**BÁO CÁO PHẢN BIỆN KHÓA LUẬN TỐT NGHIỆP**

ĐỀ TÀI: NGHIÊN CỨU KHẢ NĂNG KHÁNG MẪU ĐỐI KHÁNG CỦA CÁC TRÌNH PHÁT HIỆN XÂM NHẬP DỰA TRÊN HỌC MÁY.

1. **Giới thiệu đề tài**

* Mục tiêu đề tài: Xem khả năng nhận biết các mẫu độc hại đối kháng của các bộ phân loại, sau đó sẽ dùng các phương pháp phòng thủ lên các bộ phân loại, sau đó đánh giá lại chúng bằng dữ liệu đối kháng khác mà chúng chưa từng biết tới.
* Thực nghiệm trên Dataset CICIDS2017 và CICIOT2023.

1. **Phương pháp đề xuất**

Để tạo mẫu đối kháng và đánh giá khả năng phát hiện tấn công của các mẫu ban đầu và mẫu sau khi thực hiện biến đổi:

Quy trình đề xuất gồm 5 phase:

* Probe phase
* Create adversarial traffic phase
* Another adversarial traffic generation phase
* Excute attack phase
* Defense adversarial training.
  1. **Probe phase**

Trong giai đoạn thăm dò (probe phase), dữ liệu đầu vào là lưu lượng độc hại gốc, sau khi thực hiện quá trình xử lý dữ liệu, các đặc trưng độc hại được đưa vào Detector mục tiêu. Kết quả nhận được là các thông số đánh giá khả năng phát hiện của Detector. Giai đoạn này nhằm đánh giá hiệu suất phát hiện tấn công trước dữ liệu độc hại gốc của các Detector.

A diagram of a process

Description automatically generated

* 1. **Create adversarial traffic phase**

Trong phần này, sử dụng kiến trúc mạng Generator để tạo ra lưu lượng đối kháng từ đầu vào là các noise được tạo ngẫu nhiên. Mô hình có 3 thành phần chính:

* Multimodal Learning.
* Generator
* Discriminator
  + 1. *Multimodal Learning*

Nhằm mục đích tận dụng tối đa thông tin độc hại có trong lưu lượng độc hại ban đầu, nhóm em áp dụng Multimodal Learning để học phân phối dữ liệu từ các nhóm đặc trưng khác nhau. Trong đề tài của nhóm, dữ liệu độc hại ban đầu sẽ được chia thành nhóm là packet và traffic, sau đó các nhóm dặc trưng này sẽ được đưa vào Gaussian RBM để học phân phối dữ liệu của từng nhóm, đầu ra từ các Gaussian RBM này sẽ được đưa vào Upper Gaussian RRM để tổng hợp lại các thông tin từ các Gaussian RBM trên thành vector đại diện chung.

Việc sử dụng thông tin đa phương phức được biễu diễn dưới dạng vector đại diễn chung sẽ giúp Discriminator phân biệt tốt hơn giữa lưu lượng thật và đối kháng được tạo ra. Khi Discriminator trở nên mạnh mẽ thì Generator cần phải tạo dữ liệu đối kháng phức tạp và tinh vi hơn để đánh lừa Generator, chất lượng mẫu đối kháng của Generator sẽ được cải thiện nhiều.

* + 1. *Generator*

Generator có nhiệm vụ tạo ra dữ liệu đối kháng để đánh lừa Disriminator, đầu vào là nhiễu có giá trị được tạo ngẫu nhiên. Sử dụng **torch.randn(self.batch\_size, input\_dim)** có nghĩa là tạo một tensor chứa các giá trị ngẫu nhiên được lấy từ phân phối chuẩn. Ví dụ nếu batch\_size = 64 và input\_dim = 100, thì một tensor có kích thước (64,100) chứa các giá trị ngẫu nhiên từ phân phối chuẩn được tạo ra.

Nhóm chúng em thử nghiệm và chọn thiết kế Generator có cấu trúc như bên dưới.

|  |  |  |  |
| --- | --- | --- | --- |
| Lớp | Chiều vào | Chiều ra | Hàm kích hoạt |
| Đầu vào | Input\_dim | 64 | ReLu |
| Lớp ẩn | 64 | 128 | ReLu |
| Đầu ra | 128 | output\_dim | Tanh |

Generator sẽ nhận đánh giá từ Discriminator và điều chỉnh trọng số để cải thiện khả năng sinh dữ liệu của mình. Hàm mất mát (loss function) của Generator.

LG​= Ez∼p(z)[D(G(z))] .

* + 1. *Discriminator*

Discriminator cố gắng phân biệt giữa dữ liệu thật và dữ liệu do Generator sinh ra.

Nhóm chúng em thử nghiệm và sử dụng kiến trúc Discriminator như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| Lớp | Chiều vào | Chiều ra | Hàm kích hoạt |
| Đầu vào | Input\_dim | 64 | ReLu |
| Lớp ẩn | 64 | 128 | ReLu |
| Đầu ra | 128 | 68 | Sigmoid |

Để cải thiện khả năng phân biệt giữa dữ liệu thật và dữ liệu đối kháng được sinh ra từ Generator, sử dụng hàm Loss sau:

LD​ = Ex∼p​(x)​[D(x)]− Ez∼p​(z)​[D(G(z))]

A diagram of a diagram

Description automatically generated

* 1. **Another adversarial traffic generation phase**

A diagram of a traffic generation phase

Description automatically generated

* Ở phần này, em áp dụng các thuật toán tấn công đối kháng vào mô hình. Bằng cách trích xuất các gradient của mô hình để cần thiết cho việc tấn công đối kháng, cách tạo ra mẫu đối kháng bằng random noise.
* 4 thuật toán đối kháng bao gồm: FGSM, DEEPFOOL, CW VÀ ZOO
* Biểu đồ khi tạo mẫu đối kháng của các thuật toán như sau:
  + FGSM

Ảnh có chứa văn bản, biểu đồ, ảnh chụp màn hình, hàng

Mô tả được tạo tự động

* + Deepfool

Ảnh có chứa văn bản, biểu đồ, ảnh chụp màn hình, hàng

Mô tả được tạo tự động

* + Zoo

Ảnh có chứa văn bản, biểu đồ, hàng, ảnh chụp màn hình

Mô tả được tạo tự động

* + Cw

Ảnh có chứa văn bản, biểu đồ, ảnh chụp màn hình, Kế hoạch

Mô tả được tạo tự động

* 1. **Excute attack phase**
* Sau khi tạo mẫu đối kháng thành công, thì sẽ đem các mẫu đối kháng đi test lên các bộ phân loại.
  1. **Defense adversarial training.**

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

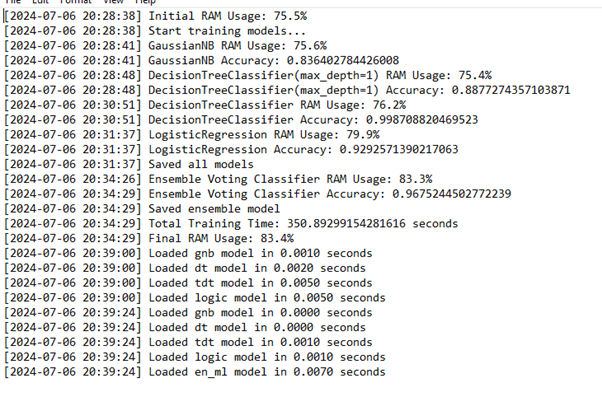
Mô tả được tạo tự động

* Có thể chia ra làm 2 phần chính là Tranning Adversarial và Testing Adversarial. Cụ thể:
  + Tranning Adversarial: Cùng với dữ liệu gốc, chúng tôi sẽ trộn dữ liệu đối kháng đã được tạo ra bởi các thuật toán đã nêu. Sau đó chúng tôi tiến hành xử lí dữ liệu và thu thập những thông tin cần thiết như label, feature để có thể nạp vào các mô hình để Tranning.
  + Testing Adversarial: Phần dữ liệu đem đi test, xử lí dữ liệu và trích xuất các mẫu là Malicious đem đi tạo mẫu đối kháng mới (khác với những mẫu đã được huấn luyện ở phần tranning). Sau đó đưa vào mô hình đã được huấn luyện với mẫu đối kháng để nó dự đoán

1. **Thực nghiệm**
   1. **Detector**

* Xây dựng các bộ phân loại
  + Machine learning
    - Gaussian Naive Bayes(GNB).
    - Decision Tree (DT).
    - Logistic Regression (LR).
  + Deep Learning
    - Deep Neural Networks (DNN) với 6 layers.
    - Long Short-Term Memory (LSTM) với 5 layers.
  + Mô hình tổng hợp
    - Ensemble Machine Learning: Từ GNB, LR và DT
    - Ensemble Deep Learning: Từ DNN 6 layers và LSTM 5 layers.
  1. **Triển khai các thuật toán tấn công đối kháng**
* Fgsm với số lượng nhiễu được thêm vào là epsilon=0.01, 0.02,0.05,0.1 và 0.2
* Zoo , CW, deepfool với số lượng vòng lặp tối ưu để tạo ra nhiễu là 10,50,100,200
  1. **Defense**
* Sẽ đem dữ liệu đối kháng đi huấn luyện với dữ liệu gốc, sau đó đánh giá bộ phân loại đã được học lên mẫu đối kháng khác mà nó chưa biết
  1. **Kết quả thu được**
     1. *Phase 1 huấn luyện và testting mô hình*

-          Log khi mô hình tranning các detector



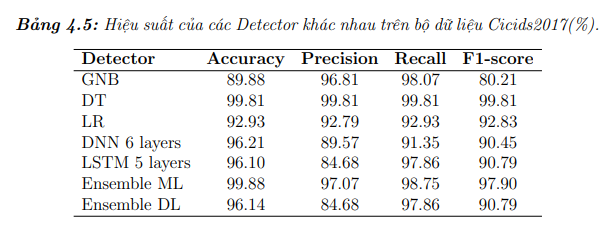
* Biểu đồ hội tụ khi tranning mô hình DNN và LSTM

Ảnh có chứa văn bản, biểu đồ, Sơ đồ, hàng

Mô tả được tạo tự độngẢnh có chứa văn bản, biểu đồ, hàng, Sơ đồ

Mô tả được tạo tự động

-          Kết quả phase1:



A table with numbers and text

Description automatically generated

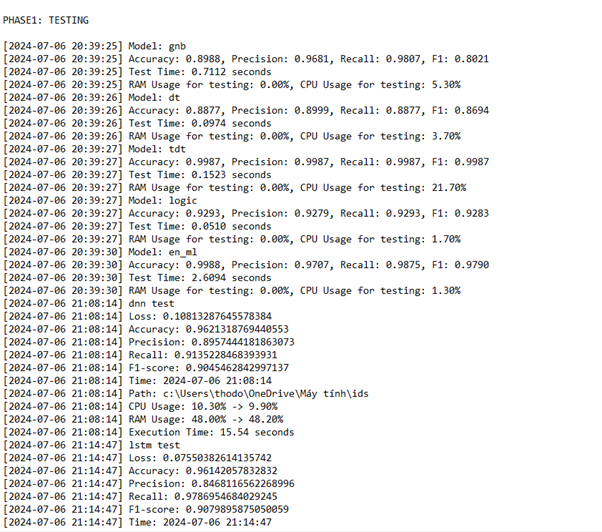
-          Một vài confusion\_matrix của bảng trên

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, biểu đồ

Mô tả được tạo tự động

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  | |  |
|  | Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, biểu đồ  Mô tả được tạo tự động |  |
|  |  |  |

-          Log:



* + 1. *Phase tấn công đối kháng*
* **Log tấn công:**

**Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động**

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, tài liệu

Mô tả được tạo tự động

* Kết quả khi tấn công đối kháng trên bộ dữ liệu CICIDS2017.

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, Phông chữ

Mô tả được tạo tự độngẢnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, Phông chữ

Mô tả được tạo tự độngẢnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

* Một vài confusion\_matrix của bảng trên (vì em chỉ đưa các mẫu độc hại đi tạo mẫu đối kháng , nên phần dự đoán mẫu bình thường (nhãn 0) sẽ ở giá trị là 0)

A screenshot of a computer

Description automatically generated

A table with text and numbers

Description automatically generated

* Mẫu đối kháng tạo ra có thể gây ra mô hình nhận diện sai.
  + 1. *Chuyển giao đối kháng*

**Ảnh có chứa văn bản, ảnh chụp màn hình, số, Phông chữ

Mô tả được tạo tự động**

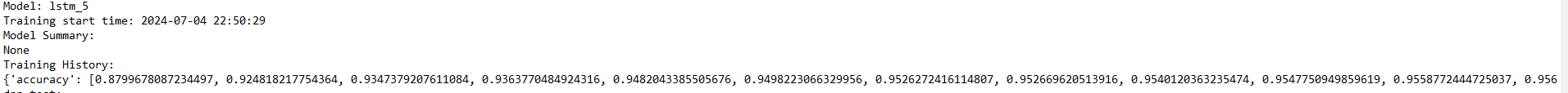
**Ảnh có chứa văn bản, ảnh chụp màn hình, số, Phông chữ

Mô tả được tạo tự động**

**Ảnh có chứa văn bản, số, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động**

* **Các thuật toán tấn công đối kháng tạo ra mẫu đối kháng có khả năng chuyển giao sang các mô hình khác và tác động lên mô hình.**
  + 1. Phase phòng thủ trước các cuộc tấn công đối kháng chưa biết
* Log huấn luyện phòng thủ

****

* Log tấn công mới

**Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động**

* Kết quả khi kiểm tra

**Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động**

**Ảnh có chứa văn bản, ảnh chụp màn hình, số, Phông chữ

Mô tả được tạo tự động**

**Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động**

**Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động**

* Một vài confusion\_matrix của bảng trên: (vì em chỉ đưa các mẫu độc hại đi tạo mẫu đối kháng , nên phần dự đoán mẫu bình thường (nhãn 0) sẽ ở giá trị là 0)

**Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, thiết kế

Mô tả được tạo tự động**

**Ảnh có chứa văn bản, ảnh chụp màn hình, Hình chữ nhật, Phông chữ

Mô tả được tạo tự động**

* Nhận xét rằng mô hình có khả năng nhận biết các cuộc tấn công đối kháng chưa biết tới từ thấp đến tốt.

1. **Kết luận**

* Ưu điểm:
  + Giải quyết được bài toán đặt ra là nghiên cứu được khả năng kháng đối kháng của các mẫu đối kháng lên các trình xâm nhập (bộ phân loại).
  + Đạt được mục tiêu là xây dựng được các bộ phân loại, triển khai được các thuật toán tấn công đối kháng đem lại hiệu quả tốt và triển khai được phương pháp phòng thủ và đánh giá tính Robustness của các mô hình.
* Nhược điểm.
  + Sử dụng nhiễu ngẫu nhiên đưa vào Generator sinh mẫu đối kháng còn đơn giản.
  + Chưa triển khai được nhiều hơn các phương pháp phòng thủ khác ngoàiAdversarial Training. Vì vậy chưa đánh giá được các phương pháp phòng thủ khác có ưu điểm hơn phương pháp phòng thủ mà chúng tôi sử dụng trong khóa luận lần này hay không.