

PHƯƠNG PHÁP LỰA CHỌN MÁY KHÁCH TỐI ƯU HƯỚNG DẪN DỮ LIỆU CHO HỌC LIÊN KẾT TRONG MẠNG ĐIỆN TOÁN BIÊN DI ĐỘNG

Hoàng Thị Thu

Học viện Công nghệ Bưu chính Viễn thông

Tóm tắt: Học liên kết (Federated Learning: FL) đã nổi lên như một phương pháp đầy hứa hẹn để huấn luyện các mô hình học máy trên các thiết bị phi tập trung, mà không yêu cầu chia sẻ dữ liệu với máy chủ trung tâm. Điều này đặc biệt có ý nghĩa trong các mạng điện toán biên đa truy cập (MEC), nơi quyền riêng tư của dữ liệu là ưu tiên hàng đầu. Tuy nhiên, sự không đồng nhất giữa các thiết bị khách về tài nguyên tính toán và dữ liệu sẵn có đã đặt ra những thách thức lớn trong việc lựa chọn thiết bị hiệu quả trong FL. Trong bài báo này, chúng tôi đề xuất một chiến lược lựa chọn máy khách mới có tên là DDrCS (Lựa chọn máy khách hướng dữ liệu), giúp tối ưu hóa việc sử dụng dữ liệu trong quá trình huấn luyện nhằm cải thiện hiệu năng mô hình đồng thời giảm thiểu chi phí giao tiếp. DDrCS được xây dựng dựa trên giao thức FedCS hiện có, nhưng thay đổi chiến lược lựa chọn máy khách dựa trên lượng dữ liệu tham gia quá trình huấn luyện. Qua các mô phỏng chi tiết, chúng tôi chứng minh rằng DDrCS vượt trội hơn FedCS về hiệu quả sử dụng dữ liệu, giảm thiểu loss function mô hình, và cải thiện độ chính xác, trong khi yêu cầu ít thiết bị tham gia vào mỗi vòng huấn luyện hơn.

Từ khóa: Học liên kết, Điện toán biên, Hướng dữ liệu, Lựa chọn máy khách, Thuật toán tối ưu hoá tổ hợp.

1. MỞ ĐẦU

Sự phát triển nhanh chóng của học máy (ML) đã tác động đáng kể đến nhiều lĩnh vực, yêu cầu phải truy cập vào các tập dữ liệu lớn để huấn luyện mô hình hiệu quả. Sự gia tăng mạnh mẽ của các thiết bị Internet vạn vật (IoT) tạo ra lượng dữ liệu khổng lồ mỗi ngày, trở thành nguồn tài nguyên quan trọng để huấn luyện các mô hình ML. Trong bối cảnh này, Điện toán biên đa truy cập (MEC) nổi lên như một khung mô hình đầy hứa hẹn, cho phép xử lý dữ liệu gần hơn với nguồn. Cách tiếp cận này giúp tăng cường hiệu suất của các ứng dụng ML bằng cách giảm thiểu độ trễ và giảm mức sử dụng băng thông, từ đó hỗ trợ phân tích dữ liệu và ra quyết định theo thời gian thực trong môi trường IoT [1]. Hơn nữa, việc tích hợp các kỹ thuật học liên kết trong MEC cho phép huấn luyện các thuật toán ML ngay trên thiết bị, tối ưu hóa việc sử dụng tài nguyên và cải thiện hiệu suất mô hình mà không làm ảnh hưởng đến quyền

riêng tư của dữ liệu [1]. Sự kết hợp này giữa IoT và MEC đóng vai trò then chốt trong việc khai thác tối đa tiềm năng của ML trong các môi trường động và có giới hạn về tài nguyên [2].

Tuy nhiên, mô hình MEC truyền thống thường dựa trên giả định rằng dữ liệu được tập trung tại một nút máy chủ. Sự tập trung này đặt ra những lo ngại đáng kể về quyền riêng tư, vì thông tin nhạy cảm có thể bị lộ trong quá trình truyền dữ liệu. Những hoạt động như vậy có thể vi phạm quyền riêng tư của người dùng và không được chấp nhận dưới các quy định bảo vệ dữ liệu nghiêm ngặt [3][4]. Do đó, đặt ra nhu cầu cấp bách về các phương pháp thay thế, vừa bảo vệ quyền riêng tư của người dùng vừa tận dụng được các lợi thế của điện toán biên.

Trước thách thức này, học liên kết (FL) ra đời, trở thành một giải pháp khả thi. FL cho phép nhiều thiết bị cùng nhau huấn luyện mô hình học máy bằng dữ liệu cục bộ của chúng mà không cần chia sẻ dữ liệu với máy chủ trung tâm. Thay vào đó, chỉ các bản cập nhật mô hình được truyền đi để tổng hợp, do đó bảo vệ quyền riêng tư và tính bảo mật của dữ liệu [5]. Cách tiếp cận phi tập trung này phù hợp với mô hình MEC, tạo điều kiện thuận lợi cho việc huấn luyện mô hình hiệu quả trong khi vẫn giải quyết các vấn đề quan trọng về quyền riêng tư và bảo vệ dữ liệu trong các mạng di động biên.

Mặc dù FL mở ra một hướng đi đầy hứa hẹn để giải quyết các vấn đề về quyền riêng tư trong các mạng MEC, nhưng nó cũng không tránh khỏi những thách thức. Một trở ngại đáng kể là sự tồn tại của các thiết bị không đồng nhất với năng lực tính toán và giới hạn tài nguyên khác nhau. Nếu các thiết bị khách có tài nguyên tính toán giới hạn, chúng sẽ mất nhiều thời gian hơn để hoàn thành việc cập nhật mô hình. Thêm vào đó, khi các thiết bị hoạt động trong điều kiện kênh truyền không dây kém, thời gian cập nhật sẽ kéo dài hơn. Những vấn đề này sẽ gây ra sự chậm trễ trong bước tổng hợp của máy chủ.

Nhiều bài báo học thuật đã đề cập đến các thách thức của FL trong các mạng MEC, đặc biệt tập trung vào các vấn đề do thiết bị không đồng nhất và điều kiện mạng không dây kém gây ra. Nghiên cứu là [6] đề xuất một cách tiếp cận phân cấp cho FL nhằm quản lý hiệu quả các khả năng khác nhau của thiết bị người dùng, đồng thời giảm thiểu độ trễ và tối ưu hóa phân bổ băng thông. Các tác giả trong [7] thảo luận về các cơ chế nhằm nâng cao hiệu quả năng lượng trong hệ thống FL, giải quyết các hạn chế về tài nguyên và tắc nghẽn truyền thông phát sinh từ tính không đồng nhất

Tác giả liên hệ: Hoàng Thị Thu,

Email: ngvanA@gmail.com

Đến tòa soạn: 08/2024, chỉnh sửa: 09/2024, chấp nhận đăng: 10/2024.

của thiết bị. Những nghiên cứu này cùng đóng góp vào việc hiểu rõ và vượt qua những trở ngại mà FL phải đối mặt trong các môi trường MEC.

Một giải pháp nổi bật mang tên FedCS, cung cấp một giao thức FL mới hướng đến việc tối đa hoá số lượng máy khách tham gia vào quá trình học đã được đề xuất trong [8]. FedCS hoạt động bằng cách khởi tạo một mô hình học máy toàn cục trên máy chủ MEC, sau đó gửi yêu cầu đến một tập các máy khách ngẫu nhiên để thu thập thông tin về tài nguyên hiện có của chúng, bao gồm lượng dữ liệu, trạng thái kết nối mạng và khả năng tính toán. Dựa trên các thông tin này, máy chủ sẽ lựa chọn các máy khách có tiềm năng đóng góp hiệu quả nhất, tức là những thiết bị có khả năng cập nhật mô hình nhanh chóng, để tham gia vào quá trình huấn luyện. Sau khi được chọn, các máy khách này sẽ tải mô hình toàn cục xuống, tiến hành cập nhật nó bằng dữ liệu cục bộ của mình và sau đó tải mô hình đã cập nhật trở lại máy chủ. Máy chủ sẽ tổng hợp các kết quả cập nhật từ nhiều máy khách để cải thiện mô hình chung. Quá trình này được lặp đi lặp lại nhiều lần, kèm theo việc quản lý tài nguyên và thiết lập các thời hạn cụ thể cho từng máy khách nhằm tối ưu hóa hiệu suất của quá trình huấn luyện.

Mặc dù FedCS là một giao thức học phân tán hiệu quả, cho phép quản lý tài nguyên của các máy khách không đồng nhất trong môi trường MEC tuy nhiên nó vẫn còn có những hạn chế nhất định trong việc chưa gom được quá nhiều data sử dụng cho quá trình huấn luyện, cũng như vẫn còn đòi hỏi sự tham gia của nhiều máy khách dẫn đến gánh nặng về giao tiếp. Những hạn chế này sẽ được đề cập kỹ hơn ở mục III sau.

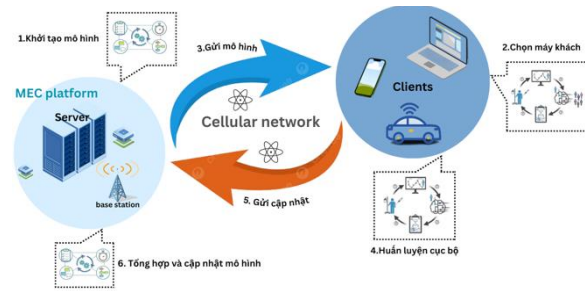
Xuất phát từ những hạn chế của FedCS, chúng tôi đề xuất sử dụng một giao thức mới mang tên DDrCS sử dụng việc lựa chọn máy khách một cách có chiến lược hơn giúp gia tăng lượng data sử dụng trong huấn luyện, từ đó cũng dẫn đến cải thiện hiệu năng của quá trình huấn luyện. Đóng góp chính trong nghiên cứu của chúng tôi bao gồm:

- Phân tích được những điểm hạn chế trong phương pháp cũ FedCS, bao gồm việc không tối ưu được lượng data sử dụng cho training; chưa tối ưu hoá được sự tham gia của các máy khách.
- Đề xuất một giao thức FL mới trên cơ sở của FedCS nhằm cải tiến bước lựa chọn máy khách giúp thực hiện học liên kết hiệu quả hơn khi tối đa hoá số lượng data tham gia vào quá trình huấn luyện ở mỗi vòng.
- Thực hiện những thử nghiệm với nhiều thông số khác nhau để chứng minh tính hiệu quả của phương pháp mới DDrCS.

Phần còn lại của bài báo được trình bày như sau: Mục II giới thiệu về mô hình học tập liên kết và vấn đề lựa chọn máy khách. Mục III trình bày chi tiết đề xuất của chúng tôi về thuật toán chọn máy khách theo hướng dữ liệu. Sau đó, Mục IV mô tả những kết quả mô phỏng thu được khi so sánh với phương pháp cơ sở khác là FedCS.

Lưu ý: Bài báo này được phát triển dựa trên cơ sở của bài báo [8]. Độc giả được khuyến khích tìm hiểu về nghiên cứu kể trên để có một cái nhìn rõ ràng hơn về FedCS và ưu điểm của phương pháp mới DDrCS được đề xuất

II. VẤN ĐỀ LỰA CHỌN MÁY KHÁCH TRONG HỌC LIÊN KẾT



Hình 1: Mô hình học liên kết cơ bản

A. Mô hình học liên kết đơn thuần

Mô hình học liên kết cơ bản được thể hiện như trong Hình 1. Quá trình bắt đầu từ việc khởi tạo mô hình học máy tại máy chủ MEC, sau đó mô hình này được gửi đến các máy khách như điện thoại di động, ô tô thông minh qua mạng di động. Một số thiết bị khách sẽ được chọn để tham gia huấn luyện cục bộ trên dữ liệu riêng của mình. Sau khi hoàn tất huấn luyện, các thiết bị gửi các bản cập nhật mô hình (trọng số đã học) trở lại máy chủ. Máy chủ MEC tiếp nhận các bản cập nhật này, tổng hợp và cải tiến mô hình toàn cục mà không cần thu thập dữ liệu gốc từ các thiết bị. Quá trình này được lặp lại cho đến khi mô hình đạt yêu cầu. Nhờ đó, FL cho phép bảo mật dữ liệu và tối ưu hoá mô hình phân tán trên nhiều thiết bị khác nhau.

B. Tầm quan trọng của việc chọn máy khách phù hợp

Trong học liên kết (Federated Learning - FL), lựa chọn máy khách đóng vai trò quan trọng đối với hiệu quả, tính công bằng và bảo mật của quá trình huấn luyện. FL hoạt động trên dữ liệu phi tập trung với các thiết bị có tài nguyên và khả năng kết nối khác nhau, do đó chọn máy khách có tài nguyên tốt và đường truyền ổn định sẽ giúp tối ưu hóa hiệu quả huấn luyện và giảm thời gian huấn luyện. Ngoài ra, dữ liệu ở các thiết bị là không đồng nhất. Vì vậy chọn máy khách có dữ liệu đa dạng giúp mô hình trở nên toàn diện và không bị thiên kiến. Sự tham gia của các máy khách yếu thế hoặc ít dữ liệu cũng rất quan trọng để đảm bảo công bằng và tính đại diện trong mô hình. Lựa chọn máy khách còn giúp tránh tình trạng một nhóm nhỏ chi phối kết quả học, đồng thời đảm bảo hệ thống mở rộng quy mô dễ dàng khi có hàng nghìn hoặc hàng triệu thiết bị tham gia. Cuối cùng, lựa chọn đúng máy khách giúp tăng cường bảo mật, bảo vệ quyền riêng tư và khả năng chống lại các cuộc tấn công từ các máy khách ác ý, đồng thời tránh sự cố hệ thống liên quan đến hạn chế tài nguyên như pin hoặc bộ nhớ.

Thuật toán 1 FedCS

Đầu vào: K là số máy khách tham gia vào giao thức. $C \in (0, 1]$ là siêu tham số kiểm soát tỉ lệ máy khách tại mỗi vòng lặp.

- 1) **Khởi tạo:** máy chủ trung tâm sẽ khởi tạo mô hình toàn cầu ban đầu (ngẫu nhiên hoặc đã tiền huấn luyện với dữ liệu công khai).
- 2) **Yêu cầu tài nguyên:** máy chủ MEC sẽ gửi cho một tập $[K \times C]$ các thiết bị trong hệ thống và hỏi về các thông số của chúng. Các thiết bị nhận được yêu cầu này sẽ gửi về cho máy chủ thông tin về tài nguyên

mà chúng có (lượng data, tốc độ tính toán, chất lượng đường truyền)

- 3) *Chọn máy khách*: dựa vào các thông tin thu được từ các thiết bị trong hệ thống, máy chủ sẽ chọn ra một tập các thiết bị làm máy khách trong quá trình FL.
- 4) *Phân phối*: máy chủ sẽ gửi mô hình toàn cầu cho các máy khách đã được chọn. Mỗi máy khách sẽ nhận mô hình này để huấn luyện cục bộ trên bộ dữ liệu của họ.
- 5) *Huấn luyện và tải lên*: mỗi máy khách sẽ sử dụng mô hình nhận được và tiến hành huấn luyện trên dữ liệu của chúng một cách độc lập sau đó cập nhật mô hình đã huấn luyện lên cho máy chủ.
- 6) *Tổng hợp*: máy chủ tổng hợp mô hình toàn cục từ các mô hình cục bộ được gửi lên từ các máy khách.
- 7) Tất cả các bước trên trừ bước *Khởi tạo* được thực hiện lặp lại cho đến khi mô hình toàn cục đạt được độ tốt mong muốn hoặc đạt giới hạn về thời gian cho phép.

C. Phương pháp chọn máy khách trong học tập liên kết (FedCS)

Nhận thấy việc lựa chọn máy khách phù hợp có thể mang lại thay đổi lớn trong hiệu suất huấn luyện của FL, các tác giả trong [8] đã đề xuất giao thức FL mang tên FedCS (Giao thức 1) phát triển lên từ giao thức thuần túy của FL.

1) *Giả định*: Như minh họa trong Hình 1, chúng tôi xem xét rằng một nền tảng MEC nhất định, nằm trong một mạng không dây và bao gồm một máy chủ và một trạm gốc (BS), quản lý hành vi của máy chủ và các máy khách trong giao thức FL. Chúng tôi sẽ đặc biệt tập trung vào việc tận dụng các mạng không dây khi chúng ổn định và không bị tắc nghẽn, chẳng hạn như vào nửa đêm hoặc sáng sớm, chủ yếu vì các mô hình học máy (ML) cần được huấn luyện và truyền thông thường có kích thước lớn. Tuy nhiên, mỗi quá trình phải được thực hiện dưới băng thông hạn chế, đặc biệt là khi có nhiều tác vụ ML được thực hiện thông qua FL. Cụ thể, chúng tôi giả định rằng lượng tài nguyên băng thông (RB; đơn vị nhỏ nhất của tài nguyên băng thông được định nghĩa trong LTE [9]) có sẵn cho mỗi quá trình là hạn chế và được quản lý bởi nhà điều hành MEC. Ngoài ra, nếu nhiều máy khách tải lên các tham số mô hình cùng lúc, thông lượng cho mỗi máy khách sẽ giảm tương ứng. Để đơn giản, chúng tôi giả định các máy khách được chọn tải mô hình lên máy chủ một cách lần lượt. Lưu ý rằng, ngay cả khi nhiều máy khách có thể tải lên song song bằng cách chia sẻ các RB, thời gian cần thiết để truyền tất cả các mô hình vẫn giống như khi tải lên tuần tự.

2) *Giao thức FedCS*

Giao thức FedCS là một phương pháp nâng cao trong Học Liên Kết (FL) nhằm tối ưu hóa việc lựa chọn máy khách và quản lý tài nguyên để cải thiện hiệu quả huấn luyện. Giao thức này đặt ra thời hạn cho các máy khách để tải xuống, cập nhật và tải lên các mô hình học máy, cho phép máy chủ tổng hợp càng nhiều cập nhật từ máy khách trong khoảng thời gian giới hạn càng tốt. Hệ thống MEC (Multi-access Edge Computing) đóng vai trò quan trọng trong việc lựa chọn máy khách, đảm bảo rằng quá trình

huấn luyện diễn ra hiệu quả và đúng hạn. Việc lựa chọn này liên quan đến việc giải quyết vấn đề lựa chọn máy khách, xác định các máy khách sẽ tham gia huấn luyện và lên lịch thời gian hoàn thành nhiệm vụ của từng máy khách.

Khác với giao thức thuần túy, nơi máy khách được chọn ngẫu nhiên, FedCS sử dụng một quy trình lựa chọn hai bước. Quy trình này bao gồm một bước *Yêu cầu tài nguyên*, trong đó các máy khách ngẫu nhiên cung cấp cho hệ thống MEC thông tin về trạng thái kênh không dây, khả năng tính toán và kích thước tài nguyên dữ liệu của chúng. Ngoài ra, bước *Phân phối* trong FedCS sử dụng phương thức multicast từ trạm gốc (BS) để phân phối mô hình toàn cầu đến các máy khách đã chọn, giúp tiết kiệm băng thông. Trong giai đoạn *Huấn luyện và tải lên*, các máy khách sẽ cập nhật mô hình song song và sử dụng các khối tài nguyên truyền thông (RBs) được phân bổ để tải lên máy chủ các tham số mới.

III. DDRCS: CHỌN MÁY KHÁCH HƯỚNG DỮ LIỆU

FedCS tuy mang lại hiệu quả rõ rệt so với phương pháp FL đơn thuần [8], nhưng có những hạn chế nhất định. Đầu tiên cần nhắc lại rằng tiêu chí lựa chọn máy khách mà FedCS áp dụng là tối đa hóa số lượng máy khách tham gia vào quá trình học liên kết. Để làm được điều này, thuật toán chọn của FedCS lần lượt chọn những máy khách sao cho thời gian gia tăng trong bước *Huấn luyện và tải lên* là nhỏ nhất. Điều này dẫn đến hai hạn chế:

- Số lượng máy khách tham gia vào quá trình học liên kết nhiều có thể làm gia tăng chi phí. Việc máy chủ phải giao tiếp với nhiều máy khách sẽ làm tiêu tốn thêm các chi phí về truyền thông cũng như làm gia tăng thời gian huấn luyện do nhiều máy khách cùng lúc tải lên mô hình của mình trong khi băng thông có giới hạn.
- Việc chỉ quan tâm đến thời lượng gia tăng trong bước *Huấn luyện và tải lên*, đã vô tình bỏ qua những tiêu chí khác, đặc biệt là lượng dữ liệu tham gia vào quá trình huấn luyện. Cụ thể, việc lựa chọn các máy khách với thời gian cập nhật nhanh có thể đồng nghĩa với việc những máy khách đó có lượng dữ liệu ít. Việc thiếu dữ liệu có thể kéo dài thời gian huấn luyện để có thể đạt được hiệu quả mong muốn.

Để khắc phục, chúng tôi đề xuất phương pháp chọn máy khách hướng dữ liệu (DDRCS) - một giao thức mới với những thay đổi đáng chú ý so với giao thức cũ.

Một số biểu thị toán học cần thiết được diễn giải như sau: Hệ thống có tổng cộng K thiết bị được đánh số index theo tập $K = \{1, 2, \dots, K\}$. K' là tập index của các thiết bị được chọn ra trong hệ thống để gửi thông báo Yêu cầu tài nguyên. $S = \{s_1, s_2, \dots, s_{|S|}\}$ là tập index của các máy khách được chọn tại mỗi vòng lặp với $S \subset K'$. Lí do cần chọn ra tập K' mà không chọn trực tiếp S từ K là vì đặt trong trường hợp số lượng thiết bị trong hệ thống quá lớn, việc gửi thông báo Yêu cầu tài nguyên tới tất cả các thiết bị sẽ tiêu tốn quá nhiều chi phí truyền thông. Thiết bị với index i có lượng dữ liệu là D_i . $D_{total} = \sum_{i=1}^{|S|} D_{s_i}$ biểu thị cho tổng lượng dữ liệu được huấn luyện có được từ tất cả các máy khách được chọn trong tập S . Trong khi đó t_i^{UL} và t_i^{DL} lần lượt là thời gian huấn luyện và thời gian tải mô hình lên máy chủ của thiết bị i . T_{round} là giới hạn thời gian dành cho mỗi vòng huấn luyện.

Các bước trong giao thức được giữ nguyên so với FedCS về tên gọi, thứ tự và ý nghĩa. Tuy nhiên chúng tôi có những thay đổi về thuật toán chọn trong bước Chọn máy khách như sau:

- Thay vì chọn tập máy khách K' một cách ngẫu nhiên, chúng tôi áp dụng một chiến lược lựa chọn có chủ đích hơn. Tại round 1, chọn tập ngẫu nhiên K' từ tất cả các máy khách trong hệ thống. Từ round 2 trở đi, giữ lại 50% các máy khách tốt nhất từ tập K' của round trước. 50% còn lại sẽ lấy ngẫu nhiên từ các client khác trong hệ thống. Chú ý: con số 50 - 50 có thể thay đổi để điều chỉnh tính (kế thừa - khám phá) của thuật toán. Việc này giúp tối ưu hóa quá trình huấn luyện bằng cách đảm bảo rằng các máy khách có khả năng đóng góp nhiều nhất vào chất lượng của mô hình sẽ được giữ lại ở những vòng huấn luyện tiếp theo. Độ tốt của máy khách i được đánh giá bằng thông số sau:

$$G_i = \frac{D_i}{t_i^{UL} + t_i^{UD}} \quad (\text{mẫu dữ liệu/giây}) \quad (1)$$

Chúng tôi lựa chọn thông số độ tốt như vậy với mong muốn chọn ra được những máy khách với khả năng huấn luyện nhiều dữ liệu nhất trong một khoảng thời gian cho trước.

- Thay vì ưu tiên chọn càng nhiều máy khách càng tốt như trong FedCS, chúng tôi tập trung vào việc lựa chọn tập máy khách S có thể đóng góp nhiều dữ liệu nhất cho quá trình huấn luyện. Mục tiêu là tối đa hóa lượng dữ liệu tốt mà mô hình có thể học được, từ đó nâng cao hiệu suất và độ chính xác của mô hình cuối cùng. Chúng tôi sẽ chọn ra tập S từ K' sao cho D_{total} lớn nhất và đồng thời thỏa mãn điều kiện tổng thời gian để các máy khách huấn luyện và tải mô hình lên máy chủ không vượt quá thời hạn giới hạn T_{round} :

$$S = \arg \max_S D_{total} \quad (2)$$

s.t.

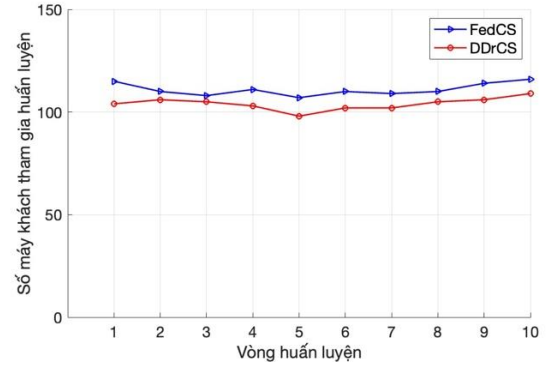
$$T_S^{total} = T_{cs} + T_S^d + \Theta_{|S|} + T_{agg} \leq T_{round}$$

$$\Theta_i = \begin{cases} 0 & \text{nếu } i = 0 \\ T_i^{UD} + T_i^{UL} & \text{còn lại} \end{cases}$$

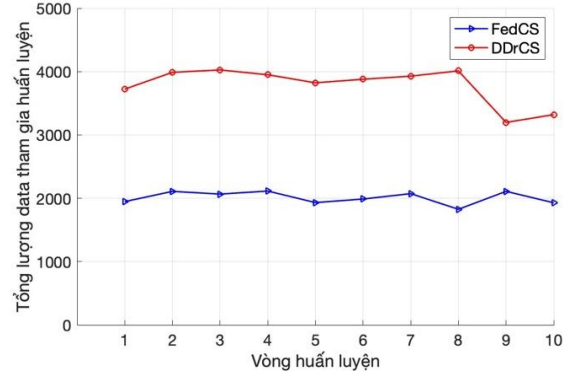
$$T_i^{UD} = \sum_{j=1}^i \max\{0, t_{s_j}^{UD} - \Theta_{j-1}\}$$

$$T_i^{UL} = \sum_{j=1}^i t_{s_j}^{UL}$$

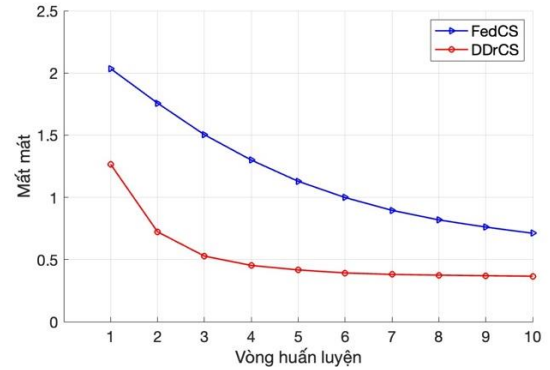
Ở đây, T_{cs} và T_{agg} lần lượt là thời gian cho bước Chọn máy khách và Tổng hợp lần. T_S^d là thời gian để thực hiện bước Phân phối tới các máy khách được chọn. Θ_i là thời gian trôi qua tính từ lúc bắt đầu của bước tải lên và cập nhật cho tới khi máy khách s_i hoàn thành việc huấn luyện và tải mô hình lên máy chủ.



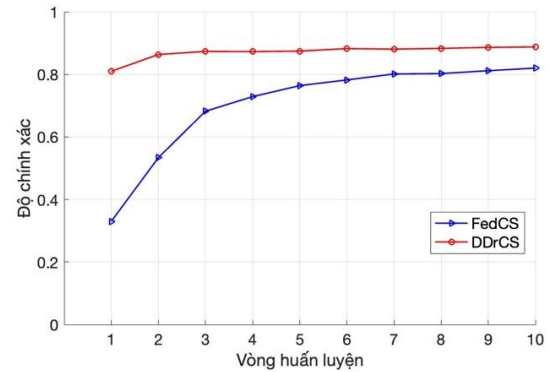
Hình 3. Số lượng máy tham gia huấn luyện theo từng vòng



Hình 4. Số lượng dữ liệu tham gia huấn luyện theo từng vòng



Hình 5. Hàm loss function của mô hình theo từng vòng



Hình 6. Độ chính xác của mô hình theo từng vòng

Việc giải quyết bài toán cực đại (2) là không đơn giản vì nó yêu cầu một quá trình tối ưu hóa tổ hợp phức tạp, trong đó thứ tự của các phần tử trong tập S ảnh hưởng đến $\Theta_{|S|}$. Vì vậy, chúng tôi thiết kế thuật toán chọn tham lam như sau (Thuật toán 2):

Thuật toán 2 Lựa chọn máy khách hướng dữ liệu (DDrCS)**Require:** Tập index K' của các máy khách.

```

1: function CLIENT_SELECTION ( $K'$ )
2:   Initialization :  $S \leftarrow \emptyset, D_{total} \leftarrow 0$ 
3:   while  $|K'| > 0$  do
4:      $x = \operatorname{argmax}_{k \in K'} (D_{total} + D_k) / T_{Suk}^{total}$ 
5:     remove  $x$  from  $K'$ 
6:     if  $T_{Suk}^{total} \leq T_{round}$  then
7:       add  $x$  to  $S$ 
8:     end if
9:   end while
10:  return  $S$ 
11: end function

```

Thuật toán bắt đầu với tập S rỗng, sau đó sẽ đi vào vòng lặp để chọn các máy khách. Tại mỗi vòng, thuật toán chọn ra máy khách nào làm tối ưu hoá nhất tỉ lệ $\frac{D_{total}}{T_S^{total}}$.

IV. KẾT QUẢ MÔ PHỎNG

Ở mục này, chúng tôi đánh hiệu năng của phương pháp lựa chọn máy khách DDrCS so với phương pháp cơ sở là FedCS. Các thông số được sử dụng để đánh giá hiệu năng của các phương pháp bao gồm: (1) Số máy khách tham gia huấn luyện; (2) Lượng dữ liệu tham gia huấn luyện; (3) Hàm loss function của mô hình; và (4) Độ chính xác của mô hình.

Cài đặt mô phỏng: chúng tôi thiết kế những cài đặt cho môi trường mô phỏng như sau: Bộ dữ liệu huấn luyện là MNIST với 42000 mẫu dữ liệu, 10% trong số đó được sử dụng làm tập dữ liệu kiểm thử; Số vòng huấn luyện là 10; mô hình có kích cỡ 6,25 Mbyte; tốc độ huấn luyện mô hình trong mỗi vòng của máy khách dao động trong khoảng từ 1 đến 50 (mẫu dữ liệu/giây); thông lượng của mỗi máy khách trong từng vòng dao động trong khoảng 15 - 50Mbps; số lượng thiết bị trong hệ thống là $K = 2000$; tham số C bằng 0,2.

A. Hiệu quả trong số lượng máy khách tham gia

Hình 3 thể hiện rõ sự biến động về số lượng máy khách tham gia giữa hai mô hình FedCS và DDrCS trong 10 vòng huấn luyện. Cả hai mô hình đều duy trì số lượng máy khách tham gia ổn định trong suốt quá trình huấn luyện, với sự chênh lệch không quá lớn giữa hai mô hình, dao động từ 100 đến 125 máy khách. Tuy nhiên, có thể nhận thấy rằng DDrCS luôn có xu hướng duy trì số lượng máy khách tham gia thấp hơn so với FedCS. Có thể lý giải bởi vì số lượng DDrCS hướng đến việc lựa chọn các máy khách có lượng dữ liệu dồi dào hơn, do đó sẽ làm tăng thêm tổng thời gian huấn luyện. Và khi thời gian cần để mỗi máy khách huấn luyện và tải mô hình lên máy chủ gia tăng, số lượng máy khách tham gia sẽ phải giảm bớt để thỏa mãn giới hạn thời gian T_{round} cho mỗi vòng. Đây là một đặc điểm tốt của DDrCS bởi nó giúp giảm bớt chi phí liên lạc nhờ duy trì ít máy khách hơn trong quá trình học liên kết.

B. Hiệu quả trong tổng lượng dữ liệu huấn luyện

Hình 4 cho thấy sự vượt trội về tổng khối lượng dữ liệu tham gia huấn luyện của DDrCS so với FedCS. Trong suốt quá trình mô phỏng, DDrCS luôn duy trì việc sử dụng khối lượng dữ liệu lớn hơn, với tỷ lệ cao hơn từ 1,5 đến 2 lần so với FedCS. Điều này phản ánh khả năng khai thác và tận dụng tài nguyên dữ liệu hiệu quả hơn của DDrCS. Cụ thể, khối lượng dữ liệu tham gia huấn luyện ở mô hình DDrCS dao động từ khoảng 3200 đến 4000 mẫu, trong khi FedCS

chỉ ở mức xấp xỉ 2000 mẫu. Xu hướng vượt trội này có được là nhờ thuật toán của DDrCS được thiết kế để tối đa hóa việc thu thập và sử dụng dữ liệu trong quá trình huấn luyện, giúp cải thiện hiệu quả huấn luyện tổng thể.

C. Hiệu quả trong mức độ giảm loss function

Hình 5 minh họa rõ ràng sự suy giảm loss function của mô hình qua các vòng huấn luyện cho cả hai phương pháp. DDrCS thể hiện khả năng giảm loss function nhanh chóng và duy trì mức độ thấp hơn so với FedCS trong suốt quá trình. Đáng chú ý, trong nửa đầu của quá trình huấn luyện (vòng 1 - 5), chỉ số loss function của DDrCS chỉ bằng từ 1/3 đến 2/3 so với FedCS, cho thấy một sự vượt trội rõ rệt. Điều này chứng tỏ DDrCS cho tốc độ học nhanh hơn đối với mô hình. Nguyên nhân chủ yếu của sự khác biệt này xuất phát từ lượng dữ liệu tham gia huấn luyện: với khối lượng dữ liệu lớn hơn, DDrCS đã tận dụng được lợi thế để cải thiện hiệu quả huấn luyện một cách vượt bậc.

D. Hiệu quả trong độ chính xác

Hình 6 cho thấy rõ sự tiến bộ vượt bậc về độ chính xác của hai mô hình trong suốt quá trình huấn luyện. Cả DDrCS và FedCS đều có xu hướng cải thiện độ chính xác qua từng vòng, nhưng DDrCS luôn tỏ ra vượt trội với tốc độ cải thiện nhanh hơn và duy trì mức độ chính xác cao hơn trong tất cả các vòng. Đặc biệt, ở vòng cuối cùng, DDrCS đạt độ chính xác ấn tượng khoảng 0.9, trong khi FedCS chỉ chạm ngưỡng gần 0.8. Điều này khẳng định ưu thế của DDrCS trong việc tối ưu hóa mô hình, giúp nó nhanh chóng đạt được kết quả tốt hơn so với FedCS. Nguyên nhân cốt lõi của sự vượt trội này, giống như trong trường hợp chỉ số loss function, chính là nhờ lượng dữ liệu dồi dào hơn mà DDrCS mang lại. Điều này đã đóng vai trò then chốt trong việc nâng cao hiệu quả huấn luyện của mô hình.

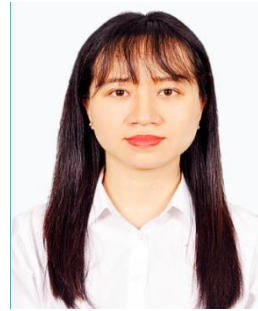
V. KẾT LUẬN

Bài báo này giới thiệu DDrCS, một phương pháp tiếp cận mới hướng dữ liệu trong việc lựa chọn máy khách cho học liên kết trong các mạng MEC. Bằng cách ưu tiên các thiết bị có lượng dữ liệu lớn và tối ưu hóa việc sử dụng tài nguyên, DDrCS đã nâng cao hiệu quả quá trình học mà không tạo gánh nặng cho mạng giao tiếp. Các kết quả mô phỏng của chúng tôi cho thấy DDrCS liên tục vượt trội hơn FedCS, với hiệu quả sử dụng dữ liệu cao hơn, giảm nhanh mô hình loss function và cải thiện độ chính xác qua nhiều vòng huấn luyện. Những kết quả này khẳng định tiềm năng của DDrCS trong việc cải thiện hiệu quả của học liên kết trong các môi trường MEC không đồng nhất.

TÀI LIỆU THAM KHẢO

- [1] M. Ficco, A. Guerriero, E. Milite, F. Palmieri, R. Pietrantuono, and S. Russo, "Federated learning for iot devices: Enhancing tinyml with onboard training," Information Fusion, vol. 104, p. 102189, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1566253523005055>
- [2] Y. Ge, Z. Li, X. Yue, H. Li, Q. Li, and L. Meng, "Tot-based automatic deeplearning model generation and the application on empty-dish recycling robots," Internet of Things, vol. 25, p. 101047, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2542660523003700>
- [3] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive

- survey,” IEEE Communications Surveys Tutorials, vol. 22, no. 3, pp. 2031–2063, 2020.
- [4] Y. Jing, J. Wang, C. Jiang, and Y. Zhan, “Satellite mec with federated learning: Architectures, technologies and challenges,” IEEE Network, vol. 36, no. 5, pp. 106–112, 2022.
- [5] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, “Federated learning of deep networks using model averaging,” CoRR, vol. abs/1602.05629, 2016. [Online]. Available: <http://arxiv.org/abs/1602.05629>
- [6] C. You, K. Guo, H. H. Yang, and T. Q. S. Quek, “Hierarchical personalized federated learning over massive mobile edge computing networks,” IEEE Transactions on Wireless Communications, vol. 22, no. 11, p. 8141–8157, Nov. 2023. [Online]. Available: <http://dx.doi.org/10.1109/TWC.2023.3260141>
- [7] X. Lin, R. Wu, H. Mei, and K. Yang, “A game incentive mechanism for energy efficient federated learning in computing power networks,” Digital Communications and Networks, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2352864823001566>
- [8] T. Nishio and R. Yonetani, “Client selection for federated learning with heterogeneous resources in mobile edge,” CoRR, vol. abs/1804.08333, 2018. [Online]. Available: <http://arxiv.org/abs/1804.08333>
- [9] S. S. I. T. M. Baker, LTE - The UMTS Long Term Evolution: From Theory to Practice. Wiley, 2011.



Hoàng Thị Thu, Tốt nghiệp Đại học-Học viện Công nghệ Bưu chính Viễn thông (2017). Thạc sĩ Học viện Công nghệ Bưu chính Viễn thông (2019). Hiện đang công tác tại Khoa Viễn thông 1, Học viện Công nghệ Bưu chính Viễn thông. Lĩnh vực nghiên cứu: IoT, WSN, Mạng viễn thông, Điện toán biên.

DATA -DRIVEN CLIENT SELECTION METHOD FOR FEDERATED LEARNING IN MULTI-ACCESS EDGE COMPUTING NETWORK

Abstract: Federated Learning (FL) has emerged as a promising method for training machine learning models on decentralized devices, without the need to share data with a central server. This is particularly meaningful in multi-access edge computing (MEC) networks, where data privacy is a top priority. However, the heterogeneity among client devices in terms of computational resources and available data has posed significant challenges in effective device selection in FL. In this paper, we propose a new client selection strategy called DDrCS (Data-Driven Client Selection), which optimizes data utilization during training to improve model performance while minimizing communication costs. DDrCS is built upon the existing FedCS protocol, but modifies the client selection strategy based on the amount of data participating in the training process. Through detailed simulations, we demonstrate that DDrCS outperforms FedCS in terms of data efficiency, reducing model loss, and improving accuracy, while requiring fewer devices to participate in each training round.

Keywords: Federated Learning, MEC, Data-driven, Client Selection, Greedy Algorithm.