Phát hiện và phân loại IoT Botnet bằng máy Học các thuật toán để cải thiện an ninh mạng

Phạm Văn Quân1[0009-0003-1625-0848] , Ngô Văn Úc2[0009-0005-0954-5618], Đỗ Phúc Hào3 [0000-0003-0645-0021], và Nguyễn Năng Hùng Văn4[0000-0002-9963-7006]

[1, 2] Đại học Đông Á, Đà Nẵng, Việt Nam
quan10.work@gmail.com, ngovanuc.1508@gmail.com

3 Đại học Viễn thông Bang Bonch-Bruevich Saint-Petersburg, Saint-Petersburg, Liên bang Nga

haodp.sut@gmail.com

[4] Đại học Bách khoa Đà Nẵng, Đà Nẵng, Việt Nam
nguyenvan@dut.udn.vn

Trừu tượng. Trong thời đại công nghệ tiên tiến này, việc kết nối các thiết bị khác nhau đã trở thành một khía cạnh quan trọng. Là một giải pháp để tối ưu hóa khả năng kết nối này, Internet of Things đã nổi lên như một giải pháp nổi bật. Tuy nhiên, việc đảm bảo tính bảo mật của các hệ thống Internet of Things cũng không kém phần quan trọng. Việc phát hiện kịp thời các cuộc tấn công tiềm ẩn vào hệ thống Internet of Things có thể giảm thiểu rủi ro và giảm thiểu thiệt hại một cách hiệu quả. Trong nghiên cứu hiện tại, chúng tôi đã nghiên cứu hiệu quả của bốn thuật toán, cụ thể là Cây quyết định, K láng giềng gần nhất, Rừng ngẫu nhiên ,và Extreme Gradient Boosting, trong việc phát hiện và phân loại Internet of Things botnet. Phát hiện của chúng tôi chứng minh rằng cả bốn thuật toán đều thể hiện hiệu quả vượt trội trong việc phát hiện và phân loại các botnet. Trong số các thuật toán này, thuật toán Tăng tốc độ dốc cực độ đạt độ chính xác cao nhất, trong khi thuật toán Cây quyết định thể hiện thời gian thực hiện ngắn nhất. Nghiên cứu này nhấn mạnh tiềm năng của các thuật toán máy học trong việc phát hiện và giảm thiểu các mối đe dọa bảo mật trong các thiết bị Internet of Things. Bằng cách tận dụng các thuật toán này, có thể phát hiện và phân loại các botnet kịp thời, do đó giảm thiểu rủi ro vi phạm an ninh trong các hệ thống Internet of Things.

Từ khóa: Internet vạn vật, Học có giám sát, Phát hiện xâm nhập, IoT Botnet, An ninh mạng.

1 Giới thiệu

Khái niệm Internet of Things (IoT) ban đầu được đề xuất bởi K.Ashton (1999) như một giải pháp cho số lượng thiết bị ngày càng tăng yêu cầu kết nối internet. Cơ quan An ninh mạng và thông tin của Liên minh châu Âu định nghĩa IoT là một hệ sinh thái vật lý không gian mạng bao gồm các cảm biến và bộ truyền động được kết nối với nhau để hỗ trợ quá trình ra quyết định. Đáng chú ý, IoT là một công nghệ quan trọng trong Công nghiệp 4.0 [1]. Trong công nghiệp IoT (IIoT) đại diện cho việc triển khai IoT chuyên biệt kết nối các động cơ và các thành phần công nghiệp để nâng cao năng suất và hiệu suất của máy móc công nghiệp.

các hoạt động. IIoT đạt được mục tiêu này bằng cách cung cấp khả năng giám sát thời gian thực, quản lý hiệu quả và kiểm soát các quy trình công nghiệp, tài sản và thời gian hoạt động. Hơn nữa, IIoT nhằm mục đích giảm chi phí vận hành đồng thời nâng cao hiệu quả tổng thể [2].

Tuy nhiên, việc phát triển các hệ thống IoT đưa ra những thách thức bảo mật đáng kể. Các thiết bị IoT thường có khả năng quản lý tài nguyên và bảo mật hạn chế. Đây cũng là cơ hội để Botnet tấn công và xâm nhập. Khi Botnet xâm nhập có thể chiếm quyền điều khiển, lấy thông tin và gây thiệt hại trên nhiều thiết bị xung quanh. Do đó, bên cạnh việc phát triển hệ thống IoT, cũng cần có các biện pháp bảo mật hiệu quả. để bảo vệ chống lại các mối đe dọa mạng và vi phạm dữ liệu [3].

Ngăn chặn các cuộc tấn công và xâm nhập khác nhau và bảo vệ dữ liệu là rất cần thiết. Ngày nay, có rất nhiều hệ thống phát hiện xâm nhập (IDS) đã được phát triển để giải quyết vấn đề này. IDS có nhiệm vụ phát hiện và báo cáo xâm nhập hệ thống. Ngoài ra, IDS ngăn chặn các cuộc tấn công độc hại và duy trì hiệu suất trong các cuộc tấn công.
IDS là một thành phần quan trọng của hệ thống bảo mật hiện đại[4].

Học máy đã giới thiệu một viễn cảnh mới cho sự phát triển của IDS.
Nghiên cứu hiện tại tập trung vào việc phân loại chín phần mềm độc hại nắm bắt được bộ dữ liệu IoT-23 trong cả hai kịch bản nhị phân và đa lớp. Các mô hình được sử dụng trong nghiên cứu này bao gồm Cây quyết định (DT), K-Hàng xóm gần nhất (KNN), Rừng ngẫu nhiên (RF) và XGBoost (XGB), đã thể hiện sự thành thạo đáng kể trong các nhiệm vụ phân loại. Kết quả của phân tích này mang lại kết quả đầy hứa hẹn, do đó làm cho nó trở thành một cách tiếp cận ap phù hợp để phát hiện xâm nhập. Nghiên cứu được cấu trúc thành năm phần riêng biệt.
Phần 2 xem xét các công trình trước đây về kỹ thuật học máy và học sâu đã được sử dụng để phát hiện xâm nhập. Phần 3 trình bày chi tiết về tập dữ liệu và các mô hình được sử dụng, bao gồm các bước tiền xử lý dữ liệu và chỉ số đánh giá. Phần 4
trình bày một phân tích toàn diện về những phát hiện. Cuối cùng, phần 5 rút ra kết luận dựa trên kết quả nghiên cứu.

2 Công việc liên quan

Trong những năm gần đây, nghiên cứu về phát hiện xâm nhập IoT đã được chú ý nhiều hơn. Nghiên cứu về chủ đề này đang trở nên cần thiết hơn. Dựa trên các kết quả từ nghiên cứu trước đây, có nhiều giải pháp để phát triển hệ thống phát hiện xâm nhập IoT. Việc hiểu rõ những điểm mạnh cũng như những vấn đề của các nghiên cứu trước giúp người nghiên cứu tìm ra hướng nghiên cứu mang lại lợi ích to lớn cho việc xây dựng hệ thống IDS. Do đó, các nghiên cứu trước đây là nền tảng trong việc cung cấp kiến thức cho sự phát triển trong tương lai.

J. Hajji et al. [5] đã tiến hành một nghiên cứu để áp dụng các thuật toán học máy không giám sát, bao gồm K-means, PCA và Autoencoder, để phát hiện lưu lượng mạng bất thường. A. Rahim và cộng sự. [6] đã sử dụng phân tích thống kê để xác định các tính năng quan trọng nhất và sau đó áp dụng một số thuật toán máy học, chẳng hạn như DT, RF và Naive Bayes (NB), để phân loại lưu lượng truy cập thông thường và độc hại. SMZ Hồi giáo et al. [7] đã nghĩ ra các mô hình tính trung bình và xếp chồng dựa trên các thuật toán Support Vector Machine (SVM), RF và Gradient Boosting. Y. Li và cộng sự. [8] đã phát triển một mô hình đóng bao từ bốn mô hình máy học dựa trên DT, KNN, Logistic Regression (LR) và RF

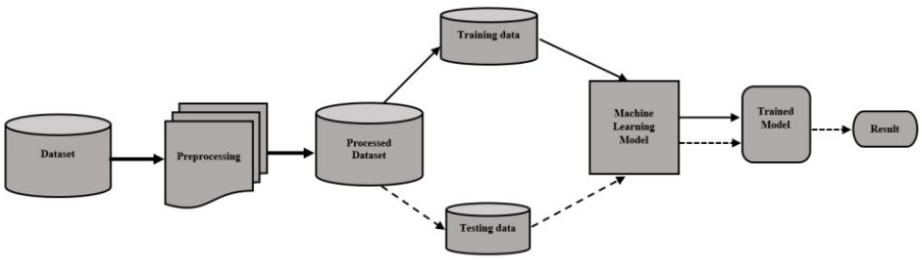
để phân loại lưu lượng mạng là bình thường hoặc độc hại. PH Do [9] đã đề xuất một phương pháp trích xuất tính năng bằng cách chia các bộ tính năng thành các lớp khác nhau, tiếp theo là ứng dụng các thuật toán học máy cho các lớp thuộc tính đó. Những nghiên cứu này đã sử dụng một loạt các thuật toán, bao gồm một số kết hợp, để đạt được độ chính xác cao trong các nhiệm vụ phân loại của chúng.

Một số nhà nghiên cứu đã khám phá tiềm năng của các mô hình học sâu để phát hiện hành vi bất thường trong lưu lượng mạng. Alotaibi et al. [10] đã phát triển mô hình dựa trên Mạng thần kinh chuyển đổi (CNN) để phân loại lưu lượng mạng là bình thường hoặc độc hại.
Lý và cộng sự. [11] đã sử dụng Bộ nhớ dài hạn ngắn hạn (LSTM) và Mạng nơ-ron dày đặc để phân loại lưu lượng mạng là bình thường hoặc độc hại. Tương tự, Abdallah et al. [12] nhân viên đã sử dụng mô hình học sâu dựa trên CNN và LSTM. Kiani et al. [13] đã đề xuất Mạng thần kinh Deep Autoencoder để tìm hiểu hành vi bình thường của lưu lượng truy cập mạng IoT và sử dụng nó để phát hiện hành vi bất thường trong thời gian thực. Ngoài ra, Rasool et al. [14] đã thử nghiệm với các mô hình học chuyển đổi như VGG16, ResNet50 và Incep tionV3. Những nghiên cứu này cho thấy tiềm năng của các mô hình học sâu trong việc phát hiện các điểm bất thường trong lưu lượng mạng.

Mỗi nghiên cứu này trình bày các phương pháp khác nhau để phát hiện và phân loại các cuộc tấn công mạng và tất cả đều cho kết quả tốt. Tuy nhiên, một số nghiên cứu chưa đề cập đến hiệu suất của mô hình cũng như thời gian thực hiện mô hình, một số ít
những người khác chỉ trình bày phát hiện hoặc phân loại.

3 phương pháp

Phần này sẽ bắt đầu với phần trình bày bộ dữ liệu sẽ được sử dụng trong phân tích tiếp theo. Các kỹ thuật tiền xử lý sau đó sẽ được áp dụng để đảm bảo tính chính xác và độ tin cậy của dữ liệu. Sau đó, chúng ta sẽ đi sâu vào quá trình lựa chọn, trực quan hóa, định dạng và chia nhỏ dữ liệu. Phân tích của chúng tôi sẽ hoàn thiện với việc đánh giá hiệu quả của các thuật toán được sử dụng, kèm theo phân tích so sánh các kết quả.



Hình 1. Phương pháp đề xuất

3.1 Bộ dữ liệu

Bộ dữ liệu IoT-23 được phát triển hoàn toàn bởi Phòng thí nghiệm Stratosphere ở Cộng hòa Séc và được xuất bản vào năm 2020. Đây là bộ dữ liệu có sẵn công khai nhằm mục đích

cung cấp cho các nhà nghiên cứu một điểm chuẩn để kiểm tra tính bảo mật và quyền riêng tư của các thiết bị Inter net of Things (IoT). Tập dữ liệu chứa các bản chụp lưu lượng mạng từ 23 loại thiết bị IoT khác nhau, chẳng hạn như phích cắm thông minh, máy ảnh, khóa thông minh và bộ điều nhiệt thông minh, v.v. Dữ liệu lưu lượng truy cập được chụp trong môi trường được kiểm soát, trong đó mỗi thiết bị được kết nối với một mạng Wi-Fi riêng biệt và chịu nhiều tình huống tấn công khác nhau, bao gồm tấn công brute-force, tấn công botnet Mirai, tấn công tiêm nhiễm, v.v. Bao gồm siêu dữ liệu cho mỗi lần chụp trong tập dữ liệu, bao gồm loại thiết bị, phiên bản chương trình cơ sở và loại tấn công. Phân phối lưu lượng được thể hiện trong Bảng 1.

Danh sách với mô tả của tất cả các tính năng được hiển thị trong Bảng 2.

Bảng 1. Nội dung lưu lượng truy cập của bộ dữ liệu IoT-23 theo các cuộc tấn công đã thực hiện.

Tên tấn công	Chảy
Part-Of-A-Horizontal-PortScan	213.852.924
Okiru	47.381.241
Okiru-Attack	13.609.479
DDoS	19.538.713
C&C-Heart Beat	33,673
C&C	21,995
Tấn công	9398
C&C-	888
C&C-Heart Beat Attack	883
C&C-Tải xuống tệp	53
C&C-Torii	30
Tập tin tải về	18
Tải xuống tệp C&C-Heart Beat	11
Tấn công Part-Of-A-Horizontal-PortScan	5
C&C-Mirai	2

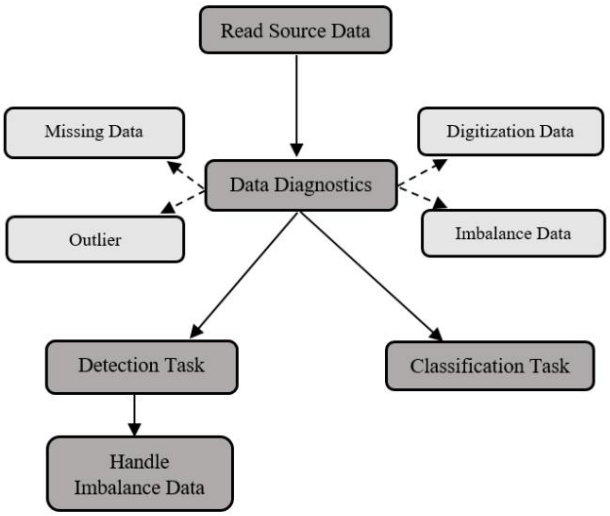
Bảng 2. Các tính năng của bộ dữ liệu IoT-23.

Tên tính năng	Sự miêu tả
trường-ts	Dòng thời gian bắt đầu
uid	ID duy nhất
id.orig-h	Nguồn Địa chỉ IP
id.orig-p	Cổng nguồn
id.resp-h	Địa chỉ IP đích
id.resp-p	cảng đích
dịch	giao thức
vụ proto	Loại dịch vụ (http, dns, v.v.)
khoảng thời gian	Tổng thời lượng lưu lượng
orig-bytes	Nguồn—byte giao dịch đích
resp-bytes	Đích—các byte giao dịch nguồn
liên kết trạng thái	trạng thái kết nối
local-orig	Nguồn địa chỉ địa phương
local-resp	Địa chỉ đích

lịch sử	gói đích
resp-pkts orig-	Dòng byte nguồn
ip-byte orig-pkts	Lịch sử của các gói nguồn
đường hầm	Thiếu byte trong khi giao dịch
byte bị bỏ lỡ-	hầm giao thông
nhãn cha mẹ	Lưu lượng byte đích
resp-ip-byte	Tên kiểu tấn công

3.2 Tiền xử lý dữ liệu

Một trong những phần quan trọng nhất của việc chuẩn bị dữ liệu cho đào tạo mô hình là tiền xử lý dữ liệu, đây được coi là quá trình tốn nhiều thời gian nhất. Tiền xử lý dữ liệu được thể hiện trong Hình 2:



Hình 2. Các bước tiền xử lý dữ liệu

Quá trình tiền xử lý dữ liệu bắt đầu bằng việc đọc nguồn dữ liệu. Tiếp theo, chúng tôi quan tâm đến chất lượng dữ liệu bằng cách kiểm tra và chẩn đoán dữ liệu. Ở bước này chúng ta sẽ giải quyết một số vấn đề nếu có như: thiếu dữ liệu, dữ liệu chứa giá trị ngoại lệ, dữ liệu phải là số và dữ liệu mất cân đối. Chúng tôi xem xét từng vấn đề và đưa ra giải pháp phù hợp, tránh ảnh hưởng đến toàn bộ dữ liệu.

Đối với các nhiệm vụ khác nhau, chúng ta phải có cách xử lý dữ liệu khác nhau. Chủ yếu ở đây là sự mất cân bằng dữ liệu tại các lớp do số lượng lớn các bản ghi khác nhau. Đối với nhiệm vụ phát hiện, sự khác biệt giữa các lớp độc hại và lành tính là quá lớn, vì vậy chúng tôi thực hiện lấy mẫu cho nhóm độc hại. Đối với tác vụ phân loại, chúng tôi sẽ hợp nhất một số lớp với một vài bản ghi lại với nhau để giảm sự mất cân bằng dữ liệu.

3.3 Phương pháp phân tích

cây quyết định

Thuật toán cây quyết định là một loại mô hình học có giám sát được sử dụng rộng rãi và phổ biến, có hiệu quả trong việc giải các bài toán phân loại và hồi quy. Cấu trúc của nó bao gồm các nút đại diện cho các biến và các nhánh đại diện cho mối quan hệ giữa các biến này. Tại nút gốc, thuật toán xem xét toàn bộ tập dữ liệu và thông qua một loạt các quyết định nhị phân dựa trên các biến đầu vào, nó phân vùng đệ quy dữ liệu thành các tập con nhỏ hơn. Các phân vùng này tạo thành các nút bên trong cây và các lá tương ứng với giá trị dự đoán của biến mục tiêu cho mỗi tập hợp con.

Để xây dựng cây quyết định, thuật toán tuân theo cách tiếp cận từ trên xuống để chọn thuộc tính tốt nhất để phân chia dữ liệu dựa trên một tiêu chí như mức tăng thông tin, chỉ số Gini hoặc entropy. Sau khi dữ liệu được phân tách, thuật toán áp dụng đệ quy quy trình tương tự cho từng tập hợp con kết quả cho đến khi đáp ứng tiêu chí dừng, chẳng hạn như độ sâu của cây được xác định trước, số lượng phiên bản tối thiểu trên mỗi lá hoặc không cải thiện thêm về hiệu suất dự đoán.

Để cập nhật và tính toán các giá trị nút trong cây quyết định, hai công thức được sử dụng: thước đo tạp chất và tiêu chí để chọn phân chia tốt nhất. Phép đo tạp chất định lượng mức độ đồng nhất hoặc không đồng nhất của biến mục tiêu trong một nút, trong khi tiêu chí để chọn phân tách tốt nhất xác định thuộc tính nào cung cấp mức tăng thông tin cao nhất hoặc tạp chất thấp nhất sau khi tách nút.

Entropy của phân phối xác suất = (1 , 2 , 3 , . ,) thỏa mãn $\sum_{=1}^{} {}_{1} = 1$

$$( ) = \sum_{=1} \times \text{nhật ký} \quad (1)$$

Mức tăng thông tin là một công thức kiểm tra nút mang lại lượng thông tin nút đó còn lại sau khi chuyển sang k nút con.

$$( ) = ( ) \left(\sum_{=1} — \times ( ) \right) \quad (2)$$

Rừng ngẫu nhiên (RF)

Thuật toán Rừng ngẫu nhiên là một kỹ thuật phân loại có giám sát nổi tiếng và được sử dụng rộng rãi, tận dụng sức mạnh của nhiều cây quyết định để cải thiện độ chính xác dự đoán của nó. Cụ thể, thuật toán tạo ra nhiều cây quyết định và tổng hợp các kết quả của chúng để có được dự đoán cuối cùng. Số lượng cây quyết định trong rừng là một tham số quan trọng ảnh hưởng đến độ chính xác của thuật toán. Điều thú vị là, khi số lượng cây quyết định tăng lên, độ chính xác của thuật toán cũng có xu hướng tăng lên, mặc dù có một điểm mà việc tăng thêm số lượng cây quyết định có ảnh hưởng không đáng kể đến độ chính xác. Một trong những ưu điểm chính của Random Forest là khả năng xử lý dữ liệu nhiều chiều, đồng thời yêu cầu ít nguồn tài nguyên tính toán hơn so với các thuật toán cạnh tranh khác. Điều này chủ yếu là do thực tế là các

thuật toán lấy mẫu một tập hợp con các tính năng tại mỗi nút của cây quyết định, điều này làm giảm tính chiều của vấn đề và dẫn đến sự hội tụ nhanh hơn.

K-Láng giềng gần nhất (KNN)

Thuật toán K-Nearest Neighbor (KNN) là một phương pháp học tập có giám sát phổ biến được sử dụng trong các lĩnh vực khai thác dữ liệu và học máy. KNN được phân loại là thuật toán "lười học", ngụ ý rằng nó không thu được kiến thức từ dữ liệu huấn luyện. Thay vào đó, tính toán chỉ được thực hiện khi cần dự đoán nhãn của dữ liệu mới. Để dự đoán nhãn của một điểm dữ liệu mới, KNN sử dụng giá trị trung bình của k nhãn gần nhất trong vùng lân cận của nó.

$$= \frac{=1}{( )} \quad (3)$$

Tăng cường độ dốc cực cao (XGBoost)

XGBoost, hay Extreme Gradient Boosting, là một thuật toán tiên tiến để giải quyết các vấn đề toán học khó hiểu với mức độ chính xác vượt trội, sánh ngang với hiệu suất của các mô hình học sâu. Nó có khả năng xử lý dữ liệu dạng bảng có kích thước và cấu trúc khác nhau, bao gồm cả dữ liệu phân loại. Phiên bản XGBoost được sử dụng trong nghiên cứu hiện tại đáng chú ý vì tốc độ đào tạo nhanh, vượt qua nhiều thuật toán khác.

3.4 Đánh giá hiệu suất

Khi đánh giá các mô hình thuật toán được mô tả trước đó, các kỹ thuật đa dạng đã được sử dụng để đánh giá độ chính xác của các kết quả và rút ra các phát hiện toàn diện cho từng mô hình. Nghiên cứu này liên quan đến một số khái niệm cơ bản, chẳng hạn như TP, TN, FP và FN. TP biểu thị số lượng dương tính thực sự đã được xác định chính xác, trong khi TN là số lượng âm tính thực sự đã được xác định chính xác. FP đại diện gửi số lượng thực tế các trường hợp dương tính đã bị phân loại nhầm là âm tính, trong khi FN biểu thị số lượng các trường hợp âm tính đã bị phân loại nhầm thành dương tính.

Độ chính xác (TRƯỚC)

Độ chính xác là thước đo hiệu suất được sử dụng để đánh giá hiệu quả của mô hình bằng cách ngăn chặn việc khai thác tỷ lệ các trường hợp tích cực được xác định chính xác. Biện pháp này có thể được tính toán bằng toán học thông qua việc sử dụng một công thức, có tính đến số lượng dương tính thật và dương tính giả. Cụ thể, độ chính xác được tính bằng cách chia số lượng dương tính thực cho tổng số dương tính thật và dương tính giả.

$$= /(+) \quad (4)$$

Độ chính xác (ACC)

Độ chính xác là thước đo hiệu suất được sử dụng để đánh giá hiệu quả của mô hình bằng cách xác định tỷ lệ dự đoán chính xác trong tổng số dự đoán. Phép đo này có thể được tính toán bằng toán học thông qua việc sử dụng một công thức, xem xét số lượng dương tính thực và âm tính thực. Cụ thể, độ chính xác được tính bằng cách chia tổng số dương thực và âm thực cho tổng số dự đoán.

$$= (TP + TN) / (TP + TN + FP + FN) \quad (5)$$

Điểm thu hồi (RE)

Điểm thu hồi là thước đo hiệu suất được sử dụng để đánh giá hiệu quả của mô hình trong việc xác định chính xác các trường hợp tích cực thực tế. Biện pháp này có thể được tính toán bằng toán học thông qua việc sử dụng một công thức, kết hợp số lượng dương tính thực và âm tính giả. Cụ thể, điểm thu hồi được tính bằng cách chia số lượng dương tính thực cho tổng số dương tính thật và âm tính giả.

$$= /(+) \quad (6)$$

Điểm F1 cho lớp nhị phân

Điểm F1 là thước đo hiệu suất thu được bằng cách tính trung bình cả điểm chính xác và điểm thu hồi. Biện pháp này được sử dụng rộng rãi để đánh giá hiệu quả tổng thể của một mô hình vì nó cung cấp một cái nhìn toàn diện về các kết quả dương tính giả và âm tính giả. Cụ thể, điểm F1 được tính là giá trị trung bình hài hòa của độ chính xác và khả năng thu hồi. Công thức tính điểm F1 tính đến cả số lượng kết quả dương tính thật, kết quả dương tính giả và âm tính giả, do đó mang lại đánh giá cân bằng hơn về hiệu suất của mô hình.

$$1 = 2 \times \frac{\times}{+} \quad (7)$$

Điểm F1 cho Đa lớp

Đối với bộ phân loại nhiều lớp, có hai tùy chọn: tính trung bình vi mô (có tính đến tần suất của từng lớp) và tính trung bình vĩ mô (tất cả các lớp được tính như nhau). Tính trung bình vĩ mô phù hợp hơn với các tập dữ liệu có kích thước tương đối đồng đều, trong khi tính trung bình vi mô phù hợp hơn với các tập dữ liệu có sự chênh lệch lớn giữa kích thước của các lớp[16]. Các công thức cho chúng là:

$$1 = \left[\sum_{=1} 1( ) \times \frac{( )}{}\right] \quad \text{(số 8)}$$

$$1 = \left[\sum_{=1} 1( ) \times \frac{1}{}\right] \quad (9)$$

4 Kết quả và thảo luận

4.1 Kết quả

Phân loại nhị phân

Khi tiến hành đào tạo và thử nghiệm mô hình, các kết quả khác nhau đã thu được. Đáng chú ý, có rất ít sự chênh lệch giữa các kết quả khác nhau. Tổng quan toàn diện về độ chính xác và thời gian đào tạo của từng mô hình được mô tả trong Bảng 3:

Bảng 3: Một số thước đo của phân loại nhị phân

đánh giá DT		RF	KNN	XGB
ACC	99,9%	89,5%	99,6%	99,8%
TRƯỚC	100,0%	88,0%	100,0%	100%
NỐT RỀ	100,0%	86,0%	99,0%	100%
F1	100,0%	87,0%	100,0%	100%
Thời gian	5,9	40.2	58.2	99,4

Trong bối cảnh phân loại nhị phân, DT nổi bật là mô hình hiệu quả nhất với thời gian xử lý ngắn nhất. Thuật toán này đặc biệt thành thạo trong việc xử lý các bộ dữ liệu có tỷ lệ lỗi cao nhờ cơ chế ra quyết định nhị phân "có" hoặc "không". Ngược lại, KNN chủ yếu dựa vào các thuộc tính dữ liệu, điều này có thể dẫn đến các lỗi nghiêm trọng khi xử lý sự khác biệt giữa các lớp dẫn đến các giá trị trung bình. Mặc dù XGB là một thuật toán mạnh mang lại kết quả đầy hứa hẹn, nhưng thời gian xử lý tương đối dài của nó khiến nó không thực tế đối với các ứng dụng trong thế giới thực.

đa phân loại

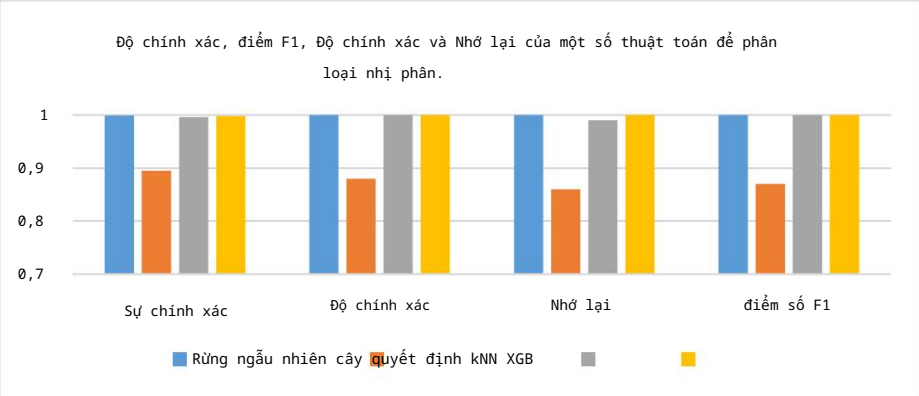
Những phát hiện thu được từ việc đưa các mô hình vào các cuộc tấn công nhiều lớp phù hợp chặt chẽ với những phát hiện được báo cáo trước đó. Cụ thể, kết quả của Random Forest tương đối kém hơn so với kết quả của các mô hình khác. Ngoài ra, thời gian đào tạo cho các mô hình được sử dụng trong các cuộc tấn công đa luồng đã được kéo dài đáng kể. Để làm rõ hơn, bảng phân tích chi tiết về kết quả và thời lượng đào tạo cho từng mô hình được trình bày trong Bảng 4:

Bảng 4: Một số thước đo của đa phân loại

Sự đánh giá	ĐT	RF	KNN	XGB
ACC	99,9%	78,2%	99,8%	99,9%
TRƯỚC	99,9%	59,6%	99,7%	100,0%
NỐT RỀ	99,7%	46,1%	98,7%	99,8%
F1	99,6%	49,8%	99,2%	99,9%
Thời gian	11,5	206,1	85,2	956.2

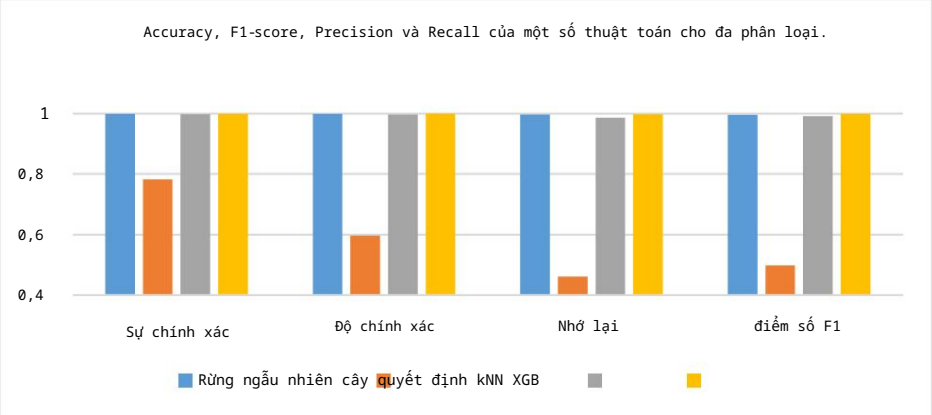
đánh giá mô hình

Để xác định sự phù hợp của một mô hình nhất định đối với một vấn đề cụ thể, các kỹ thuật đánh giá hiệu suất được sử dụng. Một loạt các phương pháp có thể được sử dụng, bao gồm nhưng không giới hạn ở các phép đo độ chính xác, độ chính xác, thu hồi và điểm F1.
Các phương pháp này đóng vai trò là chỉ số đáng tin cậy về khả năng của mô hình trong việc giải quyết vấn đề ban đầu một cách hiệu quả.



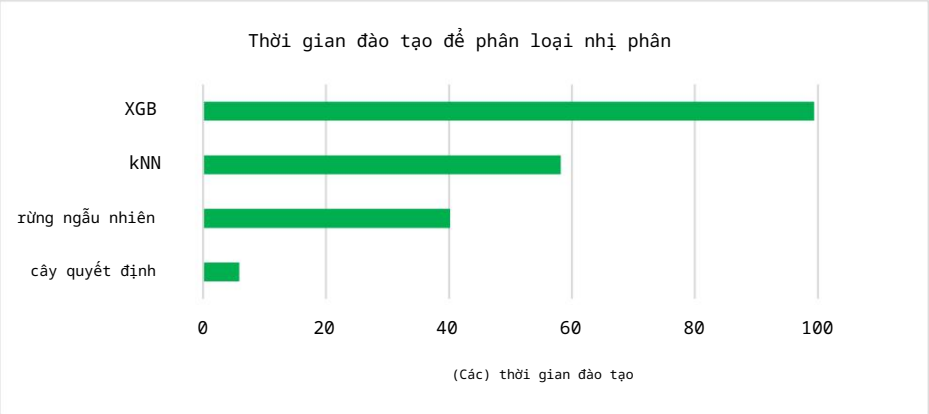
Hình 3. Accuracy, F1-score, Precision và Recall của một số thuật toán phân loại nhị phân.

Hình 3 trình bày tổng quan về độ chính xác, độ chính xác, thu hồi và số liệu điểm F1 liên quan đến thuật toán DT, RF, KNN và XGB, như được áp dụng cho phân loại nhị phân. Theo kết quả, rõ ràng là các thuật toán DT và XGB đã hoạt động rất tốt, vượt qua các đối tác của chúng. Hơn nữa, về mặt thời gian đào tạo, thuật toán Cây quyết định đã được chứng minh là hiệu quả hơn, do đó đạt được hiệu suất vượt trội về tổng thể.



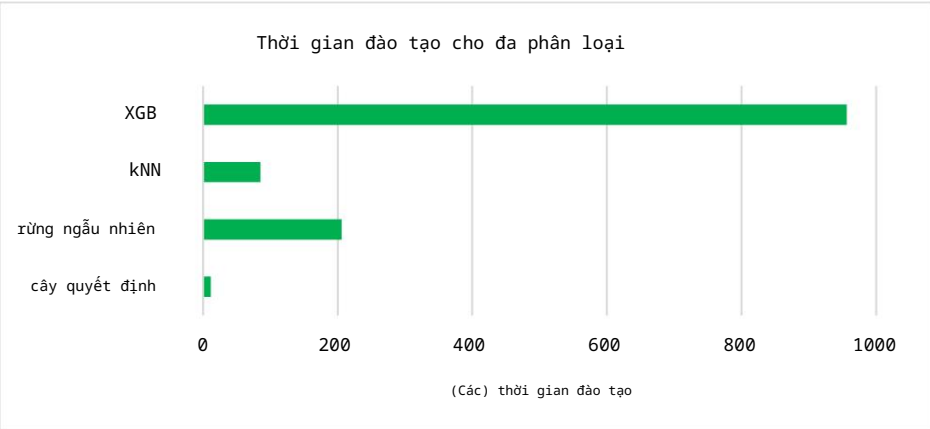
Hình 4. Accuracy, F1-score, Precision và Recall của một số thuật toán cho đa phân loại.

Hiệu suất của bốn thuật toán, cụ thể là Cây quyết định, Rừng ngẫu nhiên, KNN và XGB, được đánh giá cho vấn đề đa lớp trong hình 4. Việc đánh giá dựa trên các chỉ số về độ chính xác, độ chính xác, khả năng thu hồi và điểm F1. Kết quả chỉ ra rằng thuật toán Cây quyết định vượt trội so với các thuật toán khác về các chỉ số hiệu suất này.



Hình 5. Thời gian đào tạo để phân loại nhị phân

Thời gian đào tạo của các thuật toán cho bài toán phân loại nhị phân được trình bày trong Hình 5. Theo kết quả, thuật toán XGB mất nhiều thời gian hơn để đào tạo so với các thuật toán khác. Mặt khác, thuật toán cây quyết định đã cho thấy thời gian đào tạo tốt nhất trong số tất cả các thuật toán.



Hình 6. Thời gian đào tạo đa phân loại

Trong bài toán đa phân loại, thời gian đào tạo của các thuật toán được trình bày trong hình 6. Kết quả minh họa rằng thuật toán XGB mất nhiều thời gian hơn để đào tạo so với các thuật toán khác, với thời gian đào tạo gần gấp 100 lần so với

thuật toán cây quyết định. Mặt khác, thuật toán cây quyết định thể hiện hiệu suất tốt nhất về thời gian đào tạo.

4.2 Thảo luận

Trong bài báo này, chúng tôi đã thử nghiệm tác động của bốn thuật toán trên bộ dữ liệu IoT-23 trong nhận dạng và phân loại. Các mô hình hoạt động tốt trên các nhiệm vụ riêng biệt. Tuy nhiên, trong nhiều nhiệm vụ, các mô hình không hoạt động tốt và bị sai lệch nặng nề.

Trong tương lai, chúng tôi dự định thử nghiệm một số thuật toán máy học và mạng lưới thần kinh nhân tạo khác. Ngoài ra, chúng tôi sẽ làm việc trên một số bộ dữ liệu IoT khác và mở rộng chúng. Để triển khai thực tế các mô hình, chúng tôi dự định thử nghiệm các tính năng tốt nhất giúp cải thiện tốc độ mà không làm giảm hiệu suất quá nhiều.

5 kết luận

Sự xuất hiện của các mô hình máy học để phát hiện xâm nhập IoT là một sự phát triển tương đối gần đây, nhưng nó đã mang lại kết quả đáng kể. Trong nghiên cứu này, một số thuật toán máy học, bao gồm Cây quyết định, Rừng ngẫu nhiên, KNN và XGB, đã được áp dụng cho bộ dữ liệu Iot-23 để đánh giá hiệu quả của chúng trong việc xác định 9 lần chụp bot phần mềm độc hại khác nhau. Kết quả đã chứng minh rằng cả bốn thuật toán đều hiệu quả để phát hiện xâm nhập IoT trên bộ dữ liệu IoT-23, mặc dù hiệu suất của chúng có thể khác nhau tùy thuộc vào các tính năng cụ thể của bộ dữ liệu cũng như loại thiết bị IoT được phân tích.

Những phát hiện này hứa hẹn sẽ cải thiện các mạng bảo mật IoT thông qua việc áp dụng các thuật toán học máy. Tuy nhiên, cần có nhiều nghiên cứu hơn để khám phá hiệu suất của các thuật toán khác và để phát triển các hệ thống phát hiện xâm nhập hiệu quả và chính xác hơn nữa cho các mạng IoT.

Người giới thiệu

1. Williams, P., Dutta, IK, Daoud,. H, & Bayoumi, M.: Một cuộc khảo sát về bảo mật trong internet vạn vật tập trung vào tác động của các công nghệ mới nổi. Elsevier. (2022).
2. Khan, WZ, Rehman, MH, Zangoti, HM, Afzal, MK, Armi, N., & Salah, K.: Internet vạn vật trong công nghiệp: Những tiến bộ gần đây, hỗ trợ công nghệ và thách thức mở. Elsevier. (2020).
3. Vitorino, J., Andrade, R., Praça, I., Sousa, O., & Maia, E.: Phân tích so sánh các kỹ thuật học máy cho phát hiện xâm nhập IoT. Nền tảng và Thực hành An ninh (trang 191-207). Springer. (2022).
4. Haq, NF, Onik, AR, Hridoy, MAK, Rafni, M., Shah, FM, & Farid, DM: Ứng dụng các phương pháp học máy trong Hệ thống phát hiện xâm nhập: Khảo sát. Bài báo đăng trên Tạp chí Quốc tế về Nghiên cứu Tiên tiến về Trí tuệ Nhân tạo (IJARAI), Tập 4 Số 3. (2015).
5. Hajji, J., Khalily, M., Moustafa, N., & Nelms, T. IoT-23: Bộ dữ liệu để phân tích lưu lượng mạng IoT. lò xo. (2019).

6. Rahim, A., Razzaque, MA, Hasan, R., & Hossain, MF Bảo mật mạng IoT hiệu quả thông qua lựa chọn tính năng và kỹ thuật máy học. IEEE. (2020).
7. Islam, SMZ, Bhuiyan, MZH, & Hasan, R. Kết hợp các mô hình học máy cho Intru Phát hiện sion trong Mạng IoT bằng Bộ dữ liệu IoT-23. IEEE. (2020)
8. Li, Y., Qiu, L., Chen, Y., & Chen, Y. Hệ thống phát hiện xâm nhập dựa trên tập hợp cho IoT Mạng sử dụng Bộ dữ liệu IoT-23. IEEE. (2020)
9. PH Do, TD Dinh, DT Le, VD Pham, L. Myrova và R. Kirichek, "Một phương pháp trích xuất tính năng hiệu quả để phân loại tấn công trong mạng IoT," 2021 13th International Congress on Ultra Modern Viễn thông và Hệ thống điều khiển và Hội thảo (ICUMT)

10. Alotaibi, F., Al-Qaness, MA, Abunadi, A., & Alghazzawi, MA Một cách tiếp cận Deep Learning Ap để Phát hiện xâm nhập trong Mạng IoT bằng Bộ dữ liệu IoT-23. IEEE. (2020).
11. Li, J., Hu, C., Yang, K., Zhang, X., & Lu, J. Hệ thống phát hiện xâm nhập IoT dựa trên IoT-23 sử dụng Deep Learning. IEEE. (2020).
12. Abdallah, A., Khalil, I., Al-Emadi, N., Almohaimeed, A., & Kim, H. Botnet IoT thời gian thực Phát hiện bằng cách sử dụng Deep Learning trên Bộ dữ liệu IoT-23. IEEE. (2020)
13. Kiani, AT, Abbas, RA, Abbasi, AZ, & Khan, MK Sự bất thường dựa trên học sâu đề xuất cho Mạng IoT bằng Bộ dữ liệu IoT-23. IEEE. (2020)
14. Rasool, S., Saeed, S., Farooq, F., & Madani, A. Nghiên cứu so sánh các phương pháp học tập chuyển giao để phát hiện phần mềm độc hại IoT bằng bộ dữ liệu IoT-23. IEEE. (2021).
15. Garcia, S., Parmisano, A., & Erquiaga, MJ IoT-23: Một bộ dữ liệu được gắn nhãn với phần mềm độc hại và lưu lượng mạng IoT lành tính. IPS tầng bình lưu. (2020).
16. Stoian, NA Máy học để phát hiện bất thường trong mạng IoT: Phân tích phần mềm độc hại trên bộ dữ liệu IoT-23. EEMCS: Kỹ thuật Điện, Toán học và Khoa học Máy tính. (2020)