ĐỒ ÁN CÁ NHÂN

**ĐÁNH GIÁ TÁC ĐỘNG CỦA KĨ THUẬT CHƯNG CẤT KIẾN THỨC CHỐNG LẠI CÁC CUỘC TẤN CÔNG ĐỐI KHÁNG**

*Người thực hiện: Trần Văn Bình*

**TP. HỒ CHÍ MINH, NĂM 2024**

**MỤC LỤC**

[LỜI MỞ ĐẦU 1](#_Toc183692536)

[A. NGỮ CẢNH 1](#_Toc183692537)

[B. PHẠM VI ĐỀ TÀI 1](#_Toc183692538)

[C. MỤC ĐÍCH 2](#_Toc183692539)

[DANH MỤC HÌNH ẢNH 3](#_Toc183692540)

[DANH MỤC BẢNG 5](#_Toc183692541)

[DANH MỤC TỪ VIẾT TẮT PHỔ BIẾN 6](#_Toc183692542)

[THUẬT NGỮ LIÊN QUAN PHỔ BIẾN 7](#_Toc183692543)

[CHƯƠNG I. CƠ SỞ LÝ THUYẾT 8](#_Toc183692544)

[A. KNOWLEDGE DISTILLATION 8](#_Toc183692545)

[I. KHÁI NIỆM 8](#_Toc183692546)

[II. PHƯƠNG PHÁP HUẤN LUYỆN 16](#_Toc183692547)

[III. CÁCH HOẠT ĐỘNG 18](#_Toc183692548)

[IV. KĨ THUẬT TỐI ƯU HÓA 20](#_Toc183692549)

[V. TRƯỜNG HỢP SỬ DỤNG 21](#_Toc183692550)

[VI. ƯU NHƯỢC ĐIỂM 22](#_Toc183692551)

[VII. TẦM QUAN TRỌNG 23](#_Toc183692552)

[VIII. THUẬT NGỮ LIÊN QUAN 24](#_Toc183692553)

[B. ADVERSARIAL ATTACKS 26](#_Toc183692554)

[I. KHÁI NIỆM 26](#_Toc183692555)

[II. MỐI ĐE DỌA 28](#_Toc183692556)

[III. CÁCH HOẠT ĐỘNG 32](#_Toc183692557)

[IV. CÁCH PHÒNG VỆ 38](#_Toc183692558)

[V. CÔNG CỤ HỖ TRỢ 39](#_Toc183692559)

[CHƯƠNG II. TRIỂN KHAI THỰC NGHIỆM 41](#_Toc183692560)

[A. KIẾN TRÚC 41](#_Toc183692561)

[B. THỰC NGHIỆM 42](#_Toc183692562)

[C. KẾT QUẢ 52](#_Toc183692563)

[CHƯƠNG III. TỔNG KẾT 57](#_Toc183692564)

[A. KẾT LUẬN 57](#_Toc183692565)

[B. HƯỚNG PHÁT TRIỂN 58](#_Toc183692566)

[TÀI LIỆU THAM KHẢO 59](#_Toc183692567)

# LỜI MỞ ĐẦU

## A. NGỮ CẢNH

Trong bối cảnh hiện nay, thời đại công nghệ 4.0, các hệ thống học máy (Machine Learning) được áp dụng rộng rãi trong nhiều lĩnh vực như nhận dạng hình ảnh, phân tích dữ liệu tài chính, và điều khiển tự động,... Tuy nhiên, những mô hình này dễ bị tấn công trước các thay đổi nhỏ nhưng có chủ ý trong dữ liệu đầu vào, được gọi là **các cuộc tấn công đối kháng (Adversarial Attacks - AAs).** Những thay đổi này có thể làm suy giảm đáng kể độ chính xác và độ tin cậy của các mô hình, gây hậu quả nghiêm trọng trong các ứng dụng quan trọng như y tế, giao thông, và bảo mật.

Cùng với đó, kỹ thuật **Knowledge Distillation (KD)** ngày càng được biết đến nhiều hơn với mục đích giảm kích thước mô hình thông qua việc chuyển giao tri thức từ mô hình teacher (phức tạp) sang mô hình student (đơn giản hơn) và được kỳ vọng tăng cường độ bền vững trước các cuộc tấn công đối kháng nhờ vào việc học từ các “soft targets” của mô hình teacher, hay còn gọi giải pháp phòng vệ **Defensive Distillation**.

## B. PHẠM VI ĐỀ TÀI

Phạm vi đề tài tập trung vào việc **kiểm tra độ bền của các mô hình học máy** trước các cuộc tấn công đối kháng với kỹ thuật Knowledge Distillation. Cụ thể:

* **Mô hình sử dụng:** Hai mô hình **MLP** (Multi-Layer Perceptron) được triển khai, gồm một mô hình teacher (kiến trúc, tham số phức tạp hơn) và một mô hình student (kiến trúc, tham số đơn giản hơn).
* **Dữ liệu thử nghiệm:** Tabular data (**iotid20.csv**) được sử dụng để huấn luyện và kiểm tra mô hình.
* **Tấn công đối kháng:** Áp dụng 5 cuộc tấn công đối kháng thuộc loại trốn tránh (**Evasion Attack**) để đánh giá khả năng chống chịu của các mô hình.
* **Phân tích hiệu suất:** So sánh các tiêu chí đánh giá của mô hình học máy (Accuracy, Precision, Recall và F1-score) trước và sau các cuộc tấn công, đồng thời đánh giá mức độ cải thiện khi áp dụng KD.

Phạm vi thực nghiệm sử dụng các công cụ chính như:

* **PyTorch:** Một framework (khung) học máy dựa trên thư viện Torch được sử dụng để triển khai các mô hình học sâu (Deep Learning).
* **Adversarial Robustness Toolbox (ART):** Một library (thư viện) để thực hiện các cuộc tấn công đối kháng và đánh giá khả năng phòng thủ.

## C. MỤC ĐÍCH

Mục đích chính của đề tài là:

* **Đánh giá hiệu quả của Knowledge Distillation:** Xác định vai trò của KD trong việc bảo vệ mô hình teacher và student trước các cuộc tấn công đối kháng, đồng thời so sánh độ bền của mô hình teacher và student.
* **Kiểm tra khả năng chống chịu/độ bền của mô hình:** Phân tích và đo lường độ giảm hiệu năng của các mô hình sau khi bị tấn công.
* **Đóng góp vào nghiên cứu về độ bền của mô hình học máy:** Cung cấp góc nhìn mới về việc tích hợp KD như một phương pháp phòng thủ hiệu quả trong lĩnh vực bảo mật trí tuệ nhân tạo.

*Toàn bộ nội dung trong đồ án cá nhân chỉ được phép sử dụng cho mục đích tham khảo, vui lòng không sao chép và sửa đổi nếu chưa được sự cho phép của tác giả.*

# DANH MỤC HÌNH ẢNH

[Hình 1: Kĩ thuật Knowledge Distillation 8](#_Toc183692665)

[Hình 2: Thành phần Knowledge của kĩ thuật Knowledge Distillation 9](#_Toc183692666)

[Hình 3: Response-based knowledge 10](#_Toc183692667)

[Hình 4: Feature-based knowledge 11](#_Toc183692668)

[Hình 5: Relation-based knowledge 12](#_Toc183692669)

[Hình 6: Distillation Schemes của kĩ thuật Knowledge Distillation 16](#_Toc183692670)

[Hình 7: Cách hoạt động kĩ thuật Knowledge Distillation 19](#_Toc183692671)

[Hình 8: Mối đe dọa của học máy đối kháng 29](#_Toc183692672)

[Hình 9: Ứng dụng của ML trong nhiều lĩnh vực 30](#_Toc183692673)

[Hình 10: Cách hoạt động của tấn công đối kháng 32](#_Toc183692674)

[Hình 11: Các loại chính của tấn công đối kháng 33](#_Toc183692675)

[Hình 12: Minh họa tấn công Evasion 34](#_Toc183692676)

[Hình 13: Minh họa tấn công Poisoning 35](#_Toc183692677)

[Hình 14: Minh họa tấn công Inference 36](#_Toc183692678)

[Hình 15: Minh họa tấn công Extraction 37](#_Toc183692679)

[Hình 16: Red Teams và Blue Teams của ART 40](#_Toc183692680)

[Hình 17: Kiến trúc thực nghiệm 41](#_Toc183692681)

[Hình 18: TeacherMLP model 42](#_Toc183692682)

[Hình 19: StudentMLP model 44](#_Toc183692683)

[Hình 20: Knowledge Distillation (KD) 46](#_Toc183692684)

[Hình 21: Fast Gradient Method (FGM) Attack: 48](#_Toc183692685)

[Hình 22: Projected Gradient Descent (PGD) Attack 49](#_Toc183692686)

[Hình 23: DeepFool Attack 50](#_Toc183692687)

[Hình 24: Carlini and Wagner L\_2 Attack 51](#_Toc183692688)

[Hình 25: Sign-OPT Attack 52](#_Toc183692689)

[Hình 26: Hiệu năng ban đầu của các mô hình 53](#_Toc183692690)

[Hình 27: Độ bền của mô hình trước Fast Gradient Method (FGM) 54](#_Toc183692691)

[Hình 28: Độ bền của mô hình trước Projected Gradient Descent (PGD) 54](#_Toc183692692)

[Hình 29: Độ bền của mô hình trước Deepfool 55](#_Toc183692693)

[Hình 30: Độ bền của mô hình trước Carlini and Wagner L\_2 55](#_Toc183692694)

[Hình 31: Độ bền của mô hình trước Sign-OPT 56](#_Toc183692695)

# DANH MỤC BẢNG

[Bảng 1: Kết quả đồ án 53](#_Toc183692469)

# DANH MỤC TỪ VIẾT TẮT PHỔ BIẾN

|  |  |
| --- | --- |
| **Từ viết tắt** | **Ý nghĩa** |
| AI | Artificial Intelligence  Trí tuệ nhân tạo |
| ML | Machine Learning  Học máy |
| DL | Deep Learning  Học sâu |
| NN | Neural Network  Mạng nơ-ron nhân tạo |
| CNN | Convolutional Neural Networks  Mạng nơ-ron tích chập |
| KD | Knowledge Distillation  Chưng cất kiến ​​thức |
| AAs | Adversarial Attacks  Các cuộc tấn công đối kháng trong học máy |
| AE | Adversarial Example  Mẫu đối kháng |
| ART | Adversarial Robustness Toolbox  Một thư viện của tấn công đối kháng |
| MLP | Multilayer Perceptron  Một loại mô hình học máy |

# THUẬT NGỮ LIÊN QUAN PHỔ BIẾN

***Artificial Intelligence (AI) – Trí tuệ nhân tạo*** là khả năng giải quyết vấn đề một cách thông minh của máy tính, tương tự như khả năng giải quyết vấn đề của con người.

***Machine Learning (ML) – Học máy*** là một lĩnh vực của trí tuệ nhân tạo liên quan đến việc nghiên cứu và xây dựng các kĩ thuật cho phép các hệ thống “học” tự động từ dữ liệu để giải quyết những vấn đề cụ thể.

***Deep Learning (DL) – Học sâu*** là một cách tiếp cận để học máy được đặc trưng bởi các chồng tính toán sâu cho phép mô hình “học” sâu giải quyết các loại mẫu phức tạp.

***Neural Network (NN) – Mạng nơ-ron nhân tạo*** là hệ thống của các tế bào thần kinh nhân tạo, mô phỏng cấu trúc và hoạt động của não người, được sử dụng trong máy học để nhận diện mô hình và học từ dữ liệu.

***Convolutional Neural Networks (CNN) – Mạng nơ-ron tích chập*** là một loại mô hình học sâu hiệu quả tốt trong nhận dạng hình ảnh và dữ liệu có cấu trúc lưới. Các lớp chập giúp mô hình nhận diện đặc trưng của dữ liệu, giảm số lượng tham số và cải thiện hiệu suất đáng kể.

***Knowledge Distillation (KD) – Chưng cất kiến ​​thức*** là kỹ thuật trong học máy nhằm truyền đạt hay chuyển giao kiến thức từ một mô hình phức tạp (teacher) sang một mô hình nhỏ hơn (student).

***Adversarial Attacks (AAs) – Các cuộc tấn công đối kháng trong học máy*** là các kỹ thuật trong học máy nhằm làm cho mô hình học máy đưa ra dự đoán sai lệch và đưa ra kết quả không chính xác.

***Adversarial Example (AE) – Mẫu đối kháng*** là dữ liệu đầu vào đã được chỉnh sửa một cách tối thiểu để gây ra lỗi dự đoán cho mô hình học máy, thường sử dụng trong các cuộc tấn công đối kháng để kiểm tra độ bền của mô hình trước các thay đổi nhỏ trong dữ liệu.

***Adversarial Robustness Toolbox (ART) – Một thư viện của tấn công đối kháng*** mã nguồn mở hỗ trợ việc tạo ra các cuộc tấn công đối kháng, cũng như các kỹ thuật bảo vệ và cải thiện độ bền của mô hình trước các cuộc tấn công này.

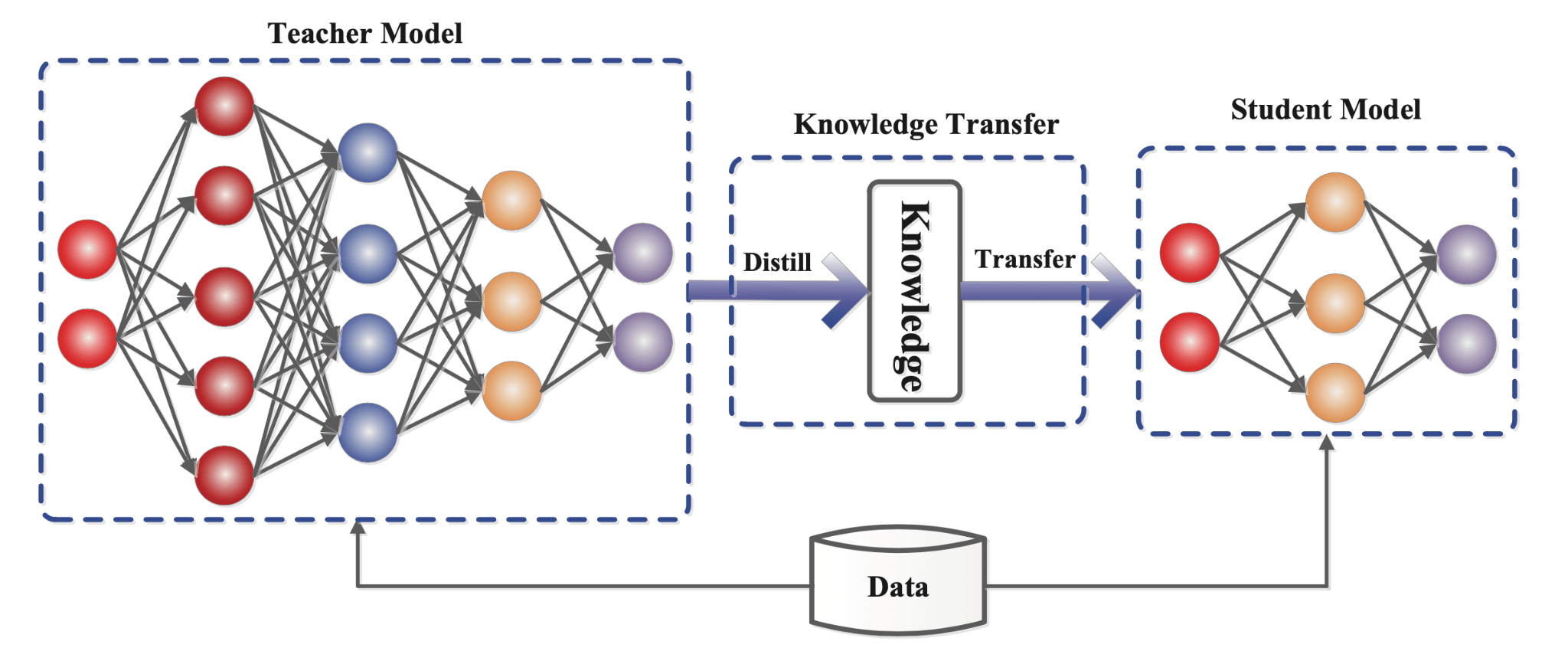
# CHƯƠNG I. CƠ SỞ LÝ THUYẾT

*Nội dung chương này sẽ trình bày* ***kiến thức tổng quan*** *về kĩ thuật Knowledge Distillation và Adversarial Attacks trong lĩnh vực Machine Learning.*

## A. KNOWLEDGE DISTILLATION

### I. KHÁI NIỆM

Knowledge Distillation (KD) là một kỹ thuật cho phép chuyển giao kiến thức từ một mô hình lớn, phức tạp hơn (thường được gọi là mô hình giáo viên – Teacher) sang một mô hình nhỏ, đơn giản hơn (thường được gọi là mô hình học sinh – Student) mà không mất hiệu lực và vẫn giữ được hiệu suất (các metrics của Student (S) tương đương hoặc cao hơn so với Teacher (T)). Điều này cho phép triển khai trên phần cứng ít mạnh mẽ hơn, làm cho đánh giá nhanh hơn và hiệu quả hơn.



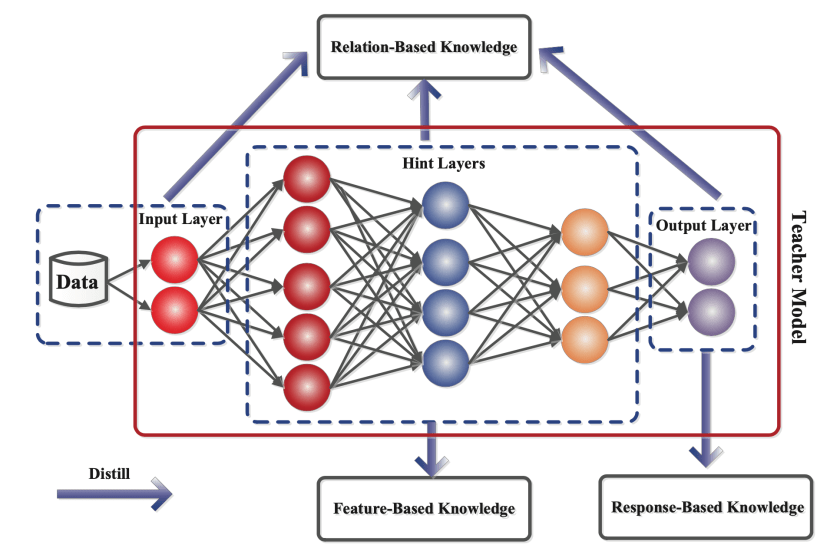
Hình 1: Kĩ thuật Knowledge Distillation

Mục tiêu của knowledge distillation là giảm dung lượng bộ nhớ, yêu cầu tính toán và chi phí năng lượng của một mô hình lớn, để một mô hình nhỏ hơn sau khi được huấn luyện với kĩ thuật KD từ mô hình lớn có thể được sử dụng trong môi trường hạn chế về tài nguyên mà không làm giảm đáng kể hiệu suất. Knowledge distillation là một cách hiệu quả để cải thiện độ chính xác của mô hình ML tương đối nhỏ, bản chất quá trình này đôi khi được gọi là việc học tập của giáo viên/học sinh (teacher/student learning).

Kiến trúc của Knowledge Distillation bao gồm 3 thành phần chính là: ***knowledge***, ***distillation algorithm*** và ***teacher-student architecture***.

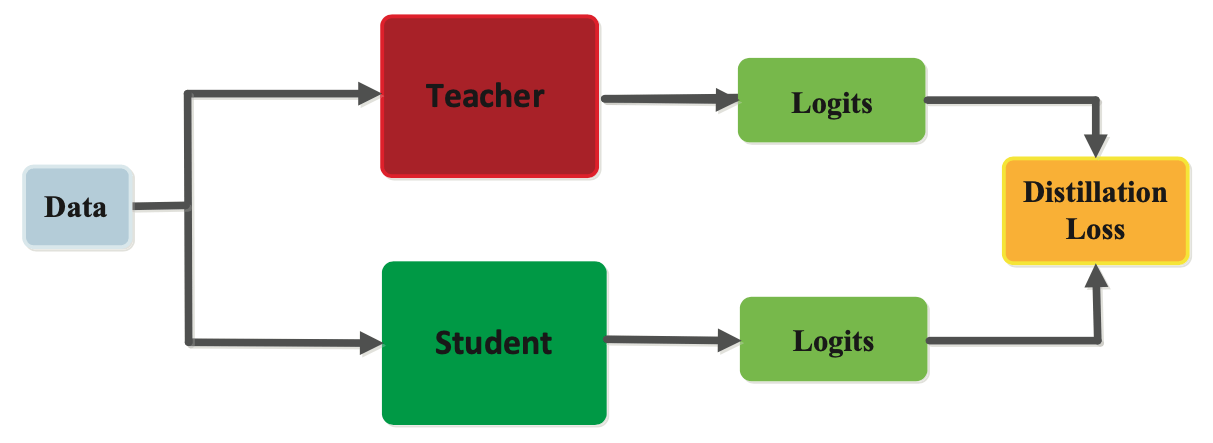
***1. Knowledge***

Có ba loại knowledge (kiến thức) chính được truyền từ mô hình giáo viên sang mô hình học sinh, dựa vào cách thông tin được thu thập từ mô hình giáo viên:



Hình 2: Thành phần Knowledge của kĩ thuật Knowledge Distillation

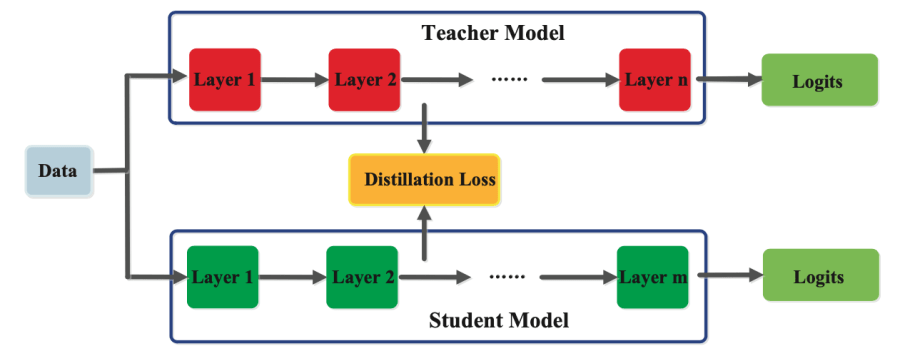
**+ Response-based knowledge:** Tập trung vào final output layer. Nắm bắt sự khác biệt giữa các log của S và T tương ứng.



Hình 3: Response-based knowledge

* **Đặc điểm:** Thông tin được thu thập từ ***lớp đầu ra*** của mô hình giáo viên, nơi chứa các xác suất dự đoán (logits). Mô hình học sinh sẽ học để ***bắt chước các xác suất*** này.
* **Cách tính mất mát (Loss):** Sử dụng Kullback-Leibler Divergence ***(KL Divergence)*** để so sánh sự khác biệt giữa các logits của mô hình giáo viên và học sinh.
* **Ứng dụng:** Thường sử dụng trong các bài toán như nhận diện đối tượng đa lớp, nơi các logits của mô hình giáo viên không chỉ dự đoán nhãn mà còn chứa thông tin về vị trí bounding box trong ảnh.

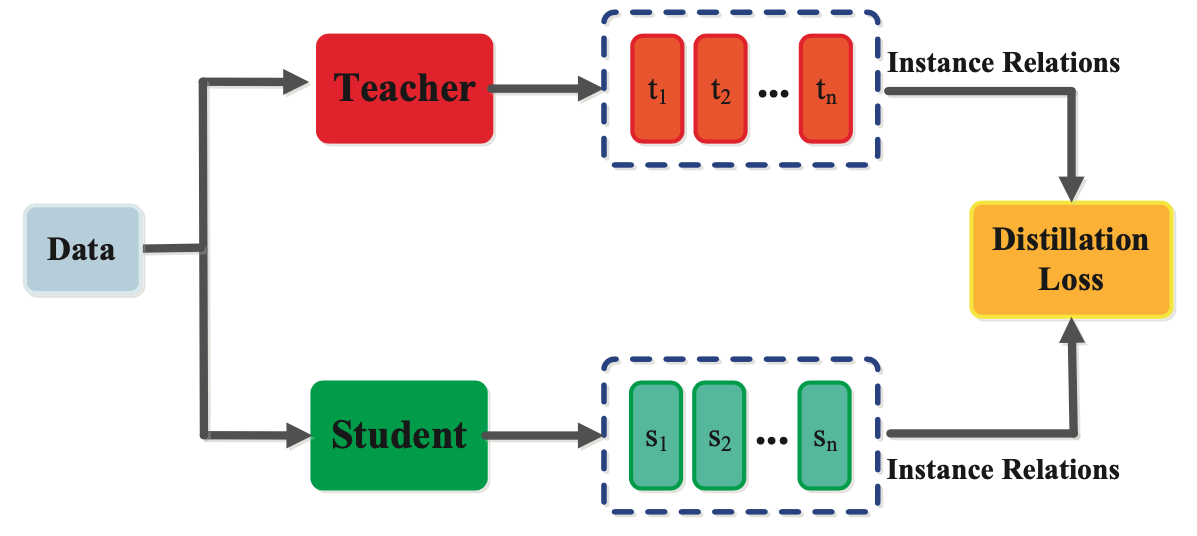
**+ Feature-based knowledge:** Tập trung vào intermediate layers. Mục tiêu: huấn luyện S cách kích hoạt đặc trưng tương tự như T. Distillation loss đạt được điều này bằng cách giảm thiểu sự khác biệt giữa kích hoạt đặc trưng của T và S.



Hình 4: Feature-based knowledge

* **Đặc điểm:** Mỗi lớp trung gian trong mô hình sâu học các đặc trưng khác nhau, từ thấp đến cao. Các bản đồ đặc trưng **(*feature maps*)** từ các lớp trung gian này được sử dụng làm kiến thức. Đây là cách để mô hình học sinh học từ các lớp trung gian, không chỉ lớp đầu ra.
* **Ứng dụng:** Ví dụ trong bài toán phân loại ảnh, các bản đồ đặc trưng từ lớp áp chót của mô hình giáo viên được sử dụng. Điều này giúp mô hình học sinh học tốt hơn khi dữ liệu có sự tương đồng trực quan giữa các lớp.
* **Lưu ý:** Hệ số nhiệt độ ***(temperature factor)*** thường được áp dụng để điều chỉnh độ “mềm” của các xác suất, kết hợp cùng chuẩn hóa L2 để giảm nhiễu trong dữ liệu.

**+ Relation-based knowledge:** Mô hình hóa dưới dạng tương quan giữa các feature maps, graphs, similarity matrix, feature embeddings, hoặc probabilistic distributions dựa trên biểu diễn đặc trưng.



Hình 5: Relation-based knowledge

* **Đặc điểm:** Mô hình này không chỉ sử dụng đầu ra hoặc bản đồ đặc trưng mà còn khai thác mối quan hệ giữa các lớp hoặc giữa các mẫu dữ liệu trong mô hình giáo viên.
* **Ứng dụng:** Ví dụ, Ma trận Gram (Gram matrix) được sử dụng để tóm tắt mối quan hệ giữa các bản đồ đặc trưng của hai lớp. Một phương pháp là sử dụng phân tích giá trị kỳ dị (SVD) để trích xuất thông tin quan trọng từ các mối quan hệ này và giúp mô hình học sinh học tốt hơn.

**- Vai trò của từng loại kiến thức:**

* **Response-based Knowledge:** Phù hợp để dạy mô hình học sinh học các kết quả cuối cùng.
* **Feature-based Knowledge:** Tập trung vào việc học các đặc trưng sâu sắc và tinh tế hơn, thường được dùng trong các bài toán thị giác máy tính.
* **Relation-based Knowledge:** Tăng cường khả năng hiểu mối quan hệ phức tạp giữa các đặc trưng hoặc lớp, giúp mô hình học sinh có kiến thức toàn diện hơn.

Tóm lại, các loại kiến thức trong Knowledge Distillation giúp tối ưu hóa quá trình học của mô hình học sinh bằng cách truyền đạt không chỉ kết quả cuối mà còn cả thông tin trung gian và mối quan hệ dữ liệu. Điều này làm cho mô hình học sinh nhỏ gọn nhưng vẫn mạnh mẽ, đáp ứng được các bài toán phức tạp với tài nguyên hạn chế.

***2. Distillation Algorithms***

Nhiều thuật toán đã được đề xuất nhằm cải thiện quá trình truyền tải kiến thức từ mô hình teacher sang student, đặc biệt trong các trường hợp phức tạp. Các thuật toán phổ biến bao gồm:

**+ Adversarial distillation:**

Học đối kháng (Adversarial Learning) đã thành công trong mô hình tạo sinh (Generative Modeling), cụ thể là các mạng GAN (Generative Adversarial Networks). Trong Knowledge Distillation, GAN được sử dụng để tăng cường tập dữ liệu huấn luyện hoặc giúp các mô hình teacher và student học tốt hơn về phân phối dữ liệu. Ý tưởng chính là huấn luyện một mô hình sinh (generator) để tạo ra các mẫu dữ liệu tổng hợp, cố gắng làm cho chúng giống như có phân phối giống như dữ liệu thực, và một mô hình phân biệt (discriminator) để phân biệt giữa các mẫu dữ liệu thực và tổng hợp.

Ví dụ, một nghiên cứu áp dụng phương pháp Adversarial Knowledge Distillation vào nhận diện sự kiện trong ngôn ngữ tự nhiên. Mô hình teacher được huấn luyện dựa trên các chú thích chính xác, trong khi mô hình student bắt chước hành vi của teacher. Một mô hình phân biệt (discriminator) so sánh đầu ra của hai mô hình, còn student (vai trò như generator) cố gắng đánh lừa discriminator bằng các đầu ra tương tự teacher.

**+ Multi-Teacher distillation:**

Sử dụng một tập hợp các T có kiến trúc khác nhau có thể cung cấp cho S các loại kiến thức khác nhau mà có thể hữu ích hơn so với kiến thức thu được từ một T duy nhất.

Thông thường, các mô hình teacher nặng được kết hợp để tạo ra đầu ra “mềm” (soft labels) bằng cách trung bình hoặc trung bình có trọng số.

Ví dụ, một nghiên cứu sử dụng hai mô hình teacher theo hai chiến lược:

* Teacher thứ nhất truyền tải kiến thức dựa trên đặc trưng (Feature-based Knowledge) thông qua học đối kháng.
* Teacher thứ hai truyền kiến thức dựa trên phân phối xác suất (Response-based Knowledge).

Cách kết hợp này giúp mô hình student học hiệu quả hơn.

**+ Cross-modal distillation:**

T được huấn luyện trong một dạng và kiến thức của nó được rút trích vào S yêu cầu kiến thức từ một dạng khác.

Trong thực tế, dữ liệu thường tồn tại ở nhiều dạng (âm thanh, hình ảnh, video). Tuy nhiên, dữ liệu từ một số phương thức có thể bị thiếu hoặc không sử dụng được. Cross-Modal Knowledge Distillation giúp truyền tải kiến thức giữa các phương thức dữ liệu.

Ví dụ, một nghiên cứu sử dụng hai mô hình teacher:

* Một mô hình học từ âm thanh.
* Một mô hình học từ hình ảnh.

Mô hình student sau đó được huấn luyện với dữ liệu video để học cách hợp nhất các đặc điểm từ nhiều phương thức, thu hẹp khoảng cách giữa chúng.

**+ Graph-Based:**

Khi truyền tải kiến thức, chỉ học từng mẫu riêng lẻ có thể không đủ để mô hình student hiểu được mối quan hệ giữa các dữ liệu. Phương pháp dựa trên đồ thị sử dụng đồ thị để biểu diễn cấu trúc thông tin và truyền tải kiến thức từ teacher sang student.

Ví dụ, một phương pháp sử dụng đồ thị liên kết vùng (Affinity Graph) trong bài toán phân đoạn đường (road segmentation). Kiến thức về cấu trúc cảnh được biểu diễn bằng đồ thị, giúp mô hình student nắm bắt tốt hơn thông tin bối cảnh.

**+ Attention-Based:**

Attention là cơ chế giúp mô hình tập trung vào các vùng quan trọng trong dữ liệu. Trong Knowledge Distillation, các bản đồ attention từ các lớp trung gian của teacher được chuyển cho student.

Ví dụ, một nghiên cứu so khớp các bản đồ attention giữa hai mô hình bằng cách tính ma trận Jacobian, thậm chí cho các kiến trúc có kích thước không đồng nhất.

**+ Data-Free:**

Trong một số trường hợp, dữ liệu huấn luyện của teacher không thể sử dụng được do vấn đề bảo mật hoặc kích thước dữ liệu quá lớn. Data-Free Knowledge Distillation (Chưng cất kiến thức không dựa vào dữ liệu gốc) tạo ra dữ liệu tổng hợp mô phỏng phân phối của dữ liệu gốc.

Ví dụ, một phương pháp sử dụng DeepInversion, tối ưu hóa từ dữ liệu nhiễu để khớp các thống kê của đặc trưng sâu được lưu trữ trong các lớp batch-normalization của mô hình.

**+ Quantized:**

Trong các thiết bị biên (edge devices), mô hình cần được tối ưu để sử dụng ít tài nguyên hơn. Quantized Knowledge Distillation (Chưng chất kiến thức với lượng tử hóa) huấn luyện mô hình student với độ chính xác thấp (như 8-bit) từ teacher có độ chính xác cao (32-bit).

Ví dụ, một nghiên cứu sử dụng phương pháp lượng tử hóa trọng số để giảm độ phức tạp, đồng thời kết hợp tổn thất distillation trong quá trình huấn luyện.

**+ Lifelong:**

Tương tự như cách con người học nhanh các đối tượng mới nhờ kiến thức có sẵn, Lifelong Knowledge Distillation (Học tập suốt đời) kết hợp học tính năng, suy luận kiến thức, và huấn luyện bộ phân loại trong một hệ thống.

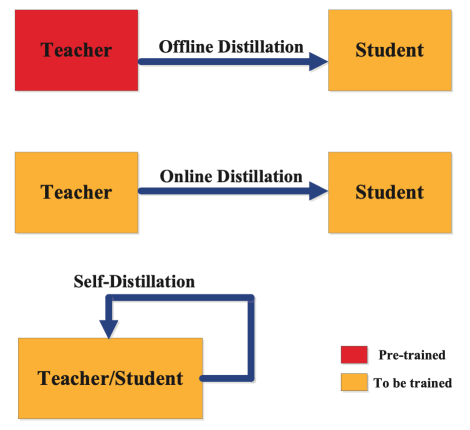
Ví dụ, một nghiên cứu áp dụng vào nhận diện hình ảnh với tập dữ liệu nhỏ (few-shot learning), sử dụng đồ thị tri thức (knowledge graphs) để mở rộng khả năng học tập từ dữ liệu trực quan.

***3. Teacher-Student Architecture:***

* Là một phiên bản nông hơn của mô hình giáo viên với ít lớp và ít nơ-ron hơn mỗi lớp.
* Là một phiên bản quantized của mô hình giáo viên.
* Là một mạng nhỏ hơn với các hoạt động cơ bản hiệu quả.
* Là các mạng nhỏ hơn với kiến trúc mạng toàn cầu được tối ưu hóa.
* Là cùng một mô hình với mô hình giáo viên.

### II. PHƯƠNG PHÁP HUẤN LUYỆN

Phương pháp huấn luyện mạng teacher-student trong Knowledge Distillation được phân loại dựa trên việc mô hình giáo viên có được cập nhật đồng thời với mô hình học sinh hay không. Vậy những phương pháp để huấn luyện S và T, bao gồm 3 Distillation Schemes chính:



Hình 6: Distillation Schemes của kĩ thuật Knowledge Distillation

**+ Offline distillation:** T được train trước đó rồi huấn luyện lại cho S.

* **Cách thức:** Trong phương pháp này, mô hình giáo viên được huấn luyện trước và sau đó được cố định ***(T không thay đổi)***. Trong quá trình huấn luyện, chỉ mô hình học sinh được cập nhật.
* **Đặc điểm:** Đây là phương pháp phổ biến nhất và cũng là cơ sở trong nghiên cứu ban đầu của Hinton et al. Phần lớn nghiên cứu tập trung vào cải thiện ***cơ chế truyền đạt kiến thức*** từ giáo viên sang học sinh, thay vì tối ưu kiến trúc của mô hình giáo viên.
* **Ví dụ:** Một nghiên cứu đã phát triển phương pháp ***truyền đạt kiến thức dựa trên xác suất***, sử dụng kiến thức dựa trên đặc trưng (feature-based). Thay vì so khớp trực tiếp các đặc trưng, phương pháp này so khớp phân phối xác suất của dữ liệu trong không gian đặc trưng, cho phép truyền đạt kiến thức giữa các dạng dữ liệu khác nhau (ví dụ: từ các công cụ trích xuất đặc trưng thủ công sang mạng nơ-ron).

**+ Online distillation:** T và S được cập nhật đồng thời trong một quá trình huấn luyện end-to-end. Sử dụng điện toán song song (parallel computing).

* **Cách thức:** Trong trường hợp không có mô hình giáo viên lớn được huấn luyện trước (điều mà phương pháp Offline giả định), mô hình giáo viên và học sinh được huấn luyện ***đồng thời.***
* **Đặc điểm:** Phương pháp này cho phép cả mô hình giáo viên và học sinh ***học hỏi lẫn nhau trong thời gian thực.*** Các mô hình con (sub-networks) thường được sử dụng làm mô hình giáo viên tạm thời, với cơ chế dạy lẫn nhau.
* **Ví dụ:** Một nghiên cứu gần đây đã đề xuất phương pháp ***Online Mutual Knowledge Distillation,*** trong đó các mô hình con được hợp nhất thông qua cơ chế kết hợp đặc trưng (feature fusion). Các mô hình con học hỏi lẫn nhau bằng cách sử dụng kiến thức dựa trên đầu ra (response-based).

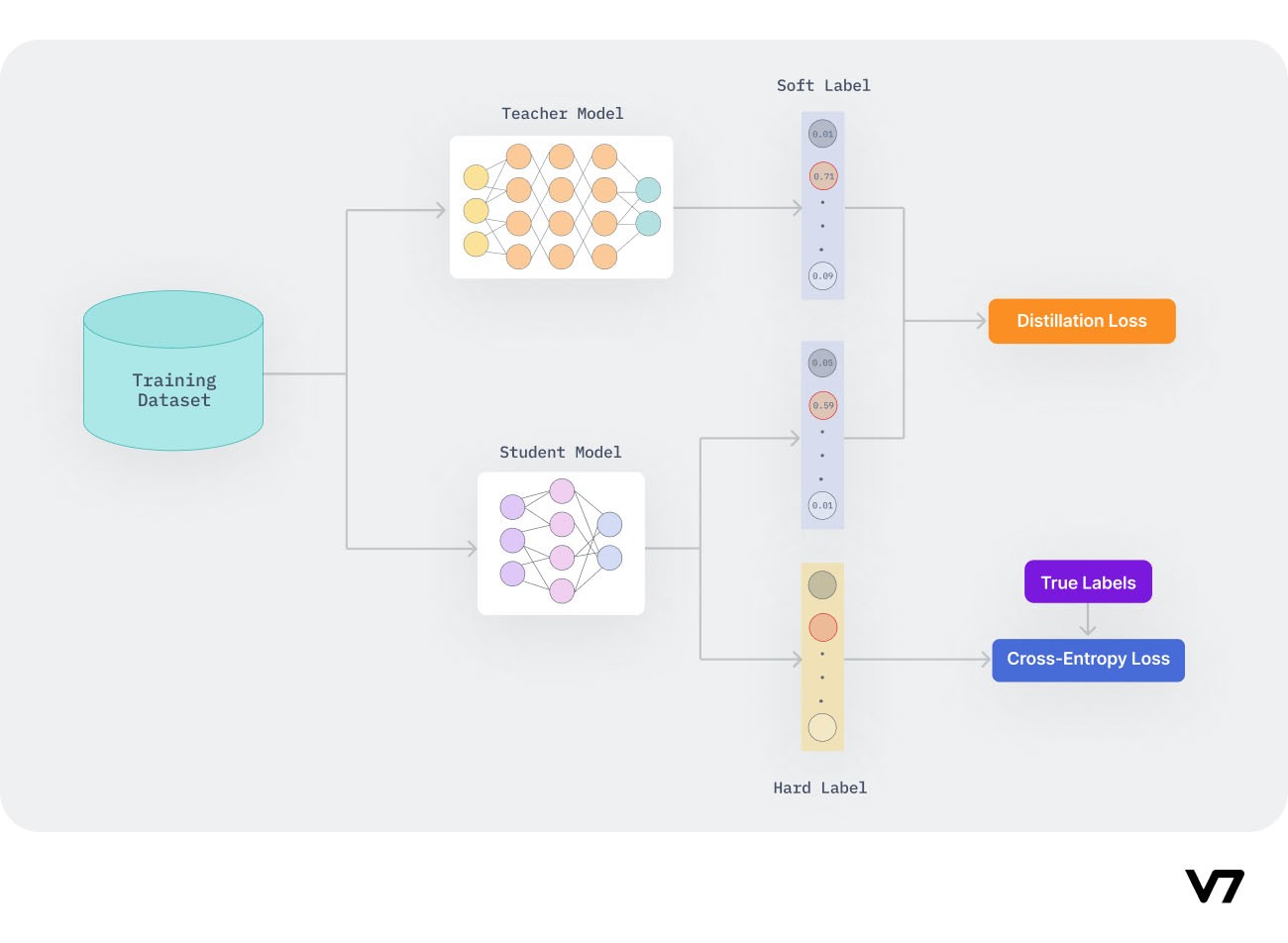
**+ Self-distillation:** Kiến ​​thức từ các lớp sâu hơn của DNN có thể được sử dụng để huấn luyện các lớp nông (chuyển kiến thức từ các epoch trước của S sang các epoch sau để huấn luyện S).

* **Cách thức:** Mô hình học sinh và giáo viên thực chất là ***cùng một mạng nơ-ron***, nghĩa là một mạng vừa đóng vai trò là giáo viên, vừa là học sinh.
* **Đặc điểm:** Giải quyết hai vấn đề lớn của chưng cất truyền thống:
* 1. Mô hình giáo viên với độ chính xác cao nhất không phải lúc nào cũng là giáo viên tốt nhất để chưng cất.
* 2. Mô hình học sinh thường không đạt được độ chính xác như mô hình giáo viên, gây suy giảm hiệu suất không chấp nhận được.
* **Cách thực hiện:** Gắn các bộ phân loại nông ***(shallow classifiers)*** tại các lớp trung gian của mạng:
* Trong giai đoạn huấn luyện, các bộ phân loại sâu hơn đóng vai trò làm giáo viên, hướng dẫn các bộ phân loại nông hơn bằng cách sử dụng hàm mất mát dựa trên độ chệch (divergence) và L2 Loss trên bản đồ đặc trưng.
* Trong giai đoạn suy luận (inference), các bộ phân loại nông sẽ được loại bỏ, chỉ giữ lại mạng chính.
* **Ví dụ:** Một phương pháp có tên Comprehensive Attention Self-Distillation ***(CASD)*** đã được đề xuất để giải quyết bài toán phát hiện đối tượng yếu giám sát (Weakly-Supervised Object Detection). Phương pháp này tận dụng chưng cất tự thân để ***củng cố tính giám sát không gian*** trên các đối tượng, giúp định vị bounding box chính xác hơn bằng cách khai thác thông tin đồng nhất qua nhiều lớp và phép biến đổi.

### III. CÁCH HOẠT ĐỘNG

Trong quy trình knowledge distillation, mô hình giáo viên (mô hình teacher) đã được huấn luyện trước tạo ra các nhãn mềm (soft labels) trên tập dữ liệu huấn luyện. Các nhãn mềm này, thực chất là các phân phối xác suất đầu ra, sau đó sẽ được sử dụng để huấn luyện mô hình học sinh (student model).

* **Nhãn cứng (Hard labels):** chỉ rõ một đối tượng là gì. Ví dụ, với một hình ảnh của cái bàn, đầu ra của mô hình giáo viên sẽ là “bàn”.
* **Nhãn mềm (Soft labels):** là các phân phối xác suất đầu ra, thể hiện độ tin cậy của mô hình đối với các nhãn. Ví dụ, với cùng một hình ảnh, mô hình giáo viên có thể đưa ra kết quả 90% là “bàn”, 8% là “bàn làm việc”, và 2% là “ghế”.



Hình 7: Cách hoạt động kĩ thuật Knowledge Distillation

Một ví dụ đơn giản về cách thức chưng cất kiến thức có thể được sử dụng để huấn luyện mô hình học sinh:

**+ Bước 1:** Huấn luyện mô hình giáo viên trên một tập dữ liệu. Sau đó, tạo nhãn mềm từ mô hình giáo viên cho cùng một tập dữ liệu.

**+ Bước 2:** Huấn luyện mô hình học sinh trên cùng một tập dữ liệu với các nhãn mềm. Sau đó, tinh chỉnh mô hình học sinh trên tập dữ liệu với nhãn cứng.

**+ Bước 3:** Đánh giá hiệu suất của mô hình học sinh thông qua một hàm mất mát (loss function), giúp định lượng mức độ mà dự đoán của mô hình phù hợp với kết quả mong muốn.

Nếu hiệu suất của mô hình học sinh đạt yêu cầu, mô hình có thể được triển khai. Nếu hiệu suất không đạt, mô hình học sinh có thể được huấn luyện lại với dữ liệu bổ sung hoặc tối ưu hóa thông qua điều chỉnh các siêu tham số, tốc độ học (learning rate), và/hoặc nhiệt độ chưng cất (distillation temperature).

Một nhiệt độ cao làm cho các phân phối xác suất mềm hơn (ít đỉnh hơn), trong khi nhiệt độ thấp làm cho các phân phối xác suất sắc nét hơn (có đỉnh rõ rệt và gần với nhãn cứng).

### IV. KĨ THUẬT TỐI ƯU HÓA

Một số kỹ thuật tối ưu hóa trong quá trình knowledge distillation bao gồm:

***1. Attention Transfer***

Attention Transfer là một kỹ thuật trong đó mô hình học sinh được huấn luyện để mô phỏng các bản đồ chú ý (attention maps) do mô hình giáo viên tạo ra. Các bản đồ chú ý này giúp làm nổi bật những khu vực quan trọng trong một hình ảnh hoặc chuỗi từ ngữ. Mục tiêu là mô hình học sinh phải nắm bắt được những vùng quan trọng mà mô hình giáo viên chú ý đến, giúp cải thiện khả năng học của mô hình học sinh.

***2. FitNets***

FitNets là một kỹ thuật trong đó mô hình học sinh được huấn luyện để khớp với các biểu diễn trung gian (intermediate representations) của mô hình giáo viên. Các biểu diễn trung gian này là những lớp ẩn của mô hình, chứa đựng các đặc trưng cơ bản của dữ liệu đầu vào. Việc sao chép các biểu diễn này giúp mô hình học sinh học được những đặc trưng quan trọng từ mô hình giáo viên mà không cần phải học lại từ đầu.

***3. Similarity-Based Distillation***

Trong kỹ thuật này, mô hình học sinh được huấn luyện để sao chép ma trận tương đồng (similarity matrix) của mô hình giáo viên. Ma trận tương đồng đo lường mức độ tương đồng giữa các mẫu đầu vào, giúp mô hình học sinh hiểu được mối quan hệ giữa các đối tượng trong dữ liệu một cách tốt hơn, từ đó cải thiện hiệu quả học tập.

***4. Hint-Based Distillation***

Chưng cất dựa trên gợi ý là một kỹ thuật trong đó mô hình học sinh được huấn luyện để dự đoán sự khác biệt giữa đầu ra của mô hình giáo viên và mô hình học sinh. Sự khác biệt này được gọi là “gợi ý” (hint). Bằng cách học từ các gợi ý này, mô hình học sinh có thể điều chỉnh và tối ưu hóa khả năng dự đoán của mình.

***5. Cross-Entropy Distillation***

Trong kỹ thuật này, mô hình học sinh được huấn luyện bằng một hàm mất mát kết hợp giữa hàm mất mát phân loại chuẩn (classification loss) và một hàm mất mát chưng cất (distillation loss). Hàm mất mát chưng cất đo lường sự khác biệt giữa các xác suất đầu ra của mô hình giáo viên và mô hình học sinh. Việc sử dụng hàm mất mát này giúp mô hình học sinh học cách dự đoán tương tự như mô hình giáo viên trong khi vẫn duy trì khả năng phân loại chính xác.

### V. TRƯỜNG HỢP SỬ DỤNG

Knowledge distillation là một kỹ thuật quan trọng giúp tạo ra các mô hình học máy nhẹ, có thể chạy hiệu quả trên các thiết bị có hạn chế về tài nguyên tính toán. Các mô hình được chưng cất này rất hữu ích đối với các hệ thống đề xuất và các thiết bị IoT tại rìa (edge devices), nơi mà khả năng tính toán có thể bị hạn chế.

Bằng cách sử dụng chưng cất kiến thức, các thiết bị như camera an ninh, hệ thống nhà thông minh và trợ lý ảo kỹ thuật số có thể thực hiện nhiều nhiệm vụ phức tạp ngay tại chỗ, bao gồm:

* **Phân loại (Classification):** Gán dữ liệu đầu vào vào một trong các danh mục đã được định nghĩa trước. Ví dụ, trong một hệ thống nhận diện hình ảnh, phân loại các hình ảnh vào các nhóm như động vật, thực vật, phương tiện giao thông,…
* **Xử lý ngôn ngữ tự nhiên (NLP):** Xử lý và phân tích lượng lớn dữ liệu ngôn ngữ tự nhiên cho các nhiệm vụ như phân tích cảm xúc (sentiment analysis) và nhận dạng thực thể có tên (named entity recognition). Chẳng hạn, trong các trợ lý ảo, NLP giúp hiểu được yêu cầu của người dùng và phản hồi chính xác.
* **Phát hiện đối tượng (Object Detection):** Nhận diện và phân loại các đối tượng trong hình ảnh hoặc video. Điều này rất hữu ích trong các ứng dụng như giám sát an ninh, nơi các camera cần phát hiện và phân loại các đối tượng như người, xe, hoặc động vật.
* **Nhận dạng giọng nói (Speech Recognition):** Chuyển đổi ngôn ngữ nói thành văn bản. Ví dụ, trợ lý ảo như Siri hay Alexa sử dụng kỹ thuật này để hiểu các lệnh bằng giọng nói và thực hiện các hành động phù hợp.
* **Dịch máy (Machine Translation):** Dịch văn bản hoặc giọng nói từ một ngôn ngữ này sang ngôn ngữ khác. Ví dụ, các ứng dụng dịch thuật như Google Translate sử dụng chưng cất kiến thức để triển khai mô hình dịch máy mạnh mẽ trên các thiết bị di động.

Nhờ vào việc sử dụng kĩ thuật knowledge distillation, các thiết bị này có thể thực hiện các tác vụ phức tạp ngay trên chính thiết bị mà không cần phải phụ thuộc vào máy chủ hoặc kết nối Internet liên tục. Điều này giúp giảm độ trễ và tiết kiệm băng thông, đồng thời cải thiện hiệu suất và tính bảo mật.

### VI. ƯU NHƯỢC ĐIỂM

***1. Ưu điểm:***

Một trong những ưu điểm chính của chưng cất kiến thức là khả năng ***tạo ra các mô hình nhỏ gọn và nhanh chóng, có hiệu suất tốt*** ngay cả trên các thiết bị IoT tại rìa (edge devices) với tài nguyên tính toán hạn chế. Các mô hình này có thể thực hiện các nhiệm vụ phức tạp mà không cần đến các máy chủ mạnh mẽ, giúp tiết kiệm băng thông và giảm độ trễ. Điều này đặc biệt hữu ích trong các ứng dụng như camera an ninh, hệ thống nhà thông minh, và trợ lý ảo, nơi mà yêu cầu về hiệu suất và tốc độ cao nhưng không có đủ tài nguyên tính toán.

***2. Nhược điểm:***

Một trong những vấn đề lớn là cần phải chấp nhận một sự đánh đổi giữa kích thước mô hình và độ chính xác. Mặc dù mô hình sau khi chưng cất thường nhỏ hơn và nhanh hơn, nhưng đôi khi độ chính xác có thể không đạt được mức cao như mô hình gốc (mô hình giáo viên). Điều này có nghĩa là trong một số trường hợp, nếu yêu cầu về độ chính xác rất cao, việc sử dụng mô hình chưng cất có thể không phải là sự lựa chọn tối ưu.

***3. Kết luận:***

Tóm lại, knowledge distillation giúp tạo ra các mô hình hiệu quả về tài nguyên, nhưng cần phải cân nhắc kỹ lưỡng giữa việc giảm kích thước mô hình và duy trì độ chính xác đủ tốt cho các ứng dụng thực tế.

### VII. TẦM QUAN TRỌNG

Khi phát triển các ứng dụng AI cho khách hàng (B2C), một thách thức lớn là các thiết bị edge như điện thoại di động hoặc máy tính bảng thường bị hạn chế về khả năng lưu trữ và xử lý. Điều này buộc các kỹ sư học máy phải tìm cách làm cho các mô hình AI phù hợp với khả năng của thiết bị, và phương pháp hiệu quả nhất là sử dụng các kỹ thuật nén để giảm kích thước mô hình. Vậy, một số kỹ thuật nén quan trọng được sử dụng trong quá trình này như:

* **Pruning (Cắt tỉa):** Đây là quá trình loại bỏ những nơ-ron nhân tạo hoặc trọng số (weights) ít quan trọng nhất trong mạng nơ-ron, tức là những thành phần ít đóng góp đến hiệu suất của mô hình. Sau khi cắt tỉa, mô hình trở nên nhỏ gọn hơn mà không làm giảm đáng kể độ chính xác.
* **Quantization (Lượng tử hóa):** Kỹ thuật này giảm độ chính xác của các trọng số trong mô hình. Ví dụ, thay vì biểu diễn trọng số với 32 bit, mô hình chỉ sử dụng 16 hoặc 8 bit. Điều này không chỉ làm giảm kích thước mô hình mà còn tăng tốc độ xử lý, đặc biệt trên các phần cứng hỗ trợ tính toán với độ chính xác thấp.
* **Low-rank factorization (Phân tích nhân tử hạng thấp):** Trong phương pháp này, các ma trận trọng số trong mô hình được xấp xỉ bằng các ma trận có hạng thấp hơn. Điều này giúp giảm số lượng tham số cần thiết, mà vẫn giữ lại được phần lớn thông tin quan trọng trong mô hình.
* **Knowledge distillation (Chưng cất kiến thức):** Đây là kỹ thuật huấn luyện một mô hình nhỏ gọn hơn (mô hình học sinh) để bắt chước hành vi của một mô hình lớn hơn (mô hình giáo viên). Mô hình học sinh được thiết kế để duy trì hiệu suất tương đương hoặc gần tương đương với mô hình giáo viên, nhưng với yêu cầu tài nguyên ít hơn.

**Vậy, tại sao nén mô hình quan trọng?**

Các kỹ thuật nén như trên, đặc biệt với kĩ thuật Knowledge distillation đóng vai trò quan trọng trong việc đưa AI vào các thiết bị edge. Chúng giúp các mô hình lớn, phức tạp có thể hoạt động trên những thiết bị hạn chế về phần cứng, mà vẫn đảm bảo hiệu suất đủ tốt. Điều này mở rộng khả năng áp dụng AI vào các ứng dụng thực tế, như nhận diện khuôn mặt trên điện thoại hoặc trợ lý ảo trong nhà thông minh.

### VIII. THUẬT NGỮ LIÊN QUAN

***1. Knowledge Distillation và Transfer Learning***

Knowledge distillation và transfer learning đều là những khái niệm quan trọng trong học máy, nhưng chúng có mục tiêu và cách ứng dụng hoàn toàn khác nhau:

**+ Knowledge Distillation:**

* **Mục tiêu:** Tạo ra một mô hình nhỏ gọn (mô hình học sinh) có thể giải quyết cùng một bài toán như mô hình lớn hơn (mô hình giáo viên).
* **Ứng dụng:** Tập trung vào việc giảm kích thước mô hình để phù hợp với các thiết bị có giới hạn tài nguyên mà vẫn giữ được hiệu suất cao.

**+ Transfer Learning:**

* **Mục tiêu:** Giảm thời gian huấn luyện mô hình lớn cho một bài toán mới bằng cách tận dụng kiến thức đã học từ một bài toán khác.
* **Ứng dụng:** Phù hợp cho việc mở rộng mô hình AI để giải quyết các tác vụ mới mà không cần xây dựng lại mô hình từ đầu.

Sự khác biệt chính:

* **Knowledge distillation** giúp giảm kích thước mô hình nhưng vẫn giữ nguyên bài toán ban đầu.
* **Transfer learning** tái sử dụng kiến thức cũ để giải quyết bài toán mới, nhằm tối ưu hóa thời gian huấn luyện.

***2. Knowledge Distillation và Data Distillation***

Knowledge distillation và data distillation đều là các phương pháp nén, nhưng chúng nhắm đến những đối tượng khác nhau:

**+ Knowledge Distillation:**

* **Mục tiêu:** Nén mô hình học máy bằng cách giảm số lượng tham số mà không làm mất đi hiệu quả hoặc độ chính xác.
* **Ứng dụng:** Giảm kích thước mô hình để sử dụng hiệu quả trên các thiết bị có giới hạn về phần cứng.

**+ Data Distillation:**

* **Mục tiêu:** Nén dữ liệu bằng cách tìm ra một tập dữ liệu nhỏ hơn nhưng vẫn đại diện đầy đủ cho toàn bộ tập dữ liệu gốc.
* **Ứng dụng:** Tối ưu hóa dữ liệu đầu vào để giảm tài nguyên cần thiết trong quá trình huấn luyện.

Sự khác biệt chính:

* **Data distillation** tập trung vào việc tối ưu hóa tập dữ liệu.
* **Knowledge distillation** tập trung vào việc giảm kích thước mô hình mà không làm giảm hiệu suất.

Nhờ đó, cả hai phương pháp đều đóng góp vào việc làm cho AI gọn nhẹ và hiệu quả hơn, nhưng với các mục tiêu khác nhau.

## B. ADVERSARIAL ATTACKS

### I. KHÁI NIỆM

Học máy đối kháng (Adversarial Machine Learning) là một phương pháp học máy nhằm đánh lừa các mô hình học máy bằng cách cung cấp đầu vào (input) lừa đảo, thường gọi là mẫu đối kháng (adversarial example) vào những “khe hở hay lỗ hổng” trong các mô hình học máy hiện tại. Điều này đặc biệt nguy hiểm trong những ứng dụng như nhận diện hình ảnh hoặc phân loại spam,… khiến cho bộ phân loại của các hệ thống ứng dụng đó đưa ra dự đoán không chính xác. Hiểu đơn giản, tấn công đối kháng (Adversarial Attacks) là một phương pháp tạo ra các mẫu đối kháng (adversarial example).

Adversarial attacks thường được phân thành hai loại dựa vào mức độ hiểu biết của kẻ tấn công với hệ thống:

* **White-Box Attacks:** Kẻ tấn công biết rõ kiến trúc, tham số, và quy trình của mô hình mục tiêu. Tận dụng toàn bộ thông tin này để thiết kế các mẫu đối kháng tinh vi nhất.
* **Black-Box Attacks:** Kẻ tấn công không có kiến thức về mô hình bên trong. Họ chỉ dựa vào dữ liệu đầu vào và kết quả đầu ra để xây dựng chiến lược tấn công.

Một mẫu đối kháng là một dữ liệu đầu vào của các mô hình học máy được tác động có chủ ý hay cố tình để làm một mô hình AI đánh giá sai và đưa ra kết quả không chính xác. Một mẫu đối kháng là một phiên bản bị lỗi của một đầu vào hợp lệ, trong đó lỗi được thực hiện bằng cách thêm một nhiễu loạn có cường độ nhỏ. Sự phiền toái ít được chú ý này được thiết kế để đánh lừa bộ phân loại bằng cách tối đa hóa xác suất của một lớp không chính xác. Mẫu đối kháng này được thiết kế để có vẻ “bình thường hay hợp lệ” đối với con người nhưng lại gây ra sự phân loại sai theo mô hình học máy mục tiêu.

Một số phương pháp phổ biến dùng để tạo ra các mẫu đối kháng:

**1. Limited-memory BFGS (L-BFGS)**

L-BFGS là một thuật toán tối ưu hóa phi tuyến tính dựa trên gradient, được sử dụng để giảm thiểu số lượng nhiễu cần thêm vào hình ảnh.

* **Ưu điểm:** Hiệu quả trong việc tạo ra các ví dụ đối kháng.
* **Nhược điểm:** Rất tốn tài nguyên tính toán và không thực tế khi triển khai do thời gian xử lý lâu.

**2. Fast Gradient Sign Method (FGSM)**

FGSM là phương pháp dựa trên gradient đơn giản, nhanh chóng. Tập trung vào việc thêm nhiễu tối thiểu vào từng điểm ảnh để khiến mô hình phân loại sai.

* **Ưu điểm:** Thời gian tính toán nhanh hơn so với các phương pháp khác.
* **Nhược điểm:** Nhiễu được thêm vào tất cả các đặc trưng, có thể làm tăng mức độ nhiễu không cần thiết.

**3. Jacobian-based Saliency Map Attack (JSMA)**

Khác với FGSM, JSMA chỉ chọn một số đặc trưng quan trọng để thêm nhiễu, nhằm giảm thiểu số lượng đặc trưng bị thay đổi nhưng vẫn đạt được mục tiêu làm sai lệch mô hình.

* **Ưu điểm:** Ít đặc trưng bị thay đổi hơn so với FGSM.
* **Nhược điểm:** Yêu cầu tài nguyên tính toán nhiều hơn FGSM.

**4. Deepfool**

Deepfool là một kỹ thuật không nhắm mục tiêu (untargeted) nhằm tạo ra các mẫu đối kháng với khoảng cách tối thiểu so với mẫu gốc. Phương pháp này ước tính ranh giới quyết định giữa các lớp và thêm nhiễu theo cách lặp lại.

* **Ưu điểm:** Hiệu quả trong việc tạo ví dụ đối kháng với ít nhiễu hơn và tỷ lệ phân loại sai cao.
* **Nhược điểm:** Tốn tài nguyên tính toán nhiều hơn FGSM và JSMA. Kết quả có thể không tối ưu.

**5. Carlini & Wagner Attack (C&W)**

C&W là một phương pháp cải tiến dựa trên L-BFGS, nhưng không sử dụng các ràng buộc hộp (box constraints) và có các hàm mục tiêu khác. Phương pháp này rất hiệu quả trong việc vượt qua các cơ chế phòng thủ tiên tiến như *distillation* hoặc *adversarial training*.

* **Ưu điểm:** Rất hiệu quả trong việc tạo ra ví dụ đối kháng. Có thể vượt qua một số biện pháp phòng thủ đối kháng.
* **Nhược điểm:** Tốn nhiều tài nguyên tính toán hơn FGSM, JSMA, và Deepfool.

**6. Generative Adversarial Networks (GAN)**

GAN được sử dụng để tạo các mẫu đối kháng bằng cách để hai mạng neural cạnh tranh với nhau: một mạng tạo ra các mẫu (generator) và một mạng phân biệt (discriminator) cố gắng phát hiện các mẫu giả.

* **Ưu điểm:** Có thể tạo ra các mẫu khác biệt với dữ liệu huấn luyện ban đầu.
* **Nhược điểm:** Huấn luyện GAN rất tốn kém về mặt tài nguyên và có thể không ổn định.

**7. Zeroth-order Optimization Attack (ZOO)**

ZOO cho phép ước tính gradient của các bộ phân loại mà không cần truy cập trực tiếp, rất hữu ích cho các cuộc tấn công “hộp đen” (black-box attacks). Phương pháp này sử dụng gradient và hessian được ước tính thông qua các truy vấn mô hình.

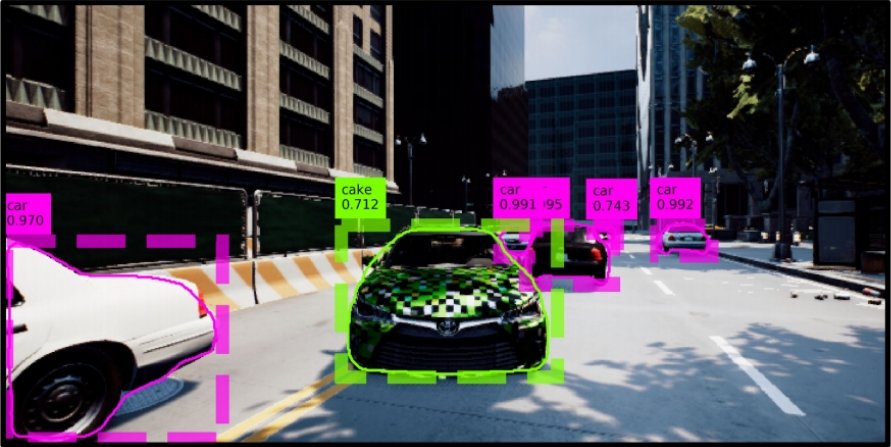
* **Ưu điểm:** Hiệu suất tương đương với C&W, không cần huấn luyện mô hình thay thế hay thông tin trực tiếp về mô hình.
* **Nhược điểm:** Yêu cầu số lượng lớn truy vấn đến mô hình mục tiêu.

### II. MỐI ĐE DỌA

Khi học máy trở thành một phần cốt lõi trong hoạt động của nhiều tổ chức, nhu cầu bảo vệ các hệ thống học máy cũng tăng lên đáng kể. Đây chính là lý do tại sao lĩnh vực học máy đối kháng (Adversarial Machine Learning) đang ngày càng được quan tâm mạnh mẽ.

Các tập đoàn công nghệ lớn như Google, Microsoft hay IBM đã bắt đầu đầu tư mạnh mẽ vào việc tăng cường bảo mật cho hệ thống học máy. Một số công ty sử dụng học máy trong các ứng dụng quan trọng như Google, Amazon, Microsoft và Tesla đã từng phải đối mặt với các cuộc tấn công đối kháng, gây ra những hậu quả nghiêm trọng.

Một ví dụ điển hình là một hệ thống học máy đã nhận diện nhầm một chiếc xe hơi có hoa văn ngụy trang thành một đối tượng hoàn toàn khác là một cái bánh (như hình bên dưới), điều này minh chứng rõ ràng rằng chỉ cần một thay đổi nhỏ trong dữ liệu đầu vào cũng có thể đánh lừa hệ thống.



Hình 8: Mối đe dọa của học máy đối kháng

Ngoài doanh nghiệp, các chính phủ cũng dần quan tâm đến vấn đề này. Liên minh Châu Âu (European Union - EU) đã phát triển một danh sách đánh giá tiêu chuẩn để kiểm tra mức độ đáng tin cậy của các hệ thống trí tuệ nhân tạo, gọi là ALTAI (Assessment List for Trustworthy Artificial Intelligence). Điều này chứng minh rằng việc bảo mật cho học máy không còn chỉ là vấn đề của doanh nghiệp mà còn là yêu cầu cấp thiết ở cấp độ pháp lý và chính sách quốc gia.

Một điều quan trọng khác là mặc dù bảo mật cho trí tuệ nhân tạo đã được quan tâm, nhưng hiện nay các tổ chức vẫn chủ yếu tập trung vào bảo mật truyền thống. Họ chưa có đủ kiến thức chuyên sâu hoặc chiến lược để đảm bảo an toàn cho hệ thống học máy khi đưa vào thực tế sản xuất.

Chính sự phát triển nhanh chóng của AI trong sản xuất đã làm nổi bật lên nhu cầu cấp thiết về Học máy bảo vệ quyền riêng tư (Privacy-Preserving Machine Learning – PPML), một hướng tiếp cận không chỉ bảo vệ dữ liệu mà còn giảm thiểu nguy cơ bị tấn công đối kháng.

Quay trở lại với thời đại công nghệ 4.0, ML đang mở ra một bề mặt tấn công (attack surface) mới khi được áp dụng trong nhiều lĩnh vực như bảo mật, y tế, tài chính, và sản xuất,... Tuy nhiên, việc tích hợp công nghệ này cũng mang lại những rủi ro bảo mật đáng kể liên quan đến thao túng dữ liệu và khai thác mô hình.



Hình 9: Ứng dụng của ML trong nhiều lĩnh vực

Các xu hướng và chiến lược chính cho tương lai của học máy đối kháng:

**1. Nhận diện rủi ro bảo mật trong ML**

Các hệ thống học máy dễ bị tấn công trước những mối đe dọa như:

* **Thao túng dữ liệu (Data Manipulation):** Tấn công vào dữ liệu huấn luyện hoặc triển khai, làm suy giảm độ chính xác của mô hình.
* **Mẫu đối kháng (Adversarial Samples):** Gây ra sai lệch trong quyết định của mô hình bằng cách thay đổi nhỏ dữ liệu đầu vào.
* **Đánh cắp mô hình (Model Theft):** Kẻ tấn công có thể trích xuất hoặc sao chép một mô hình bằng cách khai thác API.

**2. Chiến lược phòng thủ trong tương lai**

Để giảm thiểu rủi ro từ học máy đối kháng, các tổ chức cần chuẩn bị những chiến lược mạnh mẽ:

* **Tăng cường bảo mật dữ liệu:** Xây dựng các biện pháp kiểm soát quyền truy cập và phát hiện dữ liệu bị thao túng.
* **Phòng thủ mô hình (Model Robustness):** Phát triển các phương pháp làm tăng khả năng chống chịu của mô hình trước các tấn công, như *adversarial training*.
* **Giám sát và phát hiện tấn công:** Sử dụng các công cụ và hệ thống phát hiện các mẫu đầu vào bất thường hoặc nghi vấn.

**3. Ứng dụng ML an toàn**

Các ngành công nghiệp sử dụng ML cần có kế hoạch bảo mật toàn diện để bảo vệ tài sản trí tuệ và dữ liệu nhạy cảm. Điều này bao gồm:

* **Đánh giá bảo mật:** Thường xuyên kiểm tra các hệ thống ML để phát hiện lỗ hổng bảo mật.
* **Cải tiến phòng thủ:** Liên tục cập nhật các thuật toán để chống lại các kỹ thuật tấn công mới.
* **Chia sẻ thông tin:** Cộng tác giữa các tổ chức để chia sẻ các bài học và phương pháp đối phó với tấn công.

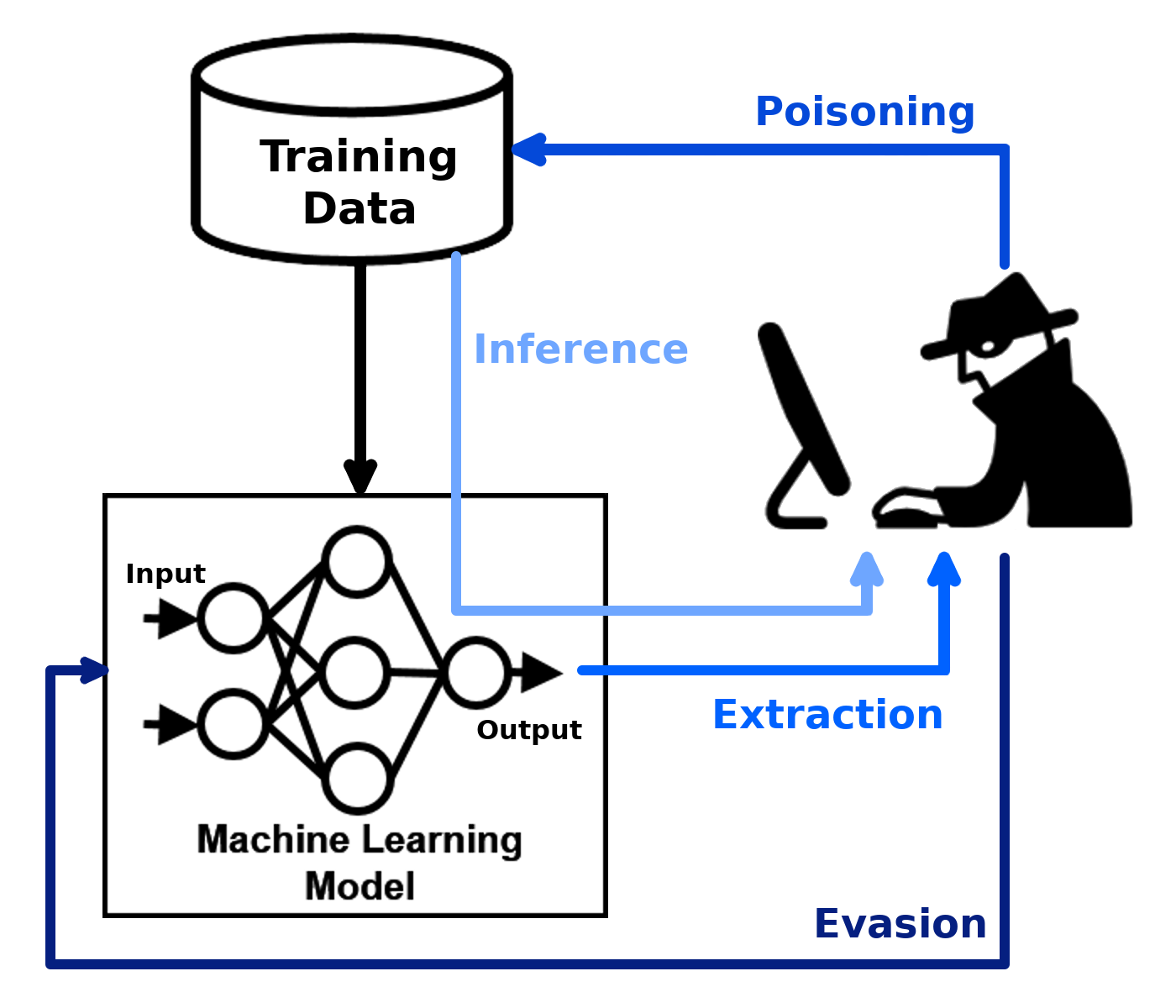
**4. Nhìn về phía trước**

Học máy đối kháng không chỉ là một thách thức mà còn là cơ hội để cải thiện hệ thống. Bằng cách hiểu rõ rủi ro và phát triển chiến lược phòng thủ, các tổ chức có thể giảm thiểu thiệt hại tiềm ẩn và đảm bảo rằng việc áp dụng ML mang lại giá trị lâu dài, an toàn và đáng tin cậy.

***Kết luận:*** *Sự phát triển của học máy đối kháng đòi hỏi các tổ chức phải chuẩn bị kỹ càng và tích cực hành động để bảo vệ hệ thống khỏi những mối đe dọa tiềm ẩn.*

### III. CÁCH HOẠT ĐỘNG

Cách thức hoạt động của các cuộc tấn công đối kháng (Adversarial Attacks) vào hệ thống học máy rất đa dạng, có thể áp dụng trên cả các mô hình học sâu (Deep Learning) và các mô hình học máy truyền thống như SVM (Support Vector Machines) hay hồi quy tuyến tính (Linear Regression).



Hình 10: Cách hoạt động của tấn công đối kháng

Mục tiêu chung của những cuộc tấn công này là làm giảm hiệu quả của mô hình, khiến mô hình đưa ra kết quả sai lệch trong các nhiệm vụ cụ thể. Nói cách khác là “đánh lừa” thuật toán học máy. Đây cũng chính là lĩnh vực nghiên cứu của học máy đối kháng, nơi tập trung vào các kỹ thuật nhằm khai thác điểm yếu của các hệ thống AI.

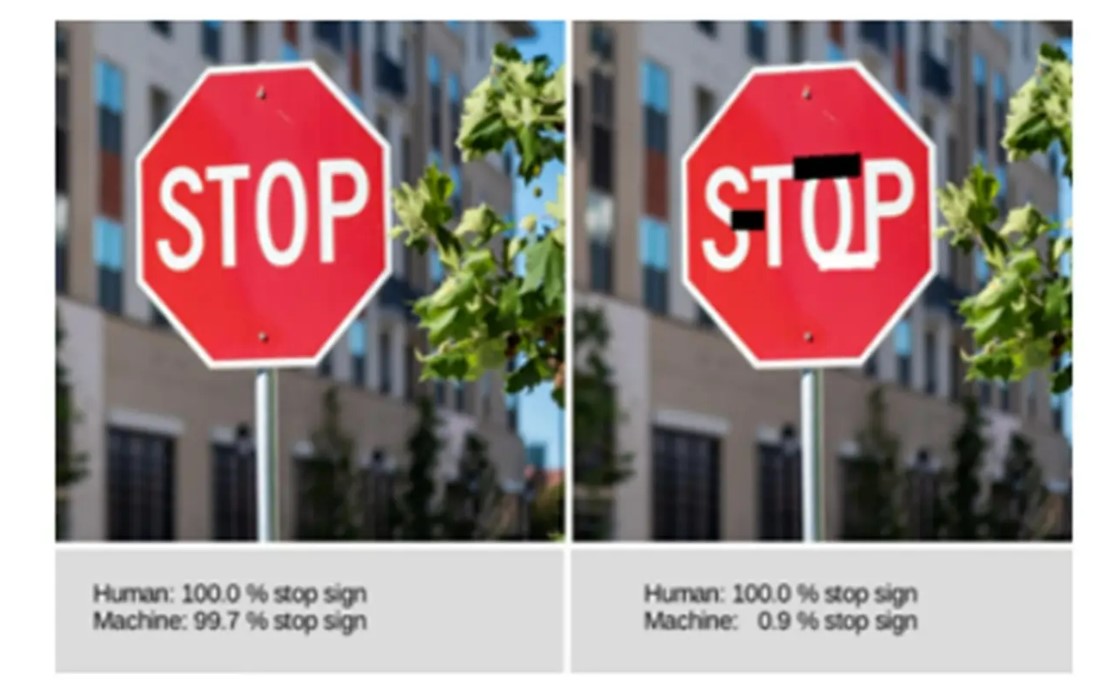
Các cuộc tấn công này thường được chia thành bốn loại chính:



Hình 11: Các loại chính của tấn công đối kháng

**1. Tấn công trốn tránh (Evasion Attacks)**

Đây là loại tấn công phổ biến và được nghiên cứu nhiều nhất. Kẻ tấn công *cố ý thay đổi dữ liệu đầu vào trong giai đoạn triển khai* để đánh lừa các mô hình đã được huấn luyện.



Hình 12: Minh họa tấn công Evasion

Đặc điểm của loại tấn công này là không làm thay đổi dữ liệu huấn luyện (training data) ban đầu mà chỉ tập trung vào dữ liệu triển khai (deployment data - dữ liệu được sử dụng để đưa vào hệ thống sau khi mô hình đã được huấn luyện và triển khai để hoạt động thực tế). Chẳng hạn, trong các hệ thống phát hiện malware hoặc spam, kẻ tấn công có thể “ngụy trang” nội dung của malware hoặc email spam để chúng được hệ thống phân loại nhầm là hợp lệ.

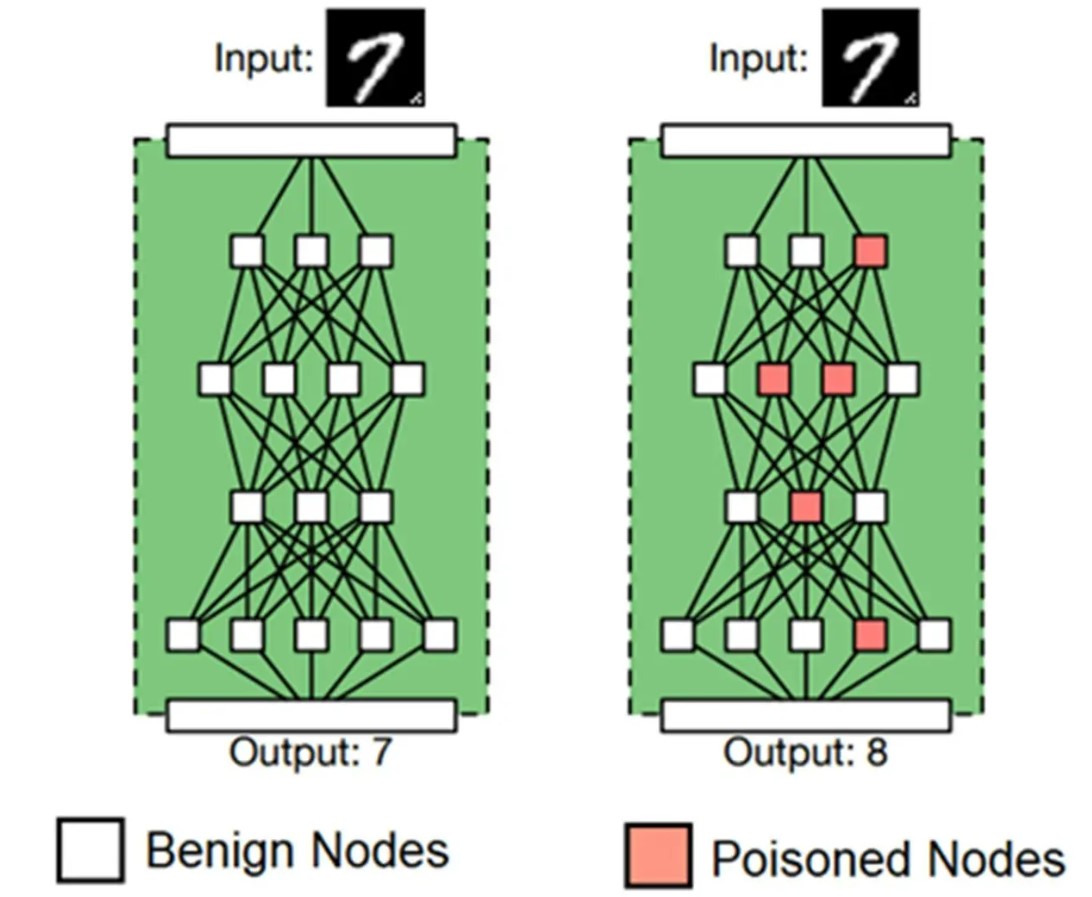
Ví dụ như các cuộc tấn công nhắm vào hệ thống xác thực sinh trắc học, nơi kẻ tấn công có thể làm giả hoặc thay đổi đặc điểm khuôn mặt để vượt qua hệ thống kiểm tra.

**2. Tấn công làm nhiễm độc dữ liệu (Poisoning Attacks)**

Đây là một hình thức mà kẻ tấn công *cố tình làm thay đổi dữ liệu huấn luyện hoặc nhãn dữ liệu* để gây sai lệch cho mô hình khi triển khai.

Hai loại tấn công nhiễm độc chính bao gồm:

* **Label Poisoning (Nhiễm độc nhãn):** Kẻ tấn công sửa đổi nhãn của một số mẫu trong tập huấn luyện, khiến mô hình học các mối quan hệ sai giữa dữ liệu đầu vào và đầu ra.
* **Data Injection (Tiêm nhiễm dữ liệu):** Kẻ tấn công thêm các mẫu dữ liệu độc hại vào tập huấn luyện. Các mẫu này có thể giống với các dữ liệu hiện có hoặc hoàn toàn khác biệt, nhằm thay đổi ranh giới quyết định của mô hình.



Hình 13: Minh họa tấn công Poisoning

Ví dụ, trong các hệ thống học máy liên tục tái huấn luyện dựa trên dữ liệu thu thập trong quá trình hoạt động, kẻ tấn công có thể chèn các mẫu dữ liệu độc hại vào luồng dữ liệu đầu vào. Khi đó, mô hình sẽ học những thông tin sai lệch và đưa ra kết quả không chính xác trong tương lai.

Điều này đặc biệt nguy hiểm trong các hệ thống tự động, nơi dữ liệu được thu thập và huấn luyện mà không có sự kiểm tra nghiêm ngặt. Loại tấn công này làm rõ rằng chỉ cần một lượng nhỏ dữ liệu bị “nhiễm độc” cũng có thể khiến mô hình hoạt động sai lệch.

**3. Tấn công suy luận (Inference) hay đảo ngược (Inversion)**

Đây là một hình thức mà kẻ tấn công *tái tạo lại dữ liệu nhạy cảm từ bộ huấn luyện* của mô hình học máy mục tiêu để thiết kế ngược lại cho mô hình học máy cho kẻ tấn công.

Bên dưới là một hình ảnh được khôi phục bằng cách sử dụng phương pháp tấn công đảo ngược hay suy luận thành một mô hình mới (bên trái) và một hình ảnh trong tập huấn luyện của mô hình gốc của nạn nhân (bên phải).



Hình 14: Minh họa tấn công Inference

Các ví dụ:

* **Model Inversion (Đảo ngược mô hình):** Kẻ tấn công tái tạo một phiên bản gần giống với dữ liệu gốc trong tập huấn luyện bằng cách khai thác các xác suất đầu ra của mô hình.
* **Membership Inference (Suy luận thành viên):** Kẻ tấn công xác định liệu một dữ liệu cụ thể có nằm trong tập huấn luyện hay không thông qua phân tích đầu ra xác suất của mô hình.

**4. Tấn công trích xuất mô hình (Model Extraction)**

Loại tấn công này đặc biệt nguy hiểm với các mô hình và dữ liệu nhạy cảm. Kẻ tấn công sử dụng kỹ thuật “điều tra hộp đen” (probing black-box) để *truy vấn mô hình và dần tái tạo lại cấu trúc hoặc dữ liệu huấn luyện ban đầu*, loại tấn công này còn gọi là “ăn cắp” mô hình.



Hình 15: Minh họa tấn công Extraction

Ví dụ, một kẻ tấn công có thể trích xuất mô hình dự đoán thị trường chứng khoán, mô hình này chứa đựng thông tin có giá trị cao. Kẻ tấn công sau đó có thể sử dụng mô hình bị đánh cắp cho mục đích tài chính cá nhân.

**Kết luận:** tấn công đối kháng không chỉ đơn giản là làm hỏng mô hình mà còn gây ra những hậu quả nghiêm trọng như mất mát dữ liệu, làm giảm độ tin cậy của hệ thống, và trong nhiều trường hợp có thể dẫn đến tổn thất kinh tế hoặc tổn hại uy tín. Vì vậy, nghiên cứu cách phòng tránh và giảm thiểu các rủi ro từ tấn công đối kháng là một bước quan trọng trong việc bảo mật hệ thống AI.

### IV. CÁCH PHÒNG VỆ

Cách phòng vệ, phòng thủ hay phòng tránh đối với các loại tấn công có thể sử dụng nhiều chiến lược khác nhau, đối với:

**1. Tấn công né tránh (Evasion Attacks)**

* **Adversarial Training:** Huấn luyện mô hình với các ví dụ đối kháng để tăng khả năng chống chịu.
* **Gradient Masking:** Làm mượt biên quyết định của mô hình để giảm hiệu quả của các tấn công dựa trên gradient. Lưu ý rằng phương pháp này bị đội ngũ OpenAI chỉ trích do tính không ổn định.
* **Defensive Distillation:** Sử dụng một mô hình phụ để huấn luyện với xác suất “mềm” từ mô hình ban đầu, giúp tăng khả năng chống lại tấn công đối kháng.

**2. Tấn công làm nhiễm độc dữ liệu (Poisoning Attacks)**

* **Data Sanitization:** Lọc bỏ các mẫu dữ liệu khả nghi hoặc không nhất quán trong tập huấn luyện.
* **Robust Optimization:** Áp dụng các kỹ thuật tối ưu hóa bền vững trong quá trình huấn luyện để giảm thiểu tác động của tấn công nhiễm độc.
* **Outlier Detection:** Nhận diện và loại bỏ các ngoại lai trong dữ liệu huấn luyện để giảm ảnh hưởng của tấn công nhiễm độc đến ranh giới quyết định của mô hình.

**3. Tấn công suy luận (Inference) hay đảo ngược (Inversion)**

* **Differential Privacy:** Áp dụng các kỹ thuật bảo mật vi sai trong quá trình huấn luyện mô hình bằng cách thêm nhiễu có kiểm soát vào dữ liệu, hạn chế thông tin có thể suy luận về các mẫu riêng lẻ.
* **Homomorphic Encryption:** Cho phép mô hình được huấn luyện và thực hiện dự đoán trên dữ liệu đã mã hóa, bảo vệ quyền riêng tư của các mẫu trong tập huấn luyện.
* **Secure Multi-Party Computation (SMPC):** Kỹ thuật này cho phép nhiều bên cùng huấn luyện một mô hình trên dữ liệu kết hợp mà không cần tiết lộ dữ liệu riêng của từng bên, giúp bảo vệ khỏi các tấn công đảo ngược.

**4. Tấn công trích xuất mô hình (Model Extraction)**

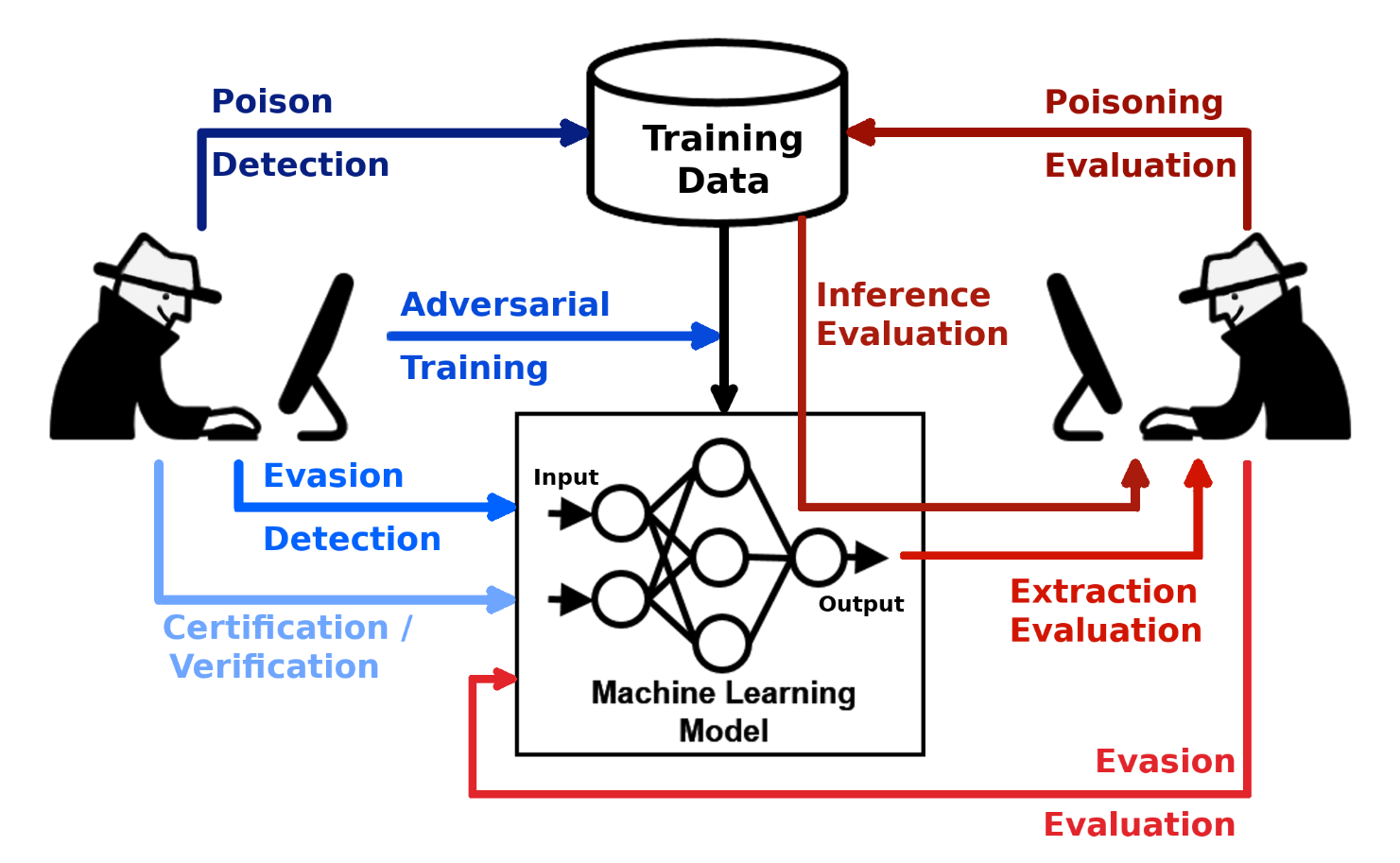
* **API Rate Limiting:** Hạn chế số lượng truy vấn mà API chấp nhận từ một người dùng cụ thể trong một khoảng thời gian nhất định, giảm hiệu quả của các tấn công trích xuất mô hình.
* **Output Perturbation:** Thêm nhiễu vào kết quả đầu ra của mô hình, khiến kẻ tấn công khó có thể sao chép chính xác mô hình mục tiêu.
* **Watermarking:** Chèn một chữ ký hoặc dấu bản quyền duy nhất vào mô hình, giúp phát hiện và truy vết các mô hình bị đánh cắp.

### V. CÔNG CỤ HỖ TRỢ

Sử dụng thư viện Adversarial Robustness Toolbox (ART) làm công cụ chính trong quá trình thực hiện các cuộc tấn công đối kháng lên các mô hình học máy mục tiêu.

Adversarial Robustness Toolbox (ART) là một thư viện Python dành cho Bảo mật máy học (Machine Learning Security). ART được tổ chức bởi Linux Foundation AI & Data Foundation (LF AI & Data). ART cung cấp các công cụ cho phép các nhà phát triển và nhà nghiên cứu bảo vệ và đánh giá các mô hình và ứng dụng Machine Learning trước các mối đe dọa bất lợi như Trốn tránh (Evasion), Nhiễm độc (Poisoning), Suy luận (Inference) và Trích xuất (Extraction). ART hỗ trợ tất cả các framework học máy phổ biến như TensorFlow, Keras, PyTorch, MXNet, scikit-learn, XGBoost, LightGBM, CatBoost, Gpy,… cùng với tất cả các loại dữ liệu như hình ảnh, bảng, âm thanh, video,... và các nhiệm vụ học máy như phân loại, phát hiện đối tượng, nhận dạng giọng nói,…

Adversarial Robustness Toolbox (ART) được sử dụng cho cả 2 lĩnh vực chính trong an toàn thông tin là Red Teams và Blue Teams.

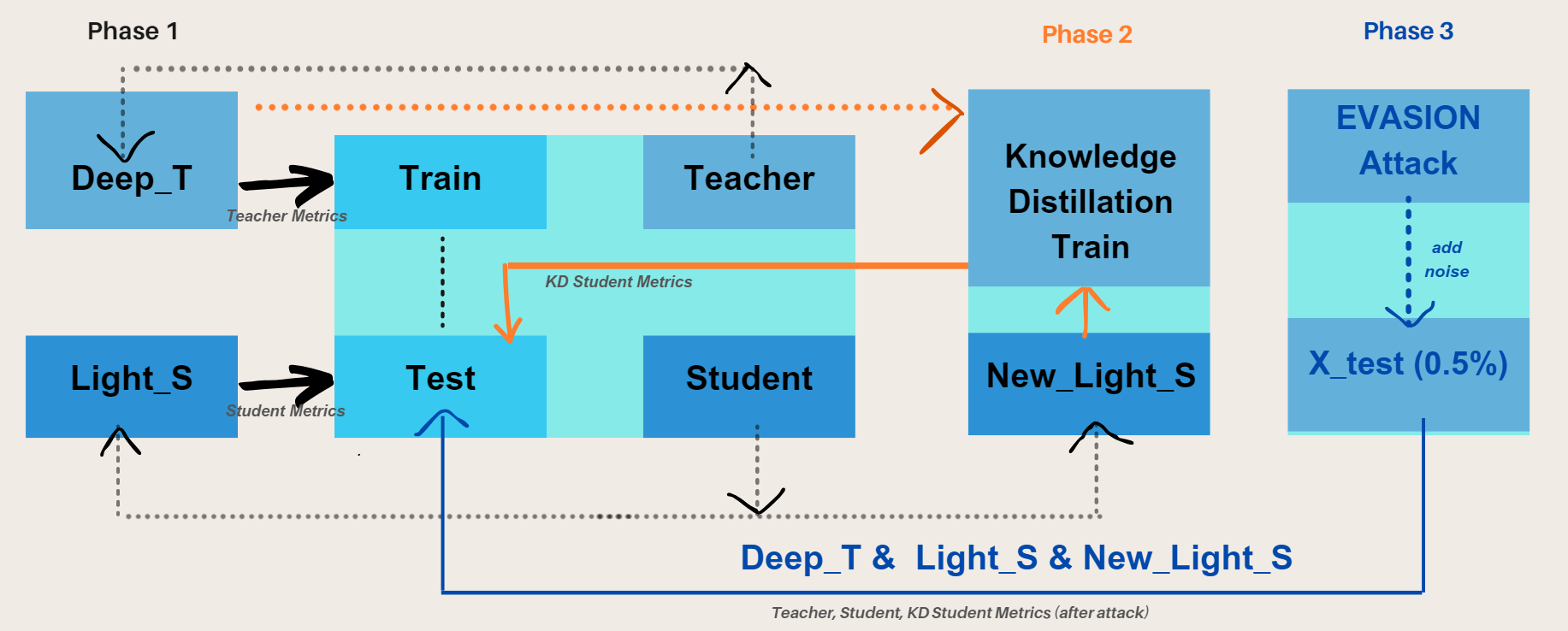


Hình 16: Red Teams và Blue Teams của ART

# CHƯƠNG II. TRIỂN KHAI THỰC NGHIỆM

*Nội dung chương này sẽ trình bày về* ***kiến trúc hệ thống, thực nghiệm và kết quả*** *của Knowledge Distillation và Adversarial Attacks trong lĩnh vực Machine Learning.*

## A. KIẾN TRÚC



Hình 17: Kiến trúc thực nghiệm

Thực hiện định nghĩa mô hình teacher và mô hình student, cũng như định nghĩa hàm train và hàm test để huấn luyện và đánh giá mô hình học máy:

Quá trình thực nghiệm diễn ra trong 3 giai đoạn chính:

**+ Giai đoạn 1:** Từ mô hình teacher đã được định nghĩa, khởi tạo một mô hình teacher (Deep\_T) để thực hiện quá trình train và test. Tương tự, từ mô hình student đã được định nghĩa, khởi tạo một mô hình student (Light\_S) để thực hiện quá trình train và test. Kết quả của giai đoạn 1 là kết quả sau khi huấn luyện mô hình Deep\_T và Light\_S.

**+ Giai đoạn 2:** Từ mô hình student đã được định nghĩa, khởi tạo một mô hình student mới (**New\_Light\_S**) để thực hiện quá trình **train với kĩ thuật Knowledge Distillation (KD)** và test. Kết quả của giai đoạn 2 là kết quả sau khi huấn luyện mô hình New\_Light\_S, **khác Light\_S** ở giai đoạn 1.

**+ Giai đoạn 3:** Thực hiện các cuộc tấn công đối kháng thuộc loại trốn tránh (Evasion) lên cả 3 mô hình: Deep\_T, Light\_S và New\_Light\_S với nhiễu loạn (mẫu đối kháng) bằng 0.5% dữ liệu gốc ban đầu và kết luận.

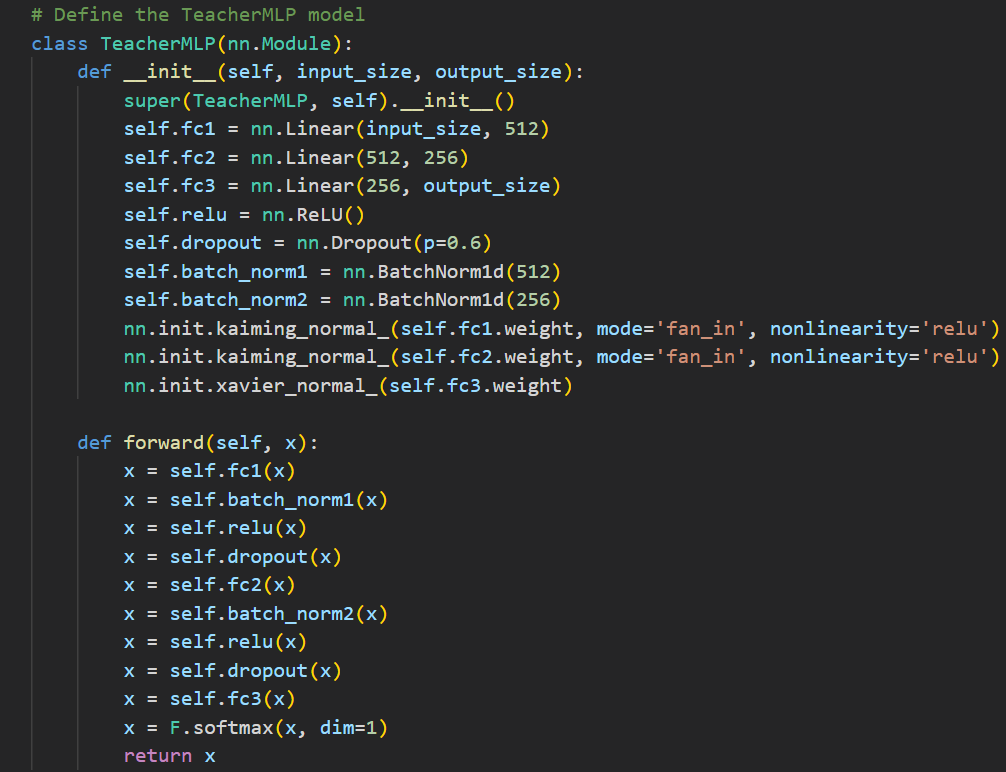
## B. THỰC NGHIỆM

*Tham khảo file code được đính kèm:* ***Evaluating the KD Technique against AAs.ipynb***

- Thực nghiệm với dataset: **iotid20.csv** (được đính kèm).

- Trong phạm vi đồ án này cả 2 mô hình teacher và student đều dựa vào kiến trúc MLP (Multi-layer Perceptron) và áp dụng kĩ thuật KD.

***+ TeacherMLP model:***



Hình 18: TeacherMLP model

**Tầng Fully Connected (Linear Layers):**

* **fc1:** Kết nối đầu vào (input\_size) với 512 neuron, là tầng đầu tiên trích xuất đặc trưng từ dữ liệu.
* **fc2:** Kết nối từ 512 neuron xuống 256 neuron, giảm kích thước để giữ lại các đặc trưng quan trọng.
* **fc3:** Là tầng đầu ra, kết nối từ 256 neuron đến số lớp phân loại (output\_size).

**Tầng kích hoạt (Activation Function):**

* **ReLU (Rectified Linear Unit):** Được áp dụng sau mỗi tầng fully connected, giúp mô hình học được các đặc trưng phi tuyến tính.
* **Dropout Layer:** Dropout(p=0.6) để loại bỏ ngẫu nhiên 60% kết nối của neuron trong mỗi bước huấn luyện để giảm overfitting. Nhằm mục đích tăng cường khả năng tổng quát (generalization) của mô hình trên dữ liệu chưa nhìn thấy (unseen data).
* **Batch Normalization:** BatchNorm1d được áp dụng sau tầng fc1 và fc2 để chuẩn hóa đầu ra của mỗi tầng, đảm bảo độ ổn định của mô hình trong quá trình huấn luyện và cải thiện tốc độ hội tụ.

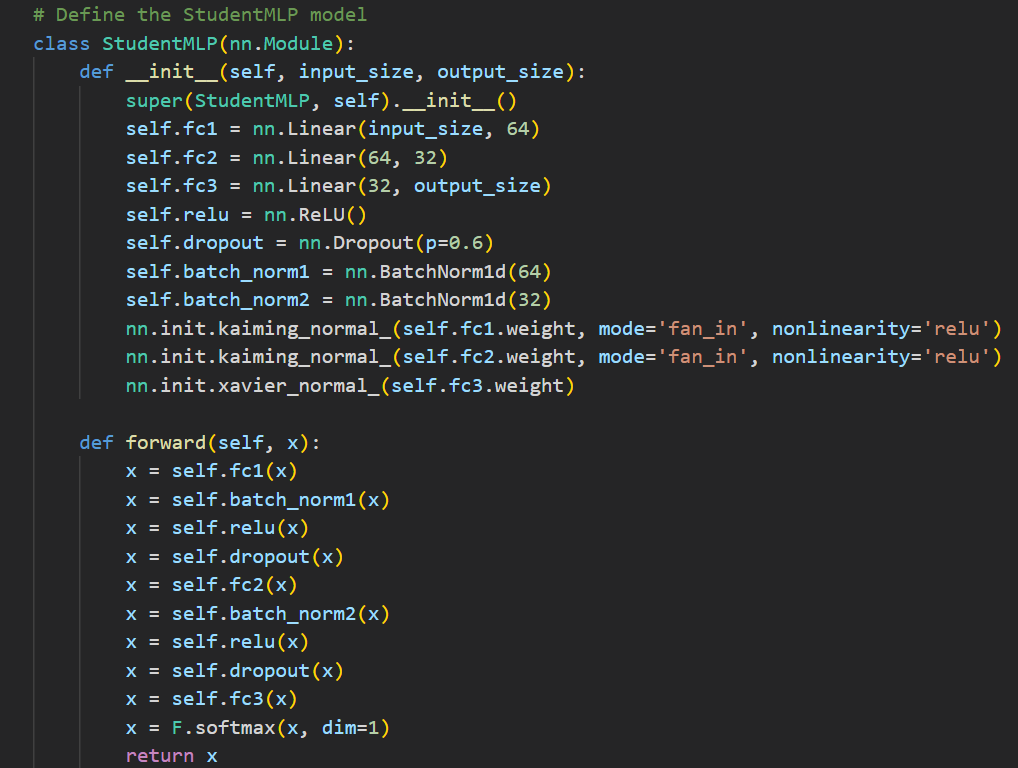
**Khởi tạo trọng số:**

* **Kaiming Normal Initialization:** Áp dụng cho fc1 và fc2. Phương pháp này phù hợp với hàm kích hoạt ReLU, giúp đảm bảo phân phối giá trị đầu ra phù hợp.
* **Xavier Initialization:** Áp dụng cho fc3. Xavier đảm bảo rằng giá trị đầu ra không bị quá lớn hoặc quá nhỏ, giúp mô hình hội tụ nhanh.

**Lan truyền tiến (Forward Pass):** Dữ liệu được xử lý qua các bước sau:

* **Tầng fc1:** Dữ liệu đầu vào được đưa qua tầng fully connected fc1. Sau đó được chuẩn hóa bởi BatchNorm1d(512), kích hoạt bởi ReLU, và giảm overfitting nhờ Dropout(p=0.6).
* **Tầng fc2:** Dữ liệu qua fc2 được chuẩn hóa, kích hoạt, và giảm overfitting tương tự tầng fc1.
* **Tầng fc3:** Tầng cuối cùng tính toán đầu ra với số lượng lớp tương ứng với output\_size, và **Softmax** để biến đổi đầu ra thành xác suất trên các lớp.

***+ StudentMLP model:***



Hình 19: StudentMLP model

**Tầng Fully Connected (Linear Layers):**

* **fc1:** Kết nối đầu vào (input\_size) với 64 neuron. Tầng này chịu trách nhiệm trích xuất đặc trưng từ dữ liệu đầu vào.
* **fc2:** Từ 64 neuron, tầng này giảm xuống 32 neuron, giúp cô đọng các đặc trưng quan trọng.
* **fc3:** Tầng đầu ra kết nối 32 neuron tới số lớp phân loại (output\_size).

**Tầng kích hoạt (Activation Function):**

* **ReLU (Rectified Linear Unit):** Hàm kích hoạt phi tuyến tính, giúp mô hình học được các đặc trưng phức tạp.
* **Dropout Layer:** Dropout(p=0.6) để loại bỏ ngẫu nhiên 60% kết nối của neuron trong mỗi bước huấn luyện, giảm nguy cơ overfitting.
* **Batch Normalization:** BatchNorm1d được áp dụng sau các tầng fc1 và fc2 để chuẩn hóa đầu ra, cải thiện sự ổn định trong huấn luyện và tốc độ hội tụ.

**Khởi tạo trọng số:**

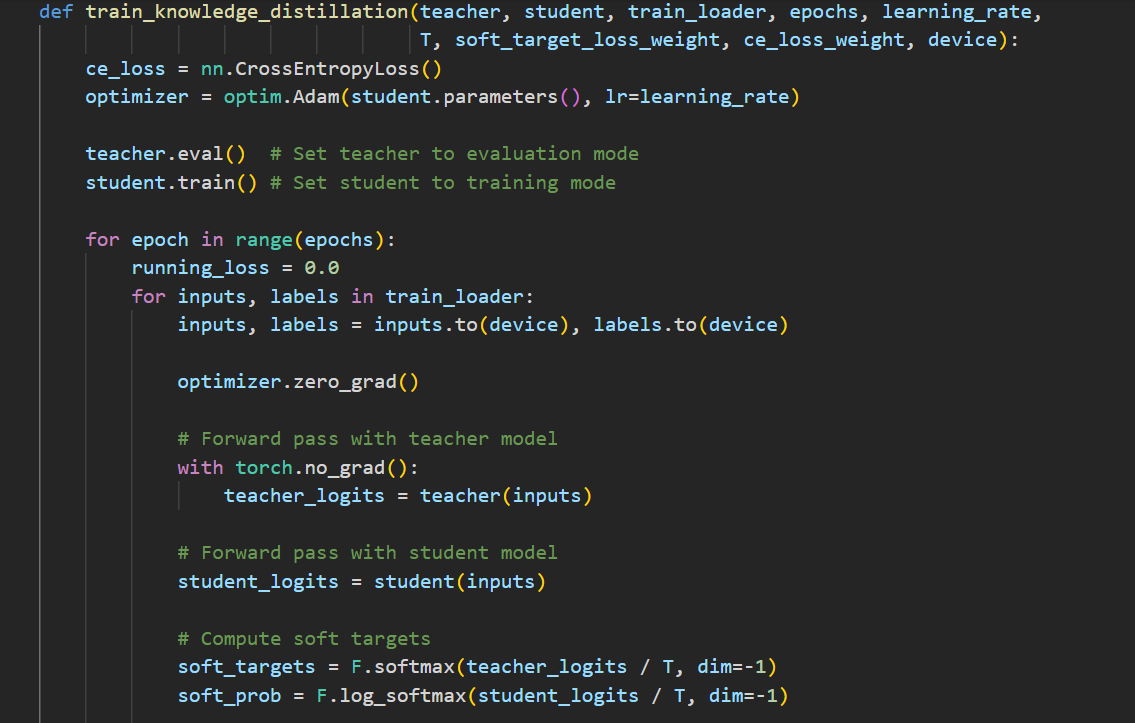
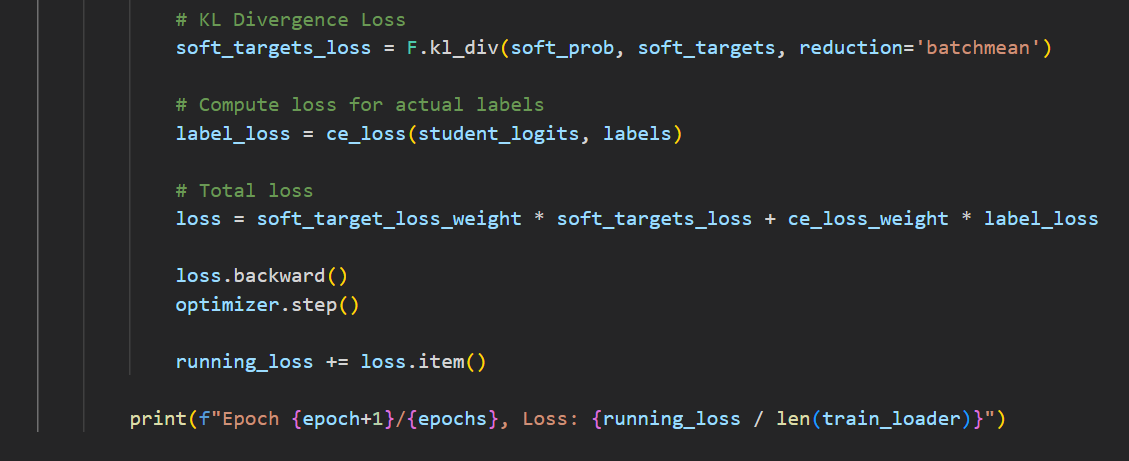
* **Kaiming Normal Initialization:** Áp dụng cho fc1 và fc2, phù hợp với hàm kích hoạt ReLU, đảm bảo các giá trị trọng số được khởi tạo tối ưu cho việc học.
* **Xavier Initialization:** Áp dụng cho fc3, giúp cân bằng giá trị đầu ra và tăng tốc độ hội tụ.

**Lan truyền tiến (Forward Pass):** Dữ liệu được xử lý qua các bước sau

* **Tầng fc1:** Dữ liệu qua tầng fully connected đầu tiên. Sau đó được chuẩn hóa bằng BatchNorm1d(64), kích hoạt bằng ReLU, giảm overfitting nhờ Dropout(p=0.6).
* **Tầng fc2:** Đầu ra từ tầng trước được đưa qua fc2, áp dụng chuẩn hóa, kích hoạt, và Dropout tương tự.
* **Tầng fc3:** Tầng cuối cùng tính toán đầu ra với số lớp tương ứng với output\_size và Softmax để chuyển đổi đầu ra thành xác suất trên các lớp.

***+ Knowledge Distillation (KD):***



Hình 20: Knowledge Distillation (KD)

**Các tham số chính của KD:**

* **T (Temperature):** Tham số điều chỉnh mức độ mềm hóa các xác suất đầu ra của mô hình teacher (soft targets). Nhiệt độ cao làm phân phối xác suất mượt hơn, cung cấp thêm thông tin cho student.
* **soft\_target\_loss\_weight:** Trọng số của loss dựa trên soft targets (thường là KL Divergence Loss).
* **ce\_loss\_weight:** Trọng số của loss dựa trên hard targets (CrossEntropy Loss).
* **soft\_targets:** Được tính bằng cách áp dụng softmax với tham số nhiệt độ T lên logits của teacher model.
* **label\_loss:** Được tính bằng CrossEntropy Loss giữa logits của student model và nhãn thực tế (hard targets).

**Quy trình KD trong đoạn code:**

*1. Khởi tạo*

* Mô hình teacher được đặt ở chế độ eval (đánh giá, không cập nhật trọng số).
* Mô hình student được đặt ở chế độ train (huấn luyện).
* Bộ tối ưu hóa Adam được sử dụng để cập nhật trọng số của student.

*2. Huấn luyện Knowledge Distillation*

Với mỗi epoch, quá trình huấn luyện diễn ra theo các bước:

**+ Bước 1: Đầu vào và dự đoán:**

* Teacher model tính toán logits (đầu ra chưa được chuẩn hóa).
* Student model cũng tính toán logits tương tự.

**+ Bước 2: Tính toán soft targets:** Áp dụng hàm softmax với nhiệt độ T để chuyển đổi logits của teacher thành soft targets.

**+ Bước 3: Tính toán loss:**

* **Soft Targets Loss:** Đo độ khác biệt giữa soft targets của teacher và student bằng KL Divergence Loss.
* **Hard Targets Loss:** So sánh logits của student với nhãn thực tế (hard labels) bằng CrossEntropy Loss.
* **Tổng Loss:** Tích hợp cả hai loại loss với trọng số tương ứng (soft\_target\_loss\_weight và ce\_loss\_weight).

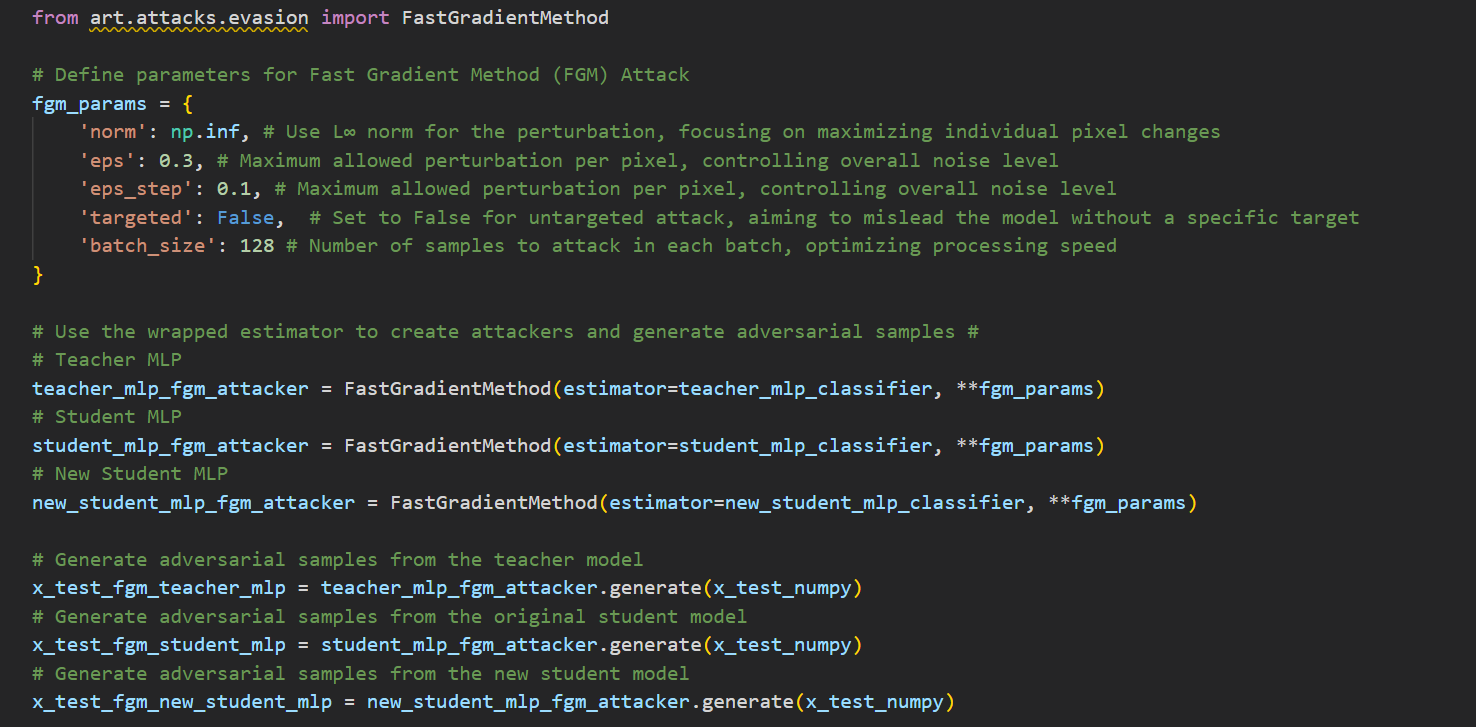
**+ Bước 4: Cập nhật mô hình:** Tính toán gradient và cập nhật trọng số của student model dựa trên tổng loss.

***+ Adversatial attacks***

- Thực hiện 5 cuộc tấn công đối kháng thuộc loại tấn công né tránh (Evasion Attacks), đều nhằm mục tiêu tạo ra mẫu đầu vào giả mạo (adversarial examples) khiến mô hình đưa ra dự đoán sai, bao gồm:

**+ Fast Gradient Method (FGM) Attack:**

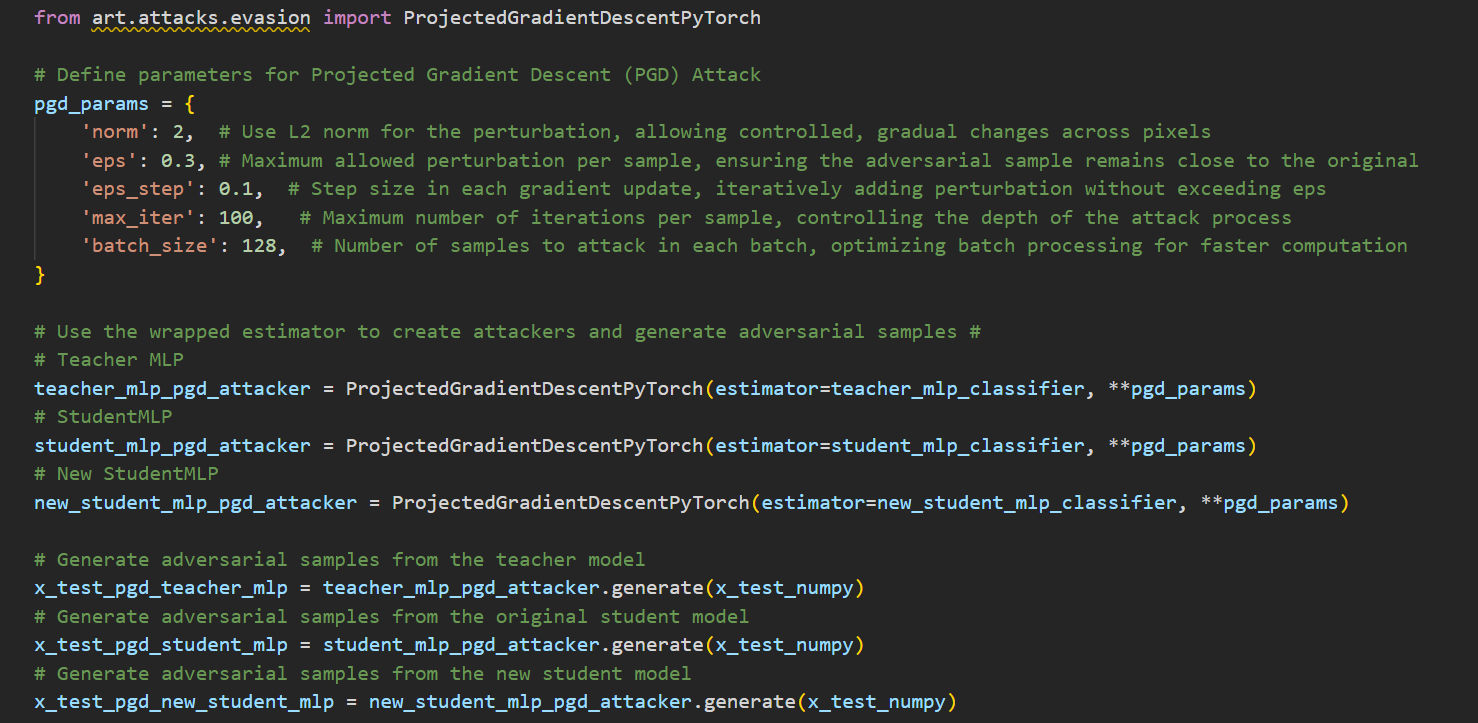
* **Cách thức hoạt động:** FGM tạo mẫu đối kháng bằng cách áp dụng một bước gradient duy nhất. Loại tấn công tối ưu hoá bằng cách di chuyển theo hướng của gradient của hàm mất mát với đầu vào ban đầu, thêm một lượng nhiễu theo hướng gradient để tối đa hóa lỗi của mô hình.
* **Tốc độ:** Rất nhanh do chỉ thực hiện một bước gradient.
* **Ưu điểm:** Thực hiện đơn giản, tốc độ cao, phù hợp để thử nghiệm và đánh giá nhanh khả năng chống chịu của mô hình.
* **Nhược điểm:** Hiệu quả có thể không cao bằng các phương pháp phức tạp hơn.



Hình 21: Fast Gradient Method (FGM) Attack:

**+ Projected Gradient Descent (PGD) Attack:**

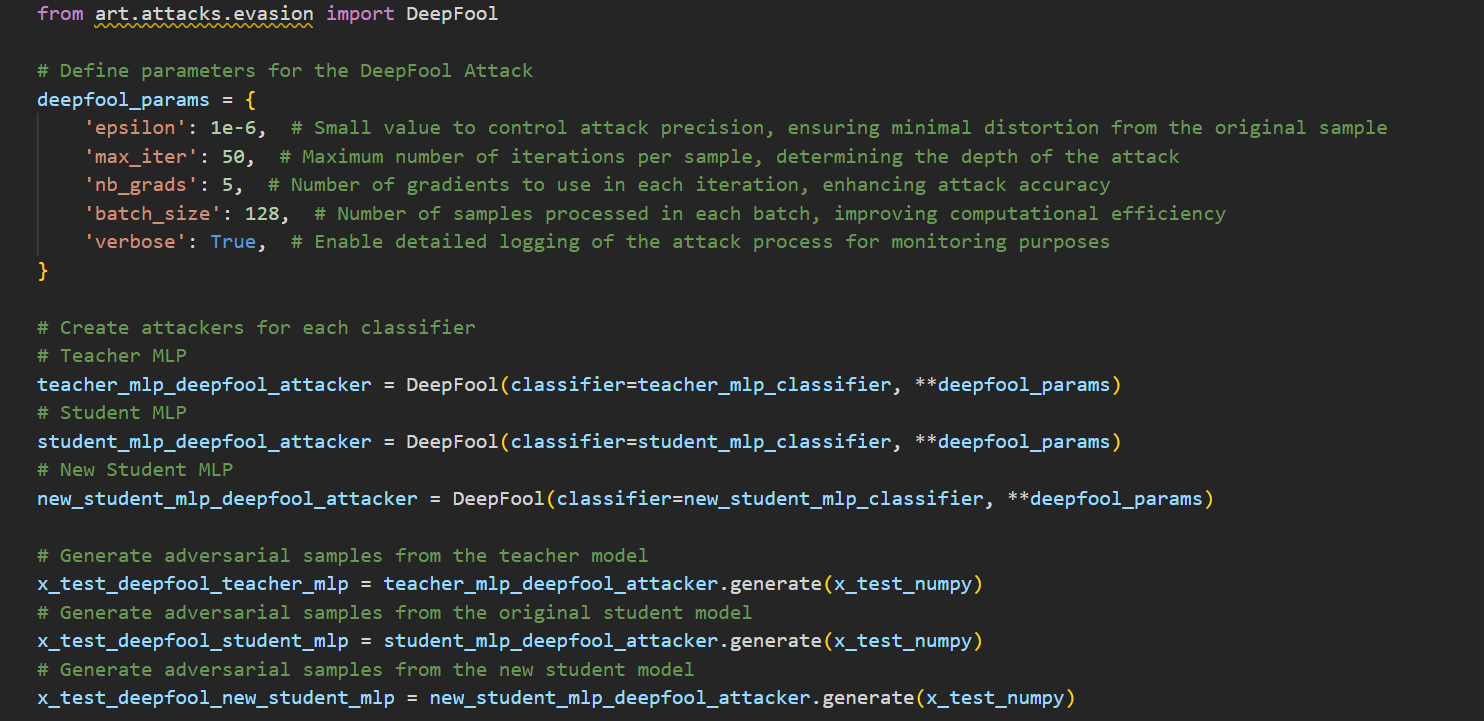
* **Cách thức hoạt động:** PGD là một biến thể của FGM với nhiều bước gradient nhỏ, cho phép tấn công được tối ưu hóa qua nhiều vòng lặp. Sau mỗi bước, loại này sẽ chiếu (project) kết quả vào một miền giới hạn để đảm bảo nhiễu thêm vào không vượt quá giới hạn.
* **Tốc độ:** Chậm hơn FGM do thực hiện nhiều bước gradient.
* **Ưu điểm:** Hiệu quả cao và rất mạnh, đặc biệt là khi đối mặt với các biện pháp phòng vệ. PGD được coi là một trong những phương pháp tấn công mạnh mẽ nhất.
* **Nhược điểm:** Tốn nhiều tài nguyên tính toán, thời gian thực hiện lâu hơn so với FGM.



Hình 22: Projected Gradient Descent (PGD) Attack

**+ DeepFool Attack:**

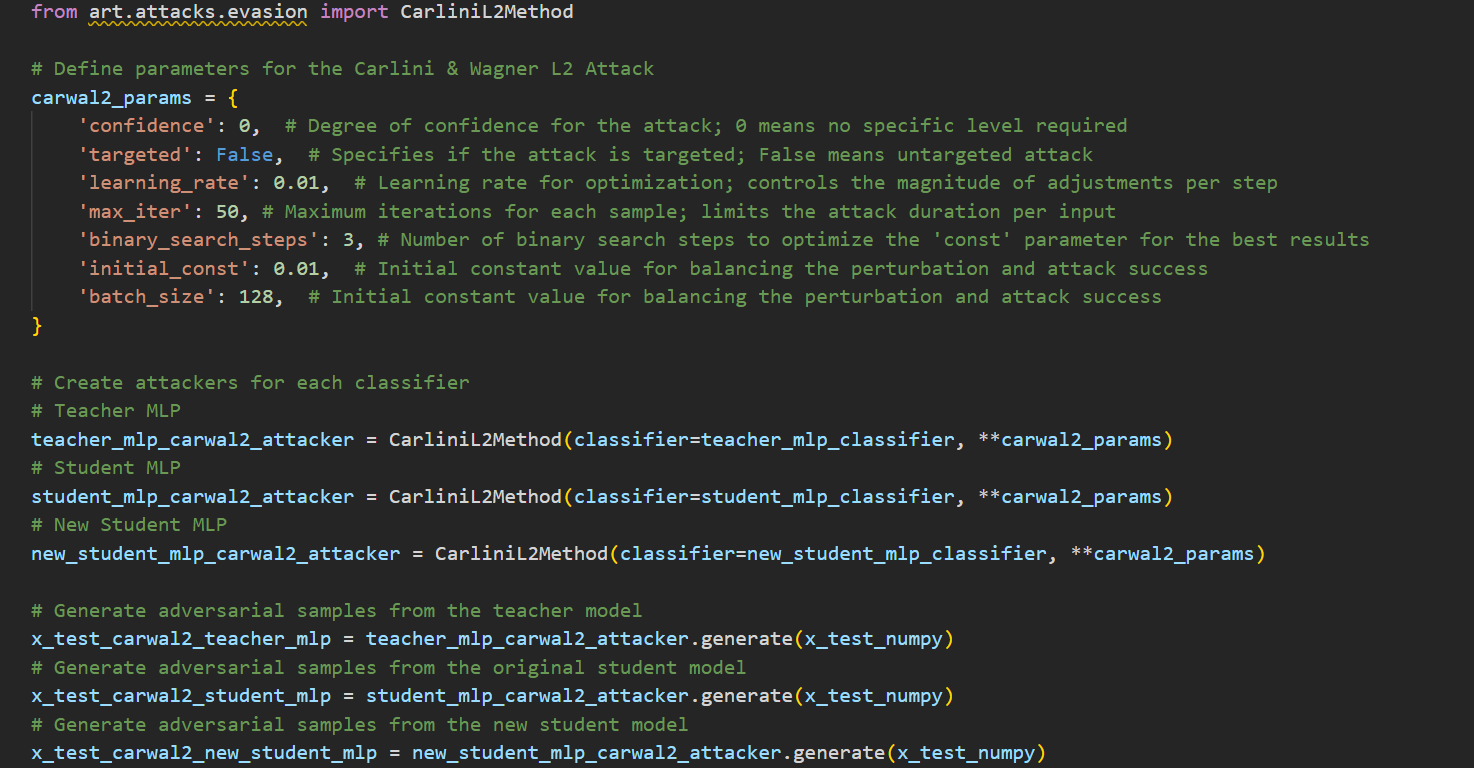
* **Cách thức hoạt động:** DeepFool tìm cách đưa điểm dữ liệu vào biên giới quyết định (decision boundary) của mô hình với lượng nhiễu nhỏ nhất. Thay vì di chuyển dọc theo gradient trực tiếp, loại này ước lượng hướng nào là hướng hiệu quả nhất để làm mô hình phân loại sai, dẫn đến lượng nhiễu ít hơn.
* **Tốc độ:** Tương đối nhanh, nhưng chậm hơn FGM do cần nhiều bước để xác định biên giới quyết định.
* **Ưu điểm:** Hiệu quả trong việc tìm nhiễu nhỏ nhất để đánh lừa mô hình, tạo ra mẫu đối kháng tinh tế.
* **Nhược điểm:** Đôi khi kém hiệu quả với các mô hình phòng vệ tốt, nhất là khi gradient bị làm tròn hoặc khi có nhiễu trong mô hình.



Hình 23: DeepFool Attack

**+ Carlini and Wagner L\_2 Attack:**

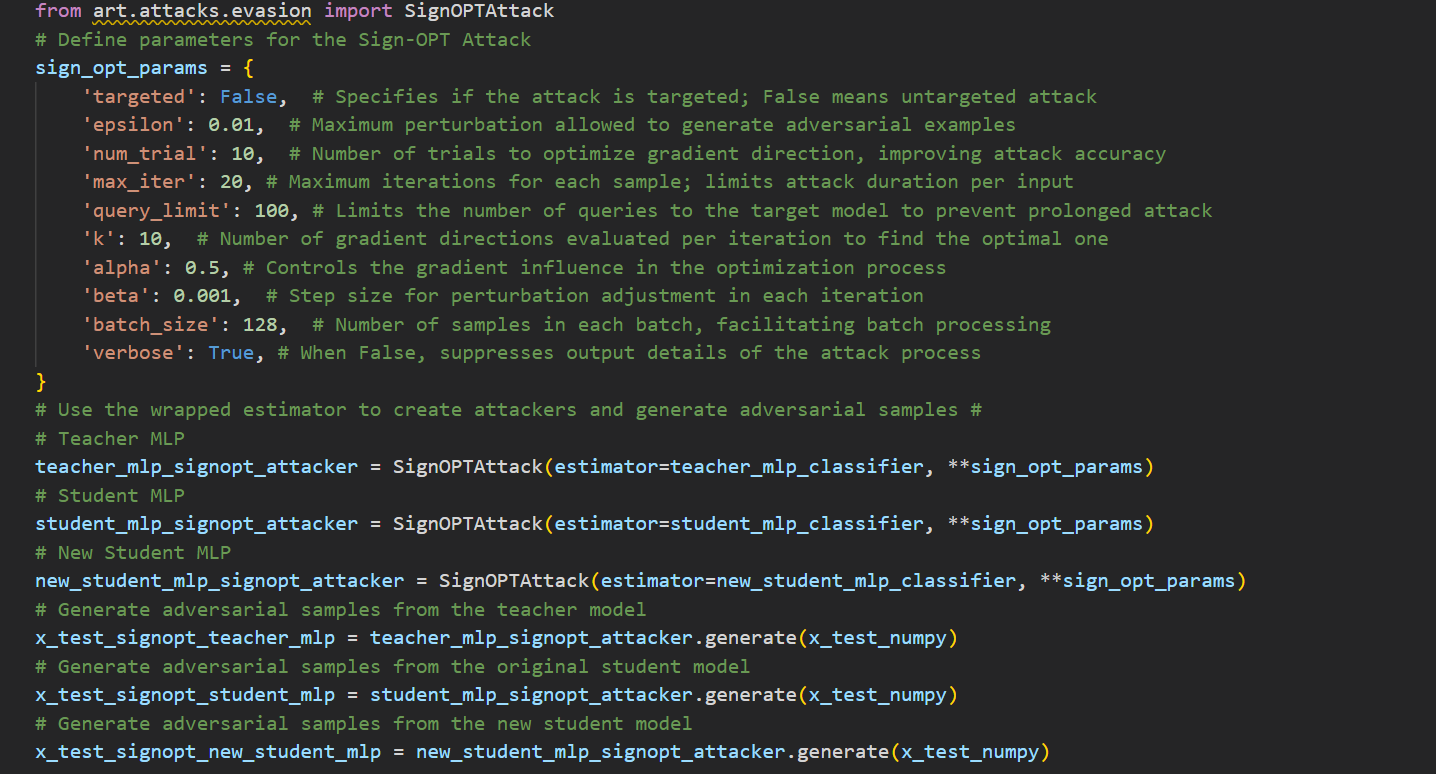
* **Cách thức hoạt động:** Phương pháp này sử dụng tối ưu hóa để tìm ra một mẫu đối kháng tối đa hóa tổn thất với mô hình, trong khi giữ mức độ biến đổi của dữ liệu gốc (lượng nhiễu) trong miền L2 nhỏ nhất. Phương pháp này áp dụng các chiến lược tối ưu hóa phi tuyến tính để vượt qua các cơ chế phòng thủ của mô hình.
* **Tốc độ:** Chậm do sử dụng tối ưu hóa phi tuyến tính phức tạp.
* **Ưu điểm:** Là một trong những tấn công mạnh nhất và rất khó phòng thủ. C&W có khả năng đánh bại nhiều mô hình với mức nhiễu nhỏ, đặc biệt với các mô hình được bảo vệ bởi các kỹ thuật làm tròn gradient hoặc mạng phòng thủ.
* **Nhược điểm:** Tốn thời gian và tài nguyên, yêu cầu tối ưu hóa phức tạp.



Hình 24: Carlini and Wagner L\_2 Attack

**+ Sign-OPT Attack:**

* **Cách thức hoạt động:** Sign-OPT là một phương pháp dựa trên tối ưu hóa gradient ước lượng và không yêu cầu gradient chính xác. Loại này thực hiện một loạt các ước lượng để tìm ra hướng tối ưu nhất cho tấn công, thường được sử dụng khi gradient không có sẵn (black-box attack).
* **Tốc độ:** Chậm, do phải ước lượng gradient qua nhiều bước thử và tối ưu hóa.
* **Ưu điểm:** Hiệu quả cao với các hệ thống không cung cấp gradient (black-box), tức là không yêu cầu thông tin chi tiết về mô hình.
* **Nhược điểm:** Tốn nhiều tài nguyên và thời gian, đặc biệt là với mô hình lớn hoặc phức tạp.



Hình 25: Sign-OPT Attack

## C. KẾT QUẢ

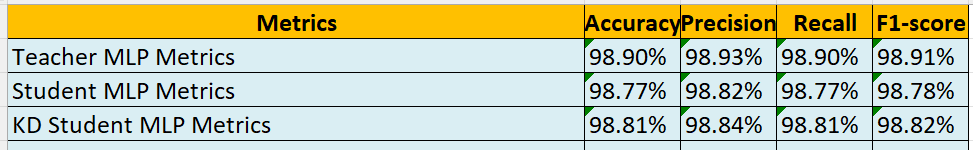
*Tham khảo file kết quả được đính kèm:* ***result\_kd\_aa\_evasion.xlsx***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Metrics** | **Accuracy** | **Precision** | **Recall** | **F1-score** |
| **Teacher - Student - KD Student** |  |  |  |  |
| Teacher MLP Metrics | 98.90% | 98.93% | 98.90% | 98.91% |
| Student MLP Metrics | 98.77% | 98.82% | 98.77% | 98.78% |
| KD Student MLP Metrics | 98.81% | 98.84% | 98.81% | 98.82% |
| **Adversarial Attacks - Evasion** |  |  |  |  |
| Teacher MLP Metrics (FGM) | 76.01% | 78.91% | 76.01% | 77.31% |
| Student MLP Metrics (FGM) | 78.96% | 83.49% | 78.96% | 80.66% |
| KD Student MLP Metrics (FGM) | 74.60% | 79.13% | 74.60% | 76.20% |
| Teacher MLP Metrics (PGD) | 86.34% | 86.18% | 86.34% | 86.20% |
| Student MLP Metrics (PGD) | 91.55% | 93.10% | 91.55% | 92.06% |
| KD Student MLP Metrics (PGD) | 87.83% | 89.55% | 87.83% | 88.51% |
| Teacher MLP Metrics (Deepfool) | 33.01% | 31.23% | 33.01% | 30.61% |
| Student MLP Metrics (Deepfool) | 69.51% | 58.28% | 69.51% | 63.28% |
| KD Student MLP Metrics (Deepfool) | 71.38% | 62.95% | 71.38% | 66.84% |
| Teacher MLP Metrics (Carlini and Wagner L\_2) | 97.05% | 97.14% | 97.05% | 97.09% |
| Student MLP Metrics (Carlini and Wagner L\_2) | 97.74% | 97.83% | 97.74% | 97.76% |
| KD Student MLP Metrics (Carlini and Wagner L\_2) | 97.48% | 97.57% | 97.48% | 97.50% |
| Teacher MLP Metrics (Sign-OPT) | 31.56% | 44.56% | 31.56% | 35.75% |
| Student MLP Metrics (Sign-OPT) | 26.21% | 40.62% | 26.21% | 31.44% |
| KD Student MLP Metrics (Sign-OPT) | 26.01% | 39.66% | 26.01% | 30.68% |

Bảng 1: Kết quả đồ án

**- Phân tích và đánh giá kết quả:**

***1. Hiệu năng ban đầu của các mô hình (Teacher - Student - KD Student)***



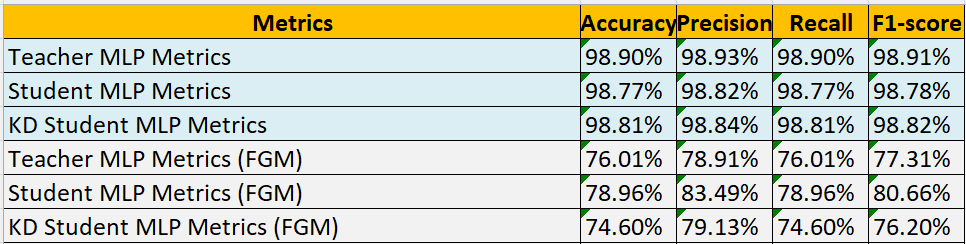
Hình 26: Hiệu năng ban đầu của các mô hình

* **Mô hình Teacher MLP:** Đạt độ chính xác (Accuracy) cao nhất với 98.90%, đồng thời các chỉ số Precision, Recall và F1-score đều ổn định ở mức xấp xỉ 98.91%. Điều này khẳng định mô hình Teacher, với kiến trúc phức tạp hơn, có khả năng tổng quát hóa tốt nhất.
* **Mô hình Student MLP:** Hiệu năng giảm nhẹ so với mô hình Teacher với Accuracy là 98.77%. Tuy nhiên, các chỉ số Precision, Recall, và F1-score vẫn duy trì ở mức tương đối cao (98.78%), chứng minh khả năng giữ lại phần lớn kiến thức từ mô hình Teacher.
* **Mô hình KD Student MLP:** Có sự cải thiện nhẹ so với mô hình Student MLP (Accuracy 98.81% **> 98.77%**), nhưng vẫn thấp hơn mô hình Teacher (98.81% **< 98.90%**). Điều này cho thấy ***kỹ thuật Knowledge Distillation giúp mô hình Student học hiệu quả hơn*** từ mô hình Teacher.

**Nhận xét:** KD Student đạt hiệu quả học tập cao hơn Student ban đầu, chứng minh hiệu quả của Knowledge Distillation trong việc truyền tải kiến thức từ Teacher sang Student.

***2. Độ bền của mô hình trước các tấn công đối kháng***

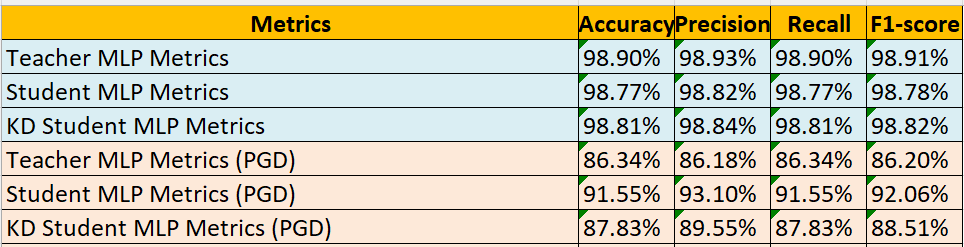
**+ Fast Gradient Method (FGM):**



Hình 27: Độ bền của mô hình trước Fast Gradient Method (FGM)

* **Teacher MLP:** Độ chính xác giảm mạnh còn 76.01%, cho thấy mô hình Teacher dễ bị tấn công trước tấn công này.
* **Student MLP:** Tăng khả năng chịu đựng tấn công với Accuracy đạt 78.96%, cao hơn Teacher.
* **KD Student MLP:** Kém bền hơn với Accuracy 74.60%, phản ánh rằng KD không giúp cải thiện hiệu quả chống lại FGM.

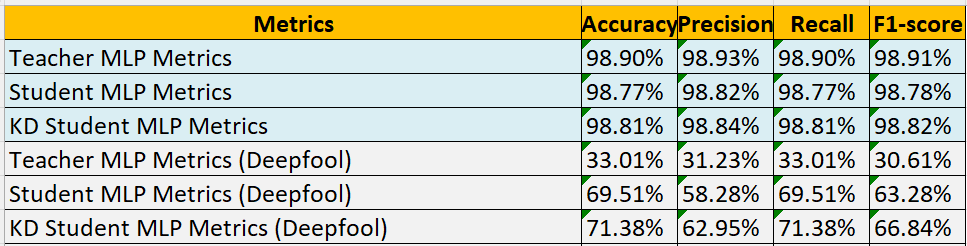
**+ Projected Gradient Descent (PGD):**

****

Hình 28: Độ bền của mô hình trước Projected Gradient Descent (PGD)

* **Teacher MLP:** Chịu ảnh hưởng nhưng vẫn duy trì hiệu năng tương đối tốt (Accuracy 86.34%).
* **Student MLP:** Thể hiện độ bền vượt trội với Accuracy 91.55%.
* **KD Student MLP:** Mức giảm nhẹ nhưng vẫn tốt hơn Teacher (Accuracy 87.83%).

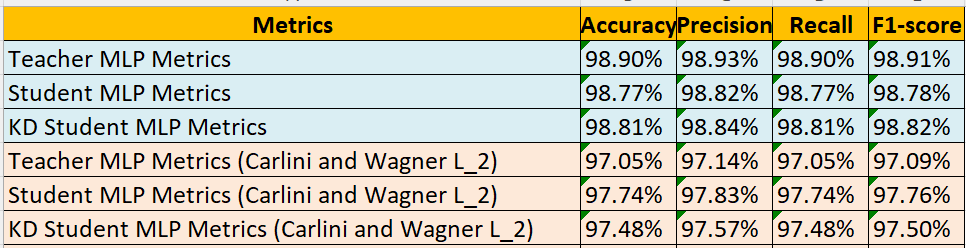
**+ Deepfool:**

****

Hình 29: Độ bền của mô hình trước Deepfool

* **Teacher MLP:** Hiệu năng giảm nghiêm trọng với Accuracy chỉ còn 33.01%.
* **Student MLP:** Tăng cường đáng kể khả năng chống chịu (Accuracy 69.51%).
* **KD Student MLP:** Hiệu năng cải thiện hơn Student với Accuracy 71.38%, cho thấy **KD giúp tăng độ bền trong tấn công Deepfool** này.

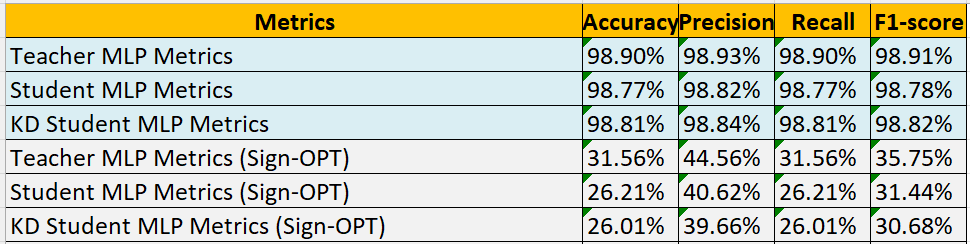
**+ Carlini and Wagner L\_2:**

****

Hình 30: Độ bền của mô hình trước Carlini and Wagner L\_2

* **Teacher MLP:** Duy trì hiệu năng cao (Accuracy 97.05%).
* **Student MLP:** Vượt qua Teacher với Accuracy 97.74%.
* **KD Student MLP:** Hiệu quả tương đối (Accuracy 97.48%), không chênh lệch đáng kể, KD vẫn có thể được coi là **chống chịu tốt đối với cuộc tấn công Carlini and Wagner L\_2** này.

**+ Sign-OPT:**

****

Hình 31: Độ bền của mô hình trước Sign-OPT

* **Teacher MLP:** Kết quả thấp nhất trong tất cả các cuộc tấn công (Accuracy 31.56%).
* **Student MLP và KD Student MLP:** Hiệu năng thấp và không có sự khác biệt đáng kể (Accuracy ~26%).

**- Kết luận từ phân tích:**

***1. Hiệu năng ban đầu:*** Mô hình KD Student MLP được huấn luyện từ mô hình teacher cho thấy hiệu quả tổng quát hóa tốt hơn Student MLP ban đầu được huấn luyện với dữ liệu gốc, khẳng định rõ vai trò của Knowledge Distillation.

***2. Độ bền của mô hình:***

* **FGM và PGD:** Mô hình Student MLP tỏ ra vượt trội hơn so với KD Student, cho thấy KD không hiệu quả với 2 loại tấn công.
* **Deepfool và Carlini & Wagner L\_2:** KD Student MLP hoạt động tốt hơn, khẳng định vai trò tích cực của KD.
* **Sign-OPT:** Tất cả các mô hình đều thể hiện độ bền rất kém, phản ánh cần cải thiện thêm khả năng phòng thủ trước cuộc tấn công Sign-OPT vô cùng mạnh mẽ này.

***3. Kết luận:*** Knowledge Distillation đóng vai trò tích cực trong việc cải thiện khả năng tổng quát và độ bền trước một số loại tấn công như Deepfool và Carlini & Wagner L\_2, nhưng không phải là giải pháp tối ưu cho mọi trường hợp.

# CHƯƠNG III. TỔNG KẾT

*Nội dung chương này sẽ trình bày về phần* ***kết luận*** *và* ***hướng phát triển*** *của đồ án.*

## A. KẾT LUẬN

Trong bối cảnh ngày càng gia tăng các cuộc tấn công đối kháng nhằm khai thác điểm yếu của các mô hình học máy, việc nghiên cứu và cải thiện khả năng phòng thủ trước những tấn công này là vô cùng cấp thiết. Đồ án đã được triển khai nhằm đánh giá hiệu quả của kỹ thuật **Knowledge Distillation (KD)** trong việc tăng cường độ bền của các mô hình học máy trước các tấn công đối kháng. Qua quá trình thực nghiệm với các mô hình **Teacher MLP**, **Student MLP**, và **KD Student MLP**, kết hợp với nhiều phương pháp tấn công đối kháng phổ biến như **FGM**, **PGD**, **Deepfool**, **Carlini & Wagner L2**, và **Sign-OPT**, các kết quả đã được thu thập và phân tích kỹ lưỡng.

Kết quả thực nghiệm cho thấy:

**1. Hiệu năng tổng quát hóa (Accuracy):**

Mô hình KD Student MLP đạt hiệu suất gần tương đương mô hình Teacher MLP, vượt trội hơn Student MLP ban đầu. Điều này khẳng định rằng kỹ thuật KD có khả năng truyền tải hiệu quả kiến thức từ mô hình Teacher sang Student, giúp cải thiện chất lượng của mô hình đơn giản hơn (Student MLP).

**2. Độ bền trước các tấn công đối kháng:**

KD Student MLP thể hiện khả năng phòng thủ vượt trội trong các tấn công như Deepfool và Carlini & Wagner L2, cho thấy KD đóng vai trò quan trọng trong việc tăng cường khả năng chịu đựng các tấn công phức tạp.

Tuy nhiên, trong các tấn công như FGM và Sign-OPT, mô hình KD Student không cho thấy sự cải thiện rõ rệt so với Student MLP. Điều này chỉ ra rằng kỹ thuật KD không hoàn toàn tối ưu hóa khả năng phòng thủ đối với mọi phương pháp tấn công.

Các mô hình, bao gồm cả Teacher, đều thể hiện độ bền rất kém với tấn công Sign-OPT, cho thấy cần có thêm nghiên cứu để khắc phục điểm yếu này.

**3. Ý nghĩa và đóng góp:**

Đồ án đã cung cấp một cái nhìn toàn diện về tác động của kỹ thuật Knowledge Distillation lên khả năng phòng thủ của các mô hình học máy.

Các kết quả thực nghiệm đã chỉ ra rằng KD có hiệu quả trong việc nâng cao hiệu năng tổng quát hóa và độ bền trước một số loại tấn công đối kháng. Tuy nhiên, cần có những giải pháp bổ trợ để cải thiện khả năng chống chịu đối với những tấn công phức tạp hơn như Sign-OPT.

Kết luận đồ án này không chỉ khẳng định vai trò của Knowledge Distillation trong việc nâng cao độ bền của các mô hình học máy (đặc biệt là mô hình teacher) mà còn mở ra nhiều hướng nghiên cứu tiếp theo nhằm tối ưu hóa khả năng phòng thủ của các mô hình trước các tấn công đối kháng ngày càng tinh vi.

## B. HƯỚNG PHÁT TRIỂN

Thực hiện cải thiện, cải tiến và bổ sung sau với 2 đề xuất:

**Đề xuất 1.** Thực hiện triển khai nhiều kiến trúc mô hình khác nhau cho cả mô hình Teacher và Student, sau đó so sánh hiệu suất giữa các kiến trúc mô hình khác nhau trong việc chống lại các cuộc tấn công đối kháng để có thể đưa ra kết luận rằng: loại kiến trúc mô hình nào sẽ chống chịu tốt hơn với các cuộc tấn công đối kháng thuộc loại tấn công né tránh (Evasion Attacks) như ở hiện tại.

**Đề xuất 2.** Thực hiện mở rộng 2 loại tấn công mới dựa vào ART, không chỉ là một loại tấn công né tránh (Evasion Attacks) như ở hiện tại.

* Tấn công làm nhiễm độc dữ liệu (Poisoning Attacks).
* Tấn công trích xuất mô hình (Model Extraction).

# TÀI LIỆU THAM KHẢO

*github*. (n.d.). Retrieved from https://github.com/Trusted-AI/adversarial-robustness-toolbox

*pytorch*. (n.d.). Retrieved from https://pytorch.org/tutorials/beginner/knowledge\_distillation\_tutorial.html