Mẫu F6

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:

ĐẢM BẢO QUYỀN RIÊNG TƯ CHO MÔ HÌNH HỌC CỘNG TÁC TRONG HỆ THỐNG PHÁT HIỆN XÂM NHẬP

Tên đề tài tiếng Anh:

Privacy Preservation for Federated Learning in Intrusion Detection System

Khoa/ Bộ môn: Mạng máy tính và Truyền thông

Thời gian thực hiện: 6 tháng

Cán bộ hướng dẫn: ThS. Phan Thế Duy

Tham gia thực hiện

TT	Họ và tên, MSSV	Chịu trách nhiệm	Điện thoại	Email
1.	Huỳnh Nhật Hào, 17520444	Chủ nhiệm	0783984689	17520444@gm.uit.edu.vn
2.	Huỳnh Minh Chủ, 17520293	Tham gia		17520293@gm.uit.edu.vn
3.		Tham gia		

Mẫu F6

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:

ĐẢM BẢO QUYỀN RIÊNG TƯ CHO MÔ HÌNH HỌC CỘNG TÁC TRONG HỆ THỐNG PHÁT HIỆN XÂM NHẬP

Tên đề tài tiếng Anh:

Privacy Preservation for Federated Learning in Intrusion Detection System

Ngày ... tháng ..... năm ....
Cán bộ hướng dẫn
(Họ tên và chữ ký)

Ngày ... tháng ..... năm ....
Sinh viên chủ nhiệm đề tài
(Họ tên và chữ ký)

Mẫu F6

THÔNG TIN KẾT QUẢ NGHIÊN CỨU

1. Thông tin chung:

- Tên đề tài: Đảm bảo quyền riêng tư cho mô hình học cộng tác trong hệ thống phát hiện xâm nhập
- Chủ nhiệm: Huỳnh Nhật Hào
- Thành viên tham gia: Huỳnh Minh Chủ
- Cơ quan chủ trì: Trường Đại học Công nghệ Thông tin.
- Thời gian thực hiện: 6 tháng

2. Mục tiêu:

- Triển khai HE và DP, cũng như tích hợp cả hai cho hệ thống phát triển xâm nhập dựa trên học cộng tác để đảm bảo an toàn và riêng tư cho dữ liệu gốc, đánh giá độ hiệu quả của mô hình huấn luyện khi áp dụng các kỹ thuật HE và DP.

3. Tính mới và sáng tạo:

- Tích hợp cả giải pháp HE và DP trong quá trình học cộng tác trong hệ thống phát hiện xâm nhập để nâng cao hơn tính riêng tư của dữ liệu trong suốt quá trình tổng hợp.

4. Tóm tắt kết quả nghiên cứu:

Mô phỏng thành công hệ thống học cộng tác cho hệ thống phát hiện xâm nhập có sử dụng mã hóa đồng cấu, differential privacy, cũng như kết hợp cả hai để đảm bảo quyền riêng tư cho mô hình học cộng tác và áp dụng và đánh giá 4 mô hình học sâu khác nhau.

5. Tên sản phẩm:

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

Kết quả cho thấy việc áp dụng các kỹ thuật đảm bảo quyền riêng tư mặc dù phải đánh đổi hiệu năng và độ chính xác nhưng vẫn cho một độ hiệu quả nhất định trên các mô hình nhỏ đến mô hình lớn.

7. Hình ảnh, sơ đồ minh họa chính:

Cơ quan Chủ trì *(ký, họ và tên, đóng dấu)*	Chủ nhiệm đề tài *(ký, họ và tên)*

MỤC LỤC

DANH MỤC HÌNH VẼ

DANH MỤC BẢNG

DANH MỤC TỪ VIẾT TẮT

FL	**F**ederated **L**earning
HE	**H**omomorphic **E**ncryption
PHE	**P**artially **H**omomorphic **E**ncryption
SWHE	**S**ome **W**hat **H**omomorphic **E**ncryption
FHE	**F**ully **H**omomorphic **E**ncryption
IDS	**I**ntrusion **D**etetion **S**ystem
DP	**D**ifferential **P**rivacy

DANH MỤC TỪ TẠM DỊCH

Học cộng tác	Federated learning
Mã hóa đồng cấu	Homomorphic encryption
Học máy	Machine learning
Trung tâm dữ liệu	Data center
Tấn công suy luận	Inference attack
Tường lửa	Firewall
Máy chủ	Server
Trọng số	Weight
Văn bản gốc	Plaintext
Bản mã	Ciphertext
Lược đồ	Scheme
Mạng nơ-ron thần kinh	Neural network
Hàm kích hoạt	Activation function
Hàm mất mát	Loss function
Máy chủ tổng hợp	Aggregator
Quyền riêng tư khác biệt	Differential Priva

Chương 1. **TỔNG QUAN**

Tóm tắt

Trong chương này, nhóm chúng tôi xin trình bày tóm tắt về bài toán đảm bảo quyền riêng tư trong hệ thống phát hiện xâm nhập dựa trên học cộng tác và nghiên cứu liên quan, các ứng dụng trong thực tế và các thách thức mà bài toán đang gặp phải. Đồng thời đưa ra mục tiêu và phạm vi nghiên cứu.

1.1. Giới thiệu bài toán

Ngày nay việc xây dựng các mô hình học máy yêu cầu việc thu thập một lượng lớn các dữ liệu huấn luyện từ nhiều nguồn khác nhau. Tuy nhiên hiện nay dữ liệu thường được phân tán và cất giữ cẩn thận trong nhiều tổ chức (ví dụ: ngân hàng, bệnh viện,...), nơi việc chia sẻ dữ liệu hoàn toàn bị cấm do quan ngại về tính riêng tư và bảo mật của dữ liệu.

Trong ngữ cảnh an ninh mạng, nhiều báo cáo ghi nhận ngày càng có nhiều cơ quan, tổ chức liên tục gặp rủi ro về đánh cắp, rò rỉ dữ liệu khi hứng chịu tác động từ các cuộc tấn công, xâm nhập diễn ra dưới nhiều kỹ thuật và hình thức khác nhau [1]. Các hệ thống phát hiện xâm nhập dựa vào dấu hiệu (signature-based) là không hiệu quả khi cố định các qui luật nhận biết tấn công mạng có sẵn. Thay vào đó, các hệ thống phát hiện xâm nhập dựa trên học máy thường được sử dụng để đưa ra một mô hình phát hiện xâm nhập dựa trên bài toán phát hiện dị thường (anomaly-based) có khả năng phát hiện tấn công mới hiệu quả hơn. Để đáp ứng được nhu cầu huấn luyện các mô hình phát hiện xâm nhập, cần có tập dữ liệu lớn và từ nhiều nguồn khác nhau, nhưng các dữ liệu này rất nhạy cảm và các tổ chức thường không mong muốn chia sẻ, tiết lộ ra bên ngoài [2].

Hiện tại, phương pháp học cộng tác cũng đã được áp dụng trong bài toán phát hiện xâm nhập, giúp đảm bảo tính riêng tư của các dữ liệu mạng vốn mang tính nhạy cảm giữa các bên tham gia, chia sẻ thông tin. Tuy nhiên việc trao đổi các tham số mô hình có thể tiết lộ các dữ liệu ban đầu, do đó đặt ra nhu cầu cần bảo vệ quá trình cập nhật mô hình cục bộ lên các mô hình trung tâm để tránh việc dịch ngược suy diễn dữ liệu. Và giống như các mô hình học cộng tác trong ngữ cảnh khác, các hệ thống phát hiện xâm nhập được xây dựng theo cách tiếp cận này cũng gặp rủi ro tương tự như trên.

Chính vì vậy, chúng tôi sẽ tập trung vào việc nghiên cứu phương pháp giúp tổng hợp tham số an toàn, trong đó có áp dụng mã hóa đồng cấu và differential privacy để đảm bảo an toàn và riêng tư cho hệ thống phát hiện xâm nhập dựa trên học cộng tác.

1.1.1. Học máy và tính cần thiết của dữ liệu

Ngày nay, nhiều vấn đề phức tạp xuất hiện trong ngành khoa học máy tính mà ta không thể giải quyết chúng bằng các thuật toán thông thường. Hệ thống nhận diện giọng nói là một vấn đề nổi bật cho ví dụ này: các bản ghi âm thanh cần được phân tích chứa một lượng rất lớn các dữ liệu nhiều chiều, việc hiểu rõ được các dữ liệu này chỉ bằng việc quan sát thông thường gần như là không thể.

Học máy ra đời nhằm cung cấp một cách tiếp cận khác để giải quyết các vấn đề phức tạp như vậy. Học máy chính là một lĩnh vực của trí tuệ nhân tạo liên quan đến việc nghiên cứu và xây dựng các kĩ thuật cho phép các hệ thống "học" tự động từ dữ liệu để giải quyết những vấn đề cụ thể, bằng việc thu thập dữ liệu và áp dụng các kĩ thuật thống kê để tìm ra những khuôn mẫu theo một cách tự động và áp dụng chúng vào vấn đề thực tiễn. Như ở ví dụ về hệ thống nhận diện giọng nói, người ta sẽ thu thập một lượng lớn các bản ghi âm thanh và tìm ra các khuôn mẫu trong đó để giúp cho việc thông dịch các tín hiệu âm thanh.

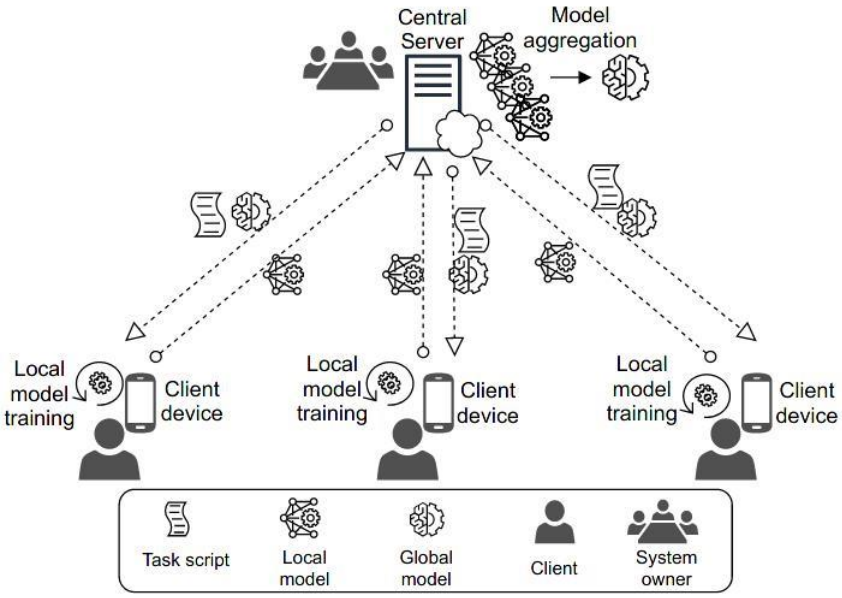
Trong những năm qua, ý tưởng này đã được áp dụng trong nhiều lĩnh vực khác nhau [3]. Ví dụ tiêu biểu như các hệ thống nhận diện giọng nói, hệ thống gợi ý ngày nay hầu hết đều dựa trên học máy, hay các giải pháp dùng để xác định vật thể và dịch tự động cũng đều dựa trên việc học từ dữ liệu.

Nhiều thuật toán liên quan tới học máy đã xuất hiện từ lâu tuy nhiên cho đến những năm gần đây mới được áp dụng một cách thành công và rộng rãi trong lĩnh vực này. Cụ thể, thuật toán lan truyền ngược mà phần lớn mô hình học máy sử dụng đã được mô tả vào đầu những năm 70 của thế kỉ trước [4]. Để giải thích cho việc tại sao những ý tưởng này chỉ được sử dụng một cách thành công ở hiện tại, đã có ba lý do được đưa ra. Đầu tiên, đã có một nhiều sự cải thiện cơ bản với các thuật toán có nhiều ảnh hưởng, một ví dụ điển hình là thuật toán Adam đã giảm đi rất nhiều lượng tinh chỉnh cần có để gradient descent hoạt động tốt [5]. Thứ hai, sự phát triển của phần cứng máy tính đã giúp gia tăng khả năng tính toán: sức mạnh xử lý đã tăng gấp đôi qua từng năm [6] và các phần cứng đặc biệt được chế tạo để dành riêng cho máy học [7].

Tuy nhiên, lí do thứ ba mới là quan trọng nhất, đó là sự gia tăng của các tập dữ liệu để huấn luyện mô hình. Việc có thêm nhiều dữ liệu cũng đồng nghĩa với việc các thuật toán sẽ lấy thêm được nhiều thông tin để quyết định những khuôn mẫu thật sự quan trọng và cần thiết. Kết quả là giảm thiểu khả năng xác định nhầm các khuôn mẫu ngẫu nhiên thành các tín hiệu có ích. Một thí nghiệm ấn tượng đã đưa ra tầm quan trọng của việc có thêm nhiều dữ liệu đã được đưa ra bởi Facebook vào tháng 5 năm 2018 [8]. Bằng việc huấn luyện một mô hình xác định vật thể sử dụng 3.5 tỉ hình ảnh lấy từ Instagram [9] đã cho ra kết quả vượt trội so với tất cả các mô hình khác trên ImageNet - một tiêu chuẩn đánh giá dành cho nhận diện vật thể. Mặc dù những phương pháp được dùng để phân tích dữ liệu hoàn toàn không mới, lượng lớn dữ liệu đã giúp họ xây dựng được một mô hình xác định vật thể tốt nhất. Và sự thật rằng việc có nhiều dữ liệu là cực kỳ quan trọng trong việc xây dựng những mô hình máy học tốt đã được mô tả và thảo luận rộng rãi [10].

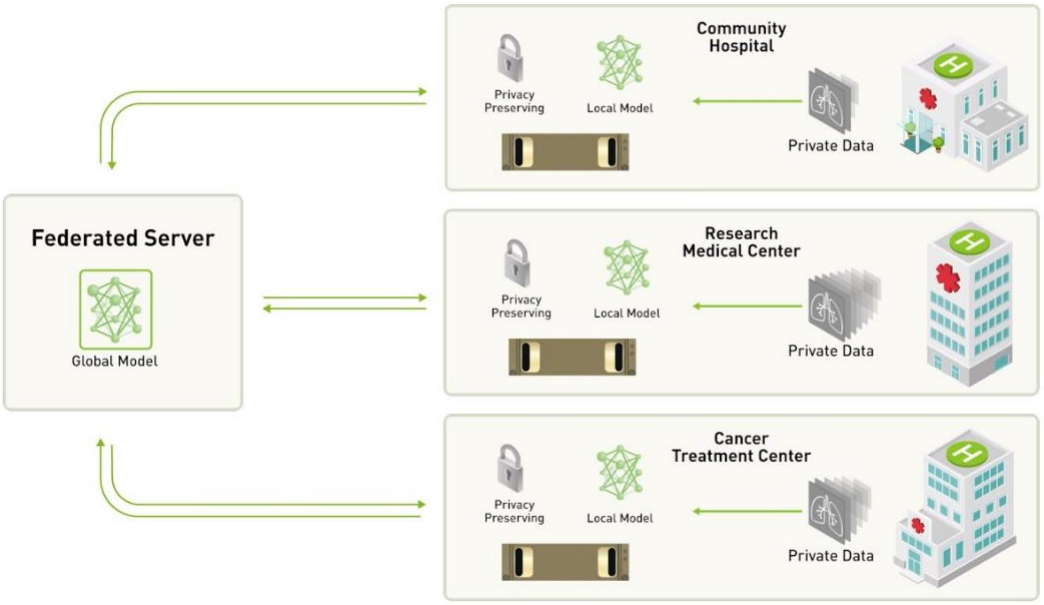
1.1.2. Học cộng tác và đảm bảo quyền riêng tư trong học cộng tác

Cách phổ biến nhất hiện nay khi dùng học máy đòi hỏi việc thu thập tất cả dữ liệu trên một trung tâm dữ liệu và sau đó mô hình máy học sẽ được huấn luyện trên những máy chủ có phần cứng mạnh. Tuy nhiên quá trình thu thập dữ liệu này thường sẽ xâm phạm đến quyền riêng tư của người dùng,thêm vào đó nhiều người dùng không muốn chia sẻ các thông tin của họ cho các công ty dẫn đến việc khó để áp dụng học máy vào những tình huống như vậy (ví dụ: dữ liệu về tình trạng sức khỏe của bệnh nhân). Không chỉ vậy, việc thu thập dữ liệu cũng trở nên bất khả thi, ví dụ như những chiếc xe hơi tự lái sản sinh ra một lượng dữ liệu quá lớn để có thể gởi đến các máy chủ ở trung tâm dữ liệu.



Hình 1.1 Kiến trúc mô hình học cộng tác [11]

Học cộng tác là một phương pháp tiếp cận máy học mà không cần phải thu thập dữ liệu. Học cộng tác (kiến trúc tham khảo hình 1.1) cho phép nhiều thành viên tham gia hợp tác đào tạo một mô hình bằng cách dùng dữ liệu cục bộ của họ và huấn luyện mô hình cục bộ, sau đó trao đổi các tham số của mô hình thay vì trao đổi dữ liệu. Cách tiếp cận này giúp các bên tham gia vẫn đảm bảo được tính riêng tư của dữ liệu (không cần phải upload dữ liệu của họ lên một server tập trung của bên thứ ba) sao cho kết quả đạt được so với cách tiếp cận truyền thống (tập trung dữ liệu về một nơi và tiến hành huấn luyện mô hình) không quá chênh lệch.



Hình 1.2 Một ví dụ về học cộng tác được áp dụng trong y tế

Một ví dụ tiêu biểu của học cộng tác đó là trong các hệ thống y tế như trong 1.2, các bản ghi y khoa là những loại dữ liệu nhạy cảm mà các bệnh viện không thể tiết lộ ra bên ngoài (do vấn đề về bảo mật thông tin bệnh nhân), lúc này các mô hình học cộng tác có thể được sử dụng để liên kết những tập dữ liệu y khoa của các cơ sở y tế lại với nhau để huấn luyện cho một mô hình học sâu liên quan đến y khoa mà vẫn có thể đảm bảo được tính bí mật do các cơ sở y tế không cần phải chia sẻ dữ liệu của mình cho các bên thứ ba.

Mô hình học cộng tác có thể cải thiện nhiều về tính riêng tư của dữ liệu so với mô hình học máy thông thường, tuy nhiên dựa vào các nghiên cứu gần đây học cộng tác vẫn có những nguy cơ về tính riêng tư bởi sự xuất hiện của các cuộc tấn công dịch ngược dữ liệu. Kẻ tấn công có thể tiết lộ một phần dữ liệu huấn luyện chỉ dựa vào các tham số được gởi lên khi tổng hợp. Cụ thể hơn, nhóm tác giả trong [12] đã khai thác được lỗ hổng tiết lộ dữ liệu không chủ ý và thanh công tái tạo lại dữ liệu gốc của các bên tham gia khác thông qua tấn công suy luận (inference attack). Hay trong [13] các bên tham gia có chủ ý xấu sử dụng mô hình toàn cục và tham số để tái cấu trúc lại dữ liệu của các bên tham gia khác. Chính vì vậy, tính riêng tư trong học cộng tác là một chủ đề cần được khai thác nhiều hơn để giảm thiểu rủi ro về tính riêng tư của dữ liệu.

1.1.3. Hệ thống phát hiện xâm nhập dựa trên mô hình học cộng tác

Ngày nay cùng với sự phát triển của Internet, các cuộc tấn công mạng cũng gia tăng theo không chỉ về số lượng mà cả về cách thức: sự trỗi dậy của mã độc tống tiền ransomware, khai thác lỗ hổng zero-day,... Các chương trình anti-virus và tường lửa dần trở nên không đủ hiệu quả để đảm bảo tính an toàn cho hệ thống mạng của một công ty - vốn nên được xây dựng dựa trên nhiều tầng bảo mật. Và một trong những tầng quan trọng, được thiết kế để bảo vệ mục tiêu của nó khỏi các cuộc tấn công tiềm ẩn thông qua một hệ thống theo dõi liên tục, được cung cấp bởi hệ thống phát hiện xâm nhập (IDS).

Các IDS được chia làm 2 loại chính: phát hiện dựa vào dấu hiệu (signature-based) hoặc dựa vào phát hiện dị thường (anomaly-based). Trong các hệ thống phát hiện xâm nhập dựa vào dấu hiệu thì dữ liệu sẽ được theo dõi và so sánh với các khuôn mẫu tấn công có sẵn, nhờ đó phát hiện được các cuộc tấn công tiềm ẩn. Phương pháp này hiệu quả và đáng tin cậy, được sử dụng rộng rãi bởi các ứng dụng như Snort [14] hay Suricata [15], tuy nhiên nó lại chỉ có thể xác định được các cuộc tấn công đã được mô tả trong cơ sở dữ liệu. Hay nói cách khác, phương pháp này không thể xác định được những cuộc tấn công mới ngày càng tinh vi và phức tạp. Chính vì vậy trong những năm qua đã có nhiều hướng nghiên cứu tập trung vào loại IDS còn lại - phát hiện xâm nhập dựa trên phát hiện dị thường.

Mặc dù học máy đã và đang được coi là một trong những phương pháp hiệu quả nhất trong việc bài toán phát hiện dị thường, việc xây dựng các mô hình học máy IDS yêu cầu một lượng rất lớn các dữ liệu huấn luyện từ nhiều nguồn khác nhau. Tuy nhiên trong phần lớn các tổ chức việc chia sẻ dữ liệu với nhau là việc không thể do quan ngại về tính riêng tư và bảo mật của dữ liệu. Chính vì vậy học cộng tác đã được áp dụng trong trường hợp này nhằm cho phép các tổ chức xây dựng một mô hình học máy IDS mà không cần chia sẻ dữ liệu.

Tuy nhiên, như đã nhắc ở phía trên, tính riêng tư của học cộng tác vẫn chưa đưa đảm bảo hoàn toàn do vẫn có rủi ro từ các cuộc tấn công dịch ngược. Để vượt qua vấn đề này, đã có nhiều giải pháp được đưa ra như trong. Trong số đó, mã hóa đồng cấu (homomorphic encryption) và differential privacy (DP) là một trong những giải pháp tiềm năng nhất trong việc đảm bảo tính riêng tư trong học cộng tác. mã hóa đồng cấu là loại mã hóa cho phép ta tính toán trên các dữ liệu đã được mã hóa mà không cần

giải mã chúng trước. DP là kĩ thuật đảm bảo riêng tư cho mỗi mẫu đơn lẻ trong tập dữ liệu bằng cách chèn thêm nhiễu. Trong nghiên cứu này, chúng tôi sẽ tập trung vào việc đảm bảo tính riêng tư cho học cộng tác trong hệ thống phát hiện xâm nhập bằng việc sử dụng mã hóa đồng cấu, differential privacy.

1.2. Các nghiên cứu liên quan

1.2.1. Học cộng tác

Học cộng tác được đề đưa ra bởi Google vào năm 2016 [16] và dần dần thu hút được rất nhiều sự quan tâm trong nghiên cứu và ứng dụng. Học cộng tác đã và đang được sử dụng rộng rãi trong nhiều lĩnh vực: bàn phím Gboard thiết kế bởi Google sử dụng học cộng tác để cải thiện khả năng gợi ý từ mà vẫn bảo vệ được tính riêng tư của người dùng, trong y học các dữ liệu của bệnh nhân rất nhạy cảm nên học cộng tác cũng rất hữu dụng, xử lý ngôn ngữ [17] tự nhiên và hệ thống gợi ý [18] cũng áp dụng học cộng tác.

1.2.2. Các giải pháp đảm bảo riêng tư trong học cộng tác

Các thuật toán hiện đại nhằm gia tăng tính riêng tư trong học cộng tác chủ yếu được chia thành 2 nhóm chính: Secure Multi-party Computation (SMC) và Dif-ferential Privacy (DP).

Khái niệm SMC (hay MPC) lần đầu được giới thiệu để đảm bảo an toàn cho dữ liệu đầu vào các các bên tham gia khi họ cùng tính toán một mô hình [19]. Trong SMC, việc giao tiếp được bảo vệ bởi các giải pháp mã hóa, hiện nay mã hóa đồng cấu là phương pháp được sử dụng nhiều nhất trong SMC. mã hóa đồng cấu cho phép thực hiện một vài phép toán toán trên dữ liệu đã mã hóa mà không cần giải mã chúng trước. Đã có nhiều công trình sử dụng các lược đồ mã hóa đồng cấu, đáng chú ý như lược đồ Paillier [20] để đảm bảo riêng tư trong học cộng tác.

Differential Privacy (DP) cũng là một kĩ thuật bảo vệ riêng tư được sử dụng rộng rãi, dựa trên ý tưởng thêm nhiễu vào các thuộc tính nhạy cảm [21]. Trong học cộng tác, DP được áp dụng bằng cách thêm nhiễu vào các tham số gởi lên của các bên tham gia. DPGAN framework được đề xuất trong [22] đã sử dụng DP để khiến các cuộc tấn công dựa trên GAN kém hiệu quả trong việc suy diễn các dữ liệu huấn luyện của người dùng khác. Ngoài ra, cả hai nghiên cứu trong có sự kết hợp giữa SMC và DP để đạt được một mô hình học cộng tác có độ chính xác cao.

1.2.3. Hệ thống phát hiện xâm nhập dựa trên học công tác

Gần đây đã có nhiều hệ thống phát hiện xâm nhập được xây dựng dựa trên học cộng tác. Cụ thể, năm 2018 Preuveneers và cộng sự đã mô tả học cộng tác dựa trên blockchain được cấp phép để phát triển mô hình học máy phát hiện dị thường trong IDS [23]. Năm 2019, Nguyen [24] thiết kế một hệ thống phân tán tự học tự động để xác định các thiết bị IoT hư hại, dựa trên cách tiếp cận học cộng tác để xác định xâm nhập. Năm 2020, Zhao và cộng sự [25] cũng đề xuất một mô hình học cộng tác trong IDS dựa trên LSTM, đạt được độ chính xác cao. Tuy nhiên các nghiên cứu này và phần lớn các nghiên cứu liên quan vẫn sử dụng các biện pháp để đảm bảo tính riêng tư trong học cộng tác. Riêng trong nghiên cứu [26] nhóm tác giả đã đề xuất framework DeepFed, giải pháp phát hiện xâm nhập dựa trên học cộng tác có sử dụng mã hóa đồng cấu Paillier để bảo vệ quyền riêng tư.

1.3. Tính ứng dụng

Nghiên cứu của chúng tôi có thể áp dụng cho các doanh nghiệp đang có nhu cầu cải thiện độ hiệu quả của IDS trong hệ thống. Các IDS được xây dựng theo cách tiếp cận học cộng tác có thể giúp các doanh nghiệp tránh việc chia sẻ dữ liệu cho các doanh nghiệp khác, ngoài ra quyền riêng tư cũng được đảm bảo hơn rất nhiều khi áp dụng thêm các kĩ thuật như mã hóa đồng cấu và differential privacy.

1.4. Những thách thức

Tuy rằng hai giải pháp chúng tôi sử dụng là HE và DP đều nâng cao tính riêng tư trong học cộng tác, song chúng vẫn có nhược điểm. Việc sử dụng HE tuy không ảnh hưởng nhiều đến độ chính xác của mô hình huấn luyện song lại gia tăng thời gian thí nghiệm, RAM sử dụng cũng như lượng dữ liệu cần trao đổi, còn giải pháp DP tuy nhanh nhưng lại ảnh hưởng lớn đến độ chính xác của mô hình. Điều này đặc biệt đúng khi mô hình càng phức tạp và có nhiều trọng số.

1.5. Mục tiêu, phạm vi nghiên cứu

1.5.1. Mục tiêu

Nghiên cứu của chúng tôi tập trung vào việc xây dựng một hệ thống học cộng tác dành cho IDS có tích hợp các giải pháp đảm bảo quyền riêng tư. Ngoài ra chúng tôi sẽ thực hiện các kịch bản thí nghiệm khác nhau để đánh giá hiệu suất, ưu và nhược điểm của từng giải pháp.

1.5.2. Phạm vi nghiên cứu

Chúng tôi thực hiện xây dựng mô hình học cộng tác dựa trên ngôn ngữ Python và các thư viện như Pytorch, Flask. Để áp dụng các kĩ thuật nâng cao quyền riêng tư cho mô hình học công tác, chúng tôi sử dụng thư viện mã hóa đồng cấu TenSEAL và thư viện hỗ trợ DP Opacus. Nghiên cứu này cũng tiến hành huấn luyện các mô hình IDS để đánh giá hiệu năng của mô hình cộng tác đưa ra kết hợp với từng kĩ thuật như đã nêu trên. Tập dữ liệu được sử dụng trong các mô hình này là CICIDS2017, các mô hình IDS được sử dụng là LSTM, Fully connected network, VGG11 và VGG16.

Chương 2. CƠ SỞ LÝ THUYẾT

Tóm tắt

Trong chương này, nhóm chúng tôi sẽ trình bày các kiến thức nền tảng và cơ sở lý thuyết có liên quan đến đề tài.

2.1. Mã hóa

2.1.1. Tổng quan

Trong mật mã học - một ngành toán học ứng dụng cho công nghệ thông tin, mã hóa là phương pháp để biến thông tin (phim ảnh, văn bản, hình ảnh...) từ định dạng bình thường sang dạng thông tin không thể hiểu được nếu không có phương tiện giải mã. Giải mã là phương pháp để đưa từ dạng thông tin đã được mã hóa về dạng thông tin ban đầu, là quá trình ngược của mã hóa.

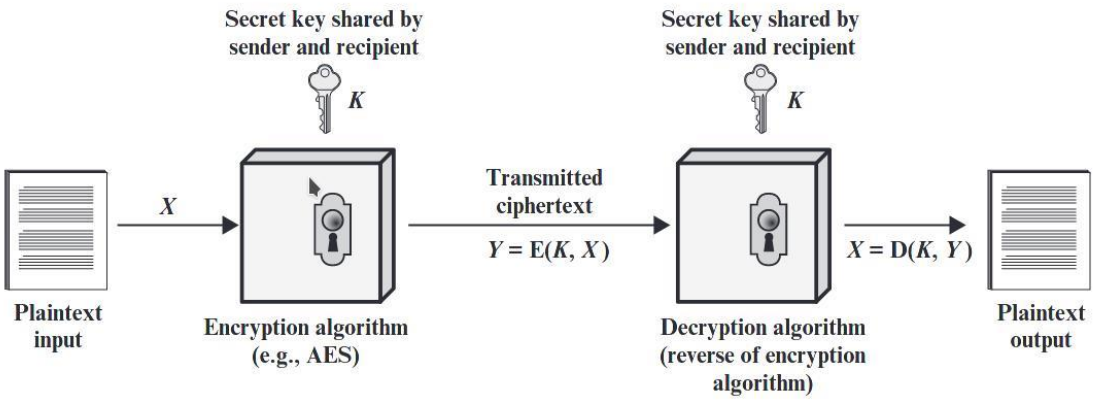
Các khái niệm liên quan đến mã hóa:

- Văn bản gốc (plaintext): thông tin nguyên bản trước khi được mã hóa.
- Bản mã (ciphertext): thông tin sau khi được mã hóa.
- Mã hóa (encryption): quá trình biến đổi từ văn bản gốc sang thành bản mã.
- Giải mã (decryption): quá trình phục hồi văn bản gốc từ bản mã. Có 2 hệ thống mã hóa hiện nay đó là mã hóa đối xứng và mã hóa bất đối xứng.

Có 2 hệ thống mã hóa hiện nay đó là mã hóa đối xứng và mã hóa bất đối xứng.

2.1.2. Mã hóa đối xứng

Mã hóa đối xứng là loại mã hóa mà sử dụng chỉ một khóa giống nhau cho việc mã hóa và giải mã. Một lược đồ mã hóa đối xứng gồm 5 thành phần (2.1):



Hình 2.1 Mô hình tổng quan của mã hóa đối xứng [27]

- **Văn bản gốc**: thông tin nguyên bản hay dữ liệu ban đầu được đưa vào làm đầu vào cho thuật toán mã hóa.
- **Thuật toán mã hóa**: thuật toán mã hóa thực hiện các phép thay thế hoặc chuyển đổi trên plaintext.
- **Khóa bí mật**: khóa bí mật cũng được dùng làm đầu vào cho thuật toán mã hóa. Khóa bí mật là một giá trị độc lập với plaintext và thuật toán. Thuật toán sẽ cho ra kết quả khác nhau tùy vào mỗi khóa được sử dụng (với cùng một plaintext).
- **Bản mã:** thông tin hay dữ liệu sau khi đã được mã hóa, phụ thuộc vào plaintext và khóa bí mật. Cùng một thông tin hay dữ liệu, dùng 2 khóa bí mật khác nhau sẽ cho ra 2 bản mã khác nhau.
- **Thuật toán giải mã**: là ngược lại của thuật toán mã hóa, thuật toán giải mã sẽ nhận bản mã làm đầu vào và cho ra kết quả là dữ liệu hay thông tin ban đầu.

Ví dụ về cách hoạt động của mã hóa đối xứng (tham khảo hình 2.1):

1. Nếu Alice muốn giao tiếp an toàn với Bob thì trước tiên cả hai sẽ thống nhất với nhau một khóa bí mật bằng một cách nào đó (thường là giao thức Diffie-Hellman)

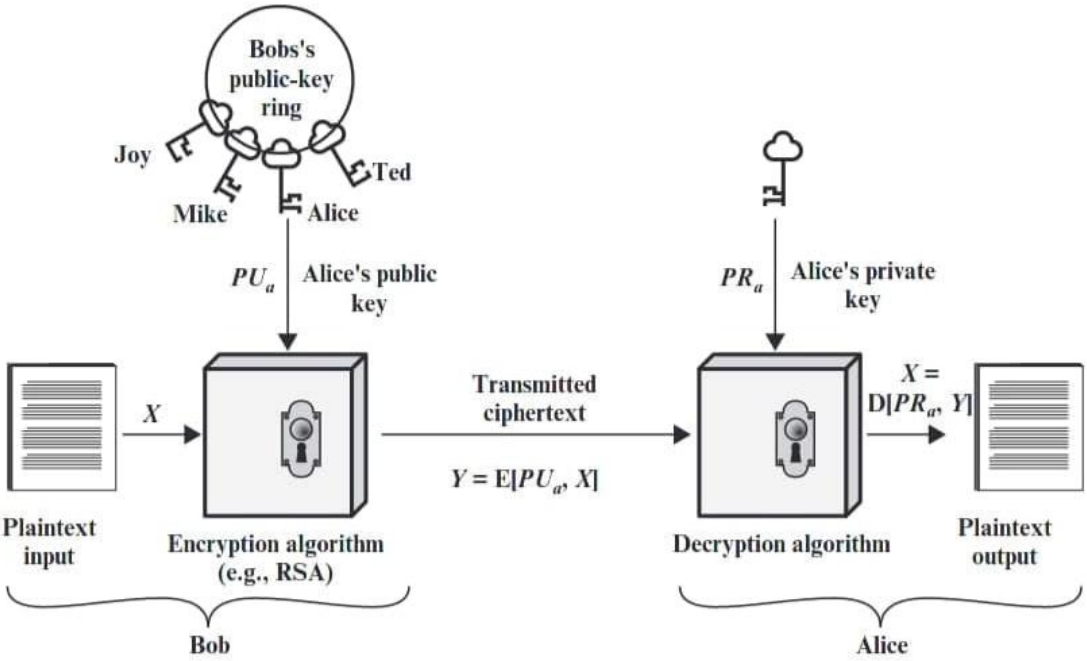
2. Nếu Alice hoặc Bob muốn gởi một thông tin mật đến phía còn lại thì chỉ cần mã hóa thông tin bằng khóa bí mật đã thống nhất

3. Khi bên nhận nhận được thông tin đã mã hóa, chỉ cần dùng khóa bí mật (mà chỉ có người gởi và người nhận biết) để giải mã và thu được thông tin mật ban đầu.

2.1.3. Mã hóa bất đối xứng

Mã hóa khóa bất đối xứng (hay mã hóa công khai) là một dạng mật mã hóa cho phép người sử dụng trao đổi các thông tin mật mà không cần phải trao đổi các khóa chung bí mật trước đó. Điều này được thực hiện bằng cách sử dụng một cặp khóa có quan hệ toán học với nhau là khóa công khai và khóa cá nhân (hay khóa bí mật).

Một lược đồ mã hóa bất đối xứng gồm 6 thành phần (2.2)



Hình 2.2 Mô hình tổng quan của mã hóa bất đối xứng [27]

- **Văn bản gốc:** thông tin nguyên bản hay dữ liệu ban đầu được đưa vào làm đầu vào cho thuật toán mã hóa.
- **Thuật toán giải mã:** thuật toán mã hóa thực hiện các phép thay thế hoặc chuyển đổi trên plaintext
- **Khóa công khai và khóa bí mật:** là một cặp khóa được chọn, một khóa sẽ được dùng để mã hóa, khóa còn lại được dùng để giải mã.
- **Bản mã:** thông tin hay dữ liệu sau khi đã được mã hóa, phụ thuộc vào plaintext và khóa bí mật. Cùng một thông tin hay dữ liệu, dùng 2 khóa bí mật khác nhau sẽ cho ra 2 bản mã khác nhau.
- **Thuật toán giải mã:** là ngược lại của thuật toán mã hóa, thuật toán giải mã sẽ nhận bản mã làm đầu vào và cho ra kết quả là dữ liệu hay thông tin ban đầu.

Ví dụ về cách hoạt động của mã bất đối xứng (tham khảo 2.2):

1. Mỗi người dùng sẽ khởi tạo một cặp khóa để dùng cho việc mã hóa và giải mã.

2. Mỗi người dùng sau đó sẽ đặt một trong hai khóa (ở ví dụ này là khóa công khai) ở một nơi công cộng để bất kì ai cũng có thể truy cập được.

3. Nếu Bob muốn gởi một thông tin mật đến Alice, Bob sẽ mã hóa thông tin bằng khóa công khai của Alice

4. Khi Alice nhận được thông tin đã mã hóa, Alice sẽ dùng khóa riêng tư của mình để giải mã và nhận được thông tin ban đầu. Không ai có thể giải mã thông tin ngoại trừ Alice.

2.2. Mã hóa đồng cấu

Trong mục này nhóm chúng tôi sẽ trình bày về các khái niệm căn bản về mã hóa đồng cấu (Homomorphic Encryption) và sự phát triển của chúng.

Trong kỷ nguyên "điện toán đám mây" (cloud computing) ngày nay, nhiều dữ liệu của các công ty, doanh nghiệp lớn được lưu trữ và tính toán bởi một bên thứ ba như Google, Microsoft, Apple, Amazon, Facebook, Dropbox,... Mã hóa thông thường cung cấp các giải pháp để bảo vệ dữ liệu khi di chuyển từ điểm A sang điểm B, nhưng các giải pháp này không đủ để đảm bảo dữ liệu khi được lưu trữ và khi được sử dụng.

Ví dụ, giả sử Alice có một vài dữ liệu $\mathbf{2}$ $\{0, 1\}^n$ (trong các ứng dụng ngày nay x thường có độ dài vài terabytes hoặc lớn hơn) và muốn lưu trữ dữ liệu này trên dịch vụ đám mây của Bob nhưng Alice lại lo ngại Bob sẽ bị tấn công hay chỉ đơn giản không tin tưởng Bob. Mã hóa thông thường không hoàn toàn giải quyết được vấn đề này: Alice có thể lưu trữ dữ liệu đã được mã hóa ở Bob và giữ lại khóa bí mật, tuy nhiên vấn đề nảy sinh khi Alice muốn làm gì đó với dữ liệu chẳng hạn như thực hiện hàm tính toán f ( ) ngay trên nơi lưu trữ thì Alice phải chia sẻ khóa bí mật với Bob, do đó vi phạm với mục đích mã hóa ban đầu. Sau sự cố hệ thống máy tính của văn phòng quản lý nhân sự Hoa Kỳ (Office of Personell Management) được phát hiện đã bị tấn công vào tháng 6 năm 2015 và làm tiết lộ nhiều thông tin nhạy cảm của khoảng 18 triệu người, chuyên gia an ninh mạng Andy Ozment đã cho rằng mã hóa thông thường cũng sẽ không giúp ngăn chặn được vụ việc [28] bởi vì "nếu kẻ xâm nhập có được thông tin xác thực của một người dùng trong hệ thống mạng thì kẻ đó sẽ có thể truy cập được vào dữ liệu kể cả khi nó được mã hóa, cũng giống như việc những người dùng trong hệ thống mạng truy cập vào dữ liệu". Vậy thì, liệu chúng ta có thể mã hóa dữ liệu theo một cách mà vẫn cho phép một vài truy cập và tính toán ngay trên đó ? Lời giải đáp cho câu hỏi trên vốn đã xuất hiện vào năm 1978 khi Rivest, Adleman, và Dertouzous đã đưa ra ý tưởng sử dụng mã hóa đồng cấu để thực hiện một số phép tính toán trên dữ liệu đã mã hóa [29]. Ý tưởng này đã truyền cảm hứng cho nhiều nhà nghiên cứu khác để tạo ra các lược đồ đồng cấu (homomorphic scheme) hỗ trợ nhiều phép tính toán.

2.2.1. Tổng quan

Mã hóa đồng cấu là một loại đặc biệt của mã hóa, có khả năng thực thi các phép toán trên dữ liệu đã mã hóa và cho ra kết quả giống như khi thực hiện phép toán trên dữ liệu ban đầu. Kết quả cho ra đã được mã hóa.

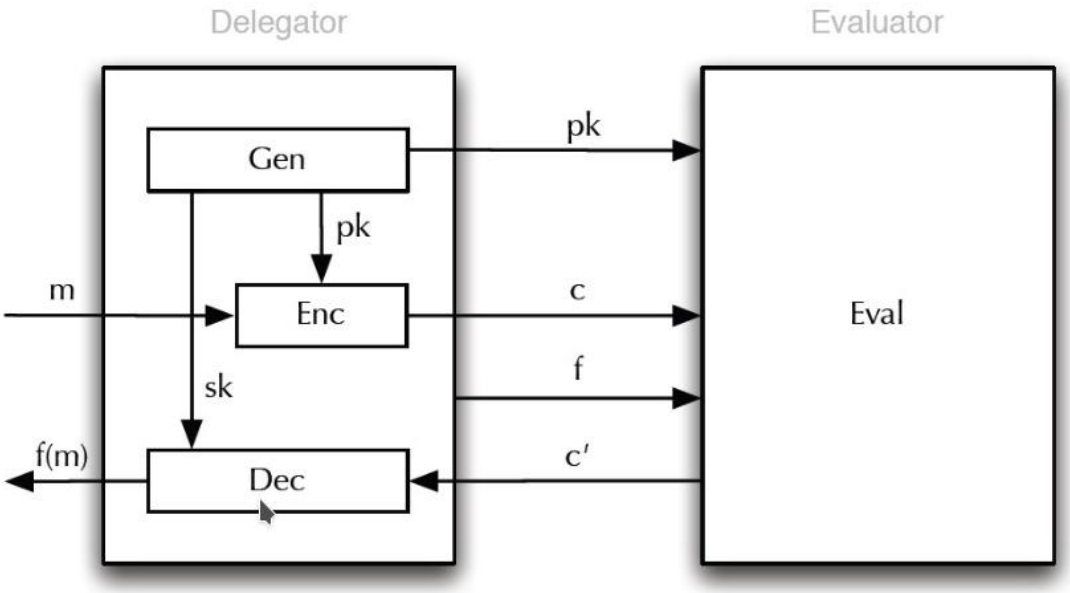
Định nghĩa: Một lược đồ mã hóa đồng cấu với thuật toán E qua một phép '*' hỗ trợ phương trình sau:

$$E(m1) * E(m2) = E(m1 * m2), \forall m1, m2 \in M,$$

với M là tập dữ liệu lớn (chứa toàn bộ thông tin cần mã hóa). [30]

Lược đồ HE có 4 thuật toán chính (tham khảo 2.3):

- Thuật toán sinh khóa - KeyGen: đầu vào là một tham số bảo mật, đối với lược đồ HE bất đối xứng đầu ra là một cặp khóa bí mật - công khai còn đối với lược đồ HE đối xứng là một khóa độc nhất.
- Thuật toán mã hóa - Enc: đầu vào là dữ liệu cần mã hóa m từ tập dữ liệu M với khóa để mã hóa để cho ra dữ liệu đã mã hóa c = E(m).
- Thuật toán giải mã - Dec: đầu vào là dữ liệu đã mã hóa c với khóa để giải mã để cho ra dữ liệu ban đầu D(c) = m.
- Thuật toán đánh giá - Eval: nhận dữ liệu đã mã hóa là đầu vào (c1, c2) và thực hiện hàm f (...) trên dữ liệu đó để cho ra dữ liệu đã được đánh giá f(c1, c2) = E(f (m1, m2)) mà không cần biết dữ liệu ban đầu (m1, m2), hay nói cách khác D(f (c1, c2)) = f (m1, m2).



Hình 2.3 Lược đồ mã hóa đồng cấu [30]

Hiện nay có ba loại lược đồ mã hóa đồng cấu (HE scheme) khác nhau dựa vào các phép tính toán và số lượng các phép tính toán có thể thực hiện:

- Partially Homomorphic Encryption (PHE): chỉ hỗ trợ những phép tính của một loại (phép nhân hoặc phép cộng) với số lần hạn chế.
- Somewhat Homomorphic Encryption (SWHE): hỗ trợ các phép tính hạn chế (ví dụ: cộng hoặc nhân) lên đến một độ phức tạp nhất định, nhưng các phép tính này chỉ có thể được thực hiện một số lần nhất định.
- Fully Homomorphic Encryption (FHE): hỗ trợ bất cứ phép tính nào với số lần không hạn chế.

2.2.2. Partially Homomorphic Encryption

Partially Homomorphic Encryption là lược đồ mã hóa đồng cấu chỉ hỗ trợ các phép cộng hoặc các phép nhân trên dữ liệu đã mã hóa. Ở đây nhóm chúng tôi xin giới thiệu 2 ví dụ phổ biến trong PHE đó là RSA và Paillier.

RSA

Mã hóa RSA chính là loại mã hóa bất đối xứng như đã đề cập ở trước, được Rivest, Shamir và Adleman giới thiệu vào năm 1978. Ngay sau đó, thuộc tính đồng cấu của loại mã hóa này cũng được giới thiệu bởi Rivest, Adleman và Dertouzous với cái tên đồng cấu riêng tư (privacy homomorphism), hay cũng chính là tiền thân của mã hóa

đồng cấu một phần như hiện nay. Lược đồ mã hóa RSA bao gồm 4 thuật toán như sau:

- Thuật toán sinh khóa: Khóa công khai là cặp số nguyên *(n, e)* với *n = pq, p, q* là 2 số nguyên tố lớn, chọn số e sao cho *gcd(e, φ(n))* = 1 (gcd: ước số chung lớn nhất) với *φ(n) = (p 1)(q 1)*. Khóa bí mật sẽ là *(d, n)* với d chính là nghịch đảo (inverse) của *e* (hay *ed* 1(mod *φ*(*n*))).
- Thuật toán mã hóa: đầu tiên dữ liệu sẽ được chuyển thành plaintext m **2** Zn, sau đó dữ liệu được mã hóa ciphertext c sẽ được tính toán như sau:

  $$E(m) = m^e \pmod{n} = c,$$

  với c **2** Z_n.
- Thuật toán giải mã: nhận đầu vào là khóa bí mật *(d, n)* và ciphertext *c* để giải mã:

  $$D(c) = c^d \pmod{n} = m$$
- Thuộc tính đồng cấu: Với m1, m2 **2** Z_n,

$$E(m_1) * E(m_2) = (m_1^e \pmod{n}) * (m_2^e \pmod{n}) \quad (2.1)$$
$$= (m_1 * m_2)^e \pmod{n} = E(m_1 * m_2)$$

Từ 2.1 có thể thấy tính chất nhân đồng cấu (multiplication homomorphic) của RSA có thể tính được $E(m_1 * m_2)$ trực tiếp từ $E(m_1)$ và $E(m_2)$ mà không cần giải mã chúng trước.

Paillier

Lược đồ mã hóa Paillier được tạo ra bởi Pascal Paillier vào năm 1999 [20], dựa trên bài toán kiểm tra một số nguyên x thỏa $x^n \equiv a \bmod n^2$. Lược đồ mã hóa Paillier bao gồm 4 thuật toán như sau:

- Thuật toán sinh khóa:
  1. Chọn 2 số nguyên tố lớn p, q sao cho gcd(pq, (p − 1)(q − 1)) = 1 (gcd: ước số chung lớn nhất).
  2. Tính n = pq và λ = lcm(p − 1, q − 1) (lcm: bội số chung nhỏ nhất).
  3. Chọn ngẫu nhiên $g \in Z^*_{n^2}$ sao cho gcd(n, L(g^{λ} ((mod n^2)))) = 1 (với L(μ) = (μ − 1)/n; ∀μ ∈ $Z^*_{n^2}$.
  4. Kết quả cho ra là khóa công khai (n, g) và khóa bí mật (p, q)
- Thuật toán mã hóa: Để mã hóa thông tin m ∈ Zn, chọn r ngẫu nhiên với r ∈ Z^*_n, ciphertext sẽ được tính toán như sau

$$E(m) = g^m r^n \pmod{n^2} = c$$

- Thuật toán giải mã: Để giải mã $c \in Z^*{}_n{}^2$, plaintext sẽ được tính toán như sau

$$D(c) = \frac{L(c^\lambda \pmod{n^2})}{L(g^\lambda \pmod{n^2})} \pmod{n} = m$$

- Thuộc tính đồng cấu:

$$E(m1) * E(m2) = (g^{m1} r^n{}_1 \pmod{n^2}) * (g^{m2} r^n{}_2 \pmod{n^2}) = g^{m1+m2} (r_1 * r_2)^n \pmod{n^2} = E(m_1 + m_2) \quad (2.2)$$

2.2.3. Somewhat Homomorphic Encryption

Somewhat HE (SWHE) là loại mã hóa đồng cấu có thể thực hiện cả các phép nhân và cộng nhưng với số lần giới hạn nhất định. Số lần giới hạn này được định nghĩa bởi khả năng của lược đồ để giải mã bản mã gắn liền với các phép toán đồng cấu một cách chính xác.

Một cách tổng quan thì bản mã của lược đồ có một tham số gây nhiễu, để giải mã được nó một cách chính xác thì tham số gây nhiễu này phải thấp hơn một giới hạn nhất định. Một lược đồ SWHE có thể thực hiện cả các phép nhân và cộng trên dữ liệu đã mã hóa nhưng sẽ gia tăng nhiễu trong bản mã sau mỗi phép tính toán. Chính vì vậy để giữ tham số gây nhiễu nhỏ hết mức có thể, lược đồ SWHE chỉ có thể thực thi một số lần giơi hạn các phép toán.

2.2.4. Fully Homomorphic Encryption

Mã hóa đồng cấu toàn phần (Fully HE) là mã hóa đồng hinh cho phép thực hiện các phép tính đồng cấu cộng và nhân không giới hạn trên dữ liệu đã được mã hóa. Vào năm 2009, Craig Gentry đã giới thiệu lược đồ FHE đầu tiên trong nghiên cứu của mình [31], tuy nhiên lược đồ này rất khó để áp dụng vào thực tiễn do yêu cầu nhiều phép tính toán. Chính vì vậy đã có nhiều nỗ lực nghiên cứu nhằm đưa ra các lược đồ FHE mới cải tiến dựa trên nghiên cứu của Craig. Các lược đồ FHE có thể được phân thành 4 loại chính dựa trên các bài toán:

1. **Ideal lattice**: được đề xuất bởi Gentry [31].
2. **Over integers**: Van Dijket đã đề xuất một lược đồ dựa trên bài toán ước số chung lớn nhất gần đúng (Approximate GCD) [32].
3. **Ring Learning with Error (RLWE):** được đề xuất bởi Brakerski và Vaikuntanathan.
4. **NTRU-like**: NTRUEncrypt là một lược đồ cũ dựa trên lattice vừa được phát hiện có tính đồng cấu gần đây [33].

Dựa trên 4 bài toán như trên, có 3 lược đồ FHE đã được phát triển và sử dụng nhiều nhất, đó là: BGV, BFV và CKKS. Tuy nhiên trong nghiên cứu này nhóm chúng tôi sẽ chỉ tập trung vào lược đồ CKKS do đây là lược đồ thích hợp nhất cho các ứng dụng học máy bởi nó hỗ trợ phép cộng và nhân trên các số thực đã được mã hóa và cho ra các kết quả gần đúng.

2.3. Học cộng tác

2.3.1. Tổng quan

Học cộng tác là một cách tiếp cận học máy mà cho phép các mô hình học máy nhận cập nhật mô hình từ nhiều nguồn dữ liệu ở các vị trí khác nhau mà không cần phải chia sẻ dữ liệu huấn luyện. Điều này cho phép dữ liệu cá nhân vẫn được giữ lại ở cục bộ, giảm thiểu khả năng vi phạm dữ liệu riêng tư.

2.3.2. Quá trình huấn luyện trong học cộng tác

Trong học cộng tác, giả sử có N bên tham gia cùng huấn luyện mô hình M_q, mỗi bên được ký hiệu bởi A_t với $t \in [1, N]$. Mỗi A_t có dữ liệu và nơi lưu trữ D_t của riêng họ.

Trước khi bắt đầu huấn luyện, tất cả A_t sẽ thống nhất một tập siêu tham số (hyperparameters) như cấu hình của mô hình và khởi tạo các trọng số (weight) ngẫu nhiên cho mô hình khởi tạo. Tại mỗi vòng huấn luyện, mô hình toàn cục hiện tại M^i_g được gởi đến một số A_t , sau đó các A_t này sẽ tính toán cập nhật cho mô hình chung này dựa trên dữ liệu cục bộ D_t và cho ra mô hình cục bộ M^t_g . Và sau đó tất cả mô hình cục bộ được thu thập một lần nữa và được tổng hợp vào mô hình M^{i+1}_g.

2.3.3. Tổng hợp mô hình

Sau khi thu thập các cập nhật mô hình từ các A_t thì chúng phải được tổng hợp lại thanh một mô hình duy nhất sử dụng thuật toán tổng hợp. Đó là, cho một tập các bản cập nhật, làm thế nào để tổng hợp các cập nhật này? Năm 2016 McMahan [34] đã giới thiệu thuật toán FederatedAveraging để giải quyết vấn đề này. Trong thuật toán này, các bên tham gia huấn luyện thực hiện tính toán với dữ liệu cục bộ và chia sẻ các trọng số với một bên tổng hợp tập trung (có thể là một máy chủ). Sau đó máy chủ sẽ kết hợp tất cả các cập nhật này bằng cách lấy trung bình các tham số mô hình. Kết quả của lần tổng hợp này sẽ là điểm bắt đầu cho lần huấn luyện kế tiếp.

2.3.4. Hệ thống học cộng tác

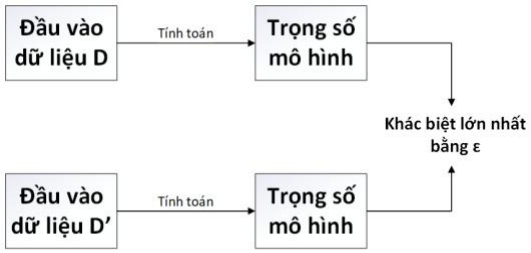
Trong khi học cộng tác mô tả các ý tưởng để hợp sức huấn luyện một mô hình, hệ thống học cộng tác (FLS) là một thiết kế thực sự để thực thi các ý tưởng này. Trong nghiên cứu của nhóm tác giả [35] đã trình bày một vài khía cạnh khi xây dựng hệ thống học cộng tác, bao gồm: phân phối dữ liệu (data distribution), mô hình học máy (machine learning model), cơ chế bảo vệ quyền riêng tư (privacy mechanism), mức độ cộng tác (scale of federation), động lực cộng tác (motivation of federation) và kiến trúc giao tiếp (communication architecture). Cụ thể hơn:

- Phân phối dữ liệu: dữ liệu có thể được phân phối theo chiều ngang (horizontal) hoặc chiều dọc (vertical). Theo chiều ngang thì dữ liệu sẽ có cùng tập đặc trưng (feature space) nhưng khác tập mẫu (set of samples) và theo chiều dọc thì ngược lại.
- Mô hình học máy: mô tả loại mô hình mà hệ thống muốn huấn luyện, ví dụ phổ biến như mạng thần kinh nhân tạo (artificial neural network), cây quyết định (decision tree) hay các mô hình tuyến tính.
- Cơ chế quyền riêng tư: bản chất của học cộng tác đã cải thiện nhiều về tính riêng tư của dữ liệu nhưng như vậy là chưa đủ. Bên cạnh đó có nhiều phương pháp để cải thiện tính riêng tư hơn nữa như: mã hóa đồng cấu, multi-party-computation hay differential privacy.
- Mức độ cộng tác: xác định số lượng đối tượng sẽ tham gia huấn luyện cũng như lượng dữ liệu mà mỗi bên sẽ huấn luyện. Thường được chia làm 2 loại: cross-silo - chỉ một vài đối tượng với lượng dữ liệu lớn hay cross-device - nhiều đối tượng tham gia với lượng dữ liệu không lớn.
- Động lực cộng tác: đưa ra được các lý do rõ ràng để khuyến khích các đối tượng tự nguyện tham gia vào học cộng tác.
- Kiến trúc giao tiếp: có 2 loại thường được sử dụng là tập trung (centralize) và phân quyền (decentralize). Trong kiến trúc tập trung sẽ có một thực thể trung tâm (máy chủ) chịu trách nhiệm cho việc tổng hợp và xử lý quá trình huấn luyện, các bên tham gia huấn luyện chỉ giao tiếp với máy chủ đó. Ngược lại, trong kiến trúc phân quyền thì không tồn tại thực thể trung tâm, thay vào đó các bên tham gia sẽ giao tiếp trực tiếp với nhau và cập nhật mô hình toàn cục một cách trực tiếp.

2.4. Quyền riêng tư khác biệt

Quyền riêng tư khác biệt (Differential Privacy) là một học thuyết cung cấp cho chúng ta một vài đảm bảo mang tính toán học về tính riêng tư của thông tin người dùng.

Mục đích chính là giảm thiểu ảnh hưởng của bất cứ dữ liệu đơn lẻ nào đến kết quả tổng thể. Điều này có nghĩa là một người sẽ cho ra cùng suy luận về một dữ liệu đơn lẻ dù cho nó có hoặc không có mặt trong đầu vào của việc phân tích (tham khảo hình 2.7). Khi số lượng các phân tích trên dữ liệu gia tăng thì rủi ro tiết lộ thông tin người dùng càng lớn. Kết quả của việc thực hiện các phép tính toán có sử dụng DP miễn nhiễm với một số lượng lớn các cuộc tấn công về quyền riêng tư. Trong FL, DP được thực hiện bằng cách thêm vào một cách cẩn thận các dữ liệu gây nhiễu đã được điều chỉnh (đặc trưng bởi chỉ số epsilon ε) vào các trọng số trước khi chúng được gởi đến nơi tổng hợp. Việc thêm nhiễu dẫn đến sự suy giảm của độ chính xác việc tính toán, do đó có sự đánh đổi giữa độ chính xác và tính riêng tư. Mức độ riêng tư được đánh giá bởi ε và tỷ lệ nghịch với mức độ bảo vệ riêng tư. Điều này có nghĩa là ε càng cao thì mức độ bảo vệ riêng tư càng thấp và khả năng tiết lộ thông tin người dùng càng cao. Việc đạt được ε-DP là một trường hợp lí tưởng và rất khó để đạt được trong các ngữ cảnh thực tế, do đó (ε, δ)-DP được sử dụng.



Hình 2.4 Differential privacy

Ví dụ, trong y tế, hồ sơ của bệnh nhân mang tính riêng tư rất cao, giả sử một nghiên cứu sử dụng các dữ liệu này tính toán về độ tuổi trung bình của một căn bệnh, kết quả cho ra tuổi trung bình của 1000 bệnh nhân là 28000, suy ra tuổi trung bình là 28. Tuy nhiên khi bỏ dữ liệu của một người ra, kết quả trung bình là 28.007. Từ đây, có thể suy ra tuổi của bệnh nhân bị bỏ ra là 21 tuổi, lúc này quyền riêng tư bị vi phạm, với DP sẽ giúp đầu ra của phép tính trên tập 1000 người và tập lân cận có kết quả khác nhau không quá nhiều, từ đó, không thể suy ra thông tin của bệnh nhân khác khi bỏ qua thông tin của một số bệnh nhân.

Định nghĩa toán học

Một cơ chế ngẫu nhiên M : X → R miền X và trong khoảng R cung cấp (ε, δ)-DP cho mọi tập hợp đầu ra S ⊆ R, và cho bất kỳ 2 tập dữ liệu lân cận nào của D, D' ∈ X, nếu M thỏa 2.3

$$\Pr[M(D_i) \in S] \leq e^{\varepsilon} \Pr[M(M(D'_i) \in S] + \delta \qquad (2.3)$$

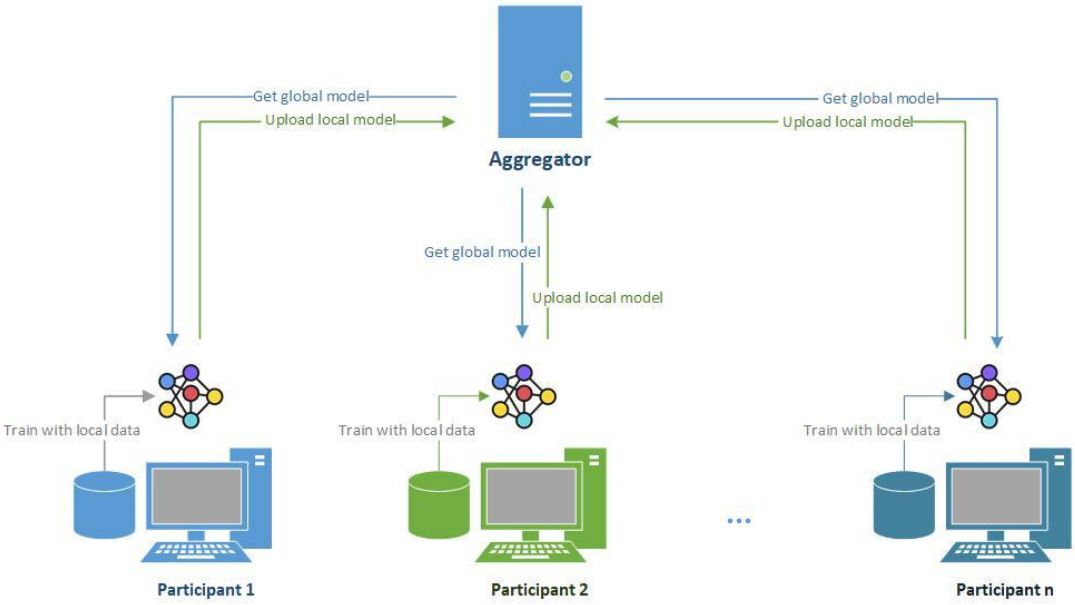
2.4.1. IDS dựa trên dị thường

Hệ thống phát hiện xâm nhập dựa trên dị thường (hay bài toán phát hiện dị thường) được thiết lập dựa trên việc phân tích thống kê, nó xác định các cuộc tấn công dựa trên sự bất thường trong khuôn mẫu so với các khuôn mẫu thông thường của lượng dữ liệu trong mạng. Hay nói cách khác, nó dựa vào phân tích thống kê lưu lượng mạng, tất cả cả trạng thái thông thường đều được đánh dấu là hành vi bình thường của hệ thống, những trường hợp còn lại đều được xem là các hành vi bất thường (anomalous activities). Loại IDS này dựa trên kĩ thuật khai thác dữ liệu (data-mining), do đó nó có thể xác định các mối đe dọa hay tấn công kiểu mới.

Chương 3. PHƯƠNG PHÁP ĐỀ XUẤT

3.1. Phương pháp học cộng tác

3.1.1. Mô hình FL tổng quan



Hình 3.1 Hình ảnh luồng hoạt động của mô hình học cộng tác

Kiến trúc học cộng tác tổng quan được đưa ra như trong hình 3.1. Sau đó dựa trên mô hình này để áp dụng các kĩ thuật HE và DP nhằm nâng cao tính riêng tư, luồng hoạt động chi tiết được mô tả trong 3.1.2 và 3.1.3.

3.1.2. Luồng hoạt động của mô hình FL có sử dụng HE

Mã hóa đồng cấu sẽ được kết hợp trong chiến lược huấn luyện của học cộng tác trong ngữ cảnh IDS. Mỗi bên tham gia (participant) sẽ mã hóa tham số mô hình (cụ thể là trọng số) trước khi gởi chúng đến máy chủ tập trung để tổng hợp. Luồng hoạt động của mô hình có sử dụng HE có thể mô tả trong bốn giai đoạn như dưới đây (và tham khảo thêm thuật toán 1) :

1. Khởi tạo: ở giai đoạn bắt đầu của quá trình học cộng tác, để đơn giản chúng tôi giả sử sẽ có một máy chủ đáng tin cậy thứ ba sẽ khởi tạo cặp khóa (SK,

PK) để dùng cho HE và phân phối chúng đến toàn bộ bên tham gia trong một kênh truyền an toàn, ngoài ra khóa công khai cũng sẽ phân phối cho máy chủ tổng hợp. Gọi R là tổng số vòng giao tiếp giữa máy chủ tổng hợp (aggregator) và một bên tham gia. Tất cả các bên tham gia sẽ chia đều tập dữ liệu của họ D_k thành R phần { d^1_k , d^2_k, ... d^R_k }. Máy chủ tổng hợp sau đó sẽ chọn một bên tham gia bất kì làm lãnh đạo (leader) để khởi tạo các trọng số khởi đầu W_0. Sau đó các trọng số khởi đầu này cũng sẽ được đồng bộ với các bên tham gia khác. Đến đây không có sự khác biệt giữa lãnh đạo và các bên tham gia còn lại.

2. Huấn luyện mô hình cục bộ: sau khi nhận được các trọng số khởi tạo W_0, mỗi bên tham gia sẽ huấn luyện mô hình IDS dựa trên học máy ở cục bộ trên sử dụng nguồn dữ liệu của riêng họ D_k. Quá trình huấn luyện chi tiết được tóm tắt ở thuật toán 1. Khi huấn luyện một mô hình, mỗi bên tham gia P_k sẽ mã hóa các trọng số W^r_k bằng Encrypt(W^r_k , PK). Sau đó các trọng số đã được mã hóa sẽ được gởi đến máy chủ tổng hợp cùng với kích cỡ của dữ liệu đã huấn luyện (được ký hiệu bởi α).
3. Tổng hợp các trọng số mô hình: khi máy chủ tổng hợp nhận được kích cỡ dữ liệu huấn luyện và các trọng số đã được mã hóa từ tất cả bên tham gia thì sẽ bắt đầu tổng hợp bằng Aggregate(W^r_1 , ..., W^r_N , α_1, ..., α_N). Kết quả cho ra sẽ là bản mã đã được tổng hợp C và được gởi lại cho tất cả bên tham gia.
4. Cập nhật mô hình cục bộ: bằng việc giải mã bản mã C sử dụng Decrypt(C, SK), mỗi bên tham gia có thể thu được các trọng số để cập nhật W^r_k và dùng chúng để cập nhật mô hình cục bộ.

Algorithm 1 HEFL

Input: Tập các bên tham gia P, nguồn dữ liệu của N bên tham gia $\{D_k | k \in (1, 2, 3...N)\}$, số vòng giao tiếp R

Output: Mô hình đã được tổng hợp

Init: $\forall k \in (1, 2, 3...N)$ P_k chia tập dữ liệu D_k thành R phần $\{d^1_k , d^2_k, ... d^R_k \}$. Một máy chủ đã được chứng thực khởi tạo cặp khóa (SK,PK) và phân phối cho các bên tham gia.

Máy chủ tổng hợp chọn ngẫu nhiên một bên tham gia làm lãnh đạo để khởi tạo trọng số ban đầu W_0 .

for $r \leq R$ **do**

  (I). Đối với các bên tham gia:

  for $\forall k \in (1, 2, 3...N)$ **do**

    P_k huấn luyện và tính toán các trọng số W^r_k của mô hình cục bộ vòng thứ r trên tập dữ liệu d^r_k .

    P_k mã hóa các trọng số: $E(W^r_k) = Encrypt(W^r_k , PK)$

    P_k gởi các trọng số đã được mã hóa $E(W^r_k)$ và kích thước của d^r_k, ký hiệu bởi α^r_k đến máy chủ tổng hợp.

  (II). Đối với máy chủ tổng hợp:

    $C = Aggregate(W^r_1, ..., W^r_N , \alpha^r_1, ..., \alpha^r_N)$.

    Phân phối bản mã đã được tổng hợp C đến tất cả P_k $(k \in (1, 2, 3...N))$

  (III). Đối với các bên tham gia:

  for $\forall k \in (1, 2, 3...N)$ **do**

      $W^r_k = Decrypt(C, SK)$

      P_k cập nhật mô hình cục bộ sử dụng các trọng số đã được cập nhật W^r_k.

  $r \leftarrow r + 1$

Sau R vòng huấn luyện giữa máy chủ tổng hợp và các bên tham gia, kết quả thu được một mô hình IDS dựa trên học máy.

3.1.3. Luồng hoạt động mô hình có sử dụng DP

Differential Privacy có thể gia tăng tính bảo mật trong mô hình FL bằng việc chèn một lượng nhiễu có kiểm soát vào các trọng số trước khi gởi đến máy chủ tổng hợp. Việc sử dụng DP là một phép đánh đổi giữa bảo vệ quyền riêng tư và độ chính xác của mô hình. Luồng hoạt động của mô hình có sử dụng DP có thể mô tả trong bốn giai đoạn như dưới đây (và tham khảo thêm thuật toán 2).

1. Khởi tạo: gọi R là tổng số vòng giao tiếp giữa máy chủ tổng hợp (aggregator) và một bên tham gia. Tất cả các bên tham gia sẽ chia đều tập dữ liệu của họ D_k thành R phần { d^1_k , d^2_k, ... d^R_k }. Máy chủ tổng hợp sau đó sẽ chọn một bên tham gia bất kì làm lãnh đạo (leader) để khởi tạo các trọng số khởi đầu W_0. Sau đó các trọng số khởi đầu này cũng sẽ được đồng bộ với các bên tham gia khác. Đến đây không có sự khác biệt giữa lãnh đạo và các bên tham gia còn lại.
2. Huấn luyện mô hình cục bộ: sau khi nhận được các trọng số khởi tạo W_0, mỗi bên tham gia sẽ huấn luyện mô hình IDS dựa trên học máy ở cục bộ trên sử dụng nguồn dữ liệu của riêng họ D_k. Quá trình huấn luyện chi tiết được tóm tắt ở thuật toán 2. Khi huấn luyện một mô hình, mỗi bên tham gia P_k sẽ thêm nhiễu vào các trọng số W^r_k bằng AddNoise(W^r_k). Sau đó các trọng số đã được gây nhiễu sẽ được gởi đến máy chủ tổng hợp cùng với kích cỡ của dữ liệu đã huấn luyện (được ký hiệu bởi α).
3. Tổng hợp các trọng số mô hình: khi máy chủ tổng hợp nhận được kích cỡ dữ liệu huấn luyện và các trọng số đã được gây nhiễu từ tất cả bên tham gia thì sẽ bắt đầu tổng hợp bằng Aggregate(W^r_1, ..., W^r_N , α^r_1, ..., α^r_N). Kết quả cho ra sẽ là trọng số toàn cục W_{global} và được gởi lại cho tất cả bên tham gia.
4. Cập nhật mô hình cục bộ: các bên tham gia sẽ nhận W_{global} từ máy chủ tổng hợp và dùng chúng để cập nhật mô hình cục bộ.

Algorithm 2 DPFL

Input: Tập các bên tham gia P, nguồn dữ liệu của N bên tham gia $\{D_k|k \in (1, 2, 3...N)\}$, số vòng giao tiếp R

Output: Mô hình đã được tổng hợp

Init: $\forall k \in (1, 2, 3...N)$ P_k chia tập dữ liệu D_k thành R phần $\{d^1_k , d^2_k, ... d^R_k \}$.

Máy chủ tổng hợp chọn ngẫu nhiên một bên tham gia làm lãnh đạo để khởi tạo trọng số ban đầu W_0.

for r ≤ R do

(I). Đối với các bên tham gia:

for ∀k ∈ (1, 2, 3...N) do

P_k huấn luyện và tính toán các trọng số W^r_k của mô hình cục bộ vòng thứ r trên tập dữ liệu d^r_k .

P_k thêm nhiễu vào các trọng số:: W^r_k = AddNoise(W^r_k)

P_k gởi các trọng số đã được gây nhiễu W^r_k và kích thước của d^r_k, ký hiệu bởi α^r_k đến máy chủ tổng hợp.

(II). Đối với máy chủ tổng hợp:

W_{global} = Aggregate(W^r_1, ..., W^r_N , α^r_1, ..., α^r_N).

Phân phối các trọng số toàn cục W_{global} đến tất cả P_k ($k \in (1, 2, 3...N)$)

(III). Đối với các bên tham gia:

for ∀k ∈ (1, 2, 3...N) do

Pk nhận các trọng số toàn cục W_{golbal} từ máy chủ tổng hợp và cập nhật mô hình cục bộ

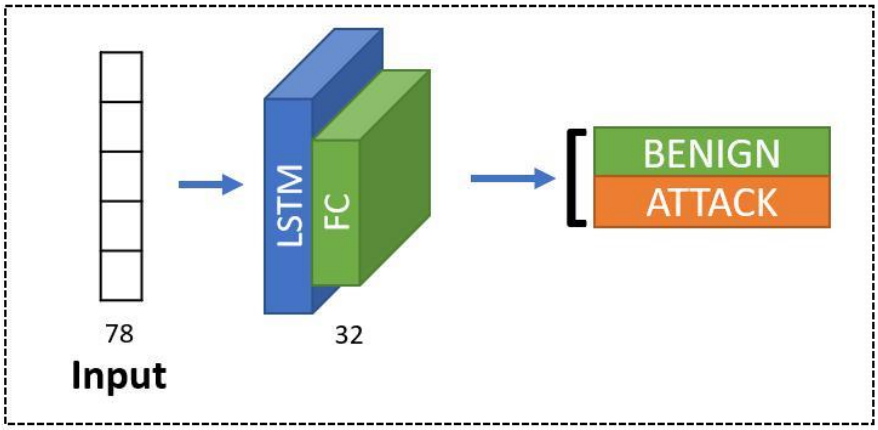
$r \leftarrow r + 1$

Sau R vòng huấn luyện giữa máy chủ tổng hợp và các bên tham gia, kết quả thu được một mô hình IDS dựa trên học máy.

3.2. Các mô hình đề xuất cho IDS

3.2.1. Long Short Term Memory networks

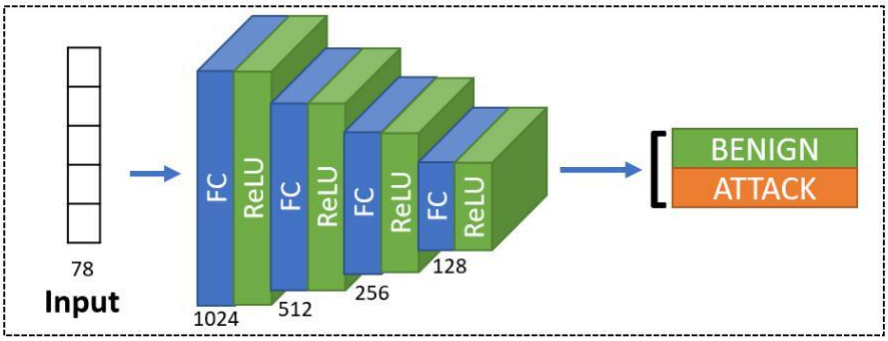
Mạng trí nhớ ngắn hạn định hướng dài hạn (Long Short Term Memory networks) thường được viết tắt LSTM, là một dạng đặc biệt của mạng hồi quy RNN có khả năng học được sự phụ thuộc dài hạn (long-term dependency), có thể nhớ được thông tin trong thời gian dài. Xây dựng mô hình với lớp LSTM có kích thước đầu vào là 78 (bằng với số lượng đặc trưng), kích thước lớp ẩn là 32, tiếp theo đó là lớp linear với kích thước đầu ra là 2, mô hình được vẽ tại hình 3.2.



Hình 3.2 Hình ảnh mô hình sử dụng LSTM

3.2.2. Fully connected network

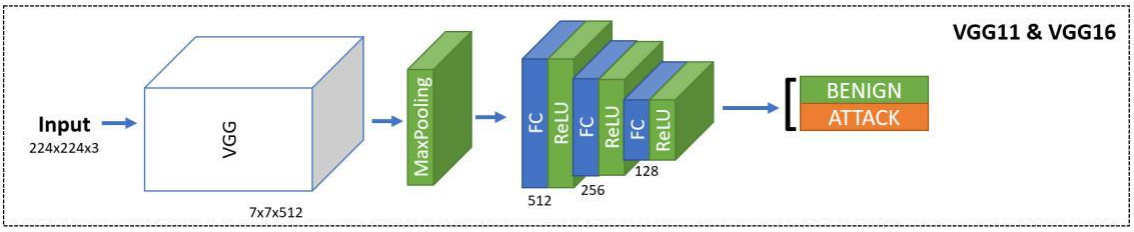
Xây dựng fully conntected network với hàm kích hoạt là ReLU, kích thước đầu vào là 78 (bằng với số lượng đặc trưng), kích thước các lớp lần lượt là 78 (lớp đầu vào), 1024, 512, 256, 128, 2 (lớp đầu ra) (Hình 3.3).



Hình 3.3 Hình ảnh fully connected network

3.2.3. VGG11 và VGG16

VGG là một convolutional neural network với một kiến trúc cụ thể được công bố trong bài báo [38] bởi một nhóm các nhà nghiên cứu từ Đại học Oxford. Nhóm đã tham gia cuộc thi thị giác máy tính ImageNet Large Scale Visual Recognition Challenge (ILSVRC), VGG đạt nằm trong 5 mô hình về độ chính xác nhất trên tập dữ liệu kiểm tra và giúp nhóm giành vị trí số 1 và vị trí số 2 trong cuộc thi ILSVRC 2014. Chúng tôi sử dụng transfer learning trong 2 mô hình cuối. Sử dụng mô hình VGG11 và VGG16 đã được huấn luyện trước, áp dùng kỹ thuật trích xuất đặc trưng (feature extraction), dữ liệu đầu vào sẽ đi qua các lớp convolution của mô hình VGG, trọng số tại các layer này sẽ được đống băng (freeze) để không điều chỉnh trong quá trình huấn luyện. Do đầu ra khi đi qua các lớpr trên là 7x7x512=25088, số lượng nơ ron này kết hợp với layer linear phía sau sẽ là rất lớn để có thể sử dụng mã hóa đồng cấu, nên chúng tôi sử dụng một lớp maxpooling kích thước 7x7 sau đó, kích thước sau lớp maxpooling là 1x512, tiếp theo là fully connected network với hàm kích hoạt là ReLU có số nơ ron tại mỗi layer lần lượt là 512 (lớp nhận đầu vào từ lớp maxpooling), 256, 128, 2 (lớp đầu ra) (Hình 3.4).



Hình 3.4 Mô hình sử dụng vgg kết hợp FC

Chương 4. **THỰC NGHIỆM VÀ ĐÁNH GIÁ**

Tóm tắt

Chương này sẽ trình bày triển khai mô hình và kết quả thực nghiệm khi thực hiện các mô hình đề xuất trên tập dữ liệu và nhiều kịch bản thử nhiệm.

4.1. Môi trường thí nghiệm

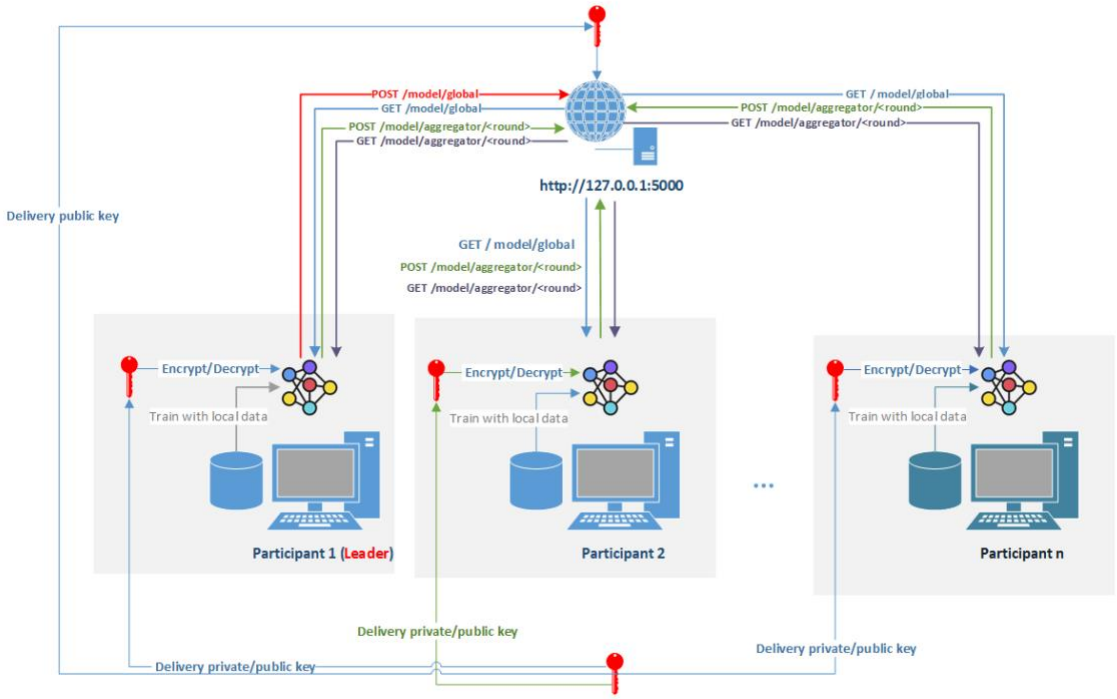
Cấu hình

Các thí nghiệm được chạy trên máy ảo Ubuntu với cấu hình CPU 16 core và RAM 64 GB và cài đặt Python 3.8

Xây dựng hệ thống học cộng tác

Các mô hình học sâu được xây dựng bằng Pytorch, Aggregator và Worker trao đổi trọng số với nhau qua giao thức HTTP/HTTPS, được xây dựng bằng framework Flask (luồng hoạt động thể hiện qua hình ảnh 4.1). Quá trình làm việc chi tiết như sau:

1. Các bên tham gia (participant) sẽ ghi danh với máy chủ tổng hợp (aggregator), mỗi participant sẽ được cung cấp thông tin ID và số lượng round cần train. Chọn ngẫu nhiên một participant làm leader.



Hình 4.1 Hình ảnh luồng hoạt động của kiến trúc đề xuất

2. Participant được chọn làm leader sẽ gửi trọng số đến aggregator thông qua POST /model/global, các participant còn lại sẽ nhận trọng số thông qua GET /model/global
3. Các participant huấn luyện với dữ liệu cục bộ, mã hóa các trọng số bằng mã hóa đồng cấu và serialize dưới dạng bytes. Sau đó gửi trọng số đến aggregator như là một tệp tin thông qua POST /model/aggregation/<round>.
4. Aggregator nhận được trọng số và lưu xuống ổ cứng, khi đủ số lượng trọng số, aggregator đọc từng file lên và thực hiện tổng hợp trung bình (FedAvg) theo công thức 4.1.

$$W_{t+1} \leftarrow \sum_{K=1}^{K} \frac{n_k}{n} w_{t+1}^{k} \qquad (4.1)$$

5. Các paritipant lấy trọng số đã tổng hợp thông qua GET /model/aggregation/<round>, giải mã và cập nhật trọng số cho mô hình cục bộ.

TenSEAL – thư viện mã hóa đồng cấu

Thư viện hỗ trợ mã hóa đồng cấu sử dụng TenSEAL, là bản triển khai hỗ trợ mã hóa đồng cấu có hỗ trợ lược đồ CKKS cho các Tensor, Vector sử dụng ngôn ngữ Python được xây dựng trên thư viện Microsoft SEAL.

Các trọng số của Pytorch được lưu dưới dạng Dictionary chứa tên các layer và giá trị các trọng số tại các layer đó dạng Tensor, một cặp khóa và giá trị sẽ có dạng [str, Tensor]. Việc mã hóa một mô hình đồng nghĩa với việc mã hóa từng từng cặp [str, Tensor] thành [str, CiphertextTensor], Dictionary này được serialize dạng bytes để có thể lưu trữ và trao đổi giữa worker và aggregator.

Chúng tôi sử dụng khóa có poly_modulus_degree = 8096 bởi vì sử dụng phương pháp batching, phương pháp này thực hiện nối nhiều số để mã hóa cùng lúc nên plaintext sẽ bị giới hạn bởi độ dài của khóa, do đó mô hình lớn như Vgg cần độ dài khóa là 8096.

Opacus - thư viện Differential Privacy

Để huấn luyện mô hình với differential privacy, chúng tôi sử dụng Opacus được xây dựng bởi Facebook, Opacus hỗ trợ thêm nhiễu vào mô hình bằng cách điều chỉnh trọng số của mô hình trong lúc huấn luyện.

Chúng tôi sử dụng Opacus để can thiệp vào optimizer trong lúc huấn luyện. Chúng tôi chọn chỉ số noise_multiplier trong quá trình huấn luyện là 1.3 tương ứng với giá trị khi huấn luyện trong khoảng $1.17 < \epsilon < 1.2$

Riêng mô đun LSTM của Pytorch không thể huấn luyện với Differentail privacy của Opacus, phải sử dụng mô đun DPLSTM của Opacus để thay thế.

4.2. Kịch bản thí nghiệm

Chúng tôi thử nghiệm qua 5 kịch bản để đánh giá sự đánh đổi giữa quyền riêng tư của dữ liệu và hiệu năng, độ chính sát.

1. **Local**: Huấn luyện các mô hình cục bộ với dữ liệu của riêng họ.

2. **Ideal**: Huấn luyện các mô hình cục bộ với toàn bộ dữ liệu.

3. **HEFL**: Huấn luyện các mô hình với phương pháp học cộng tác kết hợp mã hóa đồng cấu, dữ liệu được chia điều cho các bên tham gia.
4. **DPFL**: Huấn luyện các mô hình với phương pháp học cộng tác kết hợp làm nhiễu, dữ liệu được chia điều cho các bên tham gia.

5. **Hybrid**: Huấn luyện các mô hình với phương pháp học cộng tác sử dụng mã hóa đồng cấu kết hợp làm nhiễu, dữ liệu được chia điều cho các bên tham gia.

4.3. Đặc tả tập dữ liệu

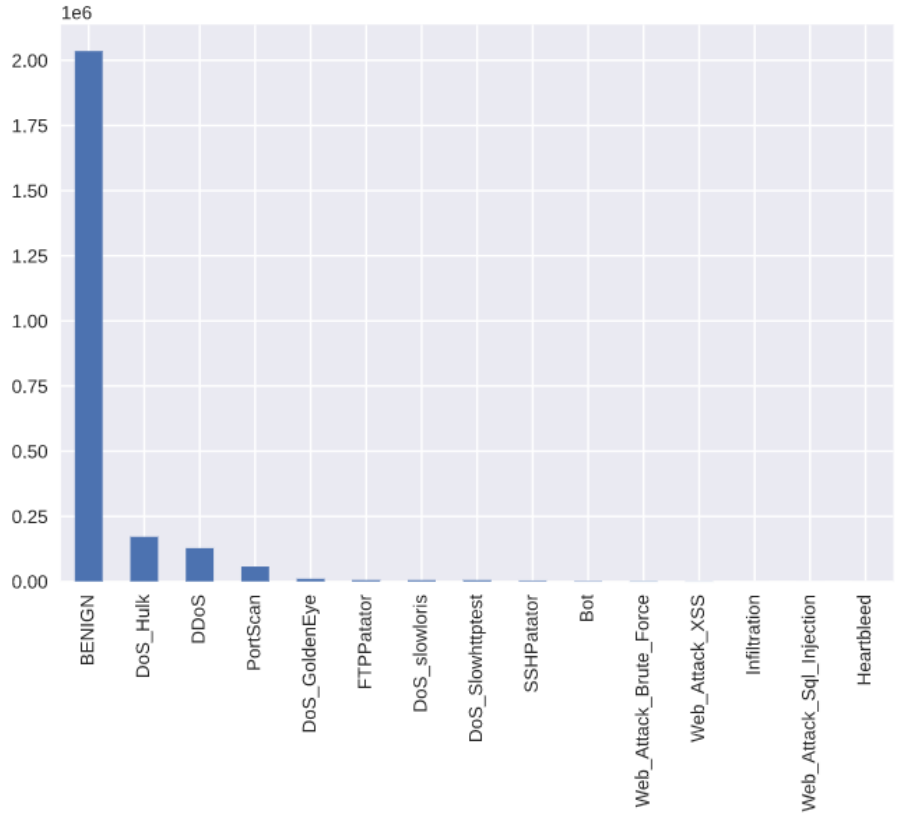
CICIDS-2017 là tập dữ liệu tạo bởi Canadian Institute for Cybersecurit. Bộ dữ liệu chứa nhiều loại tấn công đa giai đoạn như Heartbleed và các nhiều loại tấn công DOS, DDOS khác nhau. Bảng 4.1 thể hiện số lượng mẫu của mỗi nhãn trong tập dữ liệu CICIDS-2017.

File (CSV)	Type of Traffic	Number of Record
Monday-WorkingHours	Benign	529,918
Tuesday-WorkingHours	Benign	432,074
	SSH-Patator	5,897
	FTP-Patator	7,938
Wednesday-WorkingHours	Benign	440,031
	DoS Hulk	231,073
	DoS GoldenEye	10,293
	DoS Slowloris	5,796
	DoS Slowhttptest	5,499
	Heartbleed	11
Thursday-WorkingHours-Morning-WebAttacks	Benign	168,186
	Web Attack-Brute Force	1,507
	Web Attack-Sql Injection	21
	Web Attack-XSS	652
Thursday-WorkingHours-Afternoon-Infilteration	Benign	288,566
	Infiltration	36
Friday-WorkingHours-Morning	Benign	189,067

	Bot	1,966
Friday-WorkingHours-Afternoon-PortScan	Benign Portscan	127,537 158,930
Friday-WorkingHours-Afternoon-DDos	Benign DdoS	97,718 128,027
Tổng cộng		**2,830,743**

Bảng 4.1 Tóm tắt tập dữ liệu CICIDS-2017

Mỗi dòng trong tập dữ liệu bao gồm 78 đặc trưng, đã được gán nhãn sẵn, biểu đồ phân bố theo nhãn được biểu thị trong hình 4.2.



Hình 4.2 Phân bổ dữ liệu CICIDS-2017 theo nhãn

4.4. Tiền xử lý tập dữ liệu

Chúng tôi sử dụng tập dữ liệu CICIDS-2017 cho 4 mô hình, đó là mô hình dựa trên LSTM, DNN và mô hình dựa trên VGG16, VGG11 kết hợp DNN, nên phải tiền xử lý tập dữ liệu theo 2 cách khác nhau.

Đầu tiên xóa đi các mẫu có đặc trưng với giá trị không phải số (NaN) và các giá trị vô cực (Inf).

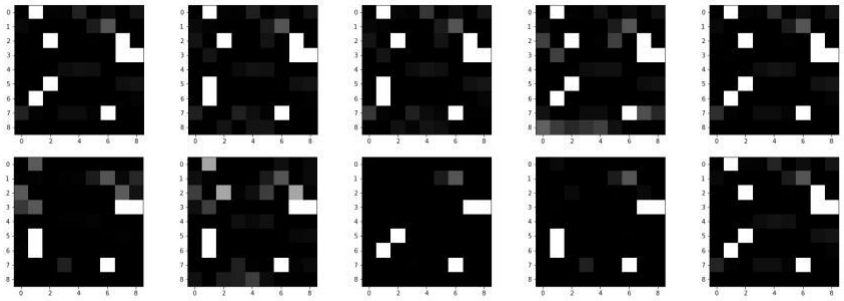
Các nhãn "BENIGN" sẽ được gán bằng 1, tất cả các nhãn còn lại được gán bằng 0. Với mô hình dựa trên LSTM, các đặt trưng được chuẩn hóa bằng lớp StandardScaler theo công thức 4.2.

$$z = (x - u)/s \qquad (4.2)$$

Với mô hình dựa trên VGG16, trước tiên sử dụng MinMaxScaler để chuyển giá trị của các đặt trưng về khoảng từ 0 đến 1 theo công thức 4.3.

$$x_{scale} = \frac{x - \min(X)}{\max(X) - \min(X)} \quad (4.3)$$

Sau đó thì dữ liệu phải chuyển về định dạng ảnh. Để chuyển về định dạng ảnh, số lượng đặt trưng của tập dữ liệu được mở rộng từ 78 thành 81, tiếp theo chuyển đổi thành mảng 2 chiều kích thước 9x9 (ảnh trắng đen kích thước 9x9), một số mẫu được chuyển thành ảnh trắng đen ở hình 4.3, bước tiếp theo thay đổi kích thước về 224x224 và chuyển đổi ảnh dạng RBG.



Hình 4.3 Hình ảnh của các đặt trưng khi chuyển về ảnh trắng đen

Sau đó, cân bằng tập dữ liệu bằng cách giảm số lượng tập dữ liệu có nhãn "BENIGN" xuống bằng với tổng số lượng nhãn tấn công.

4.5. Tiêu chí đánh giá

Để đánh giá hiệu năng của mô hình, chúng tôi sử dụng các giá trị accuracy, precision, recall, F1-score. Các chỉ số này được tính từ các thuộc tính trong confusion matrix: dương tính thật (True Positive - TP), dương tính giả (False Positive - FP), âm tính giả (False Negative - FN) và âm tính thực (True Negative - TN). Nếu chọn positive là mẫu tấn công, nagative là mẫu bình thường thì các thuộc tính này có ý nghĩa như sau:

- TP là số lượng dữ liệu tấn công được phân loại đúng.
- TN là số lượng dữ liệu bình thường được phân loại đúng.
- FN là số lượng dữ liệu tấn công bị phân loại là bình thường.
- FP là số lượng dữ kiệu bị phân loại là tấn công nhưng thực chất là bình thường.

Các công thức 4.4, 4.5,4.6, 4.7 là định nghĩa toán học của accuracy, precision, recall, F1-score.

$$\text{Accuracy} = \frac{\text{TN} + \text{TP}}{\text{TN} + \text{TP} + \text{FN} + \text{FP}} \quad (4.4)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (4.5)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (4.6)$$

$$\text{F1} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4.7)$$

Ngoài ra, chúng tôi còn đánh giá qua lưu lượng mạng mà những bên tham gia phải trao đổi với máy chủ tổng hợp, sử dụng đơn vị tính là kilobytes (KB), megabytes (MB) hoặc gigabytes (GB).

4.6. Kết quả thí nghiệm

Thực hiện lấy số lượng tham số và kích thước của các mô hình đã đề xuất ở mục 3, sau đó mã hóa với CKKS có sử dụng kỹ thuật batch để đo lường kích thước của mô hình sau khi mã hóa và thời gian mã hóa. Kết quả được thể hiện trong bảng 4.2. Qua đây có thể thấy được rằng mã hóa đồng cấu cho kích thước sau mã hóa lớn hơn rất nhiều so với mô hình ban đầu (chênh lệch lớn nhất trong thực nghiệm là gấp 200 lần với mô hình LSTM). Tuy nhiên, kiến trúc của mạng VGG đã là rất lớn nên kích thước mô hình sau mã hóa và thời gian mã hóa này là rất khả thi.

Model	Parameters	Plaintext Size	Ciphertext Size	Time to encrypt
LSTM Based	14402	254.76 KB	57.68 MB	2s
Fully connected network	204930	1.76 MB	219.5 MB	5s
VGG11 Based	9384962	96.21 MB	4.62 GB	84s

VGG16 Based	14879170	152.69 MB	7.5 GB	211s

Bảng 4.2 Bảng thể hiện số lượng tham số, thời gian mã hóa và kích trước/sau sử dụng mã hóa đồng cấu

Với mô hình LSTM, thực hiện đánh giá trên toàn bộ tập dữ liệu đã tiền xử lý và cân bằng, mỗi lần huấn luyện qua 10 epoch, sử dụng hàm mất mát Cross Entropy và optimizer Adam với learning là 1e-3. Chúng tôi thực hiện huấn luyện theo mô hình HEFL, DPFL với số lượng worker lần lượt là 2, 4, 6 và số round lần lượt là 3, 5, 7.

Kết quả (thể hiện ở Bảng 4.3) cho thấy mô hình HEFL cho kết quả cao, độ chính xác có xu hướng tăng khi tăng số round.

K	Round	Accuracy	Precision	Recall	F1
2	3	0,9824	0,9773	0,9877	0,9825
	5	0,98	0,9829	0,9768	0,9799
	7	**0,9842**	**0,9818**	**0,9866**	**0,9842**
4	3	0,9817	0,9796	0,9838	0,9817
	5	0,9815	0,9718	0,9918	0,9817
	7	0,9828	0,9787	0,987	0,9828
6	3	0,9801	0,9695	0,9913	0,9803
	5	0,9798	0,9667	0,9937	0,98
	7	0,9802	0,9688	0,9924	0,98

Bảng 4.3 Bảng kết quả thực nghiệm huấn mô hình LSTM sử dụng học cộng tác kết hợp mã hóa đồng cấu

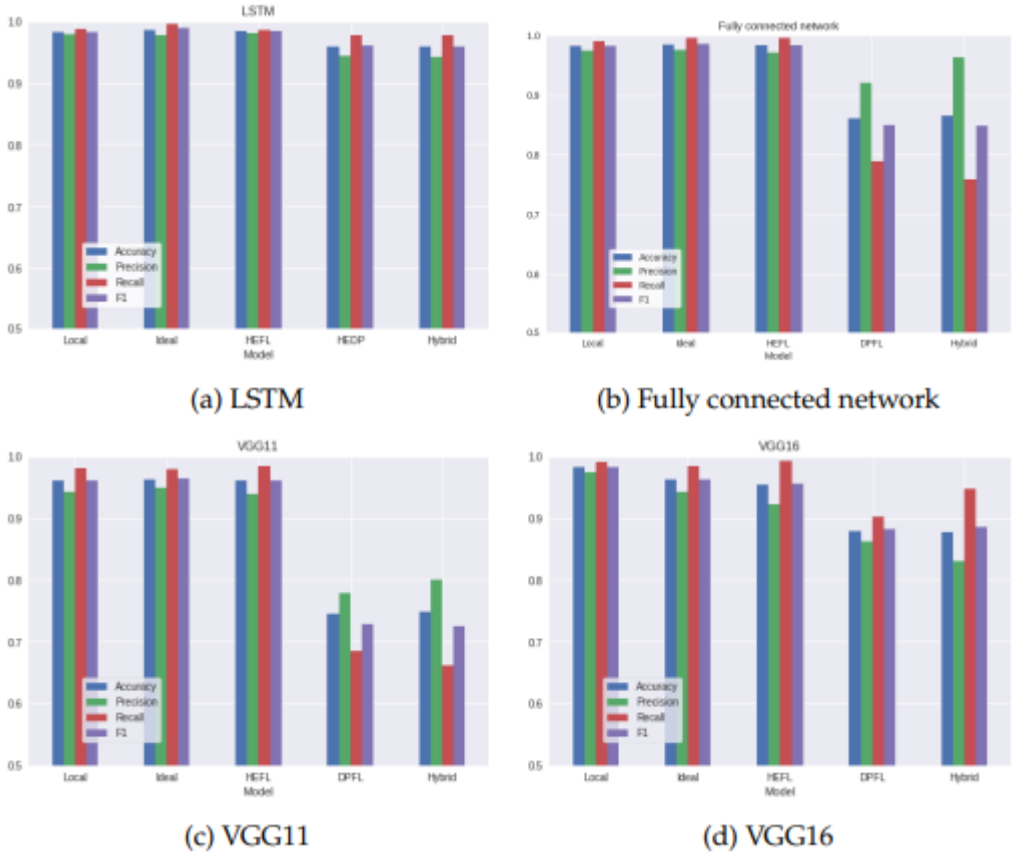
Qua kết quả huấn luyện mô hình DPFL (Bảng 4.4) cũng cho độ chính xác tăng dần khi số lượng round tăng, nhưng giảm dần khi số lượng worker tăng. Đây là hệ quả của việc mỗi worker đã thêm nhiễu vào trong quá trình huấn luyện.

K	R	Accuracy	Precision	Recall	F1
2	3	0.9565	0.9437	0.9707	0.957
	5	0.9595	0.9437	0.9772	0.9602
	7	**0.9601**	**0.944**	**0.978**	**0.9607**
4	3	0.951	0.9392	0.9642	0.9515
	5	0.9517	0.9383	0.9667	0.9523
	7	0.9532	0.9413	0.9666	0.9538
6	3	0.9489	0.9364	0.963	0.9495
	5	0.9489	0.9369	0.9625	0.9496
	7	0.9495	0.9377	0.9628	0.9501

Bảng 4.4 Bảng kết quả thực nghiệm huấn mô hình LSTM sử dụng học cộng tác kết hợp làm nhiễu

Các kết quả trên cho thấy rằng thực nghiệm trên K=2, R=7 cho kết quả tốt hơn các trường hợp khác, chúng tôi chọn K=2, R=7 để thực hiện huấn luyện cho các mô hình còn lại và so sánh với mô hình Ideal và Local.

Fully conntected network cũng được huấn luyện tương tự LSTM, tuy nhiên learning rate là 8e-4. Với các mô hình VGG, chúng tôi chỉ sử dụng 10% tập dữ liệu đã tiền xử lý và cân bằng, mỗi lần huấn luyện qua 1 epoch, sử dụng hàm mất mát Cross Entropy và optimizer Adam với learning rate là 8e-4, kết quả thực thi tất cả mô hình qua 5 kịch bản được thể hiện ở hình 4.4.



Hình 4.4 Kết quả so sánh các kịch bản của các mô hình đề xuất

Qua các kịch bản thí nghiệm trên, có thể thấy rằng mã hóa đồng cấu (HE) là một giải pháp đảm bảo riêng tư tốt trong học cộng tác mà vẫn đảm bảo độ chính xác của mô hình không quá chênh lệch so với mô hình học máy truyền thống tuy nhiên giải pháp đánh đổi bằng hiệu năng của máy tính và lưu lượng mạng vì thời gian mã hóa và kích thước của mô hình. Trong khi đó DP lại làm giảm độ chính xác mô hình mà không làm giảm nhiều hiệu năng của máy tính. Như vậy, đối với các mô hình IDS có độ phức tạp nhỏ, có thể áp dụng cả HE và DP để đạt được một mô hình có độ chính xác ổn mà vẫn đạt được mức độ bảo mật riêng tư cao. Còn đối với các mô hình có độ phức tạp lớn, nên cân nhắc khi sử dụng HE.

Chương 5. **KẾT LUẬN**

5.1. Kết quả

Việc xây dựng các mô hình học máy cho IDS bằng học cộng tác có thể giúp các tổ chức cùng nhau huấn luyện những mô hình IDS có hiệu quả cao mà không cần phải tập trung dữ liệu của mình cho một bên khác. Hơn nữa, việc tích hợp các giải pháp như mã hóa đồng cấu và differential privacy cũng nâng cao hơn tính riêng tư của dữ liệu trong suốt quá trình tổng hợp. Nghiên cứu này đạt được một số kết quả như sau:

- Xây dựng hệ thống học cộng tác cho hệ thống phát hiện xâm nhập có sử dụng mã hóa đồng cấu, differential privacy, cũng như kết hợp cả hai để đảm bảo quyền riêng tư cho mô hình học cộng tác.
- Áp dụng và đánh giá 4 mô hình: LSTM, Fully connected network, Vgg11 và Vgg16 trên tập dữ liệu CICIDS-2017 bằng phương pháp học cộng tác.
- Đã nộp 1 bài báo tại hội nghị và trong giai đoạn bình duyệt

Kết quả cho thấy việc áp dụng các kỹ thuật đảm bảo quyền riêng tư mặc dù phải đánh đổi hiệu năng và độ chính xác nhưng vẫn cho một độ hiệu quả nhất định trên các mô hình nhỏ đến mô hình lớn.

5.2. Hướng phát triển

Trong nghiên cứu này, chúng tôi đã khảo sát hiệu năng của giải pháp học cộng tác đưa ra với các mô hình có độ phức tạp khác nhau như LSTM, Fully Connected Network hay VGG. Tuy nhiên đây chỉ là một số lượng nhỏ trong số các mô hình học máy được dùng trong IDS, chính vì vậy có thể mở rộng số lượng mô hình IDS để coi khả năng đáp ứng của giải pháp được đưa ra.

Ngoài ra việc sử dụng mã hóa đồng cấu trong mô hình học cộng tác vẫn còn điểm bất cập, đó là tất cả các bên tham gia đều sử dụng chung một cặp khóa phục vụ cho việc mã hóa và giải mã, việc này vẫn dẫn đến những rủi ro về bảo mật. Vì vậy có thể áp dụng thuật toán mã hóa điểm tới điểm (end-to-end encryption) để thực hiện việc trao đổi khóa của mã hóa đồng cấu giữa các bên tham gia mà không phụ thuộc vào máy chủ.

TÀI LIỆU THAM KHẢO

[1] W. Tounsi and H. Rais, "A survey on technical threat intelligence in the age of sophisticated cyber attacks," *Computers & security,* vol. 72, pp. 212-233, 2018.

[2] N. A. A.-A. Al-Marri, B. S. Ciftler, and M. M. Abdallah, "Federated mimic learning for privacy preserving intrusion detection," in *2020 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, 2020: IEEE, pp. 1-6.

[3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature,* vol. 521, no. 7553, pp. 436-444, 2015.

[4] S. Linnainmaa, "The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors," Master's Thesis (in Finnish), Univ. Helsinki, 1970.

[5] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980,* 2014.

[6] C. A. Mack, "Fifty years of Moore's law," *IEEE Transactions on semiconductor manufacturing,* vol. 24, no. 2, pp. 202-207, 2011.

[7] N. P. Jouppi *et al.*, "In-datacenter performance analysis of a tensor processing unit," in *Proceedings of the 44th annual international symposium on computer architecture*, 2017, pp. 1-12.

[8] M. P. e. al. "Advancing state-of-the-art image recognition with deep learning on hashtags." https://engineering.fb.com/2018/05/02/ml-applications/advancing-state-of-the-art-image-recognition-with-deep-learning-on-hashtags/ (accessed.

[9] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation policies from data," *arXiv preprint arXiv:1805.09501,* 2018.

[10] A. Halevy, P. Norvig, and F. Pereira, "The unreasonable effectiveness of data," *IEEE intelligent systems,* vol. 24, no. 2, pp. 8-12, 2009.

[11] S. K. Lo, Q. Lu, L. Zhu, H.-y. Paik, X. Xu, and C. Wang, "Architectural patterns for the design of federated learning systems," *arXiv preprint arXiv:2101.02373,* 2021.

[12] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *2019 IEEE Symposium on Security and Privacy (SP)*, 2019: IEEE, pp. 691-706.

[13] A. Bhowmick, J. Duchi, J. Freudiger, G. Kapoor, and R. Rogers, "Protection against reconstruction and its applications in private federated learning," *arXiv preprint arXiv:1812.00984,* 2018.

[14] M. Roesch, "Snort: Lightweight intrusion detection for networks," in *Lisa*, 1999, vol. 99, no. 1, pp. 229-238.

[15] https://suricata-ids.org/ (accessed.

[16] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, "Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy," in *International conference on machine learning*, 2016: PMLR, pp. 201-210.
[17] M. Chen, R. Mathews, T. Ouyang, and F. Beaufays, "Federated learning of out-of-vocabulary words," *arXiv preprint arXiv:1903.10635,* 2019.
[18] M. Ammad-Ud-Din *et al.*, "Federated collaborative filtering for privacy-preserving personalized recommendation system," *arXiv preprint arXiv:1901.09888,* 2019.
[19] R. Canetti, U. Feige, O. Goldreich, and M. Naor, "Adaptively secure multi-party computation," in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, 1996, pp. 639-648.
[20] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *International conference on the theory and applications of cryptographic techniques*, 1999: Springer, pp. 223-238.
[21] C. Dwork, "Differential privacy: A survey of results," in *International conference on theory and applications of models of computation*, 2008: Springer, pp. 1-19.
[22] L. Xie, K. Lin, S. Wang, F. Wang, and J. Zhou, "Differentially private generative adversarial network," *arXiv preprint arXiv:1802.06739,* 2018.
[23] D. Preuveneers, V. Rimmer, I. Tsingenopoulos, J. Spooren, W. Joosen, and E. Ilie-Zudor, "Chained anomaly detection models for federated learning: An intrusion detection case study," *Applied Sciences,* vol. 8, no. 12, p. 2663, 2018.
[24] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A.-R. Sadeghi, "DÏoT: A federated self-learning anomaly detection system for IoT," in *2019 IEEE 39th International conference on distributed computing systems (ICDCS)*, 2019: IEEE, pp. 756-767.
[25] R. Zhao, Y. Yin, Y. Shi, and Z. Xue, "Intelligent intrusion detection based on federated learning aided long short-term memory," *Physical Communication,* vol. 42, p. 101157, 2020.
[26] B. Li, Y. Wu, J. Song, R. Lu, T. Li, and L. Zhao, "DeepFed: Federated deep learning for intrusion detection in industrial cyber–physical systems," *IEEE Transactions on Industrial Informatics,* vol. 17, no. 8, pp. 5615-5624, 2020.
[27] W. Stallings. "Cryptography and network security: principles and practice. Pearson Education." (accessed.
[28] A. Boyd. "OPM breach a failure on encryption, detection." https://www.federaltimes.com/smr/opm-data-breach/2015/06/19/opm-breach-a-failure-on-encryption-detection/ (accessed.
[29] R. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Communications,* 1978.

[30] X. Yi, R. Paulet, and E. Bertino, "Homomorphic encryption," in *Homomorphic encryption and applications*: Springer, 2014, pp. 27-46.
[31] C. Gentry, *A fully homomorphic encryption scheme*. Stanford university, 2009.
[32] M. v. Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully homomorphic encryption over the integers," in *Annual international conference on the theory and applications of cryptographic techniques*, 2010: Springer, pp. 24-43.
[33] A. López-Alt, E. Tromer, and V. Vaikuntanathan, "On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption," in *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, 2012, pp. 1219-1234.
[34] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," *arXiv preprint arXiv:1602.05629,* 2016.
[35] Q. Li, Z. Wen, and B. He, "Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection," 2019.
[36] S. Axelsson, "Intrusion detection systems: A survey and taxonomy," Citeseer, 2000.
[37] M. Rouse, "What is an intrusion detection system (ids) and how does it work," ed: Feb, 2020.
[38] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556,* 2014.