**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO CUỐI KỲ**

**MÔN: AN TOÀN HỆ THỐNG THÔNG TIN**

**Đề tài: CÁC THUẬT TOÁN TỔNG HỢP TRONG FEDERATED LEARNING**

Nhóm sinh viên thực hiện:

1. Lê Thị Ngọc Lan MSHV: 240104010

2. Vũ Mai Quỳnh Loan MSHV: 240104012

3. Lê Bá Trực MSHV: 240202016

GVHD: TS. Nguyễn Tấn Cầm

Tp. Hồ Chí Minh, 12/2024

**THÔNG TIN CHUNG**

|  |  |  |  |
| --- | --- | --- | --- |
| Họ và tên | Lê Thị Ngọc Lan | Vũ Mai Quỳnh Loan | Lê Bá Trực |
| MSHV | 240104010 | 240104012 | 240202016 |
| Email | lanltn.19@grad.uit.edu.vn | loanvmq.19@grad.uit.edu.vn | [Truclb.19@grad.uit.edu.vn](mailto:Truclb.19@grad.uit.edu.vn) |
| Ngành học | Hệ Thống Thông Tin | Hệ Thống Thông Tin | An toàn thông tin |
| Tự đánh giá điểm |  |  |  |
| Hình cá nhân |  |  |  |

Mục Lục

Danh sách các bản báo cáo

Danh sách các hình ảnh

**Chương I: Giới thiệu về Federated Learning**

**Federated Learning (FL) là gì?**

Federated Learning (FL), hay còn gọi là học liên kết, là một phương pháp học máy phân tán, trong đó các mô hình được huấn luyện trên nhiều thiết bị hoặc máy chủ mà không cần tập trung dữ liệu tại một nơi. Thay vì chia sẻ dữ liệu thô, các thiết bị gửi các bản cập nhật mô hình (model updates) sau quá trình huấn luyện cục bộ, giúp bảo vệ tính riêng tư của dữ liệu.

FL được giới thiệu bởi Google vào năm 2016, và từ đó, nó đã trở thành một trong những giải pháp tiềm năng để giải quyết các vấn đề về quyền riêng tư và bảo mật trong học máy.

**Cơ chế hoạt động của Federated Learning**

Cơ chế hoạt động của FL có thể được chia thành 4 giai đoạn chính:

**1. Khởi tạo mô hình toàn cục (Global Model Initialization)**

* Mô hình toàn cục ban đầu được tạo trên một máy chủ trung tâm. Đây có thể là một mô hình đơn giản hoặc một mô hình phức tạp được thiết kế để huấn luyện phân tán.
* Mô hình này sau đó được gửi đến tất cả các thiết bị hoặc client tham gia.

**2. Huấn luyện cục bộ tại client**

* Mỗi client nhận mô hình toàn cục và sử dụng dữ liệu cục bộ của mình để huấn luyện mô hình đó. Quá trình huấn luyện này thường diễn ra theo một hoặc nhiều epoch.
* Vì dữ liệu không rời khỏi client, quyền riêng tư của dữ liệu được bảo vệ tốt hơn.

**3. Cập nhật mô hình cục bộ (Model Update)**

* Sau khi huấn luyện xong, các client sẽ gửi các bản cập nhật mô hình (trọng số hoặc gradient) về máy chủ trung tâm.
* Chỉ có thông tin mô hình (model updates) được gửi, không có dữ liệu thô nào được chia sẻ.

**4. Tổng hợp mô hình toàn cục (Global Model Aggregation)**

* Máy chủ trung tâm nhận các bản cập nhật từ client và thực hiện quá trình tổng hợp, thường sử dụng thuật toán **FedAvg** (Federated Averaging) hoặc các chiến lược khác.
* Sau đó, mô hình toàn cục được cập nhật và phân phối lại cho các client trong vòng tiếp theo.

Quá trình này được lặp lại qua nhiều vòng (federated rounds) cho đến khi mô hình đạt được hiệu suất mong muốn.

**Ưu điểm của Federated Learning**

**1. Bảo vệ quyền riêng tư**

* Dữ liệu không bao giờ rời khỏi client, giúp giảm nguy cơ rò rỉ dữ liệu.
* Thích hợp cho các lĩnh vực nhạy cảm như chăm sóc sức khỏe, tài chính, và dữ liệu cá nhân.

**2. Giảm chi phí truyền dữ liệu**

* Thay vì truyền tải toàn bộ dữ liệu đến một trung tâm, chỉ các bản cập nhật mô hình (trọng số hoặc gradient) được gửi đi, tiết kiệm băng thông mạng.

**3. Tận dụng dữ liệu phi tập trung**

* FL tận dụng dữ liệu cục bộ trên các thiết bị hoặc client, làm tăng tính đa dạng và độ bao phủ của dữ liệu.
* Có thể huấn luyện mô hình trên các thiết bị như điện thoại di động, IoT, hoặc hệ thống doanh nghiệp.

**4. Phù hợp với các quy định pháp lý**

* FL giúp tuân thủ các quy định nghiêm ngặt về bảo vệ dữ liệu, như GDPR ở châu Âu hoặc HIPAA trong ngành y tế.

**Nhược điểm của Federated Learning**

**1. Sự không đồng nhất trong dữ liệu (Non-IID)**

* Dữ liệu trên các client có thể không đồng nhất (non-IID), gây khó khăn cho việc huấn luyện và làm giảm hiệu suất mô hình toàn cục.

**2. Tài nguyên không đồng đều**

* Các client có thể khác nhau về tài nguyên tính toán, bộ nhớ và băng thông, dẫn đến sự không đồng nhất trong quá trình huấn luyện.

**3. Độ trễ cao**

* FL phụ thuộc vào kết nối mạng giữa máy chủ và client. Các client có băng thông yếu hoặc mất kết nối có thể làm chậm quá trình huấn luyện.

**4. Tấn công và bảo mật**

* Client hoặc máy chủ trung tâm có thể trở thành mục tiêu của các cuộc tấn công như **poisoning attack** (đầu độc dữ liệu), **model inversion attack** (tái tạo dữ liệu từ mô hình), và **free-rider attack** (client không đóng góp nhưng hưởng lợi từ mô hình).

**5. Chi phí tổng hợp mô hình**

* Việc tổng hợp mô hình (aggregation) đòi hỏi sức mạnh tính toán cao ở máy chủ trung tâm, đặc biệt với các mô hình phức tạp và số lượng lớn client.

**Ứng dụng của Federated Learning**

**1. Chăm sóc sức khỏe**

* Các bệnh viện sử dụng FL để huấn luyện mô hình dự đoán bệnh dựa trên dữ liệu của từng bệnh viện mà không cần chia sẻ dữ liệu bệnh nhân.

**2. Cá nhân hóa trên thiết bị**

* Google sử dụng FL để cải thiện tính năng gõ dự đoán trên bàn phím (Gboard) mà không cần tải dữ liệu người dùng lên máy chủ.

**3. Phát hiện gian lận**

* FL có thể được dùng để phát hiện gian lận trong các hệ thống tài chính mà không cần chia sẻ dữ liệu nhạy cảm giữa các tổ chức.

**4. IoT và Smart City**

* FL hỗ trợ các thiết bị IoT huấn luyện mô hình nhận diện hành vi, tối ưu hóa năng lượng, hoặc dự đoán giao thông.

**Chương II**

1. Triển khai Federated trong An toàn hệ thống thông tin  
   1. Vòng đời Mô hình trong Federated Learning cho Hệ thống Phát hiện Xâm nhập (IDS)

Federated Learning (FL) là một hướng tiếp cận tiên tiến trong việc huấn luyện mô hình học máy, đặc biệt là trong các hệ thống có dữ liệu phân tán và yêu cầu cao về quyền riêng tư. Trong lĩnh vực phát hiện xâm nhập (Intrusion Detection Systems - IDS), FL mang lại giải pháp hiệu quả, vừa đảm bảo sự an toàn dữ liệu, vừa nâng cao hiệu suất của mô hình phát hiện. Dưới đây là phân tích chi tiết về các giai đoạn trong vòng đời của một mô hình FL được áp dụng cho IDS, từ xác định vấn đề, thu thập dữ liệu, đến triển khai và bảo trì.

1. Từ xác định vấn đề đến thu thập dữ liệu

Quá trình phát triển một mô hình FL bắt đầu từ việc xác định rõ ràng mục tiêu cụ thể mà hệ thống IDS cần đạt được. Điều này bao gồm việc trả lời các câu hỏi: Hệ thống này cần phát hiện những dạng tấn công nào? Độ chính xác mục tiêu là bao nhiêu? Có yêu cầu gì về thời gian phản hồi hoặc tỷ lệ cảnh báo sai không?

Một khi mục tiêu đã rõ ràng, các thiết bị khách trong hệ thống, chẳng hạn như tường lửa, router, hoặc thiết bị đầu cuối, được thiết lập để thu thập dữ liệu cục bộ. Dữ liệu này bao gồm nhật ký lưu lượng mạng, sự kiện bảo mật, hoặc hành vi hệ thống. Do dữ liệu nhạy cảm không thể chuyển ra khỏi thiết bị, FL đóng vai trò quan trọng trong việc đảm bảo rằng dữ liệu này vẫn nằm trong giới hạn bảo mật trong khi được sử dụng để huấn luyện mô hình.

1. Huấn luyện mô hình phân tán

Giai đoạn huấn luyện là trọng tâm của FL. Tại đây, các thiết bị khách thực hiện huấn luyện cục bộ trên dữ liệu riêng của chúng. Điều này đảm bảo rằng dữ liệu thô không bị lộ ra ngoài mà vẫn đóng góp vào việc cải thiện mô hình toàn cục.

Quá trình huấn luyện diễn ra như sau:

Khởi tạo: Một mô hình khởi đầu được gửi từ máy chủ trung tâm đến các thiết bị khách.

Huấn luyện cục bộ: Mỗi thiết bị thực hiện tối ưu hóa mô hình dựa trên dữ liệu của mình. Điều này có thể bao gồm việc tính toán gradient hoặc cập nhật trọng số.

Tổng hợp mô hình: Các thiết bị gửi bản cập nhật mô hình (model updates) – thường là gradient hoặc trọng số – về máy chủ trung tâm. Máy chủ này sẽ tổng hợp các bản cập nhật để tạo ra một mô hình toàn cục mới. Quá trình tổng hợp phổ biến nhất là Federated Averaging (FedAvg), trong đó các bản cập nhật được kết hợp dựa trên trọng số của số lượng dữ liệu từng thiết bị.

Lợi thế lớn nhất của phương pháp này là nó tận dụng được sự đa dạng của dữ liệu từ các nguồn khác nhau, giúp mô hình học hỏi từ nhiều dạng lưu lượng mạng và hành vi tấn công khác nhau.

1. Đánh giá và lựa chọn mô hình

Sau khi huấn luyện, bước tiếp theo là đánh giá hiệu suất mô hình. Có hai phương pháp chính để đánh giá:

Kiểm tra với dữ liệu chuẩn: Mô hình được đánh giá trên một tập dữ liệu chuẩn trong trung tâm dữ liệu, thường được sử dụng để so sánh các mô hình khác nhau.

Kiểm tra trên các thiết bị thực tế: Mô hình được triển khai thử nghiệm trên các thiết bị cục bộ chưa tham gia vào quá trình huấn luyện để đánh giá khả năng phát hiện các hành vi bất thường hoặc xâm nhập thực tế.

Các chỉ số đánh giá quan trọng bao gồm:

Độ chính xác (Accuracy): Đo lường khả năng phát hiện đúng các hành vi xâm nhập.

Tỷ lệ cảnh báo sai (False Positive Rate - FPR): Xác định tần suất mô hình báo động sai.

Hiệu quả tính toán: Kiểm tra xem mô hình có đáp ứng được yêu cầu thời gian thực trong môi trường sản xuất không.

Khi quá trình đánh giá hoàn tất, mô hình tốt nhất sẽ được chọn dựa trên hiệu suất tổng thể và khả năng đáp ứng các yêu cầu cụ thể của hệ thống IDS.

1. Triển khai mô hình

Giai đoạn triển khai bao gồm tích hợp mô hình vào môi trường mạng thực tế. Quá trình này không chỉ đơn thuần là đưa mô hình vào hoạt động, mà còn yêu cầu:

Kiểm tra chất lượng: Đảm bảo rằng mô hình hoạt động ổn định và không gây gián đoạn cho hệ thống mạng.

Thử nghiệm A/B: So sánh hiệu suất của mô hình mới với các mô hình hoặc phương pháp hiện có.

Triển khai dần (Staged Rollout): Đưa mô hình vào hoạt động theo từng giai đoạn, từ nhỏ đến lớn, để giảm thiểu rủi ro.

Khi được triển khai, mô hình sẽ liên tục giám sát lưu lượng mạng, phân tích nhật ký và phát hiện các mối đe dọa trong thời gian thực. Hệ thống cũng cần có cơ chế phản hồi, cho phép tự động hóa một phần hoặc toàn bộ các hành động ứng phó với xâm nhập.

1. Bảo trì và cập nhật mô hình

Môi trường an ninh mạng luôn thay đổi với các mối đe dọa mới xuất hiện liên tục. Do đó, một mô hình FL cho IDS cần được bảo trì thường xuyên để đảm bảo hiệu suất tối ưu.

Các hoạt động bảo trì bao gồm:

Giám sát hiệu suất: Theo dõi độ chính xác và khả năng phát hiện của mô hình qua thời gian.

Phát hiện vấn đề: Xác định các kịch bản mà mô hình hoạt động kém, chẳng hạn như tỷ lệ cảnh báo sai tăng cao hoặc bỏ sót các dạng tấn công mới.

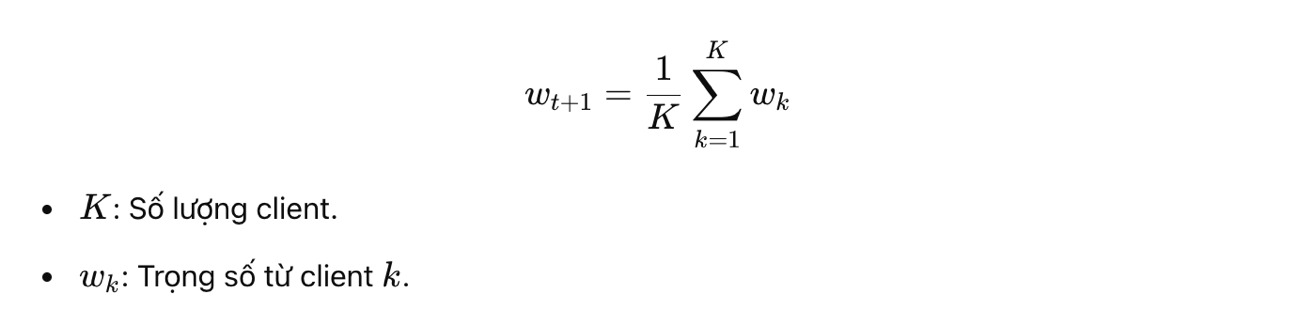
Huấn luyện lại định kỳ: Mô hình được huấn luyện lại với dữ liệu mới từ các thiết bị khách. Điều này đảm bảo rằng mô hình học hỏi được từ các xu hướng tấn công mới và duy trì hiệu quả trong phát hiện xâm nhập.

Quá trình bảo trì không chỉ cải thiện hiệu suất của mô hình mà còn tăng cường độ tin cậy của hệ thống IDS trong việc bảo vệ mạng.

**Chương 3:** So sánh các thuật toán tổng hợp: Federated Averaging, Weghted Federated Averaging, Median-based Federated Averaging, Trimmed Mean, Krum (K-Center) 🡪 bỏ

**Federated Averaging (FedAvg)**

* Mô tả:
  + Trung bình các mô hình từ các client mà không sử dụng trọng số.
  + Mỗi client có ảnh hưởng ngang nhau trong việc tổng hợp mô hình toàn cục.
* Công thức:



* Ưu điểm:
  + Đơn giản, dễ triển khai.
  + Tính toán nhanh, không yêu cầu thêm thông tin như kích thước dữ liệu.
* Nhược điểm:
  + Không hiệu quả nếu dữ liệu không cân bằng giữa các client (client có ít dữ liệu cũng đóng góp ngang bằng client có nhiều dữ liệu).

**Weighted Federated Averaging**

* Mô tả:
  + Mở rộng từ FedAvg, sử dụng trọng số dựa trên kích thước dữ liệu của từng client.
  + Các client có nhiều dữ liệu hơn sẽ đóng góp nhiều hơn vào mô hình toàn cục.
* Công thức:

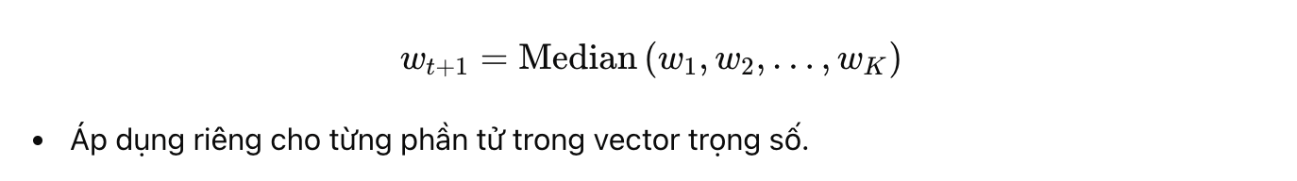
A black and white math equations

Description automatically generated with medium confidence

* Ưu điểm:
  + Giải quyết vấn đề dữ liệu không cân bằng giữa các client.
  + Cải thiện hiệu năng tổng hợp mô hình toàn cục.
* Nhược điểm:
  + Yêu cầu thông tin về kích thước dữ liệu của từng client, có thể làm giảm tính bảo mật.

**Median-based Federated Averaging**

* Mô tả:
  + Thay vì trung bình, sử dụng trung vị (median) để tổng hợp mô hình.
  + Loại bỏ ảnh hưởng của các giá trị ngoại lai (outliers).
* Công thức:



* Ưu điểm:
  + Chống lại các mô hình bất thường hoặc độc hại từ các client.
  + Đơn giản và dễ triển khai.
* Nhược điểm:
  + Hiệu năng kém trong trường hợp tất cả các client đều hợp tác và không có ngoại lệ.

**Trimmed Mean**

* Mô tả:
  + Loại bỏ một số lượng nhỏ các giá trị lớn nhất và nhỏ nhất từ các trọng số trước khi tính trung bình.
  + Bảo vệ mô hình toàn cục khỏi các outliers.
* Công thức:

A white background with black text

Description automatically generated

* Ưu điểm:
  + Tốt hơn trung vị (median) trong việc giảm thiểu ảnh hưởng của outliers, đặc biệt khi dữ liệu gần Gaussian.
* Nhược điểm:
  + Yêu cầu tham số betaβ, đòi hỏi điều chỉnh tùy thuộc vào dữ liệu.

**Krum (K-Center)**

* Mô tả:
  + Chọn một trọng số từ các client làm trọng số toàn cục sao cho nó gần với đa số các trọng số khác nhất.
  + Được thiết kế để chống lại các tấn công Byzantine (các client gửi thông tin sai lệch).
* Công thức:

A white background with black text

Description automatically generated

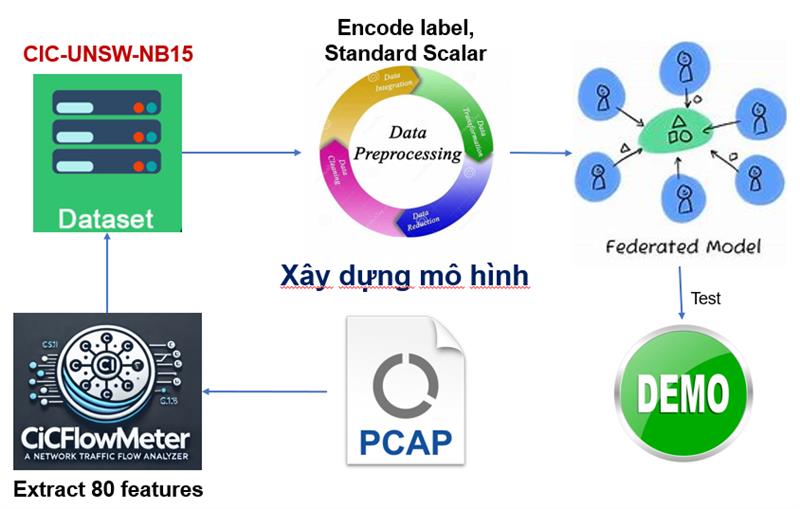
* Ưu điểm:
  + Hiệu quả trong việc bảo vệ mô hình khỏi các client độc hại.
  + Được thiết kế cho môi trường không tin cậy.
* Nhược điểm:
  + Tốn kém về tính toán do cần đo khoảng cách giữa các trọng số.
  + Không hiệu quả nếu tất cả client hợp tác.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Thuật toán | Độ đơn giản | Chống ngoại lai | Chống client độc hại | Hiệu quả với dữ liệu không cân bằng | Tính toán phức tạp |
| Federated Averaging | Cao | Thấp | Thấp | Thấp | Thấp |
| Weighted Federated Averaging | Trung Bình | Thấp | Thấp | Cao | Trung Bình |
| Median-based FedAvg | Trung Bình | Cao | Trung Bình | Trung Bình | Trung Bình |
| Trimmed mean | Trung Bình | Cao | Trung Bình | Trung Bình | Trung Bình |
| Krum | Thấp | Cao | Cao | Thấp | Cao |

*Table 01: Bảng so sánh các thuật toán tổng hợp trọng số*

**Chương II: Thiết kế và triển khai mô hình**

Quy trình xây dựng mô hình phát hiện xâm nhập mạng dựa trên học liên kết (Federated Learning) với tập dữ liệu CIC-UNSW-NB15. Dữ liệu thô từ PCAP được phân tích bằng CICFlowMeter để trích xuất 80 đặc trưng mạng. Sau đó, dữ liệu được tiền xử lý bằng cách mã hóa nhãn và chuẩn hóa, giúp chuẩn bị cho việc đào tạo mô hình. Mô hình học liên kết được xây dựng để đào tạo trên các nút phân tán, đảm bảo tính bảo mật và riêng tư. Cuối cùng, mô hình được kiểm thử với dữ liệu mới và tạo một bản demo nhằm đánh giá hiệu quả phát hiện các hành vi xâm nhập.



*Image 0x: Quy trình xây dựng mô hình*

**1. Xử lí dữ liệu**

a. Giới thiệu bộ dataset

Dữ liệu CIC-UNSW-NB15 là phiên bản mở rộng của bộ UNSW-NB15, được phát triển bởi sự hợp tác giữa Viện An ninh mạng Canada (Canadian Institute for Cybersecurity - CIC) và Đại học New South Wales (UNSW). Bộ dữ liệu này được thiết kế nhằm hỗ trợ phát triển và đánh giá các hệ thống phát hiện xâm nhập mạng (NIDS). Nó bao gồm các gói dữ liệu mạng với hoạt động bình thường và nhiều loại tấn công khác nhau, phù hợp với nhu cầu phân tích và xây dựng các mô hình học máy [7], [2].

Bộ dữ liệu UNSW-NB15 ban đầu được tạo ra bằng cách sử dụng công cụ IXIA PerfectStorm để tái hiện các hoạt động mạng hiện đại và các kịch bản tấn công. Dữ liệu thô (khoảng 100 GB) được thu thập trong hai ngày bằng công cụ **tcpdump**. Sau đó, các đặc trưng mạng được trích xuất bằng Argus và Bro-IDS, tạo ra tổng cộng 47 đặc trưng, được phân thành các nhóm như Basic, Content, Time và các đặc trưng bổ sung. Bộ dữ liệu này chứa 9 loại tấn công chính bao gồm: Fuzzers, Analysis, Backdoors, Denial of Service (DoS), Exploits, Generic, Reconnaissance, Shellcode, và Worms [7], [8].

Bộ dữ liệu CIC-UNSW-NB15 mở rộng dựa trên việc sử dụng công cụ CICFlowMeter để trích xuất thêm 80 đặc trưng, giúp tăng cường khả năng phân tích và phát hiện xâm nhập. Dữ liệu bao gồm cả cấp độ gói tin (packet-level) và luồng dữ liệu (flow-level), mang lại tính linh hoạt cao trong nghiên cứu [2].

Bộ CIC-UNSW-NB15 đã được công khai và được sử dụng rộng rãi bởi các nhà nghiên cứu và chuyên gia trong lĩnh vực an ninh mạng để phát triển và đánh giá các mô hình học máy. Tính toàn diện của bộ dữ liệu này và sự đa dạng của các loại tấn công đã giúp nó trở thành tài nguyên quý giá cho các nghiên cứu liên quan đến hệ thống phát hiện xâm nhập [7], [2], [8].

b. Quy trình xử lí dữ liệu

**2. Các thuật toán sử dụng**

a. Federated Averaging (FedAvg)

b. Weighted Federated Averaging

c. Median-based Federated Averaging (Median FedAvg)

* + 1. **Mô hình hệ thống Federated Learning**

Mô hình hệ thống của Federated Learning bao gồm hai thành phần chính: **server trung tâm** và **các client phân tán**. Hệ thống này được thiết kế để đảm bảo tính bảo mật, quyền riêng tư và khả năng tận dụng dữ liệu phân tán trên nhiều thiết bị [9].

**A diagram of a server

Description automatically generated**

Figure 1: Mô hình hệ thống Federated Learning

**Thành phần chính của hệ thống**

* **Server trung tâm**  
  Server trung tâm đóng vai trò điều phối toàn bộ quá trình học liên kết. Các chức năng chính của server bao gồm:
* Khởi tạo mô hình ban đầu: Server giữ một bản sao của mô hình toàn cục (global model) và khởi tạo mô hình này trước khi phân phối cho các client.
* Thu thập bản cập nhật từ client: Sau mỗi vòng lặp đào tạo, server thu thập các bản cập nhật mô hình từ các client. Các bản cập nhật này có thể là gradients (gradient descent) hoặc trọng số mô hình cục bộ [10].
* Tổng hợp các bản cập nhật: Server thực hiện việc tổng hợp các bản cập nhật từ các client bằng thuật toán như Federated Averaging (FedAvg), trong đó mô hình toàn cục được tính toán dựa trên trung bình có trọng số của các cập nhật từ các client [10].
* Phân phối mô hình mới: Sau khi tổng hợp, server phân phối mô hình toàn cục đã được cập nhật tới các client để tiếp tục vòng lặp đào tạo.
* **Các client phân tán**  
  Client là các thiết bị hoặc các nút lưu trữ dữ liệu cục bộ và tham gia vào quá trình đào tạo mô hình. Ví dụ về client bao gồm điện thoại thông minh, cảm biến IoT, máy tính cá nhân, hoặc các thiết bị khác trong mạng phân tán. Mỗi client có vai trò cụ thể:
* Lưu trữ dữ liệu cục bộ: Dữ liệu được lưu trữ tại client và không được chia sẻ với server hoặc các client khác, bảo đảm tính riêng tư của người dùng [9].
* Đào tạo mô hình cục bộ: Client sử dụng mô hình toàn cục từ server và thực hiện đào tạo cục bộ trên dữ liệu của mình. Việc đào tạo này thường sử dụng các thuật toán tối ưu như Stochastic Gradient Descent (SGD).
* Gửi bản cập nhật mô hình: Sau khi hoàn thành quá trình đào tạo cục bộ, client tính toán và gửi bản cập nhật mô hình (hoặc gradients) về server trung tâm [10].

**Quy trình hoạt động của hệ thống Federated Learning**

Hệ thống Federated Learning vận hành theo các bước sau:

**Bước 1: Khởi tạo mô hình toàn cục:**  
Server trung tâm khởi tạo một mô hình toàn cục ban đầu và phân phối nó đến tất cả các client tham gia [9].

**Bước 2: Đào tạo cục bộ tại các client:**  
Mỗi client nhận mô hình toàn cục từ server và sử dụng dữ liệu cục bộ của mình để cập nhật mô hình thông qua quá trình tối ưu hóa. Dữ liệu này được lưu trữ nội bộ và không bao giờ được gửi ra ngoài. Kết quả của quá trình đào tạo là các bản cập nhật mô hình hoặc gradients được tính toán từ dữ liệu riêng của client.

**Bước 3: Gửi bản cập nhật về server:**  
Các client gửi bản cập nhật mô hình (trọng số mới hoặc gradients) về server trung tâm. Quá trình này chỉ gửi các thông tin cần thiết để cập nhật mô hình, không gửi dữ liệu thô, từ đó đảm bảo tính bảo mật và quyền riêng tư [10].

**Bước 4: Tổng hợp mô hình tại server:**  
Server nhận tất cả các bản cập nhật từ các client và thực hiện tổng hợp để xây dựng mô hình toàn cục mới. Quá trình tổng hợp có thể được thực hiện bằng nhiều thuật toán khác nhau, tùy thuộc vào bài toán cụ thể, nhưng phổ biến nhất là **FedAvg**. Trong FedAvg, server tính trung bình trọng số của các bản cập nhật dựa trên kích thước dữ liệu cục bộ của từng client [10].

**Bước 5: Phân phối lại mô hình toàn cục:**  
Mô hình toàn cục mới được server phân phối lại cho tất cả các client, và quá trình lặp lại từ bước 2 đến bước 5. Chu kỳ này tiếp diễn cho đến khi mô hình đạt được độ hội tụ hoặc đạt tiêu chuẩn hiệu suất mong muốn.

* + 1. **Mô hình xử lý dữ liệu**

A diagram of a train

Description automatically generated

Figure 2: Mô hình xử lý dữ liệu

Bước 1: Thu thập và trích xuất đặc trưng

* Thu thập dữ liệu:

Các tập dữ liệu CIC-IDS2017 và UNSW-NB15 được chọn vì chúng cung cấp một lượng lớn dữ liệu về các cuộc tấn công mạng đa dạng, từ đó giúp mô hình học được các đặc trưng đặc trưng của từng loại tấn công.

Dữ liệu thô thường bao gồm các thông tin về gói tin mạng, hoạt động của hệ thống, và các thông số khác.

* Trích xuất đặc trưng:

Đặc trưng thủ công: Các chuyên gia an toàn thông tin sẽ lựa chọn và trích xuất các đặc trưng có ý nghĩa từ dữ liệu thô, ví dụ như số lượng gói tin, kích thước gói tin, các cờ TCP, địa chỉ IP nguồn và đích,...

Đặc trưng tự động: Các kỹ thuật học máy như PCA (Principal Component Analysis), t-SNE (t-Distributed Stochastic Neighbor Embedding) có thể được sử dụng để tìm ra các đặc trưng ẩn trong dữ liệu.

Bước 2: Tiền xử lý dữ liệu

* Xử lý dữ liệu thiếu: Điền vào các giá trị thiếu bằng các phương pháp như trung bình, trung vị, hoặc dự đoán dựa trên các mẫu khác.
* Mã hóa: Chuyển đổi các dữ liệu dạng danh mục (categorical) thành dạng số (numerical) để mô hình có thể xử lý, ví dụ như sử dụng one-hot encoding.
* Chuẩn hóa: Điều chỉnh dữ liệu về cùng một thang đo để các đặc trưng có đóng góp tương đương trong quá trình học. Standard Scaler là một phương pháp phổ biến để chuẩn hóa dữ liệu.

Bước 3: Học tập phân tán

* Phân chia dữ liệu: Dữ liệu được chia thành các phần nhỏ hơn có kích thước tương đương nhau. Việc phân chia này giúp tăng tốc độ huấn luyện và giảm tải cho từng máy tính.
* Huấn luyện mô hình cục bộ: Mỗi phần dữ liệu được gán cho một mô hình cục bộ để huấn luyện. Các mô hình cục bộ này có kiến trúc giống nhau nhưng các trọng số khác nhau.
* Truyền thông: Các mô hình cục bộ sẽ trao đổi thông tin với nhau trong quá trình huấn luyện để cập nhật các trọng số.

Bước 4: Học tập tập trung

* Tổng hợp mô hình: Sau khi các mô hình cục bộ đã được huấn luyện, các trọng số của chúng được tổng hợp lại để tạo thành một mô hình toàn cục. Phương pháp tổng hợp phổ biến là tính trung bình cộng hoặc weighted averaging.
* Tinh chỉnh mô hình toàn cục: Mô hình toàn cục có thể được tinh chỉnh thêm trên toàn bộ tập dữ liệu để cải thiện hiệu suất.

Bước 5: Đánh giá mô hình

* Chuẩn bị dữ liệu kiểm thử: Một phần dữ liệu được giữ lại để đánh giá hiệu suất của mô hình.
* Đánh giá:

Ma trận nhầm lẫn: Hiển thị số lượng mẫu được phân loại đúng và sai.

Độ chính xác (Accuracy): Tỷ lệ các mẫu được phân loại đúng.

Độ nhạy (Recall): Tỷ lệ các mẫu dương được phân loại đúng.

Độ chính xác (Precision): Tỷ lệ các mẫu được phân loại là dương thực sự là dương.

F1-score: Trung bình hài hòa của Precision và Recall.

AUC-ROC: Diện tích dưới đường cong ROC, đo lường khả năng phân biệt của mô hình.

**5. Các kịch bản thử nghiệm**

**a**. **Kịch bản Demo1: Mô phỏng một số cách Tổng hợp Trọng số Thường dùng trong Federated Learning**

**Mục tiêu:**

Mô phỏng quá trình huấn luyện và tổng hợp trọng số của các mô hình phân loại trong môi trường Federated Learning, sử dụng ba thuật toán tổng hợp: FedAvg, WeightedFed, và Mean-Based Avg.

Các bước thực hiện:

1. Huấn luyện mô hình cục bộ tại các client

Client 1 và Client 2 nhận dữ liệu mạng đã chia và huấn luyện mô hình Logistic Regression độc lập trên dữ liệu của mình.

Mỗi client sử dụng thuật toán tối ưu hóa (ví dụ, Stochastic Gradient Descent) để huấn luyện mô hình, tạo ra các trọng số cục bộ (weights) tương ứng với dữ liệu mà họ sở hữu.

1. Tổng hợp trọng số từ các client  
   Sau khi huấn luyện xong, Server trung tâm sẽ nhận các trọng số từ các client và thực hiện tổng hợp các trọng số này bằng ba thuật toán tổng hợp khác nhau:

FedAvg: Server tính toán mô hình toàn cầu bằng cách tính trung bình có trọng số của các trọng số từ các client, với trọng số phụ thuộc vào kích thước dữ liệu của mỗi client.

WeightedFed: Server tính toán mô hình toàn cầu dựa trên trọng số của các client, nhưng trọng số sẽ phụ thuộc vào độ chính xác của mô hình cục bộ của mỗi client.

Mean-Based Avg: Server tính toán mô hình toàn cầu bằng cách tính trung bình đơn giản của các trọng số từ các client mà không xét đến kích thước dữ liệu hay độ chính xác.

1. Kiểm tra và đánh giá mô hình toàn cầu  
   Sau khi tổng hợp trọng số từ các client, Server trung tâm phân phối mô hình toàn cầu đã cập nhật trở lại cho các client để kiểm tra. Các client sử dụng mô hình này để đánh giá hiệu quả trên tập dữ liệu kiểm tra.
2. So sánh kết quả  
   Cuối cùng, hiệu quả của mô hình toàn cầu được đánh giá dựa trên các chỉ số như Accuracy, Precision, Recall, và F1 Score. Các kết quả từ ba thuật toán tổng hợp sẽ được so sánh để xem phương pháp nào mang lại hiệu suất tốt nhất trong trường hợp cụ thể này.

**Kết quả:**  
A table of numbers and symbols

Description automatically generated with medium confidence

Figure 3: Demo1 - Mean-Based Avg

A table with numbers and symbols

Description automatically generated with medium confidence

Figure 4: Demo1 - Weighted avg

A table with numbers and symbols

Description automatically generated

Figure 5: Demo1 - Federated avg

A table with numbers and symbols

Description automatically generated

Figure 6: Demo1 - So sánh 3 thuật toán

**Nhận xét:**

1. Precision và Recall giữa các lớp:

Với lớp Benign, tất cả các phương pháp đều có Precision và Recall rất cao (gần 1), cho thấy các mẫu này được phân loại tốt.

Các lớp như Backdoor, Worms, và Shellcode có Precision và Recall rất thấp, đặc biệt là với Federated Avg. Điều này cho thấy các phương pháp gặp khó khăn trong việc nhận diện các loại tấn công nhỏ hoặc hiếm.

1. F1-Score:

Các lớp có số lượng mẫu nhỏ (như Backdoor, Worms, Shellcode) có F1-score rất thấp (< 0.1). Ngược lại, các lớp phổ biến như Benign hoặc Exploits có F1-score cao hơn.

Sự chênh lệch F1-score giữa các phương pháp (Weighted Avg và Federated Avg) không lớn, nhưng có thể thấy Weighted Avg thường có kết quả tốt hơn một chút.

1. Macro Avg và Weighted Avg:

Macro Avg thấp hơn đáng kể so với Weighted Avg, phản ánh rằng các lớp có ít mẫu (như Worms, Shellcode) làm giảm hiệu suất trung bình.

Weighted Avg có kết quả cao hơn vì được tính theo tỷ trọng số lượng mẫu, ưu tiên các lớp lớn như Benign.

1. So sánh Accuracy:

Accuracy của các phương pháp là tương đối đồng đều (khoảng 0.9). Tuy nhiên, điều này có thể không phản ánh đầy đủ hiệu suất phân loại các lớp nhỏ vì Accuracy bị chi phối bởi các lớp lớn (như Benign).  
  
Kết luận:  
• Hiệu suất tốt: Các phương pháp xử lý tốt với các lớp lớn như Benign, Exploits.  
• Vấn đề cần cải thiện: Hiệu suất cho các lớp nhỏ như Backdoor, Worms và Shellcode rất thấp. Điều này cho thấy cần áp dụng các chiến lược như cân bằng dữ liệu hoặc tối ưu hóa thêm các thuật toán để nâng cao khả năng phát hiện các lớp hiếm.  
• Federated Learning: Trong bối cảnh Federated Learning, phương pháp này có vẻ duy trì hiệu suất tổng thể nhưng cần cải thiện ở các lớp nhỏ để phù hợp hơn với thực tế.

**b.** **Kịch bản Demo2: Federated Learning - Binary vs Multi-class Classification**

**Mục tiêu:**

Mô phỏng quá trình huấn luyện mô hình phân loại nhị phân và phân loại đa lớp trong môi trường Federated Learning, so sánh hiệu suất của mô hình trên hai loại bài toán phân loại này.

**Các bước thực hiện:**

1. Huấn luyện mô hình cục bộ tại các client

Các client tham gia vào hệ thống Federated Learning sẽ nhận tập dữ liệu phân loại riêng biệt, với mỗi client có dữ liệu khác nhau cho bài toán phân loại nhị phân (Binary Classification) hoặc phân loại đa lớp (Multi-class Classification).

Client 1 huấn luyện mô hình phân loại nhị phân trên dữ liệu của mình, trong khi Client 2 và Client 3 huấn luyện mô hình phân loại đa lớp. Mỗi client sử dụng thuật toán tối ưu hóa (ví dụ, Stochastic Gradient Descent) để huấn luyện mô hình phân loại của mình trên dữ liệu cục bộ.

1. Gửi trọng số mô hình về server  
   Sau khi huấn luyện, các client gửi trọng số mô hình cục bộ (weights) về server trung tâm. Mỗi client chỉ chia sẻ các trọng số của mô hình, không chia sẻ dữ liệu thô, giúp bảo vệ quyền riêng tư của người dùng.
2. Tổng hợp mô hình toàn cầu tại server  
   Server trung tâm nhận các trọng số từ các client và thực hiện tổng hợp để xây dựng mô hình toàn cầu. Các thuật toán tổng hợp như FedAvg hoặc WeightedFed sẽ được sử dụng để tính toán mô hình toàn cầu, kết hợp các trọng số từ các client dựa trên kích thước dữ liệu hoặc hiệu suất của mô hình.
3. Đánh giá mô hình toàn cầu  
   Sau khi tổng hợp, server trung tâm phân phối mô hình toàn cầu đã cập nhật cho tất cả các client. Mỗi client sử dụng mô hình toàn cầu này để kiểm tra trên một tập kiểm thử chung mà không bị ảnh hưởng bởi dữ liệu thô từ các client khác.
4. So sánh hiệu suất giữa Binary và Multi-class Classification  
   Cuối cùng, server trung tâm đánh giá mô hình toàn cầu trên các chỉ số như Accuracy, Precision, Recall, và F1 Score. Các kết quả từ bài toán phân loại nhị phân và đa lớp sẽ được so sánh để đánh giá hiệu quả của mô hình trong mỗi trường hợp.

**Kết quả:**

A table with numbers and symbols

Description automatically generated

Figure 7: Demo2 - MULTICLASS \_ FEDAVG

A table of numbers and symbols

Description automatically generated with medium confidence

Figure 8: Demo3 - Binary\_Value

**Nhận xét:** Nhận xét chi tiết theo từng metric và class:  
  
1. Precision, Recall, F1-score của từng lớp:

• Lớp Benign:  
• Precision = 0.999, Recall = 0.974, F1-score = 0.986: Đây là lớp có hiệu suất rất cao, ANN xử lý tốt lớp phổ biến này.  
• Lớp Backdoor, Shellcode, Worms:  
• Cả Precision, Recall và F1-score đều bằng 0. Điều này cho thấy ANN không nhận diện được các lớp này, có thể do thiếu dữ liệu hoặc sự không đồng đều trong phân phối dữ liệu.  
• Lớp DoS, Fuzzers, Reconnaissance:  
• Các lớp này có hiệu suất thấp, đặc biệt lớp DoS có Recall cực kỳ thấp (0.008), dẫn đến F1-score gần bằng 0. Điều này cho thấy ANN không đủ nhạy để phát hiện các loại tấn công hiếm hoặc nhỏ.  
• Lớp Exploits và Generic:  
• F1-score dao động khoảng 0.47–0.70. Đây là mức trung bình, ANN có thể phân loại nhưng chưa đủ tốt.  
  
2. Macro Avg và Weighted Avg:  
• Macro Avg (Precision = 0.48, Recall = 0.35, F1-score = 0.35):  
• Thấp, phản ánh rằng hiệu suất trên các lớp ít dữ liệu (Backdoor, Worms, Shellcode) kém.  
• Weighted Avg (Precision = 0.93, Recall = 0.91, F1-score = 0.91):  
• Cao hơn do lớp lớn (Benign) chiếm ưu thế, làm tăng trọng số trong tính toán.  
  
3. Tổng quan Accuracy = 0.91:  
• Accuracy cao, nhưng có thể bị ảnh hưởng bởi sự chênh lệch số lượng mẫu giữa các lớp (lớp Benign chiếm ưu thế).  
  
4. Các giá trị khác:  
• F1-Score = 0.949, AUC = 0.994: Mặc dù các chỉ số này cao, chúng chủ yếu bị chi phối bởi lớp Benign.  
• False Positive Rate = 0.025, False Negative Rate = 0.005:  
• False Negative Rate thấp cho thấy khả năng nhận diện các mẫu không phải tấn công khá tốt.  
• False Positive Rate tương đối thấp, đồng nghĩa ANN ít nhầm lẫn mẫu là tấn công khi không phải.  
  
Nhận xét tổng thể:  
1. Ưu điểm:  
• Hiệu suất tốt trên lớp lớn như Benign, với Precision, Recall, và F1-score rất cao.  
• Accuracy tổng thể cao, cho thấy khả năng phân loại tổng quát ổn định.  
• False Positive và False Negative Rates thấp, phù hợp với bài toán nhị phân.  
2. Hạn chế:  
• Các lớp nhỏ hoặc hiếm như Backdoor, Shellcode, và Worms không được phân loại đúng (F1-score = 0).  
• Với các lớp như DoS, hiệu suất thấp cho thấy ANN không phù hợp với các loại tấn công nhỏ hoặc hiếm khi áp dụng Federated Learning.  
3. Cải thiện:  
• Cân bằng dữ liệu: Tăng số lượng mẫu cho các lớp nhỏ.  
• Data Augmentation: Áp dụng các kỹ thuật tăng cường dữ liệu để giảm sự mất cân bằng.  
• Tối ưu ANN: Tăng độ phức tạp mô hình hoặc sử dụng kiến trúc mạng sâu hơn (e.g., Transformer-based models).  
• Tối ưu trọng số Federated Learning: Thay đổi chiến lược tổng hợp trọng số để tăng cường nhận diện lớp nhỏ.

**c.** **Kịch bản Demo: Phát Hiện Tấn Công Đặc Biệt Từ Dữ Liệu Riêng Của Khách Hàng**

**Mục tiêu:**

Mô phỏng quá trình huấn luyện và tổng hợp trọng số trong môi trường Federated Learning để phát hiện một loại tấn công mạng mới, với sự tham gia của hai client huấn luyện mô hình Logistic Regression độc lập.

**Các bước thực hiện:**

1. Huấn luyện mô hình cục bộ tại các client

Client 1 và Client 2 nhận tập dữ liệu mạng đã chia, mỗi client sẽ huấn luyện mô hình Logistic Regression riêng biệt trên dữ liệu của mình. Dữ liệu này bao gồm các thông tin mạng được phân loại, bao gồm cả những mẫu có thể chứa các cuộc tấn công mạng.

Mỗi client sử dụng thuật toán tối ưu hóa (ví dụ, Stochastic Gradient Descent) để huấn luyện mô hình phân loại các loại tấn công trong dữ liệu mạng, đặc biệt là một loại tấn công mới mà mô hình cần nhận diện.

1. Gửi trọng số mô hình về server  
   Sau khi hoàn tất quá trình huấn luyện, Client 1 và Client 2 gửi trọng số của mô hình cục bộ về server trung tâm. Mỗi client chỉ chia sẻ trọng số của mô hình mà không tiết lộ dữ liệu thô, đảm bảo bảo mật và quyền riêng tư.
2. Tổng hợp trọng số từ các client tại server  
   Server trung tâm nhận các trọng số mô hình từ các client và sử dụng thuật toán Mean-based Avg để tính toán mô hình toàn cầu. Thuật toán này tính trung bình trọng số của các mô hình từ các client để tạo ra một mô hình toàn cầu có khả năng nhận diện tấn công mới.
3. Đánh giá mô hình toàn cầu  
   Sau khi tổng hợp, server trung tâm phân phối mô hình toàn cầu cho các client để thực hiện đánh giá. Các client sử dụng mô hình này để kiểm tra khả năng phát hiện tấn công mới trên một tập dữ liệu kiểm thử mà không cần chia sẻ thêm dữ liệu thô.
4. Đánh giá hiệu suất mô hình phát hiện tấn công  
   Cuối cùng, server trung tâm đánh giá mô hình toàn cầu dựa trên các chỉ số như Accuracy, Precision, Recall, và F1 Score để kiểm tra khả năng phát hiện loại tấn công mới. Các kết quả này sẽ giúp so sánh hiệu quả của mô hình trong việc phát hiện tấn công mạng mà không cần tập trung dữ liệu.

**Kết quả:**  
A table with numbers and symbols

Description automatically generated

Figure 9: Demo3 - WITHOUT LABEL Exploits

A table with numbers and symbols

Description automatically generated

Figure 10: Demo3 - HAVE NEW\_LABEL

**Nhận xét:**

Hiệu Quả Phát Hiện Tấn Công:

* Lớp Benign: Cả hai bảng đều cho thấy mô hình Logistic Regression có hiệu suất rất cao với Precision, Recall, và F1-score dao động khoảng 0.98-0.99. Điều này chứng tỏ mô hình xử lý tốt lớp phổ biến này.
* Lớp Backdoor, Shellcode, Worms: Hiệu suất thấp với Precision, Recall, và F1-score gần như bằng 0. Điều này cho thấy mô hình gặp khó khăn trong việc nhận diện các lớp này, có thể do thiếu dữ liệu hoặc sự không đồng đều trong phân phối dữ liệu.
* Lớp DoS, Fuzzers, Reconnaissance: Hiệu suất ở mức trung bình, với F1-score dao động từ 0.15 đến 0.80. Đặc biệt, lớp Fuzzers và Reconnaissance có kết quả tương đối khả quan.
* Lớp Exploits và Generic: Kết quả dao động từ 0 đến 0.47. Điều này cho thấy mô hình có thể phân loại nhưng chưa đủ tốt.

Macro Avg và Weighted Avg:

* Macro Avg: Kết quả Precision, Recall, và F1-score từ 0.29 đến 0.35 phản ánh rằng hiệu suất trên các lớp ít dữ liệu kém.
* Weighted Avg: Kết quả cao hơn (Precision = 0.85-0.94, Recall = 0.87-0.91, F1-score = 0.85-0.91) do lớp Benign chiếm ưu thế, ảnh hưởng đáng kể đến tính toán.

Tổng quan Accuracy:

* Accuracy dao động từ 0.87 đến 0.90. Điều này chứng tỏ mô hình có khả năng phân loại tổng quát ổn định, nhưng có thể bị ảnh hưởng bởi sự chênh lệch số lượng mẫu giữa các lớp.

Nhận xét tổng thể:

1. Ưu điểm:
   * Hiệu suất tốt trên lớp lớn như Benign, với Precision, Recall, và F1-score rất cao.
   * Accuracy tổng thể cao, cho thấy khả năng phân loại tổng quát ổn định.
2. Hạn chế:
   * Các lớp nhỏ hoặc hiếm như Backdoor, Shellcode, và Worms không được phân loại đúng (F1-score thấp).
   * Với các lớp như DoS và Exploits, hiệu suất thấp cho thấy mô hình không phù hợp với các loại tấn công nhỏ hoặc hiếm khi áp dụng Federated Learning.
3. Cải thiện:
   * Cân bằng dữ liệu: Tăng số lượng mẫu cho các lớp nhỏ.
   * Data Augmentation: Áp dụng các kỹ thuật tăng cường dữ liệu để giảm sự mất cân bằng.
   * Tối ưu mô hình: Tăng độ phức tạp mô hình hoặc sử dụng kiến trúc mạng sâu hơn.
   * Tối ưu trọng số Federated Learning: Thay đổi chiến lược tổng hợp trọng số để tăng cường nhận diện lớp nhỏ.

**Chương III: Đánh giá và kết luận**

**1. Kết Luận**

**2. Hướng phát triển trong tương lai**

**References**

[1] Statio.vn, "Federated Learning trong AI là gì? Giải thích chi tiết về học liên kết, cách hoạt động và lợi ích cho bảo mật dữ liệu," [Online]. Available: https://statio.vn/blog/federated-learning-trong-ai-la-gi-giai-thich-chi-tiet-ve-hoc-lien-ket-cach-hoat-ong-va-loi-ich-cho-bao-mat-du-lieu. [Accessed: Dec. 8, 2024].

[2] UNB, "Datasets," [Online]. Available: https://www.unb.ca/cic/datasets/. [Accessed: Dec. 8, 2024].

[3] J. McMahan, E. Moore, D. Ramage, and B. A. y Federated Learning, "Communication-efficient learning of deep networks from decentralized data," arXiv, Apr. 2021. [Online]. Available: https://arxiv.org/abs/2104.11375. [Accessed: Dec. 8, 2024].

[4] X. Zhang, Y. Li, and Z. Liu, "Application of federated learning in manufacturing," ResearchGate, 2022. [Online]. Available: https://www.researchgate.net/publication/362591286\_Application\_of\_federated\_learning\_in\_manufacturing/figures?lo=1. [Accessed: Dec. 8, 2024].

[5] S. K. S. Gupta, M. Meena, and P. Agarwal, "Federated Learning and Its Applications in Intelligent Systems," Electronics, vol. 12, no. 12, p. 5806, Dec. 2023. [Online]. Available: https://www.mdpi.com/2076-3417/12/12/5806. [Accessed: Dec. 8, 2024].

[6] H. Mohammadian, A. H. Lashkari, A. Ghorbani. “Poisoning and Evasion: Deep Learning-Based NIDS under Adversarial Attacks,” 21st Annual International Conference on Privacy, Security and Trust (PST), 2024.

[7] Moustafa, Nour, and Jill Slay. "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 dataset and the comparison with the KDD99 dataset." *Information Security Journal: A Global Perspective*, vol. 25, no. 1-3, pp. 18-31, 2016. DOI: 10.1080/19393555.2015.1125974.

[8] UNSW, "UNSW-NB15 Dataset," [Online]. Available: https://research.unsw.edu.au/projects/unsw-nb15-dataset. [Accessed: Dec. 8, 2024].

[9] K. Bonawitz, H. Eichner, W. Grieskamp, et al., "Towards Federated Learning at Scale: System Design," *Proceedings of the 2nd SysML Conference*, 2020.

[10] S. McMahan, E. Moore, D. Ramage, et al., "Communication-Efficient Learning of Deep Networks from Decentralized Data," *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.