HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

GIÁP THỊ NGỌC BÍCH

NGHIÊN CỨU HỆ THỐNG PHÁT HIỆN TẤN CÔNG IDS TRONG MÔI TRƯỜNG IOT DỰA TRÊN HỌC MÁY

CHUYÊN NGÀNH: KỸ THUẬT VIỄN THÔNG

MÃ SỐ: 8.52.02.08

TÓM TẮT ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

HÀ NỘI - 2025

Luận văn được hoàn thành tại:

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Người Hướng Dẫn Khoa Học: **PGS.TS. HOÀNG TRỌNG MINH**

Phản biện 1: ……

Phản biện 2: ……..

Luận văn sẽ được bảo vệ trước Hội đồng chấm luận văn thạc sĩ tại Học viện Công nghệ Bưu chính Viễn thông

Vào lúc: ..... giờ ..... ngày ..... tháng .... năm .....

Có thể tìm hiểu luận văn tại:

- Thư viện của Học viện Công nghệ Bưu chính Viễn thông.

CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG IDS TRONG IOT

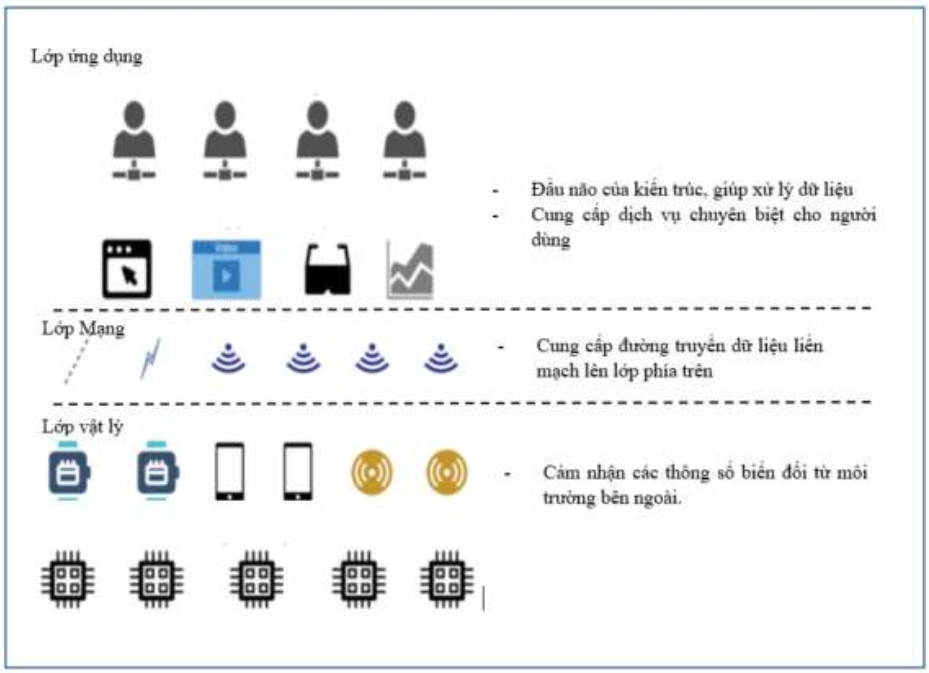
1.1. Tổng quan về IoT

1.1.Lý thuyết về IoT (Internet Vạn Vật)

1.1.1.Khái niệm và sự phát triển của IoT

1.1.2. Kiến trúc IoT

Kiến trúc cơ bản nhất trong IoT là kiến trúc ba lớp được đề cập trong nhiều bài nghiên cứu. Đây là kiến trúc được giới thiệu từ giai đoạn đầu của của các nghiên cứu về IoT. Kiến trúc được cấu thành từ ba lớp: lớp cảm biến, lớp mạng và lớp ứng dụng.



Hình 1. 1 Kiến trúc hệ thống IoT 3 lớp

Kiến trúc ba lớp đã thể hiện được ý tưởng của IoT. Tuy nhiên nó chưa đầy đủ để phục cho những nghiên cứu chuyên sâu. Để giải quyết tình trạng này, những cải tiến kỹ thuật đã được đề xuất thông qua các bài nghiên cứu kiến trúc IoT. Dựa trên kiến trúc ba lớp phổ biến, [7] đã đề xuất một kiến trúc IoT mới, cung cấp điện toán biên cơ bản, được gọi là kiến trúc ba lớp mở rộng. “Kiến trúc tuân thủ theo mô hình ba lớp bao gồm : lớp thiết bị đầu cuối IoT, lớp điện toán biên và lớp đám mây.

1.1.3 Đặc trưng dữ liệu IoT

Dữ liệu IoT mang đặc trưng của dữ liệu lớn, bao gồm: Khối lượng dữ liệu cực lớn, Tốc độ sinh dữ liệu theo thời gian thực, Tính đa dạng (từ nhiều thiết bị khác nhau),

Tính không chắc chắn và nhiễu, Giá trị khai thác cao, và Tính biến thiên. Dữ liệu này là nền tảng cốt lõi giúp IDS xây dựng mô hình hành vi mạng và phát hiện kịp thời các dấu hiệu bất thường..

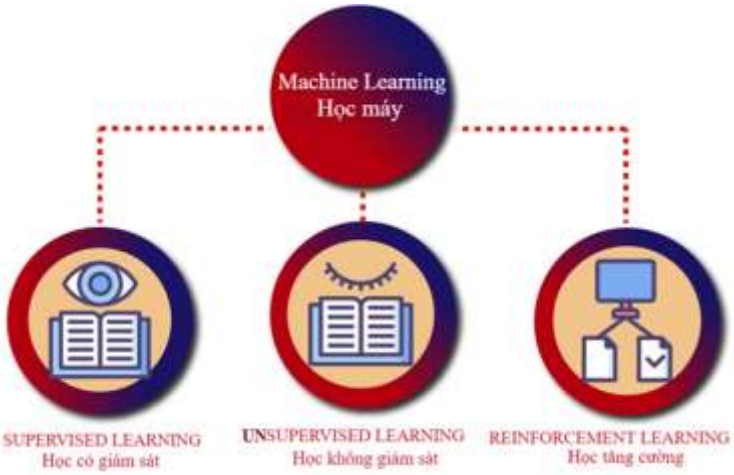
1.1.4 Các công nghệ điển hình trong IoT

a, Điện toán biên

Mục tiêu cốt lõi của điện toán biên là tối ưu hóa hoạt động của mạng IoT bằng cách hạn chế nhu cầu truyền dữ liệu lên đám mây và giảm độ trễ trong quá trình xử lý. Học máy đóng vai trò trung tâm trong mô hình IoT ba lớp mở rộng nhờ khả năng triển khai trực tiếp trên các thiết bị biên, từ đó hỗ trợ hiệu quả các ứng dụng AI yêu cầu phản hồi thời gian thực và độ trễ cực thấp. Điện toán biên bao gồm ba kiến trúc nền tảng : điện toán sương mù (fog computing), điện toán biên di động (MEC) và đám mây nhỏ (micro cloud).

b, Học máy

Các nghiên cứu cho thấy những kỹ thuật học máy khác nhau mang lại hiệu quả riêng khi được triển khai trong môi trường IoT đa dạng về cấu trúc dữ liệu và điều kiện vận hành. Một số nhóm phương pháp tiêu biểu có phạm vi ứng dụng rộng như học có giám sát, học không giám sát, học tăng cường.



Hình 1. 2 Phân loại mô hình học máy

Học máy có giám sát được biết đến là một trong những kỹ thuật ML phổ biến nhất trong ứng dụng IoT, các mô hình được đào tạo trên tập dữ liệu được gắn nhãn để phân loại hoặc dự đoán.

Khác với học có giám sát, các kỹ thuật học không giám sát không chịu phụ thuộc vào dữ liệu được gán nhãn mà dựa vào khả năng tự phát hiện cấu trúc và mẫu ẩn trong tập dữ liệu

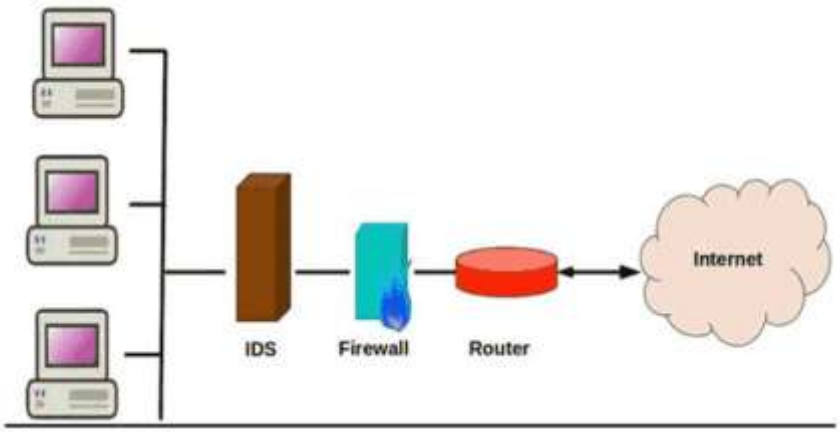
Bên cạnh đó, học tăng cường nổi lên như một hướng tiếp cận hữu hiệu cho các tình huống đòi hỏi tối ưu hóa hành động theo thời gian thực. Các thuật toán như Q-learning hoặc Deep Q-Network cho phép mô hình tự học thông qua quá trình tương tác với môi trường, từ đó dần tìm được chiến lược tối ưu.

1.2.Tổng quan hệ thống IDS

1.2.1. Lý thuyết về Hệ thống Phát hiện Xâm nhập (IDS)

a, Hệ thống phát hiện xâm nhập IDS

Hệ thống phát hiện xâm nhập được phát triển như một lớp phòng thủ chuyên dụng nhằm nhận diện các hành vi độc hại bên trong hoặc từ bên ngoài mạng máy tính. Khác với tường lửa vốn chủ yếu kiểm soát truy cập dựa trên những quy tắc cố định, IDS thực hiện quá trình phân tích sâu vào hành vi mạng và hoạt động của người dùng, từ đó phát hiện những bất thường mà các cơ chế bảo mật truyền thống khó nhận biết. IDS có thể tồn tại dưới dạng phần cứng, phần mềm hoặc kết hợp cả hai, vận hành như một "hệ thần kinh" giúp hệ thống kịp thời nắm bắt các dấu hiệu xâm nhập từ đó đưa ra cảnh báo hoặc kích hoạt phản ứng bảo vệ.



Hình 1. 3. Thành phần hệ thống IDS

b, Phân loại hệ thống IDS

- Hệ thống phát hiện xâm nhập dựa trên chữ ký (SIDS)
- Hệ thống phát hiện xâm nhập dựa trên bất thường (AIDS)

1.2.2. Hệ thống IDS dựa trên học máy tại biên

Hệ thống phát hiện xâm nhập (IDS) triển khai tại biên mạng ngày càng đóng vai trò quan trọng trong việc bảo vệ các hệ thống IoT trước sự gia tăng về số lượng và mức độ tinh vi của các cuộc tấn công mạng. Không giống các mô hình IDS truyền thống phụ thuộc vào xử lý tập trung trên đám mây, IDS tại biên tận dụng khả năng tính toán cục bộ của các thiết bị biên hoặc gateway để phát hiện tấn công thời gian thực, từ đó giảm đáng kể độ trễ xử lý, tiết kiệm băng thông truyền dữ liệu và tăng cường khả năng phản hồi nhanh trong môi trường phân tán. Song song với đó, tích hợp học máy vào biên cho phép triển khai và thực thi các mô hình học máy ngay tại nơi dữ liệu được tạo ra, như cảm biến IoT, điện thoại thông minh, các thiết bị nhúng.

1.3. Thách thức của các mô hình IDS trong môi trường IoT

- Hạn chế về tài nguyên tính toán và lưu trữ
- Yêu cầu xử lý dữ liệu thời gian thực

1.4. Kết Luận chương

Chương 1 đã cung cấp cái nhìn tổng quan về hệ sinh thái IoT, Hệ thống phát hiện xâm nhập IDS, và sự kết hợp của IDS, học máy tại biên. Qua đó có thể thấy được tính thiết thực của hệ thống IDS khi triển khai tại biên mạng. Tuy nhiên bên cạnh những lợi ích đã được đề cập như tăng tốc độ xử lý, tối ưu tài nguyên và bảo mật dữ liệu, hệ thống cũng gặp nhiều thách thức khi biên mạng bị hạn chế về tài nguyên rất lớn. Thiết bị biên gần như không thể đáp ứng các mô hình học máy phức điều này gây cản trở lớn trong việc phát triển hệ thống. Việc tích hợp các phương pháp xử lý để tối ưu hóa mô hình được nhận định là phương pháp hiệu quả để xử lý bài toán hiện tại. Các kỹ thuật giảm chiều hay phân cụm dữ liệu không chỉ giảm thiểu tài nguyên mà còn giải quyết rất tốt bài toán tối ưu, đáp ứng tốt nhu cầu của hệ thống IoT hiện tại. Chương 2 sẽ tập trung thiết kế và đề xuất "Hệ thống phát hiện tấn công trong IoT ứng dụng học máy.

CHƯƠNG 2 - HỆ THỐNG PHÁT HIỆN TẤN CÔNG TRONG IOT DỰA TRÊN HỌC MÁY

2.1. Hệ thống IDS dựa trên học máy cho thiết bị IoT tại biên

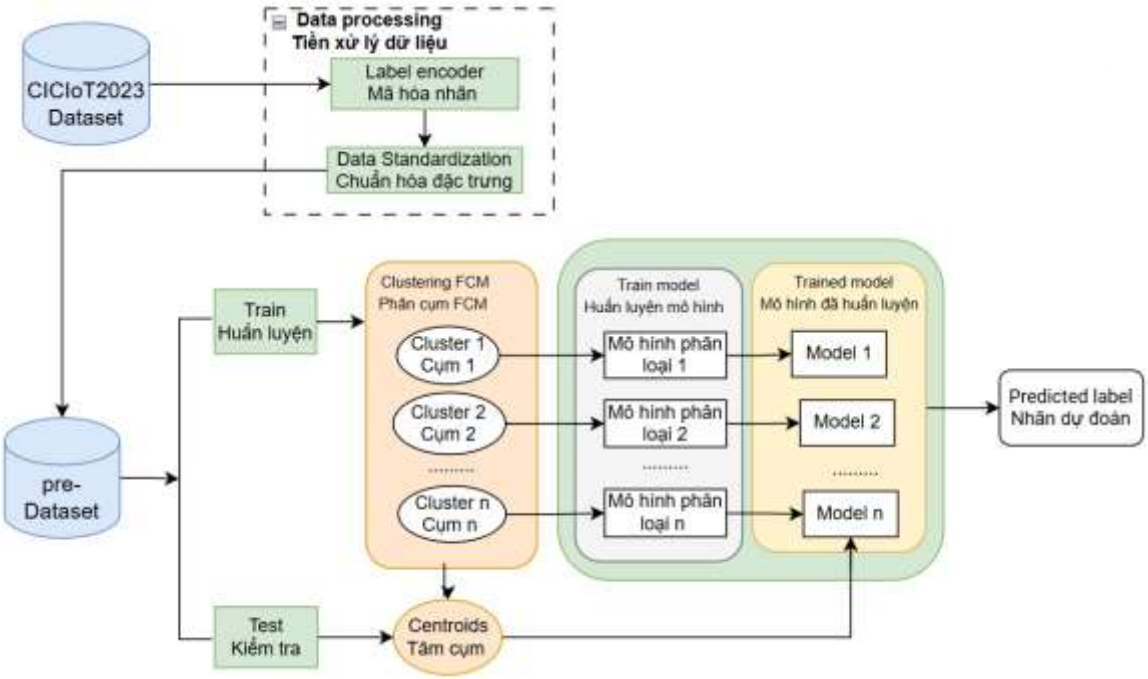
2.1.1. Cơ sở khoa học của giải pháp đề xuất

a, Cơ sở khoa học các phương pháp xử lý dữ liệu

b, Cơ sở khoa học về học máy trong hệ thống IDS

2.1.2. Hệ thống IDS ứng dụng học máy đề xuất

Trên cơ sở khoa học được trình bày ở các mục 2.1.1, đề án đề xuất hệ thống phát hiện xâm nhập (IDS) hướng tới mục tiêu tối ưu hóa hiệu quả xử lý, đồng thời đáp ứng yêu cầu triển khai trên các thiết bị biên với tài nguyên hạn chế. Kiến trúc của hệ thống được thiết kế theo dạng phân tầng, bao gồm ba khối chức năng chính: tiền xử lý dữ liệu, phân cụm dữ liệu và đánh giá hiệu năng. Hình 2.1 minh họa cấu trúc tổng thể của hệ thống. Cấu trúc phân tầng này đảm bảo mức độ mô-đun hóa cao, cho phép tối ưu hóa từng thành phần một cách độc lập.



Hình 2. 1 Mô hình hệ thống IDS đề xuất

Quá trình tiền xử lý bao gồm hai bước chính: (1) mã hóa các thuộc tính dạng phân loại bằng kỹ thuật Label Encoding để chuyển đổi chúng sang dạng số nguyên, và

(2) chuẩn hóa các đặc trưng số bằng StandardScaler nhằm đưa giá trị về cùng một miền, giúp tăng tính ổn định cho quá trình huấn luyện.

Trong khối phân cụm dữ liệu, đề án đề xuất thuật toán phân cụm mờ Fuzzy C-Means (FCM). Việc sử dụng FCM cho phép xác định mức độ phụ thuộc của từng mẫu vào các cụm khác nhau, từ đó tăng khả năng mô tả và phân biệt các dạng hành vi trong lưu lượng mạng.

Sau quá trình phân cụm, hệ thống tiến hành kiểm tra số lượng mẫu trong từng cụm. Đối với những cụm không đạt ngưỡng dữ liệu tối thiểu cần thiết cho việc huấn luyện, hệ thống sẽ tiến hành ghép cụm với cụm gần nhất và tính toán lại tâm cụm nhằm bảo đảm sự ổn định của quá trình phân tích. Khi dữ liệu đã được tổ chức phù hợp, hệ thống đưa dữ liệu vào đề huấn luyện các mô hình học máy bao gồm Random Forest và Decision Tree cho từng cụm dữ liệu.

2.2. Cơ sở lý thuyết của phương pháp phân cụm Fuzzy C-Mean

Phân cụm là kỹ thuật học máy không giám sát. Mục tiêu chính của phân cụm C-mean là phân vừng dữ liệu thành các cụm tập hợp, mỗi điểm dữ liệu được gán giá trị thành viên. FCM thường được sử dụng trong nhận dạng mẫu và phân đoạn ảnh. FCM là là phần mở rộng của của thuật toán phân cụm K-mean hay còn gọi là C-mean cứng, nhưng với việc gán các điểm dữ liệu vào các cụm được làm mềm. Trong thuật toán này, mỗi điểm dữ liệu thuộc về một cụm và quyền thành viên được biểu diễn thông qua một giá trị gọi là mức độ thành viên. Giá trị này nằm trong khoảng từ 0 đến 1. Điều này tăng tính linh hoạt cho vệc gán cụm.

2.2.1. Cơ sở toán học

FCM phân cụm bằng cách tối ưu hóa hàm mục tiêu để tạo ra miền dữ liệu rõ ràng và dày đặc. Hàm mục tiêu cho bài toán tối ưu hóa được mô tả bằng phương trình sau

$$J_{f_{cm}}(U,V) = \sum_{i=1}^{c}\sum_{j=1}^{N}(u_{ij})^m D_{ij}^2, m>1 \qquad (2.1)$$

Trong đó :

N: số lượng đối tượng, điểm dữ liệu

c : số lượng phân cụm

u_{ij} : độ phụ thuộc của điểm dữ liệu thứ x_k đối với cụm v_i

D_{ij}^2 : khoảng cách từ dữ liệu x_k đến tâm cụm v_i

m : hệ số mờ

Ràng buộc chính của hàm mục tiêu là tổng bậc thành viên của một điểm dữ liệu trên các cụm bằng 1:

$$\sum_{i=1}^{c} u_{ij} = 1 \qquad (2.2)$$

Khoảng cách Cosine dựa trên độ tương đồng vector, giúp đo lường sự khác biệt dựa trên hướng thay đổi vì độ lớn. Khoảng cách này được định nghĩa dựa trên cosine của góc 2 vector x_j và v_i :

$$D_{ij} = 1 - \frac{x_j . v_i}{\| x_j \| . \| v_i \|} \qquad (2.3)$$

Khi xác định độ phụ thuộc của điểm dữ liệu, FCM khai thác thông tin giữa các cụm, điều này đảm bảo phân bố mềm của các điểm vào cụm.

$$u_{ij} = \left[\sum_{k=1}^{c} \left(\frac{D_{ij}^2}{D_{kj}^2} \right)^{\frac{1}{m-1}} \right]^{-1} \qquad (2.4)$$

Điều kiện hội tụ.

$$v_i = \frac{\sum_{j=1}^{n} u_{ij}^m x_j}{\sum_{j=1}^{n} u_{ij}^m} \qquad (2.5)$$

Algorithm 1: FCM Algorithm

Input: $X, c, m, \varepsilon, l_{\max}$

$U = random([c, n])$

for $l = 1$ **to** $l_{\max}$ **do**

  $v_i = \frac{\sum_{j=1}^{n} u_{ij}^m x_j}{\sum_{j=1}^{n} u_{ij}^m}$

  $u_{ij} = \left[\sum_{k=1}^{c} \left(\frac{D_{ij}^2}{D_{kj}^2} \right)^{\frac{1}{m-1}} \right]^{-1}$

  if $\|V^{l-1} - V^l\| \leq \varepsilon$ **then**

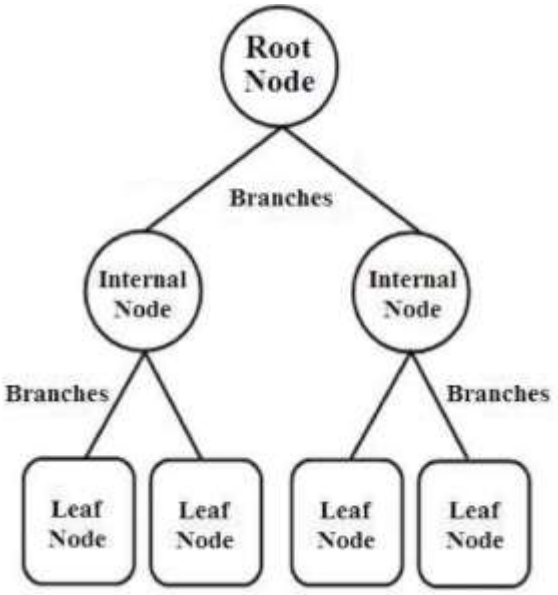
    Stop

Output: U, V

2.3. Cơ sở lý thuyết của mô hình học máy

2.3.1. Mô hình Decision Tree

Cây quyết định được biết đến với khả năng diễn giải hiệu quả. Ý tưởng chính của thuật toán là phân vùng dữ liệu bằng phương pháp đệ quy thành các tập con dựa trên giá trị của các thuộc tính khác nhau cho đến khi đáp ứng hàm mục tiêu. Quá trinh tạo ra cấu trúc dạng cây, mỗi nút đại diện cho một quyết định hoặc phép chia dựa trên thuộc tính cụ thể. Thuật toán chọn ra thuộc tính tốt nhất để sử dụng cho mỗi phép chia, như tăng mức độ thông tin, tăng tỷ lệ và chỉ số Gini. Cây quyết định có hiệu quả với trong việc xử lý dữ liệu phân loại và số.



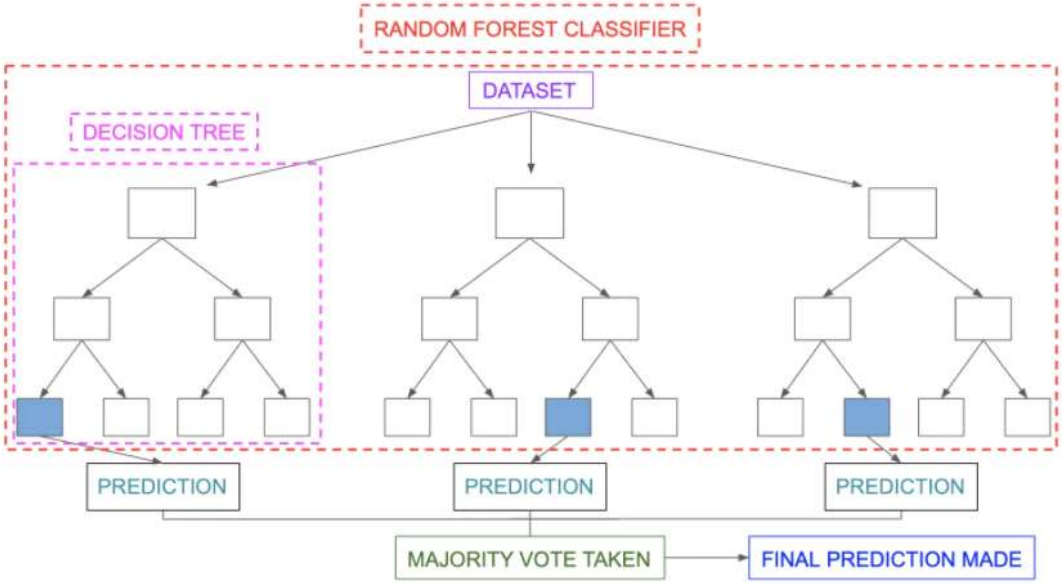
Hình 2. 2 Cấu trúc mô hình Decision Tree

a, Quy tắc phân tách

2.3.2. Mô hình Random forest

Rừng ngẫu nhiên là mô hình học máy có giám sát, xây dựng trên cơ sở các cây quyết định để hình thành mô hình học máy mạnh mẽ hơn. Mô hình lấy dự đoán của nhiều cây quyết định, dựa trên dự đoán của các cây quyết định này để đưa ra dự đoán cuối cùng.

Quy trình xây dựng mô hình được mô tả trong hình 2.5



Hình 2. 3 Cấu trúc mô hình Random Forest

(1) Khởi tạo tập cây thông qua lấy mẫu bootstrap

(2) Lựa chọn ngẫu nhiên tập đặc trưng tại mỗi điểm phân chia

(3) Xây dựng và huấn luyện các cây quyết định độc lập

(4) Tổng hợp dự đoán của mô hình

(5) Đánh giá và tinh chỉnh mô hình

2.4. Tập dữ liệu CICIoT

2.4.1. Nguồn gốc

a, Nguồn gốc dữ liệu

b, Đặc trưng dữ liệu

2.2.2. Cấu Trúc Tập Dữ Liệu

a, Số lượng mẫu

Tập dữ liệu CiCIoT2023 bao gồm các dữ liệu lành tính và lưu lượng mạng độc hại. Tập dữ liệu có tổng 46,686,579 lưu lượng, trong đó có 1,098,195 lưu lượng lành tính và 45,588,384 lưu lượng độc hại. Số lượng mẫu theo từng nhóm và chi tiết từng loại tấn công được trình bày chi tiết tại bảng 2.1:

Bảng 2. 1 Phân bổ tấn công trong tập dữ liệu CICIOT2023

Loại tấn công	Số mẫu	Loại tấn công	Số mẫu	Nhóm tấn công	Số mẫu
DDoS-ICMP_Flood	7,200,504	DoS-TCP_Flood	2,671,445	DDoS	33,984,560
DDoS-UDP_Flood	5,412,287	DoS-SYN_Flood	2,028,834	DoS	8,090,738
DDoS-TCP_Flood	4,497,667	BenignTraffic	1,098,195	Mirai	2,634,124
DDoS-PSHACK_Flood	4,094,755	Mirai-greeth_flood	991,866	Benign	1,098,195
DDoS-SYN_Flood	4,059,190	Mirai-udpplain	890,576	Spoofing	486,504
DDoS-RSTFINFlood	4,045,285	Mirai-greip_flood	751,682	Recon	354,565
DDoS-SynonymousIP_Flood	3,598,138	DDoS-ICMP_Fragmentation	452,489	Web	24,829
DoS-UDP_Flood	3,318,595	MITM-ArpSpoofing	307,593	BruteForce	13,064
Recon-PingSweep	2262	Uploading_Attack	1252	Total	46,686,579
DDoS-UDP_Fragmentation	286,925	DDoS-HTTP_Flood	28,790		
DDoS-ACK_Fragmentation	285,104	DDoS-SlowLoris	23,426		
DNS_Spoofing	178,911	DictionaryBruteForce	13,064		
Recon-HostDiscovery	134,378	BrowserHijacking	5859		
Recon-OSScan	98,259	CommandInjection	5409		
Recon-PortScan	82,284	SqlInjection	5245		
DoS-HTTP_Flood	71,864	XSS	3846		
VulnerabilityScan	37,382	Backdoor_Malware	3218		

b, Các đặc trưng

2.2.3. Các Loại tấn công trong tập dữ liệu

Bảng 2. 2 Các loại tấn công trong tập dữ liệu CICIOT2023

Lớp tấn công	Kịch bản tấn công	Cách thức tấn công
DoS	DoS-UDP_Flood	Kẻ tấn công gửi ồ ạt các gói UDP đến mục tiêu, làm cạn tài nguyên và ngăn truy cập hợp pháp.
	DoS-SYN_Flood	Gửi nhiều gói SYN để tạo kết nối nửa chừng trong TCP, làm đầy bảng kết nối của máy chủ.
	DoS-TCP_Flood	Gửi nhiều gói TCP ngẫu nhiên để làm nghẽn băng thông và giảm hiệu suất dịch vụ.
	DoS-HTTP_Flood	Tấn công ở tầng ứng dụng bằng cách gửi nhiều yêu cầu HTTP hợp lệ nhằm làm quá tải máy chủ web.
DDoS	DDoS-HTTP_Flood	Giống HTTP_Flood nhưng thực hiện từ nhiều nguồn zombie/botnet, mạnh hơn, khó đối phó hơn.
	DDoS-SlowLoris	Gửi yêu cầu HTTP rất chậm, giữ các kết nối mở để chiếm tài nguyên webserver.

	DDoS-ICMP_Fragmentation	Gửi nhiều gói ICMP bị phân mảnh khiến mục tiêu tốn tài nguyên xử lý và tái lắp ghép.
	DDoS-UDP_Fragmentation	Tương tự nhưng với giao thức UDP, làm hệ thống overload khi phải xử lý phân mảnh gói tin.
	DDoS-ACK_Fragmentation	Flood gói ACK bị phân mảnh làm nghẽn xử lý ở tầng TCP của mục tiêu.
	DDoS-SynonymousIP_Flood	Nhiều gói đến từ các địa chỉ IP đồng nghĩa/giả mạo khiến hệ thống khó lọc và quá tải.
	DDoS-ICMP_Flood	Flood ICMP (ping) khiến máy chủ phải liên tục phản hồi, làm suy giảm dịch vụ.
	DDoS-TCP_Flood	Flood các gói TCP từ nhiều nguồn, kết hợp phân tán gây ảnh hưởng lớn hơn DoS.
	DDoS-UDP_Flood	UDP Flood từ nhiều nút botnet, làm nghẽn mạng nhanh và mạnh.
	DDoS-SYN_Flood	SYN Flood phân tán, cực kỳ nguy hiểm cho hệ thống TCP handshake.
	DDoS-PSHACK_Flood	Gửi nhiều gói TCP PSH+ACK nhằm làm cạn tài nguyên xử lý dữ liệu đẩy/push.
	DDoS-RSTFINFlood	Flood gói RST/FIN để liên tục đóng kết nối, gây rối xử lý giao tiếp của mục tiêu.
Web-based	CommandInjection	Chèn lệnh hệ thống vào tham số đầu vào để chiếm quyền điều khiển server.
	SqlInjection	Chèn lệnh SQL độc hại nhằm đánh cắp hoặc thay đổi dữ liệu trong DB.
	Uploading_Attack	Tải lên file độc hại (webshell, mã độc) để chiếm quyền hoặc phá hoại hệ thống.
	XSS	Chèn script vào trang web để đánh cắp cookie, chiếm phiên hoặc điều khiển trình duyệt.
	Backdoor_Malware	Mã độc cài đặt cửa hậu trên thiết bị để truy cập và điều khiển trái phép.
	BrowserHijacking	Chiếm quyền điều hướng trình duyệt, chuyển hướng đến trang độc hại hoặc thu thập dữ liệu.
Recon	Recon-HostDiscovery	Quét tìm thiết bị đang hoạt động trên mạng để chuẩn bị tấn công.
	VulnerabilityScan	Quét lỗ hổng hệ thống nhằm tìm điểm yếu khai thác tiếp theo.
	Recon-PortScan	Quét cổng dịch vụ để xác định dịch vụ đang chạy và chuẩn bị khai thác.
	Recon-OSScan	Dò hệ điều hành mục tiêu bằng phân tích gói tin để lựa chọn kỹ thuật tấn công phù hợp.
	Recon-PingSweep	Gửi nhiều ping để xác định dải địa chỉ IP sống trong mạng.
Spoofing	MITM-ArpSpoofing	Giả mạo ARP để chen vào luồng dữ liệu, đánh cắp hoặc chỉnh sửa thông tin truyền.
	DNS_Spoofing	Giả mạo phản hồi DNS để chuyển hướng nạn nhân đến website độc hại.
Brute Force	DictionaryBruteForce	Dò mật khẩu bằng tập từ điển để chiếm quyền truy cập tài khoản dịch vụ.

Mirai	Mirai-greip_flood	Một biến thể của botnet Mirai flood gói GRE (Generic Routing Encapsulation) gây tắc nghẽn mạng.
	Mirai-udpplain	Flood các gói UDP đơn giản từ thiết bị IoT bị nhiễm Mirai.
	Mirai-greeth_flood	Flood gói GRE Ethernet — biến thể Mirai nguy hiểm ảnh hưởng lớn đến thiết bị mạng.

2.2.4. Tập Dữ Liệu CICIoT2023 trong Nghiên Cứu và Đề Án

2.2.5. Tiền xử lý dữ liệu

Mã hóa đặc trưng

Các mô hình học máy chỉ nhận và xử lý dữ liệu số. Vì vậy trước khi đưa dữ liệu vào huấn luyên, mã hóa dữ liệu là một bước quan trọng trong quá trình tiền xử lý dữ liệu. Mã hóa Label Encoder được sử dụng trong bước này. Phương pháp này ánh xạ dữ liệu phân loại từ dạng string sang dạng số nguyên.

Chuẩn hóa dữ liệu

Phương pháp Standard Scaler , nhằm đưa tất cả các biến về cùng một phạm vi đo lường. “Chuẩn hóa là quá trình chuyển đổi giá trị của từng đặc trưng sao cho có trung bình bằng 0 và phương sai bằng 1”, được biểu diễn theo công thức:

$$z = \frac{x - \mu}{\sigma} \qquad (2.6)$$

“μ là giá trị trung bình và σ là độ lệch chuẩn của đặc trưng”. Việc chuẩn hóa có vai trò quyết định trong học máy, bởi hầu hết các thuật toán phân loại và hồi quy hoạt động hiệu quả hơn và hội tụ nhanh hơn khi các đặc trưng đầu vào có cùng phạm vi giá trị. Nếu một đặc trưng có biên độ lớn hơn đáng kể so với các đặc trưng khác, nó có thể chi phối quá trình học và dẫn đến mô hình thiên lệch, làm giảm độ chính xác. Chuẩn hóa đảm bảo rằng mỗi đặc trưng đóng góp công bằng vào quá trình học, từ đó nâng cao khả năng mô hình nhận diện các mẫu và mối quan hệ tiềm ẩn một cách hiệu quả hơn.

2.5. Kết luận chương

Chương 2 đã trình bày chi tiết cơ sở lý thuyết và kiến trúc tổng thể của hệ thống phát hiện xâm nhập (IDS) dựa trên học máy trong môi trường IoT. Hệ thống được xây

dựng với mục tiêu nâng cao hiệu quả phát hiện tấn công, đồng thời đảm bảo khả năng triển khai trên các thiết bị biên với tài nguyên hạn chế. Chương 2 đã trình bày nền tảng toán học của phương pháp phân cụm FCM, bao gồm hàm mục tiêu, cơ chế cập nhật trọng số và cách thức đo khoảng cách giữa các điểm dữ liệu. Những cơ sở này đóng vai trò quan trọng trong quy trình hỗ trợ mô hình học máy cải thiện hiệu năng phân loại.

Chương 2 đã làm rõ mối liên hệ giữa các thành phần trong hệ thống IDS và cung cấp đầy đủ nền tảng lý thuyết cần thiết cho quá trình triển khai, huấn luyện và đánh giá thực nghiệm trong Chương 3. Đây là cơ sở quan trọng để kiểm chứng hiệu quả và tính khả thi của mô hình đề xuất trên tập dữ liệu CICIoT2023

CHƯƠNG 3 - THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ CỦA MÔ HÌNH IDS CHO MÔI TRƯỜNG IOT

3.1. Môi trường thực nghiệm

3.1.1. Cấu hình của mô hình huấn luyện

3.1.2. Kịch bản thử nghiệm và đánh giá

Kịch bản thử nghiệm :

Tiền xử lý dữ liệu

Bộ dữ liệu CICIoT2023 được làm sạch bằng cách loại bỏ các giá trị trống và trùng lặp, chuẩn hóa các đặc trưng số bằng StandardScaler, đồng thời mã hóa các đặc trưng phân loại bằng Label Encoding nhằm đảm bảo dữ liệu đầu vào phù hợp cho các mô hình học máy.

Phân cụm dữ liệu bằng FCM

Thuật toán phân cụm Fuzzy C-Means (FCM) sử dụng khoảng cách Cosine được áp dụng để khai thác cấu trúc tiềm ẩn trong dữ liệu. Số lượng cụm K được khảo sát trong khoảng từ 2 đến 9 để xác định cấu hình mang lại hiệu quả tối ưu.

Huấn luyện mô hình học máy

Giá trị độ phụ thuộc (membership degree) thu được từ FCM được kết hợp như một tập đặc trưng bổ sung, sau đó sử dụng để huấn luyện các mô hình học máy gồm Random Forest và Decision Tree, từ đó tăng cường khả năng phân biệt giữa lưu lượng hợp lệ và lưu lượng tấn công.

Đánh giá hiệu suất mô hình

Các mô hình được đánh giá dựa trên những chỉ số quan trọng trong hệ thống IDS: Accuracy, Precision, Recall và F1-score (chỉ số trung hòa giữa Precision và Recall) nhằm phản ánh toàn diện khả năng phát hiện tấn công của mô hình.

.3.2. Kết quả đạt được

3.2.1. Các tiêu chí đánh giá hiệu năng

a, Độ chính xác tổng thể (Accuracy)

b, Độ phủ (Recall)

c, Độ chính xác (Precision)

d, F1-score

3.2.2. Đánh giá mô hình học máy khi không dùng phân cụm FCM

Phần này đánh giá hiệu suất khi chỉ sử dụng mô hình học máy lên tập dữ liệu CICIoT2023. Đề án sử dụng hai mô hình học máy Random forest, Decision Tree.

Bảng 3. 1 Độ chính xác của mô hình học máy trên tập dữ liệu CICIOT2023

Mô hình	RandomForest	Decision Tree
Accuracy	0.9877	0.9523
Precision_Macro	0.7771	0.7072
Recall_Macro	0.7253	0.477
F1_Macro	0.7398	0.4878
Precision_Weighted	0.9865	0.9544
Recall_Weighted	0.9877	0.9523
F1_Weighted	0.9867	0.9459

3.2.3. Đánh giá mô hình học máy khi sử dụng phương pháp phân cụm

Mô hình decision Tree tối ưu khi k=4.

Bảng 3. 2 Kết quả mô hình Decision Tree kết hợp phân cụm FCM

Số cụm	Accuracy	F1_ Macro	Precision_ Macro	Recall_ Macro	F1_ Weighted	Precision_ Weighted	Recall_ Weighted
1	0.9897	0.7392	0.8009	0.7236	0.9887	0.9891	0.9897
2	0.9889	0.8586	0.9235	0.8218	0.9885	0.9889	0.9889
3	0.9914	0.84.14	0.9278	0.8334	0.9910	0.9913	0.9914
4	0.9914	0.8679	0.8912	0.8185	0.9911	0.9915	0.9914
5	0.9908	0.8496	0.9040	0.8265	0.9906	0.9912	0.9908
6	0.9906	0.8346	0.8854	0.8160	0.9904	0.9912	0.9906
7	0.9886	0.7961	0.8530	0.7786	0.9885	0.9894	0.9886
8	0.9901	0.8298	0.8727	0.8203	0.9900	0.9906	0.9901

Mô hình Random forest tối ưu với k = 8 với độ chính xác accuracy đạt 98,81%. Tham số Marco cũng đạt được kết quả khá tốt (F1_macro = 74.99%, Precision_macro = 82,82%, Recal_macro = 72,37%).

Bảng 3. 3 Kết quả mô hình Random forest kết hợp phân cụm FCM

Số Cụm	Accuracy	F1_ Macro	Precision _Macro	Recall_ Macro	F1_ Weighted	Precision_ Weighted	Recall_ Weighted
2	0.9881	0.7363	0.7931	0.7193	0.9869	0.9868	0.9881
3	0.9832	0.7107	0.8083	0.6885	0.9813	0.9830	0.9832
4	0.9868	0.7221	0.8306	0.6982	0.9851	0.9855	0.9868
5	0.9879	0.7332	0.8192	0.7146	0.9866	0.9869	0.9879
6	0.9873	0.7215	0.8582	0.7039	0.9858	0.9871	0.9873
7	0.9878	0.7270	0.7967	0.7149	0.9866	0.9870	0.9878
8	0.9879	0.7499	0.8282	0.7237	0.9867	0.9867	0.9879
9	0.9884	0.7444	0.8353	0.7231	0.9872	0.9876	0.9884

3.2.3.2. So sánh chỉ số hiệu xuất mô hình học máy kết hợp phương pháp phân cụm và mô hình học máy truyền thống

- Chỉ số hiệu suất của mô hình học máy

Kết quả bảng 3.4 cho thấy phương pháp phân cụm đã tạo ra tác động tích cực đến hiệu suất của mô hình học máy. Kết quả đều được cải thiện, đặc biệt là mô hình Decision Tree.

Bảng 3. 4. So sánh hiệu suất mô hình học máy truyền thông và mô hình học máy kết hợp phân cụm FCM

Mô hình	Decision Tree		RandomForest	
	Mô hình truyền thống	Mô hình kế hợp phân cụm (K =4)	Mô hình truyền thống	Mô hình kết hợp phân cụm (K =8)
Accuracy	0.9523	0.9914	0.9877	0.9879
F1_Macro	0.4878	0.86.79	0.7398	0.7499
Precision_Macro	0.7072	0.8912	0.7771	0.8282
Recall_Macro	0.477	0.8185	0.7253	0.7237
F1_Weighted	0.9459	0.9911	0.9867	0.9867
Precision_Weighted	0.9544	0.9915	0.9865	0.9867
Recall_Weighted	0.9523	0.9914	0.9877	0.9879

- Độ chính xác theo từng lớp tấn công

Bảng 3. 5 Báo cáo phân loại theo lớp tấn công mô hình Decison Tree

Tên lớp tấn công	Decision Tree kết hợp phân cụm FCM (k=4)			Decision Tree truyền thống			Số mẫu
	Precision	Recall	F1-score	Precision	Recall	F1-score	
Backdoor_Malware	0.81	0.53	0.64	0.33	0.02	0.04	47
BenignTraffic	0.88	0.94	0.91	0.34	0.98	0.51	2172
BrowserHijacking	0.91	0.62	0.73	0.40	0.04	0.08	47
CommandInjection	1.00	0.38	0.55	0.00	0.00	0.00	47
DDoS-ACK_Fragmentation	0.99	0.99	0.99	0.00	0.00	0.00	548
DDoS-HTTP_Flood	0.90	0.98	0.94	1.00	0.11	0.19	47
DDoS-ICMP_Flood	1.00	1.00	1.00	1.00	1.00	1.00	14378
DDoS-ICMP_Fragmentation	0.94	1.00	0.97	0.00	0.00	0.00	883
DDoS-PSHACK_Flood	1.00	1.00	1.00	0.99	1.00	0.99	8166
DDoS-RSTFINFlood	1.00	1.00	1.00	1.00	1.00	1.00	8068
DDoS-SYN_Flood	1.00	1.00	1.00	1.00	1.00	1.00	8096

DDoS-SlowLoris	1.00	0.89	0.94	0.62	0.17	0.27	47
DDoS-SynonymousIP_Flood	1.00	1.00	1.00	1.00	1.00	1.00	7174
DDoS-TCP_Flood	1.00	1.00	1.00	1.00	0.97	0.98	8972
DDoS-UDP_Flood	1.00	1.00	1.00	1.00	0.99	1.00	10800
DDoS-UDP_Fragmentation	0.98	1.00	0.99	0.00	0.00	0.00	552
DNS_Spoofing	0.74	0.68	0.71	0.86	0.05	0.10	332
DictionaryBruteForce	0.86	0.51	0.64	1.00	0.04	0.08	47
DoS-HTTP_Flood	1.00	1.00	1.00	1.00	0.91	0.95	126
DoS-SYN_Flood	0.99	1.00	0.99	1.00	0.98	0.99	4036
DoS-TCP_Flood	1.00	0.99	1.00	1.00	1.00	1.00	5319
DoS-UDP_Flood	0.99	1.00	1.00	1.00	1.00	1.00	6614
MITM-ArpSpoofing	0.76	0.81	0.79	0.78	0.32	0.45	593
Mirai-greeth_flood	1.00	1.00	1.00	1.00	0.99	1.00	1962
Mirai-greip_flood	1.00	1.00	1.00	1.00	0.98	0.99	1480
Mirai-udpplain	0.99	1.00	1.00	0.96	0.93	0.94	1760
Recon-HostDiscovery	0.96	0.73	0.83	0.52	0.44	0.48	240
Recon-OSScan	0.85	0.61	0.71	0.59	0.11	0.18	179
Recon-PingSweep	0.85	0.49	0.62	1.00	0.02	0.04	47
Recon-PortScan	0.75	0.56	0.64	0.67	0.10	0.17	140
SqlInjection	0.67	0.64	0.65	1.00	0.02	0.04	47
Uploading_Attack	0.85	0.47	0.60	0.00	0.00	0.00	47
VulnerabilityScan	0.98	0.91	0.95	0.00	0.00	0.00	47
XSS	0.90	0.60	0.72	1.00	0.06	0.12	47
accuracy			0.99			0.95	93107
macro avg	0.927	0.83	0.86	0.95	0.47	0.48	93107
weighted avg	0.99	0.99	0.99	0.95	0.95	0.94	93107

Một số lớp có mức tối ưu dưới mức trung bình có số mẫu quá nhỏ để mô hình có thể học được đặc trưng. Tỷ lệ rất nhỏ so với tổng số lượng mẫu nên có thể chấp nhận

được. Bảng 3.6 so sánh hiệu suất giữa mô hình Random Forest truyền thống và mô hình Random Forest kết hợp phương pháp phân cụm FCM. Từ bảng so sánh có thể thấy, mô hình truyền thống đã có khả năng phân loại đã đạt mức cao: phần lớn các mẫu được dự đoán chính xác tập trung trên đường chéo chính, trong đó có 9 lớp đạt độ chính xác tối (1.0) và 8 lớp đạt độ chính xác trên 0.9.

Bảng 3. 6 Báo cáo phân loại theo lớp của mô hình Random Forest

Tên lớp tấn công	**Random Forest kết hợp phân cụm FCM (K=8)**			**Random forest truyền thống**			**Số mẫu**
	Precision	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**	
Backdoor_Malware	0.38	0.17	0.24	0.26	0.28	0.27	47
BenignTraffic	0.86	0.97	0.91	0.81	0.97	0.89	2172
BrowserHijacking	0.71	0.26	0.38	0.37	0.28	0.32	47
CommandInjection	0.31	0.11	0.16	0.38	0.19	0.25	47
DDoS-ACK_Fragmentation	0.99	0.98	0.99	0.99	0.98	0.98	548
DDoS-HTTP_Flood	0.93	0.87	0.90	0.87	0.98	0.92	47
DDoS-ICMP_Flood	1.00	1.00	1.00	1.00	1.00	1.00	14378
DDoS-ICMP_Fragmentation	0.98	0.98	0.98	0.99	0.98	0.99	883
DDoS-PSHACK_Flood	1.00	1.00	1.00	1.00	1.00	1.00	8166
DDoS-RSTFINFlood	1.00	1.00	1.00	1.00	1.00	1.00	8068
DDoS-SYN_Flood	0.99	1.00	1.00	1.00	1.00	1.00	8096
DDoS-SlowLoris	0.71	0.77	0.73	0.80	0.83	0.81	47

DDoS-SynonymousIP_Flood	1.00	0.99	1.00	1.00	1.00	1.00	7174
DDoS-TCP_Flood	1.00	1.00	1.00	1.00	1.00	1.00	8972
DDoS-UDP_Flood	1.00	1.00	1.00	1.00	1.00	1.00	10800
DDoS-UDP_Fragmentation	0.99	0.99	0.99	0.99	0.99	0.99	552
DNS_Spoofing	0.58	0.62	0.60	0.65	0.60	0.63	332
DictionaryBruteForce	0.57	0.09	0.15	0.36	0.11	0.16	47
DoS-HTTP_Flood	0.92	0.98	0.95	0.95	0.96	0.96	126
DoS-SYN_Flood	1.00	1.00	1.00	1.00	1.00	1.00	4036
DoS-TCP_Flood	0.99	1.00	0.99	1.00	1.00	1.00	5319
DoS-UDP_Flood	0.98	1.00	0.99	1.00	1.00	1.00	6614
MITM-ArpSpoofing	0.83	0.78	0.80	0.81	0.75	0.78	593
Mirai-greeth_flood	1.00	1.00	1.00	1.00	1.00	1.00	1962
Mirai-greip_flood	0.99	0.99	0.99	1.00	0.99	0.99	1480
Mirai-udpplain	1.00	0.99	1.00	1.00	0.99	1.00	1760
Recon-HostDiscovery	0.82	0.68	0.74	0.70	0.65	0.68	240
Recon-OSScan	0.63	0.41	0.50	0.58	0.35	0.44	179
Recon-PingSweep	0.56	0.21	0.31	0.45	0.21	0.29	47
Recon-PortScan	0.57	0.39	0.46	0.58	0.37	0.45	140
SqlInjection	0.86	0.26	0.39	0.21	0.09	0.12	47
Uploading_Attack	0.73	0.17	0.28	0.47	0.17	0.25	47
VulnerabilityScan	0.93	0.81	0.86	0.87	0.87	0.87	47
XSS	0.36	0.17	0.23	0.33	0.09	0.14	47
accuracy	0.98			0.98			93107

macro avg	0.82	0.72	0.74	0.77	0.7	0.73	93107
weighted avg	0.98	0.98	0.98	0.98	1	0.98	93107

Khi dữ liệu được xử lý tiền xử lý bằng phương pháp phân cụm FCM, mô hình cho thấy sự cải thiện đáng chú ý hơn nữa. Cụ thể, số lớp đạt độ chính xác tối ưu tăng lên 11 lớp, và các lớp đạt trên mức 0.9 tăng lên 9 lớp. Sự gia tăng này cho thấy việc phân cụm đã góp phần giảm bớt sự chồng lấn giữa các không gian đặc trưng của các loại tấn công, từ đó cải thiện ranh giới quyết định của mô hình trong giai đoạn học.Mặc dù mức cải thiện không quá đột phá, nhưng kết quả thu được chứng minh rằng việc kết hợp phân cụm FCM mang lại hiệu quả tích cực đối với Random Forest một mô hình học máy vốn đã mạnh về khả năng tổng quát hóa.

Kết quả đã chứng minh rằng việc kết hợp phân cụm dữ liệu không chỉ giúp tối ưu hóa mô hình mà còn có thể cải thiện hiệu suất phân loại, đặc biệt khi xử lý các tập dữ liệu lớn với nhiều tham số như tập dữ liệu CICIoT2023.

Từ kết quả phân tích, đề án chọn ra mô hình tối ưu nhất là mô hình học máy Decision Tree kết hợp phương pháp phân cụm FCM.

3.3. Kết luận chương 3

Trong chương này, mô hình IDS dựa trên học máy kết hợp phân cụm FCM đã được triển khai, thử nghiệm và đánh giá trên tập dữ liệu CICIoT2023. Kết quả thực nghiệm cho thấy việc tích hợp phân cụm trước khi phân lớp giúp cải thiện hiệu năng phát hiện xâm nhập, đặc biệt trong bối cảnh dữ liệu mất cân bằng đặc trưng của môi trường IoT. Qua phân tích các chỉ số đánh giá, mô hình học máy decision Tree kết hợp phương pháp phân cụm với số cụm k=4 là cấu hình tối ưu, duy trì độ chính xác 99,14%, đặc biệt F1_Macro đạt 86.79%. Điều này chứng minh rằng kỹ thuật phân cụm giúp tăng cường nhận diện các mẫu bất thường, giảm tỷ lệ dự đoán sai và tối ưu hóa khả năng học của mô hình. Như vậy, kết quả trong chương 3 đã xác thực tính khả thi và hiệu quả của mô hình IDS đề xuất khi áp dụng trong môi trường IoT thực tế

KẾT LUẬN VÀ KIẾN NGHỊ

Trong bối cảnh các hệ thống IoT và IIoT ngày càng mở rộng, nguy cơ tấn công mạng trở nên đa dạng và tinh vi hơn, đòi hỏi sự phát triển của các hệ thống phát hiện xâm nhập (IDS) có hiệu năng cao, độ tin cậy lớn và khả năng triển khai trên môi trường tính toán biên hạn chế tài nguyên. Đề án đã tập trung nghiên cứu, đề xuất và đánh giá mô hình IDS kết hợp học máy và phân cụm nhằm nâng cao hiệu quả phát hiện bất thường trên tập dữ liệu CICIoT2023.

Trên cơ sở phân tích dữ liệu thực nghiệm, mô hình phân cụm FCM đã chứng minh được tính khả thi trong việc xử lý dữ liệu mất cân bằng đặc trưng phổ biến trong an ninh IoT. Đặc biệt, mô hình học máy Decision Tree kết hợp phương pháp phân cụm với số cụm tối ưu K=4 được xác định là tối ưu, giúp cải thiện độ chính xác, giảm lỗi nhận diện và tăng cường khả năng phát hiện các mẫu tấn công ẩn giấu. Kết quả này cho thấy kỹ thuật phân cụm đóng vai trò quan trọng trong bước tiền xử lý, góp phần tối ưu hóa việc học của mô hình và tăng hiệu năng của IDS trên môi trường IoT.

Nghiên cứu đã cung cấp đánh giá toàn diện về mô hình, chỉ ra tính hiệu quả của giải pháp đề xuất khi so sánh với mô hình gốc không sử dụng phân cụm. Qua đó khẳng định rằng phương pháp phân cụm là hướng tiếp cận phù hợp trong phát triển các hệ thống IDS thế hệ mới cho IoT..

Trong tương lai, nghiên cứu sẽ tiếp tục mở rộng theo các định hướng sau:

- Tối ưu hóa thuật toán phân cụm nhằm giảm chi phí tính toán và độ trễ, phục vụ triển khai thực tế trên thiết bị biên.
- Kết hợp phân cụm với các cơ chế học nhẹ, đảm bảo khả năng cập nhật mô hình liên tục mà không gây quá tải tài nguyên.
- Kết hợp với phương pháp giảm chiều đề giảm số đặc trưng, tăng tốc độ tính toán.
- Nghiên cứu mô hình học phân tán và ra quyết định hợp nhất từ nhiều thiết bị IoT , tối ưu hóa xử lý dữ liệu ở cấp độ mạng.
- Tiếp tục thử nghiệm trên các bộ dữ liệu lớn và kịch bản thực tế trong IoT để đánh giá khả năng tổng quát của mô hình.