hiệu suất mô hình một cách toàn diện hơn. [39]

Classification		Accuracy (Percentage)
1	Logistic Regression	0.774498
2	DecisionTree	0.862605
3	RandomForest	0.870731
4	Multi Layer Perceptron	0.849572
5	XGBoost	0.870463
6	LightGBM	0.873937
7	HistGradientBoosting	0.872783
8	KNN	0.872784

Ở bảng trên vẫn cho thấy LightGBM vẫn là thuật toán tốt nhất về Accuracy. Bảng dưới đây đã sắp xếp các thuật toán có giá trị F1 từ cao xuống thấp để dễ nhận xét:

	Model Name	CV Fit Time	CV Accuracy mean	CV Precision mean	CV Recall mean	CV F1 mean	CV AUC mean	Test Accuracy	Test Precision	Test Recall	Test F1	Test AUC
5	LightGBM	2.373257	0.956359	0.959820	0.976772	0.968221	0.993038	0.873937	0.821407	0.985264	0.895905	0.984774
6	HistGradientBoosting	4.563331	0.955127	0.958778	0.976035	0.967329	0.992793	0.872783	0.820421	0.984426	0.894972	0.983678
2	RandomForest	52.158373	0.959736	0.963049	0.978381	0.970654	0.993565	0.870731	0.818020	0.984161	0.893433	0.977035
4	XGBoost	7.940769	0.958840	0.962620	0.977485	0.969995	0.993712	0.870463	0.819376	0.980985	0.892927	0.982660
1	DecisionTree	3.954414	0.948552	0.962916	0.961438	0.962176	0.942744	0.862605	0.823384	0.955396	0.884491	0.854642
3	MultiLayerPerceptron	315.985334	0.947896	0.956921	0.967044	0.961917	0.990449	0.849572	0.801961	0.965124	0.876010	0.961672
0	LogisticRegression	6.948703	0.927421	0.912972	0.987498	0.948773	0.969048	0.774498	0.728317	0.941741	0.821391	0.886594

4.5 SO SÁNH CÁC THUẬT TOÁN BẰNG AUC

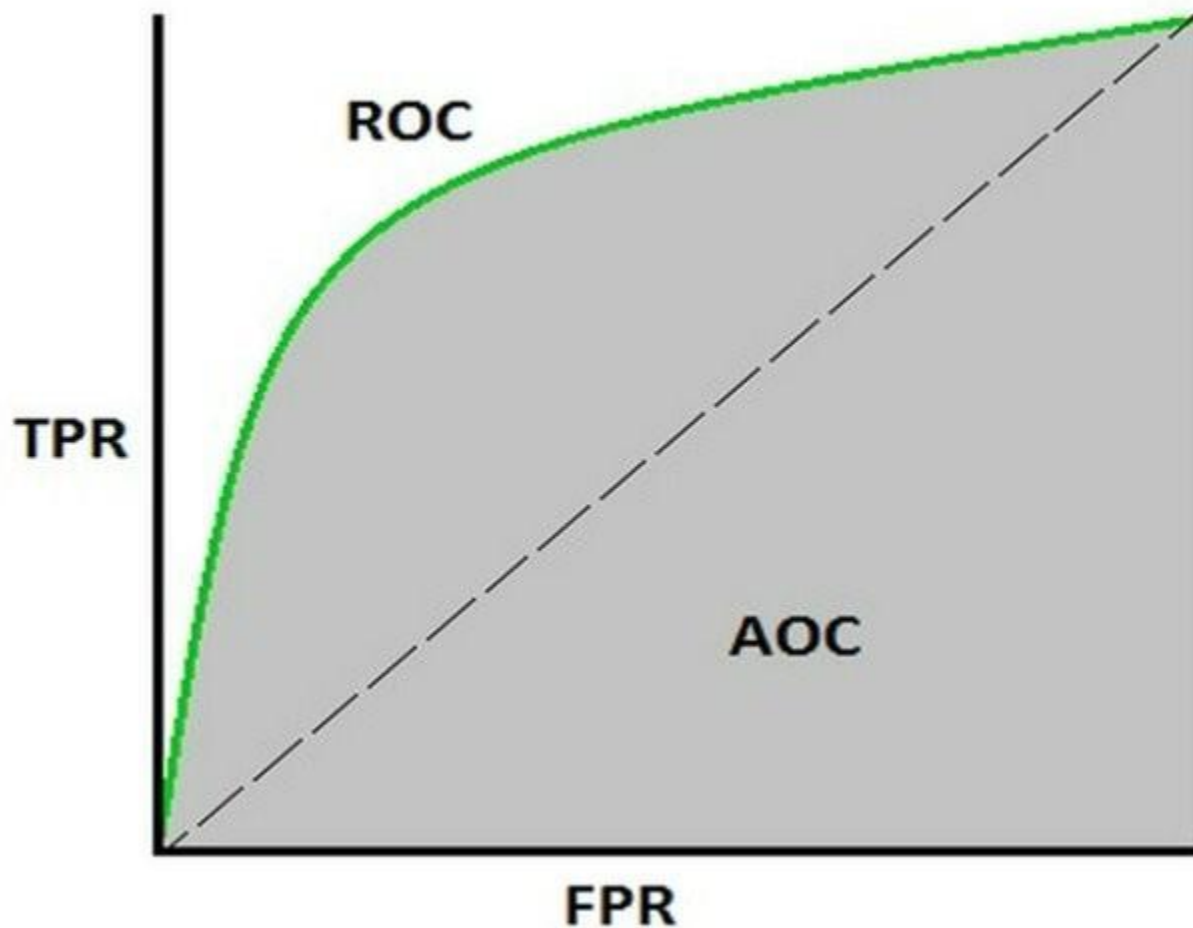
AUC là một độ đo được sử dụng để đánh giá hiệu suất của một mô hình phân loại trong việc phân loại hai lớp. Đường cong ROC (Receiver Operating Characteristic curve) biểu thị khả năng phân loại của mô hình dựa trên độ nhạy và đặc specificity. [40]

Đường cong ROC là biểu đồ thể hiện sự biến đổi của tỷ lệ True Positive (TPR) trên tung độ (trục y) và tỷ lệ False Positive (FPR) trên hoành độ (trục x) khi ngưỡng phân loại thay đổi. Một mô hình phân loại tốt sẽ có đường cong ROC gần với góc trên bên trái của biểu đồ, tức là có TPR cao và FPR thấp.[41]

AUC là diện tích tích phân dưới đường cong ROC, được tính từ các giá trị TPR và FPR tại các điểm trên đường cong. AUC cung cấp một số lượng tổng quan về hiệu suất phân loại của mô hình, mà không phụ thuộc vào ngưỡng cắt cụ thể. Nó không chỉ cho biết mức độ phân loại tốt hay xấu của mô hình, mà còn cho biết khả năng phân biệt giữa các lớp positive và negative. [40]

AUC có giá trị nằm trong khoảng từ 0 đến 1, và càng gần 1 thì mô hình càng có khả năng phân loại tốt hơn. Một AUC bằng 0.5 cho thấy mô hình có khả năng phân loại ngẫu nhiên, trong khi AUC lớn hơn 0.5 cho thấy mô hình có khả năng phân loại tốt hơn so với ngẫu nhiên. [40]

Điểm cắt thích hợp trên đường cong ROC để đưa ra quyết định phân loại thường được xác định bằng cách chọn ngưỡng giới hạn có giá trị phù hợp. AUC là một độ đo phổ biến trong việc so sánh hiệu suất của các mô hình phân loại và giúp định rõ độ chính xác và đặc specificity của mô hình.[41]

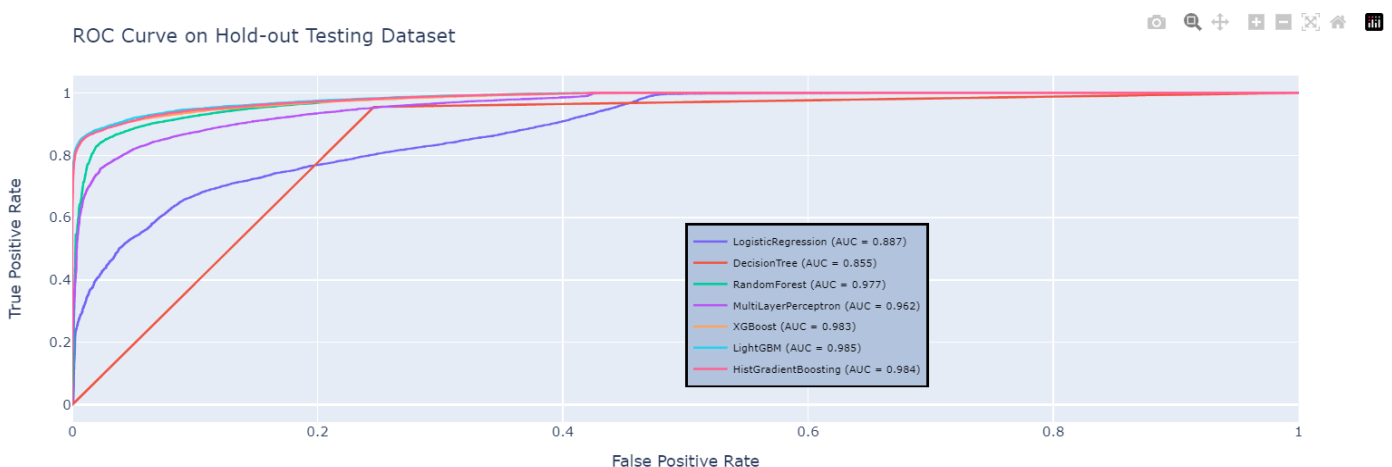


Classification	AUC
----------------	-----

1	Logistic Regression	0.886594
2	DecisionTree	0.854642
3	RandomForest	0.977035
4	Multi Layer Perceptron	0.961672
5	XGBoost	0.982660
6	LightGBM	0.984774
7	HistGradientBoosting	0.983678
8	KNN	0.872784

Như có thể thấy ở bảng trên, LightGBM là thuật toán có chỉ số AUC cao nhất. Tuy nhiên những thuật toán khác cũng có chỉ số AUC rất cao và gần như bằng 1 nên ta kết luận là các thuật toán trên vẫn rất thích hợp với dataset này.

Dưới đây là biểu đồ trực quan về giá trị AUC của các thuật toán



5 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 KẾT LUẬN

Dự kiến sẽ có 75 triệu người sử dụng thiết bị IoT vào năm 2025. Tuy nhiên, điều này cũng mang đến những rủi ro khi các khuyết điểm trong cơ sở hạ tầng có thể dễ dàng bị khai thác bằng sự phát triển không giới hạn của các sản phẩm IoT. Do đó, việc phát hiện các cuộc tấn công botnet IoT trở nên rất quan trọng theo sự phát triển của công nghệ. Ở đây bài báo cáo này đã lựa chọn phương pháp xác định dựa trên bất thường vì nó khả thi trong việc nhận ra những lỗ hổng ẩn. Hiệu suất của bộ phân loại học máy trong việc phát hiện botnet IoT đã được đánh giá trong nghiên cứu này. Ngoài ra, thời gian cần thiết để huấn luyện và xác định trong các mô hình học đã được nghiên cứu. Thuật toán LightGBM đạt đến 87% độ chính xác và 82% precision trong 2.37 giây. Tài nguyên máy tính cần thiết cho nghiên cứu này là tương đối nhỏ, làm cho nó trở thành một lựa chọn thích hợp trong môi trường IoT. Mục tiêu dài hạn của em cho mô hình này là tăng cường nó hơn nữa và làm việc để xây dựng hệ thống ngăn chặn xâm nhập song song với việc phát hiện các cuộc tấn công botnet.

Với sự phát triển mới và việc sử dụng không giới hạn các thiết bị IoT, các khuyết điểm trong hệ thống có thể dễ dàng bị khai thác. Sự phát triển của Botnet Mirai và các biến thể khác là một ví dụ về trường hợp như vậy. Gần đây, cuộc tấn công botnet lớn nhất trên Imperva, một ứng dụng web dựa trên internet, đã được tiến hành, với 400.000 thiết bị IoT được sử dụng để tạo ra cuộc tấn công DDoS .

Những cuộc tấn công botnet này vào các thiết bị IoT liên quan trực tiếp đến việc thiếu nền tảng bảo mật trong các thiết bị do tính toán chi phí. Do đó, với sự tiến

bộ của công nghệ, việc nhận dạng các cuộc tấn công botnet IoT đã trở thành một phần của công việc hàng ngày. Bài báo cáo đã chọn phương pháp xác định dựa trên bất thường vì nó hiệu quả trong việc phát hiện các khuyết điểm ẩn. Sau khi thực hiện nghiên cứu đã có một số kết quả quan trọng trong việc dự đoán trạng thái của một thiết bị IoT, có thể là "tấn công" hoặc "vô hại". Các bộ phân loại như LightGBM cho kết quả tốt nhất về độ chính xác. Mặc dù rightGBM hoạt động tốt trong việc phân loại nhưng nó vẫn có nguy cơ bị quá khớp.

Tóm lại kết quả nghiên cứu của bài báo cáo dựa trên các phương pháp giảm chiều khác nhau cung cấp các tình huống khác nhau cho việc chạy các bộ phân loại. Tài nguyên máy tính cần thiết cho nghiên cứu này là tương đối nhỏ, làm cho nó trở thành một lựa chọn thích hợp trong môi trường IoT. Mục tiêu dài hạn của em cho mô hình này là tăng cường nó hơn nữa và làm việc để xây dựng hệ thống ngăn chặn xâm nhập song song với việc phát hiện các cuộc tấn công botnet.

5.2 HƯỚNG PHÁT TRIỂN

Trong lĩnh vực IOT CYBERSECURITY, việc tích hợp các kỹ thuật Machine Learning đã cho thấy những kết quả đầy hứa hẹn trong việc phát hiện và giảm thiểu các mối đe dọa an ninh đa dạng. Khi chúng ta tiến xa hơn, có một số hướng tiềm năng cho nghiên cứu và phát triển trong lĩnh vực này.

- ◆ **Nâng cao Hệ thống Phát hiện Xâm nhập:** Một hướng đi cho công việc tương lai là tinh chỉnh và cải tiến khả năng của hệ thống phát hiện xâm nhập (IDS) cho môi trường IoT. Các thuật toán Máy học có thể được tối ưu hóa để nhận dạng và phân loại chính xác các hoạt động độc hại trong thời gian thực, đảm bảo phản ứng và giảm thiểu kịp thời các sự cố an ninh.

- ◆ **Phát hiện Bất thường dựa trên Hành vi:** Phát triển các mô hình phát hiện bất thường tiên tiến được tùy chỉnh đặc biệt cho các thiết bị và mạng IoT là một lĩnh vực quan trọng khác để khám phá trong tương lai. Bằng cách phân tích các mẫu hành vi của các thiết bị IoT, các thuật toán Máy học có thể nhận ra những sự sai lệch so với hành vi bình thường.
- ◆ **Điều chỉnh mô hình Machine Learning cho IoT:** Nhiệm vụ quan trọng trong tương lai là tối ưu hóa và điều chỉnh các mô hình Machine Learning để phù hợp với đặc thù của môi trường IoT. Điều này bao gồm việc xử lý dữ liệu thô, giảm thiểu sự ảnh hưởng của nhiễu và đảm bảo tính khả diễn giải của các mô hình để tăng cường sự tin cậy và sử dụng thực tế.
- ◆ **Học máy dựa trên mạng nơ-ron hồi quy:** Sử dụng mạng nơ-ron hồi quy (Recurrent Neural Network) trong việc phát hiện và ngăn chặn các cuộc tấn công DDoS có thể là một hướng tiếp cận tiềm năng. Học máy dựa trên mạng nơ-ron hồi quy có khả năng phân tích dữ liệu chuỗi thời gian và xác định các mô hình tấn công DDoS tiềm năng, từ đó giúp đưa ra các biện pháp phòng ngừa kịp thời.
- ◆ **Tích hợp học sâu và học tăng cường:** Kết hợp các kỹ thuật học sâu (Deep Learning) và học tăng cường (Reinforcement Learning) có thể mang lại khả năng phát hiện và phản ứng tốt hơn đối với các cuộc tấn công IoT. Sử dụng các mô hình học sâu như mạng nơ-ron gia đình Convolutional Neural Network (CNN) và mô hình học tăng cường để tối ưu hóa việc phân loại và xử lý các sự cố an ninh IoT.

6 Ứng dụng ML vào thực tế

Nhận dạng khuôn mặt là gì? Ứng dụng và cách thức hoạt động?



Nhận dạng khuôn mặt là gì?

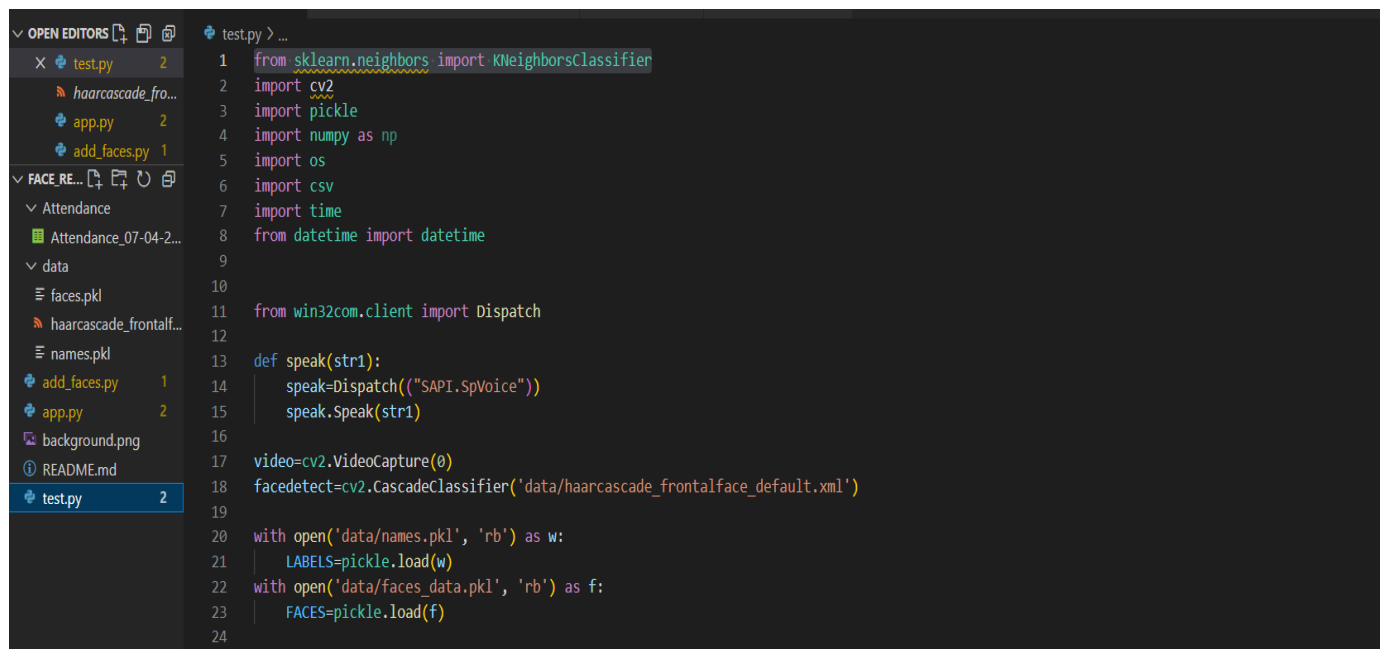
Phần mềm nhận dạng khuôn mặt có vô số ứng dụng trong thị trường tiêu dùng cũng như ngành an ninh và giám sát. Có hai nhiệm vụ chính mà các mô hình nhận dạng khuôn mặt thực hiện. Đầu tiên là xác minh, là nhiệm vụ so sánh khuôn mặt đầu vào mới với danh tính đã biết. Một ví dụ điển hình cho việc này là việc mở khóa điện thoại thông minh bằng nhận dạng khuôn mặt. Khi thiết lập hệ thống, điện thoại sẽ đăng ký khuôn mặt của bạn là chủ sở hữu điện thoại. Do đó, nhiệm vụ duy nhất khi mở khóa là so sánh khuôn mặt đầu vào mới với khuôn mặt đã đăng ký của bạn trên thiết bị. Thứ hai là nhận dạng, là nhiệm vụ so sánh một khuôn mặt đầu vào với cơ sở dữ liệu gồm nhiều nhận dạng khuôn mặt. Nhiệm vụ này thường được sử dụng cho các hệ thống an ninh và giám sát. Một ví dụ điển hình là nhận dạng khuôn mặt trong thực thi pháp luật. Trên trang web INTERPOL,

có một phần pháp y giải thích cách họ sử dụng nhận dạng khuôn mặt để xác định những người cần quan tâm tại sân bay và cửa khẩu biên giới.

Nhận dạng khuôn mặt hoạt động như thế nào?

Với rất nhiều sự quan tâm đến lĩnh vực này, các nhà khoa học dữ liệu phát triển các phương pháp tiếp cận mới để nhận dạng khuôn mặt mỗi năm. Phần này sẽ thảo luận ngắn gọn những điều cơ bản về cách hoạt động của các mô hình nhận dạng khuôn mặt và sự khác biệt chính giữa hai phương pháp tạo nhúng khuôn mặt. Ở mức cơ bản nhất, các mô hình nhận dạng khuôn mặt tuân theo các bước sau: Một hình ảnh đầu vào được đưa vào thuật toán. Thuật toán tạo ra sự nhúng khuôn mặt cho hình ảnh đầu vào. Thuật toán so sánh việc nhúng khuôn mặt của hình ảnh đầu vào với việc nhúng các khuôn mặt đã biết trong cơ sở dữ liệu.

Source code:

A screenshot of a code editor with a dark theme. The left sidebar shows a file explorer with a project structure including folders like 'Attendance', 'data', and files like 'faces.pkl', 'names.pkl', 'add_faces.py', 'app.py', 'background.png', 'README.md', and 'test.py'. The 'test.py' file is selected and open in the main editor. The code in the editor is a Python script for face recognition using K-Nearest Neighbors (KNN) classifier. It imports necessary libraries like sklearn, cv2, pickle, numpy, os, csv, time, datetime, and win32com.client. It defines a 'speak' function for audio output, initializes a video capture and a face cascade classifier, and loads pre-trained data (names and faces) from pickle files. The code is as follows:

```
1 from sklearn.neighbors import KNeighborsClassifier
2 import cv2
3 import pickle
4 import numpy as np
5 import os
6 import csv
7 import time
8 from datetime import datetime
9
10
11 from win32com.client import Dispatch
12
13 def speak(str1):
14     speak=Dispatch(("SAPI.SpVoice"))
15     speak.Speak(str1)
16
17 video=cv2.VideoCapture(0)
18 facedetect=cv2.CascadeClassifier('data/haarcascade_frontalface_default.xml')
19
20 with open('data/names.pkl', 'rb') as w:
21     LABELS=pickle.load(w)
22 with open('data/faces_data.pkl', 'rb') as f:
23     FACES=pickle.load(f)
24
```

7 TÀI LIỆU THAM KHẢO

- [1] <https://www.mdpi.com/2079-9292/11/9/1502>
- [2] <https://sci-hub.se/10.1109/CCWC.2019.8666588>
- [3] <https://sci-hub.se/10.1109/TNSM.2020.2971213>
- [4] <https://sci-hub.se/10.1109/ICoICT.2019.8835211>
- [5] Kevin P. Murphy "Machine Learning. A Probabilistic Perspective", The MIT Press
- [6] Kolias, Constantinos & Kambourakis, Georgios & Stavrou, Angelos & Voas, Jeffrey. "DDoS in the IoT: Mirai and other botnets", Computer .
- [7] <https://arxiv.org/ftp/arxiv/papers/2204/2204.03433.pdf>
- [8] <https://history-computer.com/dos-guide/>
- [9] <https://sci-hub.se/10.1109/ICOEI.2019.8862720>
- [10] S. Gibson "DRDoS: Description and analysis of a potent, increasingly prevalent, and worrisome internet attack," Gibson Research Corporation
- [11] Sách:]Machine Learning for Cybersecurity Cookbook" (Cyrus Malekpour)
- [12] Sách:Applied Machine Learning for IoT Security" (Reza M. Parizi)
- [13] Sách:What is the Internet of Things? WIRED explains – MATT BURGESS
- [14] Sivarajah. "Critical analysis of Big Data challenges and analytical methods." Journal of Business Research
- [15] Porter, M. E., & Heppelmann, J.E . "How Smart, Connected Products Are Transforming Competition."
- [16] Manyika, J., et al. "Unlocking the Potential of the Internet of Things." McKinsey Global Institute, McKinsey & Company.

- [17] <https://core.ac.uk/download/pdf/329117789.pdf>
- [18] <https://sci-hub.se/10.1109/ICCAD.2014.7001385>
- [19] <https://ieeexplore.ieee.org/abstract/document/7444889>
- [20] <https://ieeexplore.ieee.org/abstract/document/7856730/>
- [21] <https://www.mdpi.com/2073-8994/13/5/866>
- [22] https://link.springer.com/chapter/10.1007/978-3-030-22475-2_1
- [23] <https://www.sciencedirect.com/science/article/pii/B9780128189467000032>
- [24] <https://sci-hub.se/10.1109/ICCSNT47585.2019.8962457>
- [25] https://www.researchgate.net/profile/Kajal-Saraswat/publication/298175900_Decision_Tree_Based_Algorithm_for_Intrusion_Detection/links/56e68b2808ae98445c223707/Decision-Tree-Based-Algorithm-for-Intrusion-Detection.pdf
- [26] <https://sci-hub.se/10.1109/TSMCC.2008.923876>
- [27] <https://sci-hub.se/10.1109/ICCSNT47585.2019.8962457>
- [28] <https://sci-hub.se/10.1109/ICCSE.2009.5228509>
- [29] https://link.springer.com/chapter/10.1007/978-3-642-34062-8_32
- [30] https://scikit-learn.org/stable/modules/neural_networks_supervised.html#algorithms
- [31] <https://www.frontiersin.org/articles/10.3389/fgene.2019.01077/full>
- [32] <https://www.mdpi.com/2227-7390/8/5/765>
- [33] <https://proceedings.neurips.cc/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html>
- [34] <https://sci-hub.se/10.1109/MilCIS.2015.7348942>
- [35] <https://www.mdpi.com/1420-3049/28/3/1240>
- [36] <https://journals.sagepub.com/doi/pdf/10.1177/1536867X20909693>

- [37] <https://dl.acm.org/doi/abs/10.1145/3453892.3461323>
- [38] <https://link.springer.com/article/10.1186/s12864-019-6413-7>
- [39] <https://www.nature.com/articles/s41467-020-17419-7>
- [40] <https://link.springer.com/article/10.1186/s12911-019-1014-6>
- [41] <https://ieeexplore.ieee.org/abstract/document/9502525/>
- [42] <https://www.mdpi.com/397946>
- [43] <https://www.sciencedirect.com/science/article/abs/pii/S0927025619305026>