|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **BỘ GIÁO DỤC VÀ ĐÀO TẠO TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH**  **BỘ MÔN CÔNG NGHỆ THÔNG TIN**  🙢🙢🙢🕮🙠🙠🙠  Description: http://ined.utc.edu.vn/sites/ined.utc.edu.vn/files/styles/medium/public/logo.png?itok=JEfoqp8q  **BÁO CÁO THỰC TẬP TỐT NGHIỆP**  **ĐỀ TÀI: ỨNG DỤNG FEDERATED LEARNING PHÁT HIỆN TẤN CÔNG MẠNG TRONG HỆ THỐNG IOT.**   |  |  |  | | --- | --- | --- | | Giảng viên hướng dẫn | : | ThS.PHAN THANH HY | | Sinh viên thực hiện | : | NGUYỄN ĐỨC THẮNG | | Lớp | : | CQ.62.CNTT | | Mã số sinh viên | : | 6251071094 |   TP. Hồ Chí Minh, tháng 6 năm 2024  **BỘ GIÁO DỤC VÀ ĐÀO TẠO TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH**  **BỘ MÔN CÔNG NGHỆ THÔNG TIN**  🙢🙢🙢🕮🙠🙠🙠  Description: http://ined.utc.edu.vn/sites/ined.utc.edu.vn/files/styles/medium/public/logo.png?itok=JEfoqp8q  **BÁO CÁO THỰC TẬP TỐT NGHIỆP**  **ĐỀ TÀI: ỨNG DỤNG FEDERATED LEARNING PHÁT HIỆN TẤN CÔNG MẠNG TRONG HỆ THỐNG IOT.**   |  |  |  | | --- | --- | --- | | Giảng viên hướng dẫn | : | ThS. PHAN THANH HY | | Sinh viên thực hiện | : | NGUYỄN ĐỨC THẮNG | | Lớp | : | CQ.62.CNTT | | Mã số sinh viên | : | 6251071094 |   TP. Hồ Chí Minh, tháng 6 năm 2024 |

|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM** |
| **PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH** | Độc lập – Tự do – Hạnh phúc |

# NHIỆM VỤ BÁO CÁO THỰC TẬP TỐT NGHIỆP

BỘ MÔN: **CÔNG NGHỆ THÔNG TIN**

-------\*\*\*-------

|  |  |
| --- | --- |
| **Mã sinh viên**: 6251071094 | **Họ tên SV:** Nguyễn Đức Thắng |
| **Khóa:** 62 | **Lớp:** CQ.62.CNTT |

**1. Tên đề tài.**

ĐỀ TÀI: ỨNG DỤNG FEDERATED LEARNING PHÁT HIỆN TẤN CÔNG MẠNG TRONG HỆ THỐNG IOT.

**2. Mục đích, yêu cầu.**

a. **Mục đích.**

Mục tiêu của đề tài là nghiên cứu, thiết kế và triển khai một hệ thống phát hiện tấn công mạng dành cho môi trường Internet of Things (IoT) trên nền tảng học liên kết (Federated Learning). Phương pháp này cho phép huấn luyện mô hình học máy tại các thiết bị đầu cuối mà không cần truyền dữ liệu cục bộ về máy chủ trung tâm, qua đó đảm bảo tính bảo mật và riêng tư của dữ liệu người dùng.

Thông qua việc ứng dụng Federated Learning, đề tài hướng tới giải quyết bài toán phân tán trong phát hiện tấn công mạng, đặc biệt trong bối cảnh các thiết bị IoT thường bị giới hạn về tài nguyên và dễ trở thành mục tiêu của các cuộc tấn công mạng. Đồng thời, đề tài cũng đặt mục tiêu đánh giá hiệu quả, độ chính xác và khả năng thích ứng của mô hình trong các kịch bản tấn công mạng phổ biến, từ đó chứng minh tính khả thi và tiềm năng triển khai thực tế của giải pháp đề xuất trong môi trường IoT hiện đại.

b. **Yêu cầu.**

Để hiện thực hóa mục tiêu xây dựng một hệ thống học liên kết hiệu quả trong lĩnh vực an ninh mạng, đặc biệt là phát hiện các hình thức tấn công mạng phổ biến như DDoS và Brute Force, đồ án này đặt ra những yêu cầu cụ thể về kiến thức và kỹ năng. Người thực hiện cần nắm vững cơ chế hoạt động và đặc trưng hành vi của các hình thức tấn công này, đồng thời hiểu rõ bản chất của Federated Learning, từ quy trình huấn luyện mô hình phân tán đến cơ chế tổng hợp trọng số toàn cục và cách đánh giá hiệu quả mô hình trong môi trường phi tập trung.

Về mặt kỹ thuật, đồ án đòi hỏi khả năng xử lý dữ liệu từ các bộ tập dữ liệu thực tế, bao gồm cả việc tiền xử lý dữ liệu sao cho phù hợp với các thuật toán học máy. Bên cạnh đó, việc triển khai các mô hình học máy cơ bản không chỉ giúp làm quen với quy trình làm việc mà còn tạo cơ sở để so sánh hiệu quả với mô hình học liên kết được xây dựng. Một yếu tố quan trọng khác là xây dựng hệ thống mô phỏng, cho phép nhiều thiết bị IoT tham gia vào quá trình huấn luyện phân tán, từ đó tạo ra một môi trường thử nghiệm gần với thực tế.

Để đảm bảo tính khách quan và khoa học của kết quả, đồ án cũng yêu cầu thiết kế một quy trình đánh giá chính xác, dựa trên các chỉ số như Accuracy, Precision, Recall, và F1-score. Quy trình này không chỉ giúp đánh giá hiệu quả của mô hình học liên kết mà còn tạo nền tảng vững chắc cho việc mở rộng và phát triển các nghiên cứu chuyên sâu hơn trong tương lai. Sự kết hợp giữa kiến thức chuyên môn sâu rộng và kỹ năng thực hành thành thạo sẽ là chìa khóa để đạt được thành công trong đồ án này, góp phần vào việc nâng cao khả năng phòng thủ và bảo vệ hệ thống mạng trước các mối đe dọa ngày càng phức tạp.

**3. Nội dung và phạm vi đề tài**

Nội dung của đề tài bắt đầu từ việc khảo sát các hình thức tấn công mạng phổ biến hiện nay trong hệ thống IoT. Các cuộc tấn công này tuy khác nhau về bản chất kỹ thuật nhưng đều gây ảnh hưởng nghiêm trọng đến tính khả dụng và độ an toàn của thiết bị và dữ liệu người dùng. Từ việc phân tích các đặc trưng tấn công, đề tài tiến hành khảo sát các phương pháp học máy được ứng dụng trong phát hiện tấn công, từ đó chỉ ra giới hạn của phương pháp tập trung và lý do nên ứng dụng Federated Learning.

Đề tài triển khai mô hình học liên kết trong môi trường mô phỏng nhiều client (tương ứng với các thiết bị IoT), nơi mỗi client thực hiện huấn luyện mô hình cục bộ dựa trên dữ liệu nội bộ, và chia sẻ trọng số lên server trung tâm để tổng hợp mô hình toàn cục. Quy trình này giúp hạn chế việc truyền dữ liệu gốc, giảm nguy cơ bị đánh cắp hoặc rò rỉ thông tin nhạy cảm. Phạm vi nghiên cứu chủ yếu tập trung trong môi trường mô phỏng với số lượng client cố định, chưa triển khai trên các thiết bị IoT vật lý ngoài thực tế, đồng thời không đi sâu vào các hình thức tấn công khác như tấn công mã độc hoặc tấn công nâng cao có chủ đích. Tuy nhiên, mô hình thiết kế sẽ được xây dựng theo hướng mở, có thể mở rộng về sau nếu cần thiết.

**4. Công nghệ, công cụ và ngôn ngữ lập trình.**

Ngôn ngữ lập trình sử dụng chính trong đề tài là Python vì tính linh hoạt, cú pháp dễ đọc và đặc biệt là hệ sinh thái thư viện hỗ trợ mạnh mẽ cho các bài toán học máy. Các thư viện học máy như scikit-learn sẽ được sử dụng để xây dựng mô hình ban đầu và đánh giá hiệu năng, trong khi TensorFlow Federated hoặc PySyft được dùng để hiện thực hoá mô hình Federated Learning.

Các công cụ như pandas, NumPy sẽ hỗ trợ xử lý dữ liệu và trực quan hóa kết quả bằng matplotlib hoặc seaborn. Môi trường lập trình chính có thể sử dụng Jupyter Notebook hoặc Google Colab để thuận tiện theo dõi và kiểm thử mô hình. Bên cạnh đó, trong quá trình mô phỏng các thiết bị IoT, Docker có thể được sử dụng để tạo các vùng thực thi độc lập, hoặc MQTT có thể được tích hợp để mô phỏng luồng dữ liệu theo thời gian thực giữa các client và server trung tâm. Trong trường hợp mở rộng yêu cầu mô phỏng mạng ở mức sâu hơn, Mininet có thể được sử dụng để kiểm soát chi tiết các node và kênh truyền. Ngoài ra, công cụ lưu trữ mô hình có thể sử dụng SQLite hoặc tệp lưu trữ thông thường tuỳ theo quy mô đề tài.

**5. Các kết quả dự kiến sẽ đạt được.**

Kết quả mong đợi của đề tài là xây dựng thành công một hệ thống phát hiện tấn công mạng hoạt động trên nền tảng Federated Learning, có khả năng phát hiện chính xác các dấu hiệu của tấn công DDoS và Brute Force mà không cần thu thập toàn bộ dữ liệu về một nơi duy nhất. Hệ thống phải chứng minh được tính khả thi trong môi trường phân tán với độ chính xác mô hình đạt tối thiểu trên 90%, đồng thời đảm bảo hiệu suất huấn luyện không bị ảnh hưởng quá nhiều so với mô hình tập trung. Ngoài ra, đề tài kỳ vọng đưa ra phân tích toàn diện về ưu nhược điểm của Federated Learning trong bối cảnh ứng dụng vào an ninh mạng, làm rõ các yếu tố ảnh hưởng đến hiệu quả mô hình như số lượng client, số vòng huấn luyện toàn cục (global rounds), kích thước mô hình, chất lượng dữ liệu tại từng thiết bị,... Kết quả thu được sẽ không chỉ được trình bày dưới dạng bảng và biểu đồ mà còn được phân tích chi tiết để cung cấp kiến thức nền tảng cho những đề tài nghiên cứu mở rộng về bảo mật trong hệ thống IoT. Cuối cùng, đề tài cũng có thể đóng vai trò như một nền tảng thử nghiệm để mở rộng nghiên cứu sang các mô hình học tăng cường phân tán, học bán giám sát hoặc các phương pháp bảo mật như differential privacy trong học liên kết.

**6. Giảng viên và cán bộ hướng dẫn.**

Họ tên: ThS. Phan Thanh Hy

Đơn vị công tác: Bộ môn Công Nghệ Thông Tin - Trường Đại học Giao thông Vận Tải phân hiệu tại thành phố Hồ Chí Minh

Điện thoại: Email:

**Ngày tháng 06 năm 2025** **Đã giao nhiệm vụ BTL**

Trưởng BM Công nghệ Thông tin Giảng viên hướng dẫn

**ThS. Trần Phong Nhã** **ThS. Phan Thanh Hy**

Đã nhận nhiệm vụ TKTN

|  |  |
| --- | --- |
| Sinh viên: Nguyễn Đức Thắng | Ký tên: Thắng |
| Điện thoại:0374572364 | Email: 6251071094@st.utc2.edu.vn |

# LỜI CẢM ƠN

Lời đầu tiên, chúng em xin kính gửi đến toàn thể quý thầy cô Bộ môn Công nghệ thông tin - Phân hiệu Trường Đại học Giao thông vận tải lời tri ân sâu sắc nhất. Kính ơn thầy cô đã tận tâm truyền thụ tri thức, chắp cánh cho những ước mơ của chúng em trên hành trình khám phá thế giới công nghệ thông tin đầy tiềm năng. Tại mái nhà chung này, chúng em không chỉ được lĩnh hội những kiến thức uyên thâm, mà còn được học làm người, rèn luyện kỹ năng, và chuẩn bị hành trang vững chắc cho tương lai.

Trong đó, chúng em xin được bày tỏ lòng biết ơn vô bờ bến đến thầy Phan Thanh Hy, người đã luôn đồng hành, dìu dắt và truyền lửa cho chúng em trong suốt quá trình thực hiện khóa luận tốt nghiệp. Thầy đã dành trọn tâm huyết, không quản ngại khó khăn, tận tình hướng dẫn, khai sáng tri thức, giúp chúng em vững tin trên con đường chinh phục tri thức. Những ân tình cao đẹp này, chúng em xin khắc ghi mãi trong tim.

Trên hành trình học tập và nghiên cứu, chúng em đã nỗ lực hết mình để hoàn thành khóa luận một cách tốt nhất. Tuy nhiên, chúng em hiểu rằng, trên con đường trưởng thành, khó tránh khỏi những thiếu sót và sai lầm. Kính mong quý thầy cô, đặc biệt là thầy Phan Thanh Hy, rộng lòng chỉ bảo, cho chúng em những ý kiến đóng góp chân thành để khóa luận được hoàn thiện hơn.

Cuối cùng, xin kính chúc thầy Phan Thanh Hy và toàn thể quý thầy cô Bộ môn Công nghệ thông tin luôn dồi dào sức khỏe, an lạc, hạnh phúc và gặt hái được nhiều thành công hơn nữa trên sự nghiệp trồng người cao quý.

Chúng em xin chân thành cảm ơn!

# NHẬN XÉT GIÁO VIÊN

.................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................. .............................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................

.................................................................................................................................................................................................................................................................................................................................................................................................................................

*Tp. Hồ Chí Minh, ngày … tháng … năm…..*

Giáo viên hướng dẫn

ThS.Phan Thanh Hy

# MỤC LỤC

# DANH MỤC HÌNH ẢNH

# DANH MỤC BẢNG BIỂU

# DANH MỤC CHỮ VIẾT TẮT

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Mô tả | Ý nghĩa | Ghi chú |
| 1 | IoT | Internet of Things |  |
| 2 | DDoS |  |  |
| 3 | FL |  |  |
| 4 | MQTT |  |  |
| 5 | MITM |  |  |
| 6 | IP |  |  |
| 7 | MAC |  |  |
| 8 | ARP Spoofing |  |  |
| 9 | DNS Spoofing |  |  |
| 10 | HTTPS |  |  |
| 11 | SQLi |  |  |
| 12 | URL |  |  |
| 13 | WAF |  |  |
| 14 | ML |  |  |
| 15 | IDS |  |  |
| 16 | UBA |  |  |
| 17 | DBSCAN |  |  |
| 18 | NFDLM |  |  |
| 19 | SMOTE |  |  |
| 20 | EEFO |  |  |
| 21 | CSV |  |  |
| 22 |  |  |  |

# CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

## 1.1. Bối cảnh nghiên cứu

Trong bối cảnh phát triển nhanh chóng của Internet of Things (IoT), số lượng thiết bị kết nối vào mạng ngày càng tăng, từ các thiết bị gia dụng thông minh đến các hệ thống công nghiệp phức tạp. IoT không chỉ mang lại nhiều tiện ích và cải tiến trong hiệu quả hoạt động, mà còn tạo ra các thách thức nghiêm trọng về an ninh mạng. Các thiết bị IoT thường có cấu hình phần cứng và phần mềm đơn giản, với khả năng bảo mật hạn chế, khiến chúng trở thành mục tiêu dễ dàng cho các cuộc tấn công mạng. Trong số đó, tấn công từ chối dịch vụ phân tán (DDoS) và tấn công vét cạn (Brute Force) là những mối đe dọa phổ biến nhất.

Cuộc tấn công DDoS thường được thực hiện bằng cách sử dụng một mạng lưới lớn các thiết bị bị kiểm soát (botnet) để gửi lưu lượng truy cập lớn đến một mục tiêu cụ thể, gây ra tình trạng quá tải và làm giảm khả năng phục vụ người dùng hợp pháp. Tương tự, tấn công Brute Force xảy ra khi kẻ tấn công cố gắng đoán mật khẩu hoặc thông tin đăng nhập của người dùng bằng cách thử tất cả các khả năng có thể, dẫn đến việc xâm nhập hệ thống. Hậu quả của những cuộc tấn công này không chỉ dừng lại ở việc gián đoạn dịch vụ, mà còn đe dọa đến việc bảo mật thông tin cá nhân và danh tiếng của tổ chức.

Do sự phát triển không ngừng của các cuộc tấn công mạng, việc phát hiện và ngăn chặn kịp thời các mối đe dọa này trở nên cực kỳ quan trọng. Tuy nhiên, phương pháp phát hiện truyền thống gặp nhiều khó khăn do bảo mật và quyền riêng tư dữ liệu, đồng thời yêu cầu phải thu thập và phân tích dữ liệu từ nhiều nguồn khác nhau.

## 1.2. Lý do chọn đề tài

Đề tài "Ứng dụng Federated Learning phát hiện tấn công mạng trong hệ thống IoT" đã được lựa chọn cho nghiên cứu này. Đầu tiên, việc bảo đảm an toàn cho hệ thống IoT trong bối cảnh gia tăng nghiêm trọng các mối đe dọa an ninh mạng là một vấn đề không thể coi thường. Phát hiện và phòng ngừa tấn công sẽ giúp giảm thiểu thiệt hại và bảo vệ an toàn cho người dùng.

Thứ hai, Federated Learning (FL) là một phương pháp học máy tiên tiến cho phép các thiết bị IoT tự huấn luyện mô hình học máy dựa trên dữ liệu cục bộ mà không cần gửi dữ liệu về một máy chủ trung tâm. Phương pháp này không chỉ bảo vệ quyền riêng tư dữ liệu mà còn giảm thiểu rủi ro về an ninh do không yêu cầu tập trung dữ liệu, rất phù hợp với cấu hình phần cứng hạn chế của các thiết bị IoT.

Một yếu tố quan trọng khác là việc ứng dụng FL vào phát hiện các tấn công mạng có khả năng mang lại hiệu quả cao hơn so với các phương pháp hiện có. Bằng việc sử dụng mô hình học máy được huấn luyện từ nhiều nguồn dữ liệu đa dạng, FL có thể phát hiện các mẫu tấn công phức tạp mà các phương pháp truyền thống không thể nhận diện.

Đề tài này không chỉ có giá trị nghiên cứu lý thuyết mà còn có khả năng ứng dụng thực tiễn cao trong nhiều lĩnh vực. Việc phát triển các hệ thống phát hiện xâm nhập dựa trên FL có thể giúp bảo vệ nhiều ứng dụng IoT khác nhau, từ dịch vụ gia đình đến các cơ sở hạ tầng thành phố thông minh.

## 1.3. Mục tiêu nghiên cứu

Đề tài "Ứng dụng Federated Learning phát hiện tấn công mạng trong hệ thống IoT" được định hướng bởi một loạt các mục tiêu nghiên cứu cụ thể, nhằm khai thác tiềm năng của Federated Learning (FL) trong việc giải quyết các thách thức an ninh đặc thù của Internet of Things (IoT). Trước hết, một mục tiêu then chốt là tiến hành nghiên cứu tổng quan và sâu rộng về các cuộc tấn công DDoS và Brute Force trong bối cảnh IoT. Mục tiêu này không chỉ dừng lại ở việc mô tả các loại tấn công mà còn đi sâu vào phân tích các đặc điểm, kỹ thuật tấn công phổ biến, và các lỗ hổng bảo mật thường bị khai thác trong các thiết bị và hệ thống IoT. Việc hiểu rõ bản chất của các mối đe dọa này là nền tảng để xây dựng các giải pháp phát hiện và phòng ngừa hiệu quả.

Tiếp theo, đề tài hướng đến việc nghiên cứu và triển khai Federated Learning (FL) trong môi trường IoT, một mục tiêu đòi hỏi việc khám phá các thuật toán, kiến trúc và kỹ thuật tối ưu hóa FL phù hợp với các đặc tính riêng biệt của IoT. Cụ thể, nghiên cứu sẽ tập trung vào việc giải quyết các hạn chế về tài nguyên tính toán, kết nối mạng không ổn định, và sự phân tán dữ liệu trên các thiết bị IoT. Đồng thời, mục tiêu này cũng bao gồm việc xem xét các vấn đề về bảo mật và quyền riêng tư dữ liệu, đảm bảo rằng việc triển khai FL không làm phát sinh các rủi ro mới.

Trên cơ sở các nghiên cứu lý thuyết và triển khai FL, mục tiêu trọng tâm của đề tài là xây dựng một mô hình phát hiện tấn công DDoS và Brute Force dựa trên FL. Mô hình này cần có khả năng phân tích dữ liệu từ các thiết bị IoT, xác định các mẫu tấn công, và đưa ra cảnh báo kịp thời. Quá trình huấn luyện mô hình sẽ được thực hiện theo phương pháp FL, trong đó các thiết bị IoT tham gia vào việc huấn luyện mô hình một cách phân tán, không cần chia sẻ dữ liệu thô. Điều này giúp bảo vệ quyền riêng tư dữ liệu và giảm thiểu các rủi ro liên quan đến việc tập trung dữ liệu.

Để đảm bảo tính khả thi và hiệu quả của mô hình, đề tài cũng đặt ra mục tiêu đánh giá hiệu quả của mô hình phát hiện tấn công. Quá trình đánh giá sẽ dựa trên các tiêu chí khách quan và toàn diện, bao gồm độ chính xác (accuracy), độ tin cậy (reliability), tốc độ phát hiện (detection speed), và khả năng chống lại các cuộc tấn công giả mạo (robustness against adversarial attacks). Kết quả đánh giá sẽ được sử dụng để tinh chỉnh và tối ưu hóa mô hình, đảm bảo rằng nó đáp ứng được các yêu cầu thực tế.

## 1.4. Phạm vi nghiên cứu

Nhằm đạt được các mục tiêu nghiên cứu một cách hiệu quả và có tính khả thi, đề tài "Ứng dụng Federated Learning phát hiện tấn công mạng trong hệ thống IoT" được giới hạn trong một phạm vi nghiên cứu cụ thể. Về đối tượng nghiên cứu, đề tài tập trung vào các thiết bị IoT phổ biến trong các ứng dụng gia đình và công nghiệp, bao gồm nhưng không giới hạn ở các cảm biến, bộ điều khiển, thiết bị giám sát, và các thiết bị thông minh khác. Việc tập trung vào các thiết bị này cho phép nghiên cứu đi sâu vào các đặc điểm và lỗ hổng bảo mật thường gặp, từ đó xây dựng các giải pháp phát hiện tấn công phù hợp.

Về loại tấn công, đề tài tập trung vào hai loại tấn công chính là DDoS và Brute Force. DDoS được chọn vì tính chất nghiêm trọng của nó, có thể làm tê liệt toàn bộ hệ thống IoT. Brute Force được chọn vì đây là một phương pháp tấn công đơn giản nhưng hiệu quả, đặc biệt đối với các thiết bị IoT có mật khẩu yếu hoặc chưa được cấu hình bảo mật đúng cách. Mặc dù các loại tấn công khác cũng có thể được đề cập trong phần tổng quan, nhưng trọng tâm nghiên cứu sẽ đặt vào hai loại tấn công này.

Về môi trường thử nghiệm, nghiên cứu sẽ được tiến hành trong cả môi trường mô phỏng và môi trường thực tế. Môi trường mô phỏng cho phép kiểm soát các biến số và thử nghiệm các kịch bản tấn công khác nhau một cách an toàn. Môi trường thực tế, ngược lại, giúp đánh giá khả năng ứng dụng và hiệu quả của mô hình trong điều kiện thực tế, với các yếu tố như nhiễu mạng, độ trễ, và sự đa dạng của thiết bị.

Về công cụ và nền tảng, đề tài sẽ sử dụng các công cụ và nền tảng phổ biến và được hỗ trợ rộng rãi trong lĩnh vực học máy và IoT. Điều này giúp đảm bảo tính khả thi và khả năng tái sử dụng của kết quả nghiên cứu. Các công cụ và nền tảng dự kiến được sử dụng bao gồm Python (ngôn ngữ lập trình chính), TensorFlow và PyTorch (các thư viện học sâu), MQTT (giao thức truyền thông IoT), và các nền tảng đám mây (như AWS, Azure, Google Cloud) để triển khai và quản lý FL.

Về giới hạn về dữ liệu, nghiên cứu sẽ ưu tiên sử dụng các bộ dữ liệu công khai và dữ liệu mô phỏng để đảm bảo tính minh bạch và khả năng tái tạo của kết quả. Trong trường hợp sử dụng dữ liệu thu thập từ các thiết bị IoT thực tế, các biện pháp bảo mật và quyền riêng tư nghiêm ngặt sẽ được áp dụng, bao gồm ẩn danh hóa dữ liệu, mã hóa, và tuân thủ các quy định pháp luật liên quan.

## 1.5. Phương pháp nghiên cứu

Để đạt được các mục tiêu nghiên cứu và đáp ứng các yêu cầu về phạm vi, đề tài "Ứng dụng Federated Learning phát hiện tấn công mạng trong hệ thống IoT" sẽ sử dụng một tổ hợp các phương pháp nghiên cứu, bao gồm cả nghiên cứu lý thuyết, nghiên cứu thực nghiệm, phương pháp phân tích và so sánh, và phương pháp mô phỏng.

Nghiên cứu lý thuyết sẽ đóng vai trò nền tảng, cung cấp kiến thức tổng quan về các lĩnh vực liên quan. Các tài liệu khoa học, báo cáo kỹ thuật, tiêu chuẩn, và các nguồn thông tin khác sẽ được thu thập, phân tích, và tổng hợp để làm rõ các khái niệm cơ bản về IoT, an ninh mạng, DDoS, Brute Force, và FL. Đồng thời, nghiên cứu lý thuyết cũng sẽ bao gồm việc phân tích hệ thống IoT, xác định các thành phần, chức năng, và các lỗ hổng bảo mật tiềm ẩn. Đặc biệt, việc xây dựng các mô hình toán học và mô hình hóa các cuộc tấn công DDoS và Brute Force là rất quan trọng để hiểu rõ cơ chế hoạt động của các cuộc tấn công này và thiết kế các biện pháp đối phó hiệu quả.

Nghiên cứu thực nghiệm sẽ được tiến hành để kiểm chứng các giả thuyết và đánh giá hiệu quả của các giải pháp đề xuất. Quá trình nghiên cứu thực nghiệm sẽ bao gồm việc thu thập dữ liệu từ các thiết bị IoT trong môi trường mô phỏng và thực tế, xây dựng và huấn luyện mô hình phát hiện tấn công dựa trên FL, đánh giá hiệu quả của mô hình bằng các tiêu chí khách quan, và thử nghiệm mô hình trong các kịch bản tấn công khác nhau.

Phương pháp phân tích và so sánh sẽ được sử dụng để đánh giá hiệu quả của mô hình FL so với các phương pháp phát hiện tấn công truyền thống. Các kỹ thuật phân tích thống kê và khai phá dữ liệu sẽ được áp dụng để tìm ra các đặc trưng của tấn công DDoS và Brute Force, và so sánh hiệu quả của mô hình FL với các phương pháp khác về độ chính xác, độ tin cậy, tốc độ phát hiện, và khả năng chống lại các cuộc tấn công giả mạo.

Phương pháp mô phỏng sẽ được sử dụng để tạo ra các kịch bản tấn công DDoS và Brute Force trong môi trường kiểm soát, cũng như mô phỏng hoạt động của FL trong hệ thống IoT. Các công cụ mô phỏng mạng sẽ được sử dụng để tạo ra lưu lượng mạng, mô phỏng các cuộc tấn công, và đánh giá hiệu quả của mô hình phát hiện tấn công trong các điều kiện khác nhau.

## 1.6. Cấu trúc báo cáo

Để trình bày một cách rõ ràng và logic các kết quả nghiên cứu, báo cáo của đề tài sẽ được cấu trúc thành các chương sau:

Chương 1: Tổng quan đề tài: Chương này sẽ giới thiệu tổng quan về đề tài, bao gồm bối cảnh, lý do chọn đề tài, mục tiêu, phạm vi, phương pháp nghiên cứu, và cấu trúc báo cáo.

Chương 2: Cơ sở lý thuyết: Chương này sẽ cung cấp các kiến thức nền tảng về IoT, an ninh mạng, DDoS, Brute Force, và Federated Learning.

Chương 3: Nghiên cứu về Federated Learning trong IoT: Chương này sẽ đi sâu vào phân tích các thách thức và giải pháp khi triển khai FL trong môi trường IoT, bao gồm các vấn đề về tài nguyên hạn chế, kết nối không ổn định, sự phân tán dữ liệu, và bảo mật.

Chương 4: Xây dựng mô hình phát hiện tấn công DDoS và Brute Force dựa trên Federated Learning: Chương này sẽ trình bày chi tiết về kiến trúc, thuật toán, và quá trình huấn luyện mô hình phát hiện tấn công DDoS và Brute Force dựa trên FL.

Chương 5: Đánh giá và thử nghiệm mô hình: Chương này sẽ trình bày các kết quả đánh giá hiệu quả của mô hình phát hiện tấn công, cũng như các kết quả thử nghiệm trong môi trường thực tế.

Chương 6: Kết luận và hướng phát triển: Chương này sẽ tóm tắt các kết quả đạt được, đánh giá những đóng góp của đề tài, và đề xuất các hướng nghiên cứu tiếp theo.

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## 2.1. An toàn thông tin trong hệ thống mạng

Hệ thống thông tin được cấu thành từ ba bộ phận chính: phần cứng, phần mềm và kết nối. Phân chia này nhằm mục đích hỗ trợ quá trình phân loại cũng như áp dụng các tiêu chuẩn an toàn thông tin một cách dễ dàng và hiệu quả nhất. Thông thường, các thủ tục hoặc chính sách bảo mật được xây dựng để hướng dẫn con người bao gồm quản trị viên, người dùng và người vận hành cách thức sử dụng sản phẩm sao cho vừa đáp ứng yêu cầu công việc, vừa đảm bảo an toàn thông tin cho cả cá nhân lẫn tổ chức.

### 2.1.1. An toàn thông tin là gì?

An toàn thông tin (ATTT) là một lĩnh vực có vai trò then chốt và ngày càng trở nên thiết yếu trong kỷ nguyên số hiện đại, đặc biệt trong bối cảnh hệ thống mạng không ngừng phát triển mạnh mẽ và dần trở thành trung tâm của mọi hoạt động sống, từ nhu cầu cá nhân, kinh doanh cho đến vận hành các tổ chức, doanh nghiệp và cơ quan nhà nước. Sự phụ thuộc ngày càng lớn vào các hệ thống thông tin và công nghệ số khiến việc đảm bảo an toàn cho dữ liệu, hệ thống và các quy trình vận hành trở thành một nhiệm vụ cấp bách và không thể xem nhẹ.

Bản chất của an toàn thông tin không chỉ đơn thuần dừng lại ở việc bảo vệ dữ liệu khỏi các hành vi truy cập trái phép hay thất thoát, mà còn bao hàm một hệ thống toàn diện các biện pháp kỹ thuật, chính sách quản trị, quy trình vận hành và giải pháp công nghệ chuyên sâu, được thiết kế và triển khai nhằm đảm bảo ba yếu tố cốt lõi của thông tin và hệ thống thông tin: tính bảo mật (confidentiality), tính toàn vẹn (integrity) và tính khả dụng (availability).

### 2.1.2. Các yếu tố bảo mật trong hệ thống thông tin



Hình 2.1 Tam giác CIA

#### 2.1.2.1 Tính bí mật

Bí mật là khái niệm dùng để chỉ việc bảo vệ thông tin, đảm bảo rằng dữ liệu không bị tiết lộ cho những đối tượng chưa được xác thực hoặc bị lọt vào những hệ thống không được phép. Trong thực tế, việc duy trì tính bí mật giữ vai trò đặc biệt quan trọng trong môi trường giao dịch số ngày nay. Lấy ví dụ, trong một giao dịch thanh toán tín dụng trực tuyến, thông tin về số thẻ tín dụng sẽ được truyền từ người mua tới người bán, sau đó tiếp tục được chuyển đến nhà cung cấp dịch vụ thẻ tín dụng. Để đảm bảo tính bí mật trong suốt quá trình này, hệ thống sẽ áp dụng các biện pháp bảo vệ như mã hóa dữ liệu trong khi truyền tải, giới hạn phạm vi lưu trữ thông tin nhạy cảm trong các khu vực an toàn như cơ sở dữ liệu, tệp nhật ký (log files), bản sao lưu (backup) hoặc hóa đơn in ra, đồng thời thực hiện kiểm soát nghiêm ngặt quyền truy cập vào những khu vực đó.

Tuy nhiên, nếu thông tin như số thẻ tín dụng bị bên thứ ba không được xác thực, chẳng hạn như một kẻ tấn công mạng hoặc một cá nhân không liên quan tới giao dịch, chiếm đoạt bằng bất kỳ phương thức nào, thì lúc đó tính bí mật của dữ liệu coi như đã bị phá vỡ. Điều này không chỉ đe dọa đến quyền lợi của cá nhân liên quan mà còn làm suy giảm niềm tin vào toàn bộ hệ thống giao dịch.

Tính bí mật là nền tảng không thể thiếu để bảo vệ quyền riêng tư cho người có thông tin được lưu trữ trong hệ thống. Tuy vậy, chỉ riêng tính bí mật là chưa đủ để xây dựng một môi trường an toàn toàn diện. Nó cần được kết hợp với các yếu tố khác như tính toàn vẹn và tính khả dụng để tạo nên một hệ thống thông tin thực sự vững chắc và đáng tin cậy.

#### 2.1.2.2 Tính toàn vẹn

Tính toàn vẹn (Integrity) là yếu tố đảm bảo rằng dữ liệu luôn chính xác, đầy đủ và không bị chỉnh sửa, xóa bỏ hoặc tạo mới một cách trái phép. Nó đóng vai trò quan trọng trong việc duy trì độ tin cậy của thông tin, đồng thời ngăn chặn các hành vi gian lận, phá hoại hoặc làm sai lệch dữ liệu.

Khái niệm toàn vẹn trong an toàn thông tin khác với tính toàn vẹn tham chiếu trong cơ sở dữ liệu, dù chúng có mối liên hệ nhất định. Tính toàn vẹn dữ liệu trong an toàn thông tin có thể được xem như một trường hợp đặc biệt của tính nhất quán trong mô hình ACID (gồm tính nguyên tử, tính nhất quán, tính cách ly và tính lâu bền) – vốn được sử dụng để đảm bảo độ tin cậy cho các hệ thống xử lý giao dịch.

Tính toàn vẹn bị vi phạm khi dữ liệu bị thay đổi trong quá trình truyền tải hoặc lưu trữ mà không có sự nhận biết hoặc kiểm soát thích hợp. Ví dụ, nếu một thông điệp bị chỉnh sửa trong quá trình giao dịch mà hệ thống không phát hiện ra, thì tính toàn vẹn của hệ thống đã bị tổn hại.

Một hệ thống thông tin an toàn là hệ thống phải đảm bảo không chỉ tính bí mật mà còn cả tính toàn vẹn của dữ liệu, nhằm cung cấp các thông điệp đáng tin cậy, chính xác và không thể bị thay đổi ngoài ý muốn.

#### 2.1.2.3 Tính sẵn sàng.

Tính sẵn sàng là một trong những yếu tố quan trọng trong an toàn thông tin, đảm bảo rằng thông tin và các hệ thống thông tin luôn có sẵn và có thể truy cập được khi người dùng hoặc hệ thống được ủy quyền cần thiết. Yếu tố này giúp duy trì sự liên tục của các hoạt động kinh doanh, đồng thời ngăn ngừa các gián đoạn dịch vụ có thể ảnh hưởng đến hiệu quả của tổ chức.

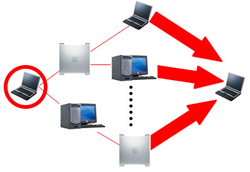
Mỗi hệ thống thông tin đều có những mục đích và yêu cầu riêng biệt. Tuy nhiên, bất kể mục đích là gì, một yêu cầu chung là thông tin cần phải luôn sẵn sàng khi người dùng yêu cầu. Điều này có nghĩa là hệ thống phần cứng dùng để lưu trữ và xử lý thông tin, các công cụ bảo mật để bảo vệ thông tin và các kênh kết nối để truy cập thông tin phải luôn hoạt động ổn định và chính xác.

Hệ thống có tính sẵn sàng cao sẽ có khả năng duy trì hoạt động liên tục bất chấp những rủi ro tiềm tàng về phần cứng, phần mềm hoặc các sự cố ngoài dự tính, như sự cố mất điện, hư hỏng phần cứng, lỗi phần mềm, hoặc trong quá trình bảo trì, nâng cấp hệ thống. Bên cạnh đó, tính sẵn sàng còn liên quan mật thiết đến khả năng ngăn chặn các cuộc tấn công từ chối dịch vụ (DoS/DDoS) - một hình thức tấn công nhằm làm gián đoạn khả năng cung cấp dịch vụ của hệ thống.

### 2.1.3 Các hình thức tấn công mạng .

#### 2.1.3.1 Tấn công Dos/Ddos

Tấn công từ chối dịch vụ (DoS) là một phương thức tấn công mạng mà mục tiêu chính là làm gián đoạn hoặc ngừng hoạt động của dịch vụ mạng hoặc hệ thống, khiến chúng không thể cung cấp dịch vụ cho người dùng hợp pháp. Mặc dù các phương pháp phân phối trực tiếp khác chủ yếu tập trung vào việc xâm nhập và truy cập vào hệ thống và dữ liệu, thì tấn công DoS chủ yếu nhằm mục đích phá hoại hệ thống bằng cách làm giảm khả năng hoạt động của chúng. Các cuộc tấn công DoS có thể được phân loại thành nhiều cấp độ khác nhau:



Tấn công từ chối dịch vu.

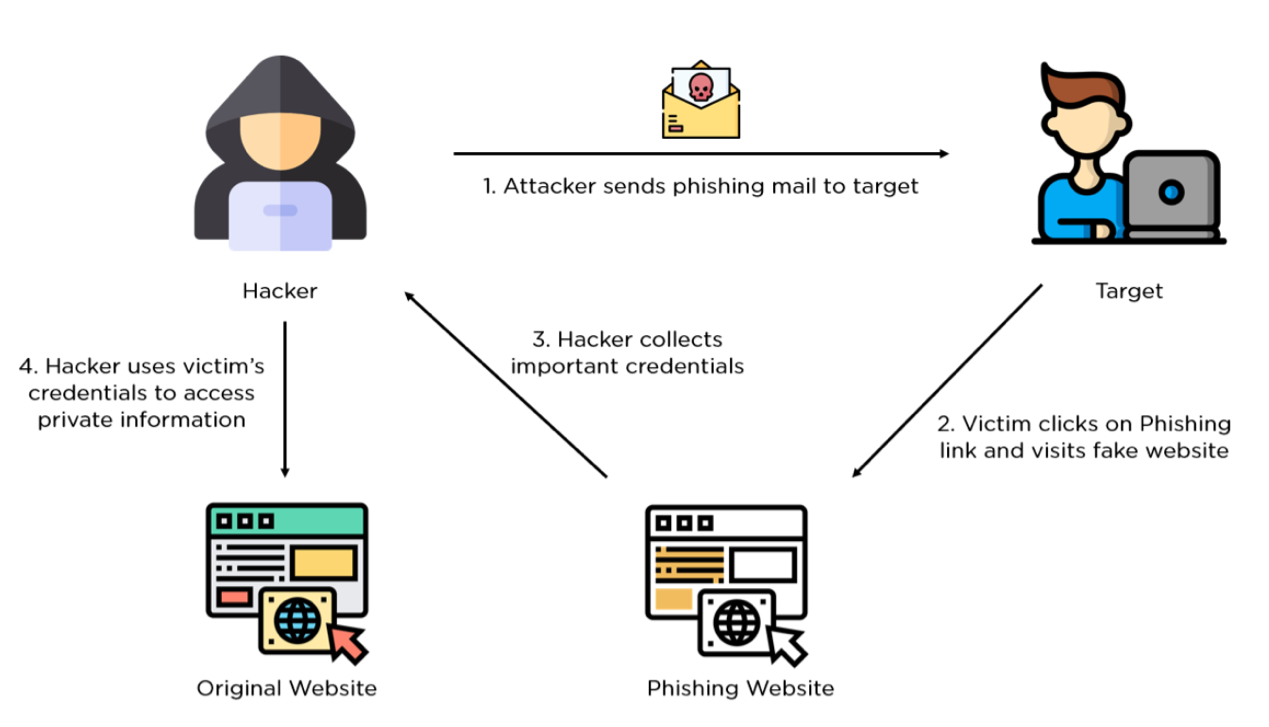
Tấn công DoS thông qua tấn công lũ dữ liệu (DDoS) là hình thức tấn công phổ biến, trong đó kẻ tấn công gửi một lượng lớn dữ liệu hoặc gói tin đến một dịch vụ hoặc mạng, khiến nó không thể xử lý hết được khối lượng dữ liệu này. Điều này làm cho hệ thống hoặc dịch vụ trở nên không khả dụng đối với người dùng hợp pháp. Một trong những phương thức phổ biến của tấn công này là tấn công DoS phân tán (DDoS), trong đó nhiều máy tính hoặc thiết bị bị xâm nhập sẽ cùng tấn công một hệ thống mục tiêu. Các kỹ thuật khuếch đại cũng có thể được sử dụng để gia tăng mức độ nghiêm trọng của tấn công, chẳng hạn như việc lợi dụng giao thức DNS hoặc giao thức NTP để tăng cường hiệu quả tấn công.

Tấn công DoS thông qua khai thác lỗ hổng (Zero-Day) là hình thức khác của tấn công DoS là việc khai thác các lỗ hổng bảo mật chưa được biết đến trong hệ thống mục tiêu. Kẻ tấn công có thể lợi dụng lỗ hổng này để gây thiệt hại nghiêm trọng, làm hỏng hoặc làm gián đoạn các dịch vụ quan trọng của hệ thống mà không thể phục hồi dễ dàng. Những lỗ hổng này có thể là những điểm yếu chưa được nhà cung cấp phần mềm hoặc tổ chức bảo mật nhận diện, khiến cho các cuộc tấn công này có thể kéo dài mà không bị phát hiện.

Tấn công DoS thông qua xâm nhập và phá hoại là loại tấn công xảy ra khi kẻ tấn công xâm nhập vào mạng và chủ động phá hoại dữ liệu hoặc hệ thống của mục tiêu. Một ví dụ điển hình là cuộc tấn công của virus Shamoon vào năm 2012, đã tấn công và làm hỏng 30.000 hệ thống của công ty dầu mỏ Saudi Aramco. Virus này xóa dữ liệu quan trọng của công ty, gây thiệt hại lớn và làm gián đoạn các hoạt động kinh doanh của họ.

#### 2.1.3.1 Tấn công Phishing

Phishing là hành vi cố gắng lừa đảo để thu thập thông tin nhạy cảm như tên người dùng, mật khẩu và chi tiết thẻ tín dụng, giả mạo là một tổ chức đáng tin cậy trong các giao tiếp điện tử. Các cuộc giao tiếp này thường giả danh các trang web xã hội phổ biến, trang đấu giá, các dịch vụ thanh toán trực tuyến hoặc quản trị viên CNTT(Công nghệ thông tin) để dụ dỗ công chúng không nghi ngờ. Email phishing có thể chứa các liên kết dẫn đến các trang web bị nhiễm phần mềm độc hại.



Phishing là một ví dụ điển hình của kỹ thuật Kỹ thuật xã hội (Social Engineering). Phishing chủ yếu được sử dụng trong các cuộc tấn công qua email. Trong tấn công phishing qua email, kẻ tấn công gửi một liên kết qua email đến người dùng, chẳng hạn như yêu cầu cung cấp thông tin ngân hàng hoặc các thông tin cá nhân khác. Người dùng sau đó nhấp vào liên kết và điền đầy đủ thông tin vào trang web giả mạo, và kẻ tấn công sẽ thu thập được tất cả thông tin của người dùng. Đây là cách thức tấn công phishing hoạt động.

Quá trình phishing được giải thích theo các giai đoạn như sau:

Giai đoạn 1: Một hacker độc hại gửi email hoặc tin nhắn đến mục tiêu, giả mạo là một nguồn đáng tin cậy. Thường thì hacker sẽ yêu cầu mục tiêu nhấp vào một liên kết của bên thứ ba để kiểm tra bảo mật hoặc cập nhật tính năng đơn giản.

Giai đoạn 2: Mục tiêu tin rằng email này đến từ một tổ chức uy tín, chẳng hạn như ngân hàng hoặc công ty, và nhấp vào liên kết độc hại dẫn đến một trang web giả mạo, được thiết kế sao cho trông giống nhất với trang web chính thức.

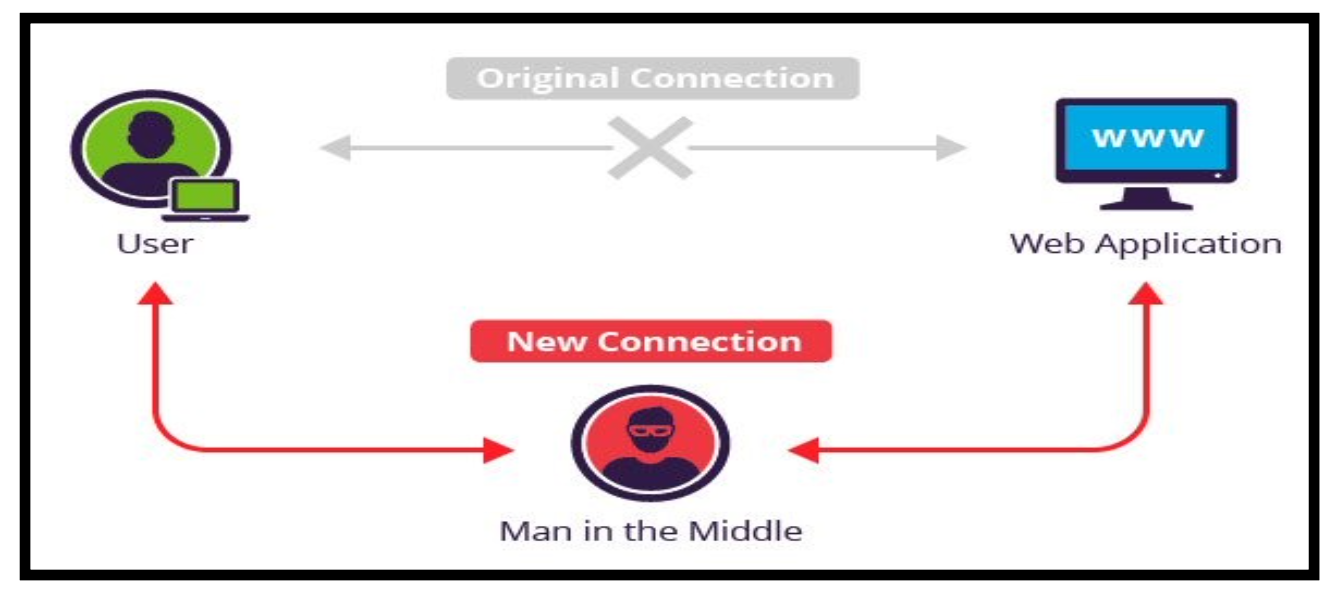
Giai đoạn 3: Trên trang web giả mạo, người dùng sẽ được yêu cầu nhập các thông tin cá nhân, như thông tin đăng nhập của tài khoản cho một dịch vụ cụ thể. Sau khi các thông tin này được cung cấp, tất cả sẽ được chuyển trực tiếp đến hacker, người đã thiết kế trang web và email độc hại.

Giai đoạn 4: Khi đã thu thập được thông tin tài khoản, hacker có thể sử dụng chúng để đăng nhập vào hệ thống hoặc bán các dữ liệu thu thập được trên internet cho những người trả giá cao nhất.

Phishing bắt đầu với một email hoặc loại thông điệp khác được thiết kế nhằm tấn công nạn nhân. Thông điệp này được làm giả sao cho giống như đến từ một người gửi đáng tin cậy. Nếu nạn nhân bị lừa, họ sẽ cung cấp thông tin cá nhân cho một trang web giả mạo.

#### 2.1.3.3 Tấn công Man-in-the-Middle (MitM)

Tấn công Man-in-the-Middle (MITM) hay còn gọi là tấn công “kẻ ở giữa” là một dạng tấn công mạng có tính chất nguy hiểm và tinh vi, trong đó kẻ tấn công bí mật chèn mình vào giữa quá trình giao tiếp của hai thực thể mà không bị phát hiện. Mục tiêu của hình thức tấn công này thường là để theo dõi, đánh cắp hoặc chỉnh sửa dữ liệu được truyền đi giữa các bên mà không làm gián đoạn kết nối, khiến nạn nhân không nhận ra sự bất thường



Sơ đồ tấn công Main-in-the-middle.

Khác với các cuộc tấn công từ chối dịch vụ (DoS) thường gây ảnh hưởng rõ ràng đến hệ thống, MITM diễn ra âm thầm, làm cho quá trình giao tiếp giữa người dùng và máy chủ vẫn diễn ra bình thường. Chính vì vậy, nếu không có hệ thống giám sát an ninh mạnh mẽ, việc phát hiện ra các hành vi MITM gần như là không thể

Tùy vào cách tiếp cận, MITM được chia thành hai loại phổ biến là MITM vật lý và MITM phần mềm. Trong trường hợp MITM vật lý, kẻ tấn công phải xuất hiện trong cùng một mạng với nạn nhân, ví dụ như trong môi trường Wi-Fi công cộng. Một tình huống điển hình là khi kẻ xấu thiết lập một điểm truy cập Wi-Fi giả mạo, tên giống với điểm truy cập thật, để dụ người dùng kết nối. Khi người dùng truy cập vào mạng giả mạo này, toàn bộ dữ liệu sẽ bị ghi lại và có thể bị chỉnh sửa hoặc chuyển hướng đến những trang web giả mạo

Trong khi đó, MITM phần mềm không yêu cầu sự hiện diện vật lý gần nạn nhân. Thay vào đó, kẻ tấn công sử dụng phần mềm độc hại đã được cài sẵn trong thiết bị để chặn dữ liệu. Một trong những hình thức phổ biến là sử dụng keylogger – chương trình ghi lại toàn bộ thao tác gõ phím của người dùng, từ đó thu thập thông tin đăng nhập, mật khẩu hoặc mã OTP

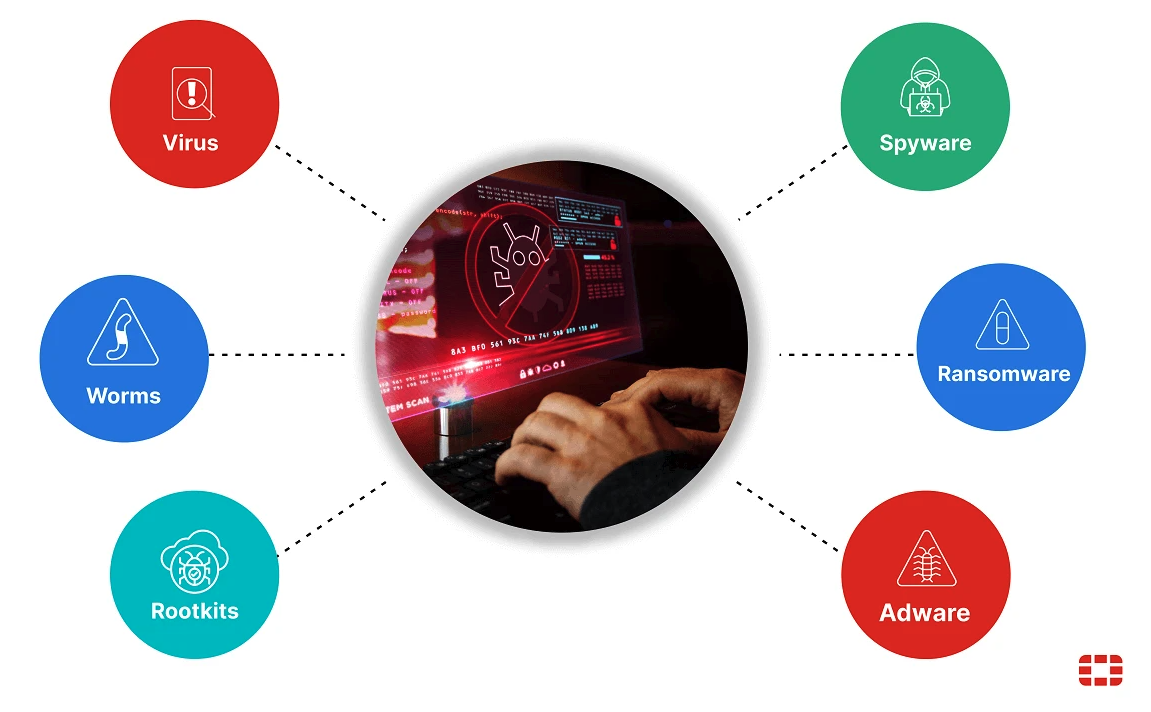
Một cuộc tấn công MITM thường diễn ra theo hai giai đoạn. Giai đoạn đầu là chặn bắt dữ liệu (Interception), trong đó kẻ tấn công sẽ cố gắng ngăn chặn hoặc theo dõi luồng dữ liệu giữa hai thiết bị. Có nhiều kỹ thuật được áp dụng như ARP Spoofing – giả mạo mối liên kết giữa địa chỉ IP và MAC để chuyển hướng dữ liệu, DNS Spoofing – giả mạo phản hồi truy vấn tên miền để chuyển hướng người dùng sang máy chủ giả, IP Spoofing – giả mạo địa chỉ IP để giành quyền truy cập, hay SSL Stripping – hạ cấp kết nối HTTPS về HTTP nhằm loại bỏ lớp mã hóa

Khi đã chặn được dữ liệu, kẻ tấn công sẽ bước sang giai đoạn giải mã và khai thác thông tin (Decryption). Nếu dữ liệu đã được mã hóa, chúng sẽ cố gắng phá vỡ lớp bảo mật này bằng cách sử dụng các kỹ thuật như HTTPS spoofing – giả mạo chứng chỉ số để đánh lừa trình duyệt rằng kết nối an toàn, hoặc SSL hijacking – chiếm quyền điều khiển luồng dữ liệu bảo mật giữa người dùng và máy chủ. Trong một số trường hợp, kẻ tấn công còn có thể lợi dụng lỗ hổng trong hệ điều hành, trình duyệt hoặc các phần mềm khác để can thiệp sâu hơn

MITM là mối đe dọa đặc biệt nghiêm trọng đối với các hoạt động cần tính bảo mật cao như giao dịch ngân hàng trực tuyến, xác thực hai bước bằng mã OTP, truy cập hệ thống email, tài khoản mạng xã hội hoặc các hệ thống lưu trữ thông tin cá nhân nhạy cảm. Một khi dữ liệu bị rơi vào tay kẻ xấu, hậu quả có thể là mất quyền truy cập vào tài khoản, bị chiếm đoạt tài sản hoặc bị rò rỉ thông tin riêng tư

Tấn công MITM thường dễ xảy ra tại các điểm truy cập công cộng thiếu bảo mật hoặc trên các thiết bị không có phần mềm bảo vệ mạnh mẽ. Các hệ thống không sử dụng mã hóa đầu cuối hay không xác thực danh tính một cách chặt chẽ cũng là mục tiêu lý tưởng cho kẻ tấn công

#### 2.1.3.4 Tấn công Mlware

****

**Các phần mềm độc hại**

Trong thế giới số hóa hiện đại, rủi ro từ phần mềm độc hại (malware) ngày càng gia tăng, không chỉ đe dọa người dùng cá nhân mà còn cả các doanh nghiệp và tổ chức. Malware là thuật ngữ bao trùm cho bất kỳ loại phần mềm hoặc mã nào được thiết kế với mục đích xâm nhập, kiểm soát, gây tổn hại hoặc làm gián đoạn hoạt động bình thường của hệ thống. Chúng bao gồm máy tính, điện thoại thông minh, máy chủ và các thiết bị kết nối mạng khác. Những kẻ đứng sau các cuộc tấn công malware có nhiều động cơ khác nhau, từ việc đơn thuần phá hoại hệ thống, đánh cắp thông tin cá nhân và tài chính nhạy cảm, đến tống tiền và thực hiện các cuộc tấn công mạng có quy mô lớn hơn nhiều.

Để hiểu rõ hơn về vấn đề này, chúng ta hãy cùng điểm qua một số loại malware phổ biến nhất hiện nay và cách chúng hoạt động. Virus là một trong những loại lâu đời nhất, chúng lây lan bằng cách gắn mã độc của mình vào các tập tin hoặc chương trình hợp pháp. Khi người dùng vô tình mở tập tin đã nhiễm virus, mã độc sẽ kích hoạt và bắt đầu sao chép chính nó sang các tập tin khác, gây ra thiệt hại như xóa dữ liệu, làm hỏng hệ điều hành hoặc thậm chí chiếm quyền điều khiển máy tính của bạn.

Một dạng khác, được biết đến với tên gọi Worm, có khả năng tự nhân bản và lây lan qua mạng mà không cần sự can thiệp của người dùng. Worm khai thác các lỗ hổng bảo mật trong hệ điều hành hoặc các ứng dụng để tự động lây lan sang các máy tính khác trong cùng một mạng, gây ra tình trạng tắc nghẽn băng thông, làm chậm hiệu suất hệ thống hoặc đánh cắp thông tin nhạy cảm.

Trojan, một thuật ngữ bắt nguồn từ câu chuyện con ngựa thành Troy nổi tiếng, là loại malware ngụy trang dưới dạng phần mềm hữu ích hoặc tập tin an toàn để lừa người dùng cài đặt. Sau khi được cài đặt, trojan có thể thực hiện các hành động độc hại như tạo cửa hậu để kẻ tấn công truy cập vào máy tính từ xa, đánh cắp thông tin cá nhân hoặc cài đặt các loại malware khác.

Spyware, như tên gọi của nó, được thiết kế để bí mật thu thập thông tin về hoạt động trực tuyến của người dùng, bao gồm lịch sử duyệt web, thông tin đăng nhập, mật khẩu và các dữ liệu cá nhân khác. Spyware thường được cài đặt thông qua các phần mềm miễn phí không rõ nguồn gốc, các trang web độc hại hoặc các email lừa đảo.

Rootkit là một tập hợp các công cụ phần mềm được thiết kế để ẩn sự hiện diện của các phần mềm độc hại khác trên hệ thống. Rootkit cho phép kẻ tấn công kiểm soát máy tính từ xa mà không bị phát hiện, đồng thời che giấu các hoạt động độc hại của các malware khác như virus, worm hoặc trojan.

Ransomware là một trong những loại malware nguy hiểm nhất hiện nay, nó mã hóa dữ liệu trên máy tính của nạn nhân và yêu cầu trả tiền chuộc để khôi phục lại quyền truy cập. Ransomware thường lây lan qua các email lừa đảo, các trang web độc hại hoặc khai thác các lỗ hổng bảo mật trong hệ thống.

#### 2.1.3.5 Tấn công cơ sở dữ liệu ( SQL Injection )

Tấn công SQL Injection (SQLi) là một kỹ thuật tấn công cho phép kẻ tấn công chèn các đoạn mã SQL độc hại vào các truy vấn cơ sở dữ liệu thông qua việc lợi dụng các lỗ hổng trong việc xử lý dữ liệu đầu vào của ứng dụng web. Khi ứng dụng web không kiểm tra hoặc xử lý dữ liệu nhập từ người dùng một cách an toàn, kẻ tấn công có thể lợi dụng điều này để thay đổi cấu trúc hoặc nội dung của các câu lệnh SQL, từ đó thực hiện các hành động trái phép trên cơ sở dữ liệu.



Tấn công cơ sở dữ liệu

Nguyên nhân chính của các lỗ hổng SQL Injection xuất phát từ việc ứng dụng web tin tưởng một cách mù quáng vào dữ liệu đầu vào từ người dùng. Dữ liệu này được sử dụng trực tiếp để xây dựng các câu lệnh SQL mà không trải qua quá trình kiểm tra, lọc bỏ hoặc mã hóa thích hợp. Kẻ tấn công có thể dễ dàng chèn các đoạn mã SQL độc hại vào các trường nhập liệu, tham số URL hoặc các nguồn dữ liệu khác mà ứng dụng sử dụng.

SQL Injection có nhiều hình thức khác nhau, tùy thuộc vào cách kẻ tấn công khai thác lỗ hổng và mục tiêu mà chúng nhắm đến. Một số hình thức tấn công SQL Injection phổ biến bao gồm:

Tấn công dựa vào lỗi (Error-based SQL Injection) cho phép kẻ tấn công lợi dụng các thông báo lỗi từ cơ sở dữ liệu để thu thập thông tin về cấu trúc cơ sở dữ liệu, phiên bản phần mềm và các thông tin nhạy cảm khác. Tấn công dựa trên Boolean (Boolean-based Blind SQL Injection) cho phép kẻ tấn công gửi các truy vấn SQL với các điều kiện Boolean (TRUE hoặc FALSE) và quan sát phản ứng của ứng dụng để suy ra thông tin về cơ sở dữ liệu. Tấn công dựa trên thời gian (Time-based Blind SQL Injection) cho phép kẻ tấn công chèn các câu lệnh SQL làm chậm quá trình thực thi của truy vấn, từ đó xác định thông tin về cơ sở dữ liệu dựa trên thời gian phản hồi. Tấn công Union-based SQL Injection sử dụng toán tử UNION trong SQL để kết hợp kết quả của truy vấn gốc với kết quả của một truy vấn do kẻ tấn công tạo ra, từ đó truy xuất dữ liệu trái phép. Tấn công Stored Procedure Injection lợi dụng các lỗ hổng trong stored procedures (các đoạn mã SQL được lưu trữ trên máy chủ cơ sở dữ liệu) để thực thi các câu lệnh SQL độc hại.

Tấn công SQL Injection có thể gây ra những hậu quả vô cùng nghiêm trọng cho cả ứng dụng web và người dùng, bao gồm đánh cắp dữ liệu nhạy cảm. Kẻ tấn công có thể truy xuất thông tin cá nhân của người dùng, thông tin tài chính, bí mật kinh doanh và các dữ liệu nhạy cảm khác. Kẻ tấn công có thể thay đổi hoặc xóa dữ liệu trong cơ sở dữ liệu, dẫn đến sai lệch thông tin, mất mát dữ liệu và ảnh hưởng đến hoạt động của ứng dụng. Kẻ tấn công có thể sử dụng SQL Injection để vượt qua cơ chế xác thực và chiếm đoạt tài khoản của người dùng. Trong một số trường hợp, kẻ tấn công có thể sử dụng SQL Injection để leo thang đặc quyền và giành quyền kiểm soát hệ thống. Kẻ tấn công cũng có thể làm quá tải cơ sở dữ liệu, dẫn đến từ chối dịch vụ cho người dùng hợp pháp. Cuối cùng, kẻ tấn công có thể sử dụng SQL Injection để chèn mã độc vào cơ sở dữ liệu, lây lan sang các ứng dụng khác và gây ra những hậu quả nghiêm trọng hơn.

Để phòng ngừa và giảm thiểu rủi ro từ tấn công SQL Injection, cần có một loạt các biện pháp bảo mật toàn diện, bao gồm sử dụng Prepared Statements (Parameterized Queries), đây là biện pháp hiệu quả nhất để ngăn chặn SQL Injection. Prepared statements cho phép tách biệt dữ liệu và câu lệnh SQL, ngăn chặn việc chèn mã SQL độc hại vào truy vấn. Luôn kiểm tra và lọc dữ liệu đầu vào từ người dùng để đảm bảo rằng nó hợp lệ và không chứa các ký tự đặc biệt có thể được sử dụng để tấn công SQL Injection. Sử dụng Stored Procedures có thể giúp giảm thiểu nguy cơ SQL Injection bằng cách hạn chế quyền truy cập trực tiếp vào cơ sở dữ liệu. Chỉ cấp quyền truy cập cơ sở dữ liệu tối thiểu cần thiết cho ứng dụng. Mã hóa dữ liệu nhạy cảm trong cơ sở dữ liệu để bảo vệ chúng khỏi bị đánh cắp hoặc truy cập trái phép. Cập nhật hệ điều hành, máy chủ web, hệ quản trị cơ sở dữ liệu và các thư viện liên quan để vá các lỗ hổng bảo mật đã biết. Sử dụng Web Application Firewall (WAF) để phát hiện và ngăn chặn các cuộc tấn công SQL Injection. Thực hiện kiểm tra bảo mật định kỳ để xác định và khắc phục các lỗ hổng bảo mật trong ứng dụng web. Nâng cao nhận thức về bảo mật cho các nhà phát triển và quản trị viên hệ thống để họ có thể xây dựng và duy trì các ứng dụng web an toàn.

Tấn công SQL Injection là một mối đe dọa nghiêm trọng đối với các ứng dụng web hiện nay. Để bảo vệ ứng dụng và dữ liệu của người dùng, cần áp dụng một loạt các biện pháp bảo mật toàn diện, từ việc sử dụng prepared statements, kiểm tra dữ liệu đầu vào, cho đến việc cập nhật phần mềm và thực hiện kiểm tra bảo mật thường xuyên. Chỉ khi nâng cao nhận thức về bảo mật và áp dụng các biện pháp phòng ngừa hiệu quả, chúng ta mới có thể giảm thiểu rủi ro từ tấn công SQL Injection và xây dựng một môi trường trực tuyến an toàn hơn.

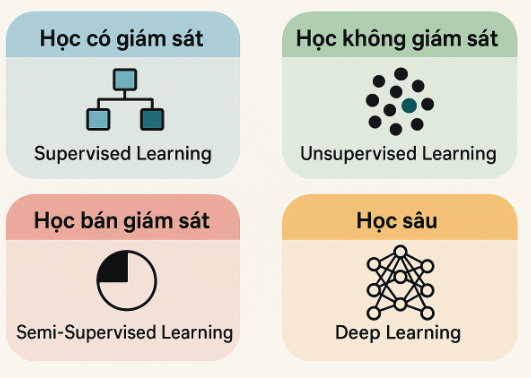
## 2.3. Ứng dụng học máy trong an toàn thông tin.

Trong bối cảnh các cuộc tấn công mạng ngày càng tinh vi và có khả năng vượt qua các hệ thống phòng thủ truyền thống, việc ứng dụng học máy (Machine Learning – ML) vào an toàn thông tin đã trở thành xu hướng tất yếu. Các mô hình ML có khả năng học từ dữ liệu quá khứ, nhận diện hành vi bất thường để phát hiện cả những cuộc tấn công chưa từng được biết đến (zero-day), trong khi các hệ thống truyền thống chỉ phát hiện được các mối đe dọa đã được định danh.

Học máy được ứng dụng rộng rãi trong nhiều lĩnh vực như hệ thống phát hiện xâm nhập (IDS), nhận diện phần mềm độc hại, phân tích hành vi người dùng (UBA), phát hiện tấn công từ chối dịch vụ (DoS/DDoS), phân loại lưu lượng mạng, xác định lỗ hổng hệ thống và đề xuất biện pháp phòng thủ chủ động.

Quy trình ứng dụng học máy bao gồm: thu thập và xử lý dữ liệu (log hệ thống, truy cập mạng, tệp tin nhật ký, dữ liệu thời gian thực...), trích xuất đặc trưng, huấn luyện – đánh giá – và triển khai mô hình. Qua đó, hệ thống có thể phân biệt giữa hành vi bình thường và hành vi tiềm ẩn nguy cơ tấn công, nâng cao khả năng bảo vệ toàn diện và linh hoạt hơn trước các mối đe dọa mạng hiện đại.

### 2.3.1. Các kỹ thuật phát hiện tấn công mạng.



Các kỹ thuật học máy được sử dụng để phát hiện tấn công mạng có thể được chia thành ba nhóm chính là học có giám sát học không giám sát và học bán giám sát mỗi nhóm có ưu điểm và nhược điểm riêng phù hợp với từng loại dữ liệu và bài toán khác nhau

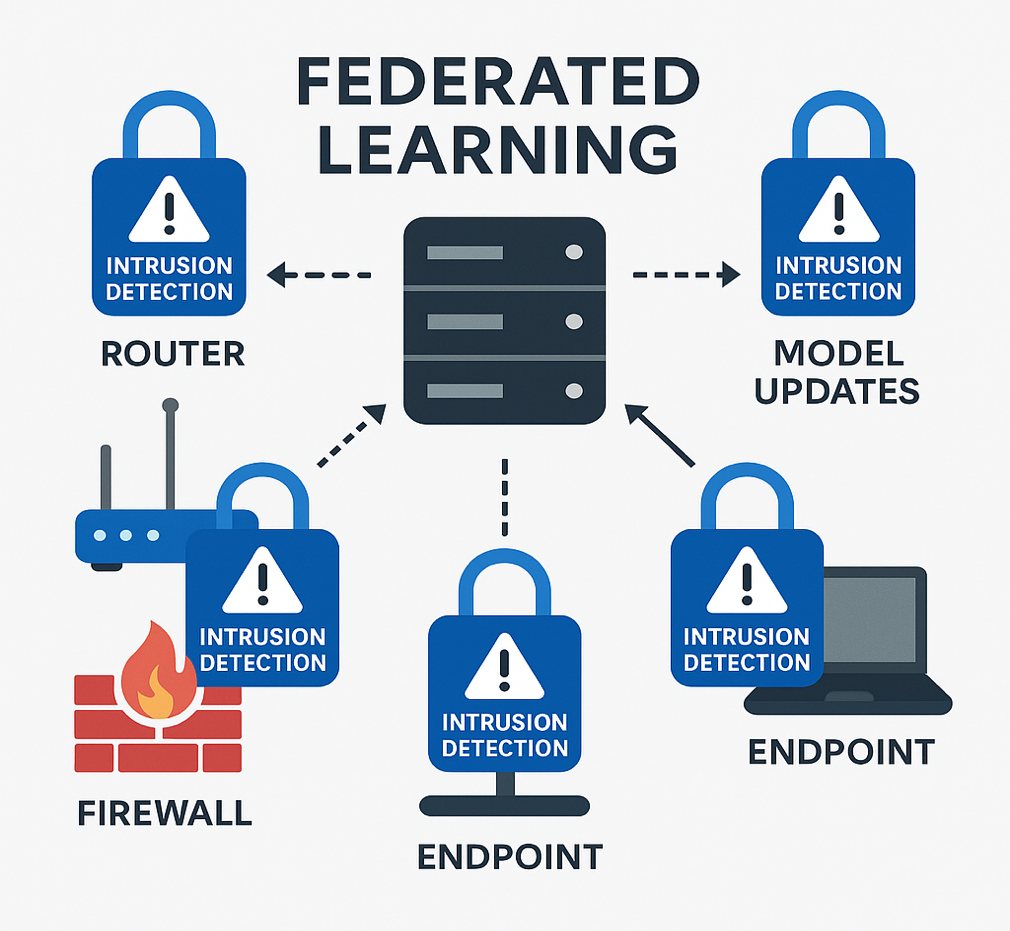
**Học có giám sát supervised learning** là phương pháp phổ biến nhất trong phát hiện tấn công mạng với yêu cầu dữ liệu huấn luyện phải được gán nhãn đầy đủ ví dụ mỗi mẫu dữ liệu đều phải được xác định là bình thường hay là một loại tấn công cụ thể những thuật toán thường được sử dụng bao gồm Decision Tree Random Forest Support Vector Machine(SVM), k-Nearest Neighbor((KNN)) và Logistic Regression ưu điểm của phương pháp này là cho độ chính xác cao và khả năng phân loại tốt tuy nhiên hạn chế lớn nhất là cần lượng dữ liệu được gán nhãn đầy đủ và chính xác điều này có thể khó khăn trong môi trường thực tế vì việc gán nhãn yêu cầu chi phí nhân lực lớn và có thể bị sai lệch chủ quan

**Học không giám sát unsupervised learning** là kỹ thuật được áp dụng khi không có hoặc có rất ít dữ liệu được gán nhãn mục tiêu của phương pháp này là phát hiện những mẫu bất thường anomaly detection trong dữ liệu học không giám sát không yêu cầu biết trước thông tin về tấn công mà sẽ tự động nhóm dữ liệu và xác định các điểm nằm ngoài nhóm chính ví dụ thuật toán K-Means Clustering Hierarchical Clustering hoặc DBSCAN có thể phát hiện ra những hành vi bất thường trong lưu lượng mạng mà trước đó chưa từng được nhận diện học không giám sát đặc biệt phù hợp với việc phát hiện tấn công zero-day nhưng độ chính xác có thể thấp hơn học có giám sát

**Học bán giám sát semi-supervised learning** là sự kết hợp giữa hai phương pháp trên nhằm tận dụng một lượng nhỏ dữ liệu được gán nhãn kết hợp với lượng lớn dữ liệu không nhãn để cải thiện hiệu quả phát hiện tấn công kỹ thuật này đang ngày càng phổ biến trong bối cảnh dữ liệu có nhãn trở nên khan hiếm

Ngoài ra trong những năm gần đây sự phát triển mạnh mẽ của học sâu deep learning cũng được ứng dụng vào việc phát hiện tấn công mạng với khả năng học các đặc trưng phức tạp từ dữ liệu thô mà không cần nhiều bước tiền xử lý các mạng neuron sâu như Convolutional Neural Network(CNN), Recurrent Neural Network(RNN), hoặc Long Short-Term Memory(LSTM) đã được chứng minh hiệu quả trong việc phân tích lưu lượng mạng chuỗi sự kiện thời gian và nhận diện hành vi bất thường

### 2.3.2 Ứng dụng Federated Learning trong phát hiện tấn công mạng.



Federated Learning hay còn gọi là học liên kết là một phương pháp học máy phân tán mới nổi cho phép việc huấn luyện mô hình được thực hiện trực tiếp trên nhiều thiết bị hoặc nút dữ liệu mà không cần phải tập trung toàn bộ dữ liệu về một máy chủ trung tâm như các phương pháp học tập truyền thống Điều này mang lại lợi thế lớn về quyền riêng tư và bảo mật dữ liệu đặc biệt phù hợp với các lĩnh vực nhạy cảm như an ninh mạng nơi mà dữ liệu thường mang tính cá nhân cao và khó có thể chia sẻ công khai

Cơ chế hoạt động của Federated Learning dựa trên nguyên tắc mỗi thiết bị hoặc tổ chức sẽ huấn luyện một mô hình cục bộ trên chính tập dữ liệu của mình sau khi hoàn tất huấn luyện các thiết bị này chỉ gửi các tham số mô hình ví dụ như trọng số mạng nơ-ron về máy chủ trung tâm để tổng hợp lại thành một mô hình toàn cục chung quá trình này hoàn toàn không yêu cầu gửi dữ liệu gốc từ thiết bị lên trung tâm do đó dữ liệu luôn được giữ an toàn tại chỗ và không bị lộ ra ngoài môi trường điều này có ý nghĩa rất lớn trong việc ngăn chặn rò rỉ thông tin và bảo vệ quyền riêng tư người dùng

Trong bối cảnh phát hiện tấn công mạng Federated Learning cho phép các thiết bị biên như router tường lửa máy trạm hoặc hệ thống giám sát an ninh trong từng tổ chức có thể tự xây dựng mô hình phát hiện tấn công dựa trên dữ liệu mạng nội bộ như log hệ thống hoặc lưu lượng mạng mỗi thiết bị này sẽ tiến hành huấn luyện độc lập trên dữ liệu riêng của nó và định kỳ gửi tham số mô hình đã huấn luyện về máy chủ trung tâm để máy chủ này thực hiện quá trình tổng hợp và cập nhật mô hình chung sau đó mô hình này được phân phối lại cho các thiết bị tham gia để tiếp tục vòng lặp học liên kết giúp mô hình toàn cục ngày càng chính xác và toàn diện hơn

Điểm nổi bật của phương pháp này là không yêu cầu chia sẻ dữ liệu nhạy cảm giữa các tổ chức hoặc thiết bị do đó các vấn đề liên quan đến bảo mật và tuân thủ pháp lý như GDPR hoặc các chính sách bảo mật nội bộ doanh nghiệp được giải quyết một cách hiệu quả bên cạnh đó việc các thiết bị được tự học từ dữ liệu của mình sẽ giúp mô hình phát hiện được những loại tấn công đặc thù mà chỉ riêng từng môi trường mới có điều này đồng nghĩa với việc khi các thiết bị đóng góp vào mô hình toàn cục thì hệ thống sẽ dần học được các loại tấn công đa dạng từ nhiều nguồn khác nhau giúp tăng khả năng bao phủ và nâng cao hiệu quả giám sát mạng

Nhiều nghiên cứu gần đây đã ứng dụng Federated Learning để phát hiện các kiểu tấn công mạng như từ chối dịch vụ DoS dò quét cổng quét mạng tấn công injection hoặc phát hiện mã độc bằng cách sử dụng các mô hình học sâu như mạng nơ-ron tích chập CNN hoặc mạng ghi nhớ dài hạn LSTM kết quả cho thấy rằng trong cả môi trường mô phỏng và thực tế mô hình học liên kết có thể đạt được độ chính xác gần tương đương hoặc thậm chí cao hơn so với các mô hình học tập trung truyền thống đồng thời đảm bảo tính riêng tư của dữ liệu một cách tối ưu

Tuy nhiên việc triển khai Federated Learning trong thực tế vẫn đang gặp nhiều thách thức ví dụ như vấn đề không đồng đều trong phân phối dữ liệu giữa các thiết bị khiến cho mô hình khó tổng hợp hiệu quả thêm vào đó là sự không đồng bộ trong khả năng tính toán và băng thông mạng giữa các thiết bị tham gia làm ảnh hưởng đến hiệu suất huấn luyện ngoài ra còn tồn tại nguy cơ bị tấn công thông qua các kỹ thuật phục hồi dữ liệu từ tham số mô hình đòi hỏi cần áp dụng các biện pháp bảo vệ bổ sung như mã hóa tham số hoặc sử dụng các cơ chế bảo mật nâng cao như tổng hợp an toàn hay privacy-preserving aggregation

## 2.4. Các công trình liên quan

### 2.4.1 Các công trình phát hiện tấn công DDoS

Tấn công từ chối dịch vụ phân tán (DDoS) hiện đang là một trong những mối đe dọa phổ biến và nguy hiểm đối với các hệ thống mạng, đặc biệt là trong môi trường Internet of Things (IoT). Với đặc điểm là số lượng lớn thiết bị kết nối và phần lớn trong số đó có cấu hình thấp, khả năng phòng thủ mạng yếu, hệ thống IoT trở thành mục tiêu lý tưởng của các cuộc tấn công DDoS. Trước thực trạng đó, nhiều nghiên cứu trong những năm gần đây đã tập trung vào việc xây dựng các mô hình phát hiện DDoS hiệu quả hơn, dựa trên các kỹ thuật học máy (Machine Learning) và học sâu (Deep Learning), nhằm mục tiêu tự động hóa quy trình giám sát và nâng cao độ chính xác trong phát hiện.

Một nghiên cứu tiêu biểu có thể kể đến là của Tariq và cộng sự (2019), sử dụng tập dữ liệu CICIDS2017 để huấn luyện các mô hình phân loại dựa trên thuật toán Decision Tree, Random Forest và Naive Bayes. Trong đó, mô hình Random Forest cho thấy hiệu quả vượt trội với khả năng phân biệt rõ ràng giữa lưu lượng mạng thông thường và lưu lượng bị tấn công.

Tiếp nối xu hướng này, nhóm nghiên cứu của Vinayakumar đã triển khai mô hình mạng nơ-ron tích chập (CNN) để phát hiện tấn công DDoS trong môi trường IoT, sử dụng phiên bản cải tiến của tập dữ liệu NSL-KDD. Kết quả cho thấy mô hình đạt độ chính xác lên tới 98.7%, đồng thời có khả năng xử lý nhanh, cho thấy tính khả thi cao khi áp dụng vào các hệ thống thực tế.

Ngoài ra, tác giả dos Santos (2021) đã tiếp cận vấn đề dưới góc độ khai thác tính liên tục theo thời gian của dữ liệu mạng, bằng cách áp dụng mạng LSTM – một dạng mạng nơ-ron hồi tiếp có khả năng ghi nhớ dài hạn. Mô hình này đặc biệt hiệu quả trong việc phát hiện các cuộc tấn công có tính chất kéo dài, tinh vi và khó nhận diện bằng các kỹ thuật truyền thống.

Không chỉ dừng lại ở việc áp dụng các thuật toán hiện có, nhiều nghiên cứu gần đây còn hướng tới việc tối ưu hóa mô hình bằng cách kết hợp các phương pháp học máy với kỹ thuật tiền xử lý dữ liệu. Cụ thể, nghiên cứu của Abiramasundari và Ramaswamy (2025) đã đề xuất khung phát hiện DDoS có tích hợp PCA để giảm chiều dữ liệu, sau đó sử dụng các thuật toán phân loại như SVM, Logistic Regression, Random Forest và KNN. Việc giảm số chiều đặc trưng không những giúp tăng tốc độ xử lý mà còn giữ được độ chính xác ổn định.

Một mô hình kết hợp đáng chú ý khác được giới thiệu bởi Sahu và cộng sự (2025) sử dụng cấu trúc mạng lai giữa Conv1D, Bi-LSTM và Bi-GRU, kết hợp với dropout nhằm giảm hiện tượng quá khớp. Mô hình được huấn luyện trên tập dữ liệu IoT-DH và đạt độ chính xác lên tới 99.41%, cho thấy tính hiệu quả và tiềm năng triển khai trong các hệ thống có yêu cầu thời gian thực.

Ở một hướng nghiên cứu khác, Ahmim và cộng sự (2023) đã xây dựng mô hình học sâu kết hợp nhiều tầng xử lý đặc trưng không gian và thời gian, từ đó nâng cao khả năng nhận diện các cuộc tấn công phức tạp mà các phương pháp đơn lẻ thường gặp khó khăn. Mô hình này cũng cho kết quả tốt khi kiểm thử với nhiều loại tấn công khác nhau trên mạng IoT.

Mihoub và nhóm nghiên cứu (2022) thì chọn tiếp cận theo hướng mô hình học máy có tính hồi tưởng – giúp hệ thống nhận diện được những hành vi bất thường lặp lại theo chu kỳ hoặc có dấu hiệu “ẩn mình”. Phương pháp này tỏ ra hiệu quả hơn trong việc xử lý các cuộc tấn công kéo dài, vốn rất phổ biến trong tấn công DDoS nhắm vào IoT.

Đối với các hệ thống IoT, vấn đề không chỉ nằm ở độ chính xác mà còn ở mức độ “gọn nhẹ” của mô hình. Với mục tiêu này, Saurabh và cộng sự (2022) đã đề xuất mô hình học sâu nhẹ NFDLM, thiết kế để hoạt động hiệu quả trên các thiết bị có tài nguyên hạn chế. Mặc dù được tối giản hóa, mô hình vẫn đạt được độ chính xác xấp xỉ 99% trong các thử nghiệm thực tế.

Cuối cùng, một điểm sáng khác đến từ nghiên cứu của Ullah và cộng sự (2023), khi nhóm tác giả phát triển kỹ thuật lựa chọn thuộc tính động, cho phép hệ thống chỉ sử dụng những đặc trưng có liên quan cao đến tấn công. Cách tiếp cận này giúp giảm tải tính toán, tối ưu tài nguyên mà vẫn đảm bảo hiệu suất phát hiện tốt.

Từ tổng hợp các công trình nêu trên có thể thấy, hướng tiếp cận bằng học máy và học sâu trong phát hiện tấn công DDoS trên môi trường IoT đang ngày càng phát triển mạnh mẽ và đa dạng. Không chỉ dừng lại ở việc nâng cao độ chính xác, các nghiên cứu hiện nay còn chú trọng đến khả năng triển khai thực tế, tối ưu hiệu năng, giảm độ phức tạp và đáp ứng yêu cầu vận hành của các thiết bị IoT vốn có giới hạn về năng lực xử lý. Đây là nền tảng quan trọng cho việc xây dựng các hệ thống phòng chống DDoS thông minh, chủ động và phù hợp với xu hướng phát triển của mạng lưới thiết bị kết nối trong tương lai.

### 2.4.2 Các công trình phát hiện tấn công Brute Force

Tấn công Brute Force (hay còn gọi là tấn công thử mật khẩu toàn diện) vẫn luôn là một trong những kỹ thuật tấn công đơn giản nhưng hiệu quả, thường xuyên được tin tặc sử dụng để khai thác lỗ hổng xác thực trong các hệ thống mạng, đặc biệt là trong môi trường Internet of Things (IoT). Với đặc trưng là cơ chế xác thực đơn giản, nhiều thiết bị sử dụng mật khẩu mặc định và ít được cập nhật thường xuyên, hệ sinh thái IoT trở thành "điểm mù" dễ bị khai thác thông qua các cuộc tấn công dò mật khẩu liên tục.

Nghiên cứu của Ahmed và cộng sự (2020) được xem là một trong những công trình sớm áp dụng học máy để phát hiện hành vi tấn công Brute Force. Trong nghiên cứu này, nhóm tác giả sử dụng thuật toán SVM để huấn luyện mô hình phân loại trên tập dữ liệu UNSW-NB15. Mặc dù kết quả cho thấy độ chính xác khá cao trong việc phân biệt truy cập hợp lệ và truy cập bất thường, nhưng thời gian huấn luyện dài và yêu cầu về tài nguyên xử lý khiến mô hình này khó áp dụng trong thực tế – đặc biệt là khi triển khai trực tiếp trên các thiết bị IoT có năng lực phần cứng hạn chế.

Tiếp nối hướng nghiên cứu này, Sahu và cộng sự (2021) tập trung khai thác khả năng học chuỗi thời gian của mạng nơ-ron hồi tiếp (RNN) nhằm nhận diện các mẫu truy cập đăng nhập liên tục, thường là dấu hiệu đặc trưng của tấn công Brute Force. Nhóm nghiên cứu huấn luyện mô hình trên tập dữ liệu CICIDS2017, trong đó kết quả cho thấy khả năng phân biệt giữa hành vi đăng nhập thông thường và hành vi dò mật khẩu là rất rõ rệt, nhờ vào việc mô hình có thể phát hiện được các chuỗi hành vi có tính lặp lại không tự nhiên.

Một hướng tiếp cận khác được Lin và các cộng sự giới thiệu là tích hợp mô hình học sâu với giải thuật phát hiện bất thường (anomaly detection). Thay vì chỉ phân loại nhị phân giữa truy cập hợp lệ và truy cập tấn công, mô hình này theo dõi hành vi mạng theo thời gian thực và cảnh báo khi phát hiện bất kỳ dấu hiệu nào vượt ngưỡng hành vi bình thường. Đây là cách tiếp cận khá phù hợp với các hệ thống yêu cầu phát hiện nhanh và phản ứng tức thời – đặc biệt trong các mạng IoT có tính phân tán cao và tài nguyên giới hạn.

Trong những năm gần đây, nhiều nghiên cứu đã hướng tới việc tối ưu mô hình theo hướng nhẹ hơn nhưng vẫn giữ được độ chính xác cao. Nổi bật trong số đó là nghiên cứu của Akwaronwu và cộng sự (2025), nơi nhóm tác giả đề xuất mô hình CNN kết hợp với kỹ thuật tăng cường dữ liệu SMOTE nhằm khắc phục vấn đề mất cân bằng trong tập dữ liệu huấn luyện. Mô hình được kiểm thử trên tập NF-UQ-NIDS và đạt độ chính xác lên tới 99.82%, với recall đạt 99.95%, cho thấy hiệu quả vượt trội trong phát hiện tấn công Brute Force ngay cả khi lượng dữ liệu tấn công chiếm tỷ lệ nhỏ trong toàn bộ lưu lượng mạng.

Bên cạnh đó, một cách tiếp cận khá mới mẻ và linh hoạt là mô hình Deep Shallow Network with LSTM (2025). Mô hình này khai thác ưu điểm của cả hai hướng tiếp cận: mạng nông giúp xử lý nhanh và tiết kiệm tài nguyên, còn LSTM lại có khả năng ghi nhớ dài hạn và nhận diện chuỗi sự kiện bất thường theo thời gian. Đặc biệt, nhóm nghiên cứu này còn lồng ghép yếu tố bảo mật dữ liệu và quyền riêng tư vào thiết kế mô hình, hướng đến khả năng triển khai trong thực tế mà không yêu cầu thu thập dữ liệu toàn bộ từ người dùng – một điểm cộng lớn trong bối cảnh hiện nay khi quyền riêng tư trở thành mối quan tâm hàng đầu.

Từ tổng quan các công trình trên, có thể thấy rằng việc phát hiện tấn công Brute Force không chỉ dừng lại ở việc phân loại đúng hay sai, mà còn đòi hỏi mô hình phải có khả năng xử lý theo thời gian thực, thích ứng với đặc thù thiết bị và bối cảnh mạng khác nhau. Các mô hình học máy cổ điển như SVM vẫn chứng minh được giá trị nhất định, nhưng dần dần nhường chỗ cho các kiến trúc học sâu như RNN, CNN hay LSTM, nhờ vào khả năng học đặc trưng sâu và linh hoạt hơn. Đặc biệt, xu hướng tích hợp nhiều thành phần – từ giảm chiều dữ liệu, xử lý mất cân bằng, cho đến tối ưu quyền riêng tư – đang mở ra một hướng đi mới cho các hệ thống phát hiện tấn công thông minh và hiệu quả hơn trong môi trường IoT.

### 2.4.3 Các công trình ứng dụng Federated Learning trong IoT

Trong bối cảnh hệ thống IoT ngày càng phổ biến và trở thành một phần thiết yếu trong nhiều lĩnh vực như nhà thông minh, y tế, giao thông hay công nghiệp 4.0, vấn đề bảo vệ an toàn cho dữ liệu và hệ thống trở nên cấp thiết hơn bao giờ hết. Tuy nhiên, đặc trưng phân tán và hạn chế về tài nguyên của các thiết bị IoT đã khiến việc triển khai các mô hình học máy truyền thống – vốn đòi hỏi dữ liệu phải được tập trung để huấn luyện – trở thành một thách thức lớn. Chính trong hoàn cảnh đó, học liên kết (Federated Learning - FL) được xem là một giải pháp đột phá, giúp khắc phục được những hạn chế nêu trên đồng thời vẫn đảm bảo hiệu quả trong việc phát hiện các mối đe dọa an ninh mạng.

Khác với mô hình học tập trung, FL cho phép các thiết bị “học” trực tiếp trên dữ liệu cục bộ mà không cần gửi dữ liệu đó ra khỏi thiết bị. Chỉ những tham số mô hình được cập nhật mới được chia sẻ để tổng hợp tại máy chủ trung tâm, qua đó vừa tiết kiệm băng thông, vừa bảo mật được thông tin nhạy cảm. Điều này đặc biệt phù hợp với môi trường IoT, nơi mà mỗi thiết bị chỉ thu thập một phần nhỏ dữ liệu và khả năng xử lý không cao, nhưng lại nắm giữ các thông tin riêng tư có giá trị.

Một trong những nghiên cứu mang tính nền tảng trong lĩnh vực này là công trình của McMahan và cộng sự (2017), nơi thuật toán FedAvg được giới thiệu. Đây là thuật toán trung bình hóa các tham số mô hình từ các thiết bị cục bộ sau mỗi vòng huấn luyện để xây dựng mô hình toàn cục. Mặc dù công trình này không đi sâu vào lĩnh vực an ninh mạng, nhưng lại tạo tiền đề vững chắc để các nhà nghiên cứu sau này triển khai FL vào các bài toán phát hiện xâm nhập trong IoT – nơi dữ liệu nhạy cảm và phân tán là điều không thể tránh khỏi.

Tiếp theo đó, vào năm 2020, Aledhari và cộng sự đã triển khai một hệ thống phát hiện tấn công mạng phân tán dựa trên FL, ứng dụng trên các tập dữ liệu phổ biến như BoT-IoT và CICIDS2018. Mô hình cho thấy hiệu quả rõ rệt trong việc phân biệt lưu lượng bất thường và bình thường mà vẫn đảm bảo quyền riêng tư, điều mà các mô hình học máy tập trung thường phải đánh đổi. Ngoài ra, khả năng huấn luyện song song trên nhiều thiết bị giúp giảm thời gian xây dựng mô hình và mở rộng quy mô hệ thống dễ dàng hơn.

Trong một hướng tiếp cận khác, nghiên cứu của T. Truong và đồng nghiệp (2022) đã kết hợp Federated Learning với mô hình LSTM để nhận diện các cuộc tấn công DDoS trên hệ thống IoT. Với khả năng xử lý dữ liệu theo chuỗi thời gian, LSTM phù hợp để phát hiện những mẫu tấn công kéo dài hoặc xảy ra theo từng đợt – vốn là đặc trưng của các cuộc tấn công từ chối dịch vụ phân tán. Khi triển khai mô hình FL-LSTM trên một môi trường giả lập phân tán, nhóm nghiên cứu ghi nhận được kết quả tích cực về độ chính xác và độ trễ phản hồi, đồng thời giảm thiểu được lượng dữ liệu phải truyền tải về máy chủ, giúp tiết kiệm tài nguyên mạng.

Đặc biệt, trong bối cảnh computing at the edge (tính toán tại thiết bị biên) ngày càng trở nên khả thi nhờ sự phát triển của phần cứng và hệ điều hành nhúng, việc kết hợp FL với các thiết bị đầu cuối đang mở ra cơ hội hiện thực hóa các hệ thống phát hiện xâm nhập phân tán, phản ứng nhanh, bảo mật cao và có thể triển khai ở quy mô lớn. Thay vì phụ thuộc vào máy chủ trung tâm, mỗi thiết bị có thể trở thành một "trạm bảo vệ" thông minh, vừa phát hiện tấn công, vừa bảo vệ dữ liệu của chính mình.

Năm 2025, xu hướng ứng dụng Federated Learning trong an ninh mạng IoT tiếp tục được đẩy mạnh với nhiều công trình nổi bật và có tính ứng dụng cao. Cụ thể:

Albogami (2025) đề xuất mô hình IDFLM-ES (Intelligent Deep Federated Learning Model for Edge Security) nhằm tăng cường khả năng phát hiện xâm nhập trong môi trường điện toán biên. Mô hình này kết hợp giữa mạng niềm tin sâu (Deep Belief Network) và các kỹ thuật tối ưu hóa tiên tiến để phát hiện chính xác các hành vi bất thường mà không làm ảnh hưởng đến quyền riêng tư dữ liệu. Kết quả thử nghiệm chỉ ra rằng phương pháp này đạt hiệu quả cao trong nhận diện tấn công đồng thời duy trì hiệu suất xử lý ổn định trên các thiết bị biên có hạn chế về tài nguyên tính toán

Trong cùng năm, Rahmati giới thiệu một khung bảo mật IoT ứng dụng Federated Learning với lõi là mạng nơ-ron hồi tiếp (RNN). Hệ thống có khả năng phát hiện các mối đe dọa trong thời gian thực với độ chính xác vượt ngưỡng 98%, đặc biệt trong các tình huống tấn công phức tạp như DDoS. Ngoài ra, khung này còn chứng minh được ưu điểm vượt trội trong việc giảm thiểu tiêu thụ tài nguyên hệ thống, giúp dễ dàng triển khai trên các nền tảng IoT hiện hữu

Một nghiên cứu khác được công bố trên tạp chí Frontiers in Big Data (2025) phát triển một khung học liên kết dành riêng cho môi trường IoT, sử dụng mô hình Time-Triggered Federated (TTF) kết hợp với thuật toán tối ưu hóa EEFO (Enhanced Elephant Foraging Optimization). Giải pháp này giúp cải thiện tốc độ và độ ổn định trong quá trình hội tụ mô hình, đồng thời tăng khả năng phát hiện các kiểu tấn công tinh vi, đặc biệt là các mối đe dọa có hành vi thay đổi theo thời gian hoặc sử dụng kỹ thuật che giấu thông minh

Các công trình gần đây đã cho thấy Federated Learning không đơn thuần là một xu hướng học máy mang tính thời sự, mà thực sự là một chiến lược hiệu quả và bền vững trong việc bảo vệ hệ thống IoT trước các mối đe dọa ngày càng phức tạp. Việc cho phép các thiết bị học từ dữ liệu cục bộ mà không cần chia sẻ thông tin nhạy cảm đang tạo nên bước tiến mới trong việc cân bằng giữa bảo mật và hiệu suất, mở ra tiềm năng ứng dụng rộng lớn trong tương lai gần.

# CHƯƠNG 3: ỨNG DỤNG FEDERATED LEARNING PHÁT HIỆN TẤN CÔNG MẠNG

## 3.1. Phân tích tập dữ liệu phát hiện tấn công bằng Federated Learning.

### 3.1.1 Tập dữ liệu CICIOT2023.

**Bộ dữ liệu CICIoT2023**, được công bố trên nền tảng **Kaggle** thông qua địa chỉ <https://www.kaggle.com/datasets/akashdogra/cic-iot-2023>, là một tập dữ liệu tiêu chuẩn do **Canadian Institute for Cybersecurity (CIC)** phát hành nhằm phục vụ mục đích nghiên cứu và phát triển các hệ thống phát hiện tấn công mạng trong môi trường Internet of Things (IoT).Đây là một trong những bộ dữ liệu IoT lớn và đa dạng, bao gồm cả các lưu lượng hợp lệ (Benign) lẫn các loại tấn công khác nhau.

Tổng số tệp dữ liệu trong CICIoT2023 là 168 tệp với định dạng .csv, mỗi tệp đại diện cho một phần nhỏ của toàn bộ lưu lượng được thu thập theo thời gian. Trong phạm vi đồ án này, một tệp duy nhất đã được sử dụng làm mẫu dữ liệu thử nghiệm,

Dữ liệu bao gồm 47 thuộc tính, trong đó có 46 đặc trưng dạng số (float64) và 1 thuộc tính nhãn (label) dạng chuỗi. Tất cả các giá trị đều đầy đủ, không có dữ liệu bị thiếu (missing values), điều này rất thuận lợi cho việc huấn luyện mô hình.

Dữ liệu trong tập CICIoT2023 được tổ chức thành các dòng, mỗi dòng tương ứng với một dòng chảy (network flow) được trích xuất từ lưu lượng mạng thực tế. Mỗi dòng chảy được mô tả bằng nhiều đặc trưng khác nhau phản ánh hành vi truyền tải dữ liệu giữa các thiết bị mạng. Đây là những đặc trưng được thiết kế một cách có chủ đích nhằm hỗ trợ quá trình phân tích, nhận diện các loại hình tấn công mạng với độ chính xác cao.

Từ toàn bộ các thuộc tính có trong tập dữ liệu, một tập hợp gồm 46 đặc trưng đã được chọn lọc để sử dụng làm đầu vào cho mô hình học máy. Nhãn phân loại label được dùng làm đầu ra cho bài toán phân lớp. Các đặc trưng đầu vào được đặt trong biến X\_columns, còn biến mục tiêu là y\_column. Dưới đây là mô tả chi tiết về các đặc trưng này theo từng nhóm chức năng

* Nhóm đặc trưng về thời gian và tốc độ truyền tải

Các đặc trưng flow\_duration và Duration lần lượt thể hiện tổng thời gian và độ dài dòng chảy. Rate, Srate và Drate đại diện cho tốc độ truyền dữ liệu tổng thể, tốc độ gửi và tốc độ nhận, giúp mô hình phát hiện các bất thường như lưu lượng tăng đột biến hoặc dòng chảy bị gián đoạn không đều

* Nhóm đặc trưng về cờ điều khiển trong giao thức TCP

Các đặc trưng như syn\_flag\_number, ack\_flag\_number, rst\_flag\_number, fin\_flag\_number, psh\_flag\_number, ece\_flag\_number và cwr\_flag\_number biểu thị số lần xuất hiện của các cờ điều khiển trong dòng chảy TCP.  
Các đặc trưng ack\_count, syn\_count, fin\_count, urg\_count và rst\_count ghi nhận số lượng gói tin có chứa từng loại cờ tương ứng. Những cờ này đóng vai trò then chốt trong việc phân biệt hành vi hợp lệ và hành vi khai thác lỗ hổng giao thức

* Nhóm đặc trưng phản ánh các giao thức mạng được sử dụng

Tập dữ liệu ghi nhận sự xuất hiện của nhiều loại giao thức thông qua các đặc trưng nhị phân như HTTP, HTTPS, DNS, Telnet, SMTP, SSH, IRC, TCP, UDP, DHCP, ARP, ICMP, IPv và LLC.  
Sự hiện diện của một hoặc nhiều giao thức trong dòng chảy có thể cho thấy đặc điểm của một loại hình tấn công cụ thể. Ví dụ, các cuộc tấn công brute-force thường xuất hiện trên giao thức SSH hoặc Telnet, trong khi tấn công DDoS có thể sử dụng UDP hoặc ICMP

* Nhóm đặc trưng thống kê về số lượng và kích thước gói tin

Các đặc trưng Tot sum, Tot size, Min, Max, AVG và Std cung cấp các thông tin thống kê cơ bản như tổng số byte, kích thước nhỏ nhất, lớn nhất, trung bình và độ lệch chuẩn trong dòng chảy  
Những đặc trưng này cho phép mô hình học máy nhận diện các kiểu tấn công có mẫu hình kích thước bất thường, ví dụ như các gói tin rất nhỏ được gửi với tần suất cao hoặc ngược lại

* Nhóm đặc trưng về đặc tính toán học và đo lường hành vi

Đặc trưng IAT phản ánh thời gian giữa hai gói tin liên tiếp, là yếu tố giúp phát hiện các hành vi như quét mạng hoặc truyền thông tin ngắt quãng có chủ đích  
Các đặc trưng Number, Magnitue, Radius, Covariance, Variance và Weight được trích xuất từ các phép đo thống kê và phân tích hình học, cung cấp cái nhìn sâu hơn về cấu trúc và phân bố của dòng chảy dữ liệu

* Đặc trưng nhãn đầu ra

Thuộc tính label là trường phân loại được sử dụng làm đầu ra trong mô hình. Mỗi bản ghi tương ứng với một trong nhiều loại lưu lượng, có thể là hợp lệ (Benign) hoặc là một loại tấn công cụ thể. Trong phạm vi đồ án này, một số nhãn tiêu biểu được đưa vào huấn luyện bao gồm DDoS-RSTFINFlood, DoS-TCP\_Flood, DDoS-ICMP\_Flood, DoS-UDP\_FloodvàoS-SYN\_Flood.  
Đây đều là những cuộc tấn công có mục đích gây gián đoạn dịch vụ, làm nghẽn hệ thống hoặc chiếm dụng tài nguyên, rất phổ biến trong môi trường mạng IoT

Do đây là bài toán phân loại đa lớp (multi-class classification), mô hình cần có khả năng học được đặc điểm phân biệt của từng loại nhãn. Điều này đòi hỏi dữ liệu phải được tiền xử lý cẩn thận và mô hình phải được thiết kế với kiến trúc đủ phức tạp để phân biệt nhiều lớp khác nhau

### 3.1.2 Tiền xử lý tập dữ liệu CICIOT2023.

Trước khi tiến hành huấn luyện mô hình học máy trong môi trường học liên kết (Federated Learning), dữ liệu từ tập CICIoT2023 cần được xử lý và chuẩn hóa một cách hệ thống để đảm bảo tính đồng nhất, độ chính xác và tối ưu hóa khả năng học của mô hình. Dữ liệu ban đầu mặc dù đã được định dạng sẵn nhưng vẫn tồn tại các đặc trưng cần sàng lọc, định dạng lại hoặc mã hóa để phù hợp với mô hình phân loại đa lớp. Quá trình tiền xử lý trong nghiên cứu này bao gồm sáu bước chính, được mô tả như sau:

**Bước 1: Đọc dữ liệu từ tệp CSV**

Tập dữ liệu CICIoT2023 được lưu trữ dưới dạng các tệp CSV, mỗi tệp tương ứng với một phần dòng chảy mạng đã thu thập. Trong quá trình huấn luyện và kiểm thử mô hình, mỗi node (client) trong hệ thống Federated Learning sẽ làm việc với một tệp riêng biệt. Ở bước tiền xử lý này, một tệp CSV mẫu đã được lựa chọn để thực hiện thử nghiệm trước. Việc đọc tệp CSV được thực hiện bằng thư viện pandas, giúp chuyển dữ liệu từ định dạng bảng tĩnh thành DataFrame linh hoạt cho việc xử lý tiếp theo.



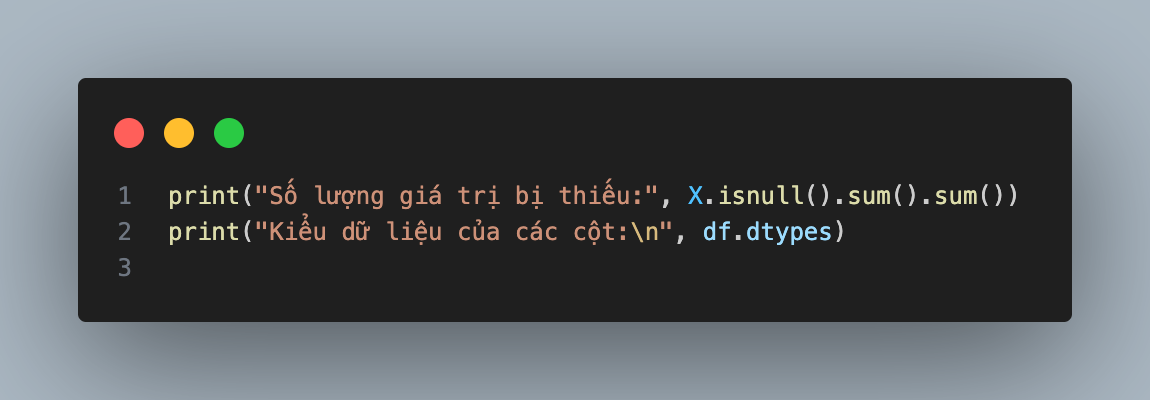
Bước 2: Lọc các đặc trưng đầu vào và nhãn mục tiêu

Tập dữ liệu gốc bao gồm hơn 80 thuộc tính, trong đó có nhiều thuộc tính không cần thiết hoặc trùng lặp về mặt thông tin. Do đó, một danh sách gồm 46 thuộc tính đặc trưng đã được lựa chọn kỹ lưỡng để phục vụ cho quá trình huấn luyện mô hình. Danh sách này bao gồm các thông số về thời gian dòng chảy, tốc độ truyền nhận, các cờ điều khiển của giao thức TCP, loại giao thức mạng, thống kê về kích thước gói tin và các chỉ số toán học dùng để mô tả hành vi. Biến mục tiêu là trường label, chứa loại tấn công tương ứng với mỗi dòng chảy mạng



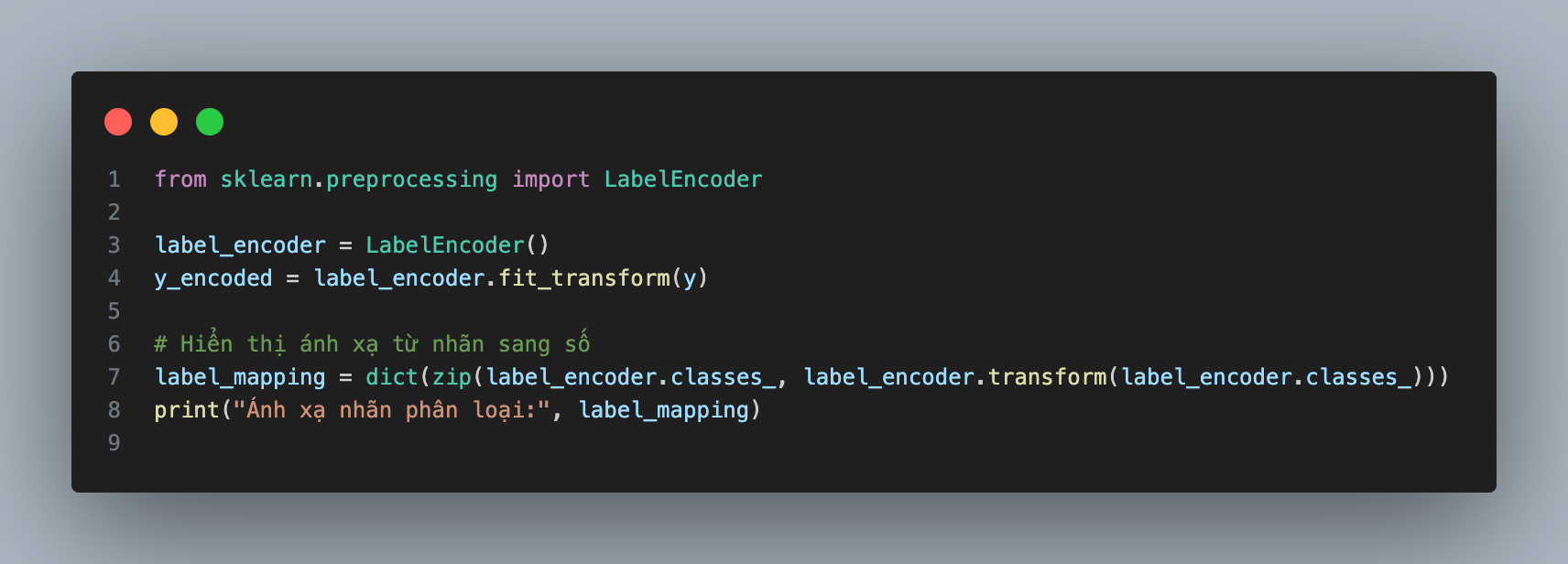
Bước 3: Kiểm tra dữ liệu thiếu và định dạng dữ liệu

Trước khi đưa vào mô hình học máy, cần xác nhận rằng dữ liệu không bị thiếu (null) cũng như các thuộc tính đang có kiểu dữ liệu phù hợp. Dữ liệu thiếu có thể gây ảnh hưởng nghiêm trọng đến độ chính xác của mô hình, và việc định dạng sai kiểu dữ liệu có thể khiến thuật toán học không hoạt động đúng. Trong tập dữ liệu CICIoT2023, kết quả kiểm tra cho thấy không có giá trị nào bị thiếu và toàn bộ các cột đều có kiểu dữ liệu phù hợp, chủ yếu là dạng số thực (float64).



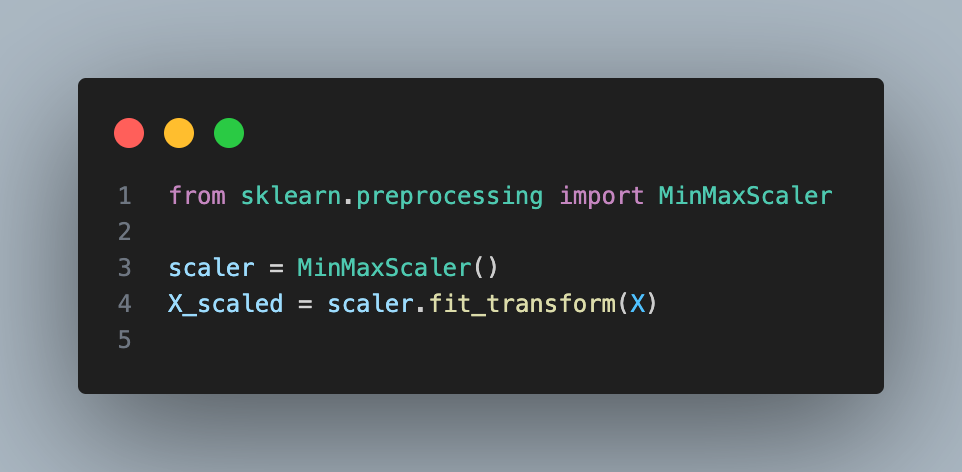
Bước 4: Mã hóa nhãn phân loại

Mô hình học máy không thể xử lý trực tiếp các chuỗi văn bản, do đó trường label (vốn chứa tên các loại tấn công như DDoS-ICMP\_Flood, DoS-TCP\_Flood, hoặc BenignTraffic) cần được chuyển sang dạng số nguyên. Để thực hiện điều này, kỹ thuật Label Encoding được áp dụng bằng cách sử dụng lớp LabelEncoder trong thư viện scikit-learn. Mỗi nhãn phân loại được ánh xạ về một chỉ số duy nhất, giúp cho mô hình dễ dàng xử lý trong bước huấn luyện.



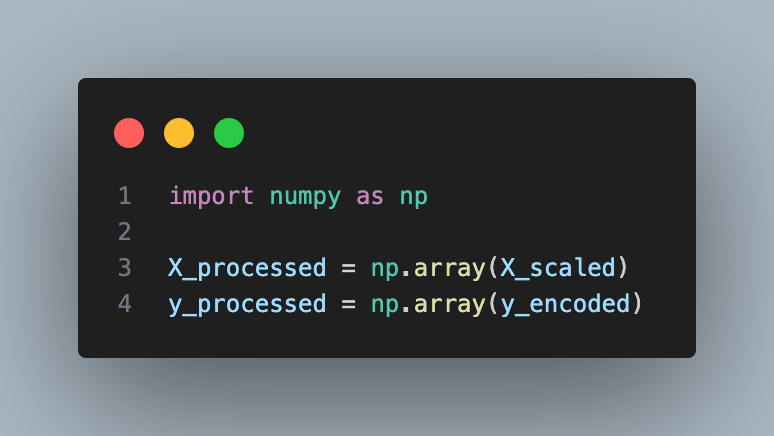
Bước 5: Chuẩn hóa dữ liệu đầu vào

Do các thuộc tính đầu vào có thang đo và đơn vị khác nhau (ví dụ như thời gian tính bằng micro-giây, tỷ lệ gói tính theo phần trăm hoặc byte), cần phải chuẩn hóa chúng để đảm bảo rằng mỗi đặc trưng đóng góp công bằng vào quá trình học. Chuẩn hóa cũng giúp thuật toán hội tụ nhanh hơn. Phương pháp chuẩn hóa được sử dụng là Min-Max Scaling, giúp đưa toàn bộ giá trị về khoảng [0, 1].



Bước 6: Chuyển dữ liệu về định dạng NumPy

Sau khi hoàn tất chuẩn hóa và mã hóa, dữ liệu được chuyển đổi từ DataFrame sang định dạng NumPy array. Đây là định dạng chuẩn cho các mô hình học máy vì giúp tăng tốc xử lý và tương thích với các thư viện như TensorFlow, PyTorch hoặc scikit-learn.



## 3.2. Xây dựng Federated Learning phát hiện tấn công.

Sau khi hoàn tất các bước tiền xử lý dữ liệu, bước tiếp theo trong quá trình phát triển hệ thống là xây dựng mô hình Federated Learning (FL) nhằm phát hiện tấn công mạng một cách phân tán. Mục tiêu của Federated Learning là huấn luyện mô hình học máy một cách hiệu quả mà không cần tập trung toàn bộ dữ liệu tại một máy chủ trung tâm, từ đó đảm bảo tính riêng tư và bảo mật của dữ liệu người dùng – điều rất quan trọng trong bối cảnh IoT.

### 3.2.1 Phân chia dữ liệu thành các client

Trong hệ thống Federated Learning (FL) áp dụng cho bài toán phát hiện tấn công mạng, việc phân chia dữ liệu thành các phần nhỏ tương ứng với từng client là một bước khởi đầu rất quan trọng. Mục tiêu chính của bước này là mô phỏng một môi trường phân tán nơi mỗi thiết bị đầu cuối (client) chỉ có quyền truy cập vào một phần cục bộ của toàn bộ tập dữ liệu gốc. Việc giới hạn dữ liệu truy cập của client mang ý nghĩa then chốt trong việc đảm bảo tính riêng tư, đồng thời phản ánh đúng thực tiễn hoạt động của các thiết bị IoT ngoài đời thật, nơi dữ liệu không thể hoặc không nên được tập trung về máy chủ trung tâm.

Quá trình chia dữ liệu được triển khai trên mã nguồn bằng cách cấu hình biến METHOD. Đây là tham số điều khiển chiến lược chia dữ liệu, có thể nhận một trong các giá trị: 'STRATIFIED', 'LEAVE\_ONE\_OUT', 'ONE\_CLASS', và 'HALF\_BENIGN'. Mỗi chiến lược thể hiện một cách tiếp cận khác nhau đối với việc tạo dựng tập dữ liệu phân tán, từ đó dẫn đến những hành vi học máy và đặc điểm hội tụ mô hình hoàn toàn khác nhau.

Với phương pháp 'STRATIFIED', dữ liệu được phân chia sao cho mỗi client nhận được một tập con có phân phối nhãn tương tự như tập dữ liệu gốc. Điều này có nghĩa là mỗi client đều sẽ sở hữu cả mẫu lưu lượng hợp lệ (benign) lẫn các mẫu lưu lượng bị tấn công (attack). Phương pháp này giúp tăng khả năng hội tụ nhanh của mô hình FL, vì mỗi client đều có thông tin đại diện cho toàn bộ các loại hành vi mạng. Để hình dung rõ hơn về sự phân bố này, bạn có thể tìm kiếm hình ảnh với từ khóa "stratified sampling in federated learning" để thấy trực quan cách các nhãn được phân phối đều vào các client.

Ở phương pháp 'LEAVE\_ONE\_OUT', chiến lược chia khác biệt hơn. Mỗi client sẽ bị loại bỏ dữ liệu liên quan đến một loại nhãn tấn công cụ thể, trong khi các client còn lại vẫn giữ nhãn đó. Điều này giúp đánh giá mức độ tổng quát hóa của hệ thống FL – tức là liệu mô hình được huấn luyện trên các client khác có khả năng phát hiện các loại tấn công chưa từng thấy ở client cụ thể đó hay không. Đây là một kịch bản quan trọng trong thực tế khi một thiết bị chỉ có thể quan sát được các kiểu hành vi mạng cụ thể, nhưng hệ thống vẫn cần đảm bảo phát hiện những bất thường khác biệt chưa từng thấy.

Phương pháp 'ONE\_CLASS' đưa ra một kịch bản cực đoan hơn khi mỗi client chỉ có dữ liệu của một loại nhãn duy nhất, ví dụ chỉ toàn dữ liệu hợp lệ hoặc chỉ dữ liệu của một loại tấn công nhất định. Mục tiêu của cách chia này là phục vụ việc huấn luyện mô hình phát hiện bất thường (anomaly detection), nơi hệ thống cần học được "bình thường là như thế nào", sau đó dựa vào sự khác biệt để phát hiện tấn công. Một hình ảnh minh họa có thể tìm kiếm để hỗ trợ hiểu phương pháp này là "federated learning one-class client data distribution".

Với chiến lược 'HALF\_BENIGN', các client được chia thành hai nhóm: nhóm đầu tiên chỉ nhận dữ liệu hợp lệ, nhóm còn lại nhận cả dữ liệu hợp lệ lẫn tấn công. Chiến lược này phản ánh rõ sự dị biệt giữa các nguồn dữ liệu: một số client hoàn toàn sạch, trong khi số khác có chứa thông tin nguy cơ. Đây là một mô hình phù hợp khi triển khai hệ thống FL trên các thiết bị IoT ngoài thực tế, nơi một số thiết bị được giám sát kỹ lưỡng và an toàn, còn số khác có thể tiếp xúc với các vùng mạng nguy hiểm. Cách chia này đồng thời kiểm tra khả năng cộng tác và lan truyền kiến thức giữa các client khi một nhóm không có dữ liệu tấn công nhưng vẫn cần phát hiện được các mối đe dọa thông qua việc học chung.

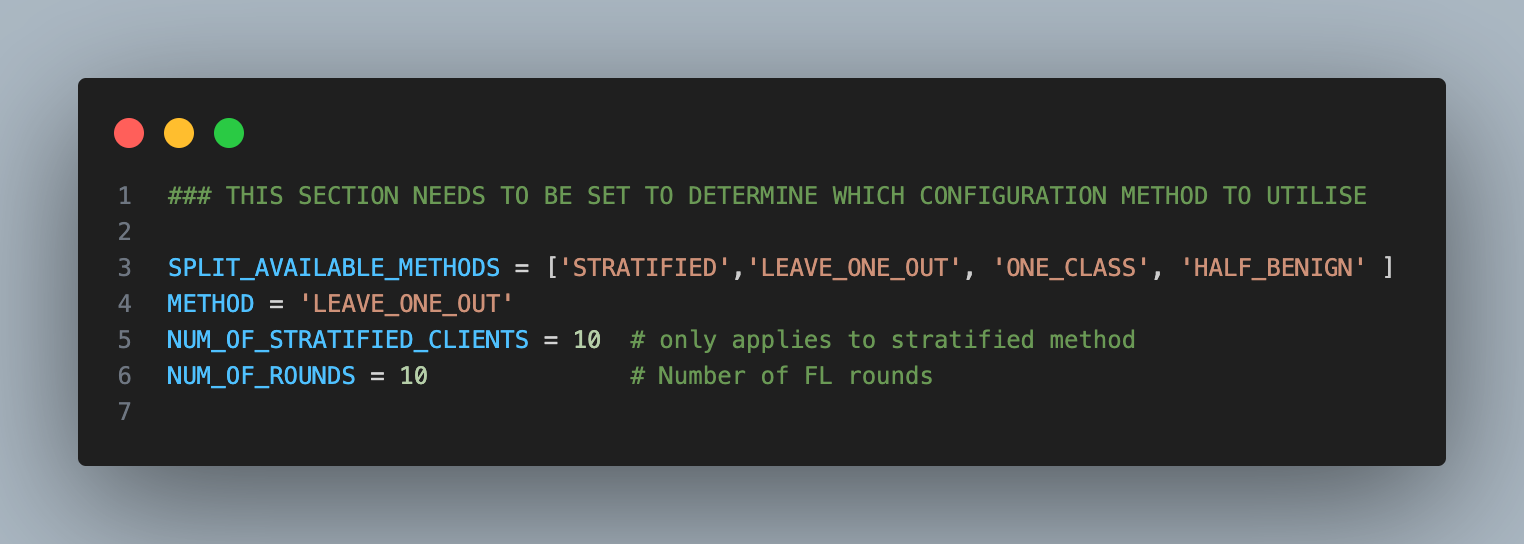
Số lượng client được sử dụng trong hệ thống FL không cố định mà phụ thuộc vào loại mô hình phân loại đang được áp dụng. Ví dụ, trong chế độ binary\_classifier, hệ thống mặc định sử dụng 10 client, trong đó 5 client chỉ chứa dữ liệu hợp lệ (benign) và 5 client chứa cả benign và dữ liệu tấn công. Trong khi đó, các chế độ phân loại khác như individual hay group có thể sử dụng số lượng client linh hoạt hơn, phù hợp với số lớp hoặc nhóm tấn công cụ thể mà bài toán đang nhắm đến. Do đó, việc xác định số lượng client không chỉ là vấn đề kỹ thuật về phân chia dữ liệu, mà còn ảnh hưởng trực tiếp đến cấu trúc của mô hình và hiệu quả huấn luyện tổng thể.

### 3.2.2 Cấu hình thuật toán Federated Learning

Hệ thống Federated Learning trong đồ án này được thiết kế hoạt động theo vòng lặp (communication rounds) giữa một máy chủ trung tâm và nhiều client. Mỗi vòng lặp mô phỏng một chu kỳ huấn luyện phân tán, trong đó mô hình được huấn luyện tại client và sau đó được đồng bộ hóa lại tại server.

Quy trình diễn ra như sau: đầu tiên, máy chủ khởi tạo một mô hình toàn cục (global model). Mô hình này được gửi tới tất cả các client tham gia. Tại mỗi client, mô hình được huấn luyện cục bộ trên dữ liệu riêng trong một số epoch nhất định. Sau đó, các client gửi lại trọng số mô hình đã được huấn luyện về cho máy chủ. Máy chủ sẽ tiến hành tổng hợp các trọng số nhận được từ client bằng thuật toán Federated Averaging (FedAvg), tạo ra một mô hình toàn cục mới. Quy trình này lặp lại qua nhiều vòng để mô hình dần được cải thiện và hội tụ.

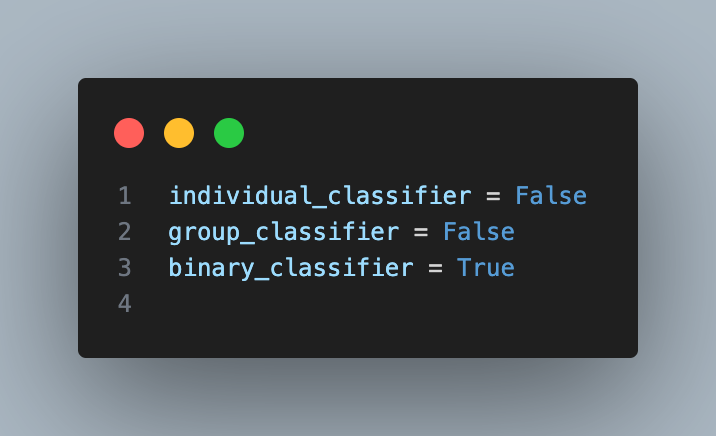
Số vòng lặp trong quá trình huấn luyện được điều chỉnh bằng biến NUM\_OF\_ROUNDS, có thể thay đổi tùy theo yêu cầu thử nghiệm.



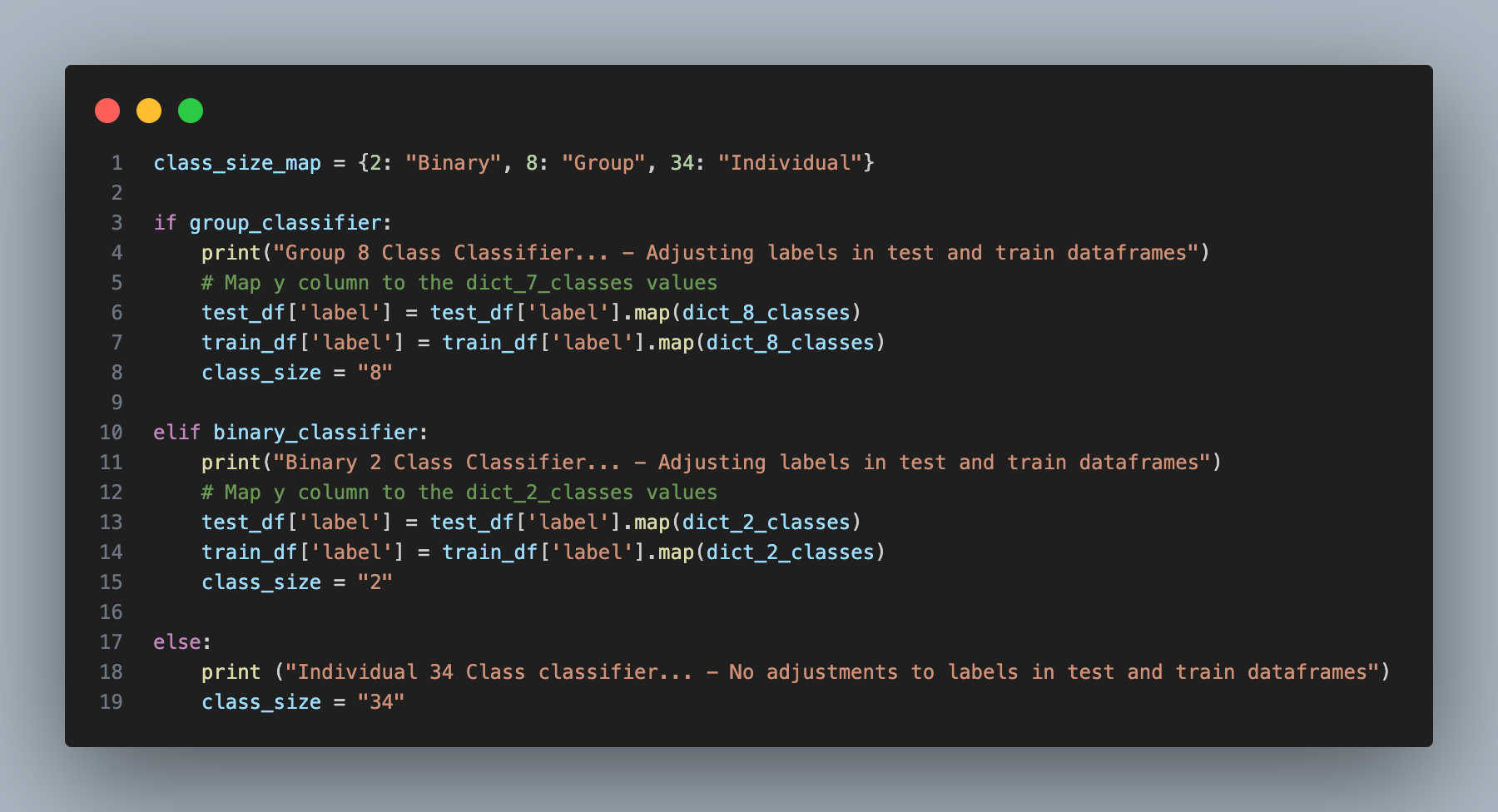
Mỗi client sẽ huấn luyện mô hình trên tập dữ liệu riêng biệt và không chia sẻ dữ liệu gốc. Điều này đáp ứng yêu cầu bảo mật và riêng tư trong môi trường IoT phân tán.

Bên cạnh đó, mô hình phân loại cũng được cấu hình phù hợp với nhiều mục đích khác nhau, hệ thống hỗ trợ ba dạng mô hình như sau:

* **Phân loại nhị phân (Binary Classifier)** – Phân biệt giữa lưu lượng hợp lệ và tấn công.
* **Phân loại theo nhóm (Group Classifier)** – Gom các loại tấn công thành nhóm và phân loại theo nhóm.
* **Phân loại từng loại (Individual Classifier)** – Phân loại riêng biệt cho từng loại tấn công cụ thể.



Dựa trên cấu hình này, hệ thống sẽ điều chỉnh nhãn (label) trong tập dữ liệu huấn luyện và kiểm tra sao cho tương ứng với từng chiến lược. Ví dụ, với mô hình phân loại nhị phân, các nhãn sẽ được ánh xạ thành Benign hoặc Attack



### 3.2.3 Thuật toán Federated Learning phát hiện tấn công.

Hệ thống Federated Learning trong đồ án được triển khai bằng thư viện **Flower (flwr)** để mô phỏng quá trình huấn luyện phân tán với nhiều client đại diện cho các tổ chức hoặc thiết bị IoT riêng biệt. Mỗi client thực hiện quá trình huấn luyện mô hình học máy trên dữ liệu cục bộ và chỉ gửi trọng số (weights) đã huấn luyện về server trung tâm. Điều này giúp đảm bảo tính riêng tư, giảm chi phí truyền tải dữ liệu, và phù hợp với môi trường IoT.

1. Khởi tạo mô hình toàn cục tại server

Trong hệ thống Federated Learning được xây dựng, bước đầu tiên là khởi tạo một mô hình học sâu (neural network) tại máy chủ trung tâm (server). Mô hình này đóng vai trò là mô hình toàn cục (global model) được phân phối đến tất cả các client trong mỗi vòng huấn luyện. Mô hình tại server và các client cần có kiến trúc giống hệt nhau để đảm bảo tính tương thích khi trao đổi trọng số trong quá trình đồng bộ hóa.

Mô hình được xây dựng bằng thư viện **Keras** thuộc TensorFlow, dưới dạng Sequential. Kiến trúc mô hình bao gồm ba tầng chính. Tầng đầu tiên là Flatten, có nhiệm vụ chuyển đổi dữ liệu đầu vào dạng ma trận thành một vector một chiều, giúp xử lý dữ liệu thuận tiện hơn cho các lớp Dense phía sau. Dữ liệu được giả định có kích thước (features, 1), trong đó features là số đặc trưng đầu vào (ví dụ 79 nếu dùng dữ liệu từ tập NSL-KDD hoặc similar IoT dataset).

Tiếp theo là hai lớp Dense dùng làm tầng ẩn, với số lượng nút lần lượt là 50 và 25. Hàm kích hoạt ReLU được sử dụng để tăng tính phi tuyến và tránh hiện tượng vanishing gradient trong quá trình huấn luyện. Cuối cùng là một lớp Dense đầu ra có số nút bằng với số lượng lớp trong bài toán phân loại, kèm theo hàm kích hoạt softmax để đưa ra xác suất dự đoán trên từng lớp.

Sau khi xây dựng xong kiến trúc, mô hình được biên dịch với trình tối ưu Adam, hàm mất mát sparse\_categorical\_crossentropy (do nhãn ở dạng số nguyên), và chỉ số đánh giá là accuracy. Việc lựa chọn cấu hình này phù hợp với các bài toán phân loại nhiều lớp và dữ liệu nhãn rời rạc.

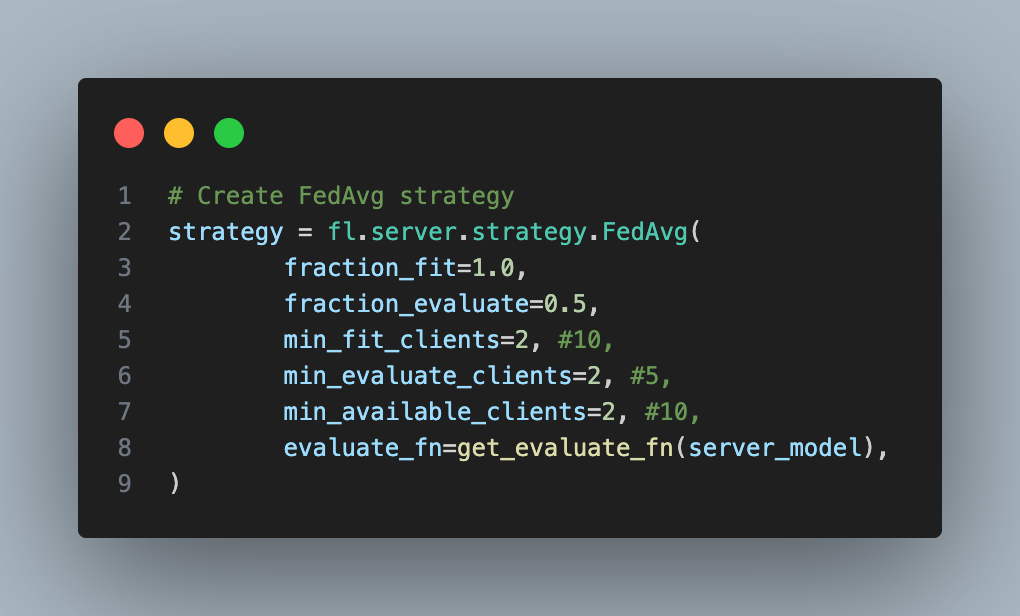


fl\_X\_train[0].shape[1] là số đặc trưng (feature) trong dữ liệu huấn luyện đã được chia cho các client. Biến num\_unique\_classes xác định số lượng lớp phân loại, có thể là 2 (binary), 7–8 (group) hoặc 34 (individual).

2. Khởi tạo chiến lược FedAvg để đồng bộ mô hình

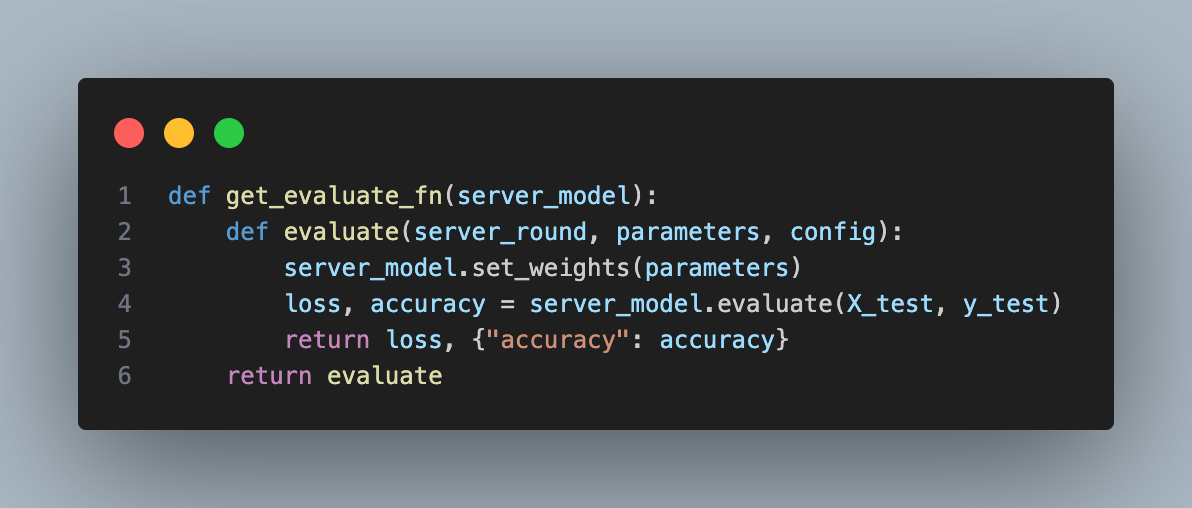
Sau khi mô hình toàn cục tại server được khởi tạo và biên dịch, bước tiếp theo là thiết lập chiến lược huấn luyện phân tán trong môi trường Federated Learning. Trong đồ án này, thuật toán đồng bộ mô hình được sử dụng là **Federated Averaging (FedAvg)**, một trong những phương pháp tiêu chuẩn và phổ biến trong FL. FedAvg có nhiệm vụ cập nhật mô hình toàn cục tại server bằng cách trung bình hóa trọng số từ các mô hình cục bộ của client sau mỗi vòng huấn luyện.

Để hiện thực điều này, thư viện Flower cung cấp sẵn một class tên là fl.server.strategy.FedAvg, cho phép cấu hình các tham số liên quan đến quá trình chọn client, tỉ lệ tham gia, cũng như cách đánh giá mô hình.



Trong đó, fraction\_fit=1.0 nghĩa là 100% client hiện có sẽ được chọn để huấn luyện trong mỗi vòng. fraction\_evaluate=0.5 chỉ định rằng chỉ một nửa số client sẽ tham gia đánh giá mô hình sau khi huấn luyện xong. Các tham số min\_fit\_clients, min\_evaluate\_clients và min\_available\_clients thiết lập số lượng tối thiểu client cần thiết để tiến hành một vòng cập nhật mô hình, đảm bảo mô phỏng hoạt động ổn định trong điều kiện client có thể rớt mạng hoặc không phản hồi.

Tham số evaluate\_fn trỏ đến một hàm tùy chỉnh được định nghĩa để đánh giá mô hình toàn cục tại server sau mỗi vòng lặp. Hàm này nhận vào mô hình đã cập nhật và tính toán độ chính xác và hàm mất mát trên tập kiểm tra toàn cục.



Hàm get\_evaluate\_fn() nhận vào mô hình server và trả về một hàm con evaluate(). Hàm này được Flower gọi sau mỗi vòng huấn luyện để đo hiệu suất mô hình, từ đó giúp theo dõi tiến trình hội tụ và hiệu quả tổng thể của hệ thống.

Nhờ cơ chế đồng bộ của FedAvg, server có thể tổng hợp kiến thức từ tất cả các client mà không cần truy cập dữ liệu gốc. Điều này đảm bảo tính riêng tư, đồng thời vẫn cho phép huấn luyện một mô hình hiệu quả và toàn diện.

3. Khởi tạo client và huấn luyện cục bộ

Sau khi server được cấu hình với mô hình toàn cục và chiến lược đồng bộ FedAvg, bước tiếp theo trong hệ thống Federated Learning là khởi tạo các client. Trong mô hình FL, mỗi client đại diện cho một thực thể độc lập, có dữ liệu cục bộ riêng và có trách nhiệm huấn luyện mô hình trên dữ liệu đó mà không chia sẻ dữ liệu ra ngoài. Việc này mô phỏng các thiết bị IoT, tổ chức, hay đơn vị mạng có quyền quản lý dữ liệu riêng tư của mình.

Mỗi client được xây dựng dựa trên lớp NumPyClient do thư viện Flower cung cấp. Lớp này cho phép hiện thực logic huấn luyện nội bộ của từng client và định nghĩa các phương thức cần thiết để giao tiếp với server. Mỗi client bao gồm ba chức năng chính: lấy trọng số mô hình (get\_parameters), huấn luyện mô hình (fit) và đánh giá mô hình (evaluate).



Trong phương thức \_\_init\_\_, mỗi client được truyền vào mô hình (model), dữ liệu huấn luyện (train\_data) và nhãn tương ứng (train\_labels). Biến cid là chỉ số định danh của từng client, dùng để theo dõi và lưu log kết quả riêng biệt.

* Hàm get\_parameters() đơn giản chỉ trả về trọng số hiện tại của mô hình cục bộ.
* Hàm fit() thực hiện nhận trọng số mới từ server, cập nhật vào mô hình, sau đó huấn luyện trên dữ liệu cục bộ trong 5 epoch với batch size là 32. Sau khi huấn luyện xong, client trả lại trọng số mô hình mới cho server cùng với số lượng mẫu đã sử dụng. Đây là dữ liệu đầu vào quan trọng cho quá trình FedAvg ở server.
* Hàm evaluate() được gọi sau khi client nhận mô hình đã đồng bộ và cần đánh giá lại mô hình đó trên dữ liệu cục bộ. Điều này giúp kiểm tra mức độ phù hợp của mô hình toàn cục đối với dữ liệu riêng của từng client.

Việc khởi tạo từng client được thực hiện thông qua một hàm gọi là client\_fn, đây là yêu cầu của Flower để sinh ra client theo chỉ số ID

  
Trong đoạn mã này, mỗi client khởi tạo sẽ được gán dữ liệu riêng thông qua chỉ số partition\_id, tương ứng với tập con đã được chia từ trước trong quá trình phân chia dữ liệu. Dữ liệu ở đây là kết quả từ bước chia dữ liệu dựa trên các chiến lược như STRATIFIED, LEAVE\_ONE\_OUT,... do đó đảm bảo rằng mỗi client mô phỏng đúng môi trường phân tán.

Mỗi client đều sử dụng mô hình học máy giống với server để đảm bảo tính đồng bộ hóa trong quá trình truyền trọng số. Cấu trúc mô hình tại client là mạng neuron ba tầng, như đã mô tả ở phần trước, với hàm mất mát là sparse\_categorical\_crossentropy phù hợp cho dữ liệu nhãn nguyên dạng nhiều lớp.

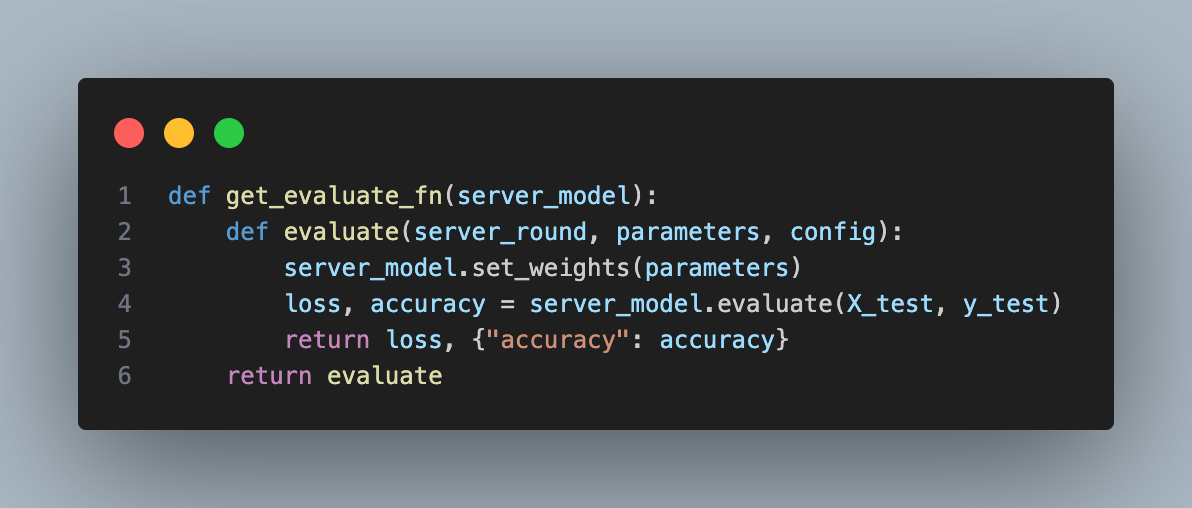
Quá trình khởi tạo client và huấn luyện cục bộ diễn ra trong từng vòng huấn luyện (round), mỗi client thực hiện học máy một cách độc lập và gửi kết quả về server. Điều này giúp tận dụng tốt dữ liệu phân tán và vẫn bảo toàn tính riêng tư của từng nguồn dữ liệu.

4. Đánh giá mô hình tại server

Sau mỗi vòng huấn luyện trong hệ thống Federated Learning, việc đánh giá mô hình toàn cục là bước then chốt để đo lường tiến trình hội tụ và hiệu quả tổng thể của quá trình huấn luyện. Đánh giá này được thực hiện tại server, sử dụng tập dữ liệu kiểm tra toàn cục được giữ riêng và không liên quan đến dữ liệu huấn luyện ở các client. Đây là cách kiểm tra khách quan nhằm xác định mô hình có đang học tốt từ dữ liệu phân tán hay không.

Trong hệ thống sử dụng Flower, việc đánh giá mô hình tại server được tích hợp thông qua tham số evaluate\_fn của chiến lược FedAvg. Đây là một hàm do người dùng định nghĩa, trả về kết quả đánh giá mô hình sau mỗi vòng cập nhật trọng số từ các client. Hàm này được gọi tự động bởi hệ thống Flower sau khi quá trình FedAvg hoàn tất.

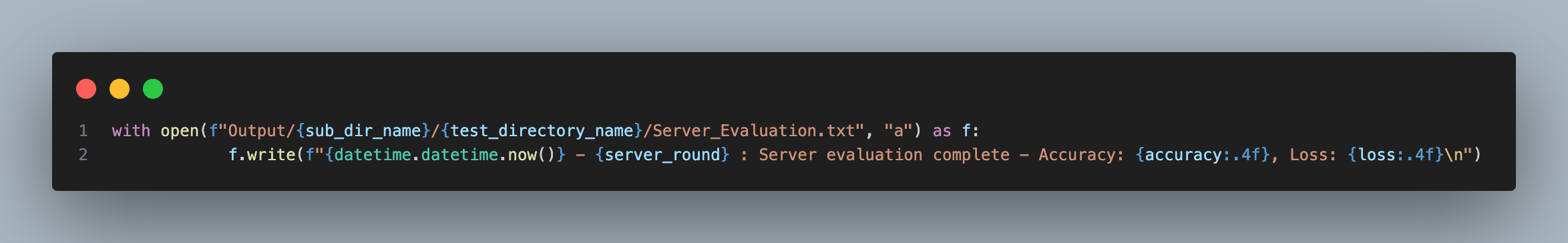
Hàm đánh giá tại server được viết dưới dạng một hàm bọc (closure), trong đó mô hình server được truyền vào và sử dụng để thiết lập lại trọng số mỗi khi có vòng huấn luyện mới



Đầu tiên mô hình server được cập nhật trọng số mới bằng hàm set\_weights. Đây là trọng số được tổng hợp từ các client sau mỗi vòng FedAvg. Sau đó, mô hình được đánh giá trên tập dữ liệu kiểm tra X\_test và y\_test, và kết quả trả về là hàm mất mát và độ chính xác.

Cấu trúc của hàm đánh giá đảm bảo rằng server không can thiệp vào dữ liệu huấn luyện của client, nhưng vẫn có thể theo dõi chính xác hiệu suất mô hình toàn cục theo thời gian. Việc cập nhật và đánh giá đều được tự động thực hiện trong vòng lặp của hệ thống Flower mà không cần gọi thủ công.

Bên cạnh việc đánh giá tức thời, hệ thống còn lưu lại kết quả đánh giá vào file văn bản để tiện theo dõi và phục vụ phân tích sau huấn luyện. Cụ thể, đoạn mã sau sẽ ghi kết quả đánh giá của từng vòng vào tệp Server\_Evaluation.txt trong thư mục Output



Ngoài việc lưu độ chính xác và loss, mô hình server còn có thể lưu lại cả kết quả dự đoán y\_pred dưới dạng .npy để sử dụng về sau cho việc tính thêm các chỉ số như confusion matrix, precision, recall hoặc trực quan hóa.

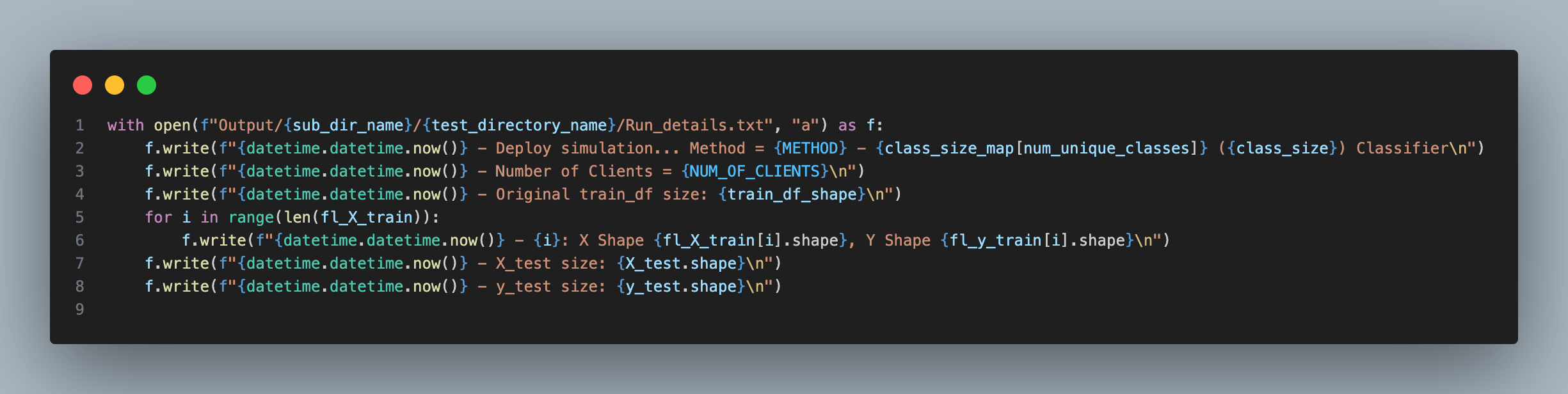
Nhờ cơ chế đánh giá này, hệ thống có thể giám sát hiệu suất của mô hình theo thời gian, đồng thời tạo cơ sở dữ liệu đánh giá để so sánh giữa các phương án huấn luyện khác nhau. Đây là một phần quan trọng trong việc đảm bảo mô hình FL không chỉ hội tụ mà còn có hiệu suất thực sự tốt

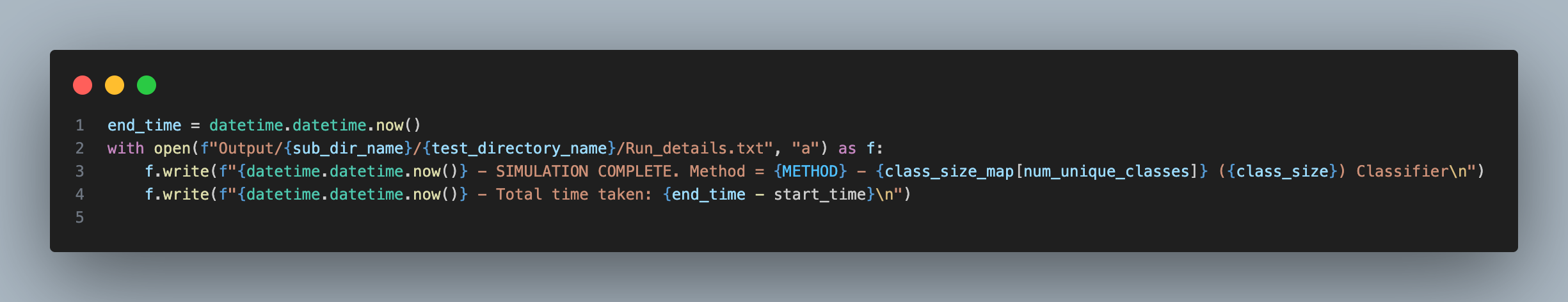
5. Khởi động mô phỏng và ghi kết quả huấn luyện

Sau khi hoàn tất khởi tạo mô hình toàn cục, chiến lược đồng bộ FedAvg, các client và hàm đánh giá tại server, hệ thống Federated Learning sẵn sàng bước vào giai đoạn chạy mô phỏng chính thức. Giai đoạn này mô phỏng toàn bộ quá trình huấn luyện phân tán giữa server và client qua nhiều vòng liên lạc. Mỗi vòng bao gồm việc gửi mô hình từ server đến các client, huấn luyện mô hình trên dữ liệu cục bộ, gửi kết quả huấn luyện trở lại server và cập nhật mô hình toàn cục.

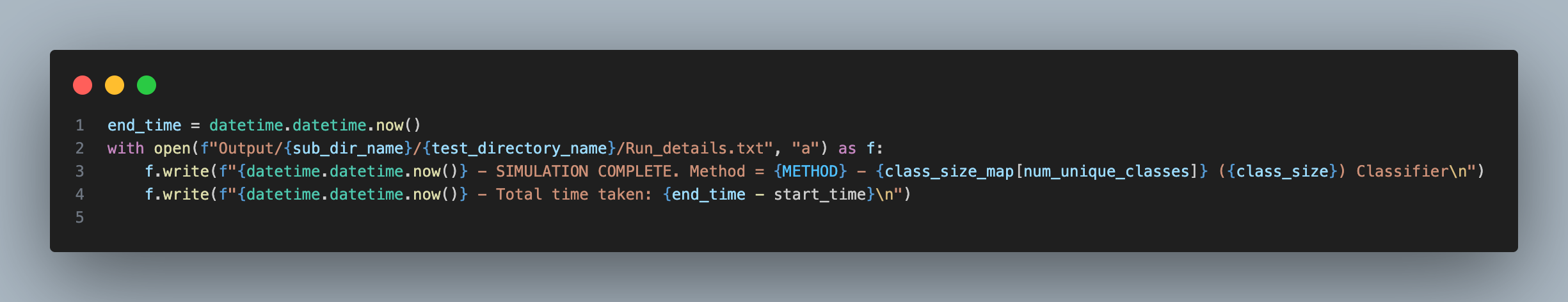
Hệ thống sử dụng hàm start\_simulation() do thư viện Flower cung cấp để khởi động quá trình mô phỏng. Hàm này yêu cầu cung cấp ba thành phần chính là hàm khởi tạo client (client\_fn), số lượng client mô phỏng (num\_clients) và cấu hình server bao gồm số vòng huấn luyện (num\_rounds) và chiến lược đồng bộ (strategy).

Trước khi khởi động, hệ thống sẽ ghi lại toàn bộ thông tin cấu hình vào tệp Run\_details.txt để tiện tra cứu sau này. Thông tin được ghi bao gồm phương pháp chia dữ liệu, loại mô hình sử dụng (binary, group, individual), số lượng client, kích thước dữ liệu huấn luyện ban đầu và chi tiết dữ liệu chia cho từng client. Ngoài ra, kích thước tập kiểm tra cũng được lưu để đảm bảo tính toàn vẹn khi đánh giá mô hình.

Sau khi lưu cấu hình, mô phỏng được chính thức bắt đầu bằng đoạn mã sau:

 Trong quá trình mô phỏng, Flower sẽ lần lượt gọi client\_fn để tạo các client, gửi mô hình từ server đến client, huấn luyện cục bộ tại từng client, thu nhận trọng số từ client và thực hiện trung bình có trọng số bằng thuật toán FedAvg tại server. Sau đó, hệ thống sẽ đánh giá mô hình bằng evaluate\_fn, ghi lại độ chính xác và mất mát, và chuyển sang vòng tiếp theo.

Sau khi mô phỏng kết thúc, hệ thống sẽ tính toán và in ra tổng thời gian huấn luyện. Thông tin này được lưu thêm vào tệp Run\_details.txt để tổng kết toàn bộ quá trình huấn luyện.

Việc lưu trữ toàn bộ log chi tiết của quá trình huấn luyện giúp dễ dàng phân tích hậu kỳ, so sánh giữa các chiến lược phân chia dữ liệu, các loại mô hình khác nhau hoặc theo dõi tiến trình hội tụ của mô hình. Điều này không chỉ hỗ trợ quá trình kiểm thử trong đồ án mà còn là thực hành tốt trong các hệ thống huấn luyện thực tế.

3.2.3. Kết quả huấn luyện và phân tích(CÁI SƯỜN)

**a) Mô tả tập dữ liệu kiểm tra và tiêu chí đánh giá**

Giới thiệu ngắn gọn về tập kiểm tra dùng để đánh giá mô hình toàn cục (số lượng mẫu, loại nhãn, có cân bằng hay không).  
Giải thích các chỉ số đánh giá sử dụng như:

* Accuracy
* Loss
* Precision, Recall, F1-score (nếu có)
* Confusion Matrix
* Biểu đồ ROC/AUC (nếu bạn có sử dụng)

**b) Kết quả huấn luyện theo từng vòng FL**

Trình bày bảng số liệu hoặc biểu đồ (line chart) về:

* Độ chính xác (accuracy) qua từng vòng huấn luyện
* Hàm mất mát (loss) theo thời gian
* So sánh các chiến lược chia dữ liệu nếu bạn thử nhiều chiến lược (STRATIFIED, LEAVE\_ONE\_OUT, v.v.)

**c) So sánh mô hình FL với mô hình tập trung (nếu có)**

Nếu bạn có xây dựng mô hình huấn luyện tập trung trên toàn bộ dữ liệu, hãy so sánh hiệu suất với FL  
Nêu ra ưu điểm, hạn chế của mỗi cách tiếp cận

**d) Phân tích ưu nhược điểm của FL trong bài toán phát hiện tấn công**

* Tính bảo mật và quyền riêng tư dữ liệu
* Khả năng tổng quát hóa trên dữ liệu phân tán
* Thách thức khi dữ liệu phân bố không đồng đều
* Chi phí tính toán và băng thông truyền thông

**e) Kết luận tạm thời từ kết quả huấn luyện**

Tóm tắt kết quả mô hình FL đã đạt được  
Đưa ra nhận định về khả năng áp dụng thực tế trong hệ thống phát hiện tấn công mạng IoT

# CHƯƠNG 4: THỰC NGHIỆM VÀ ĐÁNH GIÁ

## 4.1. Mô hình thực nghiệm và tập dữ liệu kiểm tra

4.1.1. Mô hình Federated Learning triển khai  
4.1.2. Mô tả tập dữ liệu kiểm tra  
4.1.3. Các tiêu chí đánh giá hiệu suất mô hình  
    • Accuracy  
    • Loss  
    • Precision, Recall, F1-score  
    • Confusion Matrix  
    • ROC/AUC (nếu có)

## 4.2. Kết quả huấn luyện và phân tích

4.2.1. Kết quả huấn luyện theo từng vòng liên kết FL  
    • Biểu đồ accuracy qua từng vòng  
    • Biểu đồ loss theo thời gian  
    • So sánh chiến lược chia dữ liệu: STRATIFIED, LEAVE\_ONE\_OUT,... (nếu có)

4.2.2. So sánh mô hình Federated Learning với mô hình tập trung (nếu có)  
    • So sánh hiệu suất (accuracy, loss, v.v.)  
    • Phân tích ưu nhược điểm của từng mô hình

## 4.3. Phân tích kết quả và đánh giá mô hình

4.3.1. Ưu và nhược điểm của Federated Learning trong bài toán phát hiện tấn công  
    • Bảo mật & quyền riêng tư  
    • Khả năng tổng quát hóa  
    • Thách thức dữ liệu phân tán không đều  
    • Chi phí tính toán & truyền thông

4.3.2. Kết luận tạm thời từ kết quả huấn luyện  
    • Tóm tắt hiệu suất đạt được  
    • Nhận định khả năng áp dụng thực tế trong hệ thống IoT

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

## Hạn chế của nghiên cứu

## Hướng phát triển trong tương lai

# TÀI LIỆU THAM KHẢO

1. Hoàng, “Giải pháp Federated Learning nâng cao bảo mật huấn luyện AI,” *Comlink - Đối Tác Chuyên Nghiệp*, Dec. 21, 2023. [Online]. Available: <https://comlink.vn/federated-learning/>
2. J. Mateus, L. Zodi, and A. Bagula, “Federated learning-based solution for DDoS detection in SDN,” in Proc. Int. Conf. on Computing, Networking and Communications (ICNC), Honolulu, HI, USA, 2024, pp. 875–880. doi: 10.1109/ICNC59896.2024.10556115.
3. G. Shirvani, S. Ghasemshirazi, M. A. Alipour, Carleton University, and Institute of Science and High Technology and Environmental Science, “Enhancing IoT Security Against DDoS Attacks through Federated Learning,” journal-article.
4. N. U. Ain, M. Sardaraz, M. Tahir, M. W. A. Elsoud, and A. Alourani, “Securing IoT networks against DDoS Attacks: A hybrid deep learning approach,” *Sensors*, vol. 25, no. 5, p. 1346, Feb. 2025, doi: 10.3390/s25051346.
5. Denial of Service, ScienceDirect, Elsevier. [Online]. Available: <https://www.sciencedirect.com/topics/computer-science/denial-of-service>. [Accessed: Jun. 25, 2025].
6. B. K. Jena, “What is Phishing Attack in Cyber Security - Complete Guide,” *Simplilearn.com*, Jun. 09, 2025. https://www.simplilearn.com/tutorials/cryptography-tutorial/what-is-phishing-attack
7. Mallik, “Man-in-the-middle-attack: Understanding in simple words,” Cyberspace: Jurnal Pendidikan Teknologi Informasi, vol. 2, p. 109, 2019. doi: 10.22373/cj.v2i2.3453.
8. M. Rahmati, “Federated Learning-Driven Cybersecurity Framework for IoT Networks with Privacy-Preserving and Real-Time Threat Detection Capabilities,” *arXiv.org*, Feb. 14, 2025. https://arxiv.org/abs/2502.10599?utm\_source=
9. N. N. Albogami, “Intelligent deep federated learning model for enhancing security in internet of things enabled edge computing environment,” *Scientific Reports*, vol. 15, no. 1, Feb. 2025, doi: 10.1038/s41598-025-88163-5.
10. M. A. Elaziz, I. A. Fares, A. Dahou, and M. Shrahili, “Federated learning framework for IoT intrusion detection using tab transformer and nature-inspired hyperparameter optimization,” *Frontiers in Big Data*, vol. 8, May 2025, doi: 10.3389/fdata.2025.1526480.
11. B. Akwaronwu, I. Akwaronwu, and O. Adeniyi, “Brute force attack detection in network traffic using convolutional neural networks,” Asian Journal of Research in Computer Science, vol. 18, pp. 387–402, 2025. doi: 10.9734/ajrcos/2025/v18i5662.