

Tìm hiểu học cộng tác trong phân tích mã độc trên các thiết bị IoT

Tác giả : Lê Tấn Lộc, Nguyễn Gia Huy và Vũ Tùng Lâm

|  |  |  |
| --- | --- | --- |
| Thông tin bài báo. |  | tóm tắt |
|  |  | Hàng tỷ thiết bị IoT thiếu cơ chế bảo mật phù hợp đã được sản xuất và triển khai trong những năm qua, và nhiều hơn nữa sẽ đến với sự phát triển của công nghệ 5G. Việc dễ bị tấn công của chúng đối với phần mềm độc hại đã thúc đẩy nhu cầu về các kỹ thuật hiệu quả để phát hiện các thiết bị IoT bị tấn công. Với sự riêng tư và tính toàn vẹn của dữ liệu đang trở thành mối quan tâm lớn trong những năm gần đây, các công nghệ mới như học tập liên kết và blockchain đã xuất hiện. Chúng cho phép đào tạo các mô hình học máy với dữ liệu phi tập trung trong khi vẫn bảo toàn tính riêng tư của nó theo thiết kế. Bài báo này nghiên cứu các khả năng được kích hoạt bởi học tập liên kết liên quan đến phát hiện phần mềm độc hại IoT và nghiên cứu các vấn đề bảo mật vốn có trong mô hình học tập mới này. Trong bối cảnh này, một khuôn khổ sử dụng tính năng học tập liên kết để phát hiện phần mềm độc hại ảnh hưởng đến các thiết bị IoT được trình bày. N-BaIoT, một tập dữ liệu mô hình hóa lưu lượng mạng của một số thiết bị IoT thực trong khi bị ảnh hưởng bởi phần mềm độc hại, đã được sử dụng để đánh giá khung đề xuất. Cả hai mô hình liên hợp được giám sát và không được giám sát (perceptron nhiều lớp và autoencoder) có thể phát hiện phần mềm độc hại ảnh hưởng đến các thiết bị IoT nhìn thấy và không nhìn thấy của N-BaIoT đều đã được đào tạo và đánh giá. Bên cạnh đó, các mô hình liên kết, trong khi bảo vệ quyền riêng tư của người tham gia, hiển thị kết quả tương tự như các mô hình tập trung. Như một đóng góp bổ sung và để đo lường mức độ mạnh mẽ của phương pháp liên kết, một thiết lập đối đầu với một số người tham gia độc hại đầu độc mô hình liên kết đã được xem xét. Bước trung bình tổng hợp mô hình cơ sở được sử dụng trong hầu hết các thuật toán học liên hợp dường như rất dễ bị tấn công bởi các cuộc tấn công khác nhau, ngay cả với một kẻ thù duy nhất. Do đó, hiệu suất của các chức năng tổng hợp mô hình khác hoạt động như các biện pháp đối phó được đánh giá theo các kịch bản tấn công tương tự. Các chức năng này cung cấp một cải tiến đáng kể chống lại những người tham gia độc hại, nhưng vẫn cần nhiều nỗ lực hơn để làm cho các phương pháp liên kết trở nên mạnh mẽ. |
| Từ khoá |
| *IoT Security, Federated Learning, IoT Device, Botnet Detection, Adversarial Attack* |

# giới thiệu

Đến năm 2025, các dự báo ước tính rằng sẽ có khoảng 64 tỷ thiết bị IoT trực tuyến [[1]](#mot). Việc triển khai ồ ạt các thiết bị này chắc chắn đang biến thế giới thành một môi trường siêu kết nối. Mô hình IoT, cùng với các công nghệ mạng 5G đang cho phép các mô hình công nghệ mới chưa từng thấy trước đây, chẳng hạn như Công nghiệp 4.0 và Thành phố thông minh [[2]](#hai). Tuy nhiên, đồng thời với những tiến bộ của công nghệ mới, số lượng và sự đa dạng của các cuộc tấn công mạng đã tăng lên trong những năm gần đây, khiến các phương pháp bảo mật hiện tại trở nên lạc hậu trong một thời gian ngắn [[3]](#ba). Vì lý do này, việc kiểm soát an ninh của các môi trường mạng trong tương lai đưa ra những thách thứ lớn phải được giải quyết bằng các kỹ thuật hiện đại.

Một trong những cách phổ biến khi các thiết bị đã bị phần mềm độc hại làm hỏng kẻ tấn công sẽ giám sát các hoạt động của thiết bị để lấy đi dấu vân tay hoặc thông tin cá nhân. Dấu vân tay có thể được tận dụng để tìm các sai lệch gây ra bởi các cuộc tấn công mạng hoặc các sửa đổi phần mềm độc hại [[4]](#bon). Trong các thiết bị IoT , các xử lý bất đồng bộ có thể bị giám sát, chẳng hạn như giao tiếp qua mạng, sử dụng tài nguyên, các hành động và sự kiện phần mềm hoặc tương tác của người dùng. Do đó, tùy thuộc vào mục tiêu cần đạt được mà có thể sử dụng cái này hay cái khác. Cụ thể, khi nói đến việc phát hiện các cuộc tấn công mạng, khía cạnh được sử dụng rộng rãi nhất trong tài liệu là giao tiếp qua mạng [[5]](#nam).

Sau khi các thiết bị bị giám sát, bước tiếp theo để các phần mềm độc hại tấn công một cách thành công là xử lý dữ liệu và tạo dấu vân tay hành vi của thiết bị.

* 1. **Federated Learning**

Trong bối cảnh 5G, công nghệ Trí tuệ nhân tạo (AI), chủ yếu là Machine Learning (ML) and Deep Learning (DL) đã đạt được mức độ liên quan to lớn trong những năm gần đây. Ngày nay, hầu hết các giải pháp hiện có sử dụng ML / DL để phát hiện phần mềm độc hại đều dựa vào một thực thể trung tâm chịu trách nhiệm thu thập dữ liệu từ các thiết bị khác nhau và đào tạo các mô hình toàn cầu. Sau đó, các mô hình này được phân phối giữa các máy khách riêng lẻ hoặc các máy khách này gửi dữ liệu thử nghiệm trực tiếp của họ đến máy chủ để đánh giá hành vi và phát hiện phần mềm độc hại. Tuy nhiên, cách tiếp cận này không phù hợp với các tình huống mà hành vi của thiết bị chứa dữ liệu nhạy cảm hoặc bí mật sẽ ảnh hưởng đáng kể đến an ninh môi trường và sự riêng tư trong trường hợp rơi vào tay ma licious. Tình huống tương tự cũng xảy ra trong các tình huống mà các nguồn dữ liệu được giám sát có liên quan đến con người và các hành động riêng tư có liên quan.

Khái niệm về học liên kết lần đầu tiên được các nhà nghiên cứu tại Google đưa ra vào đầu năm 2017. Trong năm qua, mối quan tâm đến học liên kết đã bùng nổ: hơn 1.000 bài nghiên cứu về học liên kết đã được xuất bản trong sáu tháng đầu năm 2020, so với chỉ 180 bài. trong tất cả năm 2018.

Cách tiếp cận tiêu chuẩn để xây dựng mô hình học máy ngày nay là tập hợp tất cả dữ liệu đào tạo ở một nơi, thường là trên đám mây, và sau đó đào tạo mô hình trên dữ liệu. Nhưng cách tiếp cận này không khả thi đối với phần lớn dữ liệu trên thế giới, vì lý do riêng tư và bảo mật không thể chuyển đến kho lưu trữ dữ liệu trung tâm. Điều này làm cho nó vượt quá giới hạn đối với các kỹ thuật AI truyền thống.

Thay vì yêu cầu một tập dữ liệu thống nhất để đào tạo một mô hình, việc học liên kết sẽ để dữ liệu ở vị trí của nó, được phân phối trên nhiều thiết bị và máy chủ. Thay vào đó, nhiều phiên bản của mô hình được gửi đi — một đến mỗi thiết bị có dữ liệu huấn luyện — và được huấn luyện cục bộ trên mỗi tập con dữ liệu. Các tham số mô hình kết quả, nhưng không phải dữ liệu đào tạo, sau đó được gửi trở lại đám mây. Khi tất cả các “mô hình nhỏ” này được tổng hợp, kết quả là một mô hình tổng thể hoạt động như thể nó đã được đào tạo trên toàn bộ tập dữ liệu cùng một lúc.

* 1. **Tình huống thiết bị IoT bị nhiễm độc**

Các thành phố của chúng ta có hàng triệu thiết bị IoT được kết nối với Internet và cảm nhận các phần dữ liệu không đồng nhất. Số lượng và tính không đồng nhất của các thiết bị sẽ tăng lên theo cấp số nhân với sự ra đời của mạng 5G [[6]](#sau). Trong bối cảnh như vậy, các vấn đề về quyền riêng tư thường xuyên xuất hiện khi các phần dữ liệu nhạy cảm được chia sẻ trong cuộc sống hàng ngày hoặc tổ chức của chúng ta [[7]](#bay). Khi bị tấn công , các thiết bị IoT bị hạn chế về tài nguyên và không được thiết kế với tính bảo mật, khiến chúng dễ bị tấn công bởi nhiều loại phần mềm độc hại. Trong các tình huống này, các phương pháp tiếp cận phát hiện dựa trên AI truyền thống không phù hợp do không thể đào tạo các mô hình tập trung với dữ liệu nhạy cảm thuộc các tổ chức hoặc đối tượng khác nhau.

Do đó, FL đang phát triển như một cơ chế chính để phát hiện các hành vi bất thường và kích hoạt các cơ chế giảm thiểu trong các tình huống nhạy cảm về quyền riêng tư được kích hoạt bởi mạng 5G.

Một số bộ dữ liệu công khai phù hợp với các kịch bản ứng dụng 5G và phần mềm độc hại IoT tồn tại trong tài liệu. Trong số đó, N-BaIoT là phù hợp nhất để đánh giá đào tạo hợp tác bảo vệ quyền riêng tư . Cụ thể, tập dữ liệu này đã phân tách dữ liệu lưu lượng truy cập của các thiết bị IoT thành các tệp khác nhau, giúp dễ dàng chia nó thành nhiều phần không được phân phối giống nhau cho một cài đặt liên kết thực tế. Vì lý do đó, tập dữ liệu N-BaIoT đã được chọn để đánh giá. Lưu ý rằng một hạn chế của tập dữ liệu này là nó chỉ chứa dữ liệu từ 9 thiết bị IoT, đây là một hạn chế đối với các thử nghiệm vì nó giới hạn số lượng khách hàng tối đa có thể được xem xét. N-BaIoT chứa các gói được xử lý trước từ lưu lượng của 9 thiết bị IoT. Tất cả các thiết bị đã tạo ra một số lưu lượng truy cập trong khi không bị hỏng (mẫu lành tính) và trong khi được Mirai và BASHLITE kiểm tra thông tin.

# triển khai và chi tiết về federated learning-based framework

Phần này sẽ trình bày chi tiết về chi tiết FL-based framework, miêu tả những thành phần của nó và cách chúng tương tác với nhau trong suốt quá trình Training model(Huấn luyện mô hình) và evaluation processes(quy trình đánh giá). Bên cạnh đó nó cũng mô tả cách framework được triển khai cho việc đánh giá những thử nghiệm với tập dữ liệu N-BaIoT dataset

* 1. **Client**

Nhìn chung các thiết bị IoT nói chung có tài nguyên hạn chề và độ tin cậy khiêm tốn khách hàng phụ trách việc training models không phải là các thiết bị được bảo vệ nhưng các thực thể khác có khả năng thu thập lưu lượng truy cập của các thiết bị IoT trong cùng một mạng. Lưu ý rằng mỗi khách hàng có thể sở hữu một số thiết bị IoT nhưng vì đơn giản các thử nghiệm chỉ thược hiện trên một thiết bị duy nhất cho mỗi khách hàng.

Diagram

Description automatically generated

Mô tả chi tiết kiến trúc của một máy khách khi thu thập dữ liệu và cách nó tương tác với máy chủ

* + 1. **Data Acquisition (thu thập dữ liệu)**

Khách hàng chịu trách nhiệm thu thập dữ liệu lưu lượng truy cập từ các thiết bị theo quan sát.

* + 1. **Dataset (tập dữ liệu)**

Có 2 tình huống. Ở tình huống thứ nhất đưa ra một thiết lập trong đó mỗi khách hàng có quyền truy cập vào dữ liệu được gắn nhãn từ thiết bị của chính mình, tình huống thứ hai, chúng ta giả định rằng mỗi máy khách chỉ có quyền truy cập vào lưu lượng lành tính của thiết bị. Vì nhận được lượng lớn dữ liệu lưu lượng truy cập lành tính dễ dàng và không cần gắn nhãn tình huống này được gọi là học không giám sát tuy nhiên nó được định nghĩa là học có giám sát trong single-class.

Trên thực tế, một tình huống được giám sát với một số máy khách có thể tạo ra một lượng lớn dữ liệu được gắn nhãn là hợp lý nhưng không phổ biến. Do đó chúng ta có hai giải pháp như sau

Đầu tiên là một điểm so sánh cho giải pháp không được giảm sát, vè tình huống được giám sát nó dễ giải quyết và dễ kiểm soát hơn tình huống không được giám sát, thứ hai là cố gắng đi sâu nhât có thể trong các cuộc thử nghiệm để có thể tiết lộ các lỗ hổng hoặt cái mối lo ngại khác về việc sử dụng FL để phát hiện phần mềm độc hại

Diagram

Description automatically generated

Trong hình trên xét cã hai tình huống chúng ta tách dữ liệu của một thiết bị duy nhất cho mục đích huấn luyện và thử nghiệm

Trong tình huống được giám sát mỗi tập dữ liệu được chia theo thứ tự thời gian giữa 3 phần:

The train set (79%), the aforementioned unused set (1%), the test set (20%)

Trong tình huống không được giám sát chỉ có dữ liệu lành tính để huấn luyện nên phần này được chia thành 4 nhóm khác nhau:

The train set (39.5%),a so-called threshold-selection set (39.5%), the unused set (1%) và the benign part of the test set (20%) tất cã dữ liệu tấn công là có sẵn cho các thử nghiệm

Để phòng tránh một số trường hợp có thể xảy ra chúng tôi cân bằng lại tập dữ liệu theo 3 cách khác nhau để giải quyết một số trường hợp. Chúng tôi đã chọn những tỉ lệ lớp sau cho mọi thiết bị:

-7.87% benign traffic và 92.12% attack traffic đây là những dữ liệu gốc ban đầu, với sự khác biệt là tỷ lệ mỗi lớp hiện không thay đổi trên các thiết bị

-50% benign traffic và 50% attack traffic (lưu lượng tấn công) giúp tập dữ liệu cân bằng để phân loại nhị phân.

-95% benign traffic và 5% attack traffic điều này phổ biến hơn nhiều mà phù hợp với thực tế bởi thường có nhiều dữ liệu lành tính hơn.

Điều cần chú ý là ba hay đổi này dẫn đến 3 vấn đề khác nhau, mục đích của việc thay đổi này không phải là so sánh tác động đến hiệu suất của mô hình khi cân bằng các lớp mà là cách để làm cho kết quả rộng hơn.

Số lượng mẫu trên mỗi thiết bị cũng được cố định không đổi theo thứ tự để làm cho kết quả ít phụ thuộc vào các trường hợp huấn luyện và giữ cho kích thước tập dữ liệu cố định theo bất kể tỉ lê lớp nào. 100 000 mẫu cho mỗi thiết bị được sử dụng cho giải pháp có giám sát và chỉ 10 000 mẫu cho giải pháp không được giám sát. Điều này là bởi vì quá trình đào tạo không có giám sát cần nhiều thời gian hơn để hội tụ và nó chỉ cần dữ liệu lành tính. Sau khi thiết lập tỷ lệ của mỗi lớp và số lượng mẫu mong muốn trên mỗi thiết bị, chúng tôi sẽ có được một tập dữ liệu trong đó số lượng mẫu và tỷ lệ của các lớp là như nhau cho tất cả các thiết bị.

Sau quá trình cân bằng này chúng ta có tập hợp khách hàng k được định nghĩa là và bộ lựa chọn ngưỡng của nó (giải pháp không được giám sát) là số lượng mẫu của khách hàng k sẽ là nk := | |

* + 1. **Data preprocessing**

Thành phần này trách nhiệm chuẩn hóa các mẫu, tính toán các tỷ lệ tối thiếu, tối đa được sữ dụng là nơi để các hoạt động trong việc element-wise(Hiểu đơn giản thì Element-wise là phép toán trên các phần tử tương ứng giữa các tensor.)các giá trị chuẩn hóa chỉ đươc tính với train set mà khách hàng sỡ hữu. lưu ý với mỗi khác hàng k thì có những giá trị chuẩn hóa và riêng

* + 1. **Model training**

Mục đích của thành phần này là huấn luyện mô hình federated ML sử dụng để phát hiện phần mềm độc hại. để làm được điều đó đầu tiên chúng ta sẽ trình bày các kiểu thiết lập được sử dụng để phân loại và phát hiện các dị thường sau đó sử dụng 2 thuật toán FL khác nhau.

Cách các thành phần này tương tác với máy chủ Trong suốt phần còn lại của công việc này, các tham số mô hình của máy khách 𝑘 được gọi là 𝑤𝑘 và các tham số của mô hình toàn cục là 𝑤. Hơn nữa, với 𝑑 số kích thước của 𝑤 và với, để xác định kích thước thứ i của w.

Supervised situation, Trong thiết lập này, một nhiệm vụ phân loại nhị phân được xem xét với bốn kiến trúc khác nhau của các perceptron nhiều lớp (MLP) với 1 neuron đầu ra:

• Classifier A: Không có lớp ẩn (mô hình tuyến tính).

• Classifier B: 1 lớp ẩn với 115 neurons ẩn.

• Classifier C: 2 lớp ẩn với 115 và 58 neurons ẩn, tương ứng.

• Classifier D: 3 lớp ẩn với 115, 58 và 29 neurons ẩn, tương ứng.

Sau mỗi lớp ẩn, chức năng kích hoạt đơn vị tuyến tính hàm mũ (ELU) được sử dụng. Lưu ý rằng số lượng tế bào thần kinh ẩn được thử, 115, 58 và 29, tương ứng với 100%, 50% và 25% của tỉ lệ đầu vào

Unsupervised situation Trong thiết lập này, mã tự động được sử dụng để phát hiện bất thường, theo một phương pháp tương tự như các tác giả của N-BaIoT..

Autoencoder là một dạng mạng neurons truyền tới, đặc biệt được có hai phần, bộ mã hóa và bộ giải mã Bộ mã hóa chuyển đổi đầu vào bằng cách giảm số thứ nguyên của nó thành một giá trị được xác định là thứ nguyên mã hóa và bộ giải mã cố gắng ánh xạ đầu vào được mã hóa trở lại đầu vào ban đầu. Nó được đào tạo bằng cách giảm thiểu Mean Squared Error (MSE) giữa các tính năng được tái tạo và đầu vào. Để sử dụng nguyên tắc này để phát hiện bất thường, một bộ mã tự động được đào tạo với dữ liệu lành tính, học cách tái tạo lại dữ liệu đó, sao cho nó có ít lỗi xây dựng lại đối với dữ liệu lành tính trong tương lai và lỗi xây dựng lại cao đối với bất kỳ thứ gì khác với dữ liệu lành tính. Khi quá trình đào tạo hoàn tất, một ngưỡng được đặt dựa trên thống kê về lỗi tái tạo của dữ liệu lành tính. Trong quá trình thử nghiệm, nếu một mẫu có sai số xây dựng lại bình phương trung bình cao hơn ngưỡng quy định thì nó được coi là bất thường (dương tính), ngược lại thì được coi là lành tính (âm tính). Công thức ngưỡng được sử dụng trong công việc này xuất phát từ [[8]](#tam) và chọn cho khách hàng 𝑘 ∈ [𝐾] ngưỡng.

thrk=mean(MSE(;wk))+std(MSE(;wk))

Trong đó 𝑀𝑆𝐸 (⋅; 𝑤𝑘) là lỗi xây dựng lại bình phương trung bình được tính với các tham số mô hình 𝑤𝑘 Nhiều kiến trúc mã tự động được điều tra trong quá trình tìm kiếm lưới.

• Autoencoder A: 1 lớp ẩn gồm 29 neurons

• Autoencoder B: 3 lớp ẩn 58, 29 và 58 neurons

• Autoencoder C: 7 lớp ẩn của tế bào thần kinh 86, 58, 38, 29, 38, 58 và 86. Số lượng tế bào thần kinh và lớp này gần tương ứng với số lượng neurons được sử dụng trong giải pháp của những người tạo ra N BaIoT.

Một lần nữa, ELU được sử dụng sau mỗi lớp ẩn. Số lượng thứ nguyên thấp là một cách để hạn chế trình mã tự động học cách biểu diễn cụ thể hơn cho dữ liệu lành tính. lựa chọn 29 kích thước mã hóa tương ứng với những gì được sử dụng Nếu không nhìn vào dữ liệu được gắn nhãn, thật khó để lựa chọn siêu tham số này tốt hơn. Các số 86, 58, 38 và 29, tương ứng tương ứng với 75%, 50%, 33% và 25% của kích thước đầu vào

Interactions with the server. Hai thuật toán FL Mini-batch aggregation và Multi-epoch aggregation, xuất phát từ Fed AVG phổ biến được xem sét. Đối với cã 2 thuật toán điểm khác biệt với Fed AVG là chúng ta coi hàm tổng hợp như một tham số của thuật toán. Do đó máy chủ có thể thử các chức năng tổng hợp khác ngoài tính năng trung bình. Đặc biệt nó còn cung cấp một triển khai thực tế hơn, nơi tỉ lệ học tập thay đổi trong quá trình đào tạo.

TRONG Mini-batch aggregation các thành phần huấn luyện sẽ huấn luyện mô hình với lượng nhỏ mini-batch dữ liệu trước khi gửi đến máy chủ để tổng hợp. Sau đó, thành phần Huấn luyện mô hình sẽ nhận được mô hình toàn cầu tổng hợp mới,qua đó quá trình huấn luyện có thể tiếp tục. Quá trình này được lặp lại cho đến khi hoàn thành một số 𝐸 epochs trên toàn bộ train set Trong tổng hợp nhiều kỷ nguyên, mô hình được huấn luyện cho tất cả các kỷ nguyên cùng một lúc trước khi được gửi đến máy chủ để tổng hợp. các mô hình tính trung bình có thể có kết quả xấu tùy ý vì tính không lồi của mục tiêu. Để cố gắng giảm thiểu điều đó, việc huấn luyện tập hợp đa kỷ nguyên được lặp lại trong 𝑇 = 30 vòng, với tốc độ học giảm dần qua các vòng.

Trong Multi-epoch aggregation những mẫu sẽ được tạo tạo cho tất cã E epochs cùng 1 lúc trước khi gửi đến máy chủ để tổng hợp. Các mô hình sử dụng tính trung bình này thì có thể có kết quả xấu vì tính không lồi của mục tiêu nó có nhiều khả năng xảy ra với tập hợp nhiều Multi-epoch. Để giảm thiếu chúng ta sẽ huấn luyện các Multi-epoch lặp lại sau mỗi T=30 vòng, với tốc độ giảm dần qua các vòng.

* + 1. **Model evaluation**

Sau khi các mô hình đã được đào tạo để đáp ứng một số điểm thông qua quy trình FL thì nó sẽ sẵn sàng để được đánh giá. Để đánh giá độ tin cậy của các mô hình được đào tạo, chúng ta sẽ đánh giá chúng trên các bộ thử nghiệm khác nhau bằng cách đánh giá các mô hình trong phần thử nghiệm dữ liệu từ các thiết bị do khách hàng sở hữu.Lưu ý hiệu suất thiết bị mới được tính toán dựa trên dữ liệu từ một thiết bị hoàn toàn mới đối với tất cả các ứng dụng khách trong liên kết

* 1. **Server**

Trong các framework được đề xuất máy chủ sẽ chịu trách nhiệm điều phối, huấn luyện các máy khách được liên kết, cụ thể nó khỏi tạo mô hình ngay từ đầu sau đó nó tổng hợp các mô hình do khách hàng gửi vào một mô hình toàn cầu. Thành phần máy chủ sẽ được đặt trong lớp CLOUD SLICING, trên Fog Nodes hoặt Cloud Data Centers, tùy thuộc vào phạm vi của khách hàng.

* + 1. **Model initialization**

Máy chủ sẽ chịu trách nhiệm khởi tạo các trọng số của mô hình ban đầu. sau khi hoàn thành mô hình sẽ được chia sẽ với tất cã các máy khách và quá trình đào tạo có thể bắt đầu, mỗi khách hàng sẽ bắt đầu với cùng một mô hình.

* + 1. **Model aggregation**

Sau khi nhận được các thông số của mỗi khách hàng {wk : }, máy chủ sẽ tổng hợp và tạo các tham số mô hình toàn cầu mới w. Với phương pháp lấy trung bình, công thức được đưa ra sẽ là w := wk khi hàm tổng này này được sử dụng chúng ta sẽ đề cập đến các thuật toán Mini-batch aggregation và Multi-epoch aggregation dưới dạng Mini-batch avg và Multi-epoch avg,respect-tively. Chúng ta có thể sử dụng giá trị trung bình có trọng số nếu số lượng mẫu khác nhau giữa các khách hàng.

Mặc dù máy chủ phải chịu trách nhiệm cho việc tổng hợp và thiết kế các framework do những lợi thế của việc điều phối quá trình chúng ta có thể di chuyển các bước Model Initialization, Model Aggregation vào thẳng trong các máy khách. Do đó các framework sẽ được phân cấp hoàn toàn nếu không có máy chủ điều phối các mô hình đã tạo.

* 1. **Additional concerns of the proposed framework**

Phần này bổ sung một số điều phát sinh khi thực hiện ML trong federated.Cụ thể các bước chuẩn hóa và lựa chọn các siêu tham số phải được chú ý. Với giải pháp không giám sát bước lựa chọn ngưỡng cũng cần được quan tâm.

* + 1. **Collaborative normalization**

Mỗi khách hàng k có thể tính toán và và máy chủ có thể tính toán những giá trị sử dụng toàn cầu nhỏ nhất lớn nhất và là những giá trị tối thiểu và tối đa cho các giá trị đó. Quá trình này đưa ra mức tối thiểu và tối đa toàn cục như thể chúng được tính toán trực tiếp trên sự kết hợp của các train sets của tất cả các máy khách. Điều này có nhược điểm là yêu cầu mỗi khách hàng phải tiết lộ các giá trị chính xác của nó là tối thiểu và tối đa cho từng tính năng trong số 115 features.

* + 1. **Collaborative grid search**

Chúng ta cần phân loại 2 loại siêu tham số: những cái cần phổ biến cho mọi khách hàng, chủ yếu là kiến trúc của mô hình và những cái có thể khác nhau đối với từng khách hàng, chủ yếu là về tối ưu hóa.

Ở loại đầu tiêu nó phải được phổ biến cho tất cã các máy khách để truyền 1 số kết quả xác nhận để đồng ý về lựa chọn.

Lưu ý rằng đối với giải pháp không được giám sát, mô hình chỉ được xác thực với dữ liệu lành tính, vì vậy các siêu tham số được chọn là những siêu tham số giảm thiểu tổn thất. Tuy nhiên, trong giải pháp được giám sát, việc lựa chọn dựa trên độ chính xác của việc xác nhận.

* + 1. **Collaborative threshold selection**

Đối với phương pháp phát hiện bất thường không có giám sát, ngoài việc đào tạo mô hình, khách hàng phải chọn ngưỡng. Vì vậy, trong khuôn khổ liên kết được đề xuất của chúng tôi, mỗi khách hàng 𝑘 tính một ngưỡng cục bộ với và truyền nó đến máy chủ, sau đó máy chủ sẽ tính ngưỡng toàn cục làm giá trị trung bình của các ngưỡng cục bộ. Ngưỡng chỉ cần được tính toán cho thử nghiệm cuối cùng sau khi mô hình đã được huấn luyện cho số lần lặp được chỉ định. Chúng tôi vẫn quyết định tính toán nó ở một số bước trong quá trình đào tạo để cho thấy sự tiến hóa của nó.

# Adversarial attacks and countermeasures

Phần này cung cấp nền tảng lý thuyết liên quan đến một số cuộc tấn công đầu độc nổi tiếng nhất, nhằm mục đích làm giảm hiệu suất của mô hình. Bên cạnh đó, nó cũng mô tả các chức năng tổng hợp mô hình khác nhau có thể cải thiện khả năng phục hồi của đào tạo mô hình liên kết chống lại các cuộc tấn công.

* 1. **Adversarial attacks**

Với một máy chủ tin cậy thì phần lớn các khách hàng tin cậy và 1 số ít các khách hàng độc hại thông đồng nhau sẽ được giả định trong phần này. Các cuộc tấn công nhiễm độc dữ liệu hoạt động thông qua phương tiện của máy khách trong tập dữ liệu. Ứng dụng khách có thể độc hại và có thể cố tình sửa đổi dữ liệu làm cho nó sai lệch. Các cuộc tấn công có thể đến từ bất kì phần nào trong đường dẫn dữ liệu bởi một thực thể đọc hại bên ngoài có quyền kiểm soát.

-• Gradient factor attack. Các ứng dụng khách độc hại với một hệ số tiêu cực abrad trước khi cập nhập mô hình cục bộ và chia sẽ nó với máy chủ các máy khách độc hại này chỉ cần có quyền truy cập dữ liệu từ một thiết bị Do đó, họ chỉ có thể tính toán ước tính của gradient trên phân phối dữ liệu của riêng họ với K tổng số khách hàng trong đó f là độc hại thì ta có yếu tố agrand được tính bằng (K – f + agrad. f) = -1.

• Model cancelling attackTrong cuộc tấn công này, các máy khách độc hại cố gắng đưa tất cả các tham số của mô hình toàn cục về giá trị 0. Họ chọn mô hình của mình theo cách mà khi được tính trung bình với các mô hình khách hàng trung thực, mô hình toàn cầu ban đầu sẽ biến mất, Để đạt được điều đó, họ chỉ cần xuất ra các tham số mô hình toàn cục ban đầu, nhân với một hệ số aparam K – f + agrad.f = 0.

* 1. **Robust model aggregation functions**

Có nhiều cách khác nhau để đảm bảo an toàn cho hệ thống trước các cuộc tấn công. Một trong những ý tưởng mở rộng nhất là sử dụng tập hợp mô hình và cập nhật các giải pháp xử lý có tính đến khả năng của các khách hàng độc hại đang cố gắng chiếm đoạt mô hình.

Coordinate-wise median chức năng này được đề xuất áp dụng giá trị trung bình cho từng thông số riêng lẻ để loại trừ hoàn toàn bất kỳ ngoại lệ tiềm năng nào.

Coordinate-wise trimmed mean một phần của các giá trị lớn nhất và nhỏ nhất bị loại bỏ trước khi tính giá trị trung bình. Do số lượng khách hàng thấp trong kịch bản mà chúng tôi xem xét, thuật toán trung bình được cắt bớt được xác định lại bằng cách sử dụng số nguyên 𝑐 của các giá trị lớn nhất và thấp nhất bị loại trừ thay vì tỷ lệ, nhưng cả hai đều tương đương. Do đó, theo định nghĩa của chúng tôi, tọa độ thứ của 𝑤 được cho bởi w(i) = u, thu được bằng cách loại bỏ 𝑐 lớn nhất và 𝑐 nhỏ nhất trong số các phần tử của nó. Số phần tử bị loại là 2𝑐. Hàm tổng hợp này được gọi là TM (𝑐).

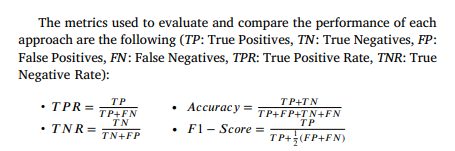
S-resampling một bước bổ sung có thể được thực hiện trước khi tổng hợp. Trong một tình huống mà tập dữ liệu của mỗi khách hàng có bản phân phối riêng, nó nhằm mục đích giảm sự không đồng nhất của các mô hình được gửi bởi mỗi khách hàng là được kết hợp với một tập hợp mạnh mẽ chức năng để giảm tác dụng phụ của việc sử dụng một chức năng như vậy trên không phải IID các mô hình. Nó hoạt động bằng cách thay thế từng mô hình bằng giá trị trung bình giữa các mô hình một cách ngẫu nhiên được lấy mẫu từ các mô hình 𝐾 khách hàng Mỗi mô hình có thể được lấy mẫu một tổng cộng tối đa là 𝑠 lần sResampling cũng có thể khiến các mô hình độc hại bị pha loãng thành một số mô hình mà nó xuất ra, tăng phạm vi tiếp cận của các máy khách độc hại. Vì điều đó lý do, nó chỉ được kỳ vọng sẽ hoạt động hài lòng với một số lượng nhỏ máy khách độc hại, giá trị 𝑠 nhỏ và có chức năng tổng hợp loại bỏ một số giá trị cực đoan cao.

# Kết quả thực nghiệm

Phần này trình bày chi tiết các kết quả thu được trong các thí nghiệm được thực hiện để xác nhận framework được đề xuất. Đầu tiên, so sánh hiệu suất khi phát hiện phần mềm độc hại giữa các phương pháp tiếp cận học liên kết và phương pháp truyền thống khi thực hiện theo cả phương pháp được giám sát và không được giám sát. Sau đó, cho thấy tác động của các cuộc tấn công đối kháng và cách các hàm tổng hợp chống lại các cuộc tấn công đó.

Ảnh có chứa văn bản

Mô tả được tạo tự động



Học liên kết bao gồm 𝐾 = 8 clients, mỗi client sở hữu dữ liệu của một trong 9 thiết bị có sẵn trong tập dữ liệu N-BaIoT. Dữ liệu của một thiết bị sẽ không được sử dụng trong quá trình đào tạo, lưu giữ nó như một thiết bị vô hình cho mục đích thử nghiệm. Trong bối cảnh này, sự kết hợp của 9 thiết bị khác nhau (với một thiết bị không nhìn thấy) đã được sử dụng trong tất cả các thí nghiệm. Hơn nữa, các thí nghiệm được lặp lại 5 lần để cải thiện tính nhất quán của kết quả. Cuối cùng, kết quả của mỗi thử nghiệm sẽ có hơn 45 lần chạy (9 thiết bị không nhìn thấy có thể có và 5 lần thực hiện).

* 1. **Hiệu suất của học tập liên kết và truyền thống để phát hiện phần mềm độc hại trong các thiết bị IoT**

Thử nghiệm này nhằm đo lường hiệu suất của giải pháp khi phát hiện phần mềm độc hại IoT bằng tập dữ liệu N-BaIoT. Để xác minh rằng phương pháp tiếp cận học tập liên kết phù hợp với kịch bản phần mềm độc hại IoT, điều đó là cần thiết để so sánh nó với các giải pháp truyền thống. Cụ thể hơn, các lựa chọn giải pháp được so sánh là:

• Naive decentralized approach. Mỗi máy khách sử dụng tập dữ liệu cục bộ của mình cho đào tạo và kiểm tra. Vì mỗi máy khách tạo ra mô hình riêng, kết quả được so sánh bằng cách lấy trung bình hiệu suất của từng máy khách.

• Centralized approach. Tất cả dữ liệu đào tạo được chia sẻ với một máy chủ phụ trách đào tạo và thử nghiệm mô hình. Ở đây không có sự riêng tư.

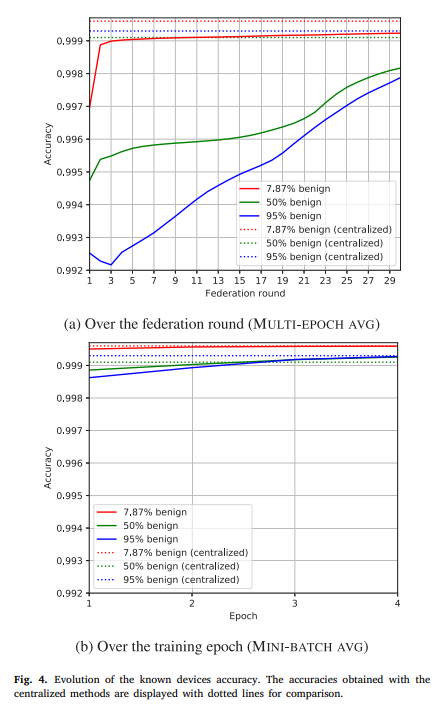
• Federated with Mini-batch avg. Các máy khách khác nhau cộng tác để tạo mô hình toàn cục bằng cách sử dụng thuật toán Mini-batch aggregation với chức năng tổng hợp AVG.

• Federated with Multi-epoch avg. Tương tự như cách tiếp cận trước nhưng đào tạo với thuật toán Multi-epoch aggregation theo thứ tự để giảm đáng kể chi phí thông tin liên lạc.

Các bước sau được thực hiện cho cả giải pháp được giám sát và không được giám sát. Đầu tiên, hai siêu tham số quan trọng (kiến trúc của mô hình và giá trị L2-regularization 𝜆) được chọn cho mỗi thiết lập bằng cách sử dụng tìm kiếm lưới. Các kiến trúc MLP và autoencoder được xét là những kiến trúc được mô tả trong Phần [2.1.4](#modeltraining). Giá trị được xét cho 𝜆 là 0, 10−5 và 10−4. Với naive method, các siêu tham số được chọn cho mỗi máy khách bởi vì các máy khách không cộng tác trong việc lựa chọn siêu tham số. Đối với các phương pháp học liên kết, mỗi federation sử dụng các tìm kiếm lưới cộng tác để chọn các siêu tham số, như được định nghĩa trong Phần [2.3.2](#collabrativegridsearch). Cuối cùng, trong centralized method, tìm kiếm lưới được thực hiện trực tiếp bởi máy chủ nhận toàn bộ tập dữ liệu. Đối với tất cả các thử nghiệm, batch size là 𝐵 = 64 được sử dụng khi đào tạo, ngoại trừ với Mini-batch avg thì kích thước lô được chia cho số lượng máy khách (𝐵 = 8), để mỗi bản cập nhật mô hình là cũng được thực hiện với tổng số 64 mẫu. Trong tất cả các thử nghiệm, cập nhật mô hình được tính bằng Stochastic Gradient Descent (SGD). Đối với giải pháp có giám sát, quá trình huấn luyện được thực hiện trong 𝐸 = 4 epochs; đối với giải pháp không giám sát 𝐸 = 120 epochs.Ảnh có chứa bàn

Mô tả được tạo tự động

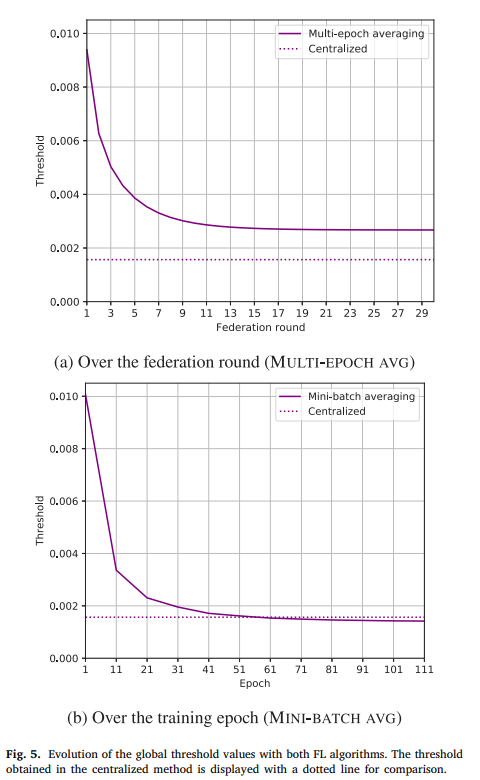
*Tình hình được giám sát.* 3 tùy chọn tách tập dữ liệu khác nhau được giải thích trong Phần [2.1.2](#dataset) (7,87%, 50% và 95% dữ liệu lành tính) được sử dụng trong các thử nghiệm, cũng kiểm tra xem class balances khác nhau ảnh hưởng như thế nào đến kết quả. Table 4 cho thấy kết quả đạt được trong các thí nghiệm này.



Ảnh có chứa bàn

Mô tả được tạo tự động

Kết quả đáng chú ý đầu tiên là hiệu suất của centralized method cao hơn naive method, đặc biệt là khi được đánh giá trên một thiết bị không nhìn thấy. Hơn nữa, trên cả ba cài đặt tập dữ liệu, kết quả Mini-batch avg rất gần với centralized method, thậm chí đôi khi vượt quá. Mặc dù Mini-batch avg thu được kết quả tốt hơn centralized method có thể gây ngạc nhiên, điều này có thể được giải thích bởi một số yếu tố, chẳng hạn như tính ngẫu nhiên của các thử nghiệm hoặc thực tế rằng các siêu tham số được tính khác nhau. Multi-epoch avg cũng tạo ra kết quả khá hài lòng, với mức giảm không đáng kể về độ chính xác trên các thiết bị đã biết, luôn được bù đắp bởi độ chính xác vượt quá centralized method trên thiết bị mới. Điều này có thể được giải thích bởi thực tế là tính trung bình của các tham số mô hình với một non-convex loss function có thể có các tác động làm hỏng mô hình một cách tùy ý, như được giải thích trong [[9]](#chin). Tuy nhiên, nó cũng có thể được xem như một dạng cơ chế hành động chống lại việc trang bị quá mức, do đó cải thiện khả năng tổng quát hóa trên một thiết bị. Hơn nữa, những kết quả này có thể được cải thiện một chút bằng cách sử dụng số lượng 𝑇 vòng liên kết cao hơn. Fig 4 cho thấy rằng bộ dữ liệu với 50% và 95% dữ liệu lành tính, độ chính xác dường như hội tụ chính xác sau 30 vòng.



Tình huống không được giám sát. Sau khi hiệu suất được giám sát được xác minh, bước tiếp theo là đánh giá cái không được giám sát. Ở đây, chỉ những dữ liệu lành tính được sử dụng để đào tạo, vì vậy mô hình cuối cùng không phụ thuộc vào class balance trong tập dữ liệu. Để làm cho kết quả của chúng tôi không phụ thuộc vào class balance, chúng tôi chỉ hiển thị TPR và TNR cho giải pháp này. Phương trình được sử dụng để xác định ngưỡng được mô tả trong Phần [2.1.4](#modeltraining). Trong số các kiến ​​trúc có thể có, kiến ​​trúc đầu tiên (Autoencoder A) luôn là kiến ​​trúc tốt nhất mất xác thực trong tất cả các lựa chọn siêu tham số. Tất cả các kết quả từ tình huống không được giám sát sẽ được tạo thêm với Autoencoder A. Table 5 cho thấy các kết quả không được giám sát của hệ thống.

Ở đây, việc tập trung dữ liệu mang lại hiệu suất tổng thể cao cải tiến so với naive method. Hơn nữa, các thuật toán FL cũng giải quyết rất thành công nhiệm vụ lấy dấu vân tay không được giám sát. Cụ thể, centralized performance gần như đạt được bởi cả Multi-epoch avg và Mini-batch avg. Một lần nữa, Multi-epoch avg dường như giúp mô hình tổng quát hóa tốt hơn, vì nó thể hiện trên thiết bị mới tốt hơn một chút so với Mini-batch avg. Điều thú vị là ngưỡng, như được hiển thị trong Fig 5, hội tụ đến một giá trị lớn hơn, trong trường hợp Multi-epoch avg hơn so với centralized method đạt được. Như đã giải thích trước đó, lựa chọn ngưỡng cộng tác không tương đương với việc chọn ngưỡng trực tiếp trên toàn bộ bộ lựa chọn ngưỡng (như trong centralized method), vì vậy kết quả này là không đáng ngạc nhiên.

Với các thử nghiệm trước, người ta đã xác minh rằng trong kịch bản cụ thể về phát hiện phần mềm độc hại trong các thiết bị IoT, sử dụng thêm dữ liệu để đào tạo mô hình cho thấy một sự cải thiện đáng kể, đặc biệt là trên các thiết bị chưa từng thấy trước đây. Bên cạnh đó, đào tạo dựa trên FL thành công đạt được centralized performance theo cách bảo vệ quyền riêng tư.

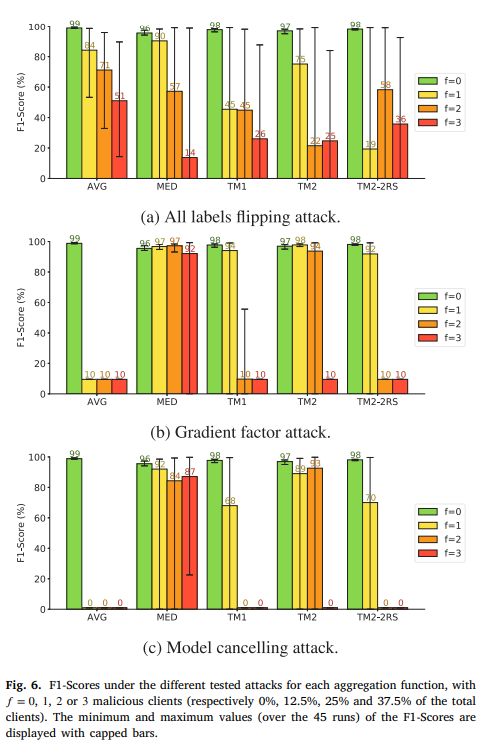
* 1. **Tác động của các cuộc tấn công đối thủ và các biện pháp đối phó khi phát hiện phần mềm độc hại**

Khi hiệu suất của phương pháp học liên kết đã được xác minh, bước tiếp theo là đánh giá cách thức các cuộc tấn công đối thủ khác nhau đề xuất trong Phần 5 ảnh hưởng đến cách tiếp cận học liên kết. Bên cạnh đó, các hàm tổng hợp khác nhau được áp dụng để kiểm tra cách chúng cải thiện mô hình khả năng phục hồi chống lại các cuộc tấn công khác nhau. Để ngắn gọn, những thử nghiệm tập trung vào tình huống được giám sát và chỉ sử dụng tập dữ liệu cân bằng với 95% dữ liệu lành tính. Hơn nữa, chúng được tiến hành với thuật toán liên kết Mini-batch aggregation. Batch size 𝐵 = 64 (thay vì 𝐵 = 8) được sử dụng cho tất cả các thử nghiệm đối kháng, vì nó cho phép cập nhật mượt mà hơn cho các hàm tổng hợp mạnh mẽ.

Như đã giải thích trước đó, s-Resampling chỉ được mong đợi hoạt động với giá trị s nhỏ và kết hợp với MED hoặc TM(2) (hoặc các hàm tổng hợp mạnh mẽ khác mà chúng tôi đã không thực hiện). Vì TM(2) tính toán đầu ra của nó bằng cách tính đến nhiều giá trị hơn MED, s-Resampling lại chỉ được thử nghiệm cho TM(2) và với 𝑠 = 2. Điều này kết hợp được gọi là TM(2) ◦ 2-Resampling.

Trong các thử nghiệm thực hiện các cuộc tấn công độc hại dữ liệu, All labels flipping attack được chọn để thử nghiệm, vì nó kết hợp cả hai yếu tố lành tính và attack label flipping. 𝑝𝑝𝑜𝑖𝑠𝑜𝑛 = 1 luôn được sử dụng. Cách tiếp cận này cho phép xác minh tác động tối đa của cuộc tấn công trong mô hình mẫu.

Về mô hình các cuộc tấn công đầu độc, trong trường hợp yếu tố gradient tấn công, giải quyết Eq. (2) cho . Với tổng số 8 máy khách bao gồm 1, 2 và 3 ứng dụng khách độc hại, các giá trị được chọn cho 𝛼𝑔𝑟𝑎𝑑 là lần lượt là −15, −7 và . Trong trường hợp mô hình cancelling attack, giải quyết Eq. (3) cho . Với tổng số 8 máy khách bao gồm 1, 2 và 3 ứng dụng khách độc hại, các giá trị được chọn cho 𝛼𝑝𝑎𝑟𝑎𝑚 tương ứng là −7, −3 và .



[Fig.6](#fig6) cho thấy F1-Score của mô hình được thử nghiệm trên các thiết bị do máy khách sở hữu (các thiết bị đã biết) khác nhau như thế nào trong các cuộc tấn công được triển khai. Nó cũng cho thấy sự phát triển của số liệu khi số lượng máy khách độc hại tăng lên từ 0 đến 3 (hoặc 37,5% tổng số máy khách trong thiết lập). Cần lưu ý rằng những kết quả này có một phương sai lớn do tính ngẫu nhiên của việc lựa chọn ứng dụng khách nào là độc hại. Thậm chí mặc dù mỗi thử nghiệm đã được chạy tổng cộng 45 lần, điều này có thể dẫn đến kết quả không mong đợi, chẳng hạn như đôi khi F1- Score trung bình tốt hơn khi nhiều máy khách độc hại hơn. Tuy nhiên, những thử nghiệm này là đủ để có được một ý tưởng có cơ sở tốt về mức độ nghiêm trọng của vấn đề đối kháng.

Như chúng ta có thể quan sát, trung bình (AVG) là hàm tổng hợp tốt nhất khi tất cả các máy khách đều trung thực. Tuy nhiên, khi các máy khách nhiễm độc liên quan, hiệu suất của nó bị ảnh hưởng nặng nề tùy thuộc vào cuộc tấn công. Cụ thể, dưới gradient factor và mô hình cancelling attacks, ngay cả một ứng dụng khách nhiễm độc cũng đủ để biến mô hình nhất quán thành một constant predictor (lưu ý rằng một constant predictor có F1-Score ∼10% và một constant negative predictor có F1-Score 0%). Điều này chứng tỏ sự cần thiết của việc sử dụng các phương pháp mạnh mẽ hơn khi giả định một mô hình óc mối đe dọa, thậm chí mô hình chỉ có một máy khách có thể nhiễm độc.

Coordinate-wise median aggregation (MED) thể hiện khả năng phục hồi cao hơn chống lại hầu hết các kịch bản tấn công được xem xét. Nhìn chung, nó có kết quả tốt nhất trong số các hàm tổng hợp được thử nghiệm trong thiết lập đối kháng. Tuy nhiên, điều này vẫn còn lâu mới đủ mạnh, đặc biệt là khi thử với 3 ứng dụng khách độc hại, vì all labels flipping attack gây ra F1-Score của nó đạt giá trị trung bình khoảng 14%. Ngay cả với một ứng dụng khách nhiễm độc, khi thực hiện all labels flipping và model cancelling attacks, mặc dù F1-Scores trung bình lần lượt là 90% và 92%, giá trị tối thiểu của chúng trong 45 lần chạy là 0% trong cả hai trường hợp, làm cho nó vẫn không đáng tin cậy.

Không có gì đáng ngạc nhiên, Coordinate-wise trimmed mean (TM (𝑐)) không thành công khi được sử dụng để chống lại nhiều hơn 𝑐 máy khách độc hại, như đã được chứng minh rõ ràng trong mô hình hủy kết quả tấn công (Hình 6c). Tuy nhiên, nó không nghĩa là hàm tổng hợp này hoạt động tốt khi 𝑐 ≥ 𝑓, như F1-Score thấp nhất là 0% ngay cả đối với một ứng dụng khách nhiễm độc dưới all labels flipping attack (Hình 6a). Lợi ích duy nhất của TM(1) và TM(2) trên MED nằm ở hiệu suất khi không có ứng dụng khách nhiễm độc nào có liên quan, điều này tốt hơn một chút vì có nhiều tham số hơn được xem xét trong quá trình tính toán mô hình toàn cục. Lợi thế này có thể là cao hơn trong trường hợp sử dụng với nhiều máy khách hơn, nhưng trong trường hợp này, nó quá thấp để biện minh cho việc sử dụng TM.

Cuối cùng, 2-Resampling cho thấy sự cải thiện về độ chính xác trên các thiết bị đã biết khi không có ứng dụng nhiễm khách độc nào tham gia. Tuy nhiên, điều này đi kèm với cái giá phải trả là giảm độ mạnh của hệ thống, làm cho TM(2) ◦ 2-Resampling có kết quả tương tự như TM(1) hầu hết thời gian. Tuy nhiên là có một cải tiến nhỏ so với TM(2), được thể hiện trong Hình 6a, lưu ý là với 2 và 3 máy khách nhiễm độc. Tương tự như TM, s-Resampling không cung cấp đủ lợi thế để được sử dụng trong một mô hình học liên kết nhỏ như vậy, nhưng nó có thể trở nên hữu ích hơn ở quy mô lớn hơn.

Nhận xét chung, mặc dù khả năng phục hồi của các mô hình đã được cải thiện đáng kể bằng cách sử dụng MED trong các mô hình tấn công đầu độc (gradient factor and model cancelling attacks), hiệu suất của mô hình là vẫn giảm về cơ bản. Ngoài ra, trong trường hợp all labels flipping attack, AVG vẫn hoạt động tốt hơn các hàm khác là tìm cách cải thiện độ chắc chắn của mô hình. Những kết quả này cho thấy rằng, mặc dù hiệu suất của mô hình đã được cải thiện, nghiên cứu thêm về các hàm tổng hợp có khả năng phục hồi trước các cuộc tấn công đối kháng là vẫn được yêu cầu. Chúng tôi tin rằng trong trường hợp các cuộc tấn công khác ảnh hưởng đến trọng số, kết quả sẽ tương tự như kết quả của gradient factor and model cancelling attacks. Tuy nhiên vẫn chưa biết hiệu suất sẽ bị ảnh hưởng như thế nào trong trường hợp các cuộc tấn công tinh vi và lén lút hơn.

* 1. **Kết quả thực hiện được bởi nhóm**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Acc. | TPR | TNR | F1-Score |
| collaborative\_classifers\_gs | 99.85% | 99.91% | 98.70% | 99.92% |
| local\_classifers\_gs | 99.81% | 99.91% | 98.68% | 99.90% |
| local\_classifer\_test | 99.96% | 99.98% | 99.70% | 99.98% |
| single\_classifer\_test | 99.96% | 99.98% | 99.70% | 99.98% |

# Thảo luận

Phần này thảo luận về các khía cạnh liên quan của hiệu suất và thiết kế kiến ​​trúc phải được xem xét khi triển khai trên môi trường B5G thực. Mặc dù hiệu suất trong các thử nghiệm phát hiện phần mềm độc hại đã được chứng minh là cao, nhưng các khía cạnh như chi phí liên lạc hoặc framework centralization nên được thảo luận.

* 1. **Số lượng máy khách và kết quả đối kháng**

Một trong những hạn chế trong thử nghiệm là số lượng máy khách được sử dụng, 8 để đào tạo, do sự sẵn có của bộ dữ liệu phù hợp cho việc học liên kết. Trong kịch bản B5G thực, việc triển khai thiết bị sẽ đạt tới 10 triệu thiết bị trên mỗi km2 theo ITU (Liên minh Viễn thông Quốc tế) yêu cầu [[10]](#muoi). Tuy nhiên, chúng tôi coi rằng các thử nghiệm là hợp lệ, mặc dù số đối thủ thấp, cụ thể là 1, 2 và 3, tỷ lệ phần trăm mà họ đại diện trên tổng số máy khách thực hiện đào tạo là tương đối cao lần lượt là 12,5%, 25% và 37,5% (xem [Fig 6](#fig6)). Như vậy, kết quả có thể được ngoại suy cho các môi trường có số lượng máy khách lớn hơn nhiều nhưng đối thủ chiếm một tỷ lệ nhỏ, không quá 50%.

Ngoài ra, các thuật toán tổng hợp mạnh mẽ khác nên được thử nghiệm vì những thuật toán hiện tại không cung cấp đủ khả năng phục hồi tấn công khi số lượng máy khách nhiễm độc vượt quá 25%. Về vấn đề này, có là những đề xuất thú vị về các thuật toán tổng hợp đã tính đến sự hiện diện có thể có của các máy khách nhiễm độc và đánh giá các biến thể trong các mô hình mà nó chia sẻ. Những điều thú vị nhất được đánh giá là công việc trong tương lai là Krum [[11]](#muoimot), Bulyan [[12]](#muoihai) và AUROR [[13]](#muoiba).

* 1. **Chi phí giao tiếp và tính toán**

Mặc dù yêu cầu thông lượng B5G (100 Mbps trong [[10]](#muoi)) vượt xa các yêu cầu của giải pháp được đề xuất, vì framework được thiết kế cho máy khách ở tại chỗ hoặc gần các điểm truy cập, chi phí truyền thông và tính toán cần được xem xét. Chúng khá quan trọng để không ảnh hưởng đến hoạt động mạng không dây của các đối tượng IoT và các phần tử mạng truy cập vào chúng.

Mini-batch aggregation có chi phí truyền thông cao hơn nhiều so với Multi-epoch aggregation vì nó yêu cầu lần truyền mô hình cho mỗi khách hàng để được đào tạo đầy đủ, trong đó 𝐵 là batch size, 𝐸 là số của epochs và là số lượng mẫu đào tạo của máy khách 𝑘. Ghi chú rằng về mặt chi phí tính toán, nó cũng chỉ ra số lượng bản cập nhật mô hình cục bộ được thực hiện bởi mỗi máy khách. Mặt khác, Multiepoch aggregation chỉ yêu cầu máy khách truyền mô hình tới máy chủ một lần mỗi vòng, với tổng số 𝑇 lần truyền với mỗi máy khách. Tuy nhiên, số lượng bản cập nhật mô hình cục bộ cũng lớn hơn 𝑇 lần, tức là cho máy khách 𝑘.

[Table 6](#table6) cho thấy sự so sánh giữa cả hai thuật toán tổng hợp trong các thí nghiệm của Phần 6.1 về chi phí tính toán và truyền thông. Multi-epoch avg cho thấy chi phí liên lạc thấp hơn nhiều so với Mini-batch avg, ít hơn ≈1300 lần trong trường hợp phương pháp tiếp cận có giám sát và ≈2000 lần trong trường hợp không giám sát. Tuy nhiên, số lần lặp lại đào tạo tại địa phương cao hơn 3,75 lần cho các siêu tham số đã được chọn. Thông lượng trong một mạng 5G hoặc B5G thực tế sẽ đủ để triển khai bất kỳ hai thuật toán. Ngoài ra, như đã nêu trong Phần 4, framework clients sẽ là các trạm gốc B5G và các điểm truy cập khác, có sức mạnh tính toán tương đối cao. Tuy nhiên, nếu chi phí giao tiếp trở thành một vấn đề quan trọng, tự nhiên là chọn cách tiếp cận Multi-epoch avg.

* 1. **Phân cấp và không đồng bộ hóa**

Mặc dù việc đào tạo các mô hình được phân cấp ở mỗi máy khách, có một máy chủ phụ trách tổng hợp mô hìn h có nhiều lợi thế, chẳng hạn như kiểm soát mô hình chung được tạo ra, điều phối giữa các máy khách, v.v. Tuy nhiên, thiết kế này cũng mang lại một số những bất lợi.

Máy chủ trở thành điểm trung tâm của sự cố, nơi tắc nghẽn hoặc tấn công có thể khiến việc không thể tổng hợp các mô hình cục bộ được nữa, và chỉ các mô hình cục bộ mới có thể được sử dụng trên mỗi máy khách. Do đó, cần phải đảm bảo mở rộng quy mô chính xác các chức năng của máy chủ để đảm bảo rằng không có tắc nghẽn và sử dụng các giải pháp bảo mật phù hợp để ngăn chặn các cuộc tấn công vào máy chủ càng nhiều càng tốt. Một giải pháp bổ sung sẽ là điều chỉnh nền tảng theo hướng tiếp cận hoàn toàn phi tập trung, nơi các mô hình được chia sẻ bằng cách sử dụng Blockchain và mỗi khách hàng thực hiện tổng hợp cục bộ, loại bỏ sự cần thiết của người điều phối trong quá trình này.

Một bất lợi khác là cần có sự đồng bộ hóa giữa các máy khách khi gửi các mô hình của họ để tổng hợp. Máy khách bị lỗi hoặc chậm do không đồng bộ có thể khiến máy chủ không thực hiện đào tạo chính xác [[14]](#muoibon). Hiện tại, framework giải quyết vấn đề này bằng cách đặt thời gian chờ để gửi các mô hình để nếu một trong các máy khách không phản hồi kịp thời, nó sẽ bị bỏ qua bước tổng hợp đó. Trong trường hợp này, một giải pháp dựa trên Blockchain cũng có lợi vì nó có thể được sử dụng như một kho lưu trữ không đồng bộ, nơi mỗi khách hàng có thể xuất bản các mô hình của mình mỗi khi đào tạo chúng ở cục bộ.

Mặc dù có những thuận tiện để giải quyết cả hai nhược điểm, nhưng điều cần thiết là phải cân nhắc rằng việc sử dụng Blockchain cũng kéo theo một loạt các mối đe dọa cần giải quyết, chẳng hạn như các cuộc tấn công đa số hoặc tấn công xác thực khối.[[15]](#muoinam). Ảnh có chứa bàn

Mô tả được tạo tự động

# Kết luận và bài học

Công trình này đề xuất một framework bảo vệ quyền riêng tư để phát hiện phần mềm độc hại IoT, thúc đẩy FL để đào tạo và đánh giá cả các mô hình được giám sát và không được giám sát mà không chia sẻ dữ liệu nhạy cảm. Khung này được thiết kế để triển khai trên các nút mạng cung cấp quyền truy cập vào các thiết bị IoT trong mạng WiFi, 5G hoặc B5G, giảm tải tính toán từ chính thiết bị IoT. Theo nghĩa này, phía máy khách được thiết kế để triển khai trên RAN trong khi phía máy chủ được thiết kế để triển khai Fog / Cloud. Để chứng minh tính khả thi của nó trong một kịch bản IoT thực tế, tập dữ liệu N-BaIoT đã được sử dụng do tính không đồng nhất và dễ phân chia về các thiết bị IoT và các mẫu phần mềm độc hại. Sử dụng N-BaIoT, chúng tôi đã so sánh hiệu suất của: (i) phương pháp học liên kết, trong đó tất cả chủ sở hữu thiết bị đào tạo mô hình của riêng họ, được tổng hợp định kỳ trong một máy chủ, (ii) thiết lập không bảo vệ quyền riêng tư, trong đó toàn bộ tập dữ liệu được máy chủ tập trung và đào tạo, và (iii) thiết lập cục bộ trong đó mỗi chủ sở hữu thiết bị đào tạo một mô hình riêng lẻ và riêng biệt. So sánh này đã chỉ ra rằng việc sử dụng dữ liệu đa dạng hơn và lớn hơn, như được thực hiện trong các federated methods và centralized methods, tác động tích cực đáng kể đến hoạt động của mô hình cả trong kịch bản được giám sát và không được giám sát. Bên cạnh đó, nó đã được chứng minh rằng tính riêng tư của dữ liệu có thể được bảo toàn mà không làm mất hiệu suất của mô hình bằng cách tuân theo phương pháp học liên kết. Khả năng phục hồi của các mô hình được liên kết chống lại các máy khách độc hại đã được kiểm tra thông qua các cuộc tấn công đối nghịch sau: (i) a data poisoning attack flipping all labels, (ii) a model poisoning attack multiplying gradients by a negative factor, and (iii) a model cancelling attack. Kết quả cho thấy rằng nếu không sử dụng một kỹ thuật mạnh mẽ để tổng hợp các mô hình, một ứng dụng khách độc hại trong liên kết có thể làm hỏng mô hình. Một số chức năng tổng hợp mạnh mẽ, hoạt động như một biện pháp đối phó với các cuộc tấn công của đối thủ, đã được áp dụng để giải quyết vấn đề này, với tính năng tổng hợp trung bình cho thấy những cải tiến đầy hứa hẹn nhưng vẫn chưa đủ. Bước đầu tiên này theo hướng làm cho hệ thống mạnh mẽ chống lại các cuộc tấn công cho thấy rằng vẫn cần nhiều nỗ lực để đạt được kết quả thỏa mãn.

Trong tương lai, chúng tôi có kế hoạch đánh giá tác động của các cuộc tấn công đối thủ trong kịch bản không được giám sát để xác minh rằng chúng ảnh hưởng đến kết quả theo cách tương tự như đối với đối tác được giám sát. Hơn nữa, thử nghiệm độ bền của mô hình chống lại các cuộc tấn công trốn tránh, sử dụng các mẫu đối thủ giả mạo để tránh bị phát hiện tại thời điểm đánh giá, cũng có thể là một hướng đi thú vị trong tương lai. Ngoài ra, công việc này có kế hoạch nghiên cứu sâu hơn về các biện pháp đối phó hiện có chống lại các cuộc tấn công của đối thủ, chẳng hạn như Krum, Bulyan và AUROR.

Khả năng mở rộng trong các kịch bản B5G thực cũng là một vấn đề mà không thể nghiên cứu với bất kỳ bộ dữ liệu có sẵn nào, làm nảy sinh nhu cầu tạo ra một bộ dữ liệu lớn hơn và đa dạng hơn nhiều. Việc triển khai kiến ​​trúc theo cách phân tán hoàn toàn bằng cách sử dụng Blockchain để trao đổi các mô hình liên kết cũng được xem xét. Bên cạnh đó, việc kết hợp Blockchain vào khuôn khổ có thể cải thiện các mối quan tâm về bảo mật và quyền riêng tư có thể có của khách hàng.

# tài liệu tham khảo

**[1]** K. Riad, T. Huang, and L. Ke, “A dynamic and hierarchical access control for iot in multi-authority cloud storage,” Journal of Network and Computer Applications, vol. 160, p. 102633, 2020.

**[2]** C. L. Stergiou, K. E. Psannis, and B. B. Gupta, “Iot-based big data secure management in the fog over a 6g wireless network,” IEEE Internet of Things Journal, vol. 8, no. 7, pp. 5164–5171, 2020.

**[3]** V. Adat and B. B. Gupta, “Security in internet of things: issues, challenges, taxonomy, and architecture,” Telecommunication Systems, vol. 67, no. 3, pp. 423–441, 2018.

**[4]** P. M. S. Sánchez, J. M. J. Valero, A. H. Celdrán, G. Bovet, , M. G. Pérez, and G. M. Pérez, “A Survey on Device Behavior Fingerprinting: Data Sources, Techniques, Application Scenarios, and Datasets,” IEEE Communications Surveys & Tutorials, In press.

**[5]** M. F. Elrawy, A. I. Awad, and H. F. Hamed, “Intrusion detection systems for iot-based smart environments: a survey,” Journal of Cloud Computing, vol. 7, no. 1, pp. 1–20, 2018.

**[6]** K. Samdanis and T. Taleb, “The road beyond 5g: A vision and insight of the key technologies,” IEEE Network, vol. 34, no. 2, pp. 135–141, 2020.

**[7]** Z. M. Fadlullah and N. Kato, “Hcp: Heterogeneous computing platform for federated learning based collaborative content caching towards 6g networks,” IEEE Transactions on Emerging Topics in Computing, 2020.

**[8]** Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, and Y. Elovici, “N-baiot—network-based detection of iot botnet attacks using deep autoencoders,” IEEE Pervasive Computing, vol. 17, no. 3, pp. 12–22, 2018.

**[9]** B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in Artificial Intelligence and Statistics, ser. Proceedings of Machine Learning Research, A. Singh and J. Zhu, Eds., vol. 54. Fort Lauderdale, FL, USA: PMLR, 20–22 Apr 2017, pp. 1273–1282. [Online]. Available: http: //proceedings.mlr.press/v54/mcmahan17a.html

**[10]** M. Series, “Detailed specifications of the terrestrial radio interfaces of international mobile telecommunications-2020 (imt-2020),” Report, pp. 2410–0, 2021.

**[11]** P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, “Machine learning with adversaries: Byzantine tolerant gradient descent,” in Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 118–128.

**[12]** R. Guerraoui, S. Rouault et al., “The hidden vulnerability of distributed learning in byzantium,” in International Conference on Machine Learning. PMLR, 2018, pp. 3521–3530.

**[13]** S. Shen, S. Tople, and P. Saxena, “Auror: Defending against poisoning attacks in collaborative deep learning systems,” in Proceedings of the 32nd Annual Conference on Computer Security Applications, 2016, pp. 508–519.

**[14]** A. Al-Qerem, M. Alauthman, A. Almomani, and B. Gupta, “Iot transaction processing through cooperative concurrency control on fog– cloud computing environment,” Soft Computing, vol. 24, no. 8, pp. 5695–5711, 2020.

**[15]** M. Saad, J. Spaulding, L. Njilla, C. Kamhoua, S. Shetty, D. Nyang, and D. Mohaisen, “Exploring the attack surface of blockchain: A comprehensive survey,” IEEE Communications Surveys & Tutorials, vol. 22, no. 3, pp. 1977–2008, 2020.