ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC VÀ KĨ THUẬT THÔNG TIN

BÁO CÁO ĐỒ ÁN
MÔN NHẬP MÔN ĐẢM BẢO VÀ AN NINH THÔNG TIN
Đề tài: Triển khai framework federated learning cho phân loại mã độc PE

GVHD: TS. Nguyễn Tấn Cầm
Nhóm sinh viên thực hiện:

1. Âu Trường Giang MSSV: 21522019
2. Nguyễn Nguyễn Thành An MSSV: 21521806

Tp. Hồ Chí Minh, 12/2023

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Thành phố Hồ Chí Minh, ngày 01 tháng 12 năm 2023

Người nhận xét

(Ký tên và ghi rõ họ tên)

BẢNG PHÂN CÔNG, ĐÁNH GIÁ THÀNH VIÊN

Bảng 1: Bảng phân công, đánh giá thành viên

Họ và tên	MSSV	Phân công	Đánh giá
Âu Trường Giang	21522019	- Làm báo cáo Word. - Tìm tài liệu tham khảo. - Chạy code triển khai,	Hoàn thành tốt, đóng góp 100%.
Nguyễn Nguyễn Thành An	21521806	- Tìm hiểu các mô hình. - Làm slide.	Hoàn thành tốt, đóng góp 100%.

DANH MỤC CÁC BẢNG, HÌNH ẢNH

TRIỂN KHAI FRAMEWORK FEDERATED LEARNING CHO PHÂN LOẠI MÃ ĐỘC PE

Deploy federated learning framework for PE malware classification

Âu Trường Giang[ab*], Nguyễn Nguyễn Thành An[ac*]

[a]Trường Đại học Công nghệ Thông tin, Đại học Quốc gia Thành phố Hồ Chí Minh

[b]21522019@gm.uit.edu.vn, [c]21521806@gm.uit.edu.vn

Abstract

Trong bài báo này, nhóm chúng tôi đề xuất triển khai framework federated learning để phân loại mã độc PE. Federated learning là một phương pháp học máy tiên tiến, cho phép mô hình học từ dữ liệu phân tán mà không cần phải chuyển dữ liệu về một trung tâm. Nghiên cứu giúp đóng góp cho việc bảo vệ sự riêng tư của dữ liệu và đồng thời tận dụng sức mạnh của dữ liệu phân tán từ nhiều nguồn.

Keywords: *PE file, Malware, Federated learning framework, Classification.*

1. Introduction

Ngày nay, với sự bùng nổ của công nghệ thông tin và việc sử dụng ngày càng phổ biến của Internet, rủi ro về an ninh mạng ngày càng trở nên nghiêm trọng. Trong số các mối đe dọa này, mã độc PE (Portable Executable) là một loại mã độc nguy hiểm đặc biệt, thường được sử dụng để thực hiện các cuộc tấn công mục tiêu hệ điều hành Windows. Để ngăn chặn và phòng tránh các cuộc tấn công này, cần có các phương pháp hiệu quả trong việc phân loại và phát hiện mã độc PE.

Các phương pháp truyền thống để phân loại phần mềm độc hại thường liên quan đến việc tập trung các bộ dữ liệu lớn để đào tạo các mô hình học máy. Tuy nhiên, mô hình tập trung này đưa ra những lo ngại về quyền riêng tư và an ninh dữ liệu. Việc triển khai các mô hình học máy phụ thuộc vào các bộ dữ liệu tập trung có thể tiết lộ thông tin nhạy cảm đến rủi ro bị xâm phạm, đe dọa chính quyền riêng tư mà những mô hình này nhằm bảo vệ. Hơn nữa, tính thay đổi liên tục của phần mềm độc hại đòi hỏi mô hình phải thích ứng nhanh chóng, một thách thức được làm nặng thêm bởi quá trình đào tạo tập trung cứng nhắc và tốn thời gian.

Federated learning là một khái niệm ngày càng trở nên quan trọng, đặc biệt là khi cần phải xử lý dữ liệu nhạy cảm và tách biệt trên nhiều thiết bị hoặc nơi lưu trữ. Federated learning cho phép mô hình học máy được đào tạo trên các thiết bị địa phương mà không cần chuyển gửi dữ liệu về một trung tâm trung ương, giảm thiểu rủi ro về bảo mật và bảo vệ quyền riêng tư.

Đề tài này tập trung vào triển khai framework federated learning để phân loại mã độc PE. Việc áp dụng federated learning vào bài toán này mang lại nhiều lợi ích, bao gồm sự bảo mật cao hơn, giảm bớt áp lực về băng thông mạng, và tăng tính quản lý quyền riêng tư cho người dùng. Chúng ta sẽ khám phá cách triển khai và tối ưu hóa framework federated learning để đảm bảo hiệu suất cao trong việc phân loại mã độc PE, đồng thời đảm bảo tính an toàn và bảo mật của hệ thống.

2. Related works

Vào năm 2019, Irina Baptista, Stavros Shiaele, Nicholas Kolokotronis đã đề xuất phương pháp phương pháp được sử dụng để phát triển hệ thống phát hiện phần mềm độc hại dựa vào việc hiển thị nhị phân của tệp trên không gian hai chiều và sau đó sử dụng SOINN, sau khi thực hiện trích xuất tính năng, để phân biệt giữa các tệp lành tính và độc hại.[1]

Vào năm 2020, Tina Rezaei và Ali Hamze[2] đã đề xuất một phương pháp hiệu quả để phát hiện malware bằng cách sử dụng các thông số PE Header. PE Header (Portable Executable Header) là một thành phần quan trọng trong các tệp tin thực thi của hệ điều hành Windows. Bằng cách phân tích các thông số này, bài nghiên cứu đề xuất một cách tiếp cận mới để xác định và ngăn chặn malware một cách hiệu quả.

Vào tháng 5 năm 2020, Wei-Ting Chen, Wei-Chih Huang, và Chi-Yuan Huang đề xuất một khuôn khổ crowdsourcing mới cho federated learning (FL). Khi các thiết bị tham gia triển khai chiến lược tính toán không được phối hợp, khó khăn là xử lý hiệu quả giao tiếp (tức là số lần giao tiếp mỗi vòng lặp) trong khi trao đổi các tham số mô hình trong quá trình tổng hợp. Do đó, một thách thức quan trọng trong FL là làm thế nào người dùng tham gia để xây dựng một mô hình toàn cầu chất lượng cao với hiệu quả giao tiếp.[3]

Vào tháng 6 năm 2020, phương pháp phát hiện mã độc sử dụng đám mây và học máy (cloud computing and machine learning-based malware detection) được đề xuất bởi các nhà nghiên cứu tại Đại học Công nghệ Nanyang, Singapore. Phương pháp này sử dụng kết hợp các kỹ thuật phân tích đám mây và học máy để phát hiện mã độc hiệu quả hơn, ngay cả khi mã độc đó được thiết kế để tránh các phương pháp phát hiện truyền thống. Kỹ thuật phân tích đám mây được sử dụng để phân tích dữ liệu lớn về mã độc và mã sạch. Kỹ thuật học máy được sử dụng để đào tạo các mô hình học máy để phân loại mã độc và mã sạch. Các mô hình học máy được đào tạo trên dữ liệu đã được phân tích bởi kỹ thuật phân tích đám mây.[4]

Tháng 7 năm 2020, DJ Beutel, T Topal, A Mathur, X Qiu đã có bài giới thiệu về Flower,[5] một framework nghiên cứu về Federated Learning (FL) được thiết kế để hỗ trợ việc thực hiện các nghiên cứu trong lĩnh vực FL một cách thuận lợi và hiệu quả. Nhóm các tác giả đã mở rộng FL đến 15 triệu client, so sánh giữa các framework FL trong môi trường đơn máy tính, đo lường năng lượng tiêu thụ trên cụm thiết bị Nvidia Jetson TX2, tối ưu hóa thời gian hội tụ với băng thông hạn chế, và triển khai Flower trên các thiết bị di động Android trong AWS Device Farm.

Năm 2021, Priyanka Mary Mammen từ Trường Đại học Massachusetts, Amherst đã nói federated learning (FL) mở ra nhiều cơ hội trong các lĩnh vực quan trọng như chăm sóc sức khỏe, tài chính, v.v.,[6] nơi việc chia sẻ thông tin người dùng riêng tư với các tổ chức hoặc thiết bị khác có rủi ro. Mặc dù FL có vẻ là một kỹ thuật machine learning hứa hẹn đảm bảo dữ liệu cục bộ riêng tư, nhưng nó cũng dễ bị tấn công giống như các mô hình ML khác.

Năm 2021, Nureni Ayofe Azeez, Oluwanifise Ebunoluwa Odufuwa, Sanjay Misra, Jonathan Oluranti, andRobertas Damaševičius cũng đã đưa ra Phương pháp phát hiện mã độc Windows PE sử dụng học tập tập hợp. Các nhà nghiên cứu đã đánh giá hiệu quả của hệ thống này bằng cách sử dụng một tập dữ liệu gồm 100.000 tệp PE, bao gồm 50.000 tệp mã độc và 50.000 tệp mã sạch. Kết quả cho thấy hệ thống này đạt được độ chính xác phát hiện mã độc là 99,8% và tỷ lệ phát hiện sai là 0,2%.[7]

Năm 2022, nhóm tác giả Valerian Rey, Pedro Miguel Sánchez Sánchez, Alberto Huertas Celdrán và Gérôme Bovet đã thử nghiệm và đánh giá hiệu suất của Federated Learning trong việc phát hiện và ngăn chặn malware trên các thiết bị IoT. Độ chính xác phát hiện được đo bằng tỷ lệ số thiết bị IoT bị nhiễm được phát hiện thành công. Phương pháp đạt được độ chính xác 99% trên một bộ dữ liệu gồm 10.000 thiết bị IoT, trong đó có 1.000 thiết bị bị nhiễm phần mềm độc hại.[8]

Tháng 10 năm 2022, Leon Witt và các cộng sự đã phân tích các khung FL áp dụng cả công nghệ blockchain để phân tán quá trình và cơ chế phần thưởng để khích lệ tham gia trong bài "Decentral and Incentivized Federated Learning Frameworks: A Systematic Literature Review". Tìm kiếm từ 12 cơ sở dữ liệu khoa học lớn đã thu được 422 công trình. Sau quá trình lọc, 40 bài viết được xem xét sâu hơn. Bài báo cung cấp một cách tiếp cận hệ thống để phân loại và đo lường sự khác biệt giữa các khung FL, tiết lộ hạn chế của các công trình hiện tại và đề xuất hướng nghiên cứu tương lai cho lĩnh vực mới này.[9]

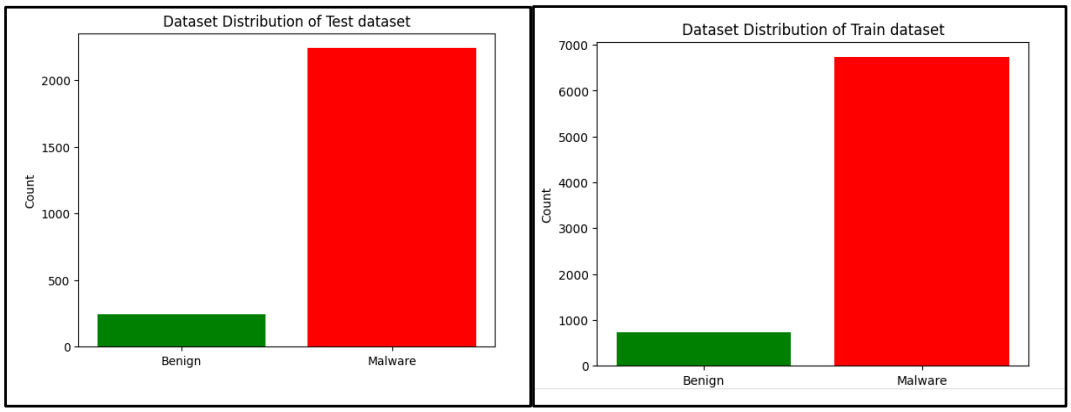
Tháng 9 năm 2023, Dimitrios Serpanos và Georgios Xenos trong bài báo "Federated Learning in Malware Detection" đã phát biểu về FL kết hợp các tổ chức để phát triển DNN phát hiện phần mềm độc hại mà không chia sẻ dữ liệu.[10] Mỗi tổ chức huấn luyện DNN trên dữ liệu riêng. FL kết hợp DNN cục bộ thành DNN toàn cầu. Dữ liệu từ nhiều nguồn được tổng hợp mà không chia sẻ mẫu hoặc đặc trưng. Sử dụng tập EMBER, phương pháp này đạt độ chính xác trên 93%. Mô hình kết hợp tăng cường phát hiện phần mềm độc hại. Quyền riêng tư của các tổ chức được đảm bảo trong quá trình. FL mở ra cách mới để bảo vệ dữ liệu riêng và tối ưu hóa hiệu suất.

3. Proposed system

Nhóm đã sử dụng bộ dữ liệu Dike Dataset[1], nơi có những file PE cho đề tài nghiên cứu.

3.1. Thống kê dữ liệu

Bộ dữ liệu ban đầu của bao gồm 1082 dữ liệu về Benign và 10841 dữ liệu về Malware. Trong đó, nhóm đã chia thành 20% cho Test và 80% cho Train.



Hình 1: Bộ dữ liệu tiền xử lý

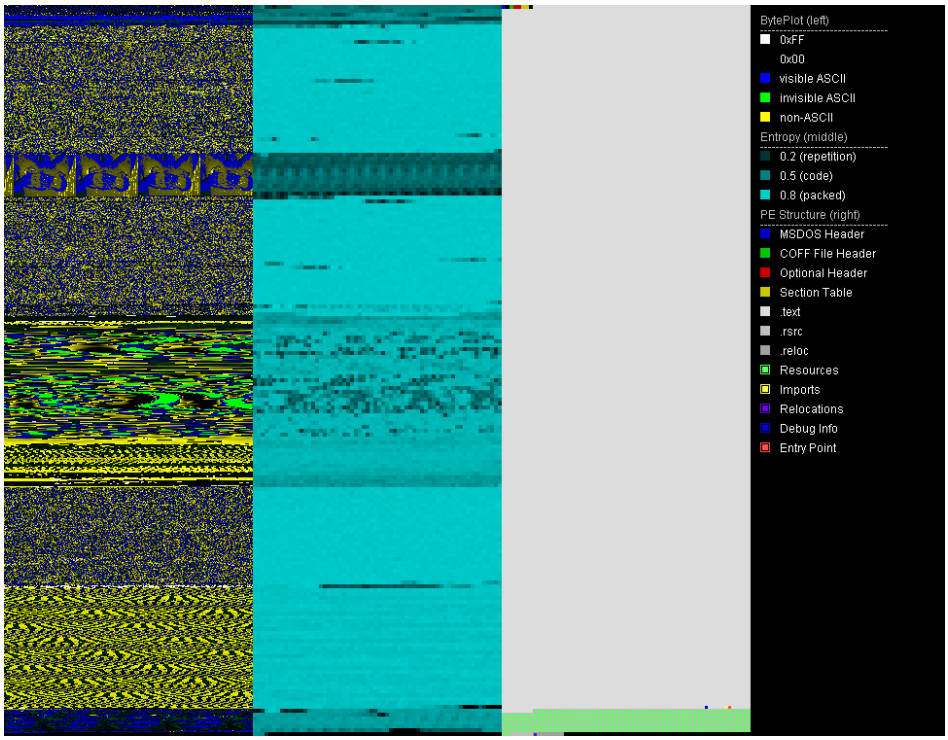
Nhóm chúng tôi có một bộ dữ liệu không cân bằng 11923 được thể hiện như bảng 2.

Bảng 2: Thông tin bộ dữ liệu

	Bộ dữ liệu
Tổng số dữ liệu	11923
Dữ liệu Benign	1082
Dữ liệu Malware	10841
Malware cân bằng	Không

3.2. Tiền xử lý dữ liệu

Chúng tôi quyết định sử dụng công cụ PortEx[1], một thư viện của Java để biến các file PE thành ảnh.



Hình 2: Hình ảnh một file PE được chuyển thành ảnh bằng PortEx

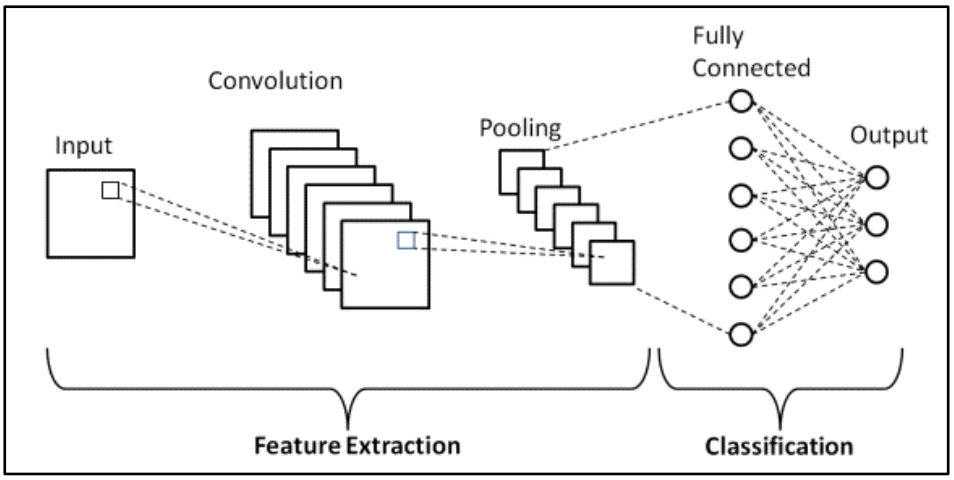
Từ các file ảnh, chúng tôi tiếp tục tìm hiểu các mô hình học máy để áp dụng các thuật toán nhận diện file là loại Malware hay Benign thông qua phân tích các điểm ảnh.

3.3. Lựa chọn mô hình

CNN (Convolutional Neural Network)[11]**:** mô hình học máy có giám sát được sử dụng cho các nhiệm vụ nhận dạng hình ảnh và video. CNN được thiết kế để học các mẫu trong dữ liệu hình ảnh (các cạnh, đường cong, các đối tượng,...).

[1] https://github.com/struppigel/PortEx

CNN hoạt động bằng cách sử dụng các bộ lọc (filter) để quét qua dữ liệu hình ảnh. Các bộ lọc này hoạt động như các bộ cảm biến, tìm kiếm các mẫu cụ thể trong dữ liệu hình ảnh. CNN có một số lớp, mỗi lớp có một số bộ lọc. Các lớp ở phía trước của CNN thường có các bộ lọc nhỏ, tìm kiếm các mẫu đơn giản. Các lớp ở phía sau của CNN thường có các bộ lọc lớn hơn, tìm kiếm các mẫu phức tạp hơn.

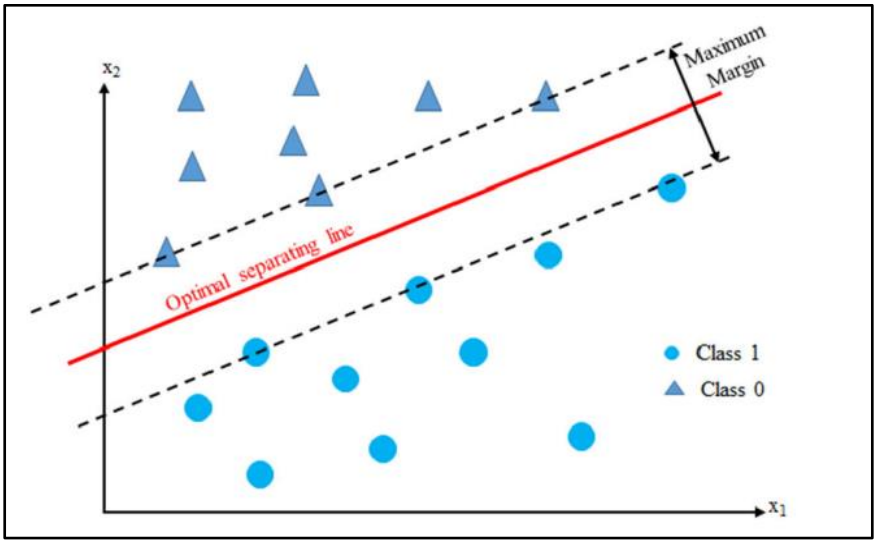


Hình 3: Mô hình CNN

Mô hình SVMs (Support Vector Machines)[12]: mô hình máy học phân loại và hồi quy. SVMs hoạt động bằng cách tìm một ranh giới quyết định tối ưu giữa hai lớp dữ liệu.

Ranh giới quyết định này được chọn sao cho các điểm dữ liệu của mỗi lớp càng xa ranh giới càng tốt.

Các điểm dữ liệu gần nhất với ranh giới quyết định được gọi là các vector hỗ trợ. SVMs đã được sử dụng thành công trong nhiều nhiệm vụ phân loại, bao gồm phân loại mã độc qua hình ảnh.



Hình 4: Mô hình SVMs

Chúng tôi chọn CNN để triển khai vì nó phù hợp với các tập dữ liệu khác nhau, có thể học các mẫu phức tạp hơn so với SVMs. Ngoài ra, theo tìm hiểu nhóm biết được SVMs có thể nhanh hơn nhưng độ chính xác không thể bằng CNN.

3.4. Xây dựng CNN đơn giản

Nhóm chúng tôi đã tìm hiểu một CNN đơn giản trích từ tài Flower framework nhằm để học đặc trưng từ đặc điểm của các tệp thực thi PE và được huấn luyện trên các clients phân tán mà không cần truyền tải dữ liệu độc lập về trung tâm. Trong quá trình thiết lập mô hình có những công thức tính như sau:

- Công thức tính mất mát (Cross-Entropy Loss):

$$loss(x, class) = -log(\frac{e^{x}class}{\sum_j e^{x}j}) \quad (1)$$

Trong đó:

- *x* là đầu ra của mô hình.
- *class* là lớp thực tế.

- Công thức tính đạo hàm cho quá trình lan truyền ngược (backpropagation):

  Đạo hàm của loss theo đầu ra được tính bằng hàm softmax và one-hot encoding của lớp thực tế:

$$\frac{\partial loss}{\partial x} = Softmax(x) - OneHot(class) \qquad (2)$$

  Các đạo hàm liên quan đến trọng số và bias của mỗi tầng được tính thông qua lan truyền ngược thông thường.

- Công thức tính hàm đơn vị tuyến tính chỉnh lưu (ReLU):

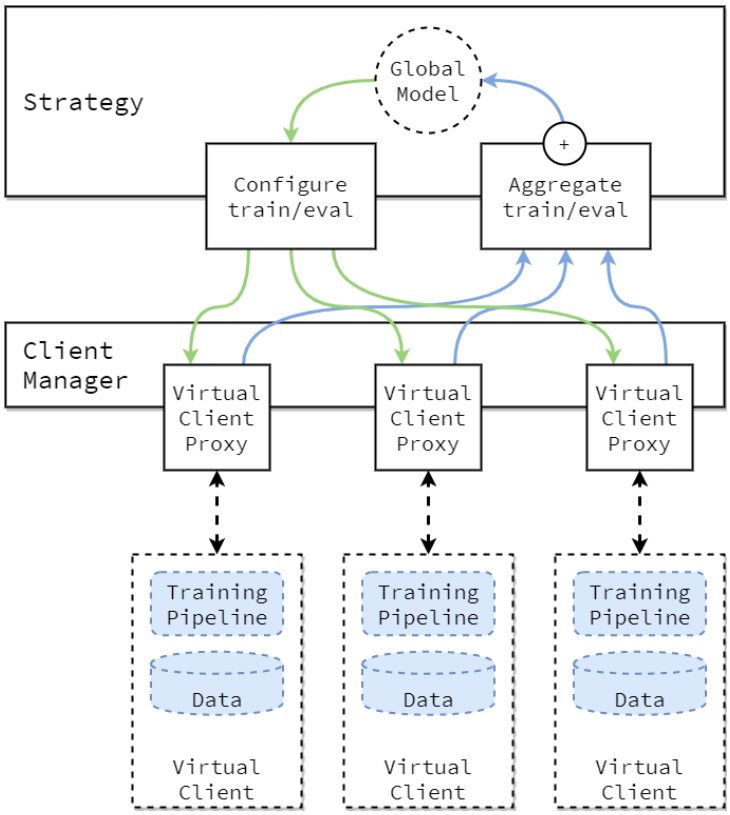
$$ReLU(x) = max(0, x) \qquad (3)$$

- Công thức tính gộp tối đa (MaxPooling):

$$MaxPooling(x, kernel\_size, stride) \qquad (4)$$

  Với mỗi phần tử trong vùng kernel, chọn giá trị lớn nhất.

3.5. Virtual Client Engine



Hình 5: Kiến trúc Flower cho federated learning

Virtual Client Engine (VCE): Công cụ được tích hợp cho phép ảo hóa Flower Clients để tối đa hóa việc sử dụng phần cứng có sẵn. VCE là một mô-đun chính trong framework Flower, giúp chạy các workload FL quy mô lớn với chi phí tối thiểu.

VCE sẽ tải một mô hình Neural Network, chọn dữ liệu huấn luyện và kiểm tra cụ thể của client, và tạo ra một client Flower với cấu hình này.

3.6. Môi trường sử dụng

Nhóm đã sử dụng môi trường được cung cấp sẵn bởi Google Colab để triển khai các thuật toán.

Chi tiết thông số hệ thống như sau:

- Hệ điều hành: Ubuntu 22.04.3 LTS
- Bộ nhớ: 79 GB
- RAM: 12982 MB
- CPU: Intel(R) Xeon(R) CPU @ 2.30GHz
- GPU: NVIDIA Tesla T4

3.7. Thông số huấn luyện

Nhóm đã quyết định huấn luyện bằng mô hình CNN với các thông số như sau:

- Epoch: 5
- Num clients: 10
- Batch size: 24
- Height: 224
- Width: 224
- Val size: 0.1
- Training set: 80%
- Test set: 20%

3.8. Federated averaging (FedAvg)

- Mô phỏng bắt đầu với cấu hình cho 5 vòng lặp (num_rounds=5) mà không giới hạn thời gian cho mỗi vòng lặp (round_timeout=None).
- Một phiên bản Ray local được khởi tạo để hỗ trợ việc phân phối tính toán trên nhiều thiết bị với một số tài nguyên nhất định, bao gồm CPU, GPU, bộ nhớ, và các nguồn tài nguyên khác.

- Cung cấp hướng dẫn để tối ưu hóa quá trình mô phỏng sử dụng Flower VCE. Mỗi client được cấu hình với 1 GPU và 1 CPU.
- Flower tạo một pool của các Virtual Client Engine Actor với 1 actor.
- Tham số toàn cục được khởi tạo trước khi bắt đầu quá trình FL.Server yêu cầu tham số ban đầu từ một client ngẫu nhiên trong hệ thống và đánh giá chúng.
- Đầu tiên, chiến lược FedAvg chọn một số lượng client (ở đây là 10) để tham gia vào quá trình huấn luyện.
- Cho mỗi vòng lặp huấn luyện, mỗi client thực hiện một số công việc huấn luyện và cập nhật trọng số mô hình. Mỗi client mất khoảng 12-13 giây cho mỗi vòng lặp. Sau đó mô hình sẽ được debug nếu cần.

4. Evaluation

4.1. Độ đo Accuracy

Độ đo Accuracy là chỉ số hiệu suất được sử dụng trong học máy để đánh giá độ chính xác của một mô hình. Accuracy được tính bằng cách chia số lượng dự đoán chính xác cho tổng số dự đoán.

$$Accuracy = \frac{Số\ lượng\ dự\ đoán\ chính\ xác}{Tổng\ số\ dự\ đoán}$$

4.2. Độ đo F1

Độ đo F1 là một chỉ số hiệu suất được sử dụng trong học máy để đánh giá hiệu suất của một mô hình phân loại. Độ đo F1 được tính bằng cách lấy trung bình của Precision và Recall.

$$\boldsymbol{F1} = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

+ Trong đó Precision là một chỉ số hiệu suất được sử dụng trong học máy để đánh giá hiệu suất của một mô hình phân loại. Độ đo Precision được tính bằng cách lấy tỷ lệ các positive đúng (TP) so với dự đoán..

$$Precision = \frac{TP}{TP + FP}$$

TP là số lượng dự đoán positive thực sự của mô hình là positive.

FP là số lượng dự đoán positive sai của mô hình.

+ Trong đó Recall là một chỉ số hiệu suất được sử dụng trong học máy để đánh giá hiệu suất của một mô hình phân loại. Độ đo Recall được tính bằng cách lấy tỷ lệ TP so với thực tế..

$$Recall = \frac{TP}{TP + FN}$$

TP là số lượng dự đoán positive thực sự của mô hình là positive.

FN là số lượng đối tượng positive thực sự bị mô hình bỏ sót.

4.3. Bảng so sánh

Bảng 3: So sánh chi tiết các độ đo

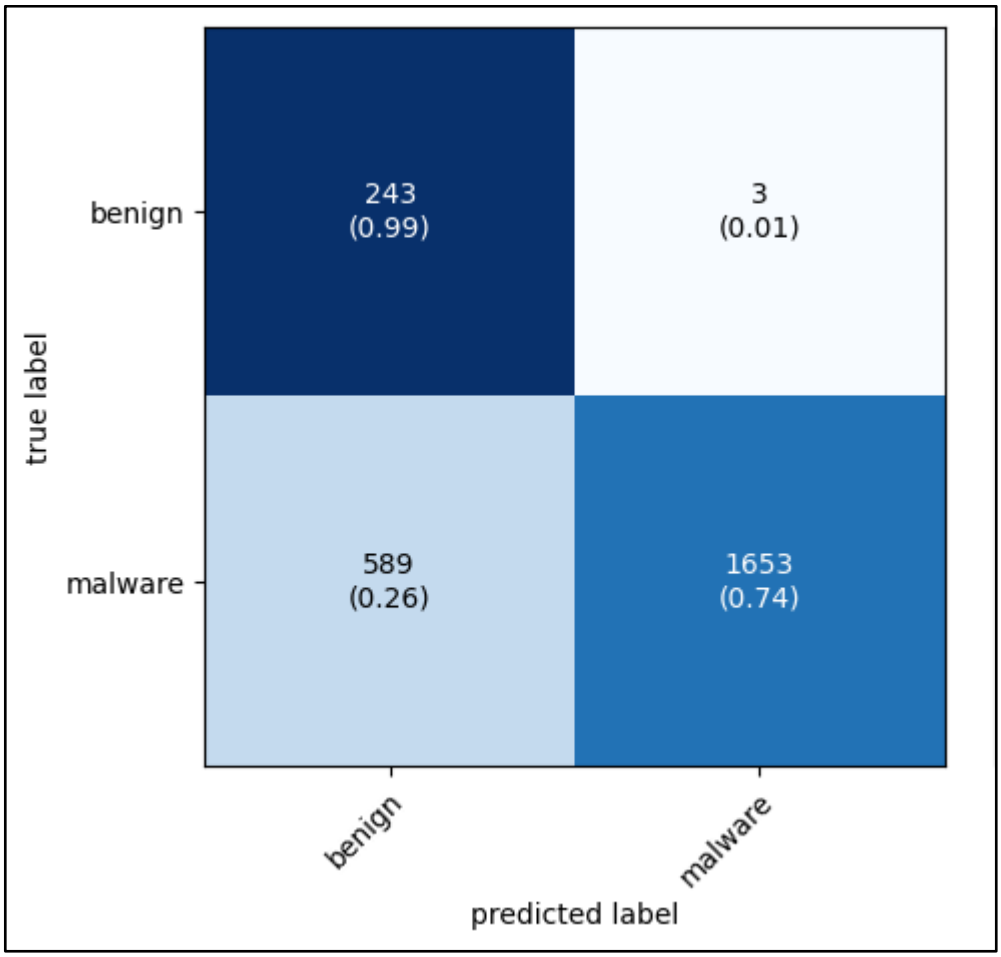
	Precision	Recall	F1-Score	Accuracy
Mô hình gốc	0.81	0.90	0.85	0.90
10 client	0.92	0.96	0.94	0.96

Qua bảng trên phương pháp sử dụng Flower client cho thấy điểm số gần như cao hơn ở mọi khía cạnh. Ban đầu thì Precision chỉ có 81% và F1-Score là 85% nhưng sau khi sử dụng phương pháp thì Precision và F1-Score đã tăng mạnh lần lượt là 11% và 9%. Ngoài ra Recall và Accuracy cũng tăng nhẹ 6% khi từ 90% lên 96%. Cuối cùng, so sánh chi tiết cho thấy phương pháp sử dụng Flower client có nhiều ưu điểm khiến cho ra được kết quả tốt hơn so với mô hình gốc.

Tuy nhiên, trong quá trình kiểm nghiệm vô số lần, không phải lúc nào cũng cho ra những chỉ số cao như trên. Thực tế khi chạy mô hình tập trung mà không cần FL vẫn có thể chạy ra các độ đo tốt.

Dù vậy, FL lại có một số lợi ích như: tiết kiệm băng thông, giảm độ trễ (latency), phân phối và cập nhật mô mô hình nhanh chóng, ổn định. Federated learning theo nhiều bài báo nghiên cứu cho thấy nó phù hợp với mô hình trên ứng dụng di động và IoT. Ngoài ra, do đặc tính không tập trung nên được cho là một giải pháp tốt cho các yêu cầu về bảo mật.

4.4. Confusion matrix



Hình 6: Confusion matrix của mô hình huấn luyện

Dữ liệu trong biểu đồ cho thấy rằng, đối với các nhãn "benign", có 243 nhãn dự đoán chính xác và 3 nhãn dự đoán không chính xác. Tỷ lệ dự đoán chính xác là 99%. Đối với các nhãn "malware", có 589 nhãn dự đoán không chính xác và 1653 nhãn dự đoán chính xác. Tỷ lệ dự đoán chính xác là 26%.

5. Conclusion

Sau khi triển khai, nhóm đã giới thiệu được một framework của phương pháp federated learning. Thông qua việc sử dụng thêm mô hình CNN đã thu được những kết quả khả quan.

Bằng cách phân phối quá trình học máy trên các thiết bị phân tán và tổng hợp cập nhật mô hình thông qua phương pháp Flower client, framework đã mang lại hiệu suất và tính chính xác khá hiệu quả đối với việc nhận diện mối đe dọa.

Thông qua đề tài, nhóm hy vọng rằng nghiên cứu của mình sẽ đóng góp phần nào cho việc phát triển framework cho mô hình học liên kết (federated learning).

REFERENCES

[1] *Irina Baptista, Stavros Shiaele, Nicholas Kolokotronis. A Novel Malware Detection System Based on Machine Learning and Binary Visualization. In: 2019 IEEE International Conference on Communications Workshops (ICC Workshops). IEE, 2019.*

[2] *REZAEI, Tina; HAMZE, Ali. An efficient approach for malware detection using PE header specifications. In: 2020 6th International Conference on Web Research (ICWR). IEEE, 2020. p. 234-239.*

[3] *Pandey, Shashi Raj, et al. "A crowdsourcing framework for on-device federated learning." IEEE Transactions on Wireless Communications 19.5 (2020): 3241-3256.*

[4] *Dr. P. Indirapriyadarsini, Mohammed Uzair Mohiuddin, Mohammed Taqueeuddin, Ch Srikanth Reddy, T Koushik. Malware Detection using Machine Learning and Cloud Computing. ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429*

[5] *BEUTEL, Daniel J., et al. Flower: A friendly federated learning research framework. arXiv preprint arXiv:2007.14390, 2020.*

[6] *Mammen, Priyanka Mary. "Federated learning: Opportunities and challenges." arXiv preprint arXiv:2101.05428 (2021).*

[7] *Nureni Ayofe Azeez, Oluwanifise Ebunoluwa Odufuwa, Sanjay Misra, Jonathan Oluranti, andRobertas Damaševičius. Windows PE Malware Detection Using Ensemble Learning. Informatics 2021, 8(1), 10*

[8] *REY, Valerian, et al. Federated learning for malware detection in IoT devices. Computer Networks, 2022, 204: 108693.*

[9] *Witt, Leon, et al. "Decentral and incentivized federated learning frameworks: A systematic literature review." IEEE Internet of Things Journal (2022).*

[10] *Serpanos, Dimitrios, and Georgios Xenos. "Federated Learning in Malware Detection." 2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA). IEEE, 2023.*

[11] *Tianmei Guo; Jiwen Dong; Henjian Li; Yunxing Gao. Simple convolutional neural network on image classification. 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA).*

[12] *Lahouari Ghouti, Muhammad Imam. Malware classification using compact image features and multiclass support vector machines. Volume14, Issue4, Pages 419-429*