TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN – ĐHQG-HCM
Khoa Mạng máy tính & Truyền thông

Tổng quan: Học máy đối kháng

NT204 – Hệ thống tìm kiếm, phát hiện và ngăn ngừa xâm nhập

GV: Đỗ Hoàng Hiển

hiendh@uit.edu.vn

Nội dung hôm nay…

Học máy đối kháng

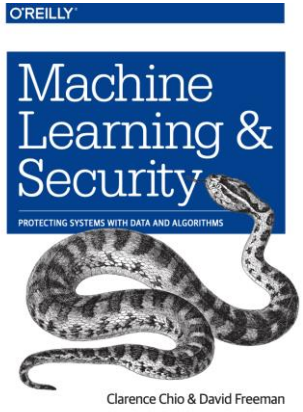
Hôm nay có gì?

Học máy đối kháng

- Tổng quan về các nghiên cứu gần đây

Tham khảo:

1. Chio, C., & Freeman, D. (2018). *Machine Learning & Security* book (Chapter 8)
2. Y. Vorobeychik and M. Kantarcioglu, "*Adversarial Machine Learning*," Synthesis Lectures on AI and Machine Learning
3. Materials (slides and related papers) from AAAI 18 (https://aaai18adversarial.github.io/)



Học máy trong thực tế

Xe tự hành

Y tế

Thành phố thông minh

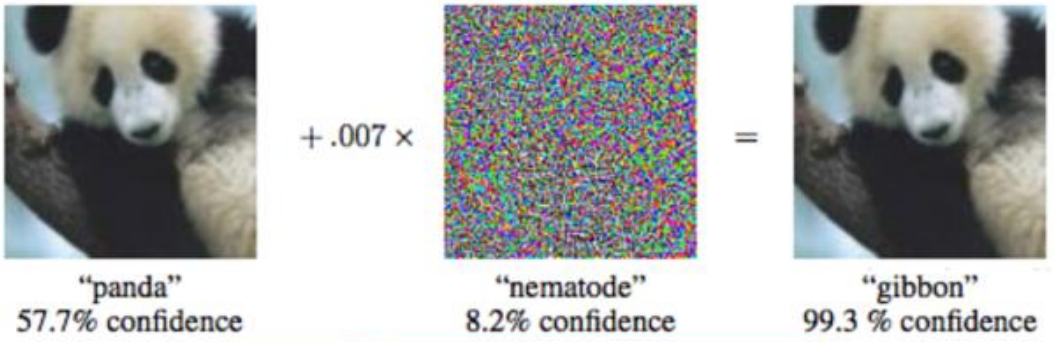
Phân loại mã độc

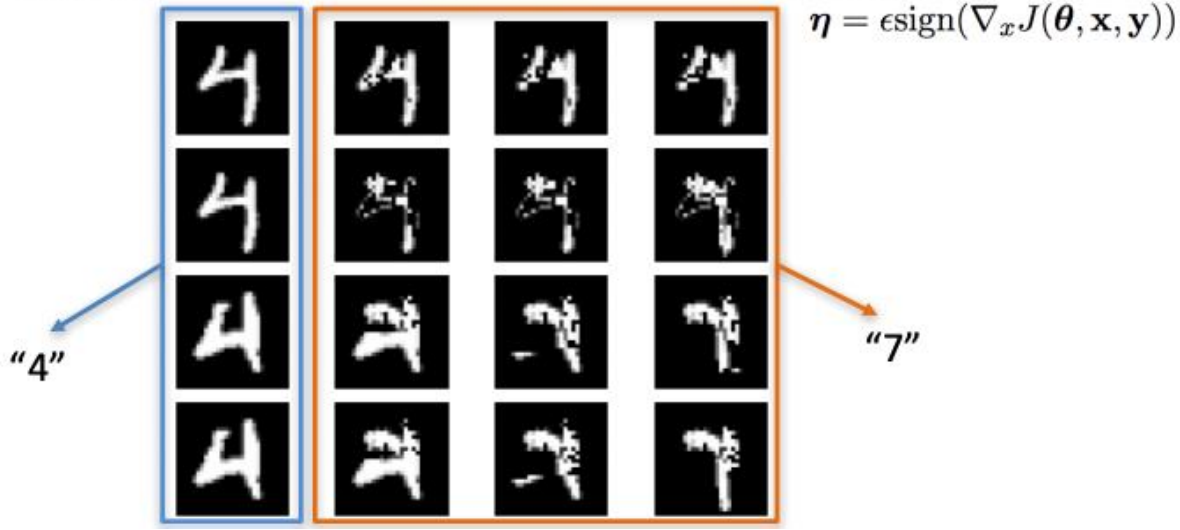
Phát hiện lừa đảo

Nhận diện sinh trắc học

Ví dụ về đối kháng – Adversarial (1)

GoogLeNet

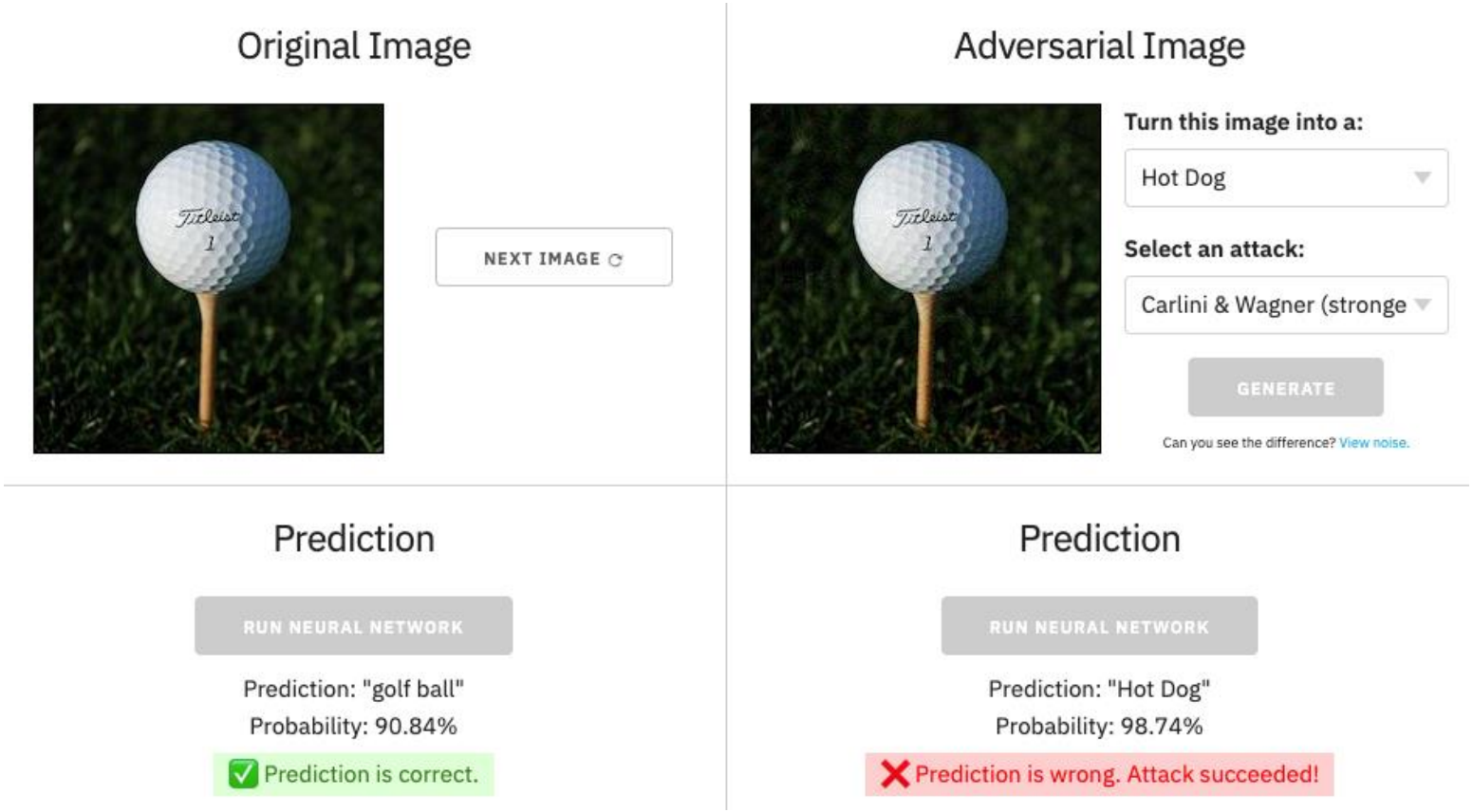




- Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." ICLR 2015
- Li, Bo, Yevgeniy Vorobeychik, and Xinyun Chen. "A General Retraining Framework for Scalable Adversarial Classification." ICLR. (2016).

Ví dụ về đối kháng – Adversarial (2)

adversarial.js



Tham khảo: https://kennysong.github.io/adversarial.js/

Thế giới thực rất … lộn xộn

Adversarial perturbations – Xáo trộn đối kháng luôn có trong thực tế dưới những điều kiện hay góc nhìn khác nhau, bao gồm khoảng cách và góc độ

Điều kiện vật lý khác nhau (góc độ, khoảng cách, ánh sáng…)

Độ nhạy

Giả mạo/ Lỗi nhận diện (tái tạo màu sắc, v.v…)

Nhiễu kỹ thuật số

Màu mong muốn | Màu khi in | Màu camera thấy

Thay đổi background

Evtimov, Eykholt, Fernandes, Kohno, Li, Prakash, Rahmati, and Song, 2017

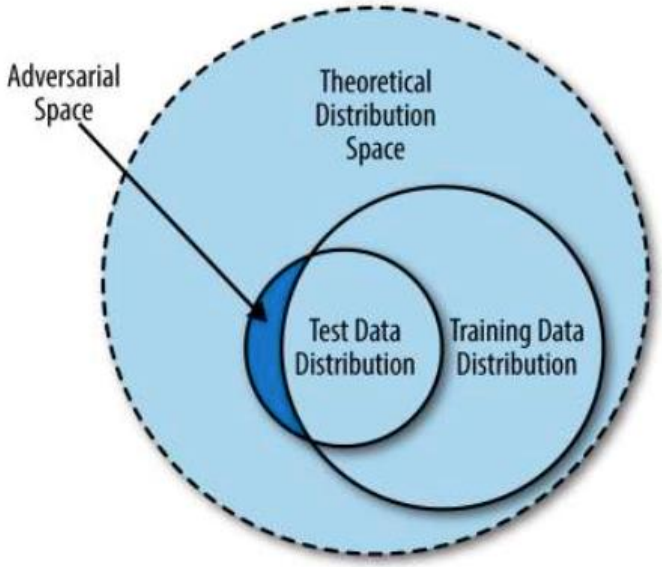
Học máy đối kháng

- *Học máy đối kháng - Adversarial machine learning (AML)* nghiên cứu về các lỗ hổng của học máy trong **môi trường đối kháng**
  - Các nhà nghiên cứu bảo mật cũng như học máy đã công bố nhiều nghiên cứu thực nghiệm tấn công vào các *giải pháp dựa trên học máy như antivirus, spam filter, hệ thống phát hiện xâm nhập, bộ phân loại ảnh, bộ phân tích tâm lý, v.v…*
- Vấn đề chính trong AML: **kẻ xấu** (thực hiện hành vi xấu) có **mục đích:**
  - Để không bị phát hiện (là chủ yếu)
  - Có thể thay đổi hành vi để tránh bị phát hiện
- Các lỗ hổng trong các hệ thống học máy có thể do: lỗi khi thiết kế hệ thống, giới hạn của thuật toán hoặc kết hợp cả 2 nguyên nhân
- Học máy đối kháng rất khó vì hầu hết các giải pháp học máy đều hoạt động dạng **black box**

Tham khảo: Chio, C., & Freeman, D. (2018). *Machine Learning & Security* book (Chapter 8)

Các lỗ hổng bảo mật trong các thuật toán ML

- Các kỹ thuật ML thường được phát triển với giả định về *dữ liệu ổn định*, *các thuộc tính độc lập* và *ít ngẫu nhiên*
  - Các tập training và testing được giả định lấy từ không gian mẫu có phân bố không đổi theo thời gian, và các thuộc tính được lựa chọn cũng được giả định độc lập với nhau và phân bố đều
  - Các thuật toán ML thường không được thiết kế để hoạt động hiệu quả trong các môi trường đối kháng khi các giả định trên thường bị phá vỡ
- Nhiều lỗ hổng của học máy phát sinh từ vấn đề cơ bản của việc **học không triệt để - imperfect learning**

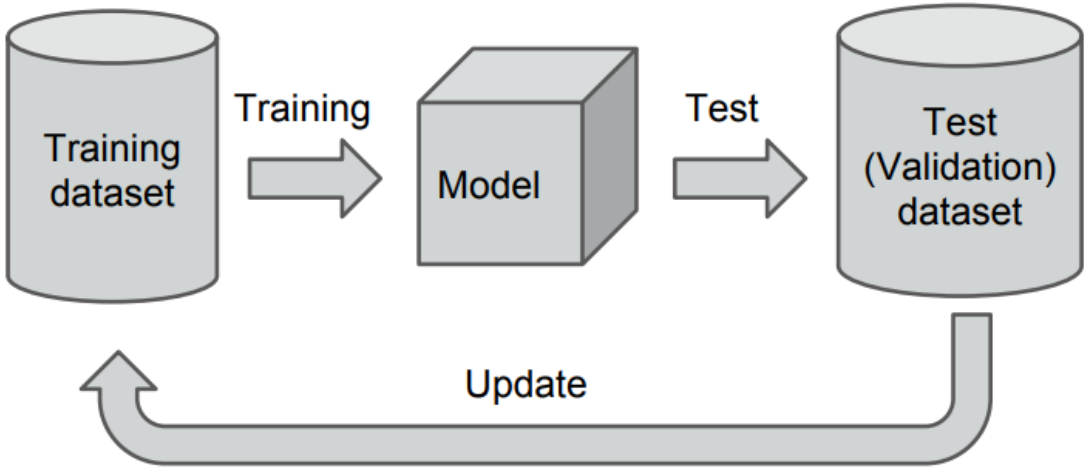


Không gian đối kháng là kết quả của biểu diễn không triệt để trong dữ liệu training

Tham khảo: Chio, C., & Freeman, D. (2018). *Machine Learning & Security* book (Chapter 8)

Bức tranh tổng quan

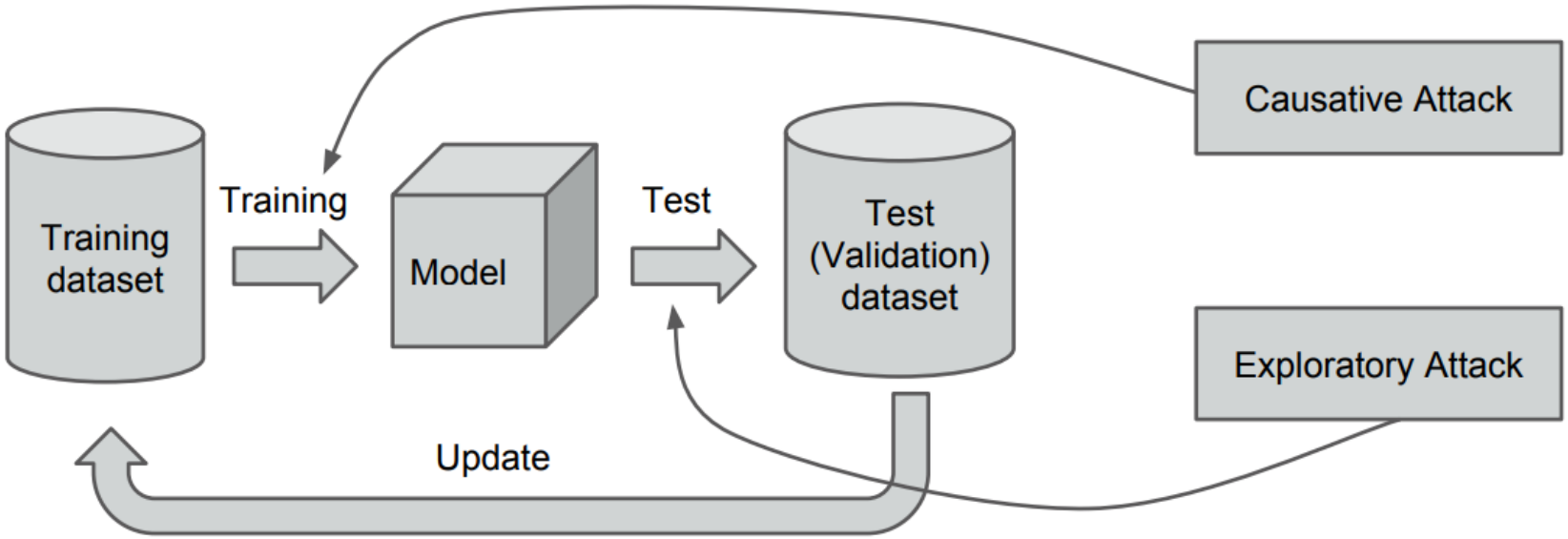
- Các hệ thống học hiện đại có thực sự an toàn?



- Tăng lỗi trong giai đoạn test
- Giảm độ chính xác của quá trình học
- Đánh lừa hệ thống thông minh
- Đạt được mục đích cá nhân

Bức tranh tổng quan

- Các hệ thống học hiện đại có thực sự an toàn?



Khái niệm

- **Ảnh hưởng**
  - ***Causative attacks (Poisoning – Đầu độc)***: các tấn công từ các tác nhân đối kháng nhắm đến **quá trình train** bằng cách giả mạo **dữ liệu train** hoặc **tham số train**
    - Hiểu được cách hoạt động của thuật toán học
    - Thao tác trên các thuộc tính hoặc nhãn của tập train
    - Thay đổi chức năng phân loại/phân biệt
    - Ví dụ: tấn công Flipping-label, backdoor
  - ***Exploratory attacks (Evasion – Qua mặt)*** dựa trên giao tiếp với hệ thống ML **sau khi được train,** tìm và khai thác *không gian đối kháng* để khiến model tạo ra các lỗi không mong muốn
    - Thao tác trên các thuộc tính của dữ liệu test
    - Ngăn hoạt động phát hiện thông thường
    - Thay đổi kết quả phân loại/phân biệt
    - Ví dụ: brute-force fuzzing không gian input của bộ phân loại học máy để tìm các mẫu bị phân loại sai

Khái niệm

- **Đặc trưng**
  - ***Indiscriminate attacks (Non-targeted – không mục tiêu cụ thể)*** là các tấn công không cụ thể của các tác nhân đối kháng, muốn các model đưa ra các quyết định sai nhưng không quan tâm kết quả cuối cùng của hệ thống
    - Mục tiêu là khiến bộ phân loại dự đoán nhầm thành bất kỳ nhãn nào khác ngoài nhãn đúng
    - Hầu hết các nghiên cứu hiện nay đều xem xét mục tiêu này
    - Ví dụ: khiến các sample thuộc họ malware A bị phân loại nhầm thành *bất kỳ loại nào khác nhãn của họ malware A*
  - ***Targeted attacks (Tấn công có mục tiêu)*** thường cố gắng khiến kết quả dự đoán của model chuyển hướng sang 1 kết quả thay thế cụ thể
    - Mục tiêu là khiến bộ phân loại dự đoán nhầm thành 1 nhãn mục tiêu → khó hơn
    - Ví dụ: tấn công có mục tiêu vào 1 bộ phân loại họ malware có thể khiến các mẫu thuộc họ malware A chắc chắn bị phân loại nhầm sang họ malware B

Khái niệm

- **Vi phạm an ninh thông tin**
  - Tấn công vào đặc tính **Bảo mật - Confidentiality** (hoặc **privacy**) thường cố gắng lấy được các thông tin nhạy cảm/quan trọng từ các hệ thống ML
  - Tấn công vào đặc tính **Toàn vẹn – Integrity** thường khiến mô hình ML hoạt động lỗi, tuy nhiên quan trọng là **âm thầm** mắc lỗi
    - **Phân loại sai có source/target cụ thể:** từ 1 lớp cụ thể sang 1 lớp cụ thể khác
    - **Phân loại sai có mục tiêu:** từ 1 lớp cụ thể sang bất kỳ lớp nào khác
    - **Phân loại sai:** từ bất kỳ lớp nào sang bất kỳ lớp nào khác
  - Tấn công vào đặc tính **Sẵn sàng – Availability** thường giảm khả năng sử dụng/hoạt động của hệ thống, nhằm hạ gục hoàn toàn hệ thống ML

Đọc thêm: https://towardsdatascience.com/will-my-machine-learning-be-attacked-6295707625d8

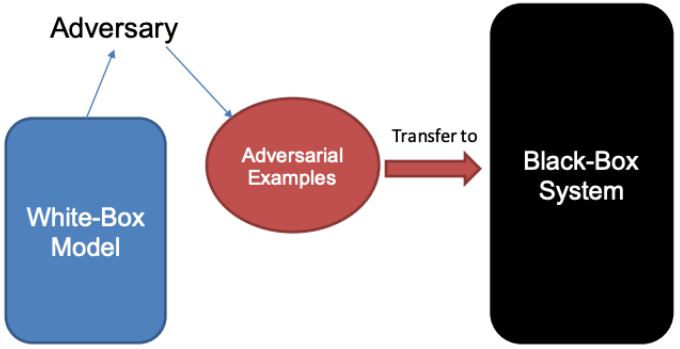
Khả năng tấn công của attacker

- Tấn công **White-Box** giả định rằng attacker biết về phân bố dữ liệu (có thể truy cập được 1 số phần dữ liệu), kiến trúc của mô hình, thuật toán tối ưu được dùng, trọng số và bias
- Tấn công **Black-Box** giả định rằng attacker không biết gì về hệ thống ML (không biết bất kỳ thông tin gì ở trên)
  - Có 2 dạng: **hard label** (attacker chỉ nhận được nhãn đã được dự đoán từ model phân loại) và **confidence** (attacker nhận được nhãn dự đoán cũng như điểm tin cậy từ model phân loại)
- Tấn công **Grey-Box** ở giữa 2 trường hợp tấn công trên
  - Ví dụ, attacker có thể biết mô hình nhưng không biết về dữ liệu được dùng – hoặc ngược lại

Read more: https://towardsdatascience.com/will-my-machine-learning-be-attacked-6295707625d8

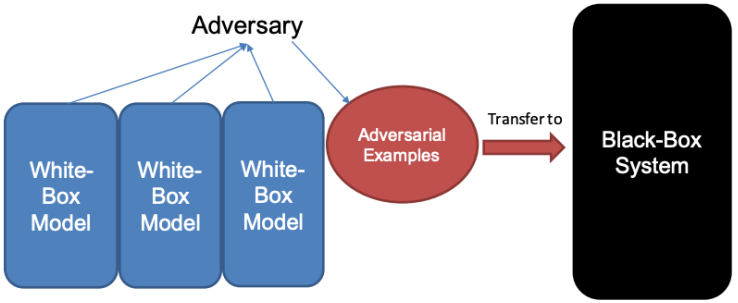
Framework chung cho tấn công Black-box

- **Zero-Query Attack**
  - Thêm nhiễu ngẫu nhiên
  - Chênh lệch trung bình
  - Tấn công dựa trên chuyển nhượng
    - Các tấn công Black-box thực tế vào ML
    - Tấn công dựa trên chuyển nhượng kết hợp



Black-box Attacks Based On Transferability

- **Query Based Attack**
  - Ước tính gradient chênh lệch hữu hạn
  - Ước tính gradient giảm truy vấn
  - Kết quả: hiệu quả giống tấn công white-box



Ensemble Targeted Black-box Attacks Based On Transferability

Zero-query attack có thể xem là trường hợp đặc biệt của query-based attack, khi số lượng query = 0

Liu, Chen, Liu, Song. Delving into Transferable Adversarial Examples and Black-box Attacks,ICLR 2017

Chuyển nhượng tấn công

- **Attack transferability – Chuyển nhượng tấn công:** các mẫu đối kháng (từ không gian đối kháng) được thiết kế đặc biệt nhằm khiến 1 mô hình phân loại sai cũng có khả năng khiến các model khác cũng phân loại sai – ngay cả khi các mô hình có thuật toán hoặc kiến trúc khác nhau

- Tấn công đối kháng qua mặt trên các model Black-box: các bộ phân loại chưa có thông tin về kỹ thuật ML hoặc mô hình

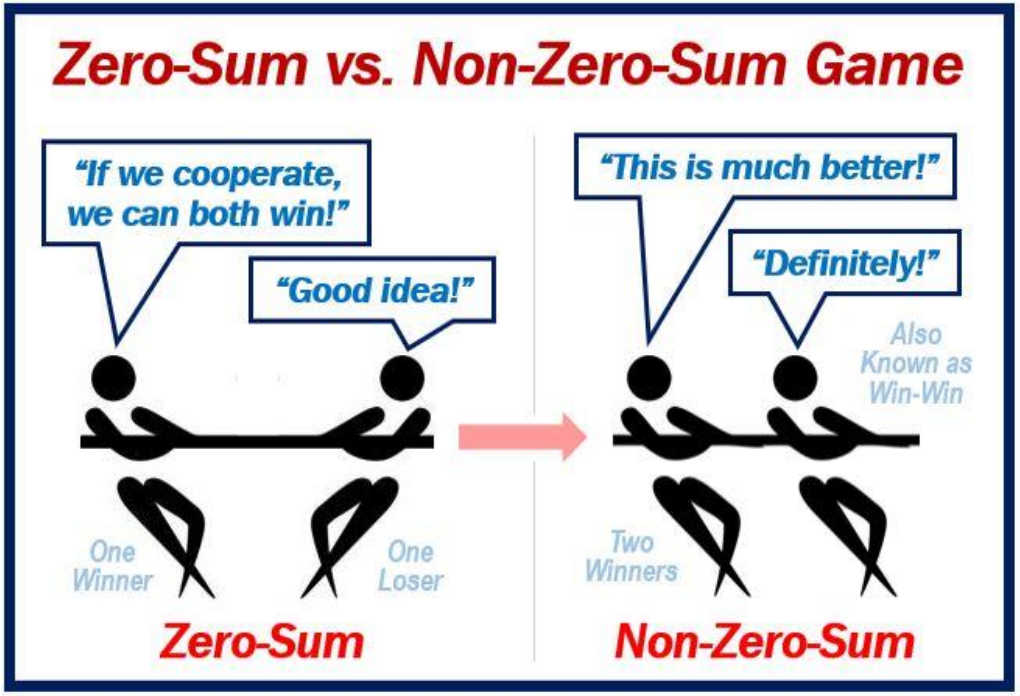
  →Tạo tập dữ liệu training đã có gán nhãn để huấn luyện 1 mô hình nội bộ *tương tự*

  →Tìm kiếm các mẫu thuộc không gian đối kháng để đánh lừa mô hình black-box mục tiêu

  ➔ Sử dụng **Mạng tạo sinh đối kháng - Generative Adversarial Network (GAN)**

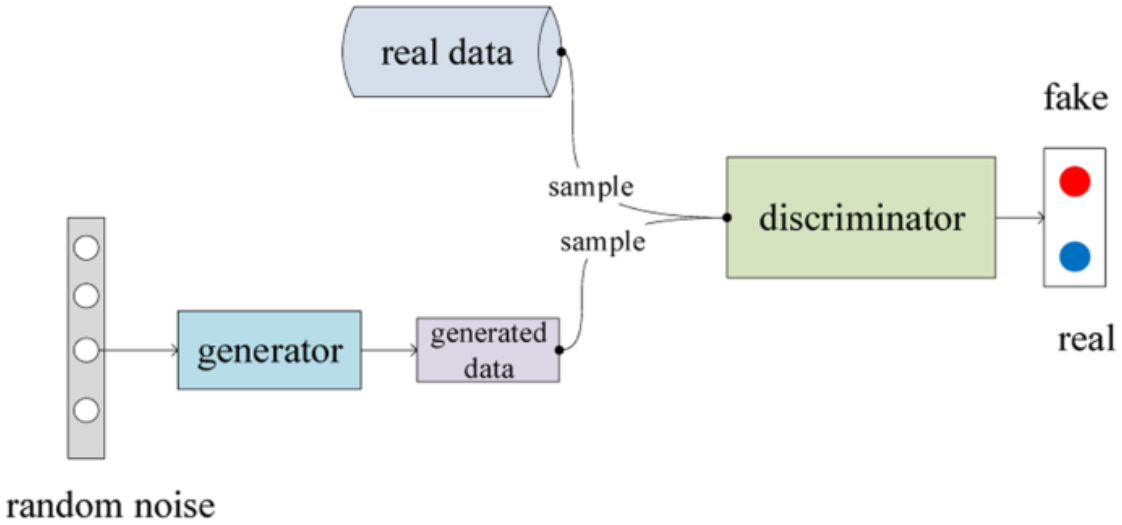
Mạng tạo sinh đối kháng - Generative Adversarial Network

- Thuật toán máy học không giám sát sử dụng 2 mạng neuron "đấu tay đôi với nhau" trong 1 trò chơi **zero-sum-game**



Mạng tạo sinh đối kháng - Generative Adversarial Network

- Thuật toán máy học không giám sát sử dụng 2 mạng neuron "đấu tay đôi với nhau" trong 1 trò chơi **zero-sum-game**



- GAN không có quan hệ trực tiếp với học máy đối kháng, nhưng kỹ thuật này trong thực tế đã được dùng để tạo ra tên miền C&C để tấn công qua mặt các model phát hiện tấn công dựa trên học máy

Kỹ thuật tấn công

Tấn công Đầu độc trên ML

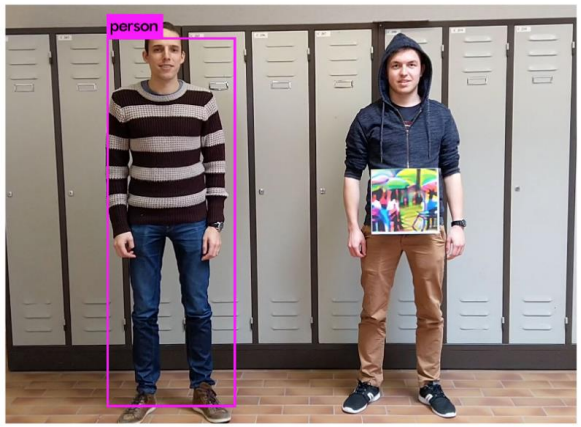
- Tấn công **đầu độc** xảy ra khi tác nhân đối kháng có thể chèn **dữ liệu xấu** vào dữ liệu train của mô hình, khiến mô hình học 1 số thứ không nên học
  - Tấn công đầu độc thường là ***cố ý*** và có thể khác nhau tuỳ vào đặc điểm hoặc loại **vi phạm an ninh thông tin**
- Trong đó, kẻ tấn công tạo ra các thay đổi độc hại trong dữ liệu huấn luyện
  - Thêm dòng dữ liệu (ví dụ, gửi các email – bình thường hoặc tấn công – được thiết kế đặc biệt)
  - Thay đổi dòng dữ liệu (tấn công vào 1 trong các server lưu trữ 1 phần dữ liệu)
  - Có thể loại bỏ 1 số dòng dữ liệu có chọn lọc
- Trong tấn công đầu độc, attacker thường được giả định có thể kiểm soát 1 phần dữ liệu huấn luyện được dùng cho thuật toán ML

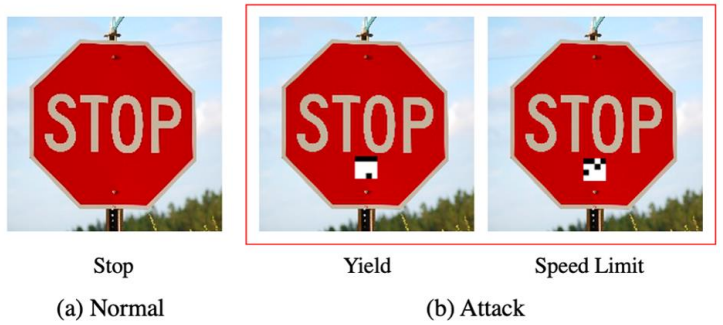
Tham khảo:
- https://aaai18adversarial.github.io/
- https://towardsdatascience.com/poisoning-attacks-on-machine-learning-1ff247c254db

Kỹ thuật tấn công

Tấn công Đầu độc trên ML

- **Phân loại:**
  - Tấn công nhắm vào tính **availability** của ML: thêm rất nhiều dữ liệu xấu vào hệ thống ML, từ đó bất kỳ ranh giới phân loại nào mô hình học được đều vô dụng
    - Ví dụ. **Label-flipping attacks**
  - Tấn công nhắm vào tính **integrity** của ML: **backdoor attacks**
    - Backdoor là 1 dạng input người thiết kế model không để ý, nhưng attacker có thể lợi dụng để khiến hệ thống ML làm bất kỳ điều gì mà attacker muốn



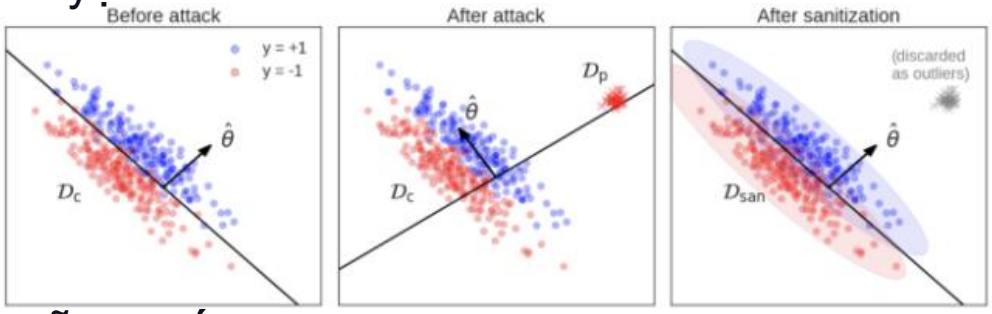


Source: https://portswigger.net - https://arxiv.org/abs/1904.08653

Kỹ thuật tấn công

Đối phó với Tấn công Đầu độc trên ML

- Phát hiện bên ngoài: **Làm sạch dữ liệu** và **phát hiện bất thường (Data sanitization** và **anomaly detection)**
  - **Ý tưởng:** khi tấn công đầu độc hệ thống ML, attacker thường thêm các thứ rất khác thường vào dữ liệu huấn luyện
  - **Thách thức**: "bên trong"?



- Phân tích ảnh hưởng của các mẫu huấn luyện mới trên **độ chính xác của mô hình**
  - **Ý tưởng:** nếu 1 input đã bị đầu độc, nó sẽ phá huỷ độ chính xác của mô hình trên tập test, và có thể phát hiện ra bằng cách chạy với mẫu mới trên sandbox trước khi thêm vào tập huấn luyện.

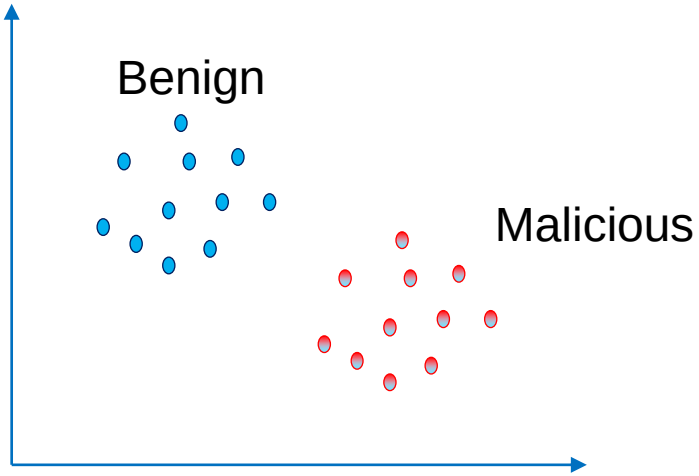
→ Không có rule đủ **đúng và chính xác** chắc chắn chặn được tấn công đầu độc

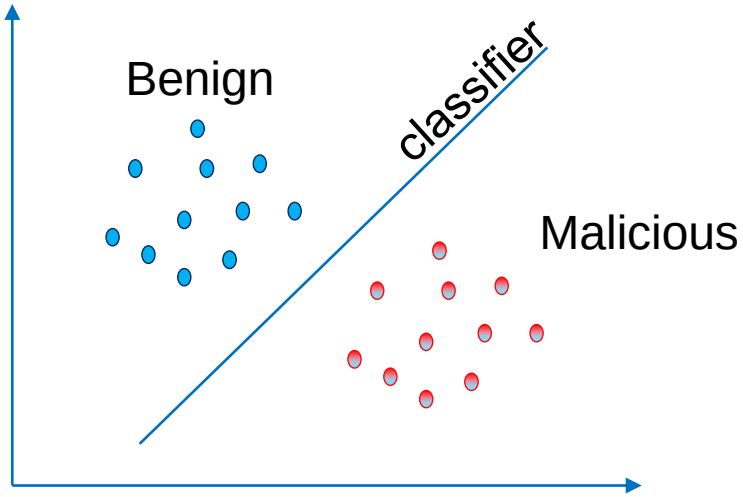
Đọc thêm: https://towardsdatascience.com/poisoning-attacks-on-machine-learning-1ff247c254db

Kỹ thuật tấn công

Evasion Attack – Tấn công qua mặt

- ***Evasion attack – Tấn công qua mặt:*** Khai thác không gian đối kháng để tìm các **mẫu đối kháng,** khiến bộ phân loại ML phân loại sai
- Kẻ thù trước đó đã chọn thể hiện x (hiện được phân loại là độc hại), giờ tìm một thể hiện x' khác được phân loại là lành tính





Ref: https://aaai18adversarial.github.io/

Ví dụ tấn công Evasion

1. From: spammer@example.com
Cheap mortgage now!!!

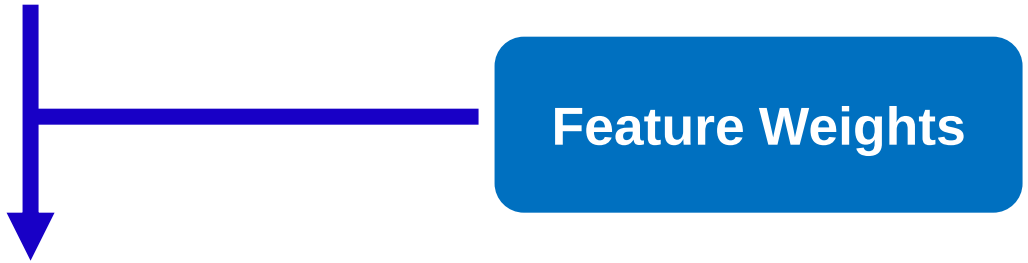
2. cheap = 1.0
mortgage = 1.5

3. Total score = 2.5 > 1.0 (threshold)

Ref: https://aaai18adversarial.github.io/

Ví dụ tấn công Evasion

1. From: spammer@example.com
Cheap mortgage now!!!
Joy Oregon



2. cheap = 1.0
mortgage = 1.5

Joy= -1.0
Oregon = -1.0

3. Total score = **0.5** < 1.0 (threshold)

OK

Ref: https://aaai18adversarial.github.io/

Kỹ thuật tấn công

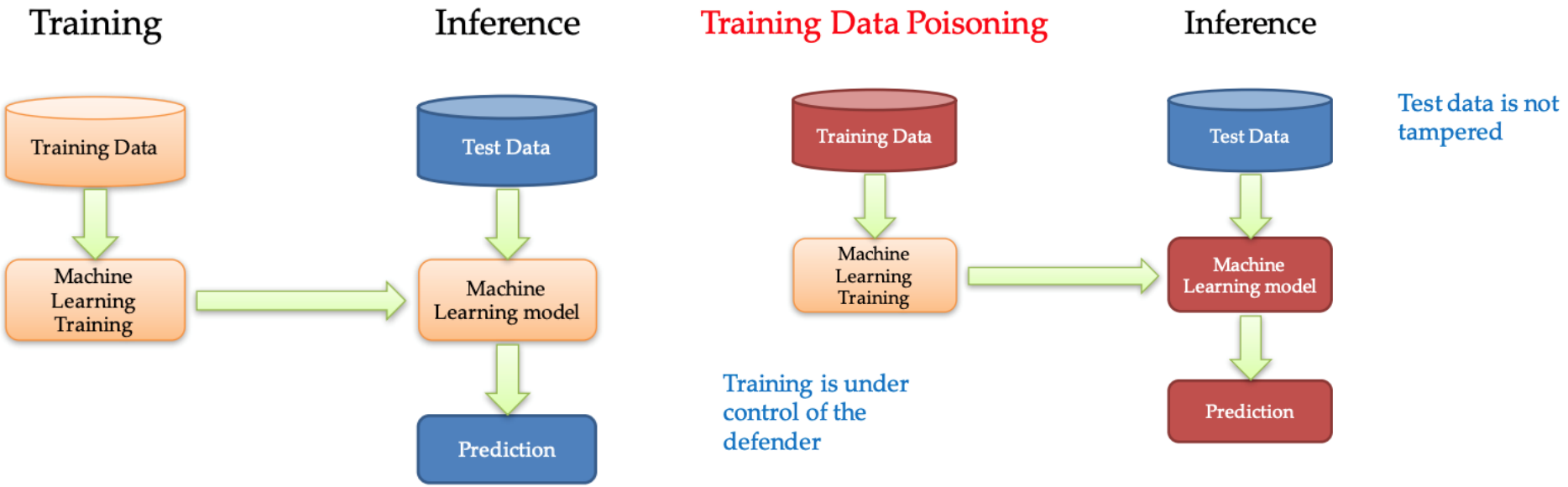
Đối phó với tấn công Evasion

- **Retraining - Tái huấn luyện** (*Adversarial training - Huấn luyện đối kháng)*
  - Bắt đầu với dữ liệu gốc
  - Sử dụng *bất kỳ thuật toán ML nào* để học được mô hình *f*
  - Với các dữ liệu độc hại, sử dụng *bất kỳ phương pháp qua mặt nào* để tạo dữ liệu mới x' để thêm vào tập dữ liệu
  - Lặp lại quá trình
  - Dừng khi:
    - Không có dữ liệu mới để thêm
    - Đạt tới giới hạn về số vòng lặp
    - Bộ phân loại ít thay đổi giữa các vòng lặp liên tiếp
- Robust Learning bằng **Regularization**

Tham khảo: https://aaai18adversarial.github.io/

Vẫn còn 1 chặng đường dài để đối phó…

- Các mẫu đối kháng **không dễ phát hiện**
- Model tấn công tốt hơn
- Ngày càng hiểu về các mạng neural



Tổng kết

- Điều kiện tiên quyết cho các giải pháp bảo mật dựa trên ML là bản thân ML phải an toàn và mạnh mẽ.
- **Tấn công poisoning** và **evasion** không phải là minh chứng cho "sự thất bại" của học máy, mà chỉ ra những quy chuẩn không đúng về kỳ vọng đối với những gì học máy có thể thực hiện trong các tình huống thực tế.
- Biết về các loại lỗ hổng khác nhau của ML trong các môi trường đối kháng có thể thúc đẩy các thiết kế hệ thống tốt hơn và giúp giảm số lượng các giả định sai lầm về khả năng của ML.

Câu hỏi/thắc mắc (nếu có)???

Today end, **See you next week!**