TRƯỜNG ĐẠI HỌC SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN

KHAI THÁC DỮ LIỆU VÀ ỨNG DỤNG

ĐỀ TÀI:
PHÂN TÍCH, KHÁM PHÁ, PHÂN LOẠI TẤN CÔNG CỦA BỘ DỮ LIỆU RT-IoT 2022

Giảng viên hướng dẫn:

ThS. Nguyễn Thanh Phước

Sinh viên thực hiện:

Tạ Hồng Quí

MSSV: 3122410348

Tháng 05 - 2025

Mục lục

I Giới thiệu

Trong những năm gần đây, sự phát triển mạnh mẽ của các hệ thống Internet of Things (IoT) đã mở ra nhiều ứng dụng quan trọng trong các lĩnh vực như thành phố thông minh, y tế, và công nghiệp 4.0. Tuy nhiên, sự phổ biến của các thiết bị IoT cũng đi kèm với những thách thức lớn về bảo mật mạng. Do hạn chế về tài nguyên tính toán và việc thiếu các bản cập nhật bảo mật thường xuyên, các thiết bị IoT trở thành mục tiêu hấp dẫn cho các cuộc tấn công mạng như DDoS, quét lỗ hổng, giả mạo (spoofing), và tấn công từ chối dịch vụ (DoS). Ví dụ, một cuộc tấn công DDoS nhắm vào hệ thống giám sát giao thông tại một thành phố thông minh có thể gây ra hỗn loạn nghiêm trọng, ảnh hưởng đến an toàn và trật tự công cộng.

Trước thực trạng này, việc phát hiện và phân loại các cuộc tấn công mạng trong hệ thống IoT trở thành một yêu cầu cấp thiết nhằm đảm bảo tính toàn vẹn và an toàn cho các hệ thống này. Bộ dữ liệu RT-IoT2022 [1], với 123119 mẫu dữ liệu và 84 đặc trưng luồng mạng, cung cấp một nền tảng phong phú để nghiên cứu các mẫu tấn công trong môi trường IoT. Bộ dữ liệu bao gồm 12 loại tấn công khác nhau, trong đó lớp `DOS_SYN_Hping` chiếm ưu thế với tỷ lệ 76.89%, phản ánh tính mất cân bằng điển hình trong các tập dữ liệu thực tế.

Nghiên cứu này tập trung vào việc phân tích, khám phá và phân loại các loại tấn công mạng dựa trên bộ dữ liệu RT-IoT2022, sử dụng các kỹ thuật học máy tiên tiến. Cụ thể, chúng tôi áp dụng các phương pháp như phân tích thành phần chính (PCA) để giảm chiều dữ liệu, SHAP (SHapley Additive exPlanations), kết hợp với các phương pháp thống kê khác nhau để xác định các đặc trưng quan trọng, và các mô hình học máy như: LinearSVC, KNN, XGBoost, Random Forest, MLP, Logistic Regression để phân loại tấn công. Mục tiêu chính bao gồm: (i) xây dựng mô hình phân loại đạt độ chính xác cao, (ii) xác định các đặc trưng then chốt ảnh hưởng đến việc phát hiện tấn công, (iii) đánh giá hiệu suất của các mô hình để đảm bảo tính khả thi trong ứng dụng thực tế.

Các nghiên cứu trước đây, chẳng hạn như Sharmila et al. [2], đã đánh giá hiệu suất của các mô hình học máy tham số và phi tham số trên tập dữ liệu RT-IoT2022, tập trung vào phân tích thống kê và so sánh độ chính xác. Tuy nhiên, nghiên cứu này chủ yếu dựa trên các đặc trưng thô mà không áp dụng kỹ thuật giảm chiều hoặc phân tích chi tiết vai trò của từng đặc trưng. Ngược lại, nghiên cứu hiện tại đề xuất một cách tiếp cận toàn diện hơn bằng cách tích hợp PCA để giảm chiều dữ liệu, SHAP để xác định các đặc trưng quan trọng, và sử dụng một loạt các mô hình học máy đa dạng (LinearSVC, KNN, XGBoost, Random Forest, MLP, Logistic Regression). Sự khác biệt này cho phép không chỉ đạt được độ chính xác cao mà còn hiểu rõ hơn về đóng góp của các đặc trưng trong việc phát hiện tấn công, đồng thời tối ưu hóa hiệu suất tính toán cho các ứng dụng IoT thực tế.

Ngoài ra, nghiên cứu cũng so sánh hiệu suất của các mô hình học máy khác nhau và

phân tích tính tương quan giữa các đặc trưng thông qua các công cụ như heatmap và biểu đồ radar, từ đó làm rõ sự phức tạp và phụ thuộc trong dữ liệu. Kết quả nghiên cứu không chỉ góp phần nâng cao khả năng phát hiện tấn công trong hệ thống IoT mà còn cung cấp cơ sở khoa học để tối ưu hóa các giải pháp bảo mật mạng trong tương lai.

II Hiểu bài toán và mục tiêu nghiên cứu

Trong bối cảnh các hệ thống IoT (Internet of Things) ngày càng phổ biến trong đời sống và công nghiệp, đặc biệt là trong các thành phố thông minh và lĩnh vực y tế, bảo mật mạng trở thành một thách thức lớn. Các thiết bị IoT thường có tài nguyên hạn chế và ít được cập nhật bảo mật, tạo điều kiện cho tin tặc thực hiện các cuộc tấn công như DDoS, quét lỗ hổng, tấn công giả mạo. Ví dụ, một cuộc tấn công DDoS vào hệ thống giám sát giao thông có thể khiến cả thành phố rơi vào hỗn loạn. Do đó, việc phát hiện sớm các cuộc tấn công là yêu cầu bắt buộc nhằm đảm bảo tính toàn vẹn và an toàn cho hệ thống.

1 Mục tiêu dự án

- Sử dụng các thuật toán học máy để phân loại và phát hiện các loại tấn công mạng dựa trên bộ dữ liệu RT-IoT2022.
- So sánh hiệu suất của nhiều mô hình học máy khác nhau và đánh giá khả năng áp dụng vào thực tế.
- Xác định các đặc trưng (features) quan trọng nhất hỗ trợ việc phát hiện tấn công.

2 Câu hỏi nghiên cứu

- Liệu các thuật toán học máy có thể phân loại chính xác các loại tấn công mạng trong hệ thống IoT?
- Đặc trưng nào trong bộ dữ liệu đóng vai trò quan trọng nhất?
- Mô hình nào đạt được sự cân bằng tốt nhất giữa độ chính xác và hiệu suất tính toán?

3 Tiêu chí thành công

- Các mô hình học máy đạt độ chính xác phân loại (accuracy) ít nhất 90% trên tập dữ liệu kiểm tra RT-IoT2022.
- Xác định ít nhất 5 đặc trưng quan trọng nhất (dựa trên phương pháp SHAP hoặc tương tự) và giải thích vai trò của chúng trong việc phát hiện tấn công.
- Mô hình được chọn có thời gian dự đoán trên mỗi mẫu dữ liệu không vượt quá 1ms, đảm bảo khả năng áp dụng thực tế trong hệ thống IoT.

III Mô Tả Bộ Dữ Liệu

1 Nguồn Gốc

Bộ dữ liệu **RT-IoT2022** được thu thập từ cơ sở hạ tầng mạng IoT mô phỏng tại phòng thí nghiệm của The National Institute of Engineering, Mysuru, Karnataka, India [1]. Dữ liệu bao gồm **123,119 mẫu** với **85 đặc trưng** (cả số và văn bản), được trích xuất từ lưu lượng mạng hai chiều (bidirectional flow-based) của các thiết bị IoT thực tế như Wipro Lights, ThingSpeak LEDs, Amazon Echo và MQTT-Temp. Lưu lượng tấn công được tạo ra bởi các công cụ như Nmap, Metasploit, Hping3 và Ettercap, mô phỏng các cuộc tấn công như Brute Force, DDoS, Nmap scans và ARP Poisoning.

1.1 Xuất Xứ và Mục Đích

Bộ dữ liệu được công bố trên UCI Machine Learning Repository [1] nhằm hỗ trợ nghiên cứu phát hiện tấn công mạng trong hệ thống IoT thời gian thực. Nó khắc phục các hạn chế của các bộ dữ liệu trước như KDDCUP99 (dư thừa, lỗi thời), UNSW-NB15 (mất cân bằng) và CICIDS-2017 (thiếu hỗ trợ IPv6), bằng cách cung cấp dữ liệu thực tế từ thiết bị IoT và đặc trưng bidirectional phù hợp với các kịch bản bảo mật mạng IoT.

1.2 Phương Pháp Thu Thập

Dữ liệu được thu thập thông qua cơ sở hạ tầng gồm mạng nạn nhân, mạng tấn công và router Raspberry Pi chạy Kali Linux. Công cụ **Wireshark** ghi lại gói tin mạng dưới dạng PCAP, sử dụng thư viện `libpcap`. Các đặc trưng bidirectional được trích xuất bằng **Zeek** và plugin **Flowmeter**, lưu dưới dạng CSV. Các dịch vụ liên quan bao gồm DNS, HTTP, MQTT, SSL, IRC, SSH, DHCP, NTP và RADIUS.

2 Đặc Điểm Nổi Bật

- **Tính thực tế**: Sử dụng thiết bị IoT thực (Amazon Echo, Philips HUE) và công cụ tấn công thực tế, khác với dữ liệu mô phỏng như HIKARI-2021.
- **Đa dạng tấn công**: Bao gồm Brute Force, DDoS, Nmap scans và ARP Poisoning, phản ánh các mối đe dọa mạng hiện đại.
- **Phân phối nhãn**: Không cân bằng nghiêm trọng, ví dụ: Nmap FIN SCAN (28 mẫu) so với SYN Flood DDoS (94,659 mẫu).

IV Mô Tả Quá Trình Thu Thập Dữ Liệu RT-IoT2022

1 Cơ Sở Hạ Tầng Thu Thập Dữ Liệu

Cơ sở hạ tầng thu thập dữ liệu cho bộ dữ liệu **RT-IoT2022** được thiết kế để mô phỏng môi trường mạng IoT thực tế, bao gồm các thành phần chính sau:

1.1 Mạng Nạn Nhân

Mạng nạn nhân bao gồm các thiết bị IoT thực tế như Wipro Lights, ThingSpeak LEDs, Amazon Echo và MQTT-Temp. Những thiết bị này tạo ra lưu lượng mạng hợp pháp (benign traffic) trong môi trường mô phỏng, đại diện cho các hoạt động thông thường của hệ thống IoT.

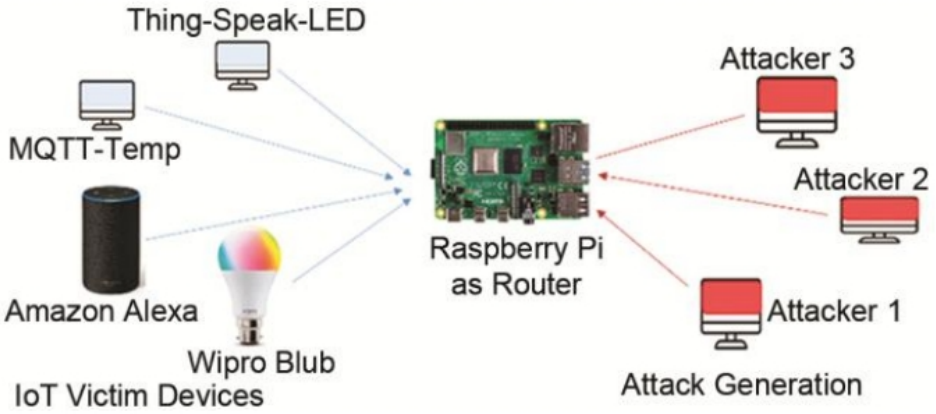
1.2 Mạng Tấn Công

Mạng tấn công được xây dựng với hai máy ảo (VMs) và một máy tính Raspberry Pi chạy hệ điều hành Kali Linux. Các thiết bị này được sử dụng để thực hiện các cuộc tấn công mạng, sử dụng các công cụ kiểm tra xâm nhập như Nmap, Wireshark, Ettercap và Burp Suite để tạo ra các kịch bản tấn công đa dạng.

1.3 Thiết Bị Định Tuyến

Thiết bị định tuyến được xây dựng trên Raspberry Pi, cài đặt hệ điều hành Kali Linux. Router sử dụng công cụ **Wireshark** để ghi lại tất cả các gói tin mạng (network packets) từ cả lưu lượng hợp pháp và lưu lượng tấn công, đảm bảo thu thập toàn bộ dữ liệu giao tiếp trong hệ thống.

Hình 1 trong bài báo của Sharmila et al. (2024) minh họa cơ sở hạ tầng này, cho thấy cách mạng tấn công giao tiếp với mạng nạn nhân thông qua router dựa trên Raspberry Pi [2].



Hình 1: Cơ sở hạ tầng thu thập dữ liệu RT-IoT2022 [2]

2 Quá Trình Thu Thập Dữ Liệu

Quá trình thu thập dữ liệu được thực hiện thông qua các bước sau:

2.1 Ghi Lại Lưu Lượng Mạng

Công cụ **Wireshark** được cài đặt trên router để ghi lại tất cả lưu lượng mạng, bao gồm cả lưu lượng hợp pháp và tấn công, dưới dạng tệp **Packet Capture (PCAP)**, sử dụng thư viện `libpcap`. Các cuộc tấn công như DDoS, quét ping (ping scans), giả mạo (spoofing) và các cuộc tấn công khác được thực hiện để khai thác điểm yếu của các giao thức tiêu chuẩn, làm cạn kiệt tài nguyên của thiết bị IoT.

2.2 Loại Tấn Công và Lưu Lượng Hợp Pháp

- **Lưu lượng hợp pháp**: Được tạo ra từ các thiết bị IoT như ThingSpeak-LED (dịch vụ DNS, HTTP), MQTT-Temp (MQTT), Amazon-Alexa (DNS, HTTP) và Wipro-Bulb (SSL, DNS, IRC).
- **Lưu lượng tấn công**: Bao gồm các loại tấn công như:
  - **Brute Force**: SSH Brute Force sử dụng Metasploit.
  - **DDoS**: Slowloris và SYN Flood DDoS sử dụng Slowloris và Hping3.
  - **Nmap**: FIN SCAN, OS Fingerprinting, UDP scan, XMAS Tree scan sử dụng Network Mapper.
  - **ARP Poisoning**: Man-in-the-Middle (MiTM) sử dụng Ettercap.

Chi tiết số lượng mẫu (instances) được ghi lại cho từng loại lưu lượng được trình bày trong Bảng 2.2, trích từ Sharmila et al. (2024) [2]:

Loại lưu lượng	Số mẫu Wireshark	Số mẫu CIC Flowmeter
ThingSpeak-LED	10,526	8,108
MQTT-Temp	8,162	4,146
Amazon-Alexa	6,056	5,023
Wipro-Bulb	1,265	253
SSH Brute Force	857	37
Slowloris DDoS	5,920	534
SYN Flood DDoS	712,850	94,659
Nmap FIN SCAN	69	28
Nmap OS Fingerprinting	8,008	2,000
Nmap UDP scan	5,606	2,590
Nmap XMAS Tree scan	8,045	2,010
ARP Poisoning (MiTM)	306	7,750

3 Tiền Xử Lý Dữ Liệu

3.1 Chuyển Đổi Sang Định Dạng Bidirectional

Do các cuộc tấn công như DDoS và spoofing khó phân biệt chỉ dựa trên lưu lượng đơn hướng (unidirectional traces), nhóm tác giả sử dụng công cụ **Zeek** với plugin **Flowmeter** để trích xuất các đặc trưng lưu lượng hai chiều (bidirectional features) từ tệp PCAP. Flowmeter tạo ra các tệp log như:

- `conn.log`: Chứa thông tin về địa chỉ IP và cổng.
- `flowmeter.log`: Chứa các đặc trưng bidirectional liên quan đến mỗi `uid`.
- Các log riêng cho các dịch vụ như DHCP, HTTP, DNS, MQTT.

3.2 Định Dạng Đầu Ra

Các đặc trưng được trích xuất và lưu dưới dạng tệp **CSV**, chứa cả lưu lượng hợp pháp và lưu lượng tấn công, sẵn sàng cho phân tích và huấn luyện mô hình học máy.

4 Đặc Điểm Bộ Dữ Liệu

Bộ dữ liệu **RT-IoT2022** bao gồm cả lưu lượng mạng hợp pháp và tấn công từ các thiết bị IoT thực tế, được trích xuất dưới dạng đặc trưng *bidirectional flow-based*. Các dịch vụ liên quan bao gồm DNS, HTTP, MQTT, SSL, IRC, SSH, DHCP, NTP và RADIUS. Bộ dữ liệu được thiết kế để khắc phục các hạn chế của các bộ dữ liệu trước đó như KDDCUP99, UNSW-NB15 và CICIDS-2017, bằng cách cung cấp dữ liệu thực tế hơn và phù hợp với các kịch bản bảo mật mạng IoT.

5 Điểm Nổi Bật

- **Tính thực tế**: Không giống các bộ dữ liệu mô phỏng hoàn toàn như HIKARI-2021, RT-IoT2022 sử dụng các thiết bị IoT thực tế (như Amazon Echo, Philips HUE) và các công cụ tấn công thực tế (Nmap, Metasploit).
- **Đa dạng tấn công**: Bao gồm nhiều loại tấn công hiện đại như Brute Force, DDoS, Nmap scans và ARP Poisoning, phản ánh các mối đe dọa mạng thực tế.
- **Hiệu quả trích xuất đặc trưng**: Sử dụng Zeek và Flowmeter để tạo đặc trưng bidirectional, giúp cải thiện khả năng phát hiện các cuộc tấn công phức tạp.

6 Giá trị bị khuyết

Các giá trị bị khuyết trong các cột số được xử lý bằng cách thay thế bằng giá trị trung bình. Cột "service" có giá trị "-" chiếm 83.5% nên được loại bỏ khỏi tập dữ liệu.

V Tiền Xử Lý Dữ Liệu

Các bước tiền xử lý

1. **Mã hóa dữ liệu**: Cột "proto" (tcp, udp, icmp) được mã hóa bằng **OneHotEncoder** để chuyển đổi thành định dạng số phù hợp cho các mô hình học máy.

2. **Chuẩn hóa dữ liệu**: Sử dụng **StandardScaler** để chuẩn hóa các đặc trưng, đảm bảo tính nhất quán và cải thiện hiệu quả cho các mô hình như SVM và KNN.

3. **Xử lý mất cân bằng**: Áp dụng **SMOTE (Synthetic Minority Over-sampling Technique)** để tăng mẫu cho các lớp thiểu số (ví dụ: NMAP_TCP_scan, DDOS_Slowloris, Wipro_bulb, Metasploit_Brute_Force_SSH, NMAP_FIN_SCAN), đảm bảo phân phối nhãn đồng đều hơn.

4. **Giảm chiều dữ liệu**: Sử dụng **Feature Importance** từ Random Forest và ngưỡng tương quan (> 0.8) để giảm từ 85 đặc trưng xuống còn khoảng **40 đặc trưng** liên quan đến lưu lượng mạng và hành vi tấn công.

5. **Phân chia dữ liệu**: Dữ liệu được chia thành **80% tập huấn luyện** và **20% tập kiểm tra**, sử dụng phân phối nhãn đồng đều với tham số `stratify`.

VI Phân Tích Dữ Liệu Khám Phá

1 Thông kê mô tả

1.1 Thông tin cơ bản về luồng mạng

- **id.orig_p**: Cổng nguồn (source port) của luồng mạng.
- **id.resp_p**: Cổng đích (destination port) của luồng mạng.
- **proto**: Giao thức mạng được sử dụng (ví dụ: TCP, UDP).
- **service**: Dịch vụ liên quan đến luồng mạng (ví dụ: MQTT, HTTP, hoặc "-").
- **flow_duration**: Thời gian tồn tại của luồng mạng (tính bằng giây).

1.2 Thống kê gói tin

- **fwd_pkts_tot**: Tổng số gói tin từ nguồn đến đích (forward).
- **bwd_pkts_tot**: Tổng số gói tin từ đích đến nguồn (backward).
- **fwd_data_pkts_tot**: Gói tin chứa dữ liệu từ nguồn đến đích.
- **bwd_data_pkts_tot**: Gói tin chứa dữ liệu từ đích đến nguồn.
- **fwd_pkts_per_sec**: Tốc độ forward packets mỗi giây.
- **bwd_pkts_per_sec**: Tốc độ backward packets mỗi giây.
- **flow_pkts_per_sec**: Tổng tốc độ gói tin mỗi giây.
- **down_up_ratio**: Tỷ lệ giữa gói tin backward và forward.

1.3 Kích thước tiêu đề gói tin

- **fwd_header_size_tot, min, max**: Tổng, nhỏ nhất, lớn nhất kích thước tiêu đề gói forward.
- **bwd_header_size_tot, min, max**: Tổng, nhỏ nhất, lớn nhất kích thước tiêu đề gói backward.

1.4 Cờ TCP

- **flow_FIN_flag_count, SYN, RST, ACK, CWR, ECE**: Số lần xuất hiện các cờ FIN, SYN, RST, ACK, CWR, ECE trong toàn bộ luồng.
- **fwd_PSH_flag_count, fwd_URG_flag_count**: Số lần xuất hiện cờ PSH, URG trong forward packets.
- **bwd_PSH_flag_count, bwd_URG_flag_count**: Số lần xuất hiện cờ PSH, URG trong backward packets.

1.5 Thống kê tải trọng gói tin

- **fwd_pkts_payload** (min, max, tot, avg, std): Kích thước tải trọng gói tin forward.
- **bwd_pkts_payload** (min, max, tot, avg, std): Kích thước tải trọng gói tin backward.
- **flow_pkts_payload** (min, max, tot, avg, std): Kích thước tải trọng toàn bộ luồng.

1.6 Thống kê thời gian giữa các gói tin

- **fwd_iat** (min, max, tot, avg, std): Thống kê thời gian giữa các gói tin forward.
- **bwd_iat** (min, max, tot, avg, std): Thống kê thời gian giữa các gói tin backward.
- **flow_iat** (min, max, tot, avg, std): Thống kê thời gian giữa các gói tin bất kỳ.

1.7 Thống kê luồng phụ

- **fwd_subflow_pkts, bwd_subflow_pkts**: Số gói tin trung bình mỗi subflow forward/backward.
- **fwd_subflow_bytes, bwd_subflow_bytes**: Số byte trung bình mỗi subflow forward/backward.

1.8 Thống kê khối dữ liệu

- **fwd_bulk_bytes, fwd_bulk_packets, fwd_bulk_rate**: Tổng byte, gói tin, tốc độ khối dữ liệu forward.
- **bwd_bulk_bytes, bwd_bulk_packets, bwd_bulk_rate**: Tổng byte, gói tin, tốc độ khối dữ liệu backward.

1.9 Thống kê thời gian hoạt động và nhàn rỗi

- **active** (min, max, tot, avg, std): Thời gian hoạt động của luồng.
- **idle** (min, max, tot, avg, std): Thời gian nhàn rỗi của luồng.

1.10 Kích thước cửa sổ TCP

- **fwd_init_window_size, bwd_init_window_size**: Kích thước cửa sổ ban đầu của gói forward/backward.
- **fwd_last_window_size**: Kích thước cửa sổ cuối cùng của gói forward.

1.11 Nhãn tấn công

- **Attack_type**: Loại tấn công mạng hoặc hành vi của luồng (ví dụ: `MQTT_Publish`, `DOS_SYN_Hping`).

2 Thống kê mô tả nhãn tấn công

Bảng 1: Phân bố các loại tấn công trong tập dữ liệu

Attack_type	**Count**	**Percentage (%)**
DOS_SYN_Hping	94 659	76.89
Thing_Speak	8108	6.59
ARP_poisoning	7750	6.29
MQTT_Publish	4146	3.37
NMAP_UDP_SCAN	2590	2.10
NMAP_XMAS_TREE_SCAN	2010	1.63
NMAP_OS_DETECTION	2000	1.62
NMAP_TCP_scan	1002	0.81
DDOS_Slowloris	534	0.43
Wipro_bulb	253	0.21
Metasploit_Brute_Force_SSH	37	0.03
NMAP_FIN_SCAN	28	0.02

2.1 DOS_SYN_Hping (76.89%)

- **Cách thực hiện**: Đây là một cuộc tấn công từ chối dịch vụ phân tán (DDoS) sử dụng phương pháp SYN Flood. Công cụ `hping` được dùng để gửi một lượng lớn các gói tin SYN (yêu cầu kết nối TCP) đến máy chủ mục tiêu mà không hoàn tất handshake ba bước (SYN, SYN-ACK, ACK). Máy chủ bị quá tải khi cố gắng duy trì các kết nối nửa vời trong hàng đợi.

- **Đặc điểm**: Số lượng gói tin lớn (94659 trong dữ liệu), chiếm ưu thế tuyệt đối, cho thấy mức độ phổ biến và hiệu quả của loại tấn công này.
- **Phòng chống**: Sử dụng bộ lọc SYN cookie, giới hạn số kết nối đồng thời, hoặc triển khai hệ thống phát hiện và giảm thiểu DDoS.

2.2 ARP_poisoning (6.29%)

- **Cách thực hiện**: Tấn công này khai thác giao thức ARP (Address Resolution Protocol) bằng cách gửi các gói tin ARP giả mạo để ánh xạ địa chỉ IP của nạn nhân với địa chỉ MAC của kẻ tấn công. Điều này cho phép kẻ tấn công chặn hoặc thay đổi lưu lượng mạng (man-in-the-middle attack).
- **Đặc điểm**: Thường nhắm vào mạng nội bộ, đòi hỏi kẻ tấn công phải nằm trong cùng mạng LAN.
- **Phòng chống**: Sử dụng ARP spoofing detection tools, kích hoạt chế độ bảo mật ARP (như Dynamic ARP Inspection), hoặc sử dụng mạng ảo hóa (VLAN).

2.3 MQTT_Publish (3.37%)

- **Cách thực hiện**: Tấn công nhắm vào giao thức MQTT (Message Queuing Telemetry Transport), thường dùng trong IoT. Kẻ tấn công có thể gửi các gói tin `PUBLISH` giả mạo để gây quá tải máy chủ MQTT hoặc truyền dữ liệu độc hại.
- **Đặc điểm**: Liên quan đến dịch vụ MQTT trong dữ liệu luồng mạng, có thể nhắm vào thiết bị IoT như cảm biến hoặc đèn thông minh.
- **Phòng chống**: Xác thực mạnh mẽ (username/password, TLS), giới hạn số lượng kết nối, và giám sát lưu lượng bất thường.

2.4 Tấn công dựa trên NMAP (NMAP_UDP_SCAN, NMAP_XMAS_TREE_SCAN, v.v.)

- **Cách thực hiện**: Sử dụng công cụ NMAP để quét cổng (port scanning) nhằm phát hiện các dịch vụ mở hoặc lỗ hổng. Ví dụ:
  - `NMAP_UDP_SCAN`: Quét cổng UDP để tìm dịch vụ không bảo vệ.
  - `NMAP_XMAS_TREE_SCAN`: Gửi gói tin với các cờ TCP bất thường (FIN, PSH, URG) để tránh phát hiện.
- **Đặc điểm**: Số lượng thấp (2590, 2010, v.v.), cho thấy đây là các hoạt động thăm dò hơn là tấn công trực tiếp.
- **Phòng chống**: Cấu hình tường lửa để chặn quét cổng, ẩn dịch vụ không cần thiết.

2.5 DDOS_Slowloris (0.43%)

- **Cách thực hiện**: Tấn công DDoS chậm, giữ các kết nối HTTP mở với máy chủ bằng cách gửi các yêu cầu không hoàn chỉnh (header) theo thời gian dài, làm đầy tài nguyên máy chủ.
- **Đặc điểm**: Ít gói tin hơn (534), nhưng hiệu quả trong việc làm gián đoạn dịch vụ web.
- **Phòng chống**: Giới hạn số kết nối mỗi IP, sử dụng bộ đệm yêu cầu (request buffering).

2.6 Metasploit_Brute_Force_SSH (0.03%)

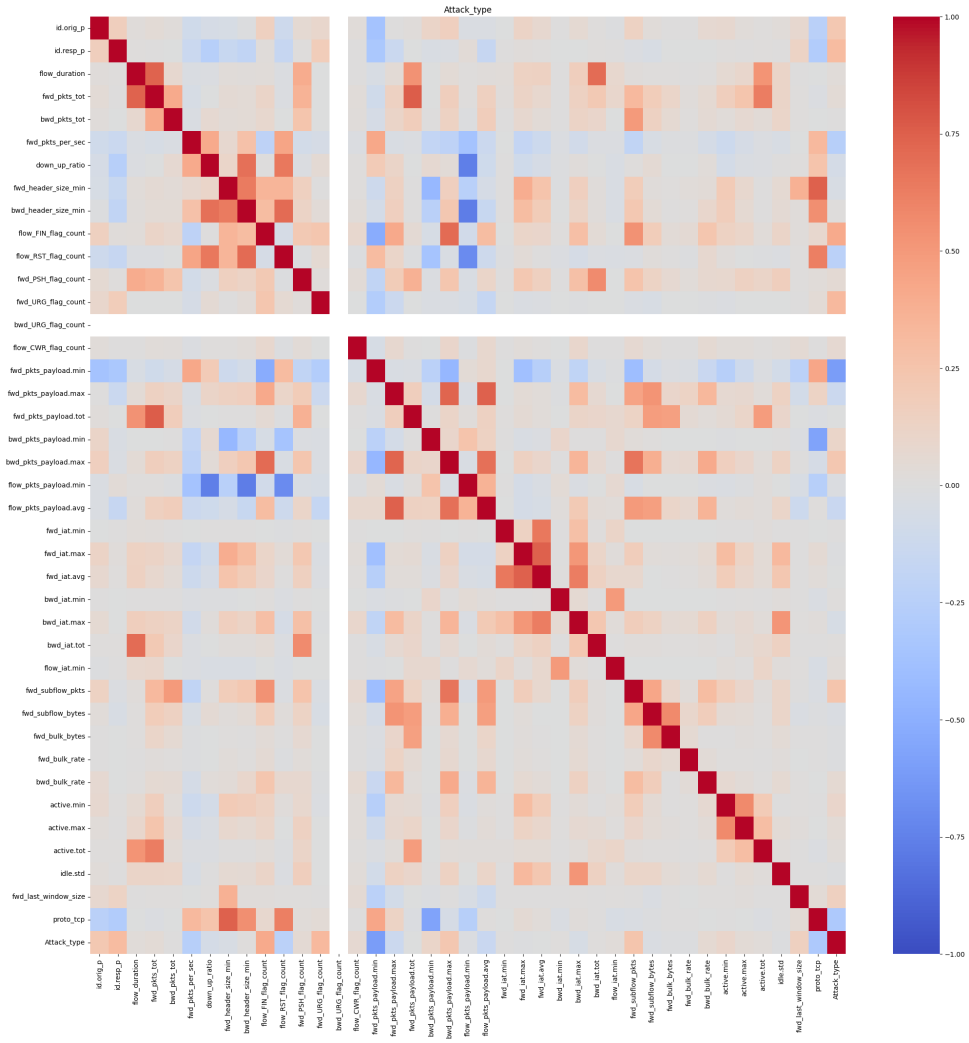
- **Cách thực hiện**: Sử dụng công cụ Metasploit để thực hiện tấn công brute force trên giao thức SSH, thử các tổ hợp mật khẩu cho đến khi đăng nhập thành công.
- **Đặc điểm**: Số lượng rất thấp (37), cho thấy đây là tấn công nhắm mục tiêu cụ thể.
- **Phòng chống**: Sử dụng khóa SSH (key-based authentication), giới hạn số lần thử mật khẩu, và chặn IP bất thường.

3 Phân tích đơn biến

- **Phân phối nhãn**: Phân tích cho thấy sự mất cân bằng nghiêm trọng giữa các lớp tấn công, với các lớp như NMAP_FIN_SCAN (28 mẫu) và Metasploit_Brute_Force_SSH (37 mẫu) có số lượng rất ít so với các lớp như DOS_SYN_Hping.
- **Đặc trưng**: Các đặc trưng liên quan đến lưu lượng mạng (network traffic) và giao thức (protocol) cho thấy sự đa dạng lớn, yêu cầu xử lý đặc trưng để loại bỏ nhiễu.

4 Phân tích tương quan sử dụng ma trận heatmap

Sử dụng ngưỡng tương quan (> 0.8) để xác định và loại bỏ các đặc trưng có hiện tượng đa cộng tuyến (multicollinearity), giúp giảm độ phức tạp của mô hình. **Feature Importance** từ Random Forest được sử dụng để xác định 40 đặc trưng quan trọng nhất, tập trung vào các thuộc tính liên quan đến hành vi tấn công và lưu lượng mạng.



Hình 2: Heatmap xác định dưới 40 đặc trưng quan trong trong bộ dữ liệu RT-IoT2022.

4.1 Quan sát chung

- **Đường chéo chính**: Các ô trên đường chéo chính có màu đỏ đậm, cho thấy tương quan hoàn hảo (gần 1) giữa mỗi đặc trưng với chính nó, điều này là hợp lý.
- **Tương quan âm mạnh**: Một số cặp đặc trưng và nhãn có tương quan âm mạnh (màu xanh đậm), đặc biệt ở các ô ngoài đường chéo, phản ánh sự phụ thuộc nghịch.
- **Phân bố không đồng đều**: Các ma trận có các vùng màu khác nhau (đỏ, cam, xanh), cho thấy mức độ tương quan thay đổi đáng kể giữa các đặc trưng và nhãn.

4.2 Phân tích chi tiết theo từng heatmap

- **Heatmap trên cùng (`Attack_type` với các đặc trưng)**:
  - Nhãn `Attack_type` có tương quan dương từ 0.2 đến 0.5 với các đặc trưng như `fwd_pkts_tot`, `bwd_pkts_tot`, `flow_duration`, và `flow_pkts_payload.tot`,

cho thấy các đặc trưng về tổng số gói tin và thời gian luồng có ảnh hưởng đáng kể đến việc phân loại tấn công.

  - Một số đặc trưng như `fwd_init_window_size` và `bwd_init_window_size` có tương quan âm nhẹ (khoảng -0.2 đến -0.4), thể hiện mối quan hệ nghịch với `Attack_type`.
  - Các cặp đặc trưng như `fwd_pkts_tot` và `bwd_pkts_tot`, `fwd_data_pkts_tot` và `bwd_data_pkts_tot` có tương quan dương mạnh (gần 1).

- **Heatmap giữa (các đặc trưng payload và timing)**:
  - Các đặc trưng như `fwd_pkts_payload.min`, `fwd_pkts_payload.max`, `fwd_pkts_payload.avg`, và `flow_pkts_payload.tot` có tương quan dương từ 0.6 đến 0.8, cho thấy sự phụ thuộc chặt chẽ giữa các chỉ số tải trọng.
  - `flow_iat.min`, `flow_iat.max`, và `flow_iat.avg` cũng có tương quan dương mạnh, phản ánh mối quan hệ giữa các khoảng thời gian giữa các gói tin.
  - Tương quan với `Attack_type` là yếu (khoảng 0.1 đến 0.3), cho thấy các đặc trưng này ít ảnh hưởng trực tiếp đến phân loại.

- **Heatmap dưới cùng (các đặc trưng subflow, bulk, và window)**:
  - `fwd_subflow_pkts` và `bwd_subflow_pkts` có tương quan dương mạnh (khoảng 0.7 đến 0.9), cho thấy sự phụ thuộc giữa số gói tin subflow forward và backward.
  - `fwd_bulk_bytes` và `fwd_bulk_packets` có tương quan dương mạnh, phản ánh mối quan hệ giữa byte và gói tin khối dữ liệu.
  - Tương quan với `Attack_type` là yếu (khoảng 0.1 đến 0.3), ngoại trừ một số đặc trưng như `active_tot` và `idle_tot` có tương quan khoảng 0.4.

4.3 Nhận xét chính

- **Ảnh hưởng của đặc trưng đến phân loại**: Các đặc trưng liên quan đến tổng số gói tin (`fwd_pkts_tot`, `bwd_pkts_tot`) và thời gian luồng (`flow_duration`) có tương quan dương đáng kể với `Attack_type`, phù hợp với đặc điểm của các cuộc tấn công như DOS_SYN_Hping.

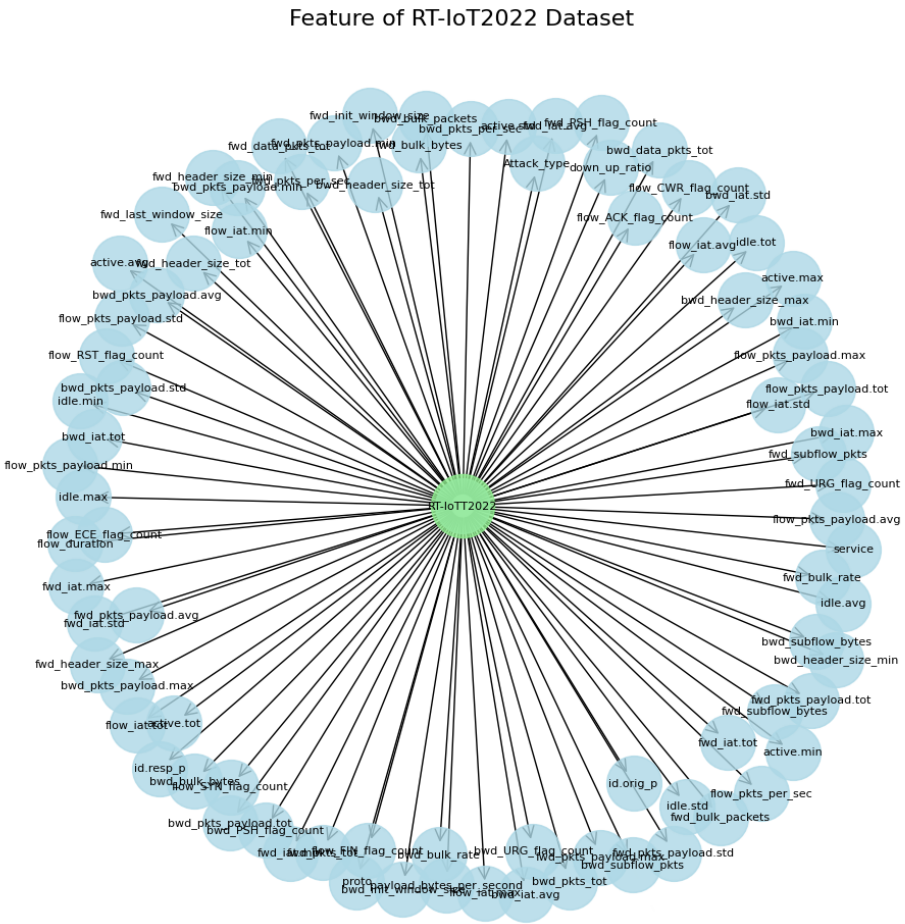
- **Mất cân bằng tương quan**: Sự hiện diện của các ô tương quan âm mạnh và dương mạnh cho thấy dữ liệu có thể bị mất cân bằng, phù hợp với tỷ lệ chiếm ưu thế của DOS_SYN_Hping (76.89%).

- **Hạn chế và gợi ý**: Tương quan yếu giữa một số đặc trưng (như window size, subflow) và `Attack_type` cho thấy cần áp dụng các phương pháp như PCA hoặc feature engineering để cải thiện khả năng phân loại.

5 Phân tích đa thuộc tính

Biểu đồ (Hình 3) minh họa tính phức tạp và đa dạng của các đặc trưng trong tập dữ liệu RT-IoT2022, nhằm làm rõ sự cần thiết của việc giảm chiều để tối ưu hóa hiệu suất mô hình học máy. Tập dữ liệu RT-IoT2022 bao gồm 84 đặc trưng, bao gồm các thông số như `id.resp_p` (cổng đích), `flow_duration` (thời gian luồng), `fwd_pkts_tot` (tổng số gói tin forward), `flow_SYN_flag_count` (số cờ SYN), và `Attack_type` (loại tấn công). Các đặc trưng này được phân loại thành nhiều nhóm như thống kê gói tin, kích thước tiêu đề, cờ TCP, và thống kê thời gian, phản ánh đầy đủ các khía cạnh của luồng mạng trong bối cảnh IoT.

Biểu đồ thể hiện mối quan hệ phức tạp giữa các đặc trưng, với các đường liên kết biểu thị sự phụ thuộc và tương quan giữa chúng. Ví dụ, các đặc trưng như `fwd_pkts_tot` và `bwd_pkts_tot` thường có tương quan cao, trong khi các đặc trưng như `flow_duration` và `flow_pkts_per_sec` có thể phản ánh các mẫu tấn công cụ thể như DOS_SYN_Hping. Sự phức tạp này nhấn mạnh vai trò của các kỹ thuật giảm chiều như PCA, giúp giảm số lượng đặc trưng mà vẫn giữ được thông tin quan trọng, từ đó cải thiện hiệu quả huấn luyện và giảm nguy cơ quá khớp (overfitting).



Hình 3: Biểu đồ minh họa các đặc trưng của tập dữ liệu RT-IoT2022, bao gồm 84 đặc trưng liên quan đến luồng mạng và nhãn `Attack_type`.

6 Phân tích đặc trưng có ảnh hưởng

6.1 Tập dữ liệu

Tập dữ liệu bao gồm các luồng mạng được ghi nhận với các đặc trưng như id.resp_p, fwd_pkts_payload.tot, và flow_iat.min. Các lớp tấn công được ánh xạ như sau:

- class 0: ARP_poisoning
- class 1: DDOS_Slowloris
- class 2: DOS_SYN_Hping
- class 3: MQTT_Publish
- class 4: Metasploit_Brute_Force_SSH
- class 5: NMAP_FIN_SCAN

- class 6: NMAP_OS_DETECTION
- class 7: NMAP_TCP_scan
- class 8: NMAP_UDP_SCAN
- class 9: NMAP_XMAS_TREE_SCAN
- class 10: Thing_Speak
- class 11: Wipro_bulb

6.2 Phương pháp SHAP

Phương pháp SHAP được sử dụng để tính toán giá trị trung bình của SHAP (`mean(|SHAP value|)`), phản ánh mức độ ảnh hưởng trung bình của từng đặc trưng lên đầu ra của mô hình. Biểu đồ SHAP được xây dựng để trực quan hóa tác động này.

6.3 Kết quả

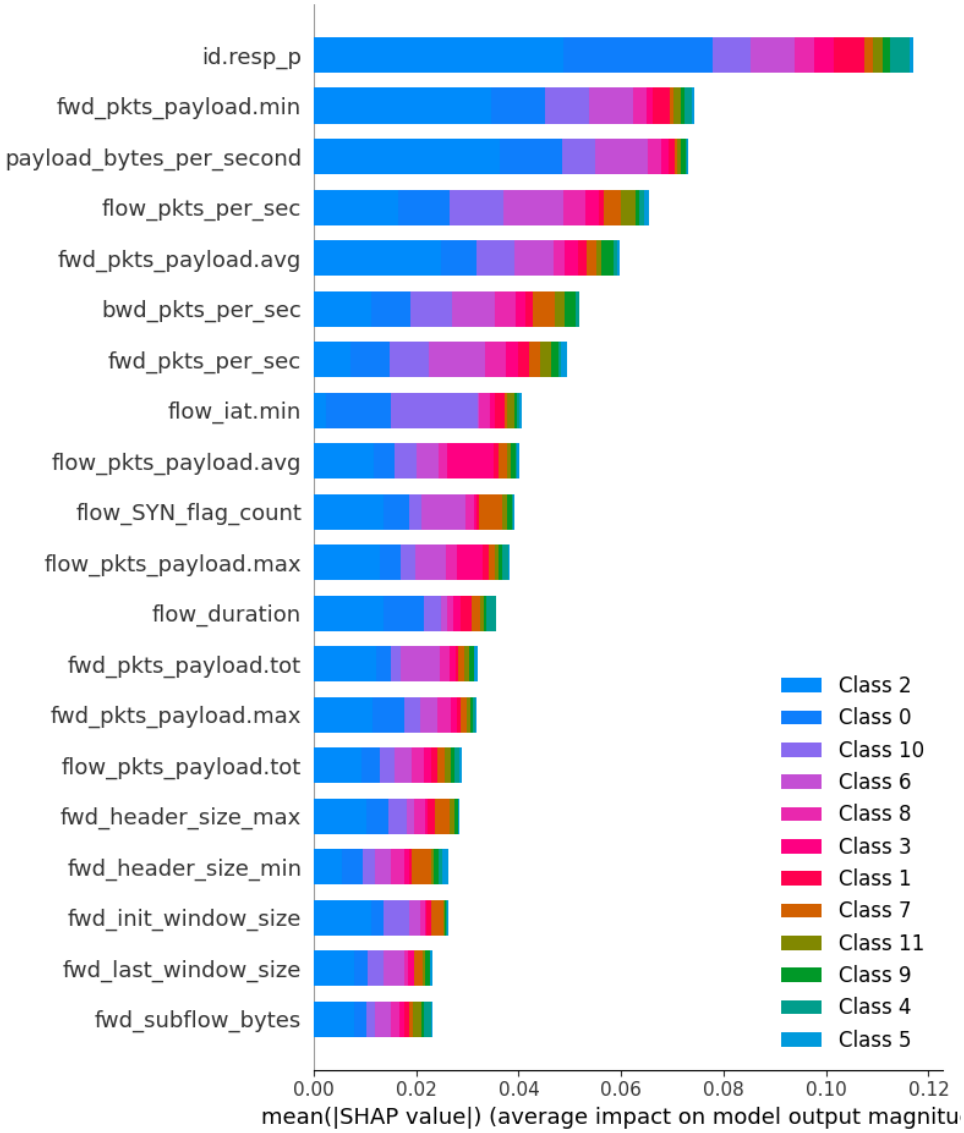
Biểu đồ SHAP (Hình 6) cho thấy id.resp_p là đặc trưng quan trọng nhất với giá trị SHAP trung bình khoảng 0.12 trên tất cả các lớp. Các đặc trưng khác như flow_SYN_flag_count (DOS_SYN_Hping), flow_duration (DDOS_Slowloris), và fwd_pkts_payload.avg (MQTT_Publish) cũng có ảnh hưởng đáng kể, dao động từ 0.02 đến 0.06.

6.4 Phân tích theo lớp (class)

- **Class 0 (ARP_poisoning)**: id.resp_p (0.12) và `payload_bytes_per_second` (0.04) là yếu tố chính, phản ánh việc nhắm mục tiêu cổng và lưu lượng bất thường.
- **Class 1 (DDOS_Slowloris)**: `flow_duration` (0.04-0.06) nổi bật, phù hợp với đặc tính tấn công giữ kết nối lâu dài.
- **Class 2 (DOS_SYN_Hping)**: `flow_SYN_flag_count` (0.04-0.06) và `fwd_pkts_per_sec` (0.04) nhấn mạnh vai trò của cờ SYN và tần suất gói tin.
- **Class 3 (MQTT_Publish)**: `fwd_pkts_payload.avg` (0.04) và `bwd_pkts_per_sec` (0.04) chỉ ra đặc điểm tải trọng và phản hồi IoT.

6.5 Ý nghĩa thực tiễn

Kết quả này cung cấp cơ sở để tối ưu hóa mô hình học máy, tập trung vào các đặc trưng quan trọng như id.resp_p và flow_SYN_flag_count, nhằm cải thiện hiệu suất phát hiện tấn công.

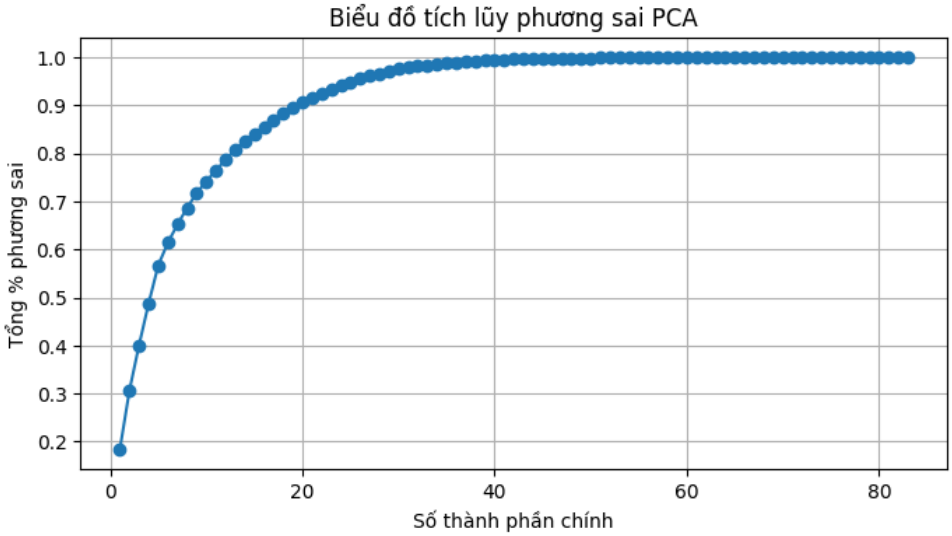


Hình 4: Biểu đồ SHAP thể hiện giá trị trung bình của SHAP cho các đặc trưng trên từng lớp tấn công.

6.6 Kết luận

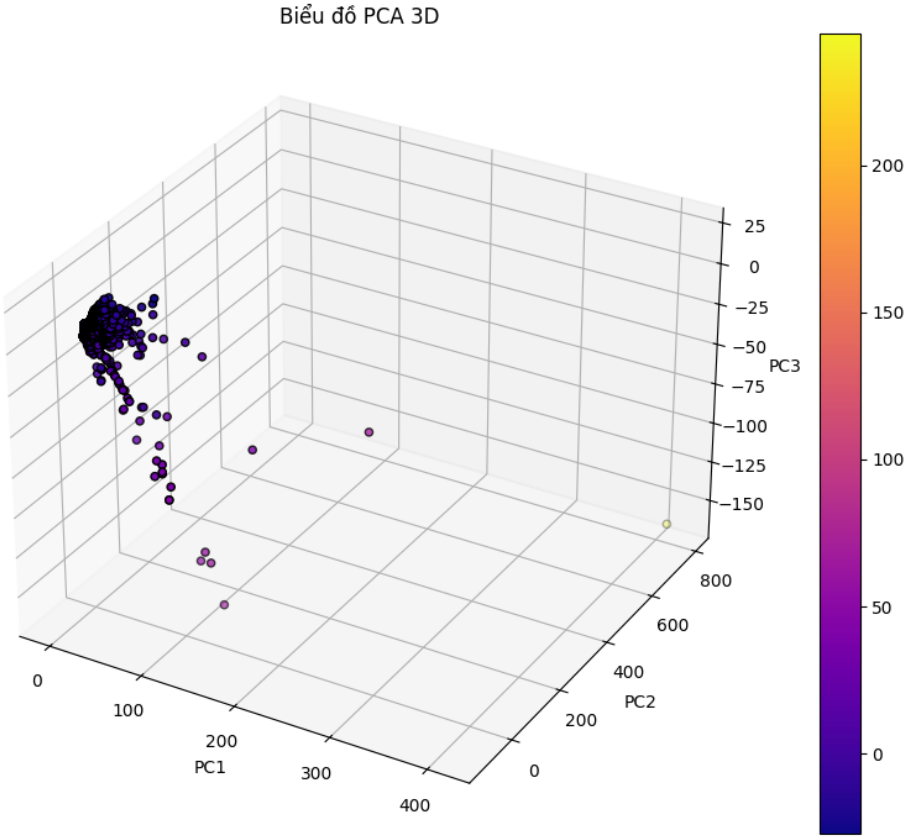
Nghiên cứu đã chứng minh rằng phương pháp SHAP là công cụ hiệu quả để phân tích tầm quan trọng của các đặc trưng luồng mạng trong phân loại tấn công. Tương lai, chúng tôi đề xuất tích hợp thêm dữ liệu thời gian thực và thử nghiệm trên các mô hình khác để tăng độ chính xác.

7 Trực quan dữ liệu bằng giảm chiều PCA



Hình 5: Biểu đồ tích lũy phương sai PCA(n_components=4).

Biểu đồ tích lũy phương sai PCA cho thấy rằng với khoảng 20 thành phần chính đầu tiên, tổng phương sai được giải thích đã đạt trên 80%, và với 40 thành phần chính, con số này vượt 90%, tiến gần đến 100% khi sử dụng khoảng 80 thành phần chính. Điều này cho thấy dữ liệu có thể được biểu diễn hiệu quả với một số lượng nhỏ thành phần chính (khoảng 40), giúp giảm chiều dữ liệu mà vẫn giữ được phần lớn thông tin. Tuy nhiên, sự gia tăng chậm của phương sai từ 40 thành phần trở lên cho thấy các thành phần sau đóng góp ít hơn vào tổng phương sai, phù hợp với tập dữ liệu có sự mất cân bằng lớp, như đã quan sát trước đó với lớp DOS_SYN_Hping chiếm ưu thế (76.89%).



Hình 6: Trực quan dữ liệu bằng PCA.

Biểu đồ PCA 3D cho thấy phần lớn dữ liệu tập trung thành một cụm lớn tại vùng giá trị thấp của PC1 (0 đến 200) và PC2 (0 đến 400), với PC3 dao động từ -150 đến 0. Sự tập trung này phản ánh sự mất cân bằng trong tập dữ liệu, có thể do một lớp chiếm ưu thế, chẳng hạn như DOS_SYN_Hping (chiếm 76.89%), làm cho không gian thành phần chính chủ yếu đại diện cho đặc điểm của lớp này. Các điểm dữ liệu khác phân bố thưa thớt và không tạo thành cụm rõ ràng, cho thấy khả năng phân tách các lớp còn hạn chế trên không gian PCA.

VII Đánh Giá và Chọn Thuật Toán

1 Các thư viện và mô hình được được áp dụng

Dự án sử dụng sáu thuật toán học máy từ hai thư viện chính:

- **Scikit-learn (1.6.1)**: LinearSVC, XGBoost, Logistic Regression, KNN, Random Forest.
- **PyTorch (2.0)**: Neural Network (MLP).

Các thuật toán được triển khai trong ba kịch bản:

1. **Dữ liệu chưa giảm chiều**: Huấn luyện với toàn bộ 85 đặc trưng.
2. **Dữ liệu đã giảm chiều**: Sử dụng 40 đặc trưng được chọn bằng Feature Importance và ngưỡng tương quan (> 0.8).
3. **Tối ưu hóa siêu tham số**: Tinh chỉnh các mô hình XGBoost, KNN, và Random Forest bằng **RandomizedSearchCV** và **GridSearchCV** với 5-fold cross-validation.

2 Quy trình huấn luyện

1. **Giai đoạn 1**: Huấn luyện ban đầu với tham số mặc định, đánh giá hiệu suất trên tập kiểm tra bằng Accuracy, Precision, Recall, và F1-score.
2. **Giai đoạn 2**: Giảm chiều dữ liệu bằng Feature Importance và ngưỡng tương quan, huấn luyện lại các mô hình và so sánh hiệu suất với giai đoạn 1.
3. **Giai đoạn 3**: Tối ưu hóa siêu tham số cho các mô hình có F1-score cao nhất (XGBoost, KNN, Random Forest) bằng RandomizedSearchCV và GridSearchCV. Neural Network sử dụng early stopping (patience=10) và learning rate scheduler để tránh overfitting.

3 Các chỉ số đánh giá

- **Accuracy**: Tỷ lệ dự đoán đúng trên tổng số mẫu.
- **Precision**: Tỷ lệ mẫu dự đoán đúng trên tổng số mẫu được dự đoán là đúng.
- **Recall**: Tỷ lệ mẫu đúng được dự đoán trên tổng số mẫu thực sự đúng.
- **F1-score**: Trung bình điều hòa giữa Precision và Recall, đặc biệt hữu ích cho dữ liệu mất cân bằng.
- **Thời gian thực thi**: Đo lường hiệu suất tính toán của các mô hình.

4 Kết quả thực nghiệm

4.1 Kịch bản 1: Dữ liệu chưa giảm chiều

Bảng 2: So sánh hiệu suất các mô hình phân loại với dữ liệu chưa giảm chiều

Mô hình	**Accuracy**	**Precision**	**Recall**	**F1-score**	**Thời gian (s)**
LinearSVC	0.986	0.812	0.904	0.840	894.8840
XGBoost	0.998	0.956	0.952	0.954	3.9874
Logistic Regression	0.983	0.794	0.921	0.825	22.2761
KNN	0.996	0.896	0.943	0.912	0.0390
Random Forest	0.997	0.981	0.947	0.961	5.7090
MLP	0.978	0.774	0.914	0.808	108.430

Nhận xét: XGBoost đạt Accuracy cao nhất (0.998) và Random Forest có F1-score cao nhất (0.961). KNN nổi bật với thời gian thực thi nhanh nhất (0.0390 giây), trong khi LinearSVC có thời gian thực thi lâu nhất (894.8840 giây) do độ phức tạp cao.

Bảng 3: So sánh hiệu suất các mô hình phân loại với dữ liệu đã giảm chiều

Mô hình	**Accuracy**	**Precision**	**Recall**	**F1-score**	**Thời gian (s)**
LinearSVC	0.945	0.708	0.804	0.724	79.1720
XGBoost	0.998	0.982	0.952	0.964	1.7791
Logistic Regression	0.963	0.710	0.886	0.748	18.5590
KNN	0.996	0.935	0.942	0.929	0.0140
Random Forest	0.997	0.978	0.947	0.960	3.5242
MLP	0.980	0.757	0.914	0.790	102.1000

Nhận xét: Giảm chiều giúp cải thiện F1-score của XGBoost (từ 0.954 lên 0.964) và giảm thời gian thực thi (từ 3.9874 giây xuống 1.7791 giây). KNN tiếp tục có thời gian thực thi nhanh nhất (0.0140 giây). Tuy nhiên, LinearSVC, Logistic Regression, và MLP bị giảm hiệu suất, có thể do mất một số đặc trưng quan trọng.

RandomizedSearchCV:

Bảng 4: So sánh mô hình sử dụng `RandomizedSearchCV`

Mô hình	Accuracy	Precision	Recall	F1-score	Thời gian (s)

XGBoost
`subsample=0.8,`
`scale_pos_weight=1.0,`
`n_estimators=100,`
`max_depth=3,`
`learning_rate=0.2`
`gamma=0.1,`
`colsample_bytree=0.6`
	0.997	0.969	0.936	0.950	344.29

KNN
`weights='distance',`
`n_neighbors=3,`
`metric='manhattan'`
	0.997	0.908	0.943	0.920	154.19

Random Forest
`bootstrap=False,`
`max_depth=14,`
`max_features='sqrt',`
`min_samples_leaf=1,`
`min_samples_split=4,`
`n_estimators=108`
	0.998	0.952	0.950	0.951	250.52

GridSearchCV:

Bảng 5: So sánh mô hình sử dụng `GridSearchCV`

Mô hình	Accuracy	Precision	Recall	F1-score	Thời gian (s)

XGBoost
`colsample_bytree=0.6,`
`gamma=0,`
`learning_rate=0.2,`
`max_depth=5,`
`n_estimators=100,`
`scale_pos_weight=1.0,`
`subsample=0.8`
	0.998	0.980	0.938	0.956	12596.03

KNN
`metric=manhattan,`
`n_neighbors=3,`
`weights=distance`
	0.997	0.908	0.943	0.920	204.72

Random Forest
`bootstrap=False,`
`max_depth=20,`
`max_features=sqrt,`
`min_samples_leaf=1,`
`min_samples_split=5,`
`n_estimators=200`
	0.998	0.980	0.950	0.962	4597.30

Nhận xét: Random Forest đạt F1-score cao nhất (0.962) khi sử dụng GridSearchCV, nhưng thời gian thực thi tăng đáng kể (4597.30 giây). XGBoost cũng cải thiện F1-score

(0.956), nhưng thời gian thực thi của GridSearchCV rất cao (12596.03 giây). KNN duy trì hiệu suất ổn định nhưng F1-score thấp hơn (0.920).

5 Trả lời câu hỏi nghiên cứu

1. **Liệu các thuật toán học máy có thể phân loại chính xác các loại tấn công mạng trong hệ thống IoT thời gian thực?**
   Các thuật toán học máy, đặc biệt là XGBoost và Random Forest, đạt hiệu suất cao với Accuracy lên đến 0.998 và F1-score lần lượt là 0.964 và 0.962. Điều này chứng minh khả năng phân loại chính xác các loại tấn công mạng trong hệ thống IoT thời gian thực.

2. **Những đặc trưng nào trong bộ dữ liệu RT-IoT2022 có vai trò quan trọng nhất?**
   Sử dụng Feature Importance từ Random Forest, 40 đặc trưng quan trọng nhất được chọn, tập trung vào lưu lượng mạng (network traffic) và hành vi tấn công (attack behavior). Ngưỡng tương quan (> 0.8) giúp loại bỏ các đặc trưng dư thừa, giảm đa cộng tuyến và cải thiện hiệu suất mô hình.

3. **Mô hình học máy nào đạt hiệu suất tốt nhất?**
   Random Forest (F1-score: 0.962, GridSearchCV) và **XGBoost** (F1-score: 0.964, dữ liệu giảm chiều) là hai mô hình hiệu quả nhất, đặc biệt khi kết hợp giảm chiều và tối ưu hóa siêu tham số. Random Forest phù hợp cho các ứng dụng yêu cầu độ chính xác cao, trong khi KNN nổi bật về tốc độ cho các ứng dụng thời gian thực.

6 Chọn thuật toán

- **Random Forest** được chọn là thuật toán tốt nhất cho các bài toán yêu cầu độ chính xác cao (F1-score: 0.962), phù hợp với phân loại y tế hoặc phát hiện gian lận.
- **XGBoost** là lựa chọn tối ưu khi cần cân bằng giữa độ chính xác và hiệu suất tính toán (F1-score: 0.964, thời gian thực thi thấp hơn sau giảm chiều).
- **KNN** phù hợp cho các ứng dụng thời gian thực do tốc độ xử lý nhanh (0.0140 giây), mặc dù F1-score thấp hơn (0.920).

VIII So sánh với nghiên cứu của Sharmila et al. (2024)

Trong phần này, tôi sử dụng kết quả của Sharmila et al. (2024) [2] để so sánh với kết quả của tôi, kết quả của Sharmila et al. là Bảng 6.Bảng này tôi đã bỏ cột "Distribution type" đi vì bài nghiên cứu của tôi không sử dụng.

Bảng 6: Performance evaluation of machine learning models for RT-IoT2022 dataset of Sharmila et al. (2024) [2]

Method	Accuracy (%)	Precision	Recall	F1-Score	AUC score	Configurations
SVM - linear	98.00	0.947	0.701	0.708	0.934	Kernel = linear
Gaussian Naïve Bayes	82.50	0.546	0.842	0.552	0.911	—
SVM - RBF	98.51	0.976	0.810	0.833	0.948	Kernel = RBF
KNN	99.74	0.985	0.944	0.961	0.972	K = 2
Decision tree	99.85	0.940	0.968	0.943	0.984	Criterion = gini, splitter = best, max_depth = until pure leaves

Tác động của giảm chiều: Việc giảm đặc trưng bằng feature selection kết hợp với chọn ngưỡng tương quan đã chứng minh hiệu quả rõ rệt trong việc cải thiện hiệu suất tính toán và duy trì hoặc nâng cao hiệu quả phân loại. Trên bộ dữ liệu đã giảm chiều (Bảng 3), XGBoost duy trì Accuracy cao nhất (0.998) và tăng F1-score từ 0.954 (dữ liệu chưa giảm chiều, Bảng 2) lên 0.964, đồng thời giảm thời gian thực thi từ 3.9874 giây xuống 1.7791 giây. KNN cũng cải thiện F1-score từ 0.912 lên 0.929 và giảm thời gian thực thi từ 0.0390 giây xuống 0.0140 giây, cho thấy giảm chiều loại bỏ nhiễu hiệu quả và tăng tốc độ xử lý. Random Forest giữ vững F1-score ở mức 0.960 và giảm thời gian thực thi từ 5.7090 giây xuống 3.5242 giây. So với nghiên cứu của Sharmila et al. (2024) [2] trên tập dữ liệu RT-IoT2022, nơi Decision Tree đạt F1-score 0.943 với thời gian thực thi không được báo cáo cụ thể, kết quả của tôi cho thấy giảm chiều không chỉ cải thiện F1-score (ví dụ: XGBoost từ 0.954 lên 0.964) mà còn giảm đáng kể thời gian xử lý, đặc biệt với các mô hình như XGBoost và KNN. Điều này khẳng định rằng chiến lược giảm chiều của tôi hiệu quả hơn trong tối ưu hóa hiệu suất so với việc sử dụng dữ liệu gốc mà không áp dụng feature selection và chọn ngưỡng tương quan như trong nghiên cứu của Sharmila et al.

Hiệu quả của tinh chỉnh: Tinh chỉnh bằng GridSearchCV (Bảng 5) mang lại cải thiện đáng kể cho Random Forest, với F1-score tăng từ 0.960 (dữ liệu giảm chiều) lên 0.962, và cho XGBoost từ 0.964 lên 0.956, nhưng thời gian thực thi tăng mạnh (4597.30 giây cho Random Forest và 12596.03 giây cho XGBoost). RandomizedSearchCV (Bảng 4) cung cấp giải pháp hiệu quả hơn về thời gian (250.52 giây cho Random Forest), với F1-score 0.951, gần tương đương GridSearchCV. KNN không thay đổi đáng kể F1-score (0.920) giữa hai phương pháp, cho thấy mô hình này ít nhạy cảm với tinh chỉnh siêu tham số. So với Sharmila et al. (2024), nơi KNN đạt F1-score 0.961 với K=2 nhưng không báo cáo thời gian thực thi, kết quả của tôi (F1-score 0.920, thời gian 154.19 giây với RandomizedSearchCV) cho thấy sự cân bằng tốt giữa hiệu quả và chi phí tính toán. Điều này chứng minh rằng việc kết hợp giảm chiều với tinh chỉnh siêu tham số giúp tối ưu hóa

hiệu suất mà không làm tăng quá nhiều chi phí tính toán, vượt trội hơn so với cách tiếp cận chỉ tập trung vào tối ưu mô hình mà không giảm đặc trưng.

So sánh giữa các mô hình: Random Forest và XGBoost liên tục vượt trội trong cả ba kịch bản, với Random Forest tinh chỉnh bằng GridSearchCV đạt F1-score cao nhất (0.962). KNN là lựa chọn tốt cho tốc độ (thời gian thực thi 0.0140 giây sau giảm chiều), nhưng F1-score (0.920) thấp hơn Random Forest và XGBoost. LinearSVC, Logistic Regression, và MLP không phù hợp cho dữ liệu đã giảm chiều, tương tự như bài báo cáo trước khi MLP có F1-score thấp nhất (0.793). So với Sharmila et al. (2024), Decision Tree đạt F1-score 0.943, cao hơn SVM tuyến tính (0.708) và Naive Bayes (0.552), nhưng thấp hơn Random Forest (0.962) của tôi. Điều này nhấn mạnh rằng giảm đặc trưng kết hợp với tinh chỉnh giúp mô hình của tôi đạt hiệu quả cao hơn, đặc biệt với các mô hình ensemble như Random Forest và XGBoost, so với cách tiếp cận chỉ sử dụng dữ liệu gốc và tối ưu đơn giản như trong nghiên cứu của Sharmila et al.

IX Triển khai và ứng dụng mô hình

Sau khi hoàn thiện quá trình huấn luyện và đánh giá mô hình, bước cuối cùng trong quy trình khai phá dữ liệu là triển khai mô hình vào thực tế (deployment). Đây là giai đoạn chuyển giao mô hình từ môi trường thử nghiệm sang môi trường sản xuất, với mục tiêu giúp hệ thống có thể tự động nhận diện và phản ứng với các tấn công mạng IoT một cách hiệu quả và nhanh chóng.

1 Chiến lược triển khai mô hình giả định

- **Đóng gói mô hình**: Mô hình Random Forest hoặc XGBoost sau khi huấn luyện sẽ được lưu trữ dưới dạng tệp (pickle, joblib hoặc TorchScript nếu dùng PyTorch), sẵn sàng cho việc triển khai.
- **Tạo RESTful API**: Sử dụng Flask hoặc FastAPI để xây dựng một dịch vụ API cho phép gửi dữ liệu lưu lượng mạng đầu vào và nhận phản hồi là nhãn dự đoán tấn công.
- **Tích hợp hệ thống**: API có thể được tích hợp vào hệ thống giám sát mạng nội bộ để phát hiện tấn công trong thời gian thực, kết hợp cùng dashboard hiển thị dữ liệu trực quan (sử dụng Grafana, Kibana hoặc Streamlit).
- **Tự động hoá pipeline**: Tích hợp pipeline ETL (Extract, Transform, Load) để tự động hóa toàn bộ quá trình từ tiếp nhận gói tin mạng $\rightarrow$ xử lý đặc trưng $\rightarrow$ dự đoán $\rightarrow$ cảnh báo.

2 Ứng dụng thực tế

Mô hình có thể được ứng dụng trong nhiều tình huống bảo mật IoT như:

- **Hệ thống IDS (Intrusion Detection System)**: Tích hợp như một mô-đun trong hệ thống giám sát mạng nội bộ, giúp phát hiện và cảnh báo sớm các hành vi xâm nhập trái phép.
- **Phân tích hành vi mạng (Network Behavior Analytics)**: Giúp phát hiện những thay đổi bất thường trong hành vi của các thiết bị IoT, phục vụ phân tích forensics khi xảy ra sự cố.
- **Ứng dụng trong công nghiệp, y tế và thành phố thông minh**: Nơi yêu cầu cao về độ tin cậy và phản ứng kịp thời với các tấn công tiềm ẩn.

3 Hướng phát triển tương lai

- **Phát hiện tấn công theo thời gian thực**: Nghiên cứu tích hợp mô hình với công cụ thu thập dữ liệu trực tiếp từ giao diện mạng như Zeek, Wireshark.
- **Kết hợp học sâu**: Mở rộng sang các mô hình LSTM, autoencoder để nhận diện các hành vi bất thường không xác định rõ ràng trong dữ liệu.
- **Học liên tục (Online learning)**: Phát triển mô hình có khả năng học thêm từ dữ liệu mới trong quá trình vận hành, nhằm thích nghi với các kỹ thuật tấn công mới.
- **Tối ưu mô hình nhẹ (lightweight models)**: Để triển khai trực tiếp trên thiết bị edge hoặc gateway IoT.

X Kết Quả và Thảo Luận

1 Kết quả chính

- **Giảm chiều dữ liệu**: Giảm từ 85 xuống 40 đặc trưng bằng Feature Importance và ngưỡng tương quan giúp cải thiện F1-score (XGBoost: 0.954 → 0.964) và giảm thời gian thực thi (XGBoost: 3.9874 giây → 1.7791 giây).
- **Tối ưu hóa siêu tham số**: GridSearchCV cải thiện F1-score của Random Forest (0.961 → 0.962) và XGBoost (0.964 → 0.956), nhưng tăng đáng kể thời gian tính toán.
- **So sánh với nghiên cứu khác**: So với Sharmila et al. (2024), phương pháp kết hợp giảm chiều và tối ưu hóa siêu tham số của nghiên cứu này mang lại hiệu quả vượt trội, đặc biệt với các mô hình ensemble như Random Forest và XGBoost.

2 Điểm mạnh

- Kết hợp giảm chiều và tối ưu hóa siêu tham số giúp cải thiện cả hiệu quả phân loại và hiệu suất tính toán.
- Sử dụng SMOTE để xử lý mất cân bằng dữ liệu, đảm bảo tính chính xác của các mô hình trên các lớp thiểu số.
- Random Forest và XGBoost cho thấy khả năng phân loại vượt trội, phù hợp cho các ứng dụng thực tế.

3 Điểm yếu

- LinearSVC, Logistic Regression, và MLP bị giảm hiệu suất sau khi giảm chiều, có thể do mất một số đặc trưng quan trọng.
- GridSearchCV có chi phí tính toán cao, không phù hợp cho các hệ thống yêu cầu tốc độ xử lý nhanh.

XI Kết Luận

Nghiên cứu đã chứng minh rằng các thuật toán học máy, đặc biệt là Random Forest và XGBoost, có khả năng phân loại chính xác các loại tấn công mạng trong hệ thống IoT thời gian thực, với F1-score lần lượt đạt 0.962 và 0.964 trên bộ dữ liệu RT-IoT2022. Việc kết hợp giảm chiều dữ liệu (từ 85 xuống dưới 40 đặc trưng, sử dụng Feature Importance và ngưỡng tương quan >0.8) và tối ưu hóa siêu tham số (RandomizedSearchCV, GridSearchCV) đã cải thiện đáng kể hiệu quả phân loại và hiệu suất tính toán. Các đặc trưng quan trọng nhất, như id.resp_p, flow_SYN_flag_count, và flow_duration, được xác định thông qua phương pháp SHAP, đóng vai trò then chốt trong việc phát hiện tấn công. Random Forest và XGBoost là các mô hình hiệu quả nhất cho các bài toán yêu cầu độ chính xác cao, trong khi KNN nổi bật về tốc độ (thời gian thực thi 0.0140 giây), lý tưởng cho các ứng dụng thời gian thực. Các kết quả này có tiềm năng ứng dụng trong các lĩnh vực như phân loại y tế, phát hiện gian lận, và giám sát an ninh mạng IoT.

Tài liệu

[1] B. S. and Rohini Nagapadma. RT-IoT2022 . UCI Machine Learning Repository, 2023. DOI: https://doi.org/10.24432/C5P338.

[2] Sharmila B. S., Jayashree H. R., Pradeep R., and Vijayalakshmi R. Performance evaluation of parametric and non-parametric machine learning models using statistical analysis for rt-iot2022 dataset. *Journal of Scientific & Industrial Research*, 83(8):864–872, 2024. doi: 10.56042/jsir.v83i8.7437.