ĐẠI HỌC QUỐC GIA TP. HCM
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

BÁO CÁO TỔNG KẾT
ĐỀ TÀI KHOA HỌC VÀ CÔNG NGHỆ SINH VIÊN NĂM 2021

Tên đề tài tiếng Việt:

AN TOÀN TRONG TRAO ĐỔI DỮ LIỆU ĐỐI VỚI MÔ HÌNH HỌC MÁY FEDERATED LEARNING

Tên đề tài tiếng Anh:

TOWARDS SECURE AND PRIVACY-PRESERVING DATA COMMUNICATION IN FEDERATED LEARNING

Khoa/ Bộ môn: Mạng máy tính và truyền thông

Thời gian thực hiện: 6 tháng

Cán bộ hướng dẫn: ThS. Nguyễn Thanh Hoà

Tham gia thực hiện

TT	Họ và tên, MSSV	Chịu trách nhiệm	Điện thoại	Email
1.	Nguyễn Thái Tài	Chủ nhiệm	0795386962	17521001@gm.uit.edu.vn
2.	Nguyễn Thành Nhân	Tham gia	0336585836	17520840@gm.uit.edu.vn

ĐẠI HỌC QUỐC GIA TP. HCM
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

Ngày nhận hồ sơ	
Mã số đề tài	
(Do CQ quản lý ghi)	

BÁO CÁO TỔNG KẾT

Tên đề tài tiếng Việt:

AN TOÀN TRONG TRAO ĐỔI DỮ LIỆU ĐỐI VỚI MÔ HÌNH HỌC MÁY FEDERATED LEARNING

Tên đề tài tiếng Anh:

TOWARDS SECURE AND PRIVACY-PRESERVING DATA COMMUNICATION IN FEDERATED LEARNING

Ngày ... tháng ..... năm	***Ngày ... tháng ..... năm ....***
Cán bộ hướng dẫn	Sinh viên chủ nhiệm đề tài
(Họ tên và chữ ký)	***(Họ tên và chữ ký)***
Nguyễn Thanh Hoà	**Nguyễn Thái Tài**

THÔNG TIN KẾT QUẢ NGHIÊN CỨU

1. Thông tin chung:

- Tên đề tài:

AN TOÀN TRONG TRAO ĐỔI DỮ LIỆU ĐỐI VỚI MÔ HÌNH HỌC MÁY FEDERATED LEARNING

- Chủ nhiệm: **Nguyễn Thái Tài**

- Thành viên tham gia: **Nguyễn Thành Nhân**

- Cơ quan chủ trì: Trường Đại học Công nghệ Thông tin.

- Thời gian thực hiện: 6 tháng

2. Mục tiêu:

Trong những năm gần đây, với sự bùng nổ về dữ liệu và phát triển vượt bậc về công nghệ đã tạo nên điều kiện để thúc đẩy mạnh mẽ các thiết bị di động và Internet of Things (IoT). Ở thời điểm hiện tại, các thiết bị có khả năng cảm biến và tính toán ngày càng tiên tiến, kết hợp chung với sự thăng tiến của kỹ thuật Machine Learning (ML - Học máy), Deep Learning (DL - Học sâu) đã giải quyết được nhiều bài toán và mở ra vô số khả năng để giải quyết các vấn đề trong cuộc sống. Về mô hình ML truyền thống, dữ liệu được gửi từ nhiều nguồn như thiết bị di động hay IoT và được đưa về xử lý ở máy chủ trung tâm. Mặc dù mô hình này có sự hiệu quả và những điểm ưu việt riêng, thế nhưng vẫn có những khuyết điểm và không còn thực sự phù hợp cho hệ thống hiện nay. Vào năm 2019, hơn 590 triệu hồ sơ người dùng Facebook đã bị lộ trên hệ thống cloud của Amazon [1]. Đó cũng là yếu điểm đầu tiên của mô hình ML truyền thống: không đảm bảo được tính riêng tư với dữ liệu nhạy cảm của người dùng [2]. Trong thực tế, các dữ liệu được phân tán qua nhiều tổ chức khác nhau dưới sự bảo vệ về quyền riêng tư bởi các luật như General Data Protection Regulation (GDPR - Luật tổng hợp bảo vệ dữ liệu chung) [3]. Điều này làm cho các thuật toán ML hiện nay đang trong tình trạng bị đói dữ liệu (data hungry). Thứ hai, vấn đề liên quan tới việc thu thập dữ liệu đối mặt với sự chậm trễ trong lan truyền và các độ trễ không cần thiết khiến cho mô hình này đòi hỏi rất nhiều thời gian và chi phí [4].

Để giải quyết những thách thức trên, một mô hình học máy mới đã được đề xuất, phát triển và giới thiệu lần đầu tiên vào năm 2016 bởi McMahan [5] có tên là Federated Learning (FL - Học liên kết). Trong FL, việc học sẽ được diễn ra trên chính thiết bị người dùng và các thiết bị này sẽ cùng tham gia cộng tác với nhau thông qua sự điều phối của máy chủ trung tâm. Những dữ liệu riêng tư của người dùng vẫn sẽ lưu trữ ở thiết bị mà không bị chuyển đi bất kỳ nơi nào khác, chỉ những thay đổi của mô hình sau quá trình huấn luyện mới được chia sẻ với máy chủ trung tâm. Do đó, mô hình FL sẽ đảm bảo được tính riêng tư cho dữ liệu nhạy cảm của người dùng [6].

Theo [1], FL hiện tại là một hướng nghiên cứu rất mới và thu hút được rất nhiều sự quan tâm từ cộng đồng nghiên cứu Machine Learning. Sự gia tăng nhu cầu về công nghệ sử dụng mô hình Federated Learning đã dẫn đến việc ra đời của nhiều framework, công cụ. Nổi bật trong số đó có thể kể tên Federated AI Technology Enabler (FATE) được phát triển bởi Webank, TensorFlow Federated (TFF) được phát triển bởi TensorFlow thuộc Google, PySyft phát triển bởi OpenMined, PaddleFL của nhà phát triển Baidu.

Để tiếp cận cũng như đánh giá hiệu quả trong việc đảm bảo tính riêng tư dữ liệu người dùng của hệ thống dựa trên mô hình FL, trong đề tài nghiên cứu khoa học này, nhóm tác giả sẽ nghiên cứu mô hình FL đồng thời tích hợp các kỹ thuật phù hợp để đảm bảo an toàn cho giai đoạn tổng hợp cũng như trao đổi dữ liệu giữa các thành phần trong hệ thống. Sau đó sẽ thực hiện đánh giá để so sánh với mô hình học máy phân tán truyền thống**.**

3. Tính mới và sáng tạo:

Theo khảo sát [17], Federated Learning đang là một hướng nghiên cứu rất mới và thu hút được nhiều sự quan tâm từ cộng đồng nghiên cứu Machine Learning. Ví dụ như Google đã giới thiệu một sơ đồ tổng hợp an toàn để bảo vệ quyền riêng tư của các bảncập nhật của người dùng được tổng hợp bên dưới thư viện hỗ trợ Federated Learning của họ [25]. CryptoNets [7] đã điều chỉnh các tính toán mạng nơ-ron để hoạt động với dữ liệu được mã hóa thông qua Mã hóa đồng hình (HE). SecureML [21] là một sơ đồ tính toán đa bên sử dụng tính năng chia sẻ bí mật và Yao’s Garbled Circuit để mã hóa và hỗ trợ đào tạo cộng tác cho quá trình tuyến tính, hồi quy hậu cần và mạng nơ-ron.

Sự gia tăng nhu cầu về công nghệ sử dụng mô hình FL đã dẫn đến việc ra đời của nhiều nền tảng (framework), công cụ. Các nền tảng mã nguồn mở được nghiên cứu,phát triển nhanh chóng, nổi bật trong số đó có

thể kể đến như Federated AI TechnologyEnabler (FATE) được phát triển bởi Webank, TensorFlow Federated (TFF) được phát triển bởi TensorFlow thuộc Google, PySyft phát triển bởi OpenMined, PaddleFL của nhóm nghiên cứu tại Baidu.

Federated Learning có thể giải quyết vấn đề bảo vệ quyền riêng tư trong mô hình học sâu bằng cách kết hợp mô hình học máy phân tán, các cơ chế kỹ thuật mật mã và bảo mật. Vì vậy, FL có thể trở thành nền tảng của thế hệ học máy tiếp theo phục vụ cho nhu cầu công nghệ, xã hội để phát triển và ứng dụng trí tuệ nhân tạo [18].

4. Tóm tắt kết quả nghiên cứu:

Trong đề tài này, nhóm tác giả đã xây dựng được các mô hình thực nghiệm như: 1) Mô hình học máy phân tán truyền thống, 2) Mô hình FL cross-silo. Nghiên cứu và ứng dụng thành công kỹ thuật HE và DP vào trong hệ thống đã xây dựng, đồng thời tiến hành thực nghiệm và thu thập các số liệu cần thiết. Nhóm đánh giá các kết quả thu được qua 3 kịch bản chính là:

- Kịch bản 1: So sánh mô hình học máy phân tán truyền thống và mô hình FL cross-silo nhằm đánh giá được ưu và nhược điểm của 2 mô hình này.
- Kịch bản 2: Ứng dụng kỹ thuật HE và DP vào mô hình FL.

Từ 3 kịch bản trên nhóm sẽ rút ra được các đánh giá khách quan về các điểm vượt trội của mô hình FL cross-siloso với mô hình học máy phân tán truyền thống. Nhóm còn đưa ra các đánh giá về 2 kỹ thuật DP và HE khi ứng dụng vào FL.

4.1. Cơ sở lý thuyết và các nghiên cứu liên quan

4.1.1. Mô hình hắc máy phân tán (Distributed Machine Learning)

Mô hình học máy phân tán là mô hình đề cập đến các thuật toán và hệ thống học máy đa bên được thiết kế để cải thiện hiệu suất, tăng độ chính xác và mở rộng quy mô kích thước dữ liệu đầu vào lớn hơn. Việc tăng kích thước dữ liệu đầu vào có thể khiến cho nhiều thuật toán giảm đáng kể lỗi học tập và thường có hiệu quả so với việc sửdụng các phương pháp phức tạp hơn [12]. Mô hình học máy phân tán cho phép các công ty, nhà nghiên cứu và những người trong bộ phận đưa ra quyết định sáng suốt và kết luận rút ra từ một lượng lớn dữ liệu.

Nhiều hệ thống tồn tại để thực hiện các tác vụ học máy trong môi trường phân tán. Các hệ thống này chia thành ba loại chính: cơ sở dữ liệu, hệ thống chung và hệ thống được xây dựng theo mục đích. Mỗi loại hệ thống có những ưu điểm và nhược điểm riêng biệt, nhưng tất cả đều được sử dụng trong thực tế tùy thuộc vào các trường hợp sử dụng cá nhân, yêu cầu hiệu suất, kích thước dữ liệu đầu vào và lượng nỗ lực triển khai.

Trong mô hình học máy phân tán thì tất cả dữ liệu sẽ được đưa lên máy chủ trung tâm và đây cũng là một trong những thử thách lớn của mô hình này. Vì những dữ liệu có thể chứa thông tin nhạy cảm của người dùng cuối hoặc tổ chức, chẳng hạn như hình ảnh khuôn mặt, dịch vụ dựa trên vị trí, thông tin chữa bệnh [19] hoặc tình trạng kinh tế cá nhân. Việc di chuyển dữ liệu thô từ các thiết bị cá nhân hoặc trung tâm dữ liệu của nhiều tổ chức sang máy chủ hoặc trung tâm dữ liệu tập trung có thể gây rò rỉ thông tin tức thì hoặc tiềm ẩn. Điều này làm cho việc tổng hợp dữ liệu từ các thiết bị phân tán, nhiều khu vực hoặc tổ chức hầu như không thể computing [27]. Chính vì những thách thức trên mà mở ra tiềm năng vô cùng lớn cho mô hình học máy Federated Learning.

4.1.2. Học hợp tác (Federated Learning)

Học hợp tác (FL - Federated Learning) là mô hình học máy mà trong đó việc huấn luyện sẽ được diễn ra trên đối tượng tham gia (thiết bị di động hoặc tổ chức) và các đối tượng này sẽ cùng tham gia cộng tác với nhau để huấn luyện một mô hình thông qua sự điều phối của máy chủ trung tâm. Những dữ liệu riêng tư của người dùng (tổ chức) vẫn sẽ được lưu trữ ở thiết bị mà không bị chuyển đi bất kỳ nơi nào khác, chỉ những thay đổi của mô hình sau quá trình huấn luyện mới được chia sẻ với máy chủ trung tâm. Mô hình FL giải quyết đáng kể các vấn đề của các mô hình học máy truyền thống như đảm bảo tính riêng tư của người dùng và độ trễ lớn trong lan truyền [13].

Ban đầu thuật ngữ Federated Learning được giới thiệu để ứng dụng vào các thiết bị di động hoặc các thiết bị ngoại biên (edge device) nhưng gần đây việc ứng dụng mô hình FL cho một nhóm các người dùng đáng tin cậy (ví dụ: các tổ chức y tế, tài chính) bắt đầu thu hút được nhiều sự quan tâm hơn từ giới nghiên cứu. Hai dạng ứng dụng này được gọi là "cross-device" và "cross-silo" [13].

4.1.2.1. Mô hình FL cross-device

Mô hình cross-device FL là một dạng mô hình học máy phân tán với một máy chủ trung tâm cùng rất nhiều đối tượng tham gia (thường là các thiết bị di động hoặc IoT). Quá trình huấn luyện sử dụng dữ liệu của

thiết bị và diễn ra cũng tại thiết bị. Mặc dù quá trình huấn luyện, cập nhập hoàn toàn được máy chủ trung tâm điều hành nhưng dữ liệu của thiết bị sẽ không bị đọc hay gửi đi bất kỳ đâu trong suốt thời gian diễn ra các quá trình này. Ngoài ra, các thiết bị này sẽ không được định danh bằng bất cứ dạng thông tin gì bởi máy chủ trung tâm. Tuy nhiên, số lượng thiết bị quá lớn khiến cho quá trình huấn luyện và cập nhập mô hình khó có thể diễn ra đồng loạt trên tất cả thiết bị. Điều này xảy ra do các thiết bị không đáp ứng đủ các điều kiện cần thiết cho để thựchiện các quá trình trên (không đủ lượng pin tối thiểu, không được kết nối mạng,...) [11].

4.1.2.2. Mô hình FL cross-silo

So với mô hình cross-device FL thì mô hình cross-silo FL lại có quy mô nhỏ hơn nhiều với chỉ khoảng từ 2-100 đối tượng tham gia[27]. Mô hình này thường được sử dụng bởi nhiều tổ chức y tế hoặc tài chính vì độ tin cậy cao mà nó mang lại. Bởi vì đối tượng tham gia mô hình chủ yếu là các tổ chức cho nên kết nối đến máy chủ trung tâm luôn được duy trì ổn đỉnh, điều này có nghĩa là dữ liệu luôn có tính sẵn sàng cao.Vì vậy các quá trình huấn luyện, cập nhật, trao đổi dữ liệu sẽ diễn ra suôn sẻ. Các đốitượng tham gia sẽ được định danh cụ thể và tham gia vào mô hình.

4.1.3. Các kỹ thuật đảm bảo tính an toàn cho mô hình học hợp tác

4.1.3.1. Mã hoá đồng hình (Homomorphic Encryption)

Mã hoá đồng hình là tên gọi chung của các loại mã hoá cho phép tính toán trên dữliệu được mã hoá mà không cần thao tác giải mã trước đó [20]. Một điều kiện tối quan trọng trong HE chính là kết quả của một phép toán thực hiện trên dữ liệu được mã hoá phải tương đồng với kết quả của cùng một phép toán thực hiện trên dữ liệu gốc. Với những tính chất trên, HE mang một tiềm năng ứng dụng to lớn bởi nó cho phép mộtbên thứ ba có thể thực hiện phép tính, thuật toán trên dữ liệu mã hoá mà không cần thực thi bất kỳ kiểu truy cập nào vào dữ liệu gốc. Vì vậy, dữ liệu của người dùng được bảo vệ và đảm bảo an toàn trong khi bên thứ ba thực hiện các tác vụ (nghiên cứu, thống kê, dùng cho học máy, ...). Mã hoá đồng hình đã được ứng dụng vào nhiều lĩnh vực khác nhau như tài chính, kinh doanh, y tế hoặc bất kỳ lĩnh vực nào cần phải làm việc với những dữ liệu nhạy cảm.

Đi sâu hơn vào hình thức của mã hoá đồng hình, một biểu thứcg: $A \rightarrow B$ sẽ được gọi là đồng hình nếu như thoả điều kiện sau:

Ngoài một số các thuật toán có trên các hệ thống mã hoá thông thường khác như: sinh khoá, mã hoá hay giải mã, mã hoá đồng hình còn có một thuật toán cộng có tên là evaluation (Eval) [20], thuật toán này là mô tả chính thức của quy tắc phía trên. Đầu vào và đầu ra của thuật toán Eval là dạng dữ liệu đã mã hoá. Trong thuật toán Eval,biểu thức g được thực hiện lên dữ liệu mã hoá c1và c2 mà không cần truy cập vào dữ liệu gốc m1và m2, tính chất này được thể hiện như sau:

$$Dec(key_{priv}, Eval_g(key_{eval}, c_1, c_2)) = f(m_1, m_2)$$

Trong mã hoá đồng hình, chỉ hai phép tính cần phải mang tính đồng hình là phép cộng (OR) và phép nhân (AND). Mã hoá đồng hình có thể phân làm 3 dạng [20]:

- **Mã hoá đồng hình một phần** (PHE - Partial homomorphic encryption): Các loại mã hoá được gọi là PHE khi nó chỉ hỗ trợ một phép tính thực hiện trên dữ liệu mã hoá với số lần thực hiện không giới hạn. Các loại mã hoá tiêu biểu thuộc dạng này là: RSA, Goldwasser-Micali và El-Gamal.
- **Phần nào đó là mã hoá đồng hình** (SWHE - Somewhat homomorphic encryption): là dạng cho hỗ trợ cả hai phép toán nhưng chỉ có thể thực hiện một số lần nhất định các phép toán trên ciphertext, nếu vượt quá số lần được cho phép kết quả sẽ sai lệch.
- **Mã hoá đồng hình toàn phần** (FHE - Fully homomorphic encryption): là loại mã hoá có thể hỗ trợ cả hai phép tính với số lần tính toán không giới hạn. FHE được xem là "Dao xếp đa năng Thuỵ Sĩ của ngành mật mã học" bởi khả năng tính toán không giới hạn trên dữ liệu mã hoá [20]. Các loại mã hoá thuộc dạng FHE vẫn đang được phát triển và hoàn thiện bởi tính phức tạp và tính trừu tượng của mã hoá đồng toàn phần.

4.1.3.2. Quyền riêng tư khác biệt (Differential Privacy)

Quyền riêng tư khác biệt (DP) là một định nghĩa toán học về quyền riêng tư, có mục đích chung chính là đảm bảo các loại phân tích thống kê khác nhau thực hiện trên một tập dữ liệu sẽ không ảnh hưởng đến quyền riêng tư. DP mô tả một lời hứa của những người giữ dữ liệu (Dataholders) đối với một thông tin dữ liệu cá nhân (Data subject) và lời hứa đó là: "Bạn sẽ không bị ảnh hưởng, gặp bất lợi khác, khi cho phép dữ liệu củabạn được sử dụng cho bất kỳ nghiên cứu hoặc phân tích nào, mặc cho các nghiên cứu, các tập dữ liệu hoặc các nguồn thông tin có liên quan đã có sẵn." (Dwork). Tập dữ liệu đảm bảo DP thì thông tin cá nhân của một người sẽ không bị lộ khi tập dữ liệu chứa thông tin của người đó được tìm

hiểu, nghiên cứu [8]. Cụ thể hơn, DP đảm bảo một tập dữ liệu bị thêm hoặc bớt đi một mục sẽ không gây ảnh hưởng gì đến kết quả của bất kỳ phân tích hay thuật toán nào thực hiện trên tập dữ liệu đó [8]. Qua đó, cá nhân tham gia vào tập dữ liệu sẽ không gặp bất cứ rủi ro nào.

Quyền riêng tư khác biệt cục bộ (Local Differential Privacy)

Local DP là một phương thức đảm bảo tính DP bằng cách thêm nhiễu (noise) vào dữ liệu của cá nhân trước khi thêm dữ liệu này vào tập dữ liệu, cơ sở dữ liệu [8]. Phương thức này mang lại sự riêng tư cho dữ liệu cá nhân nhưng lại ảnh hưởng đến độ chính xác của tập dữ liệu, cơ sở dữ liệu. Độ nhiễu càng tăng thì độ chính xác của tập dữ liệu càng giảm, nhưng Local DP lại đặc biệt hiệu quả với các tập dữ liệu có quy mô lớn, quy mô của tập dữ liệu càng lớn thì độ chính xác của tập dữ liệu sau khi áp dụng Local DP lại càng cao. Với các tập dữ liệu có quy mô lớn, số lượng cá nhân có thống kê dữ liệu giống nhau sẽ càng nhiều cho nên sau khi làm nhiễu tập dữ liệu sẽ vẫn giữ được độ chính xác cao so với tập dữ liệu ban đầu. Phương thức này thường được sử dụng trong trường hợp cá nhân cụ thể không hoàn toàn tin tưởng vào Dataholders.

Quyền riêng tư khác biệt toàn cục (Global Differential Privacy)

Global DP là một phương thức đảm bảo tính DP bằng cách thêm nhiễu (noise) vào đầu ra của truy vấn được thực hiện trên tập dữ liệu, cơ sở dữ liệu [8]. Phương thức này tăng sự riêng tư cho dữ liệu cá nhân mà vẫn đảm bảo độ chính xác của tập dữ liệu. Phương thức này thường được sử dụng cho các trường hợp yêu cầu độ chính xác cao của tập dữ liệu và khi cá nhân cung cấp dữ liệu tin tưởng tuyệt đối vào Dataholders.

4.1.4. Các thư viện hỗ trợ

4.1.4.1. PyTorch

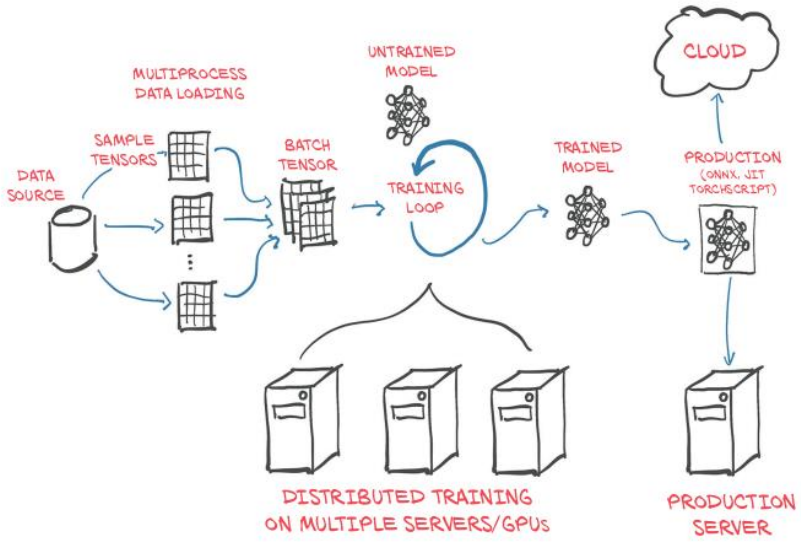
PyTorch[1] là một thư viện tensor được tối ưu hoá cho học sâu sử dụng GPU và CPU. PyTorch được xem như là một giao diện người dùng của Torch, nó tập trung chủ yếu vào việc cung cấp các thuật toán và tính gradient của chúng. Thư viện PyTorch hỗ trợ cho bộ xử lý đồ hoạ (GPU) khá tốt giúp cho khả năng tính toán của thư viện này là tương đối nhanh [14].

[1] https://pytorch.org

Hình 4.1: PyTorch

PyTorch tương đối khác với các thư viện hỗ trợ cho DL khác như Theano hay Tensorflow, chủ yếu là trong mẫu hình lập trình. Các thư viện như Theano và Tensorflow về cơ bản là tuân theo mô hình định nghĩa-biên dịch-thực thi (define-compile-run), còn PyTorch là một thư viện động (define-by-run), điều này nghĩa là không có bước biên dịch trước khi thực thi. Người dùng có thể định nghĩa các biểu thức toán học và trực tiếp gọi ra một toán tử để tính toán gradient của một biểu thức cụ thể [14].

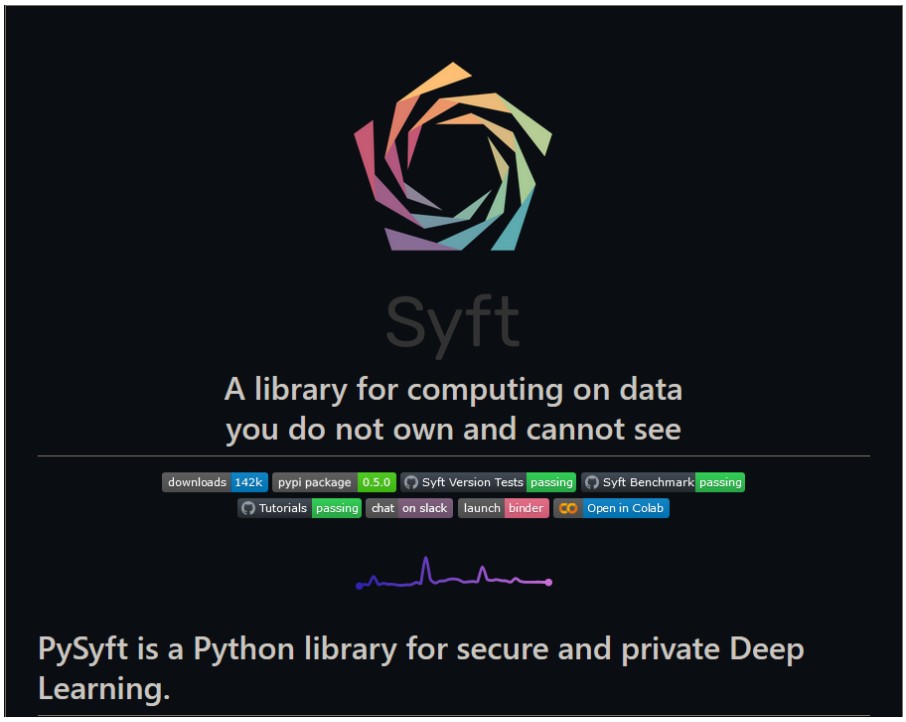
PyTorch là một thư viện hỗ trợ phù hợp cho mục đích nghiên cứu, bởi nó khá dễ dàng cho việc phát triển và thử nghiệm các kiến trúc DL mới. Mã nguồn được viết bằng PyTorch tương đối trực quan, các biểu thức toán học trong PyTorch mô tả khá chặt chẽ các mạng Nơ-ron nhân tạo. Thư viện PyTorch cũng dễ dàng để sửa lỗi hơndo tính chất động của nó. Tuy nhiên, so với mô hình define-by-run của PyTorch thì mô hình define-compile-run lại cung cấp nhiều không gian hơn để tối ưu hoá các tính toán cơ bản [14].



Hình 4.2: Một dự án học máy triển khai với Pytorch đơn giản

4.1.4.2. PySyft

PySyft[2] là một thư viện hỗ trợ cho phép tính toán riêng tư, bảo mật trong các mô hình Học sâu. PySyft kết hợp Học hợp tác (Federated Learning), Tính toán đa bên an toàn (Secure Multi-party Computation) và Quyền riêng tư khác biệt (Differential Privacy) trong một mô hình lập trình duy nhất được tính hợp vào trong thư viện hỗ trợ Học sâu khác nhau như PyTorch, Keras hoặc TensorFlow. Các nguyên tắc của PySyft ban đầu được phát thảo trong bài nghiên cứu [24] và được triển khai đầu tiên bởi Openmined - một trong những cộng đồng nghiên cứu Trí tuệ nhân tạo phi tập trung hàng đầu.



Hình 4.3: Thư viện PySyft được phát triển bởi Openmined trên Github

Thành phần cốt lõi của PySyft là SyftTensors. SyftTensors đại diện cho một trạng thái hoặc sự chuyển đổi dữ liệu và có thể được liên kết với nhau. Cấu trúc chuỗi luôn có tensor PyTorch ở đầu và các phép biến đổi hoặc trạng thái được thể hiện bởi SyftTensors được truy cập từ trên xuống bằng cách sử dụng thuộc tính con và trở lên bằng cách sử dụng thuộc tính cha [24].

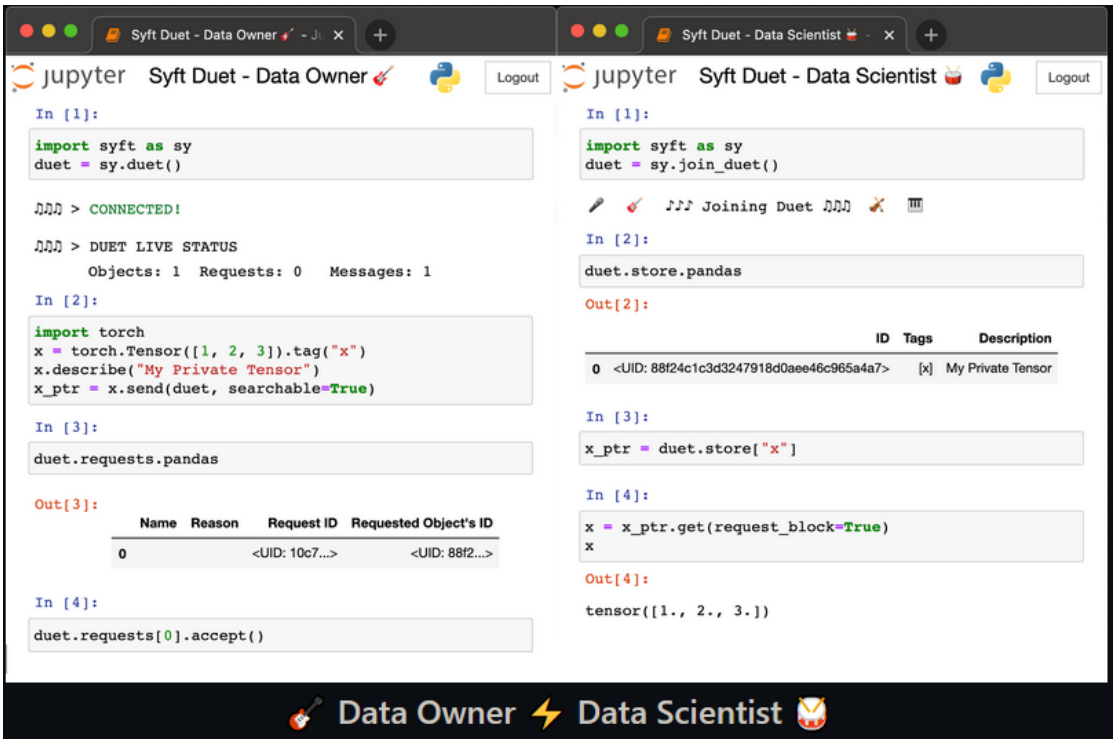
PySyft đại diện cho một trong những nỗ lực đầu tiên nhằm tăng cường các mô hình bảo mật mạnh mẽ trong các chương trình học sâu. Trong quá trình phát triển, quyền riêng tư có khả năng trở thành một

[2] https://github.com/OpenMined/PySyft

trong những nền tảng cơ bản của thế hệ tiếp theo của thư viện hỗ trợ học sâu.

Duet

Duet[3] là công cụ peer-to-peer trong PySyft cung cấp API nghiên cứu cho DataOwner để có thể gửi dữ liệu một cách riêng tư, trong khi Data Scientist có thể truy cập hoặc thao tác dữ liệu từ phía chủ sở hữu thông qua cơ chế kiểm soát truy cập zero-knowlegde. Duet được phát triển để làm giảm đi rào cản giữa nghiên cứu và các cơ chế bảo vệ quyền riêng tư, nên các nghiên cứu khoa học có thể được thực hiện trên dữ liệu mà không ảnh hưởng đến quyền riêng tư. Lợi ích chính của việc sử dụng Duet là cho phép sử dụng PySyft mà không cần quản lý việc triển khai PyGrid đầy đủ. Đây là cách đơn giản nhất để sử dụng Syft mà không cần cài đặt bất cứ thứ gì.



Hình 4.4: Ứng dụng minh hoạ Duet

4.1.4.3. TenSEAL

TenSEAL[4] là một thư viện kết nối các thư viện hỗ trợ học máy cổ điển với các khả năng mã hóa đồng hình. Thư viện này dễ dàng hoá việc thực hiện các phép tính trên tensor lên các dữ liệu được mã hoá.

[3] https://blog.openmined.org/duet-opengrid-infrastructure-for-easy-remote-data-science/
[4] https://github.com/OpenMined/TenSEAL

TenSEAL hoạt động dựa trên Microsoft SEAL[5], cụ thể hơn là lượt đồ CKKS và BFV của Microsoft SEAL. Người dùng có thể làm việc với dữ liệu gốc hoặc được mã hoá bằng ngôn ngữ C++ hoặc Python thông qua TenSEAL. Các API của TenSEAL được xây dựng xoay quanh ba thành phần chính là: ngữ cảnh(context), tensor gốc (plain tensor) và tensor mã hoá (encrypted tensor) [3].

TenSEAL Context

TenSEAL context là một thành phần chính của thư viện TenSEAL. Nó có chức năng tạo và lưu trữ các khoá cần thiết trong cho việc thực hiện các phép tính trên dữ liệu mã hoá. Context tạo ra khoá bí mật để giải mã dữ liệu, khoá công khai để mã hoá dữ liệu, các khoá Galois để thực hiện phép quay ma trận và các khoá tái định dạng (relinearization key) để tái cấu trúc các dữ liệu mã hoá. Ngoài ra, context còn có thể chứa các nhóm luồng (threat-pool) dùng để điều khiển số lượng công việc sẽ được chạy song song khi thực hiện các hoạt động có thể song song hóa. Nó còn có thể được cấu hình để tự động tái định dạng và thay đổi tỉ lệ (rescaling) trong quá trình tính toán.

TenSEAL Plain Tensor và Encrypted Tensor

PlainTensor là một lớp thực hiện chức năng chuyển đổi các tensor chưa được mã hoá thành dạng phù hợp để thực hiện quá trình mã hoá. Đây là một bước tiền xử lý trước khi mã hoá các tensor.

EncryptedTensor là một lớp chứa TenSEALContext, đây là một thành phần không thể thiếu khi thực hiện bất kỳ hoạt động mã hoá hay tính toán nào. Ngoài ra, lớp này còn cung cấp một API cần được triển khai cho việc hiển thị các tensor thông qua thưviện TenSEAL.

4.1.4.4. Opacus

Opacus[6] là một thư viện cho phép huấn luyện các mô hình PyTorch với DP. Thư viện này hỗ trợ huấn luyện mô hình mà không cần phía client phải thực hiện quá nhiều thay đổi trong mã code, ít ảnh hưởng đến hiệu suất huấn luyện và cho phép khách hàng theo dõi trực tuyến ngân sách riêng tư (privacy budget) đã sử dụng tại bất kỳ thời điểm nào.

[5] https://www.microsoft.com/en-us/research/project/microsoft-seal/
[6] https://opacus.ai/

Hình 4.5: Thư viện Opacus

Mục đích chính của Opacus là bảo vệ quyền riêng tư của mỗi mẫu huấn luyện trong khi hạn chế những sai lệch trong kết quả dự đoán. Opacus thực hiện điều này bằng cách sửa đổi trình tối ưu hoá của PyTorch để thực thi (và đo lường) DP trong suốt quá trình huấn luyện mô hình. Opacus tập trung chủ yếu vào DP-SGD.

Opacus định nghĩa một API nhẹ (lightweight API) bằng cách khai báo một PrivacyEngine. PrivacyEngine này sẽ làm nhiệm vụ theo dõi ngân sách riêng tư cũng như xử lý các *gradient* của mô hình. Ta chỉ cần gắn API này vào trình tối ưu hoá của PyTorch, và nó sẽ làm việc tự động trong quá trình huấn luyện mô hình.

4.1.5. Các nghiên cứu liên quan

BatchCrypt: Efficient Homomorphic Encryption for Cross-Silo Federated Learning

BatchCrypt là một hệ thống được áp dụng cho mô hình học hợp tác cross-silo. Hệ thống này được phát triển bởi các tác giả thuộc Đại học Khoa học và Công nghệ Hồng Kông, Đại học Nevada và Webank. Cụ thể hơn về hệ thống này, các tác giả thay vì mã hoá các gradient với độ chính xác cao thì sẽ chuyển các gradient thành một số nguyên lớn rồi mới mã hoá. Các phương thức chuyển dạng và mã hoá cũng được phát triển riêngđể phù hợp nhất cho quá trình tổng hợp gradient. BatchCrypt được thử nghiệm trên 3 dạng mô hình khác nhau là: Linear Regresstion trên tập dữ liệu FMNIST, AlexNet trên tập dữ liệu CIFAR10 và LSTM trên tập dữ liệu Shakespeare. Hiện tại, BatchCrypt đã được bổ sung làm một mô đun thêm của thư viện FATE.

Privacy-preserving and communication-efficient federated learning in Internet of Things

Nhóm tác giả Fang et al. phát triển một công cụ mang tên PCFL (privacy-preserving and communication-efficient scheme for federated learning). PCFL bao gồm 3 thành phần chính: (1) Bộ chọn lọc bản cập nhật, (2) Bộ nén hai chiều, (3) Giao thức bảo mật quyền riêng tư. Nghiên

cứu này được phát triển nhằm đảm bảo được tính chính xác của mô hình cũng như quyền riêng tư, thực nghiệm trên 2 dạng mô hình là CNN trên tậpdữ liệu MNIST và LSTM trên tập dữ liệu Shakespeare.

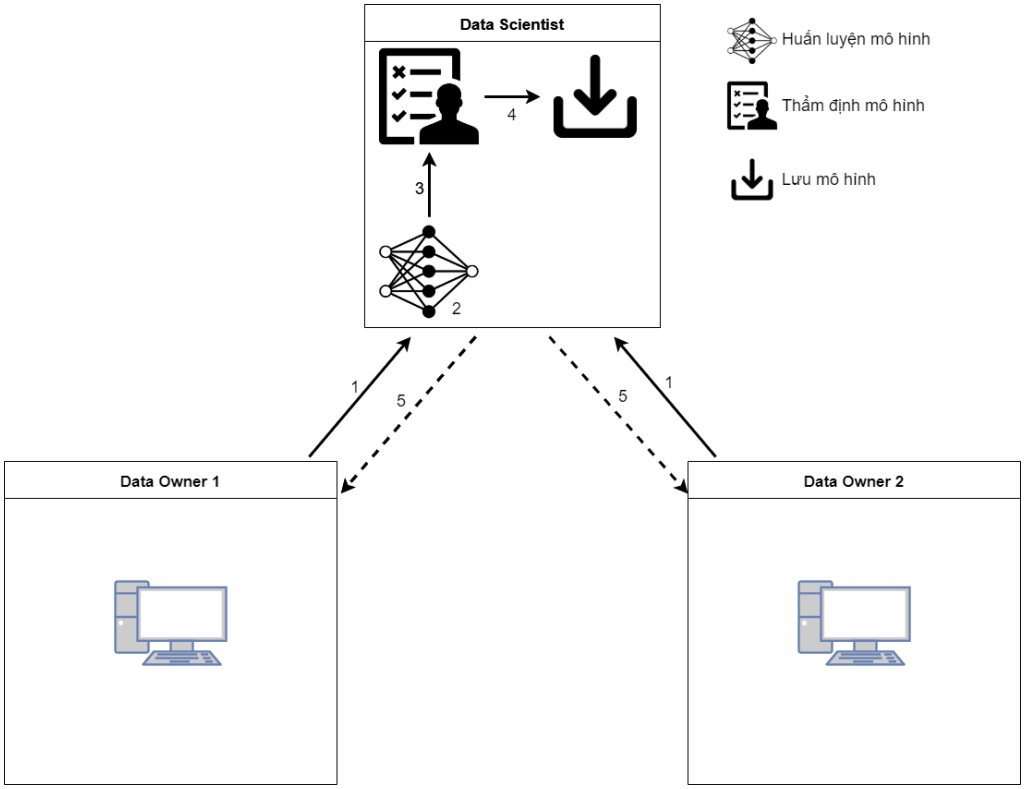
FedOpt: Towards Communication Efficiency and Privacy Preservation in Federated Learning

FedOpt (Federated Optimisation) được phát triển bởi Asad, Moustafa, and Ito. Cụ thể hơn về nghiên cứu này, các tác giả đã phát triển một thuật toán nén gọi là Thuật toán nén thưa (SCA - Sparse Compression Algorithm) để giúp tối ưu hoá việc trao đổi dữ liệu, thuật toán này kết hợp với phép cộng trong mã hoá đồng hình vào DP để ngăn chặn dữ liệu bị rò rỉ. Nghiên cứu này được thực nghiệm trên tập dữ liệu MNIST và CIFAR-10 với mô hình FL giả lập được phát triển bởi TensorFlow. Kết quả thu được từ thựcnghiệm sẽ được so sánh với các kỹ thuật khác như: Practical Secure Aggregation (PSA), Federated Extreme Boosting (XGB), Efficient and Privacy-Preserving Federated DeepLearning (EPFDL) và Privacy-preserving collaborative learning (PPCL).

4.2. Thiết kế hệ thống

4.2.1. Mô hình học máy phân tán truyền thống

Mô hình học máy phân tán truyền thống thực nghiệm được nhóm tác giả xây dựng là mô hình học máy với hai thành phần chính là người sở hữu dữ liệu (Data Owner) và người nghiên cứu dữ liệu (Data Scientist). Mô hình được xây dựng với sự hỗ trợ của các thư viện như: PyTorch, PySyft. Trong đó PyTorch đóng vai trò xây dựng mô hình DL, PySyft với chức năng Duet sẽ tạo dựng kết nối peer-to-peer giữa các Data Owner và Data Scientist.



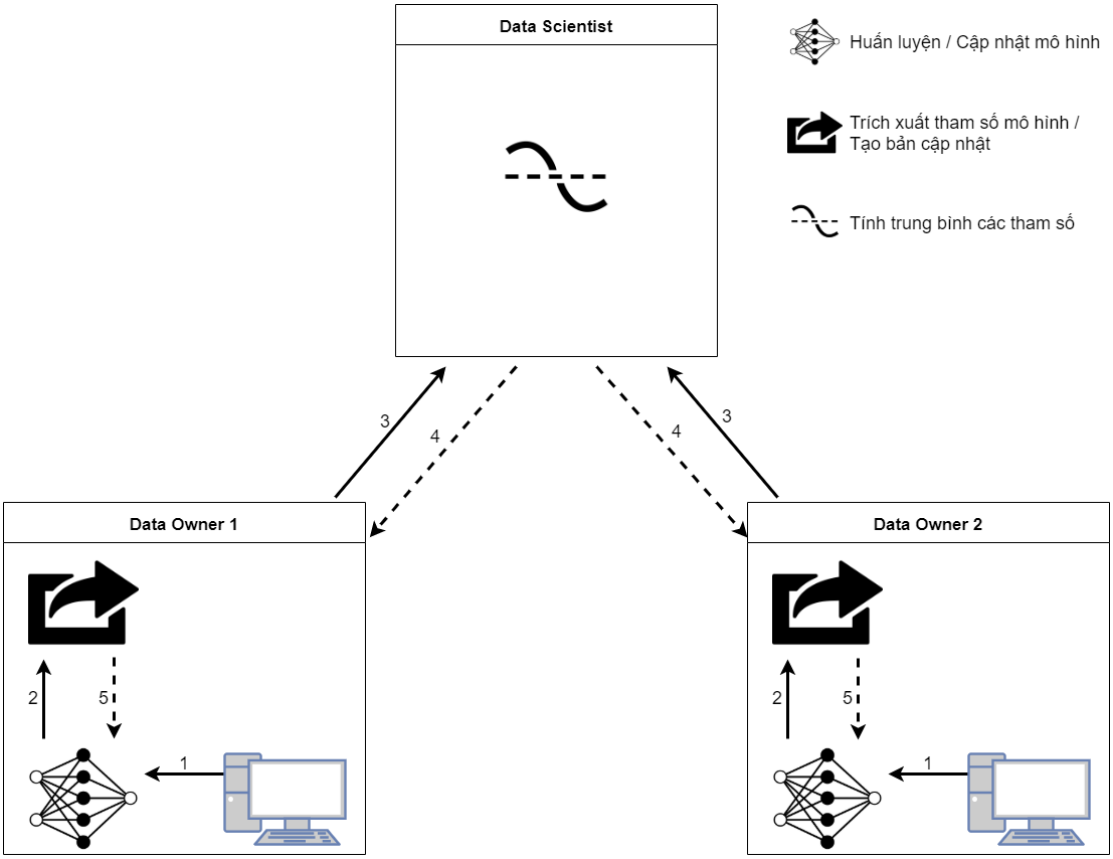
Hình 4.6: Tổng quan mô hình học máy phân tán truyền thống thực nghiệm

Các giai đoạn chính trong mô hình học máy phân tán truyền thống:

1. Dữ liệu từ các Owner sẽ được gửi đến Scientist.
2. Data Scientist gộp các dữ liệu của Owner và tiến hành huấn luyện mô hình DL.
3. Data Scientist thẩm định mô hình với tập dữ liệu kiểm tra của mình.
4. Data Scientist lưu lại mô hình DL.
5. Data Scientist gửi mô hình DL cho các Owner.

4.2.2. Mô hình FL cross-silo

Mô hình FL cross-silo thực nghiệm được nhóm tác giả xây dựng là mô hình học hợp tác theo dạng cross-silo với hai thành phần chính là người sở hữu dữ liệu (Data Owner) và người nghiên cứu dữ liệu (Data Scientist). Mô hình được xây dựng với sự hỗ trợ của các thư viện như: PyTorch, PySyft. Trong đó PyTorch đóng vai trò xây dựng mô hình DL, PySyft với chức năng Duet sẽ tạo dựng kết nối peer-to-peer giữa các Data Owner và Data Scientist.



Hình 4.7: Tổng quan mô hình FL cross-silo thực nghiệm

Các giai đoạn chính trong mô hình FL cross-silo:

1. Dữ liệu từ các Owner sẽ được dùng để huấn luyện mô hình DL.
2. Data Owner trích xuất các tham số mô hình.
3. Data Owner gửi các tham số mô hình đến Scientist để tổng hợp.
4. Data Scientist gửi bản cập nhật về các Owner.
5. Data Owner giải mã bản cập nhật và cập nhật vào mô hình DL.

4.2.2.1. Data Owner

Trong mô hình này, sau khi hai Data Owner đã thiết lập kết nối thành công với Data Scientist, các Data Owner sẽ tiến hành xử lý tập dữ liệu của mình cho phù hợp với đầu vào của mô hình. Sau khi xử lý dữ liệu, mô hình DL sẽ được khởi tạo và thực hiện các bước chuẩn bị cho bước huấn luyện. Bước vào quá trình huấn luyện, các bước lan truyền tiến (Forward propagation), tínhloss(loss function), lan truyền ngược (Backward propagation) và tối ưu hoá mô hình sẽ được lần lượt thực hiện, các bước này sẽ được thực hiện với từng mẫu có trong tập dữ liệu. Song song đó, quá trình thẩmđịnh sẽ diễn ra để kiểm tra tỉ lệ dự đoán chính xác của mô hình cũng như giá trịlossthực tế khi thực hiện trên tập dữ liệu thử nghiệm, quá trình thẩm định sẽ được thực hiện sau khi một epoch huấn

luyện kết thúc. Tiếp đến, các tham số của mô hình (weight, bias của từng lớp) sẽ được rút trích ra và gửi đến Data Scientist. Sau bước này các Data Owner sẽ tạm nghỉ để đợi Data Scientist tạo ra bản cập nhật. Các Data Owner sẽ hoạt động trở lại khi nhận được bản cập nhật. Bản cập nhật sẽ được cập nhật vào mô hình và thực hiện thẩm định với tập dữ liệu thử nghiệm. Các kết quả của quá trình thẩm định sẽ được lưu lại và các giai đoạn từ gia đoạn huấn luyện sẽ được lặp lại cho đến khi đủ số lượng round nhất định.

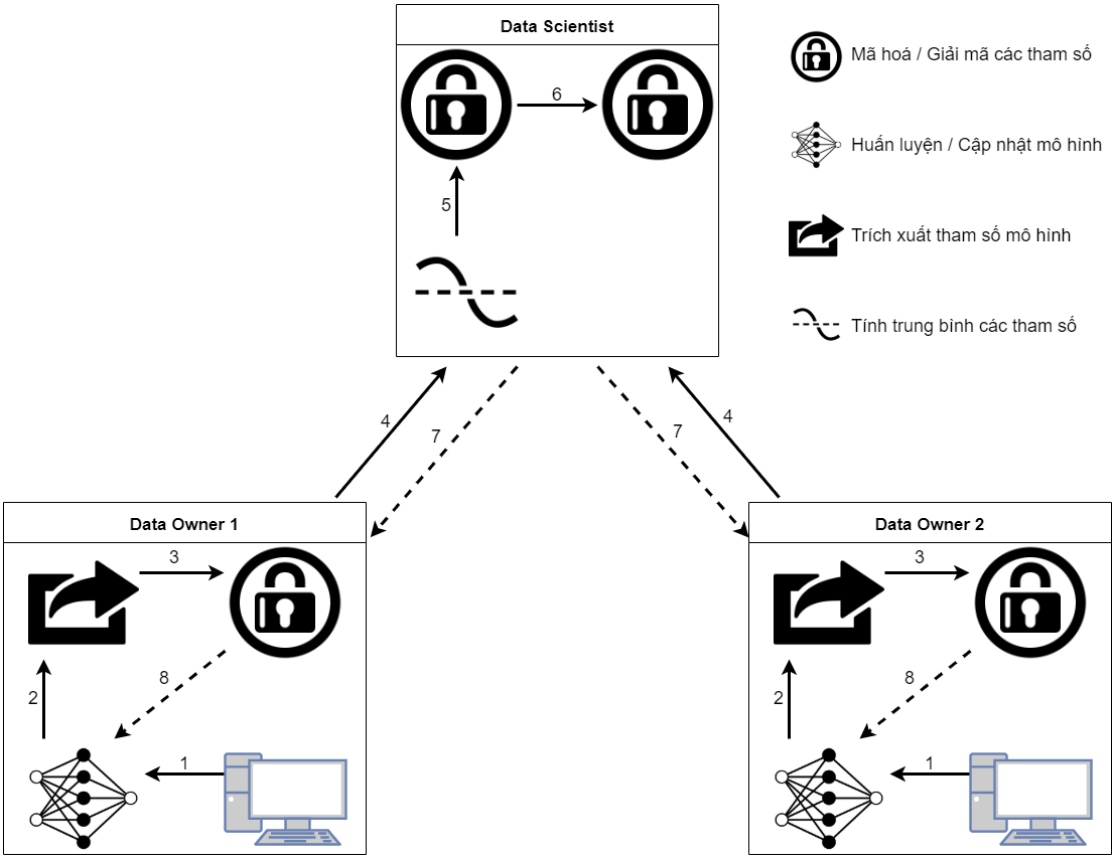
4.2.2.2. Data Scientist

Khi các Data Owner đã hoàn tất quá trình huấn luyện và gửi các tham số mô hình, Data Scientist sẽ lấy các tham số này về và tiến hành tính trung bình. Sau khi quá trình tính toán hoàn tất, Scientist sẽ gửi bản cập nhật cho các Owner. Quá trình này sẽ được lặp lại với số round nhất định.

4.2.3. Thiết kế các kỹ thuật đảm bảo an toàn trong traođổi dữ liệu cho mô hình FL

4.2.3.1. Áp dụng kỹ thuật mã hoá đồng hình CKKS với TenSEAL

TenSEAL sẽ được ứng dụng vào mô hình thực nghiệm tại quá trình trao đổi dữ liệu giữa Data Owner và Scientist. Data Owner và Scientist sẽ khởi tạo TenSEAL context của riêng mình. Context của Scientist sẽ được dùng để mã hoá các tham số mô hình của Owner trước khi gửi. Context của các Owner sẽ được dùng để mã hoá bản cập nhật trước khi gửi cho các Owner tương ứng.



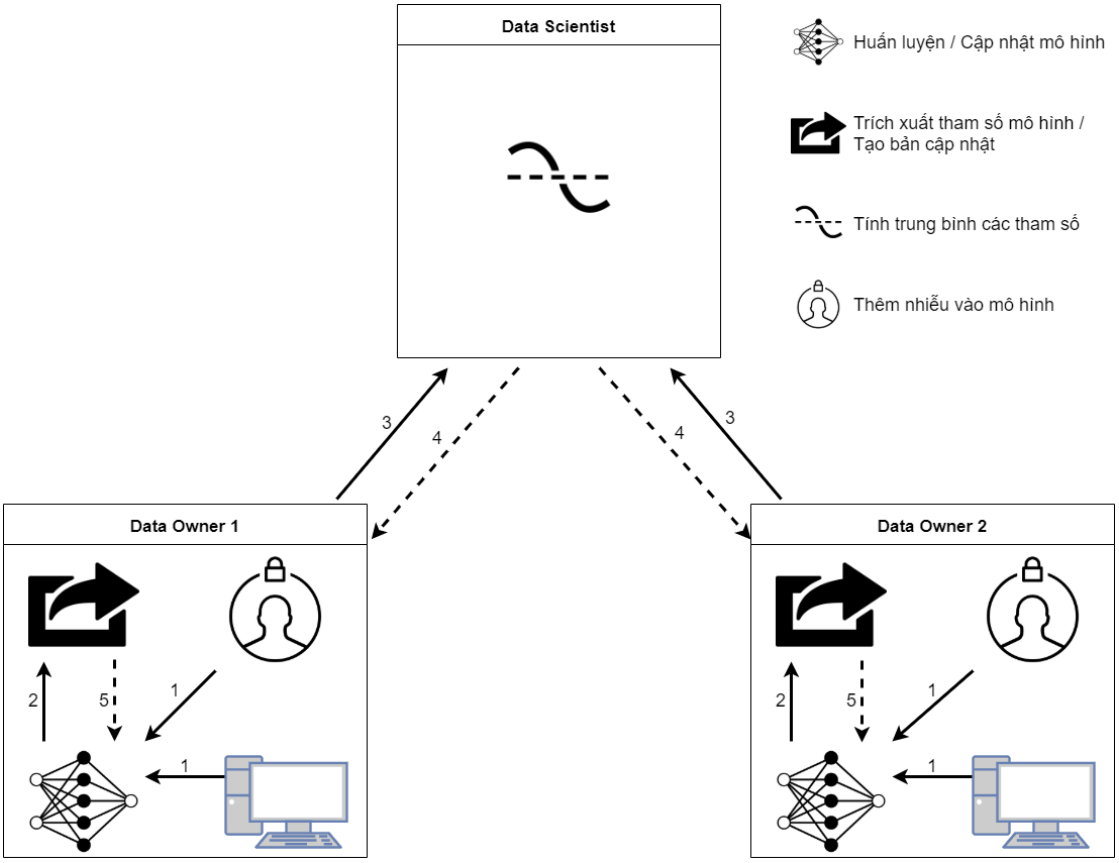
Hình 4.8: Tổng quát mô hình thực nghiệm ứng dụng HE.

Các giai đoạn chính trong mô hình FL ứng dụng HE:

1. Dữ liệu từ các Owner sẽ được dùng để huấn luyện mô hình DL.
2. Data Owner trích xuất các tham số mô hình.
3. Data Owner mã hoá tham số mô hình với context của Scientist.
4. Data Owner gửi các tham số đã được mã hoá đến Scientist để tổng hợp.
5. Data Scientist giải mã bản cập nhật sau khi tính trung bình.
6. Data Scientist mã hoá lại bản cập nhật với context của các Owner.
7. Data Scientist gửi bản cập nhật đã mã hoá về các Owner tương ứng.
8. Data Owner giải mã bản cập nhật và cập nhật vào mô hình DL.

4.2.3.2. Áp dụng kỹ thuật Quyền riêng tư khác biệt (DP) với Opacus

Opacus sẽ được ứng dụng vào quá trình huấn luyện mô hình DL của các Data Owner. Cụ thể hơn, các Owner sẽ tạo ra một PrivacyEngine, sau đó sẽ dung PrivacyEngine vừa tạo để gắn vào hàm tối ưu hoá của mô hình DL và tiến hành huấn luyện mô hình. PrivacyEngine sẽ thay thế quy trình huấn luyện của mô hình thành phương pháp DP-SGD của Opacus, từ đó các tham số mô hình sẽ được thêm nhiễu để đảm bảo rằng các tham số này sẽ không ghi nhớ được các dữ liệu của Owner.



Hình 4.9: Tổng quát mô hình thực nghiệm ứng dụng DP.

Các giai đoạn chính trong mô hình FL ứng dụng DP:

1. PrivacyEngine sẽ được khởi tạo và gán vào hàm tối ưu hoá. Dữ liệu từ các Owner sẽ được dùng để huấn luyện mô hình DL.
2. Data Owner trích xuất các tham số mô hình.
3. Data Owner gửi các tham số đến Scientist để tổng hợp.
4. Data Scientist gửi bản cập nhật về các Owner sau khi tổng hợp xong.
5. Data Owner cập nhật bản cập nhật vào mô hình DL.

5. Tên sản phẩm: SecureFL – Học hợp tác an toàn

6. Hiệu quả, phương thức chuyển giao kết quả nghiên cứu và khả năng áp dụng:

6.1. Thông tin phần cứng và đánh giá kết quả

6.1.1. Thông tin phần cứng

Mô hình thực nghiệm được triển khai trên máy tính được cài đặt Anaconda và Jupyter notebook với thông số cụ thể như sau:

- RAM 32GB
- HDD 500GB
- 8 CPU(s)
- Hệ điều hành Windows 10

Quá trình thực nghiệm sẽ sử dụng các thư viện hỗ trợ:

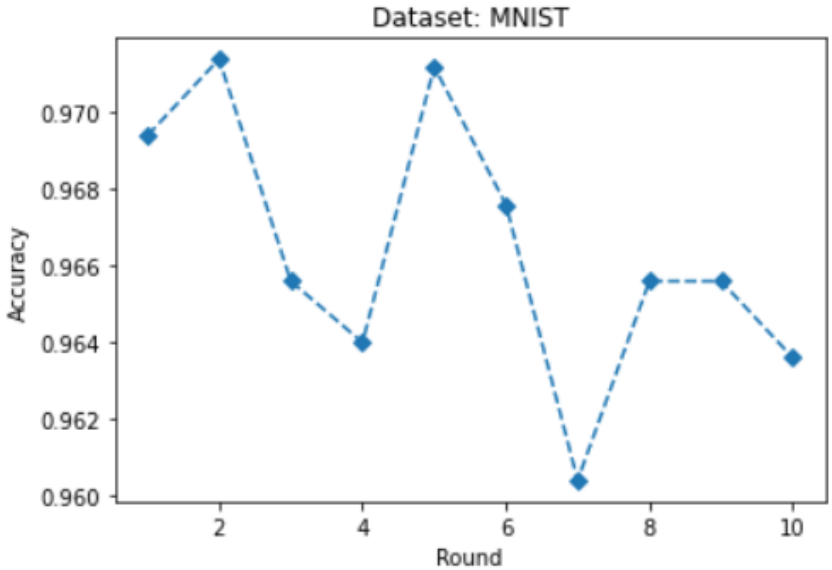
- PyTorch
- PySyft (bao gồm Duet)
- TenSEAL
- Opacus

6.2. Đánh giá kết quả

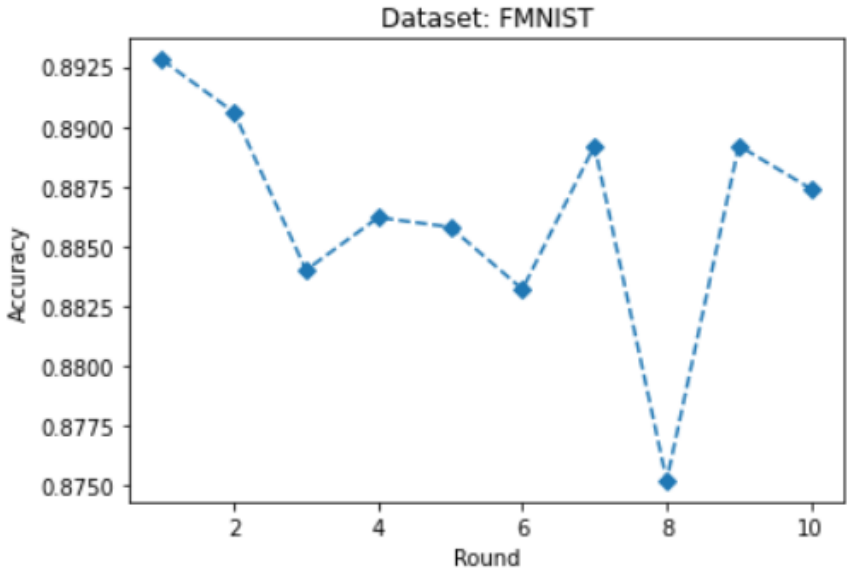
6.2.1. So sánh mô hình học máy phân tán truyền thống với mô hình FL cross-silo

6.2.1.1. Mô hình học máy phân tán truyền thống

Số liệu thu được khi thực nghiệm trên tập dữ liệu MNIST và FMNIST. Mỗi round của mô hình sẽ bao gồm 15 epochs.



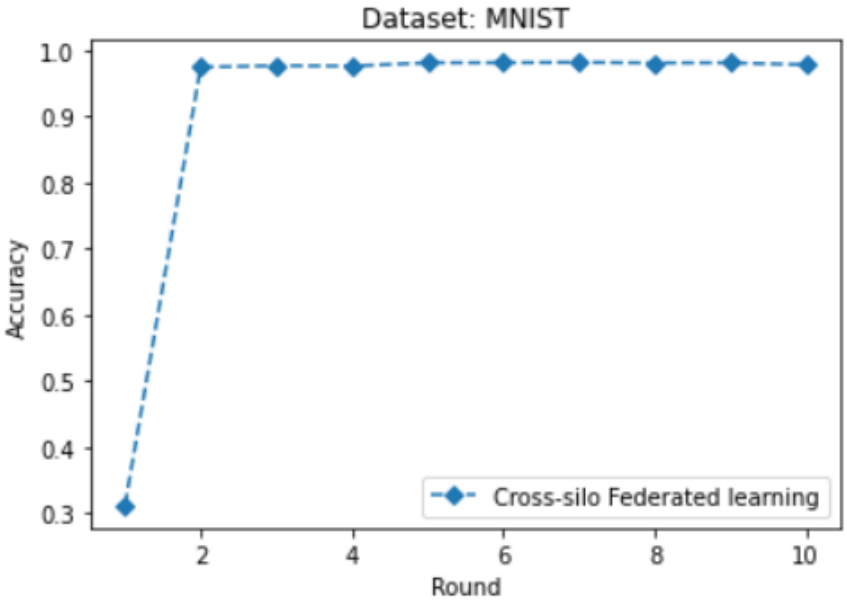
Hình 4.10: Biểu đồ độ chính xác của mô hình học máy phân tán truyền thống sử dụng tập dữ liệu MNIST.



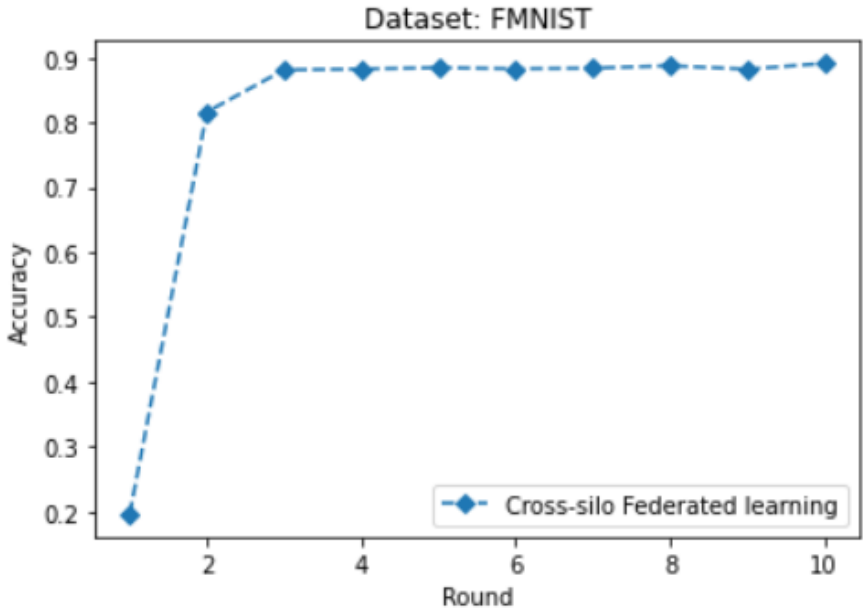
Hình 6.1: Biểu đồ độ chính xác của mô hình học máy phân tán truyền thống sử dụng tập dữ liệu FMNIST.

6.2.1.2. Mô hình FL cross-silo

Số liệu thu được khi thực nghiệm trên 2 tập dữ liệu MNIST và FMNIST. Mô hình FL sẽ sử dụng mô hình DL ANN. Mỗi round của mô hình ANN sẽ bao gồm 15 epochs.



Hình 6.2: Biểu đồ độ chính xác của mô hình FL cross-silo sử dụng tập dữ liệu MNIST.



Hình 6.3: Biểu đồ độ chính xác của mô hình FL cross-silo sử dụng tập dữ liệu FMNIST.

6.2.1.3. Đánh giá

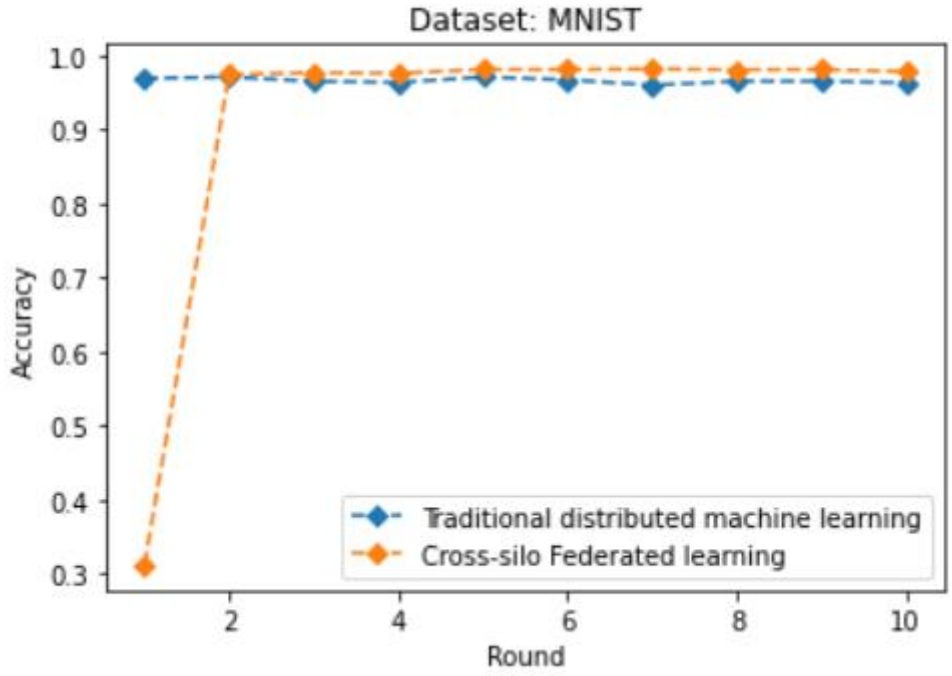
Mô hình học máy phân tán truyền thống

Mô hình học máy phân tán truyền thống mang lại một độ chính xác cao (97.12% sau 5 round huấn luyện) cùng với đó là tốc độ thực hiện mỗi round nhanh (trung bình 1.024 phút cho một round chưa tính thời gian gửi dữ liệu). Điều này cho thấy mô hình DL ở mô hình học máy phân tán truyền thống hội tụ khá nhanh cùng với thời gian thựchiện mỗi round ngắn. Tuy nhiên, mô hình này lại không đảm bảo an toàn được cho dữ liệu người dùng, các dữ liệu người dùng cần được gửi cho máy chủ trung tâm để thực hiện huấn luyện. Điều này cho thấy mô hình học máy phân tán truyền mang lỗ hổng bảo mật nghiêm trọng cần được khắc phục.

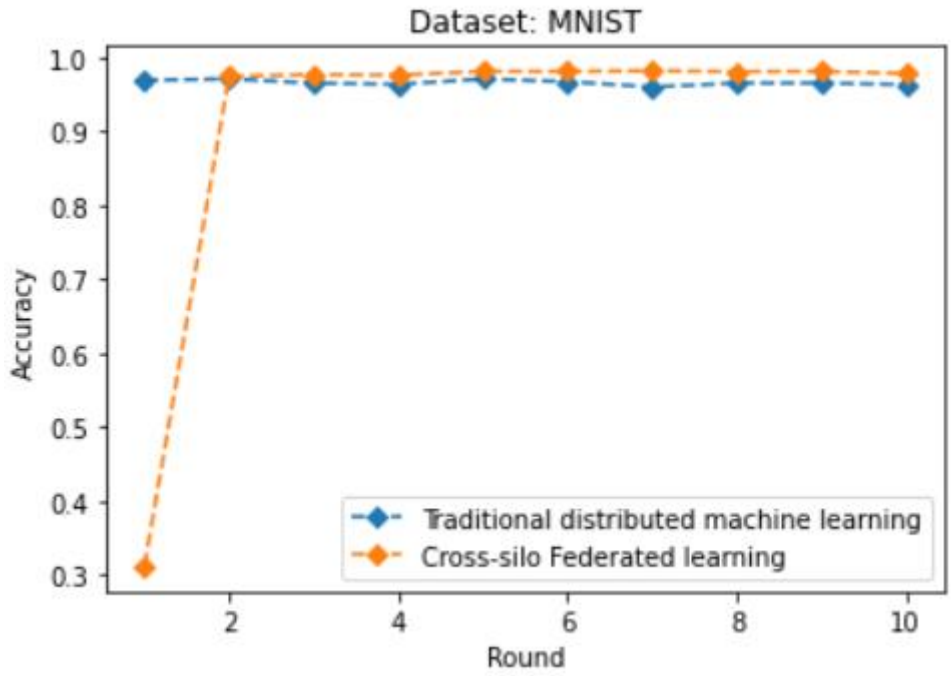
Mô hình FL cross-silo

Mô hình FL cross-silo có độ chính xác mô hình cao (98.15% và 88.48% sau 5 round với tập dữ liệu MNIST và FMNIST), thời gian thực hiện một round tương đối nhanh (trung bình 4.48 phút cho mỗi round trên tập dữ liệu MNIST). Tuy nhiên khác với mô hình học máy phân tán truyền thống, mô hình FL có độ chính xác thấp trong những round đầu nhưng lại tăng dần trong những round kế tiếp. Mặc dù mất nhiều round để đạt được độ chính xác cao nhưng mô hình FL lại mang đến độ chính xác cao hơn so với mô hình học máy phân tán truyền thống khi thực hiện nhiều round tổng hợp hơn. Quá trình huấn luyện của mô hình FL được

diễn ra trên chính các máy Data Owner, dữ liệu người dùng vẫn được giữ nguyên trên máy người dùng sẽ đảm bảo được tính riêng tư của người dùng. Sau khi huấn luyện xong chỉ các tham số mô hình sẽ đượcgửi đến Data Scientist, nhưng các tham số mô hình này vẫn có thể bị suy luận ra các dữ liệu nếu như những kẻ tấn công đối chiếu với các tham số mô hình trước khi huấn luyện. Điều này cho thấy mô hình FL cross-silo vẫn tồn tại lỗ hổng bảo mật cần được giải quyết.



Hình 6.4: Biểu đồ độ chính xác của mô hình học máy phân tán truyền thống so với mô hình FL cross-silo sử dụng tập dữ liệu MNIST.

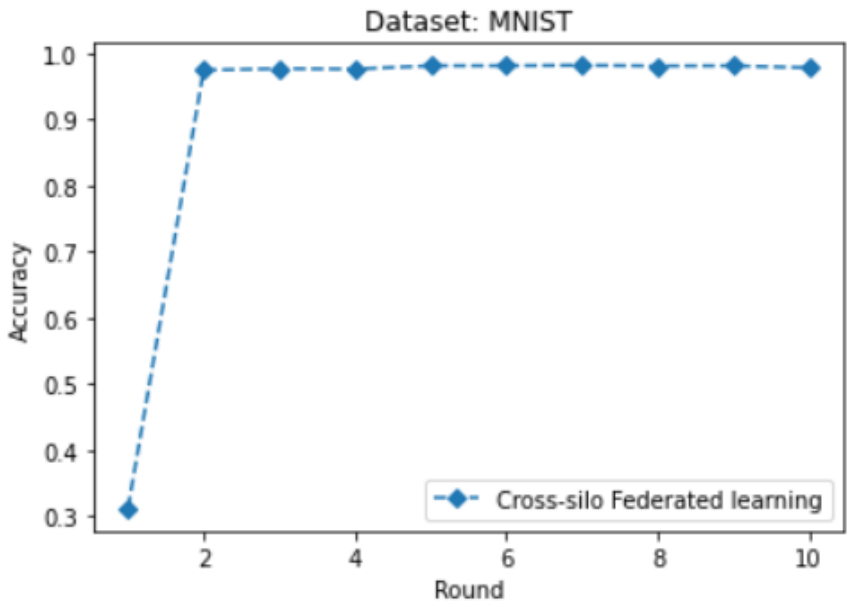


Hình 6.5: Biểu đồ độ chính xác của mô hình học máy phân tán truyền thống so vớimô hình FLcross-silosử dụng tập dữ liệu FMNIST.

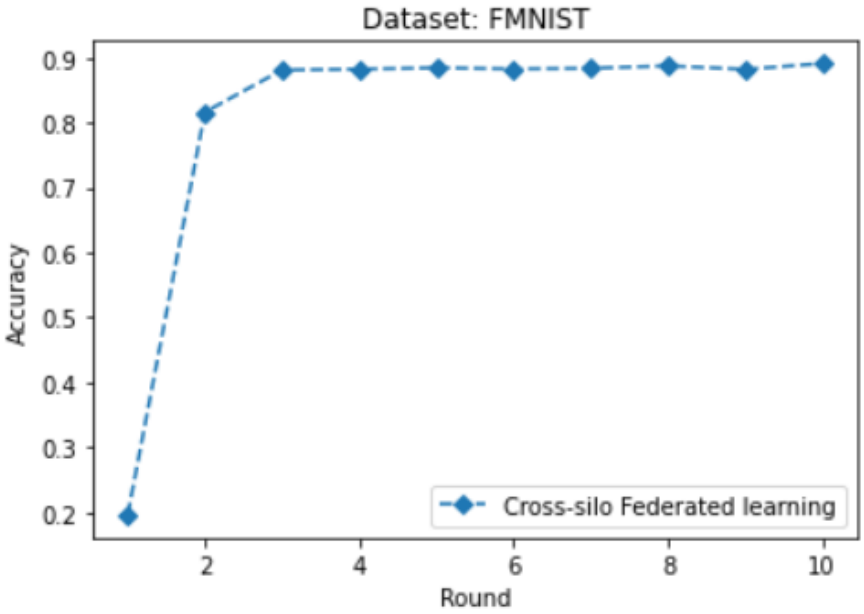
6.2.2. So sánh mô hình FL cross-silo với mô hình mô hình FL cross-silo ứng dụng HE và DP

6.2.1. Mô hình FL cross-silo

Số liệu thu được khi thực nghiệm trên 2 tập dữ liệu MNIST và FMNIST. Mô hình FL sẽ sử dụng mô hình DL ANN và CNN. Mỗi round của mô hình ANN sẽ bao gồm 15 epochs, mô hình CNN sẽ bao gồm 10epochs.



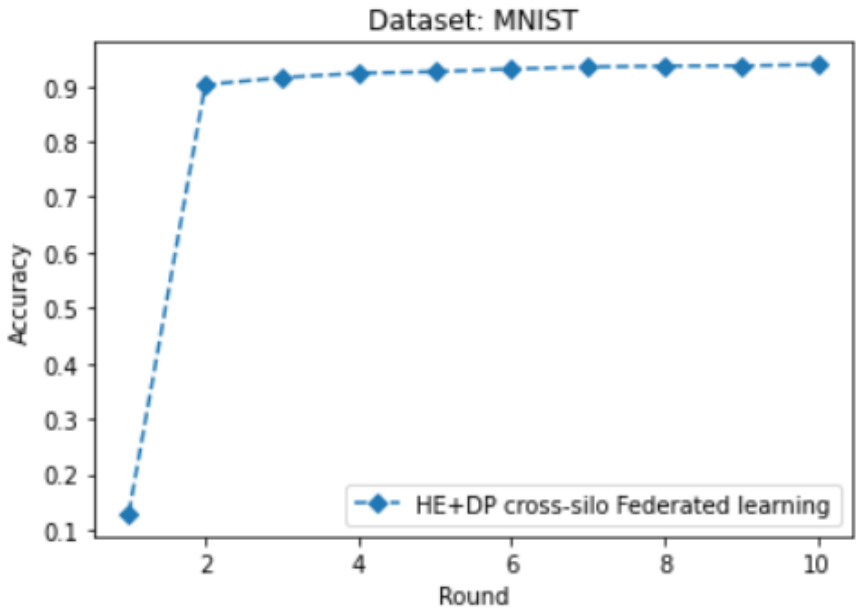
Hình 6.6: Biểu đồ độ chính xác của mô hình FL cross-silo sử dụng tập dữ liệu MNIST.



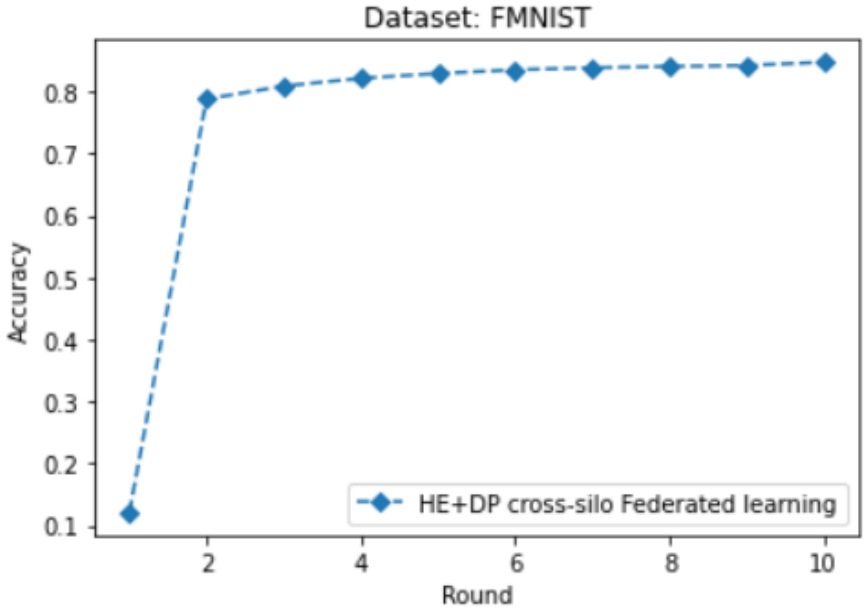
Hình 6.7: Biểu đồ độ chính xác của mô hình FL cross-silo sử dụng tập dữ liệu FMNIST.

6.2.2. Mô hình FL cross-silo ứng dụng HE và DP

Số liệu thu được khi thực nghiệm trên 2 tập dữ liệu MNIST và FMNIST. Mô hình FL sẽ sử dụng mô hình DL ANN. Mỗi round của mô hình sẽ bao gồm 15 epochs.



Hình 6.8: Biểu đồ độ chính xác của mô hình FL cross-silo ứng dụng HE và DP sử dụng tập dữ liệu MNIST.

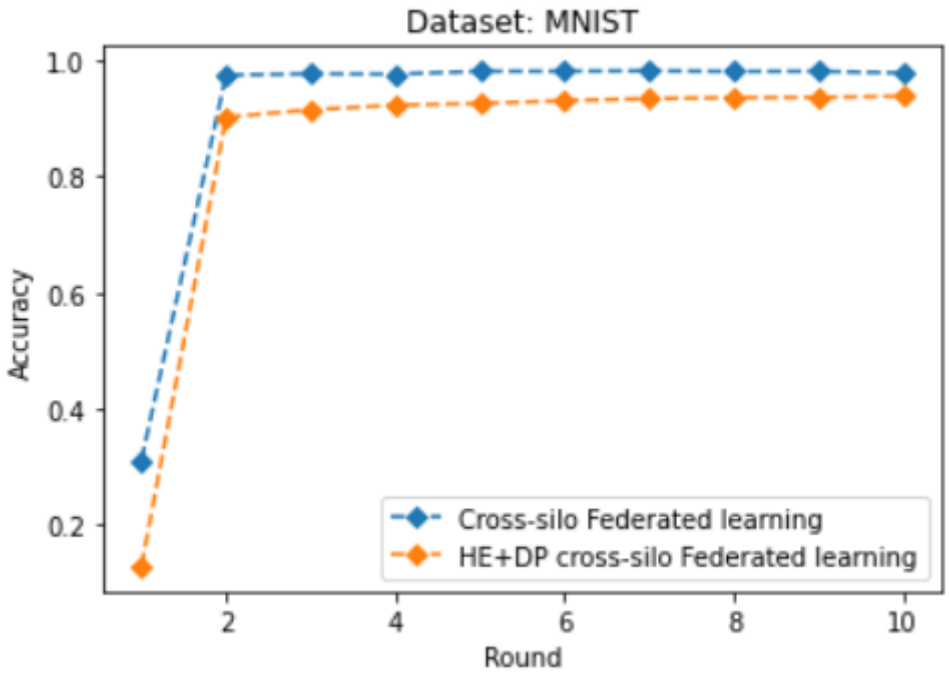


Hình 6.9: Biểu đồ độ chính xác của mô hình FL cross-silo ứng dụng HE và DP sử dụng tập dữ liệu FMNIST.

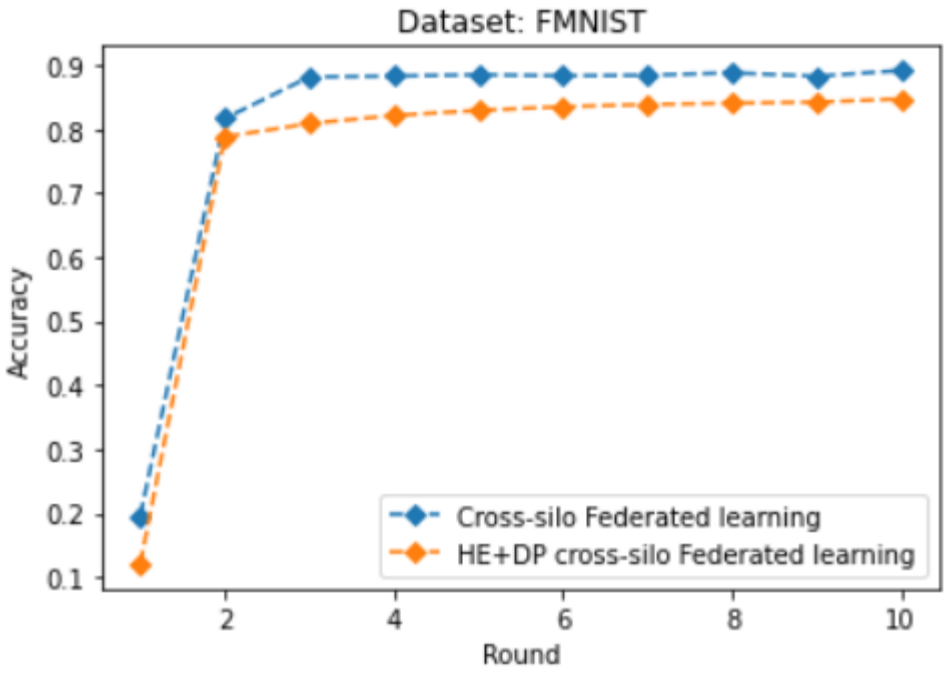
6.2.3. Đánh giá

Các số liệu thu được cho thấy mô hình được ứng dụng kỹ thuật HE và DP có độ chính xác khá cao (kém hơn khoảng 2-3% so với mô hình FL cơ bản, tăng dần khi thực hiện nhiều round tổng hợp, chỉ đạt đến hội tụ khi ngân sách riêng tư tiến về 0) song song đó thời gian của mỗi chu kỳ tổng hợp là tương đối cao (mất trung bình khoảng 5.1 phút cho một chu

kỳ tổng hợp so với mô hình FL cơ bản là 2 phút), thời gian này tập trung chủ yếu tại quá trình trao đổi dữ liệu giữa các thành phần trong hệ thống. Thời gian trao đổi dữ liệu tại mô hình này lớn hơn so với mô hình FL cross-silo (2.84 phút so với mô hình FL cross-silo là 0.92 phút). Thời gian mã hoá mất từ 2-3 giây và mất hơn 1 giây cho quá trình giải mã các tham số mô hình tại Data Owner và Scientist. Việc tính toán trên dữ liệu được mã hoá thông thường sẽ mất nhiều thời gian hơn so với tính toán trên dữ liệu thô. Tuy nhiên bởi vì việc tính trung bình không quá phức tạp cho nên thời gian tính toán chỉ mất 2-3 giây.

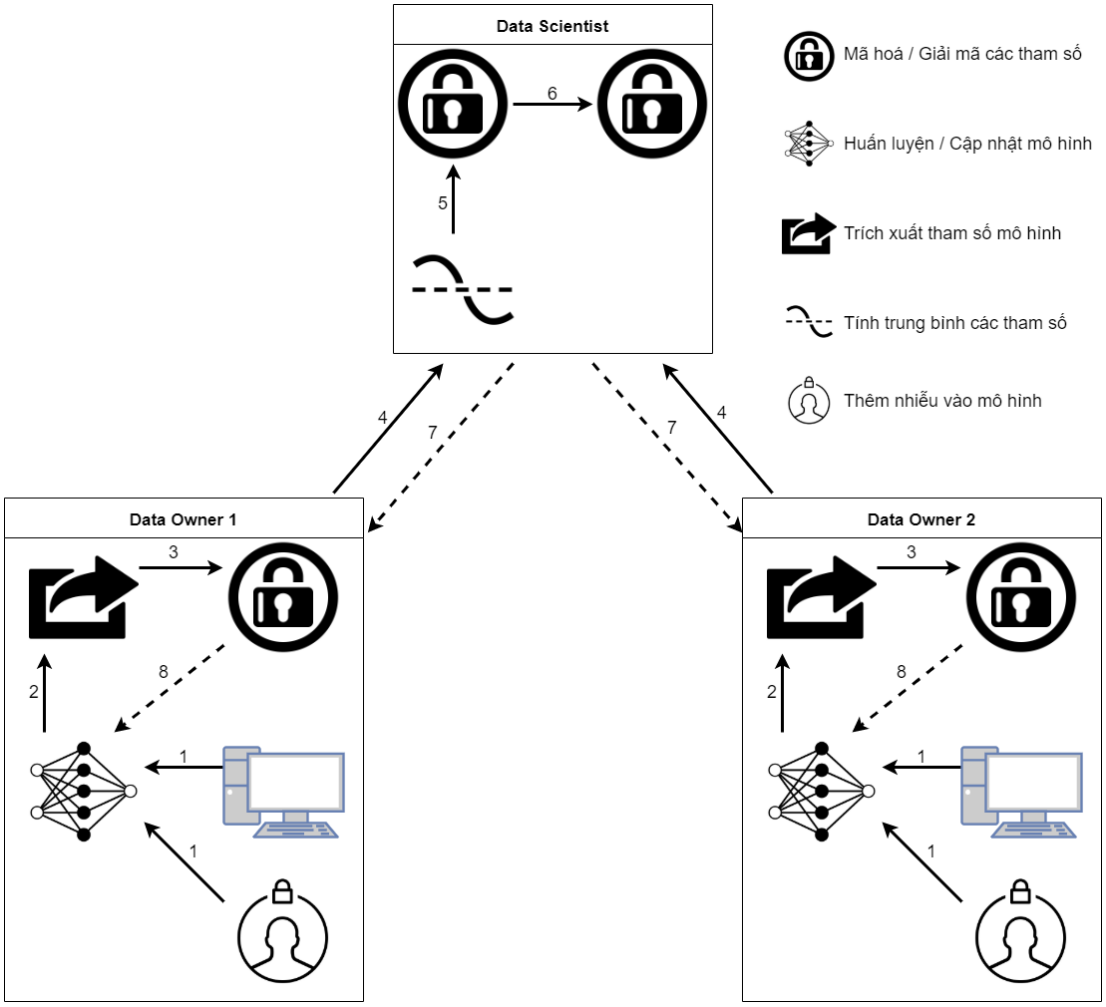


Hình 6.10: Biểu đồ độ chính xác của mô hình FL cross-silo so với mô hình FL cross-silo sử dụng tập dữ liệu MNIST.



Hình 6.11: Biểu đồ độ chính xác của mô hình FL cross-silo so với mô hình FL cross-silo sử dụng tập dữ liệu FMNIST.

7. Hình ảnh, sơ đồ minh họa chính



Hình 7.1: Tổng quát mô hình FL cross-silo ứng dụng HE và DP thực nghiệm.

Cơ quan Chủ trì

(ký, họ và tên, đóng dấu)

Chủ nhiệm đề tài

(ký, họ và tên)

Nguyễn Thái Tài