| **ĐẠI HỌC QUỐC GIA HÀ NỘI**  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**      **Ngô Thành Văn**  **SỬ DỤNG HỌC LIÊN HỢP THÍCH NGHI ĐỂ PHÁT HIỆN CÁC TẤN CÔNG TỪ CHỐI DỊCH VỤ PHÂN TÁN**  **KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**  **Ngành: Công nghệ thông tin định hướng thị trường Nhật Bản**  **HÀ NỘI – 2024** |
| --- |

| **ĐẠI HỌC QUỐC GIA HÀ NỘI**  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**    **Ngô Thành Văn**  **SỬ DỤNG HỌC LIÊN HỢP THÍCH NGHI ĐỂ PHÁT HIỆN CÁC TẤN CÔNG TỪ CHỐI DỊCH VỤ PHÂN TÁN**  **KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**  **Ngành: Công nghệ thông tin định hướng thị trường Nhật Bản**  **Cán bộ hướng dẫn: TS. Nguyễn Đại Thọ**  **HÀ NỘI – 2024** |
| --- |

# LỜI CẢM ƠN

Lời đầu tiên, em xin gửi lời cảm ơn sâu sắc tới thầy TS. Nguyễn Đại Thọ đã luôn tận tình chỉ dẫn em từ định hướng, kiến thức, và phương pháp nghiên cứu, cũng như luôn dành cho em những lời khuyên quý báu trong suốt thời gian em thực hiện đề tài này.

Em xin bày tỏ lòng biết ơn tới các thầy cô Khoa Công nghệ thông tin, trường Đại học Công nghệ, Đại học Quốc gia Hà Nội đã giảng dạy và đã tạo cho em những điều kiện tốt nhất trong suốt quá trình học tập cũng như hoàn thành khóa luận.

Con xin gửi lời cảm ơn đến gia đình, người thân đã luôn bên cạnh động viên và giúp đỡ con trong suốt thời gian qua. Tình cảm của mọi người là niềm khích lệ to lớn giúp con có được thành quả như ngày hôm nay.

Cuối cùng, tôi xin cảm ơn đến tất cả bạn bè, đặc biệt là các bạn lớp K65J đã luôn bên cạnh và giúp đỡ tôi trong học tập cũng như trong cuộc sống.

Trong quá trình thực hiện khóa luận, do thời gian, kiến thức và điều kiện nghiên cứu còn hạn chế nên không tránh khỏi thiếu sót, rất mong nhận được ý kiến đóng góp từ người đọc để tác giả có thể tiếp tục phát triển đề tài này.

Hà Nội, tháng 6 năm 2024

Ngô Thành Văn

# TÓM TẮT

**Tóm tắt:** Học liên hợp đã dần có chỗ đứng trong ngành an toàn thông tin như một cách để hợp tác huấn luyện cho mô hình học máy với các dữ liệu phân tán về tấn công mạng. Tuy nhiên, những ứng dụng của học liên hợp trong an toàn thông tin ẩn chứa một vài vấn đề chưa được đề cập tới. Với việc thuật toán nền tảng FedAvg yêu cầu phải có một tập dữ liệu kiểm tra trên hệ thống để đánh giá hiệu quả của mô hình. Đối với một số ngành khác, điều kiện này có thể dễ dàng đạt được nhưng đối với riêng ngành an toàn thông tin, việc chia sẻ những dữ liệu nhạy cảm là không thể đáp ứng. FLAD (Federated Learning Approach to DDoS) hay học liên hợp thích nghi ra đời với mục tiêu loại bỏ rào cản trên khi không yêu cầu các bên phải chia sẻ bất kỳ dữ liệu nào ở bất kỳ thời điểm nào. FLAD đáp ứng tốt hơn yêu cầu về đảm bảo tính riêng tư khi không đặt ra yêu cầu server cần sở hữu một tập dữ liệu để đánh giá. Tuy nhiên, FLAD vẫn sử dụng thuật toán trung bình của FedAvg làm nền móng, dẫn tới sự mất mát lớn của mô hình toàn cục sau mỗi lần cập nhật. Đối với hệ thống bao gồm hàng nghìn tới hàng triệu khách hàng, điều này có thể là một vấn đề ảnh hưởng tới hiệu năng. Hơn nữa, để đáp ứng được những nhu cầu hiện nay về an toàn thông tin, nhu cầu về việc phân loại tấn công thay vì phát hiện để tăng độ hiểu biết về kẻ tấn công ngày càng được gia tăng. Mô hình FLAD được xây dựng cho bài toán phát hiện đã không còn phù hợp với bối cảnh hiện nay. Trung bình phân đoạn (Partial Average) là thuật toán có thể khắc phục được những hạn chế của FLAD, đồng thời có thể đáp ứng được bài toán phân loại. Khóa luận này đưa ra một đề xuất áp dụng thuật toán trung bình phân đoạn vào FLAD để gia tăng độ chuẩn xác khi phân loại các kiểu tấn công DDoS hiện nay với bài toán học liên hợp mà máy chủ không có tập dữ liệu kiểm tra. Kết quả thu được độ chính xác của FLAD khi kết hợp với thuật toán trung bình phân đoạn là 0.769, so với 0.601 của mô hình gốc.

***Từ khóa:*** học liên hợp, thích nghi, FLAD, FedAvg, Partial Average

# LỜI CAM ĐOAN

Tôi xin cam đoan rằng những nghiên cứu và nội dung trình bày đều được tôi thực hiện dưới sự hướng dẫn của TS. Nguyễn Đại Thọ. Các số liệu, những kết luận nghiên cứu được trình bày trong luận văn này hoàn toàn trung thực và không sao chép từ các tài liệu, công trình nghiên cứu của người khác.

Tôi xin hoàn toàn chịu trách nhiệm về lời cam đoan này.

Hà Nội, ngày 24 tháng 06 năm 2024

Sinh viên

Ngô Thành Văn

# MỤC LỤC

[LỜI CẢM ƠN 3](#_30j0zll)

[TÓM TẮT 4](#_1fob9te)

[LỜI CAM ĐOAN 5](#_3znysh7)

[MỤC LỤC 6](#_2et92p0)

[Danh sách hình vẽ 7](#_tyjcwt)

[Danh sách bảng 7](#_a76rxwsp0351)

[Danh sách thuật toán 7](#_2syquavnynht)

[Chương 1. Mở đầu 8](#_1t3h5sf)

[1.1. Lý do chọn đề tài 8](#_4d34og8)

[1.2. Đóng góp và kết quả của khóa luận 9](#_2s8eyo1)

[1.3. Cấu trúc của khóa luận 10](#_17dp8vu)

[Chương 2. Tổng quan về học liên hợp 11](#_3rdcrjn)

[2.1. Bối cảnh chung 11](#_26in1rg)

[2.2. Khái niệm học liên hợp 14](#_35nkun2)

[2.3. Các nền tảng của học liên hợp 16](#_1ksv4uv)

[2.3.1. Các mạng neuron học sâu 16](#_pilnouj07c2e)

[2.3.2. Các mô hình học máy truyền thống 19](#_dm8ye31e7qpe)

[2.3.3 Phân đoạn dữ liệu trong học liên hợp 21](#_20fi14rp4wqg)

[2.3.4. Cá biệt hóa mô hình 23](#_n8m0qq3gvvpv)

[2.4. Các phương pháp học liên hợp liên quan đến khóa luận 25](#_44sinio)

[2.4.1. FedAvg 25](#_wle3p1mprgn7)

[2.4.2. Học liên hợp thích nghi - FLAD 26](#_yqzljfdijbpe)

[2.4.3. Trung bình phân đoạn - Partial Average 30](#_3evqnc7ey5ob)

[Chương 3: Đề xuất áp dụng thuật toán trung bình phân đoạn vào mô hình học liên hợp thích nghi 33](#_z337ya)

[3.1. Phân tích hạn chế của mô hình học liên hợp thích nghi FLAD và khả năng áp dụng thuật toán trung bình phân đoạn 33](#_3j2qqm3)

[3.1.1 Hạn chế của học liên hợp thích nghi FLAD 33](#_q0hxnnco9x3k)

[3.1.2 Áp dụng thuật toán trung bình phân đoạn 33](#_aibkbao2sw9a)

[3.2. Phương pháp áp dụng 34](#_e8nftzaxm3yc)

[3.2.3 Thay thế FedAvg trong FLAD bằng thuật toán trung bình phân đoạn 34](#_lsp87lk9r925)

[Chương 4: Thực nghiệm và đánh giá kết quả 37](#_111kx3o)

[4.1. Tập dữ liệu 37](#_3l18frh)

[4.2 Thực nghiệm 39](#_ayiyy1y6p9m)

[4.3 Đánh giá kết quả 41](#_43ky6rz)

[Chương 5: Kết luận 43](#_2iq8gzs)

[TÀI LIỆU THAM KHẢO 44](#_xvir7l)

# 

# Danh sách hình vẽ

[Hình 1. Tổng quan về học liên hợp 12](#_5zew8lrmm80q)

[Hình 2. Khái niệm học liên hợp 14](#_bauty6q1rxfx)

[Hình 3. Dữ liệu được phân vùng theo chiều ngang 22](#_e97yuawcs7x)

[Hình 4. Dữ liệu được phân vùng theo chiều dọc 23](#_h891b5tpwlku)

[Hình 5 Đặc điểm các mối đe dọa tiềm tàng của học liên hợp cũng như các đối thủ tiềm ẩn. 24](#_aci0v1cqto5q)

[Hình 6. Mô hình trung bình toàn phần và phân đoạn trên không gian tham số 32](#_rea466q4mhy6)

## 

# Danh sách bảng

[Bảng 1. Các tham số trong mô hình FLAD 28](#_8962ll38q7n1)

[Bảng 2. Cấu trúc mô hình 34](#_eokeuq1zm3dv)

[Bảng 3. Thống kê các kiểu tấn công trong tệp dữ liệu CIC-DDoS2019 38](#_vd1az7qbaa5i)

[Bảng 4: Kích thước bộ dữ liệu đầu vào 39](#_keeyomg5ga0j)

[Bảng 5: Tham số huấn luyện 40](#_a1ym5252xtz4)

[Bảng 6. Kết quả thực nghiệm 41](#_s4xilhkb203)

# Danh sách thuật toán

[Thuật toán 1. Học liên hợp thích nghi 28](#_dn5ro7618v4y)

[Thuật toán 2. Chọn khách hàng cần huấn luyện 29](#_htxb45tpz3v0)

[Thuật toán 3. Huấn luyện ở phía khách hàng 30](#_y6k4ex2aigby)

[Thuật toán 4. Trung bình phân đoạn 31](#_qd0fxv4fr2op)

# 

# Chương 1. Mở đầu

## 1.1. Lý do chọn đề tài

.Vấn đề về tấn công DDoS (Distributed Denial of Service) luôn là một trong những mối lo lớn của các tổ chức, doanh nghiệp và cả cá nhân khi truy cập Internet. Đây không chỉ là một trở ngại kỹ thuật, mà còn là một thách thức về an ninh mạng, có thể gây ra những hậu quả nghiêm trọng cho sự hoạt động thông tin và kinh doanh. Điều này đã thúc đẩy sự quan tâm lớn hơn đến việc nghiên cứu và phát triển các phương pháp phát hiện và phòng chống tấn công DDoS.

Lý do chọn đề tài này là vì tấn công DDoS không chỉ đang trở nên phổ biến hơn mà còn ngày càng trở nên phức tạp và tinh vi hơn. Trong bối cảnh hiện nay, khi mà sự phụ thuộc vào mạng Internet ngày càng lớn, đặc biệt là trong thời đại của Công nghệ Thông tin và Cuộc cách mạng Công nghiệp 4.0, việc bảo vệ hệ thống mạng trở nên quan trọng hơn bao giờ hết.

Một trong những lý do chính là sự gia tăng về tính phức tạp và tầm ảnh hưởng của các cuộc tấn công DDoS. Các hacker và nhóm tội phạm mạng đã không ngừng phát triển các kỹ thuật tấn công mới, từ việc sử dụng botnet (mạng bot) lớn đến việc sử dụng các kỹ thuật tấn công phức tạp như tấn công cổ chai trên cơ sở dữ liệu hoặc tấn công ứng dụng. Những cuộc tấn công này không chỉ gây ra sự cố kỹ thuật mà còn có thể làm ngưng trệ hoạt động kinh doanh, gây mất uy tín và thiệt hại tài chính lớn cho các tổ chức.

Ngoài ra, việc tăng cường về tính toàn cầu của Internet cũng làm cho các tấn công DDoS trở nên nguy hiểm hơn. Với sự kết nối mạng lan rộng và việc sử dụng dịch vụ đám mây ngày càng phổ biến, một cuộc tấn công có thể có tác động toàn cầu, ảnh hưởng đến hàng triệu người dùng và hàng ngàn tổ chức trên khắp thế giới. Do đó, việc phát hiện và ngăn chặn các cuộc tấn công trở nên cấp bách và đòi hỏi sự chú ý đặc biệt.

Cùng với đó, sự phát triển của Internet of Things (IoT) cũng tạo ra một mối đe dọa mới cho việc bảo vệ mạng. Với hàng tỷ thiết bị được kết nối, các hacker có thể tận dụng sức mạnh của các botnet IoT để thực hiện các cuộc tấn công DDoS lớn và phức tạp mà không cần nhiều kỹ năng kỹ thuật. Điều này làm cho việc phát hiện và ngăn chặn các cuộc tấn công trở nên khó khăn hơn, đặc biệt là khi phải đối mặt với sự đa dạng và khả năng phát triển nhanh chóng của các loại tấn công này.

Một yếu tố quan trọng khác là sự cần thiết của việc nghiên cứu và phát triển các công nghệ phát hiện tiên tiến để đối phó với các tấn công DDoS. Trong khi các hacker không ngừng cải tiến các kỹ thuật tấn công của họ, thì việc phát triển các phương pháp phát hiện mới là cực kỳ quan trọng để duy trì tính hiệu quả của hệ thống bảo mật mạng. Các phương pháp phát hiện mới có thể bao gồm việc sử dụng trí tuệ nhân tạo và học máy để phát hiện các mẫu tấn công mới, phân tích dữ liệu lớn để xác định các hành vi bất thường, và sử dụng các phương pháp tiên tiến khác để nhanh chóng phát hiện và ngăn chặn các cuộc tấn công.

Tóm lại, việc chọn đề tài về phát hiện và phân loại tấn công DDoS là cực kỳ phù hợp trong bối cảnh hiện nay do sự phổ biến và nguy hiểm của các cuộc tấn công này. Nghiên cứu về vấn đề này giúp cải thiện sự hiểu biết về công nghệ và an ninh mạng để đảm bảo rằng các biện pháp phòng chống tấn công DDoS được phát triển và triển khai một cách hiệu quả và đúng đắn.

## 

## 1.2. Đóng góp và kết quả của khóa luận

Khóa luận đề xuất việc áp dụng phương pháp trung bình phân đoạn vào FLAD để nâng cao độ chuẩn xác của mô hình học máy cho bài toán liên hợp phân loại khi máy chủ tập trung không có tập dữ liệu để đánh giá.

Mô hình có nhiệm vụ phân loại các kiểu tấn công DDoS với đầu vào là thông tin của các gói tin độc hại, và đầu ra của mô hình là kiểu tấn công. Phương pháp trung bình phân đoạn kết hợp với học liên hợp thích nghi cho kết quả về độ chuẩn xác khi phân loại là 0.769 so với 0.601 của mô hình gốc.

## 1.3. Cấu trúc của khóa luận

Nội dung chính của khóa luận được trình bày qua 5 chương:

**Chương 1:** Mở đầu

Chương này nêu lên lý do lựa chọn đề tài, đóng góp và cấu trúc của khóa luận.

**Chương 2:** Tổng quan về học liên hợp

Chương này trình bày về những kiến thức chung của học liên hợp

**Chương 3:** Đề xuất áp dụng thuật toán trung bình phân đoạn vào mô hình học liên hợp thích nghi

Chương này trình bày tổng quan bài toán, cơ sở lý thuyết và mô hình được sử dụng trong khóa luận.

**Chương 4:** Thực nghiệm và đánh giá kết quả

Chương này đưa ra kết quả, đánh giá phương pháp đã sử dụng.

**Chương 5:** Kết luận

Chương này kết luận về những kết quả thu được trong đề tài.

# Chương 2. Tổng quan về học liên hợp

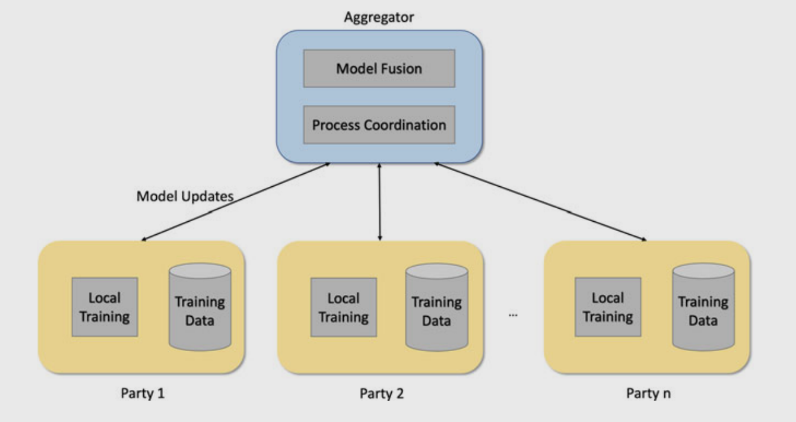
Những nội dung trong chương này phần lớn được trích dẫn từ cuốn Federated Learning: A Comprehensive Overview of Methods and Applications [22] và cuốn Federated Learning Fundamentals and Advances [23]

## 2.1. Bối cảnh chung

Học máy (Machine Learning - ML) đã trở thành một kỹ thuật quan trọng để phát triển nhận thức và giải quyết những khó khăn trong việc phân tích và phát triển thuật toán. Những ứng dụng trong thị giác máy, phát hiện giọng nói, và ngôn ngữ tự nhiên được phát triển mạnh mẽ nhờ vào Mạng neuron sâu (Deep Neural Networks - DNNs) và phần cứng dùng để tính toán phức tạp. Đồng thời, những kỹ thuật học máy cổ điển như cây quyết định, hồi quy tuyến tính, và máy hỗ trợ véc-tơ (SVMs) được sử dụng nhiều hơn ở một số vấn đề liên quan đến cấu trúc dữ liệu.

Ứng dụng của học máy phụ thuộc nhiều vào sự sẵn có của dữ liệu đào tạo chất lượng cao. Tuy nhiên, đôi khi, những cân nhắc về quyền riêng tư ngăn cản việc đưa dữ liệu đào tạo đến kho lưu trữ dữ liệu trung tâm để quản lý và quản lý cho quy trình học máy. Học liên hợp (FL) là một cách tiếp cận được đề xuất bởi McMahan et al [1], để huấn luyện các mô hình học máy về dữ liệu huấn luyện ở các vị trí khác nhau, không yêu cầu thu thập dữ liệu tập trung.

Theo cách tiếp cận học liên hợp, một nhóm các bên riêng biệt kiểm soát dữ liệu đào tạo tương ứng của họ sẽ hợp tác để đào tạo mô hình học máy. Họ làm điều này mà không chia sẻ dữ liệu đào tạo của mình với các bên hoặc bất kỳ tổ chức bên thứ ba nào khác. Các bên tham gia hợp tác còn được gọi là khách hàng hoặc thiết bị. Các bên có thể là nhiều thứ khác nhau, bao gồm các thiết bị tiêu dùng như điện thoại thông minh hoặc ô tô, cũng như dịch vụ đám mây của các nhà cung cấp khác nhau, trung tâm dữ liệu xử lý dữ liệu doanh nghiệp ở các quốc gia khác nhau, kho ứng dụng trong công ty hoặc các hệ thống nhúng như robot sản xuất trong một cơ sở nhà máy ô tô.

Trong khi việc hợp tác học liên hợp có thể được tiến hành theo nhiều cách khác nhau, hình thức phổ biến nhất của nó được phác thảo trong Hình 1.1. Theo cách tiếp cận này, một bộ tổng hợp, đôi khi được gọi là máy chủ hoặc điều phối viên, sẽ tạo điều kiện thuận lợi cho việc cộng tác. Các bên thực hiện quy trình đào tạo cục bộ dựa trên dữ liệu đào tạo riêng của họ. Khi quá trình đào tạo cục bộ của họ hoàn tất, họ sẽ gửi các tham số mô hình của mình đến bộ tổng hợp dưới dạng cập nhật mô hình. Loại cập nhật mô hình phụ thuộc vào loại mô hình học máy được đào tạo. Ví dụ, đối với mạng nơ-ron, các bản cập nhật mô hình có thể là trọng số của mạng. Sau khi đơn vị tổng hợp nhận được các bản cập nhật mô hình từ các bên, chúng có thể được hợp nhất thành một mô hình chung, quá trình này được gọi là tổng hợp mô hình. Trong ví dụ về mạng thần kinh, điều này có thể đơn giản như việc lấy trung bình các trọng số, như được đề xuất trong thuật toán FedAvg [2]. Sau đó, mô hình hợp nhất thu được sẽ được phân phối lại cho các bên dưới dạng cập nhật mô hình để tạo cơ sở cho vòng học tập tiếp theo. Quá trình này có thể được lặp lại qua nhiều vòng cho đến khi quá trình huấn luyện hội tụ. Vai trò của bộ tổng hợp là điều phối quá trình học tập và trao đổi thông tin giữa các bên và thực hiện thuật toán tổng hợp để hợp nhất các tham số mô hình từ mỗi bên thành một mô hình chung. Kết quả của quá trình học liên hợp là một mô hình dựa trên dữ liệu huấn luyện của tất cả các bên, trong khi dữ liệu huấn luyện không bao giờ được chia sẻ.

#### Hình 1. Tổng quan về học liên hợp

Cách tiếp cận học liên hợp dường như có liên quan đến việc học phân tán trên các cụm, đây là cách tiếp cận phổ biến cho các nhiệm vụ ML lớn. Học tập phân tán sử dụng một cụm nút điện toán để chia sẻ nỗ lực tính toán của ML và từ đó đẩy nhanh quá trình học tập. Học tập phân tán thường sử dụng máy chủ tham số để tổng hợp kết quả từ các nút, không giống như mô hình liên hợp. Tuy nhiên, nó khác ở một số điểm quan trọng. Trong học liên hợp, việc phân phối và số lượng dữ liệu không được kiểm soát tập trung và có thể không xác định được liệu tất cả dữ liệu đào tạo có được giữ kín hay không. Không thể đưa ra các giả định về tính độc lập và phân phối dữ liệu giống hệt nhau (IID) giữa các bên. Tương tự, một số bên có thể có nhiều dữ liệu hơn những bên khác, dẫn đến sự mất cân bằng về dữ liệu giữa các bên. Trong học tập phân tán, dữ liệu được quản lý tập trung và phân phối đến các nút khác nhau trong các phân đoạn, với thực thể trung tâm nhận thức được các thuộc tính ngẫu nhiên của dữ liệu. Sự mất cân bằng và tính không IID của dữ liệu bên phải được tính đến khi thiết kế thuật toán đào tạo học liên hợp.

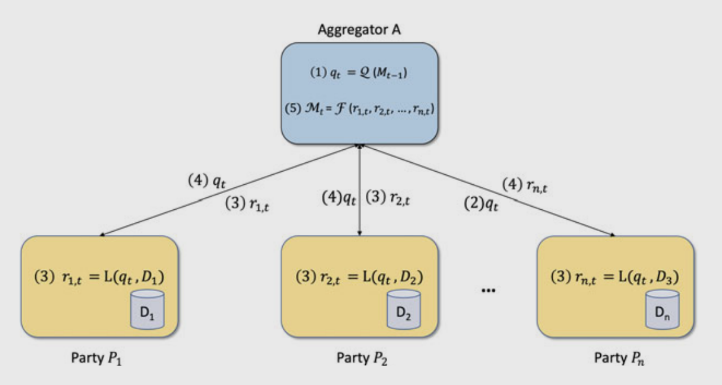
Ngược lại, ở học liên hợp, số lượng bên có thể khác nhau tùy theo trường hợp sử dụng. Đào tạo một mô hình về bộ dữ liệu ở các trung tâm dữ liệu khác nhau của một công ty đa quốc gia có thể có ít hơn mười bên. Điều này thường được gọi là trường hợp sử dụng doanh nghiệp[3] hoặc cross-silo[4]. Đào tạo về dữ liệu của một ứng dụng điện thoại di động có thể có hàng trăm triệu bên đóng góp. Điều này thường được gọi là trường hợp sử dụng nhiều thiết bị [4]. Trong trường hợp sử dụng doanh nghiệp, điều quan trọng chung là phải xem xét các cập nhật mô hình từ tất cả hoặc hầu hết các bên trong mỗi vòng. Trong trường hợp sử dụng thiết bị, mỗi vòng học liên hợp sẽ chỉ bao gồm một mẫu phụ - có thể lớn - trong tổng số bộ thiết bị. Trong trường hợp sử dụng doanh nghiệp, quy trình học liên hợp xem xét danh tính của các bên liên quan và có thể sử dụng danh tính này trong quá trình đào tạo và xác minh. Trong trường hợp sử dụng trên nhiều thiết bị, nhận dạng bên thường không quan trọng và một bên có thể chỉ tham gia vào một vòng đào tạo.

Trong trường hợp sử dụng thiết bị, hơn là trong trường hợp doanh nghiệp, có thể giả định lỗi giao tiếp của một số thiết bị do có số lượng lớn người tham gia. Điện thoại di động có thể bị tắt hoặc thiết bị có thể nằm trong vùng phủ sóng kém. Điều này có thể được quản lý bởi các bên lấy mẫu và thiết lập giới hạn thời gian để thực hiện tổng hợp hoặc các kỹ thuật giảm thiểu khác. Trong trường hợp sử dụng của doanh nghiệp, lỗi giao tiếp phải được quản lý cẩn thận vì sự đóng góp của từng bên có liên quan do số lượng người tham gia ít.

## 2.2. Khái niệm học liên hợp

Giống như bất kỳ nhiệm vụ học máy nào, học liên hợp huấn luyện một mô hình M đại diện cho hàm dự đoán f trên dữ liệu huấn luyện D. M có thể có cấu trúc của mạng thần kinh hoặc bất kỳ mô hình phi thần kinh nào khác. Ngược lại với học máy tập trung, D được phân vùng giữa n bên *P* = {*P*1*, P*2*,...,Pn*}, trong đó mỗi bên *Pk* ∈ P sở hữu một tập dữ liệu đào tạo riêng *Dk*. Một quy trình học liên hợp bao gồm một tập hợp A và một tập hợp các bên P. Điều quan trọng cần lưu ý là *Dk* chỉ có thể được truy cập bởi bên *Pk*. Nói cách khác, không bên nào có kiến thức về bất kỳ tập dữ liệu nào khác ngoài tập dữ liệu của mình và A không có kiến thức về bất kỳ tập dữ liệu nào.

Quá trình học liên hợp được tiến hành như thế nào ở mức trừu tượng này được thể hiện trong Hình 1.2.



#### Hình 2. Khái niệm học liên hợp

Để huấn luyện mô hình học máy toàn cầu M, đơn vị tổng hợp và các bên thực hiện thuật toán học liên hợp được thực thi theo cách phân tán tại đơn vị tổng hợp và các bên. Các thành phần thuật toán chính là hàm huấn luyện cục bộ L của mỗi bên, thực hiện huấn luyện cục bộ trên tập dữ liệu *Dk* và hàm tổng hợp F của trình tổng hợp, kết hợp các kết quả của L của mỗi bên thành một mô hình chung mới. Có thể có một tập hợp các lần lặp lại quá trình huấn luyện và kết hợp cục bộ được gọi là các vòng, sử dụng chỉ số t. Việc thực hiện thuật toán được điều phối bằng cách gửi tin nhắn giữa các bên và đơn vị tổng hợp. Quá trình tổng thể diễn ra như sau:

1. Quá trình bắt đầu tại bộ tổng hợp. Để huấn luyện mô hình, bộ tổng hợp sử dụng hàm Q lấy đầu vào là mô hình của vòng huấn luyện trước M*t*−1 ở vòng t và tạo truy vấn *qt* cho vòng hiện tại. Khi quá trình bắt đầu, M0 có thể trống hoặc chỉ được gieo hạt ngẫu nhiên. Ngoài ra, một số thuật toán học liên hợp có thể bao gồm các đầu vào bổ sung cho Q và có thể điều chỉnh các truy vấn cho mỗi bên, nhưng để đơn giản cho việc thảo luận và không mất tính tổng quát, phương pháp đơn giản hơn này được sử dụng.

2. Truy vấn *qt* được gửi đến các bên và yêu cầu thông tin về mô hình cục bộ tương ứng của họ hoặc thông tin tổng hợp về tập dữ liệu của mỗi bên. Các truy vấn mẫu bao gồm các yêu cầu về độ dốc hoặc trọng số mô hình của mạng thần kinh hoặc số lượng cây quyết định.

3. Khi nhận *qt*, quy trình huấn luyện cục bộ thực hiện chức năng huấn luyện cục bộ L lấy truy vấn đầu vào *qt* và tập dữ liệu cục bộ *Dk* và đưa ra bản cập nhật mô hình *rk,t*. Thông thường, truy vấn *qt* chứa thông tin mà bên có thể sử dụng để khởi tạo quy trình đào tạo cục bộ. Ví dụ, điều này bao gồm trọng số mô hình của mô hình chung, mới Mt để khởi tạo quá trình huấn luyện cục bộ hoặc thông tin khác cho các loại mô hình khác nhau.

4. Khi L hoàn thành, *rk,t*  được gửi lại từ bên *Pk* đến tập hợp A, tập hợp này sẽ thu thập tất cả *rk,t*  từ tất cả các bên.

5. Khi bộ tổng hợp nhận được các bản cập nhật mô hình của tất cả các bên dự kiến *Rt* = *(r*1*,t, r*2*,t,...,rn,t)* , chúng sẽ được xử lý bằng cách áp dụng hàm tổng hợp F lấy làm đầu vào *Rt*  và trả về *Mt*.

Quá trình này có thể được thực hiện qua nhiều vòng và tiếp tục cho đến khi đáp ứng tiêu chí kết thúc, ví dụ: số vòng huấn luyện tối đa *tmax* đã trôi qua, dẫn đến mô hình toàn cầu cuối cùng M = M*tmax*. Số vòng cần thiết có thể rất khác nhau, từ việc hợp nhất một mô hình theo cách tiếp cận Naive Bayes đến nhiều vòng đào tạo cho các thuật toán học máy dựa trên độ dốc điển hình.

Hàm huấn luyện cục bộ L, hàm tổng hợp F và hàm tạo truy vấn Q thường là một bộ miễn phí được thiết kế để hoạt động cùng nhau. L tương tác với tập dữ liệu thực tế và thực hiện huấn luyện cục bộ, tạo ra bản cập nhật mô hình *rk,t*. Nội dung của *Rt* là đầu vào của F và do đó phải được F giải thích, điều này tạo ra mô hình tiếp theo *Mt* của đầu vào này. Nếu cần một vòng nữa, Q sẽ tạo một truy vấn khác.

Các biến thể khác nhau cho cách tiếp cận cơ bản này với học liên hợp: Trong trường hợp học liên hợp đa thiết bị, số lượng bên tham gia thường lớn, lên tới hàng triệu. Không phải tất cả các bên tham gia vào mọi vòng. Trong trường hợp này, Q không chỉ xác định truy vấn mà còn xác định bên nào *Ps* ⊂ P sẽ đưa vào vòng truy vấn tiếp theo. Việc lựa chọn đảng có thể là ngẫu nhiên, dựa trên đặc điểm của đảng hoặc dựa trên thành tích đóng góp trước đó.

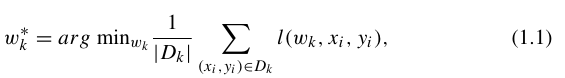
Ngoài ra, các truy vấn đối với mỗi bên có thể khác nhau, trong đó F cần tích hợp kết quả của các truy vấn khác nhau để tạo ra một mô hình mới *Mt*.

Trong khi cách tiếp cận với một bộ tổng hợp duy nhất được sử dụng phổ biến và thiết thực nhất cho hầu hết các kịch bản, các kiến trúc học liên hợp thay thế khác đã được đề xuất. Ví dụ: mỗi bên *Pk* có thể có tập hợp *Ak* tổng hợp được liên kết của riêng mình, truy vấn các bên khác; tập hợp các bên có thể được phân chia giữa các bộ tổng hợp và quá trình tổng hợp theo cấp bậc có thể diễn ra.

## 2.3. Các nền tảng của học liên hợp

### 2.3.1. Các mạng neuron học sâu

DNN đã trở nên rất phổ biến và phù hợp với học liên hợp theo cách tương đối đơn giản trong cách tiếp cận cơ bản của nó bằng cách đào tạo cục bộ ở mỗi bên và tổng hợp các kết quả đào tạo cục bộ tại công cụ tổng hợp. Huấn luyện cục bộ L thường tương ứng với huấn luyện thường xuyên - tập trung của mạng nơ-ron ở bên *Pk* và các tham số wk của nó trong mỗi vòng t.

**

###### 

Tối ưu hóa ở mỗi bên *Pk* giảm thiểu hàm mất mát l cho tham số *wk*, vectơ trọng số của mạng nơron, trên tập dữ liệu huấn luyện của bên *Dk*. Nếu sử dụng thuật toán giảm độ dốc, trong mỗi kỷ nguyên τ của vòng t cho trước, wk được cập nhật như sau:

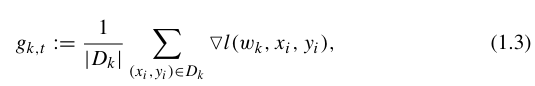
**

Hàm mất mát l được tính toán dựa trên dữ liệu cục bộ *Dk*, bao gồm các mẫu *Xk* và nhãn *Xk*, và có thể là bất kỳ hàm phù hợp nào, chẳng hạn như Lỗi bình phương trung bình (MSE) thường được sử dụng. Các tham số cho vòng này wt,τ k được cập nhật bằng cách sử dụng tốc độ học tập của từng bên cụ thể ηk. Mỗi vòng cục bộ được đưa vào một bản cập nhật mô hình mới từ công cụ tổng hợp, wt,0 k, cung cấp điểm khởi đầu mới cho quá trình đào tạo cục bộ cho mỗi vòng.

Có thể đưa ra các lựa chọn liên quan đến các siêu tham số bên-địa phương khi thiết lập hệ thống học liên hợp hoặc một dự án học liên hợp cụ thể, ví dụ:

* Nên chọn kích thước lô nào cho thuật toán giảm dần độ dốc cục bộ của nhóm: tức là Giảm dần độ dốc ngẫu nhiên (SGD) ban đầu toàn bộ; hoặc một kích thước lô nhỏ phù hợp?
* Có bao nhiêu kỷ nguyên cục bộ được chạy trước khi gửi các bản cập nhật mô hình *rk,t* tới bộ tổng hợp? Tất cả các bên có nên sử dụng cùng một số kỷ nguyên trong mỗi vòng không? Chỉ đào tạo một kỷ nguyên duy nhất trong mỗi bên sẽ ngăn các mô hình cục bộ khác nhau nhiều nhưng gây ra nhiều lưu lượng truy cập mạng hơn và hoạt động tổng hợp thường xuyên hơn. Việc chạy nhiều kỷ nguyên hoặc thậm chí một số kỷ nguyên khác nhau ở các bên khác nhau có thể gây ra sự khác biệt lớn hơn nhưng có thể được sử dụng để thích ứng với sự khác biệt về khả năng tính toán và quy mô của các bên trong tập dữ liệu huấn luyện.
* Chọn tốc độ học *ηk*  nào cho mỗi bên? Sự khác biệt trong phân phối dữ liệu giữa các bên có thể mang lại lợi ích cho tỷ lệ học tập khác nhau.
* Các thuật toán tối ưu hóa khác có thể sử dụng các siêu tham số cục bộ khác nhau như động lượng hoặc tốc độ phân rã [5].

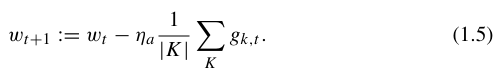
Xem xét trường hợp SGD liên kết đơn giản, trong đó giống như SGD tập trung [2] - mỗi mẫu mới dẫn đến nâng cấp mô hình. Bộ tổng hợp sẽ chọn một bên *Pk*  và gửi truy vấn *qt,k*<*wt*> cho bên được chọn. *Pk* chọn mẫu huấn luyện tiếp theo *(xi, yi)* ∈ *Dk*  và thực hiện huấn luyện cục bộ L, tính toán gradient mất mát l*(wt, xi, yi)* cho mẫu này. Gradient của một bên Pk trong vòng t cụ thể như sau:



Gradient trung bình của các mẫu dữ liệu huấn luyện trong Dk. Bên Pk trả lại nó dưới dạng trả lời rk,t<gt,k> cho bộ tổng hợp. Sau đó, trình tổng hợp sẽ tính toán nội dung truy vấn mới với trọng số mô hình dựa trên phản hồi từ Pk và tốc độ học của trình tổng hợp:



Sau đó, vòng tiếp theo bắt đầu với việc người tổng hợp chọn một bên khác để đóng góp. Theo cách đơn giản này, nó khá kém hiệu quả vì nó gây ra chi phí liên lạc và không tận dụng được việc đào tạo đồng thời. Để làm cho SGD liên kết hiệu quả hơn, có thể huấn luyện theo các đợt nhỏ ở mỗi bên, tăng tính toán ở mỗi bên trong mỗi vòng. Cũng có thể đào tạo đồng thời tại tất cả hoặc một tập hợp con của các bên *Ps* ⊂ *P*, lấy trung bình độ dốc phản hồi của các bên khi tính toán trọng số mô hình mới:



Mặc dù cách này hiệu quả hơn cách tiếp cận đơn giản, nhưng nó vẫn liên quan đến việc giao tiếp nhiều với đơn vị tổng hợp và khả năng phối hợp bị chậm trễ ít nhất một lần trong mỗi kỷ nguyên khi kích thước lô là Dk đầy đủ hoặc nhiều lần khi sử dụng lô nhỏ.

FedAvg hiệu quả hơn bằng cách tận dụng khả năng xử lý độc lập ở mỗi bên [1]. Mỗi bên chạy nhiều kỷ nguyên trước khi trả lời. Thay vì trả lời bằng gradient, các bên có thể tính toán một tập trọng số mới wt,k trực tiếp tại mỗi bên *Pk*, sử dụng tốc độ học tập chung η và trả lời với *rk,t*<*wk,t, nk*>, mô hình của họ và số của các mẫu. Thuật toán tổng hợp F của trình tổng hợp tính trung bình các tham số của mỗi bên, được tính theo số lượng mẫu ở mỗi bên, cho vòng tiếp theo:



Các thử nghiệm cho thấy phương pháp này hoạt động tốt với các loại mô hình khác nhau [2]. FedAvg sử dụng hầu hết các biến được trình bày trong phương trình 2, nhưng có thể tưởng tượng việc giới thiệu các tham số khác như tốc độ học cục bộ hoặc biến đổi cho thuật toán giảm độ dốc.

Các phương pháp tiếp cận khác có thể mở rộng trên các thuật toán đào tạo cục bộ và hợp nhất học liên hợp cơ bản này để thích ứng với các thuộc tính khác nhau của phân phối dữ liệu, lựa chọn khách hàng và yêu cầu về quyền riêng tư. Bài báo trong [6] đề xuất cách tiếp cận học liên hợp dựa trên động lượng để tăng tốc độ hội tụ, lấy cảm hứng từ tối ưu hóa ML tập trung như [5]. Các thuật toán tối ưu hóa trạng thái như ADMM thường chỉ có thể áp dụng được nếu tất cả các bên tham gia cộng tác mọi lúc, duy trì trạng thái của bên đó. Các phương pháp tiếp cận khác nhau, bao gồm [7] và [8], điều chỉnh ADMM cho phù hợp với cài đặt học liên hợp thực tế. FedProx [9] giới thiệu một thuật ngữ chính quy hóa gần nhất để giải quyết tính không đồng nhất về dữ liệu giữa các bên đối với các trường hợp sử dụng không phải IID. Các cách tiếp cận khác như [10] vượt xa các phương pháp giảm độ dốc để tối ưu hóa.

Đối với mỗi cách tiếp cận học liên hợp nhằm giải quyết các khía cạnh cụ thể của tính không đồng nhất dữ liệu, cấu trúc mô hình và các bên, cần xác định một thuật toán bao gồm L, F và giao thức tương tác giữa các bên và bộ tổng hợp, tức là định dạng của *qk* và *rk*.

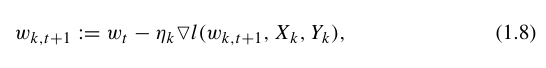
### 2.3.2. Các mô hình học máy truyền thống

Các kỹ thuật học máy cổ điển cũng có thể được áp dụng cho các tình huống học liên hợp. Một số kỹ thuật này có thể được tiếp cận rất giống với DNN. Các kỹ thuật khác phải được xem xét lại hoàn toàn về việc huấn luyện phi tập trung.

Các mô hình tuyến tính, bao gồm hồi quy và phân loại, có thể được huấn luyện trong học liên hợp bằng cách điều chỉnh quy trình huấn luyện tương tự như cách điều chỉnh quy trình huấn luyện mạng thần kinh. Dữ liệu huấn luyện với vectơ đặc trưng *xi* = *(x*1*i , x*2*i ,...,xmi )* có thể được sử dụng để huấn luyện bộ dự đoán cho hồi quy tuyến tính, chẳng hạn như hình dạng:

**

Nó dự đoán *yi* cho m biến tuyến tính *xji* và độ lệch b và kéo theo việc giảm thiểu hàm mất mát cho vectơ trọng số w và b. w thường nhỏ hơn nhiều so với vectơ trọng số của DNN. Với dữ liệu D được phân chia giữa các bên là *Dk*, có thể làm theo các cách tiếp cận được nêu trong phần trước. Huấn luyện ở mỗi bên, giảm thiểu cục bộ hàm mất l*(wk, bk, xi, yi)* đối với dữ liệu huấn luyện cục bộ. Giống như DNN, có thể lựa chọn cách kết hợp các mô hình địa phương thành mô hình toàn cầu. Ví dụ: sử dụng FedAvg làm hàm tổng hợp F, sau đó có thể tính toán cục bộ các trọng số mô hình cục bộ mới như áp dụng tốc độ học tập dành riêng cho từng nhóm *ηk* cho độ dốc của các trọng số.

**

Tất cả các bên gửi trọng số mô hình của họ đến đơn vị tổng hợp nơi các trọng số được tính trung bình và mô hình M mới, được xác định bởi *(wt, bt)*, được phân phối lại cho các bên. Cũng có thể áp dụng các phương pháp tổng hợp khác như SGD liên kết hoặc bất kỳ phương pháp nâng cao nào được thảo luận trong tiểu mục trước. Điều này thường hội tụ nhanh hơn trong trường hợp DNNs do w nhỏ hơn. Các mô hình tuyến tính cổ điển khác như hồi quy logistic hoặc Máy vectơ hỗ trợ tuyến tính (SVM) [11] có thể được chuyển đổi thành phương pháp học liên hợp theo cách tương tự.

Cây quyết định và các mô hình dựa trên cây nâng cao hơn yêu cầu cách tiếp cận học liên hợp khác với các loại mô hình có cấu trúc tham số tĩnh, chẳng hạn như những cấu trúc đã được thảo luận cho đến thời điểm này. Cây quyết định là một loại mô hình phân loại đã được thiết lập và thường được sử dụng cho các vấn đề phân loại. Nó đặc biệt có liên quan trong các lĩnh vực mà khả năng giải thích của quyết định là quan trọng về mặt xã hội như chăm sóc sức khỏe, tài chính và các lĩnh vực khác mà quy định yêu cầu chứng minh các quyết định dựa trên tiêu chí nào. Mặc dù DNN và mô hình tuyến tính có thể được huấn luyện cục bộ và các tham số cục bộ có thể được hợp nhất tại bộ tổng hợp, nhưng không có thuật toán tổng hợp tốt nào được đề xuất để hợp nhất các mô hình cây được huấn luyện độc lập thành một cây quyết định duy nhất.

Tài liệu [3] mô tả cách tiếp cận liên kết cho thuật toán ID3 [12] trong đó việc hình thành cây diễn ra tại bộ tổng hợp và vai trò của các bên là phản hồi bằng số lượng đối với việc phân chia lớp được đề xuất dựa trên dữ liệu đào tạo cục bộ của họ. Nó hoạt động cho dữ liệu số và phân loại. Trong phiên bản gốc, tập trung của nó, cây quyết định ID3 tính toán mức thu được thông tin cho từng tính năng chia tập dữ liệu huấn luyện thành các lớp nhất định. Nó chọn đặc điểm có nhiều thông tin nhất và tính toán các giá trị cho đặc điểm này để phân chia D tốt nhất. Một thuộc tính thường không phân chia D đủ. Đối với mỗi nhánh của cây vừa tạo, áp dụng cách tiếp cận tương tự một cách đệ quy. Tính năng tiếp theo nào sẽ phân chia tập hợp con dữ liệu trong mỗi cây con một cách tốt nhất bằng cách tính toán mức thu được thông tin của từng tập dữ liệu của cây con đối với các tính năng còn lại. Thuật toán tiếp tục tinh chỉnh phân loại đệ quy cho đến khi dừng lại khi tất cả các thành viên của nút cây có cùng nhãn lớp hoặc đạt đến độ sâu tối đa.

Trong phiên bản liên kết, hàm tổng hợp F ở bộ tổng hợp sẽ tính toán mức tăng thông tin và chọn tính năng tiếp theo để phát triển cây. Để có được đầu vào nhằm tính toán mức tăng thông tin, bộ tổng hợp sẽ truy vấn tất cả các bên về các tính năng được đề xuất và các giá trị phân chia. Các bên đếm các thành viên của mỗi cây con được đề xuất và nhãn của chúng là hàm huấn luyện cục bộ F và trả về số lượng này dưới dạng phản hồi cho bộ tổng hợp. Bộ tổng hợp cộng số lượng cho từng tính năng được đề xuất từ tất cả các bên và sau đó tiến hành tính toán mức thu được thông tin trên số lượng tổng hợp này. Giống như trong phiên bản tập trung, tính năng tốt nhất tiếp theo được chọn và cây con lại được phân chia, v.v.

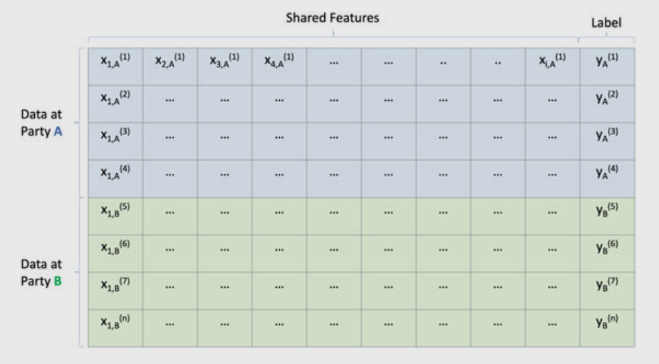
Trong cách tiếp cận này, bộ tổng hợp đóng vai trò nổi bật và thực hiện phần lớn việc tính toán, trong khi các bên chủ yếu cung cấp số lượng liên quan đến các tính năng và giá trị phân chia. Giống như các phương pháp học liên hợp khác, dữ liệu huấn luyện không bao giờ rời khỏi bất kỳ bên nào. Tùy thuộc vào số lượng tập dữ liệu đào tạo và số lượng thành viên của lớp, điều này có thể yêu cầu các biện pháp bảo vệ quyền riêng tư hơn nữa để đảm bảo không tiết lộ quá nhiều thông tin theo cách tiếp cận đơn giản này. Tuy nhiên, đây là một ví dụ điển hình về cách học liên hợp có thể được tiếp cận khác với trường hợp DNN và mô hình tuyến tính.

### 2.3.3 Phân đoạn dữ liệu trong học liên hợp

Cho đến nay, phân phối dữ liệu giữa các bên thường giả định rằng dữ liệu đào tạo của tất cả các bên bao gồm các tính năng giống nhau cho từng mẫu và các bên có dữ liệu liên quan đến các mẫu khác nhau. Ví dụ, bệnh viện A có hồ sơ, hình ảnh sức khỏe của một số bệnh nhân; bệnh viện B có trên các bệnh nhân khác, như minh họa trong Hình 1.3.

Trong trường hợp mạng nơ-ron, giả định rằng mỗi bên có các mẫu có kích thước và nội dung tương đương.

Tuy nhiên, trong một số trường hợp, các bên có thể có những đặc điểm khác nhau khi đề cập đến cùng một thực thể. Sử dụng lại ví dụ về chăm sóc sức khỏe, các bác sĩ chăm sóc sức khỏe ban đầu có thể có hồ sơ sức khỏe điện tử liên quan đến các lần thăm khám của bệnh nhân theo thời gian, trong khi bác sĩ X quang có những hình ảnh liên quan đến bệnh của bệnh nhân. Một bác sĩ phẫu thuật chỉnh hình có thể có hồ sơ phẫu thuật của bệnh nhân. Khi tìm kiếm các yếu tố dự đoán về kết quả sức khỏe cho phẫu thuật chỉnh hình, có thể có ích khi dựa trên dự đoán dựa trên dữ liệu của cả ba bên, bác sĩ chăm sóc ban đầu, bác sĩ X quang và bác sĩ phẫu thuật chỉnh hình. Trong trường hợp này, chỉ có một bên, bác sĩ phẫu thuật chỉnh hình, có thể có kết quả của cuộc phẫu thuật thực sự. Tập dữ liệu này được phân vùng theo chiều dọc.



#### Hình 3. Dữ liệu được phân vùng theo chiều ngang



#### Hình 4. Dữ liệu được phân vùng theo chiều dọc

Hình 1.4 minh họa phân vùng theo chiều dọc, với các tính năng chồng chéo trong khóa nhận dạng để khớp với hồ sơ của cả hai bên, ví dụ: mã định danh của chính phủ. Vì không phải tất cả các tính năng liên quan đều có ở bất kỳ bên nào nên việc học hỏi không thể diễn ra độc lập ở mỗi bên. Hơn nữa, khóa nhận dạng phải được khớp để hiểu các tính năng của mỗi bên bổ sung cho nhau như thế nào. Để bảo vệ quyền riêng tư dữ liệu của mỗi bên, cần một phương pháp mã hóa để khớp dữ liệu và thực hiện quá trình học tập. Hardy và cộng sự. đề xuất một cách tiếp cận ban đầu dựa trên mã hóa đồng hình một phần với những người khác như Xu và cộng sự [13] đề xuất các biến thể hiệu quả hơn nhiều, giảm các yêu cầu liên lạc và tính toán đến mức nó trở nên khả thi trong thực tiễn doanh nghiệp thực tế.

Có liên quan phần nào đến học liên hợp theo chiều dọc, Học phân tách đã được đề xuất bởi Vepakomma et al. [14]. Trong học phân tách, DNN được phân vùng giữa máy khách và máy chủ theo cách máy khách duy trì phần “trên” của DNN thành một lớp phân tách và máy chủ có lớp phân tách và các lớp bên dưới. Ở dạng cơ bản, máy khách có dữ liệu đầu vào và máy chủ có nhãn. Khi sử dụng SGD làm thuật toán huấn luyện, quá trình chuyển tiếp bắt đầu ở máy khách với đầu vào và được truyền đến máy chủ ở lớp phân tách. Việc truyền ngược diễn ra từ máy chủ đến máy khách thông qua lớp phân tách. Với cách tiếp cận này, dữ liệu của một bên cũng có thể được giữ kín, trong khi bên kia có một phần cấu trúc mô hình. Việc học phân tách có thể được thay đổi tùy theo máy khách cũng có nhãn, với lớp cuối cùng, được kết nối đầy đủ ở phía máy khách thông qua lớp phân tách thứ hai hoặc nhiều máy khách có dữ liệu được phân vùng theo chiều dọc và giao tiếp với máy chủ bằng cách sử dụng các phân vùng của lớp phân tách. Trường hợp thứ hai có thể được coi là sự khái quát hóa của việc học liên kết theo chiều dọc.

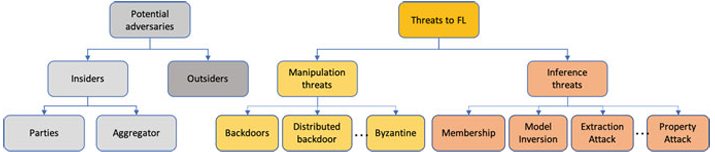
### 2.3.4. Cá biệt hóa mô hình

Cá biệt hóa mô hình đề cập đến việc điều chỉnh mô hình toàn cầu (được đào tạo liên bang) để phân phối dữ liệu của các bên cụ thể tham gia vào quy trình học liên hợp. Mặc dù việc tham gia vào quy trình học liên hợp cho phép tất cả các bên được hưởng lợi từ một lượng lớn dữ liệu đào tạo, nhưng đôi khi việc cá biệt hóa mô hình cuối cùng để đảm bảo nó phản ánh dữ liệu thuộc sở hữu của một bên cụ thể sẽ có ích. Điều này đặc biệt có liên quan nếu các bên tương ứng với người dùng hoặc tổ chức cá nhân. Trong trường hợp đơn giản, các bên riêng lẻ có thể chạy các giai đoạn đào tạo cục bộ bổ sung trên dữ liệu cục bộ cho đến cuối quy trình học liên hợp. Wang và cộng sự. đề xuất cách tiếp cận để đánh giá lợi ích của việc cá nhân hóa cho mỗi bên [15].

Mansour và cộng sự [16] phân tích ba cách tiếp cận khác nhau để cá nhân hóa: phân cụm người dùng, đào tạo về dữ liệu nội suy (giữa toàn cầu và cục bộ) và nội suy mô hình. Cách tiếp cận đầu tiên yêu cầu nới lỏng các yêu cầu về quyền riêng tư hoặc các kỹ thuật bảo mật nâng cao để phân nhóm người dùng dựa trên dữ liệu đào tạo. Nội suy dữ liệu dựa trên việc tạo ra một tập dữ liệu toàn cầu. Mặc dù tất cả các phương pháp đều có tác dụng tuy nhiên phép nội suy mô hình có khả năng ứng dụng rộng rãi nhất từ góc độ quyền riêng tư. Grimberg và cộng sự đề xuất phương pháp tối ưu hóa tính trung bình của mô hình toàn cầu và mô hình cục bộ cho mục đích cá biệt hóa bằng cách xác định trọng số tối ưu hóa, mở rộng các phương pháp đã thảo luận trước đây [17].

Mặc dù các phương pháp tiếp cận cá biệt hóa vẫn đang phát triển nhưng đây là phần bổ sung quan trọng cho quy trình học liên hợp.

Bằng cách để lại dữ liệu ở nguyên vị trí, học liên hợp cung cấp mức độ riêng tư vốn có ngay từ đầu. Tuy nhiên, vẫn có nguy cơ xâm phạm quyền riêng tư của dữ liệu. Điều quan trọng là phải hiểu các mô hình mối đe dọa khác nhau có thể phát sinh trong quá trình áp dụng học liên hợp để đảm bảo rằng các rủi ro liên quan được giảm thiểu một cách thích hợp bằng các biện pháp phòng vệ phù hợp.



#### Hình 5 Đặc điểm các mối đe dọa tiềm tàng của học liên hợp cũng như các đối thủ tiềm ẩn.

Cần phân tích rủi ro trước tiên bằng cách tìm hiểu các bề mặt tấn công tiềm ẩn mà đối thủ có thể khai thác. Hệ thống học liên hợp được thiết lập tốt sẽ sử dụng các kênh an toàn và được xác thực để đảm bảo tất cả các tin nhắn trao đổi giữa các bên và đơn vị tổng hợp không thể bị các thực thể khác chặn đồng thời ngăn chặn hành vi mạo danh. Do đó, có thể giả định rằng đơn vị tổng hợp và các bên là hai thực thể duy nhất có thể truy cập các thông điệp được trao đổi giữa họ và các tạo phẩm được tạo ra trong quá trình đào tạo. Với suy nghĩ này, có thể phân loại các đối thủ tiềm năng là người trong cuộc hay người ngoài cuộc. Đối thủ nội bộ là những thực thể tham gia vào quá trình đào tạo có quyền truy cập vào các tạo phẩm được tạo ra trong quá trình đào tạo và các thông điệp hướng tới họ. Tất cả các đối thủ tiềm năng khác đều được coi là người ngoài cuộc. Trong phân loại này, các thực thể nhận được mô hình cuối cùng do quá trình đào tạo học liên hợp tạo ra được coi là người ngoài.

Có thể phân loại các mối đe dọa đối với học liên hợp thành các mối đe dọa thao túng và suy luận, trong đó các mối đe dọa thao túng là những mối đe dọa mà mục tiêu của người trong cuộc là tác động đến mô hình để có lợi cho họ bằng cách thao túng bất kỳ tạo phẩm nào mà cô ấy có thể truy cập trong quá trình đào tạo, trong khi các mối đe dọa suy luận là những mối đe dọa mà một người trong cuộc hoặc người ngoài cố gắng trích xuất thông tin cá nhân về dữ liệu đào tạo.

## 2.4. Các phương pháp học liên hợp liên quan đến khóa luận

### 2.4.1. FedAvg

Federated Averaging là một thuật toán học liên hợp được đề xuất bởi McMahan et al; cho phép một lượng lớn khách hàng có thể thực hiện việc đào tạo mô hình mà không cần phải chia sẻ thông tin trên máy chủ nội bộ. FedAvg chạy Stochastic Gradient Descent (SGD) song song giữa các thiết bị và tính trung bình các giá trị tham số sau một khoảng thời gian nhất định. Sau khi cập nhật mô hình trên máy chủ tập trung, mô hình mới này sẽ được gửi về cho các bên khách hàng để tiếp tục thực hiện quá trình học. Quá trình này được lặp lại nhiều lần cho đến khi đạt được hội tụ mục tiêu.

Tuy nhiên phương pháp này yêu cầu máy chủ tập trung phải có một tập dữ liệu đánh giá mô hình học máy. Điều này có thể làm hạn chế tiềm năng của học liên hợp trong trường hợp mà chỉ có một phần của dữ liệu được dùng để kiểm tra bởi máy chủ tập trung. Khi một bên khách hàng bị tấn công, hệ thống của họ sẽ chứa những dữ liệu mạng quan trọng nhưng không thể chia sẻ với máy chủ tập trung để đánh giá mô hình. Điều này dẫn đến việc mô hình trung tâm không thể được kiểm tra với những dữ liệu tấn công mới nhất.

Vì vậy, có 2 điều mà FedAvg không đáp ứng được cho lĩnh vực an toàn thông tin:

* Tốc độ: Thời gian hội tụ ngắn để mô hình máy chủ được phân phối cho các bên khách hàng với mục đích chống lại những đợt tấn công mới nhất. FedAvg khiến tất cả khách hàng cùng thực hiện việc học cho mô hình với số bước giống nhau, bỏ qua độ chuẩn xác của mô hình máy chủ trên tập dữ liệu kiểm tra của khách. Điều này làm cho thời gian huấn luyện mô hình kéo dài một cách đáng kể mà không đem lại sự đóng góp cho việc tăng độ chuẩn xác cho mô hình máy chủ.
* Bảo mật: FedAvg được hoạt động dựa trên việc máy chủ tập trung có thể truy cập tới dữ liệu kiểm tra để xác nhận rằng mô hình máy chủ đã đạt được độ chuẩn xác nhất định và kết thúc việc huấn luyện ở phía khách. Điều này hiếm khi có thể xảy ra trong lĩnh vực an toàn thông tin. Trong trường hợp máy chủ khách hàng bị tấn công bởi dữ liệu có chứa mã khai thác 0-day mà không thể chia sẻ trong lúc huấn luyện, chỉ có một cách để máy chủ tập trung có thể biết rằng mô hình máy chủ có thể phát hiện cuộc tấn công trên là cho mô hình huấn luyện với dữ liệu của bên khách hàng nạn nhân.

### 2.4.2. Học liên hợp thích nghi - FLAD

FLAD (Federated Learning Approach to DDoS) là phương pháp học liên hợp mà máy chủ trung tâm đánh giá độ chuẩn xác của mô hình trung tâm mà không cần trao đổi tập dữ liệu học hoặc đánh giá giữa máy chủ và khách hàng. FLAD đáp ứng được nhu cầu của tình hình an toàn thông tin hiện nay khi mà không một dữ liệu tấn công nào ở bất kỳ thời điểm nào được chia sẻ giữa khách hàng và máy chủ. Đồng thời áp dụng một phương án dừng để tăng tốc cho quá trình học của các bên khách [21].

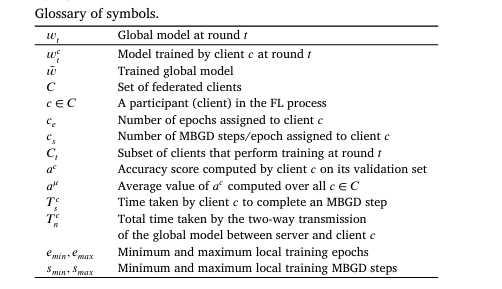
Ý tưởng của phương pháp này là việc chọn lọc và chỉ thực hiện quá trình học cho những bên khách có kết quả kiểm tra kém với mô hình trung tâm hiện tại. Với những bên được chọn, lượng tính toán (số bước) phải trải qua phụ thuộc vào độ chuẩn xác [21]. Lưu ý là độ chuẩn xác được tính dựa trên tập dữ liệu của chính khách hàng. Điều này đảm bảo được không có dữ liệu mạng nhạy cảm nào được trao đổi giữa khách hàng và máy chủ, kể cả cho việc kiểm tra.

Phương pháp này giả định một tình huống khi mà những khách hàng tới từ nhiều bên khác nhau và một thực thể đóng vai trò như một máy chủ tập trung. Máy chủ này phụ trách điều phối hoạt động học liên hợp. Thêm vào đó, không một khách hàng nào muốn hoặc có quyền chia sẻ bất kì thông tin hay dữ liệu với các bên khác. Mặt khác, mục tiêu của nhóm là để nâng cao khả năng phát hiện DDoS của mỗi IDS phía bên khách với những dữ liệu riêng biệt của họ.

Với tình huống trên, bên khách hàng có thể bị tấn công bởi lỗ hổng 0-day DDoS bất kỳ lúc nào. Để đảm bảo khả năng bảo mật tốt nhất, hệ thống cần phải có một mô hình được cập nhật liên tục với những dữ liệu tấn công mới nhất và phân phối cho tất cả các bên khách. Nhưng điều quan trọng là máy chủ trung tâm đôi khi không thể sử dụng dữ liệu mạng liên quan tới những kĩ thuật tấn công mới đang được phát triển. Điều này là một thách thức lớn cho việc phân loại những loại tấn công này của mô hình trung tâm.

Tính huống này cũng cho rằng cả máy chủ và các bên khách đều trong sạch và không cố gắng đầu độc mô hình trung tâm bằng những dữ liệu sai lệch. Tấn công đầu độc là một vấn đề nghiêm trọng đối với những IDS phụ thuộc vào việc hợp tác huấn luyện cho mô hình. Bên khách hàng độc hại có thể ảnh hưởng tới quá trình huấn luyện bằng cách tạo ra những bộ dữ liệu giả mạo dẫn đến việc mô hình trung tâm không thể phân loại chuẩn xác một số loại tấn công. Vấn đề này ảnh hưởng tới hầu hết những ứng dụng của học máy đối với mảng an toàn thông tin. Tuy nhiên, phạm vi đề cập của FLAD không bao gồm vấn đề này.

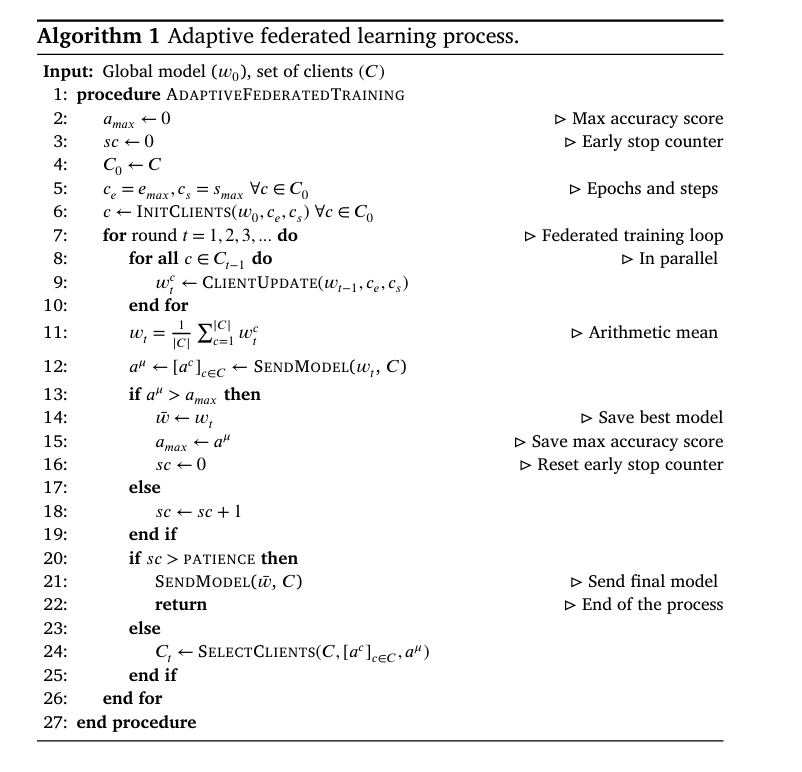
Khi huấn luyện một mô hình học máy, trọng số của mạng lưới có thể xem là một điểm ở trên không gian nhiều chiều, với mỗi chiều tương ứng với một trọng số riêng biệt. Mục tiêu của quá trình huấn luyện là để tìm một điểm trên không gian có thể tối ưu độ chuẩn xác của mô hình toàn cục trên tập dữ liệu cục bộ. Tính thích nghi của mô hình FLAD được thể hiện qua hai biến số epoch (ce) và steps (cs) [21]. Hai biến trên khác nhau ở mỗi bên mô hình khách, đồng thời được thay đổi dựa trên khả năng dự đoán của mô hình. Việc xác định hai biến số trên phụ thuộc vào độ lớn của khoảng cách giữa độ chuẩn xác của mô hình toàn cục trên tập dữ liệu cục bộ và độ chuẩn xác lớn nhất có thể đạt được. Nhờ vào đó, bên khách hàng có khoảng cách lớn sẽ phải thực hiện nhiều bước huấn luyện để tiến gần tới điểm hội tụ. Tương tự, những bên khách hàng có khoảng cách nhỏ sẽ phải thực hiện ít bước hơn, và thậm chí là không cần huấn luyện.



##### Bảng 1. Các tham số trong mô hình FLAD

Mô hình FLAD hoạt động dựa trên ba giai đoạn chính: xử lý dữ liệu, lựa chọn bên khách cần huấn luyện và huấn luyện, đánh giá. Trong đó, hai giai đoạn đầu được xử lý bởi máy chủ tập trung, bên khách hàng sẽ đảm nhiệm việc huấn luyện cùng với đánh giá. Mô hình FedAvg không có giai đoạn lựa chọn bên khách hàng cần huấn luyện.

Giai đoạn xử lý dữ liệu có cấu trúc tương đồng với FedAvg , tuy nhiên không có sự xuất hiện của quá trình đánh giá trên máy chủ.

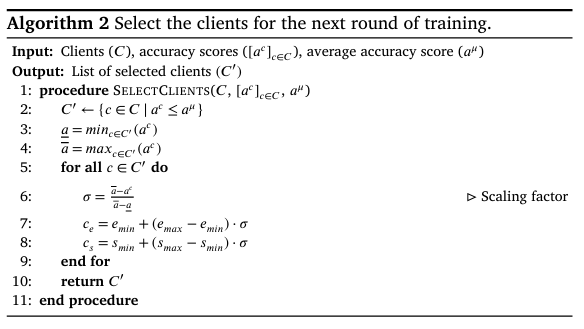


###### Thuật toán 1. Học liên hợp thích nghi

Thuật toán 1 mô tả quá trình học liên hợp chính được thực hiện bởi máy chủ, điều phối hoạt động của các bên khách. Thuật toán nhận vào một mô hình toàn cục (w0) và một tập các khách hàng tham gia vào quá trình huấn luyện. Quá trình này được chạy cho đến khi hội tụ, được điều khiển bởi biến PATIENCE - số vòng quá trình được chạy trước khi kết thúc mà không tiến triển. Ở dòng số 5, lượng tính toán cho mỗi bên khách được khởi tạo bằng với số epoch và step tối đa. Vòng lặp ở dòng 8-10 gọi hàm ClientUpdate (quá trình huấn luyện) cho một tệp khách hàng đã được chọn.

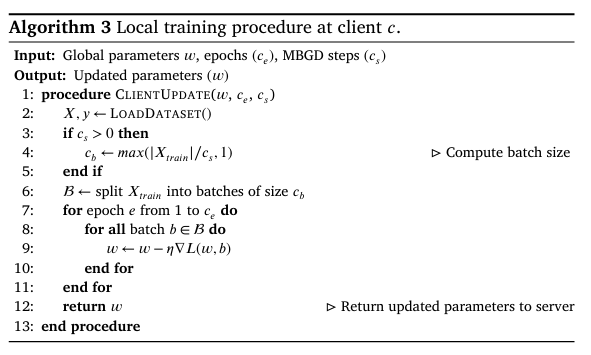
Ở mỗi vòng, máy chủ trung tâm tính toán trung bình toàn bộ các tham số của tất cả các bên khách, kể cả những bên không tham gia huấn luyện trong vòng đấy (dòng 11). Mô hình toàn cục được cập nhật và gửi tới các bên khách hàng, sau đó độ chuẩn xác (ac) được gửi lại máy chủ. Máy chủ sau đó tính trung bình các kết quả này để đánh giá độ hiệu quả quá trình học liên hợp (dòng 13-19). Nếu giá trị trung bình mới lớn hơn giá trị cũ, mô hình toàn cục sẽ được lưu lại và biến sc được đặt trở về giá trị 0. Nếu không, biến sc sẽ được tăng lên 1 đơn vị cho thấy không có sự tiến triển. Khi sc > PATIENCE, quá trình học liên hợp kết thúc và mô hình toàn cục tốt nhất được gửi về các bên khách (dòng 21).

Giai đoạn lựa chọn khách hàng xác định những bên sẽ tham gia vào quá trình huấn luyện ở vòng tiếp theo và gán epochs, steps tương ứng cho từng bên.



###### Thuật toán 2. Chọn khách hàng cần huấn luyện

Quá trình lựa chọn bắt đầu với việc chọn một danh sách các bên khách hàng C’ tham gia vào vòng huấn luyện sau. C’ là một tập con của C bao gồm những bên có độ chuẩn xác trên tập dữ liệu cục bộ thấp hơn một giá trị trung bình (dòng 2). Số epoch và steps được gán cho mỗi bên phụ thuộc vào giá trị ac . Ý tưởng là giá trị ac càng cao, khối lượng tính toán cần thiết càng thấp. Điều này được thực hiện ở dòng 5-9, nơi mà mỗi bên được gán cho khối lượng thấp nhất cộng thêm một khối lượng có giá trị phụ thuộc vào ac. Quá trình này trả về một danh sách các bên cần tham gia huấn luyện cùng với khối lượng tính toán tương ứng cho từng bên.



###### Thuật toán 3. Huấn luyện ở phía khách hàng

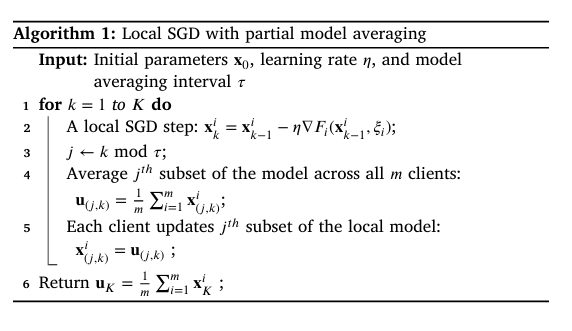
Thuật toán 3 thể hiện quá trình huấn luyện được thực hiện trên hệ thống khách. Quá trình này bắt đầu từ việc nhận các giá trị về trọng số w từ mô hình toàn cục nhận được từ máy chủ trung tâm. Sau đó thực hiện huấn luyện với khối lượng được định sẵn bởi máy chủ trung tâm (dòng 7-11). Lượng mẫu dữ liệu trong một bước huấn luyện cb được xác định dựa trên cs ở dòng 4.

### 2.4.3. Trung bình phân đoạn - Partial Average

SGD và FedAvg là hai thuật toán tạo nên nền móng cho mô hình học liên hợp. Thuật toán trên chạy SGD ở nhiều bên khách hàng khác nhau rồi định kỳ tính một giá trị trung bình giữa các bên khách. Điều đó tạo ra một sự khác biệt lớn giữa các mô hình của các bên khách hàng, đồng thời khiến cho quá trình huấn luyện hội tụ tốn nhiều thời gian. Partial Average hay trung bình phân đoạn là một thuật toán tính trung bình có thể làm giảm thiểu vấn đề sự khác biệt của mô hình trong học liên hợp. Thuật toán trung bình phân đoạn giúp cho các mô hình của bên khách gần nhau hơn trong không gian tham số. Thuật toán tập trung vào khả năng phân loại khi thay thế FedAvg trong học liên hợp.

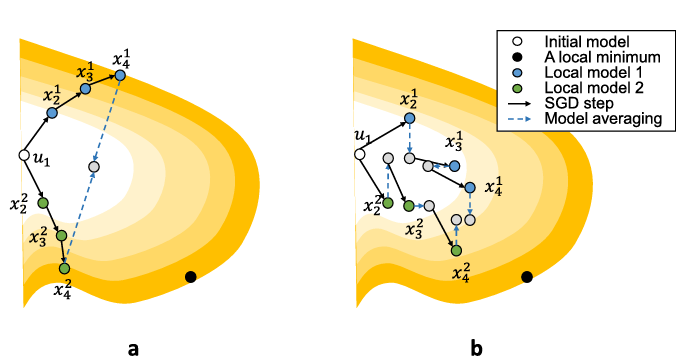
Đối với thuật toán FedAvg, các bên khách sẽ tham gia huấn luyện theo vòng. Đồng thời, sau một số vòng nhất định, dữ liệu về mô hình bên phía khách sẽ được gửi về phía máy chủ trung tâm để xử lý và thực hiện tính trung bình tham số của các bên. Sau đó mô hình này sẽ được gửi lại về phía bên khách để tiếp tục quá trình huấn luyện. Thuật toán Partial Average thay vì sử dụng toàn bộ các tham số của mô hình cho việc tính trung bình thì chỉ sử dụng một phần của tập tham số ở mỗi vòng[20]. Điều này làm tăng tốc quá trình hội tụ của mô hình. Với một giới hạn về số vòng, tốc độ hội tụ nhanh khả năng cao sẽ dẫn tới độ chuẩn xác cao cho mô hình.

Ở thuật toán trung bình toàn phần, sau một vài vòng định sẵn - được biểu diễn bởi kí hiệu ꞇ thể hiện tốc độ cập nhật, toàn bộ tham số của các bên được tính trung bình và cập nhật vào mô hình toàn cục. Thuật toán trung bình phân đoạn dựa vào giá trị ꞇ trên để chia tập tham số thành các phần bằng nhau [20].



###### Thuật toán 4. Trung bình phân đoạn

Ở mỗi vòng, các bên khách sẽ thực hiện việc huấn luyện (dòng 2). Một tập các tham số *j* được xác định dựa trên kết quả phép chia lấy dư của thứ tự vòng hiện tại *k* và tốc độ cập nhật ꞇ(dòng 3). Điều này đảm bảo tất cả các tham số đều được tham gia cập nhật ít nhất một lần trong toàn bộ quá trình học. Sau đó, chương trình tính trung bình dựa trên tệp tham số được chọn (dòng 4) và cập nhật vào mô hình (dòng 5). Cuối cùng, sau khi kết thúc K vòng, thuật toán thực hiện tính trung bình và cập nhật mô hình toàn cục (dòng 6).



#### Hình 6. Mô hình trung bình toàn phần và phân đoạn trên không gian tham số

Hình 1.6 [20] biểu diễn sự khác biệt giữa hai thuật toán trung bình toàn phần (a) và trung bình phân đoạn (b). Nó thể hiện hướng đi của hai mô hình cục bộ sau một vòng cập nhật (t = 3). Trung bình phân đoạn liên tục đồng bộ một phần của các mô hình cục bộ so với việc cập nhật không bị lệ thuộc của trung bình toàn phần. Điều này dẫn đến các mô hình cục bộ tiến gần đến nhau trên không gian tham số hơn so với trung bình toàn phần.

# Chương 3: Đề xuất áp dụng thuật toán trung bình phân đoạn vào mô hình học liên hợp thích nghi

## 3.1. Phân tích hạn chế của mô hình học liên hợp thích nghi FLAD và khả năng áp dụng thuật toán trung bình phân đoạn

### 3.1.1 Hạn chế của học liên hợp thích nghi FLAD

FLAD là một phương pháp liên hợp có khả năng đẩy mạnh quá trình học của một hệ thống liên hợp có nhiều khách hàng. Tuy nhiên, việc vẫn sử dụng thuật toán trung bình FedAvg có nhiều hạn chế trong việc phân loại các kiểu dữ liệu tấn công DDoS. Sau một số vòng nhất định, các mô hình khách có sự khác nhau lớn và nằm cách xa nhau trên không gian tham số. Việc tính trung bình toàn bộ các tham số này ảnh hưởng lớn tới tốc độ hội tụ của mô hình. Nếu số lượng khách hàng lớn lên đến hàng nghìn hoặc hàng triệu, vấn đề này trở nên rất nghiêm trọng.

FLAD được thiết kế cho bài toán phát hiện dữ liệu mạng độc hại liên quan đến kiểu tấn công DDoS. Trên thực tế, ngoài việc đánh giá một tệp dữ liệu có phải DDoS hay không, phân loại kiểu tấn công cũng là một nhu cầu thiết yếu cho công việc phân tích dữ liệu để đưa ra những phương án đối phó kịp thời.

### 3.1.2 Áp dụng thuật toán trung bình phân đoạn

Đối với bài toán áp dụng thuật toán trung bình phân đoạn cho FedAvg, sau mỗi vòng huấn luyện một phần các tham số sẽ được đưa vào tính trung bình và cập nhật mô hình cục bộ. Khi được áp dụng thuật toán trung bình phân đoạn, thời điểm cập nhật được kéo ngắn lại đồng thời chỉ một phần tham số được tham gia cập nhật.

Không có một nguyên tắc cụ thể về cách chia lượng tham số cập nhật hoặc thời điểm cập nhật mới, chỉ cần đáp ứng được tiêu chí toàn bộ tham số được tham gia cập nhật ít nhất một lần sau một khoảng số vòng cập nhật. Khi thỏa mãn tiêu chí trên, hiệu năng của thuật toán được đảm bảo về độ chuẩn xác [20].

Vì cần thay đổi mục đích từ phát hiện thành phân loại, mô hình học máy của FLAD cần phải thay đổi về dữ liệu đầu ra. Thay vì phát hiện có (là dữ liệu độc hại) hoặc không (dữ liệu sạch), mô hình cần phải phân loại và gán nhãn cho các loại dữ liệu độc hại khác nhau.

## 3.2. Phương pháp áp dụng

### 3.2.3 Thay thế FedAvg trong FLAD bằng thuật toán trung bình phân đoạn

Đối với thuật toán trung bình phân đoạn, cần xác định rõ thời điểm các bên khách thực hiện việc cập nhật mô hình toàn cục để tiến hành phân đoạn. Dựa vào thuật toán 1, dòng 11 thể hiện việc tính trung bình các trọng số giữa các bên khách hàng. Dòng này được gọi một lần mỗi vòng sau khi các bên đã thực hiện xong việc huấn luyện, cho thấy giá trị của t = 1. Đối với mỗi vòng, những bên tham gia huấn luyện sẽ thực hiện ce lần, mỗi lần cs bước huấn luyện. Vì vậy, có thể phân đoạn giai đoạn huấn luyện dựa trên giá trị của ce .

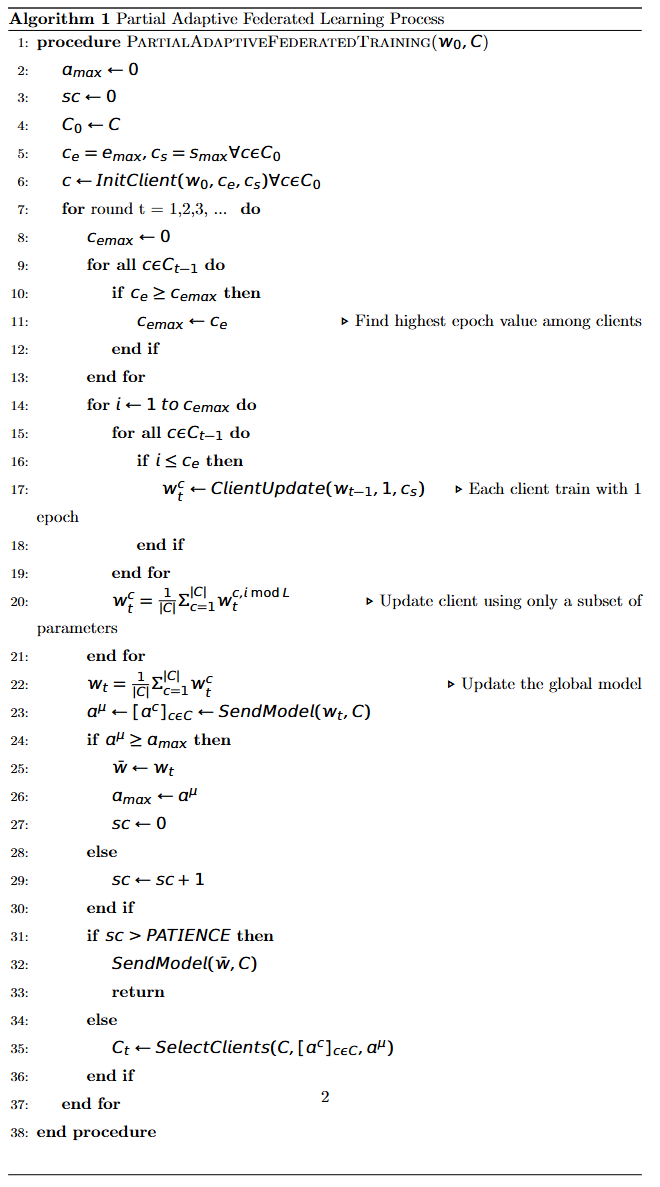
Một phương án để chia tệp tham số thành các tệp con cho mỗi lần cập nhật có thể sử dụng là chia theo lớp tham số. Ở mô hình FLAD, tổng cộng có 4 lớp: 1 lớp đầu vào, 2 lớp ẩn, 1 lớp đầu ra. Nhưng trong đó chỉ có 3 lớp cuối có trọng số. Vì vậy, trong trường hợp này có thể chia tập tham số thành 3 phần.

| **Lớp** | **Đầu ra** | **Tham số** |
| --- | --- | --- |
| flatten (Flatten) | (None, 110) | 0 |
| fc0 (Dense) | (None, 32) | 3552 |
| fc1 (Dense) | (None, 32) | 1056 |
| fc3 (Dense) | (None, 10) | 330 |

##### Bảng 2. Cấu trúc mô hình

So với cấu trúc mô hình của bài toán FLAD được công bố trong bài báo, mô hình trên có đầu ra với 10 nơ-ron thể hiện các giá trị nhãn từ 0 đến 9, đại diện cho các loại tấn công khác nhau.

Ý tưởng chung của quá trình này là sau mỗi epoch, các bên khách sẽ sử dụng 1/L số tham số để cập nhật mô hình cục bộ, với L là số lớp có trọng số.



Thuật toán 4: Trung bình phân đoạn áp dụng trong học liên hợp thích nghi

Quá trình học liên hợp thích nghi phân đoạn khởi tạo các tham số giống như mô hình gốc. Tuy nhiên, trước khi bắt đầu quá trình huấn luyện, một giá trị cemax được xác định nhằm phân nhỏ quá trình học của các bên khách (dòng 11). Như vậy, mỗi bên khách sẽ thực hiện quá trình huấn luyện với khối lượng bằng *1 x cs* (dòng 17). Biến đếm *i* giúp đảm bảo cho việc các bên khách được thực hiện đúng với khối lượng *ce  x cs* ban đầu được giao. Đồng thời, bằng việc sử dụng biến đếm *i* chia lấy dư cho số lượng lớp của mô hình *L* có thể xác định được tệp tham số cần tham gia vào quá trình cập nhật. Ở dòng 20, chỉ một bộ phận các tham số được sử dụng để cập nhật mô hình cục bộ, so với toàn bộ tham số ở thuật toán gốc.

# Chương 4: Thực nghiệm và đánh giá kết quả

## 4.1. Tập dữ liệu

Đề xuất được kiểm tra với tập dữ liệu tấn công DDoS mới nhất, CIC-DDoS2019 [18] , được cung cấp bởi Cơ quan an toàn thông tin Canada thuộc trường đại học New Brunswick. CIC-DDoS2019 bao gồm dữ liệu mạng của một vài ngày, đồng thời chứa 13 loại tấn công DDoS khác nhau. Dữ liệu trên được công bố dưới dạng các gói dữ liệu mạng thu được, đồng thời bao gồm một tệp văn bản chứa thông tin của từng gói dữ liệu. Dữ liệu tấn công được tạo ra bởi công cụ của bên thứ ba và có thể được chia thành 2 loại: tấn công phản hồi và tấn công khai thác. Loại đầu tiên bao gồm những kiểu tấn công dựa trên giao thức UDP, kẻ tấn công yêu cầu những phản hồi từ máy chủ tới IP của nạn nhân. Điều này khiến băng thông của nạn nhân bị tràn ngập bởi phản hồi từ máy chủ. Loại thứ hai liên quan đến những kiểu tấn công khai thác vào những điểm yếu của một vài loại giao thức mạng (bắt tay 3 chiều của TCP).

Ở bảng 1 thống kê về các kiểu tấn công được cung cấp bởi CIC-DDoS2019. Cột *Luồng* thể hiện số phiên giao tiếp TCP hoặc dòng UDP hai chiều. Mỗi luồng được định danh bằng một dãy gồm 5 thành phần: IP nguồn, cổng nguồn, IP đích, cổng đích, giao thức.

| **Kiểu tấn công** | **Luồng** | **Giao thức** | **Thông tin** |
| --- | --- | --- | --- |
| DNS | 441931 | UDP | Tấn công DDoS khai thác một hệ thống sử dụng giao thức UDP để khủng bố nạn nhân bằng rất nhiều những truy vấn được gửi bởi kẻ tấn công đến máy chủ xác định bởi IP thu được của nạn nhân |
| LDAP | 11499 |
| MSSQL | 9559537 |
| NTP | 1194836 |
| NetBIOS | 7553086 |
| Portmap | 186449 |
| SNMP | 1334534 | UDP | Tấn công khuếch đại sử dụng giao thức quản lý mạng đơn giản SNMP để cấu hình thiết bị mạng |
| SSDP | 2580154 | UDP | Tấn công dựa trên giao thức tìm kiếm dịch vụ đơn giản SSDP. Giao thức trên cho phép thiết bị UPnP gửi và nhận thông tin qua UDP. Thiết bị dính lỗ hổng gửi các gói tin hồi đáp UPnP đến nạn nhân |
| TFTP | 6503575 | UDP | Tấn công được thực hiện bằng cách gửi yêu cầu tệp đến máy chủ giao thức truyền tệp đơn giản ở phía nạn nhân |
| Syn Flood | 6056402 | TCP | Tấn công khai thác hoạt động bắt tay ba lần của TCP để khủng bố nạn nhân bằng rất nhiều gói tin SYN |
| UDP Flood | 6056402 | UDP | Tấn công sử dụng nhiều gói tin UDP để làm tắc nghẽn tài nguyên mạng của nạn nhân |
| UDPLag | 474018 | UDP | Dữ liệu UDP được tạo ra để làm chậm băng thông tới máy chủ trò chơi của nạn nhân |
| WebDDoS | 146 | TCP | Kiểu tấn công bao gồm khoảng 3100 gói tin tới một máy chủ web ở cổng 80 |
| Tổng | 42865789 |  | Mặc dù có rất nhiều luồng, tệp dữ liệu rất không cân bằng. Có 8 kiểu tấn công chứa nhiều luồng hơn những loại khác với hơn một triệu luồng mỗi loại. Trong khi đó WebDDoS chỉ có 146 luồng. |

##### Bảng 3. Thống kê các kiểu tấn công trong tệp dữ liệu CIC-DDoS2019

Dữ liệu của CIC-DDoS2019 có thể được xử lý bởi công cụ LUCID [19]. Dữ liệu sau khi được xử lý được thể hiện bằng một mảng 2 chiều với *n = 10* dòng và *f = 11* cột . Mỗi dòng đại diện cho một gói dữ liệu với 11 thông tin: Thời gian, độ dài gói tin, giao thức cao nhất, cờ IP, giao thức, độ dài TCP, TCP Ack, cờ TCP, kích cỡ cửa sổ TCP, độ dài UDP và kiểu ICMP. Nếu số gói tin thuộc một luồng nhỏ hơn *n,* mảng được thêm các giá trị *0.* Công cụ LUCID chia tách các luồng dữ liệu thành các gói nhỏ hơn để phù hợp với thực tế, khi mà thuật toán cần phải xử lý những mảnh của luồng nằm rải rác theo thời gian.

Đối với thực nghiệm của đề xuất này, dữ liệu của 9 loại tấn công được sử dụng bao gồm NTP, DNS, LDAP, MSSQL, NetBIOS, SNMP, SSDP, UDP và TFTP. Mô hình tổng quát của hệ thống bao gồm 1 máy chủ tập trung và 2 khách thực hiện huấn luyện mô hình. Dữ liệu của cả hai bên khách đều chứa đầy đủ 9 loại tấn công được sử dụng, được tách thành 3 loại huấn luyện (Train), đánh giá (Validation) và kiểm tra (Test). LUCID phân bổ kích thước của dữ liệu huấn luyện và đánh giá theo tỉ lệ 90 / 10.

Bảng dưới đây thống kê kích thước của các tệp dữ liệu bên khách theo đơn vị gói tin (Packet)

|  | **Huấn luyện** | **Đánh giá** | **Kiểm tra** |
| --- | --- | --- | --- |
| **Khách 1** | 2612542 | 290165 | 322481 |
| **Khách 2** | 2588059 | 287401 | 319504 |

##### Bảng 4: Kích thước bộ dữ liệu đầu vào

## 

## 

## 

## 

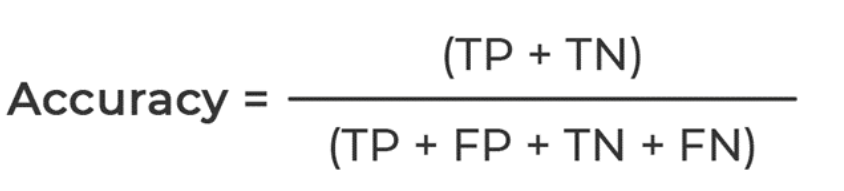
## 

## 

## 4.2 Thực nghiệm

Phần thực nghiệm tập trung vào việc tính toán độ chuẩn xác của thuật toán trung bình phân đoạn khi áp dụng vào mô hình FLAD so với mô hình FLAD gốc để phân loại kiểu tấn công DDoS. Với mục đích trên, dữ liệu CIC-DDoS2019 đã qua xử lý được sử dụng cho việc đánh giá.

Hiệu năng của mô hình được đánh giá bằng độ chuẩn xác của mô hình khi dự đoán trên tập dữ liệu kiểm tra sau khi đã được huấn luyện trên tập dữ liệu huấn luyện. Chỉ số này tương ứng với phần trăm số kết quả dự đoán đúng (True positive + True negative) với kết quả thực tế trên tổng số mẫu dự đoán (True positive + True negative + False positive + False negative).



Các tham số huấn luyện của quá trình như sau:

| **Tên** | **Giá trị** | **Thông tin** |
| --- | --- | --- |
| PATIENCE | 25 | Số vòng tối đa không có tiến triển |
| Min epochs | 1 | Số epoch nhỏ nhất |
| Max epochs | 12 | Số epoch lớn nhất |
| Min steps | 10 | Số step nhỏ nhất |
| Max steps | 1000 | Số step lớn nhất |
| *n x f* | 10 x 11 | Kích cỡ của lớp đầu vào MLP |
| l | 2 | Số lớp ẩn |
| m | 32 | Số nơ-ron trên một lớp |

##### Bảng 5: Tham số huấn luyện

Môi trường thực nghiệm là máy chủ chạy hệ điều hành Ubuntu 20.04, CPU Intel Broadwell x86/64 8 nhân, RAM 64GB. Môi trường phần mềm được cài đặt dựa theo yêu cầu của FLAD [21].

Kết quả thu được từ hai mô hình FLAD gốc và FLAD kết hợp với trung bình phân đoạn:

| **Phương pháp** | **Độ chuẩn xác (Accuracy)** |
| --- | --- |
| FLAD | 0.60134 |
| FLAD + PartialAvg | 0.76903 |

##### Bảng 6. Kết quả thực nghiệm

## 4.3 Đánh giá kết quả

Với bài toán học liên hợp thích nghi phân loại, kết quả thực nghiệm cho thấy khi áp dụng công thức trung bình phân đoạn cho kết quả về độ chuẩn xác tướng ứng với 0.769 tốt hơn so với mô hình gốc là 0.601.

Kết quả này là hoàn toàn hợp lý do Partial Average được xây dựng cho bài toán phân loại còn với FLAD là phát hiện. Khi biến đổi FLAD để giải quyết vấn đề phân loại, FLAD cho thấy những mặt hạn chế của mình. Partial Average là một sự lựa chọn tốt để bổ sung cho những mặt ưu điểm cũng như hạn chế những khuyết điểm của FLAD khi phải đối đầu với vấn đề phân loại.

|  |  | |  |
| --- | --- | --- | --- |
|  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Chương 5: Kết luận

Trong đề tài này, chúng tôi đã thực hiện thử nghiệm biện pháp áp dụng trung bình phân đoạn để áp dụng cho mô hình FLAD đối với bài toán phân loại. Với mục tiêu đề xuất và so sánh độ chuẩn xác của các mô hình, khóa luận đã rút ra kết luận là kết quả từ phương pháp trung bình phân đoạn được đề xuất là có độ chuẩn xác tốt hơn so với mô hình gốc với kết quả lần lượt là 0.769 và 0.601.

Từ kết quả đạt được, chúng tôi nhận thấy có những điểm có thể nghiên cứu thêm để phát triển đề tài như: tối ưu tốc độ học và độ chuẩn xác dựa vào hay đổi cấu trúc mô hình cùng các tham số, thử nghiệm thêm một số thuật toán trung bình khác để nâng cao độ chuẩn xác.

Thông qua khóa luận này, chúng tôi đã trau dồi được thêm kiến thức về các mạng học sâu và về lĩnh vực thông tin về các kiểu tấn công DDoS. Chúng tôi mong rằng nội dung của khóa luận có thể đóng góp vào sự phát triển của lĩnh vực.

# TÀI LIỆU THAM KHẢO

[1] Koneˇ cn` y J, McMahan HB, Yu FX, Richtárik P, Suresh AT, Bacon D (2016) Federated learning: strategies for improving communication efficiency. arXiv preprint arXiv:161005492

[2] McMahan B, Moore E, Ramage D, Hampson S, y Arcas BA (2017) Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics. PMLR, pp 1273–1282

[3] Ludwig H, Baracaldo N, Thomas G, Zhou Y, Anwar A, Rajamoni S, Ong Y, Radhakrishnan J, Verma A, Sinn M et al (2020) IBM federated learning: an enterprise framework white paper v0. 1. arXiv preprint arXiv:200710987

[4] Kairouz P, McMahan HB, Avent B, Bellet A, Bennis M, Bhagoji AN, Bonawitz K, Charles Z, Cormode G, Cummings R et al (2019) Advances and open problems in federated learning. arXiv preprint arXiv:191204977

[5] Kingma DP, Ba J (2017) Adam: a method for stochastic optimization. 1412.6980

[6] Liu W, Chen L, Chen Y, Zhang W (2020) Accelerating federated learning via momentum gradient descent. IEEE Trans Parallel Distrib Syst 31(8):1754–1766

[7] Elgabli A, Park J, Bedi AS, Bennis M, Aggarwal V (2020) GADMM: fast and communication efficient framework for distributed machine learning. J Mach Learn Res 21(76):1–39

[8] Elgabli A, Park J, Ahmed S, Bennis M (2020) L-FGADMM: layer-wise federated group ADMM for communication efficient decentralized deep learning. In: 2020 IEEE wireless communications and networking conference (WCNC). IEEE, pp 1–6

[9] Li T, Sahu AK, Zaheer M, Sanjabi M, Talwalkar A, Smith V (2018) Federated optimization in heterogeneous networks. arXiv preprint arXiv:181206127

[10] Malinovskiy G, Kovalev D, Gasanov E, Condat L, Richtarik P (2020) From local SGD to local fixed-point methods for federated learning. In: International conference on machine learning. PMLR, pp 6692–6701

[11] Ge N, Li G, Zhang L, Liu YLY (2021) Failure prediction in production line based on federated learning: an empirical study. arXiv preprint arXiv:210111715

[12] Quinlan JR (1986) Induction of decision trees. Mach Learn 1(1):81–106

[13] Xu R, Baracaldo N, Zhou Y, Anwar A, Joshi J, Ludwig H (2021) FedV: Privacy-preserving federated learning over vertically partitioned data. arXiv preprint arXiv:210303918

[14] VepakommaP, Gupta O, Swedish T, Raskar R (2018) Split learning for health: distributed deep learning without sharing raw patient data. arXiv preprint arXiv:181200564

[15] Wang K, Mathews R, Kiddon C, Eichner H, Beaufays F, Ramage D (2019) Federated evaluation of on-device personalization. arXiv preprint arXiv:191010252

[16] Mansour Y, Mohri M, Ro J, Suresh AT (2020) Three approaches for personalization with applications to federated learning. arXiv preprint arXiv:200210619

[17] Grimberg F, Hartley MA, Karimireddy SP, Jaggi M (2021) Optimal model averaging: towards personalized collaborative learning. In: Proceedings of the international workshop on federated learning for user privacy and data confidentiality. https://fl-icml.github.io/2021/papers

[18] I. Sharafaldin, A. H. Lashkari, S. Hakak and A. A. Ghorbani, "Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy," 2019 International Carnahan Conference on Security Technology (ICCST), Chennai, India, 2019, pp. 1-8, doi: 10.1109/CCST.2019.8888419

[19] R. Doriguzzi-Corin, S. Millar, S. Scott-Hayward, J. Martínez-del-Rincón and D. Siracusa, "Lucid: A Practical, Lightweight Deep Learning Solution for DDoS Attack Detection," in IEEE Transactions on Network and Service Management, vol. 17, no. 2, pp. 876-889, June 2020, doi: 10.1109/TNSM.2020.2971776.

[20] Sunwoo Lee, Anit Kumar Sahu, Chaoyang He, Salman Avestimehr, Partial model averaging in Federated Learning: Performance guarantees and benefits, Neurocomputing, Volume 556, 2023, 126647, ISSN 0925-2312, <https://doi.org/10.1016/j.neucom.2023.126647>.

[21] Roberto Doriguzzi-Corin, Domenico Siracusa, FLAD: Adaptive Federated Learning for DDoS attack detection, Computers & Security, Volume 137, 2024, 103597, ISSN 0167-4048, https://doi.org/10.1016/j.cose.2023.103597.

[22] Ludwig, Heiko., & Baracaldo, Nathalie. (Eds.). (2022). *Federated Learning A Comprehensive Overview of Methods and Applications* (1st ed. 2022.). Springer International Publishing. <https://doi.org/10.1007/978-3-030-96896-0>

[23] Yaochu, Jin., Han-chao, Zhu., Jinjin, Xu., Yang, Chen. (2023). Federated Learning: Fundamentals and Advances. Machine Learning: Foundations, Methodologies, and Applications, doi: 10.1007/978-981-19-7083-2