**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CUỐI KỲ**

**MÔN: AN TOÀN HỆ THỐNG THÔNG TIN**

**Đề tài: CÁC THUẬT TOÁN TỔNG HỢP TRONG FEDERATED LEARNING**

Nhóm sinh viên thực hiện:

1. Lê Thị Ngọc Lan MSHV:

2. Vũ Mai Huỳnh Loan MSHV:

3. Lê Bá Trực MSHV: 240202016

GVHD: TS. Nguyễn Tấn Cầm

Tp. Hồ Chí Minh, 10/2024

**THÔNG TIN CHUNG**

|  |  |  |  |
| --- | --- | --- | --- |
| Họ và tên | Lê Thị Ngọc Lan | Vũ Mai Huỳnh Loan | Lê Bá Trực |
| MSHV | 240104010 |  | 240202016 |
| Email | Lanltn.19@grad.uit.edu.vn |  | [Truclb.19@grad.uit.edu.vn](mailto:Truclb.19@grad.uit.edu.vn) |
| Ngành học | Hệ Thống Thông Tin |  | An toàn thông tin |
| Tự đánh giá điểm |  |  |  |
| Hình cá nhân |  |  |  |

Mục Lục

Danh sách các bản báo cáo

Danh sách các hình ảnh

Chương I  
Chương II

1. Triển khai Federated trong An toàn hệ thống thông tin  
   1. Vòng đời Mô hình trong Federated Learning cho Hệ thống Phát hiện Xâm nhập (IDS)

Federated Learning (FL) là một hướng tiếp cận tiên tiến trong việc huấn luyện mô hình học máy, đặc biệt là trong các hệ thống có dữ liệu phân tán và yêu cầu cao về quyền riêng tư. Trong lĩnh vực phát hiện xâm nhập (Intrusion Detection Systems - IDS), FL mang lại giải pháp hiệu quả, vừa đảm bảo sự an toàn dữ liệu, vừa nâng cao hiệu suất của mô hình phát hiện. Dưới đây là phân tích chi tiết về các giai đoạn trong vòng đời của một mô hình FL được áp dụng cho IDS, từ xác định vấn đề, thu thập dữ liệu, đến triển khai và bảo trì.

1. Từ xác định vấn đề đến thu thập dữ liệu

Quá trình phát triển một mô hình FL bắt đầu từ việc xác định rõ ràng mục tiêu cụ thể mà hệ thống IDS cần đạt được. Điều này bao gồm việc trả lời các câu hỏi: Hệ thống này cần phát hiện những dạng tấn công nào? Độ chính xác mục tiêu là bao nhiêu? Có yêu cầu gì về thời gian phản hồi hoặc tỷ lệ cảnh báo sai không?

Một khi mục tiêu đã rõ ràng, các thiết bị khách trong hệ thống, chẳng hạn như tường lửa, router, hoặc thiết bị đầu cuối, được thiết lập để thu thập dữ liệu cục bộ. Dữ liệu này bao gồm nhật ký lưu lượng mạng, sự kiện bảo mật, hoặc hành vi hệ thống. Do dữ liệu nhạy cảm không thể chuyển ra khỏi thiết bị, FL đóng vai trò quan trọng trong việc đảm bảo rằng dữ liệu này vẫn nằm trong giới hạn bảo mật trong khi được sử dụng để huấn luyện mô hình.

1. Huấn luyện mô hình phân tán

Giai đoạn huấn luyện là trọng tâm của FL. Tại đây, các thiết bị khách thực hiện huấn luyện cục bộ trên dữ liệu riêng của chúng. Điều này đảm bảo rằng dữ liệu thô không bị lộ ra ngoài mà vẫn đóng góp vào việc cải thiện mô hình toàn cục.

Quá trình huấn luyện diễn ra như sau:

Khởi tạo: Một mô hình khởi đầu được gửi từ máy chủ trung tâm đến các thiết bị khách.

Huấn luyện cục bộ: Mỗi thiết bị thực hiện tối ưu hóa mô hình dựa trên dữ liệu của mình. Điều này có thể bao gồm việc tính toán gradient hoặc cập nhật trọng số.

Tổng hợp mô hình: Các thiết bị gửi bản cập nhật mô hình (model updates) – thường là gradient hoặc trọng số – về máy chủ trung tâm. Máy chủ này sẽ tổng hợp các bản cập nhật để tạo ra một mô hình toàn cục mới. Quá trình tổng hợp phổ biến nhất là Federated Averaging (FedAvg), trong đó các bản cập nhật được kết hợp dựa trên trọng số của số lượng dữ liệu từng thiết bị.

Lợi thế lớn nhất của phương pháp này là nó tận dụng được sự đa dạng của dữ liệu từ các nguồn khác nhau, giúp mô hình học hỏi từ nhiều dạng lưu lượng mạng và hành vi tấn công khác nhau.

1. Đánh giá và lựa chọn mô hình

Sau khi huấn luyện, bước tiếp theo là đánh giá hiệu suất mô hình. Có hai phương pháp chính để đánh giá:

Kiểm tra với dữ liệu chuẩn: Mô hình được đánh giá trên một tập dữ liệu chuẩn trong trung tâm dữ liệu, thường được sử dụng để so sánh các mô hình khác nhau.

Kiểm tra trên các thiết bị thực tế: Mô hình được triển khai thử nghiệm trên các thiết bị cục bộ chưa tham gia vào quá trình huấn luyện để đánh giá khả năng phát hiện các hành vi bất thường hoặc xâm nhập thực tế.

Các chỉ số đánh giá quan trọng bao gồm:

Độ chính xác (Accuracy): Đo lường khả năng phát hiện đúng các hành vi xâm nhập.

Tỷ lệ cảnh báo sai (False Positive Rate - FPR): Xác định tần suất mô hình báo động sai.

Hiệu quả tính toán: Kiểm tra xem mô hình có đáp ứng được yêu cầu thời gian thực trong môi trường sản xuất không.

Khi quá trình đánh giá hoàn tất, mô hình tốt nhất sẽ được chọn dựa trên hiệu suất tổng thể và khả năng đáp ứng các yêu cầu cụ thể của hệ thống IDS.

1. Triển khai mô hình

Giai đoạn triển khai bao gồm tích hợp mô hình vào môi trường mạng thực tế. Quá trình này không chỉ đơn thuần là đưa mô hình vào hoạt động, mà còn yêu cầu:

Kiểm tra chất lượng: Đảm bảo rằng mô hình hoạt động ổn định và không gây gián đoạn cho hệ thống mạng.

Thử nghiệm A/B: So sánh hiệu suất của mô hình mới với các mô hình hoặc phương pháp hiện có.

Triển khai dần (Staged Rollout): Đưa mô hình vào hoạt động theo từng giai đoạn, từ nhỏ đến lớn, để giảm thiểu rủi ro.

Khi được triển khai, mô hình sẽ liên tục giám sát lưu lượng mạng, phân tích nhật ký và phát hiện các mối đe dọa trong thời gian thực. Hệ thống cũng cần có cơ chế phản hồi, cho phép tự động hóa một phần hoặc toàn bộ các hành động ứng phó với xâm nhập.

1. Bảo trì và cập nhật mô hình

Môi trường an ninh mạng luôn thay đổi với các mối đe dọa mới xuất hiện liên tục. Do đó, một mô hình FL cho IDS cần được bảo trì thường xuyên để đảm bảo hiệu suất tối ưu.

Các hoạt động bảo trì bao gồm:

Giám sát hiệu suất: Theo dõi độ chính xác và khả năng phát hiện của mô hình qua thời gian.

Phát hiện vấn đề: Xác định các kịch bản mà mô hình hoạt động kém, chẳng hạn như tỷ lệ cảnh báo sai tăng cao hoặc bỏ sót các dạng tấn công mới.

Huấn luyện lại định kỳ: Mô hình được huấn luyện lại với dữ liệu mới từ các thiết bị khách. Điều này đảm bảo rằng mô hình học hỏi được từ các xu hướng tấn công mới và duy trì hiệu quả trong phát hiện xâm nhập.

Quá trình bảo trì không chỉ cải thiện hiệu suất của mô hình mà còn tăng cường độ tin cậy của hệ thống IDS trong việc bảo vệ mạng.

**Chương 3:** So sánh các thuật toán tổng hợp: Federated Averaging, Weghted Federated Averaging, Median-based Federated Averaging, Trimmed Mean, Krum (K-Center)

**1. Federated Averaging (FedAvg)**

* **Mô tả**:
  + Trung bình các mô hình từ các client mà không sử dụng trọng số.
  + Mỗi client có ảnh hưởng ngang nhau trong việc tổng hợp mô hình toàn cục.
* **Công thức**:
* **Ưu điểm**:
  + Đơn giản, dễ triển khai.
  + Tính toán nhanh, không yêu cầu thêm thông tin như kích thước dữ liệu.
* **Nhược điểm**:
  + Không hiệu quả nếu dữ liệu không cân bằng giữa các client (client có ít dữ liệu cũng đóng góp ngang bằng client có nhiều dữ liệu).

**Weighted Federated Averaging**

* **Mô tả**:
  + Mở rộng từ FedAvg, sử dụng trọng số dựa trên kích thước dữ liệu của từng client.
  + Các client có nhiều dữ liệu hơn sẽ đóng góp nhiều hơn vào mô hình toàn cục.
* **Công thức**:
* **Ưu điểm**:
  + Giải quyết vấn đề dữ liệu không cân bằng giữa các client.
  + Cải thiện hiệu năng tổng hợp mô hình toàn cục.
* **Nhược điểm**:
  + Yêu cầu thông tin về kích thước dữ liệu của từng client, có thể làm giảm tính bảo mật.

**Median-based Federated Averaging**

* **Mô tả**:
  + Thay vì trung bình, sử dụng trung vị (median) để tổng hợp mô hình.
  + Loại bỏ ảnh hưởng của các giá trị ngoại lai (outliers).
* **Công thức**:
* **Ưu điểm**:
  + Chống lại các mô hình bất thường hoặc độc hại từ các client.
  + Đơn giản và dễ triển khai.
* **Nhược điểm**:
  + Hiệu năng kém trong trường hợp tất cả các client đều hợp tác và không có ngoại lệ.

**Trimmed Mean**

* **Mô tả**:
  + Loại bỏ một số lượng nhỏ các giá trị lớn nhất và nhỏ nhất từ các trọng số trước khi tính trung bình.
  + Bảo vệ mô hình toàn cục khỏi các outliers.
* **Công thức**:
* **Ưu điểm**:
  + Tốt hơn trung vị (median) trong việc giảm thiểu ảnh hưởng của outliers, đặc biệt khi dữ liệu gần Gaussian.
* **Nhược điểm**:
  + Yêu cầu tham số betaβ, đòi hỏi điều chỉnh tùy thuộc vào dữ liệu.

**Krum (K-Center)**

* **Mô tả**:
  + Chọn một trọng số từ các client làm trọng số toàn cục sao cho nó gần với đa số các trọng số khác nhất.
  + Được thiết kế để chống lại các tấn công Byzantine (các client gửi thông tin sai lệch).
* **Công thức**:
* **Ưu điểm**:
  + Hiệu quả trong việc bảo vệ mô hình khỏi các client độc hại.
  + Được thiết kế cho môi trường không tin cậy.
* **Nhược điểm**:
  + Tốn kém về tính toán do cần đo khoảng cách giữa các trọng số.
  + Không hiệu quả nếu tất cả client hợp tác.

A white grid with black text

Description automatically generated