**CÂU HỎI ÔN TẬP**

**Môn học: NT522 – Phương pháp học máy trong An toàn thông tin**

**Câu 1. Phân biệt các loại AI khác nhau? Trình bày được định nghĩa của các loại học máy (Học giám sát, không giám sát, bán giám sát, học tăng cường)?**

*Trả lời:*

Artificial Intelligence (AI) là sự mô phỏng quá trình trí tuệ của con người bằng máy móc, đặc biệt là các hệ thống hệ máy tính. Mỗi loại AI có những đặc điểm và ứng dụng riêng, phục vụ cho các mục đích khác nhau trong cuộc sống và công nghệ.

1. Phân loại theo khả năng và sự thông minh:

* Artificial Narrow Intelligence (Trí tuệ nhân tạo hẹp hay còn được gọi là Weak AI): được thiết kế để thực hiện một nhiệm vụ cụ thể.

Ví dụ: Siri của Apple, hệ thống đề xuất của Netflix, các chatbot hỗ trợ khách hàng.

* Artificial General Intelligence (Trí tuệ nhân tạo chung hay còn được gọi là Strong AI): có khả năng hiểu, học và áp dụng kiến thức ở nhiều lĩnh vực khác nhau, tương tự như con người.

VD: robot.

* Artificial Super Intelligence (Siêu trí tuệ nhân tạo): là phiên bản trí tuệ nhân tạo vượt trội hơn trí tuệ con người và có thể thực hiện mọi nhiệm vụ tốt hơn con người. Ngày nay, Super AI chỉ là khái niệm mang tính giả thuyết và chưa có trong thực tế.

2. Phân loại theo phương pháp:

* Học Máy (Machine Learning): là một nhánh của AI cho phép các hệ thống tự động học hỏi và cải thiện từ kinh nghiệm mà không cần lập trình rõ ràng.
* Học Sâu (Deep Learning): là một phần của học máy, sử dụng các mạng nơ-ron sâu để học từ dữ liệu phức tạp. Áp dụng nhiều trong các lĩnh vực như nhận dạng hình ảnh, xử lý ngôn ngữ tự nhiên và chơi game.
* Mạng Nơ-ron (Neural Networks): là một mô hình tính toán lấy cảm hứng từ cấu trúc và chức năng của não người. Có thể là mạng nơ-ron truyền thống (NN) hoặc mạng nơ-ron sâu (DNN),…

3. Phân loại theo công nghệ:

* Reactive Machine – AI phản ứng: đây là loại AI cơ bản và cổ xưa nhất. Nó có khả năng phản ứng với các kích thích trong môi trường xung quanh nhưng không lưu trữ kinh nghiệm hay ký ức trong quá khứ.
* Limited Memory – AI với ký ức ngắn hạn: có khả năng phản ứng với cách kích thích của môi trường, loại trí tuệ nhân tạo này có thể lưu trữ dữ liệu và các dự đoán từ trước trong một khoảng thời gian ngắn, từ đó phân tích dữ liệu.
* Theory of mind – AI lý thuyết về tâm trí: có khả năng mô phỏng cảm xúc, suy nghĩ và cách con người tương tác với xã hội.
* Self aware – AI tự nhận thức: sẽ phát triển cảm xúc của riêng nó và nhận dạng được những cảm xúc này. Nó sẽ có nhu cầu, niềm tin, cảm xúc, phát triển ý thức, hiểu về sự tồn tại của chính nó trên thế giới này.

Trình bày định nghĩa của các loại học máy (Học giám sát, không giám sát, bán giám sát, học tăng cường):

1. Học giám sát (Supervised Learning)

Học có giám sát là loại học mà máy tính được đào tạo trên một tập dữ liệu đã được gán nhãn. Nói cách khác, trong bài toán này, chúng ta cung cấp cho máy tính cả dữ liệu đầu vào và đầu ra mong muốn (nhãn). Mục đích của việc học có giám sát là dự đoán kết quả đầu ra cho dữ liệu mới chưa được gán nhãn.

Học có giám sát lại được chia thành hai nhóm chính:

* Hồi quy (Regression): khi kết quả đầu ra là một giá trị liên tục, chúng ta sử dụng thuật toán hồi quy.

Ví dụ: dự đoán giá nhà, dự đoán doanh số.

* Phân loại (Classification): khi kết quả đầu ra là một giá trị rời rạc, chúng ta sử dụng thuật toán phân loại.

Ví dụ: phân loại văn bản, nhận dạng khuôn mặt.

1. Học không giám sát (Unsupervised Learning)

Học không giám sát xử lý vấn đề khi chúng ta không biết kết quả đầu ra mong muốn và không có nhãn cho dữ liệu. Mục tiêu của học không giám sát là khám phá cấu trúc ẩn trong dữ liệu và tìm ra một cách biểu diễn dữ liệu phù hợp.

Học không giám sát chủ yếu được chia thành ba nhóm:

* Phân cụm (Clustering): nhóm các đối tượng tương tự lại với nhau.

Ví dụ: phân loại khách hàng, phân loại tài liệu.

* Giảm chiều dữ liệu (Dimensionality Reduction): giảm số lượng biến trong dữ liệu, giữ lại các thông tin quan trọng nhất.

Ví dụ: giảm chiều dữ liệu ảnh, tìm các yếu tố tiềm ẩn trong thông tin người dùng.

* Nhận dạng mẫu (Pattern Recognition): quá trình phát hiện và phân loại các cấu trúc hoặc xu hướng ẩn trong dữ liệu không có gán nhãn trước, giúp khám phá mối quan hệ và mẫu hình trong tập dữ liệu.
* Ví dụ: nhận diện chữ viết tay, phân loại hình ảnh y tế, nhận dạng khuôn mặt trong an ninh.

1. Học bán giám sát (Semi-supervised Learning)

Học bán giám sát kết hợp cả hai phương pháp học có giám sát và học không giám sát, trong đó một phần dữ liệu được gán nhãn và phần còn lại không. Mục tiêu của học bán giám sát là tận dụng lợi thế của cả hai phương pháp để cải thiện độ chính xác của mô hình.

* Phân cụm bán giám sát (Semi-Supervised Clustering): phương pháp phân cụm kết hợp dữ liệu có gán nhãn và không gán nhãn để cải thiện độ chính xác của việc nhóm các đối tượng tương tự lại với nhau.

VD: phân loại văn bản với một số ít văn bản được gán nhãn, phân nhóm khách hàng với một số ít khách hàng đã được phân loại.

**Câu 2. Các tiêu chí để đánh giá mô hình học máy? Cách chọn tiêu chí đánh giá phù hợp trong một số ngữ cảnh bài toán An toàn thông tin.**

*Trả lời:*

*Tại sao đánh giá là cần thiết cho mô hình thành công?*

Hai nguyên nhân lớn nhất dẫn đến hiệu suất kém của các thuật toán học máy là:

* Overfitting: xảy ra khi Mô hình hoạt động tốt đối với một tập hợp dữ liệu cụ thể (Dữ liệu đã biết) mà không phù hợp với dữ liệu bổ sung (Dữ liệu không xác định).
* Underfitting: xảy ra khi mô hình không thể nắm bắt đầy đủ cấu trúc cơ bản của dữ liệu.

**1. Các chỉ số phân loại**

Số liệu phân loại

Khi thực hiện các dự đoán phân loại, có bốn loại kết quả có thể xảy ra:

* True positives (TP) là khi bạn dự đoán một quan sát thuộc về một lớp và nó thực sự thuộc về lớp đó.
* True Negative (TN) là khi bạn dự đoán một quan sát không thuộc về một lớp và nó thực sự không thuộc lớp đó.
* False Positive (FP) xảy ra khi bạn dự đoán một quan sát thuộc về một lớp trong khi thực tế thì không.
* False Negative (FN) xảy ra khi bạn dự đoán một quan sát không thuộc về một lớp trong khi thực tế là nó có.

Bốn kết quả này thường được vẽ trên một ma trận nhầm lẫn (confusion matrix).

Trong thực tế có ba độ đo chủ yếu để đánh giá một mô hình phân loại là Accuracy, Precision and Recall.

* Accuracy được định nghĩa là tỷ lệ phần trăm dự đoán đúng cho dữ liệu thử nghiệm.

accuracy = =

* Precision được định nghĩa là phần nhỏ của các ví dụ có liên quan (tích cực thực sự) trong số tất cả các ví dụ được dự đoán là thuộc một lớp nhất định.

precision = =

* Recall được định nghĩa là phần nhỏ của các ví dụ được dự đoán thuộc về một lớp so với tất cả các ví dụ thực sự thuộc về lớp đó.

recall = =

Tùy thuộc vào ứng dụng, bạn có thể muốn ưu tiên Recall hoặc Precision cao hơn.

A mathematical equation with a line and a number

Description automatically generated with medium confidence

Tham số β cho phép chúng ta kiểm soát sự cân bằng giữa Precision và Recall.

* β < 1 tập trung nhiều hơn vào Precision.
* β > 1 tập trung nhiều hơn vào Recall.
* β = 1 tập trung vào cả Precision và Recall.

Khi β = 1, ta sử dụng F1-score, là kỳ vọng harmonic (harmonic mean) của Precision và Recall. F1-score lớn khi cả 2 giá trị Precision và Recall đều lớn. Ngược lại, chỉ cần 1 giá trị nhỏ sẽ làm cho F1-Score nhỏ.

A black text on a white background

Description automatically generated

* Đường cong PR

Đường cong PR chỉ đơn giản là một đồ thị có giá trị Precision trên trục y và giá trị Recall trên trục x. Nói cách khác, đường cong PR chứa TP / (TP + FN) trên trục y và TP / (TP + FP) trên trục x.

* Đường cong ROC

Đường cong ROC ( đường đặc tính hoạt động của máy thu ) là một đồ thị thể hiện hiệu suất của một mô hình phân loại ở tất cả các ngưỡng phân loại. Đường cong này vẽ hai tham số:

Tỷ lệ Tích cực Thực sự (Recall)

Tỷ lệ Tích cực giả (FPR)

A black line with black text

Description automatically generated

* AUC

Diện tích dưới đường cong (AUC), là thước đo tổng hợp hiệu suất của bộ phân loại nhị phân trên tất cả các giá trị ngưỡng có thể có (và do đó nó là ngưỡng bất biến ) . Nghĩa là, AUC đo toàn bộ diện tích hai chiều bên dưới toàn bộ đường cong ROC.

Vì AUC là một phần của diện tích hình vuông đơn vị, nên giá trị của nó sẽ luôn nằm trong khoảng từ 0 đến 1,0.

**2. Chỉ số hồi quy**

* Mean Squared Error: về cơ bản, nó tìm thấy sai số bình phương trung bình giữa các giá trị được dự đoán và thực tế. MSE là thước đo chất lượng của một công cụ ước tính - nó luôn không âm và các giá trị càng gần 0 càng tốt.
* Mean Absolute Error: đo độ lớn trung bình của các lỗi trong một tập hợp các dự đoán mà không cần xem xét hướng của chúng. Đó là giá trị trung bình trên mẫu thử nghiệm về sự khác biệt tuyệt đối giữa dự đoán và quan sát thực tế, trong đó tất cả các khác biệt riêng lẻ có trọng số bằng nhau.
* Root Mean Square Error (RMSE) hoặc Root Mean Square Deviation (RMSD) là căn bậc hai của mức trung bình của các sai số bình phương. RMSE là độ lệch chuẩn của các phần dư (sai số dự đoán).

**Cách Chọn Tiêu Chí Đánh Giá Phù Hợp Trong Ngữ Cảnh An Toàn Thông Tin**

Trong lĩnh vực an toàn thông tin, các tiêu chí đánh giá mô hình có thể khác nhau tùy thuộc vào mục tiêu cụ thể của bài toán. Dưới đây là một số ngữ cảnh cụ thể và cách chọn tiêu chí phù hợp:

1. **Phát hiện xâm nhập mạng (Intrusion Detection):**
   * **Tiêu chí:** Độ nhạy (Recall) và F1 Score.
   * **Lý do:** Mục tiêu chính là phát hiện tất cả các hành vi xâm nhập, do đó cần chú trọng vào Recall. F1 Score giúp cân bằng giữa Recall và Precision, tránh quá nhiều cảnh báo giả.
2. **Phát hiện phần mềm độc hại (Malware Detection):**
   * **Tiêu chí:** Độ chính xác của dự đoán dương tính (Precision) và AUC-ROC.
   * **Lý do:** Việc dự đoán chính xác các phần mềm độc hại quan trọng để tránh đánh dấu sai các phần mềm hợp pháp. AUC-ROC giúp đánh giá hiệu suất phân loại tổng quát của mô hình.
3. **Phân loại email spam (Spam Email Classification):**
   * **Tiêu chí:** F1 Score và Precision.
   * **Lý do:** Cần cân bằng giữa việc phát hiện email spam và tránh lọc nhầm email hợp pháp, do đó F1 Score là phù hợp. Precision cao để đảm bảo email hợp pháp không bị lọc nhầm.
4. **Phát hiện gian lận tài chính (Fraud Detection):**
   * **Tiêu chí:** Precision và F1 Score.
   * **Lý do:** Việc phát hiện gian lận cần độ chính xác cao để giảm thiểu các cảnh báo sai, do đó Precision là quan trọng. F1 Score giúp duy trì cân bằng với Recall để không bỏ sót quá nhiều giao dịch gian lận.
5. **Xác thực danh tính (Identity Verification):**
   * **Tiêu chí:** Độ đặc hiệu (Specificity) và AUC-ROC.
   * **Lý do:** Đảm bảo rằng những người hợp pháp không bị từ chối, do đó Specificity cao là cần thiết. AUC-ROC giúp đánh giá khả năng phân biệt giữa các danh tính hợp pháp và giả mạo.

**Câu 3. Trình bày cách phân chia dữ liệu và tiền xử lí dữ liệu phổ biến?**

*Trả lời:*

1. Phân chia dữ liệu

Phân chia dữ liệu là quá trình chia tách tập dữ liệu thành các tập con để sử dụng cho việc huấn luyện mô hình, đánh giá hiệu suất mô hình và kiểm định. Các phương pháp phân chia dữ liệu thường gặp bao gồm:

* Phân chia theo tỷ lệ (Train-Validation-Test split): Phương pháp phổ biến nhất, tách dữ liệu thành tập huấn luyện, tập validation (kiểm định) và tập kiểm tra theo tỷ lệ nhất định. Thường dùng tỷ lệ như 70-15-15% hoặc 80-10-10%.
* Phân chia chuỗi thời gian: Đối với dữ liệu chuỗi thời gian, việc phân chia cần duy trì thứ tự thời gian, ví dụ như lấy dữ liệu cũ hơn để huấn luyện và dữ liệu gần đây hơn để kiểm tra.
* Phân chia chéo (Cross-validation): Dùng để kiểm tra tính tổng quát của mô hình bằng cách chia dữ liệu thành nhiều phần, huấn luyện trên các phần và kiểm tra trên phần còn lại.

2. Tiền xử lý dữ liệu

Tiền xử lý dữ liệu là quá trình chuẩn bị và làm sạch dữ liệu để chuẩn bị cho việc phân tích và huấn luyện mô hình. Các bước tiền xử lý thường gặp bao gồm:

* Xử lý dữ liệu thiếu (Missing data): Điền giá trị thiếu hoặc loại bỏ các mẫu dữ liệu thiếu.

Trong một số bộ dữ liệu, chúng ta có thể gặp giá trị NA trong các đặc trưng. Điều này đơn giản là dữ liệu bị thiếu. Khi xử lý loại dữ liệu này, có nhiều cách tiếp cận:

- Thay thế giá trị thiếu bằng giá trị trung bình hoặc trung vị đối với dữ liệu số và bằng mode đối với dữ liệu phân loại.

- Loại bỏ toàn bộ các hàng chứa giá trị NA.

- Loại bỏ toàn bộ các đặc trưng chứa giá trị NA (điều này hữu ích nếu số lượng giá trị NA lớn hơn 50% trong một đặc trưng).

- Thay thế giá trị NA bằng 0.

Nếu bạn chọn phương án loại bỏ, có nguy cơ mất thông tin quan trọng từ dữ liệu. Do đó, tốt hơn hết là lựa chọn các phương án thay thế giá trị thiếu.

* Xử lý dữ liệu mất cân bằng (Handling Imbalanced Data)

Xử lý dữ liệu mất cân bằng để giảm thiểu vấn đề overfitting và underfitting. Một số phương pháp bao gồm:

- Lấy mẫu dưới (Under-sampling) lớp đa số: Lấy mẫu dưới lớp đa số sẽ lấy lại mẫu các điểm của lớp đa số trong dữ liệu để làm cho chúng bằng với lớp thiểu số.

- Lấy mẫu quá mức (Over-sampling) lớp thiểu số bằng cách nhân bản: Lấy mẫu quá mức lớp thiểu số sẽ lấy lại mẫu các điểm của lớp thiểu số trong dữ liệu để làm cho chúng bằng với lớp đa số.

- Lấy mẫu quá mức lớp thiểu số bằng kỹ thuật Oversampling Tổng hợp Lớp Thiểu số (SMOTE): Các mẫu tổng hợp được tạo ra cho lớp thiểu số và bằng với lớp đa số.

Đây là những phương pháp để điều chỉnh sự mất cân bằng trong dữ liệu, giúp cải thiện hiệu suất của mô hình máy học bằng cách làm cho dữ liệu của các lớp khác nhau trở nên cân bằng hơn.

* Xử lý ngoại lệ (Handling Outliers)

- Loại bỏ ngoại lệ: Các mục chứa ngoại lệ được xóa khỏi phân phối.

- Thay thế giá trị: Một cách tiếp cận khác là xử lý ngoại lệ như các giá trị thiếu và thay thế chúng bằng các giá trị thay thế phù hợp.

- Giới hạn (Capping): Sử dụng một giá trị tùy ý hoặc giá trị từ phân phối biến để thay thế các giá trị tối đa và tối thiểu.

- Rời rạc hóa (Discretization): Là quá trình chuyển đổi các biến liên tục, mô hình và hàm số thành các biến rời rạc. Điều này được thực hiện bằng cách xây dựng một loạt các khoảng liên tục (hoặc thùng) mà bao gồm phạm vi của biến/mô hình/hàm số mong muốn của chúng ta

* Chuẩn hóa dữ liệu (Normalization/Standardization): Đưa các đặc trưng về cùng một khoảng giá trị để tránh ảnh hưởng của độ lớn của đặc trưng đến mô hình.
* Biến đổi dữ liệu (Data transformation):

Biến đổi Log là một trong những kỹ thuật phổ biến nhất. Đây thường được sử dụng để biến đổi phân phối lệch thành phân phối chuẩn hoặc ít lệch hơn.

Chúng ta lấy logarit của các giá trị trong một cột và sử dụng những giá trị đó làm cột trong biến đổi này. Nó được dùng để xử lý dữ liệu lệch và làm cho dữ liệu trở nên gần hơn với phân phối chuẩn.

* Bining

Phân khối đặc trưng hoặc phân khối dữ liệu là một kỹ thuật tiền xử lý dữ liệu. Nó có thể được sử dụng để giảm thiểu các tác động của các sai số quan sát nhỏ, tính các giá trị thông tin.

- Phân khối là cách để chuyển đổi các biến số liên tục số thành các biến rời rạc bằng cách phân loại chúng dựa trên phạm vi giá trị của cột mà chúng rơi vào.

- Trong loại biến đổi này, chúng ta tạo ra các thùng. Mỗi thùng cho phép một phạm vi cụ thể của các giá trị số liên tục. Điều này ngăn chặn việc quá khớp và tăng tính ổn định của mô hình.

Phân khối có thể được thực hiện theo các cách khác nhau được liệt kê dưới đây:

- Phân khối độ rộng cố định

- Phân khối theo phân vị

- Phân khối theo cảm nhận

* Encoding

Trong các bộ dữ liệu, chúng ta có thể có các kiểu dữ liệu đối tượng (object). Để xây dựng một mô hình, chúng ta cần có tất cả các đặc trưng ở dạng kiểu số nguyên. Do đó, Label Encoder và OneHot Encoder được sử dụng để chuyển đổi kiểu dữ liệu đối tượng thành kiểu dữ liệu số nguyên.

- Label Encoding: là quá trình chuyển đổi các biến phân loại thành các giá trị số nguyên một cách tuần tự.

- One-hot Encoding: tạo ra một biến giả cho mỗi giá trị duy nhất trong biến phân loại.

* Feature Scalling

Feature Scaling là quá trình điều chỉnh tỷ lệ hoặc chuyển đổi tất cả các giá trị trong bộ dữ liệu về một tỷ lệ nhất định.

- Standardization: là quá trình tỷ lệ hoá các giá trị dữ liệu sao cho chúng có tính chất của phân phối chuẩn chuẩn hóa.

Mean Normalization: là một phương pháp để tỷ lệ lại các đặc trưng vào một khoảng cố định [-1, 1] với giá trị trung bình bằng 0.

- Normalization:

Min-max Normalization: là một phương pháp để tỷ lệ lại các đặc trưng vào một khoảng cố định [0, 1] bằng cách trừ đi giá trị tối thiểu của đặc trưng và sau đó chia cho phạm vi của đặc trưng.

* Loại bỏ nhiễu (Noise removal): Loại bỏ nhiễu từ dữ liệu, chẳng hạn như loại bỏ outlier.
* Tạo đặc trưng mới (Feature engineering): Tạo ra các đặc trưng mới từ dữ liệu ban đầu, giúp mô hình học được hiệu quả hơn.
* Giảm chiều dữ liệu (Dimensionality reduction): Giảm số chiều của dữ liệu để giảm chi phí tính toán và cải thiện hiệu suất mô hình.

**Câu 4. Tác dụng của hàm kích hoạt (activation function) trong các mạng nơ-ron? Một số hàm activation phổ biến?**

*Trả lời:*

Hàm kích hoạt (activation function) đóng vai trò là cầu nối giữa các lớp nơ-ron trong mạng, giúp xác định xem liệu nơ-ron nào nên được kích hoạt và truyền tiếp thông tin sang lớp tiếp theo.

1. Các tác dụng chính của hàm kích hoạt trong các mạng nơ-ron:

* Khả năng phi tuyến (Non-linearity):

Mạng nơ-ron với các lớp kích hoạt phi tuyến có khả năng học và biểu diễn các mô hình phức tạp hơn so với các mô hình tuyến tính. Điều này là do các hàm kích hoạt phi tuyến như ReLU (Rectified Linear Unit), Sigmoid, Tanh, etc., có thể biểu diễn các mối quan hệ không tuyến tính giữa đầu vào và đầu ra của mạng.

* Chuyển đổi và chuẩn hóa dữ liệu:

Hàm kích hoạt có thể chuyển đổi các giá trị đầu vào thành một dạng phù hợp để truyền tiếp sang lớp tiếp theo.

* Độ dốc gradient (Gradient Descent):

Các hàm kích hoạt phải có tính chất có đạo hàm (derivative) để tính toán gradient cho việc cập nhật trọng số trong quá trình học.

* Điều tiết quá trình học:

Việc lựa chọn hàm kích hoạt phù hợp có thể cải thiện hiệu suất của mạng bằng cách điều tiết quá trình học và giảm thiểu hiện tượng overfitting.

2. Một số hàm activation phổ biến

* A symbol of a mathematical equation

  Description automatically generated with medium confidenceSigmoid

- Hàm kích hoạt Sigmoid được xác định bởi

- Là hàm liên tục, đưa giá trị đầu ra về khoảng (0, 1), thường được sử dụng cho các bài toán phân loại nhị phân.

- Dễ dàng bị vấn đề vanishing gradient khi giá trị đầu vào rất lớn hoặc rất nhỏ.

* A blue text with a white background

  Description automatically generated with medium confidenceReLU Rectified Linear Unit

- Hàm kích hoạt ReLU được xác định bởi

- Là hàm kích hoạt phi tuyến tính đơn giản và hiệu quả, range 0 -> infinity

- Thường được sử dụng trong các mạng nơ-ron sâu (deep neural networks) do tính khả năng học nhanh và hiệu quả tính toán.

* A mathematical equation with black text

  Description automatically generatedTanh (Hyperbolic Tangent)

- Hàm kích hoạt Tanh được xác định bởi

- Đưa giá trị đầu ra về khoảng (-1, 1), có tính chất tương tự Sigmoid nhưng với khoảng giá trị mở rộng hơn.

- Thường được sử dụng trong các lớp ẩn của mạng nơ-ron.

* Softmax

- Hàm kích hoạt Softmax được sử dụng cho lớp đầu ra của mạng nơ-ron phân loại đa lớp (classification).

- Chuyển đổi các giá trị thành xác suất, tổng các xác suất bằng 1.

- Phù hợp để xử lý bài toán phân loại nhiều lớp.

* Leaky ReLU

- Hàm kích hoạt Leaky ReLU có dạng với a là một hằng số nhỏ dương (thường là 0.01).

- Giúp khắc phục vấn đề "dying ReLU" (ReLU bị chết) khi giá trị đầu vào nhỏ hơn 0.

**Câu 5. Phân biệt sự khác nhau của các dạng thuật toán học máy như Phân cụm (clustering), phát hiện sự dị thường (anomaly detection), phân loại (classification).**

*Trả lời:*

|  |  |  |  |
| --- | --- | --- | --- |
| **Loại thuật toán** | **Mục đích** | **Phương pháp hoạt động** | **Ví dụ các thuật toán** |
| **Phân cụm (Clustering)** | Phân nhóm dữ liệu thành các nhóm con dựa trên sự tương đồng | Các thuật toán phân cụm cố gắng tìm ra cách chia tập dữ liệu sao cho các điểm trong cùng một nhóm có tính chất tương tự nhau (gần nhau), và các điểm khác nhóm có tính chất khác biệt (xa nhau). | K-means, Hierarchical Clustering, DBSCAN |
| **Phát hiện sự dị thường**  **(Anomaly detection)** | Nhận diện các điểm dữ liệu không tuân theo mẫu chung của dữ liệu | Các thuật toán phát hiện sự dị thường cố gắng xác định các điểm dữ liệu hiếm gặp hoặc không giống như hầu hết các điểm dữ liệu khác. | Isolation Forest, One-Class SVM, Autoencoders |
| **Phân loại (Classification)** | Dự đoán lớp hoặc nhãn của một điểm dữ liệu dựa trên đặc trưng | Các thuật toán phân loại học từ dữ liệu đã được gán nhãn và cố gắng tạo ra một mô hình có khả năng phân loại đúng nhãn cho các điểm dữ liệu mới chưa biết nhãn. | SVM, Decision Trees, Random Forest, Logistic Regression, Neural Networks, CNN, RNN |

**Câu 6. Học sâu (DL) là gì? Xác định được các lớp, đầu ra, đầu vào trong một số mô hình mạng nơ-ron nhân tạo, học sâu (DL)? Một số ứng dụng cơ bản trong lĩnh vực ATTT.**

*Trả lời:*

Học sâu (Deep Learning) là một nhánh của học máy (Machine Learning) tập trung vào việc sử dụng các mô hình mạng nơ-ron sâu (Deep Neural Networks) để học từ dữ liệu. Các thuật toán học sâu là các mạng nơ-ron được lập mô hình theo bộ não con người. Ví dụ: một bộ não con người chứa hàng triệu nơ-ron được kết nối với nhau, làm việc cùng nhau để tìm hiểu và xử lý thông tin. Tương tự, các mạng nơ-ron học sâu, hay mạng nơ-ron nhân tạo, được tạo thành từ nhiều lớp nơ-ron nhân tạo hoạt động cùng nhau bên trong máy tính.

Một mạng nơ-ron có các thành phần sau:

* Lớp đầu vào

Một mạng nơ-ron nhân tạo sẽ có một số nút để nhập dữ liệu đầu vào. Các nút này tạo nên lớp đầu vào của hệ thống.

* Lớp ẩn

Lớp đầu vào xử lý và chuyển dữ liệu đến các lớp sâu hơn trong mạng nơ-ron. Các lớp ẩn này xử lý thông tin ở các cấp độ khác nhau, thích ứng với hành vi của mình khi nhận được thông tin mới. Các mạng học sâu có hàng trăm lớp ẩn có thể được dùng để phân tích một vấn đề từ nhiều góc độ khác nhau.

Ví dụ: nếu bạn phải phân loại hình ảnh của một loài vật chưa xác định, bạn sẽ cần so sánh hình ảnh này với các loài vật đã biết. Chẳng hạn, bạn sẽ quan sát hình dáng của cặp mắt, đôi tai, kích thước, số chi và hình mẫu lông của loài vật đó. Bạn sẽ cố gắng xác định các hình mẫu, chẳng hạn như sau:

* Loài vật có móng guốc nên nó có thể là bò hoặc hươu.
* Loài vật có mắt mèo nên nó có thể là một loài mèo hoang dã nào đó.

Các lớp ẩn trong mạng nơ-ron chuyên sâu hoạt động theo cùng một cách. Nếu một thuật toán học sâu đang cố gắng phân loại một hình ảnh động vật, mỗi lớp ẩn của thuật toán này sẽ xử lý một đặc điểm khác nhau của con vật và cố gắng phân loại chính xác nó.

* Lớp đầu ra

Lớp đầu ra bao gồm các nút xuất dữ liệu. Các mô hình học sâu xuất ra đáp án "có" hoặc "không" chỉ có hai nút trong lớp đầu ra. Mặt khác, các mô hình xuất ra nhiều đáp án hơn sẽ có nhiều nút hơn.

Một số ứng dụng của Deep Learning trong lĩnh vực ATTT:

* Phát hiện xâm nhập (Intrusion Detection)

Sử dụng các mạng nơ-ron sâu để phân tích lưu lượng mạng và phát hiện các hành vi xâm nhập bất thường. Các mô hình học sâu có thể phát hiện các mẫu xâm nhập phức tạp mà các hệ thống truyền thống có thể bỏ qua.

Ví dụ: Sử dụng mạng nơ-ron tích chập (CNN) để phân tích các gói dữ liệu hoặc sử dụng mạng nơ-ron hồi quy (RNN) để phân tích chuỗi sự kiện.

* Phát hiện mã độc (Malware Detection)

Học sâu có thể được sử dụng để phát hiện mã độc dựa trên phân tích tệp và hành vi của phần mềm. Các mô hình học sâu có thể phân tích mã nguồn, hành vi thực thi và các dấu hiệu khác để xác định mã độc.

Ví dụ: Sử dụng mô hình học sâu để phân tích mã byte của tệp và phát hiện các đặc trưng của mã độc.

* Phân tích hành vi người dùng (User Behavior Analytics)

Sử dụng học sâu để phân tích hành vi người dùng và phát hiện các hoạt động bất thường hoặc nguy cơ tiềm ẩn. Các mô hình học sâu có thể học các mẫu hành vi bình thường và cảnh báo khi có hành vi bất thường xảy ra.

Ví dụ: Sử dụng mạng nơ-ron hồi quy (RNN) để phân tích chuỗi hành vi người dùng và phát hiện các hoạt động bất thường.

* Phát hiện lừa đảo (Fraud Detection)

Học sâu có thể được sử dụng để phát hiện các hành vi lừa đảo trong các giao dịch tài chính. Các mô hình học sâu có thể học các mẫu hành vi bình thường và phát hiện các hành vi lừa đảo bất thường.

Ví dụ: Sử dụng mạng nơ-ron đa lớp (MLP) để phân tích dữ liệu giao dịch và phát hiện các mẫu lừa đảo.

* Phân tích nhật ký (Log Analysis)

Học sâu có thể được sử dụng để phân tích các tệp nhật ký hệ thống và phát hiện các hoạt động bất thường hoặc dấu hiệu xâm nhập. Các mô hình học sâu có thể học từ dữ liệu lịch sử và phát hiện các mẫu bất thường trong nhật ký.

Ví dụ: Sử dụng mạng nơ-ron tự mã hóa (Autoencoder) để phát hiện các mẫu bất thường trong dữ liệu nhật ký.

Những ứng dụng trên cho thấy tiềm năng của học sâu trong việc nâng cao hiệu quả và độ chính xác của các hệ thống bảo mật, giúp bảo vệ hệ thống và dữ liệu khỏi các mối đe dọa và tấn công.

**Câu 7. Nguyên tắc hoạt động của mạng sinh đối kháng? Một số ứng dụng của mô hình mạng sinh đối kháng (GANs) trong ATTT.**

*Trả lời:*

Mạng sinh đối kháng - hay mạng đối kháng tạo sinh (GAN - Generative Adversarial Network) là một mô hình sinh mẫu trong học máy, được quan tâm rất nhiều trong xu hướng áp dụng trí tuệ nhân tạo vào giải quyết các vấn đề trong đời sống hiện nay, từ nhận diện xử lý ảnh, tới các vấn đề bảo mật, an toàn thông tin cho các hệ thống.

A diagram of a sample

Description automatically generated

**Nguyên tắc hoạt động của GAN**

GANs bao gồm hai mạng nơ-ron đối kháng: mạng sinh (generator) và mạng phân biệt (discriminator). Nguyên tắc hoạt động của GANs dựa trên cuộc cạnh tranh giữa hai mạng này nhằm cải thiện hiệu suất của cả hai

* Mạng sinh (Generator)

Mục đích: học cách tạo ra dữ liệu giả sao cho ngày càng giống dữ liệu thật.

Đầu vào: Một vector ngẫu nhiên z được lấy từ một phân phối xác suất (thường là phân phối chuẩn Gaussian).

Đầu ra: Một mẫu dữ liệu giả G(z).

* Mạng phân biệt (Discriminator)

Mục đích: Phân biệt giữa dữ liệu thật và dữ liệu giả do mạng sinh tạo ra.

Đầu vào: Một mẫu dữ liệu (có thể là thật hoặc giả).

Đầu ra: Xác suất mà mẫu dữ liệu đầu vào là dữ liệu thật.

**Quy trình huấn luyện**

* Khởi tạo: Bắt đầu với các trọng số ngẫu nhiên cho cả mạng sinh và mạng phân biệt.
* Huấn luyện mạng phân biệt

Mạng phân biệt được huấn luyện để phân biệt chính xác giữa dữ liệu thật và dữ liệu giả.

Dữ liệu thật từ tập huấn luyện và dữ liệu giả từ mạng sinh được đưa vào mạng phân biệt.

Mạng phân biệt điều chỉnh trọng số của nó để tối đa hóa xác suất gán nhãn đúng cho dữ liệu thật và dữ liệu giả.

* Huấn luyện mạng sinh

Mạng sinh được huấn luyện để tạo ra dữ liệu giả mà mạng phân biệt không thể phân biệt được với dữ liệu thật.

Đầu vào của mạng sinh là các vector ngẫu nhiên, và đầu ra là dữ liệu giả.

Mục tiêu của mạng sinh là tối thiểu hóa khả năng mạng phân biệt phát hiện dữ liệu giả.

* Lặp lại: Quy trình trên được lặp lại nhiều lần, với mỗi lần huấn luyện cả hai mạng theo hướng cải thiện hiệu suất của mình:

Mạng sinh cố gắng tạo ra dữ liệu giả ngày càng chân thực hơn.

Mạng phân biệt ngày càng trở nên tốt hơn trong việc phân biệt dữ liệu thật và giả.

**Hàm mất mát (Loss Function)**

Hàm mất mát của GANs thường bao gồm hai phần:

* Mất mát của mạng phân biệt (Discriminator Loss): Mạng phân biệt cố gắng tối đa hóa hàm mất mát này.
* Mất mát của mạng sinh (Generator Loss): Mạng sinh cố gắng tối thiểu hóa hàm mất mát này.

**Ứng dụng của GAN**

* Phát hiện và phòng chống mã độc (Malware Detection and Prevention)

GANs có thể được sử dụng để tạo ra các mẫu mã độc giả nhằm huấn luyện các hệ thống phát hiện mã độc. Bằng cách tạo ra các mẫu mã độc mới và chưa biết trước, các mô hình học sâu có thể được cải thiện để nhận diện và phòng chống các loại mã độc mới xuất hiện.

* Tăng cường dữ liệu (Data Augmentation)

Trong nhiều trường hợp, các hệ thống an ninh cần một lượng lớn dữ liệu để huấn luyện các mô hình học sâu. GANs có thể được sử dụng để tạo ra các dữ liệu giả nhưng có tính chất tương tự dữ liệu thật, giúp tăng cường dữ liệu huấn luyện và cải thiện hiệu suất của mô hình.

* Phát hiện xâm nhập (Intrusion Detection)

GANs có thể được sử dụng để tạo ra lưu lượng mạng giả nhằm kiểm tra và nâng cao hiệu quả của các hệ thống phát hiện xâm nhập. Ví dụ: Sử dụng GANs để tạo ra các cuộc tấn công giả nhằm kiểm tra độ nhạy của hệ thống IDS.

* Phát hiện bất thường (Anomaly Detection)

GANs có thể được sử dụng để phát hiện các hành vi bất thường trong hệ thống, giúp xác định các cuộc tấn công hoặc hoạt động đáng ngờ. Mạng sinh có thể tạo ra các mẫu dữ liệu bình thường, trong khi mạng phân biệt được huấn luyện để phát hiện các mẫu dữ liệu bất thường.

Ví dụ: Sử dụng GANs để phát hiện các giao dịch tài chính bất thường hoặc các hoạt động mạng không bình thường.

**Câu 8. Điều gì xảy ra khi kẻ tấn công sử dụng kỹ thuật tấn công đối kháng để đánh lừa mô hình ML? Các biện pháp phòng ngừa là gì?**

*Trả lời:*

Khi sử dụng kỹ thuật tấn công đối kháng (adversarial attack) để đánh lừa mô hình học máy (Machine Learning - ML), kẻ tấn công tạo ra các mẫu đối kháng (adversarial examples) – các mẫu dữ liệu được thiết kế đặc biệt để gây ra lỗi trong mô hình ML. Dưới đây là những điều có thể xảy ra khi mô hình ML bị tấn công đối kháng:

* Giảm độ chính xác của mô hình

Mô hình ML có thể đưa ra dự đoán sai khi gặp phải các mẫu đối kháng. Điều này làm giảm độ chính xác và hiệu suất tổng thể của mô hình.

* Phát hiện sai lệch (Misclassification)

Mẫu đối kháng có thể khiến mô hình phân loại sai một cách đáng kể. Ví dụ, một hình ảnh của một con mèo có thể bị mô hình nhận dạng là một con chó chỉ vì những nhiễu rất nhỏ không thể nhận thấy bằng mắt thường.

* Khai thác lỗ hổng bảo mật

Kẻ tấn công có thể lợi dụng các điểm yếu trong mô hình ML để xâm nhập và khai thác hệ thống. Điều này có thể dẫn đến mất dữ liệu, rò rỉ thông tin nhạy cảm, hoặc phá hoại hệ thống.

**Phòng chống tấn công đối kháng**

* Tăng cường đào tạo với mẫu đối kháng (Adversarial Training)

Huấn luyện mô hình với các mẫu đối kháng để nó học cách chống lại các tấn công. Bằng cách làm như vậy, mô hình trở nên mạnh mẽ hơn trước các mẫu đối kháng.

* Sử dụng các kỹ thuật phát hiện và lọc mẫu đối kháng

Triển khai các hệ thống giám sát và phát hiện các mẫu đối kháng để lọc bỏ chúng trước khi chúng ảnh hưởng đến mô hình.

* Cải tiến kiến trúc mô hình

Sử dụng các mô hình phức tạp hơn và kỹ thuật học sâu tiên tiến để giảm thiểu tác động của các tấn công đối kháng.

* Sử dụng phương pháp phòng vệ dựa trên xác suất (Probabilistic Defense)

Áp dụng các kỹ thuật phòng vệ dựa trên xác suất để làm cho mô hình khó bị tấn công hơn bằng cách làm nhiễu loạn đầu vào hoặc đầu ra của mô hình.

Tấn công đối kháng là một mối đe dọa nghiêm trọng đối với các hệ thống học máy, và việc hiểu rõ về các kỹ thuật tấn công cũng như phát triển các biện pháp phòng chống là rất quan trọng để bảo vệ các ứng dụng học máy khỏi các cuộc tấn công tiềm tàng.

**Câu 9. Học tăng cường là gì? Một số ứng dụng của Reinforcement Learning (RL) trong An toàn thông tin? Xác định được các khái niệm cơ bản, định nghĩa của RL trong các bài toán điển hình của ATTT.**

*Trả lời:*

* 1. **Học tăng cường là gì?**

Học tăng cường (Reinforcement Learning - RL) là một nhánh của học máy tập trung vào việc đào tạo các tác nhân (agents) để đưa ra quyết định bằng cách tương tác với môi trường của chúng. Mục tiêu của tác nhân là học cách hành động để tối đa hóa phần thưởng tích lũy (cumulative reward) theo thời gian.

Thành phần cơ bản của học tăng cường:

* ***Tác nhân (Agent)****:* là thực thể đưa ra các quyết định dựa trên chính sách (policy) của mình. Tác nhân tương tác với môi trường bằng cách chọn các hành động và nhận phản hồi dưới dạng phần thưởng.
* ***Môi trường (Environment)****:* là tất cả mọi thứ mà tác nhân tương tác. Môi trường cung cấp phản hồi cho các hành động của tác nhân dưới dạng phần thưởng và trạng thái mới.
* ***Trạng thái (State)****:* là một biểu diễn của môi trường tại một thời điểm cụ thể. Trạng thái chứa tất cả thông tin cần thiết để tác nhân đưa ra quyết định.
* ***Hành động (Action)****:* là các lựa chọn mà tác nhân có thể thực hiện. Mỗi hành động dẫn đến một trạng thái mới của môi trường và có thể mang lại phần thưởng.
* ***Phần thưởng (Reward)****:* là phản hồi của môi trường đối với một hành động cụ thể của tác nhân. Phần thưởng có thể là dương hoặc âm và được sử dụng để hướng dẫn quá trình học của tác nhân.
* ***Chính sách (Policy)****:* là chiến lược mà tác nhân sử dụng để quyết định hành động nào nên thực hiện trong mỗi trạng thái. Chính sách có thể là xác suất hoặc quyết định cụ thể.
* ***Hàm giá trị (Value Function)****:* đánh giá mức độ tốt của một trạng thái (hoặc trạng thái-hành động). Hàm giá trị ước tính phần thưởng tích lũy mà tác nhân có thể nhận được từ một trạng thái cụ thể.

Các loại thuật toán học tăng cường:

* **RL dựa trên mô hình (Model-based learning)**

RL dựa trên mô hình thường được sử dụng trong trường hợp môi trường được xác định rõ ràng và không thay đổi, đồng thời khó có thể kiểm thử môi trường trong thế giới thực.

**Dyna-Q**: Kết hợp giữa học có mô hình và ngoài mô hình, sử dụng mô hình ước tính để tạo ra các trải nghiệm giả lập nhằm tăng cường quá trình học.

* **RL không mô hình (Model-free learning)**

RL không mô hình là lựa chọn phù hợp nhất để sử dụng cho môi trường lớn, phức tạp và không dễ mô tả. Ngoài ra, RL không mô hình cũng là lựa chọn lý tưởng khi môi trường không xác định và hay thay đổi, đồng thời thử nghiệm dựa trên môi trường không đi kèm với nhược điểm lớn.

**Q-learning**: Một thuật toán học ngoài mô hình phổ biến, sử dụng hàm giá trị hành động (action-value function) để ước tính giá trị của các hành động trong mỗi trạng thái.

**SARSA**: Tương tự như Q-learning, nhưng cập nhật giá trị hành động dựa trên chính sách hiện tại của tác nhân.

* 1. **Một số ứng dụng của Reinforcement Learning (RL) trong An toàn thông tin**
* Phát hiện và phản ứng xâm nhập

RL có thể phát triển hệ thống phát hiện xâm nhập (IDS) để nhận diện lưu lượng mạng bất thường. Hệ thống RL tự động thực hiện các biện pháp đối phó khi phát hiện xâm nhập.

* Tự động hóa bảo mật

- Phát hiện và ngăn chặn mã độc: RL phát triển hệ thống tự động nhận diện và chặn mã độc.

- Tối ưu hóa quy trình bảo mật: RL tối ưu hóa các quy trình như quét lỗ hổng và đánh giá rủi ro.

* Phát hiện và phòng chống tấn công mạng

- Phát hiện tấn công DDoS: RL phát hiện và ngăn chặn các cuộc tấn công DDoS.

- Mô phỏng tấn công mạng: RL mô phỏng các cuộc tấn công để kiểm tra hệ thống bảo mật.

* Bảo vệ quyền riêng tư

- Kiểm soát truy cập thông minh: RL phát triển hệ thống kiểm soát truy cập.

- Bảo vệ dữ liệu cá nhân: RL phát hiện và ngăn chặn xâm phạm quyền riêng tư.

Một ví dụ :bài toán "Taxi" (Taxi-v3) từ OpenAI Gym. Trong bài toán này, tác nhân (agent) phải học cách đón và trả khách trong một lưới 5x5 ô vuông.

* **Tác nhân (Agent)**: chiếc taxi
* **Môi trường (Environment)**: lưới 5x5 ô vuông chứa các vị trí cố định để đón và trả khách.
* **Trạng thái (State)**: trạng thái bao gồm vị trí hiện tại của taxi, vị trí của hành khách và điểm đến của hành khách.
* **Hành động (Action)**: các hành động bao gồm: di chuyển lên, xuống, trái, phải, đón khách, và trả khách.
* **Phần thưởng (Reward)**: phần thưởng là +20 khi hành khách được trả đúng nơi, -10 khi hành động đón hoặc trả khách sai chỗ, và -1 cho mỗi bước đi để khuyến khích tối ưu hóa số bước đi.
* **Chính sách (Policy)**: Là chiến lược mà tác nhân sử dụng để quyết định hành động nào nên thực hiện trong mỗi trạng thái. Chính sách có thể là xác suất hoặc quyết định cụ thể..
* **Hàm giá trị (Value Function)**: Đánh giá mức độ tốt của một trạng thái (hoặc trạng thái-hành động). Hàm giá trị ước tính phần thưởng tích lũy mà tác nhân có thể nhận được từ một trạng thái cụ thể.

**Câu 10. Một số phương pháp phát hiện mã độc sử dụng các thuật toán học máy, học sâu?**

*Trả lời:*

1. **Phương pháp sử dụng Học máy (Machine Learning)**

a. **Phân loại dựa trên các đặc trưng tĩnh (Static Feature-Based Classification)**

* **Chiết xuất đặc trưng (Feature Extraction):** Bao gồm các đặc trưng như chữ ký mã (signature), chuỗi ký tự, API gọi, cấu trúc PE (Portable Executable).
* **Thuật toán phân loại:** Sử dụng các thuật toán như Random Forest, Support Vector Machine (SVM), và K-Nearest Neighbors (KNN).
* **Ưu điểm:** Khả năng phát hiện nhanh và hiệu quả các mẫu mã độc đã biết.
* **Nhược điểm:** Khó phát hiện mã độc mới và mã độc đã bị biến đổi.

b. **Phân loại dựa trên các đặc trưng động (Dynamic Feature-Based Classification)**

* **Chiết xuất đặc trưng động:** Bao gồm hành vi thực thi của mã (execution behavior), các hành động ghi file, truy cập mạng, thay đổi registry.
* **Thuật toán phân loại:** Sử dụng các thuật toán như Hidden Markov Model (HMM), Decision Trees, và Gradient Boosting.
* **Ưu điểm:** Khả năng phát hiện mã độc mới thông qua hành vi bất thường.
* **Nhược điểm:** Quá trình phân tích động tốn thời gian và tài nguyên.

c. **Phương pháp học tập có giám sát (Supervised Learning)**

* **Thuật toán:** Sử dụng các mô hình như Logistic Regression, Naive Bayes, và Neural Networks.
* **Dữ liệu huấn luyện:** Được gán nhãn với các mẫu mã độc và mẫu lành tính.
* **Ưu điểm:** Mô hình học từ dữ liệu có gán nhãn, cho kết quả chính xác cao khi dữ liệu huấn luyện đủ lớn và đa dạng.
* **Nhược điểm:** Đòi hỏi dữ liệu gán nhãn lớn và chất lượng.

2. **Phương pháp sử dụng Học sâu (Deep Learning)**

a. **Mạng nơ-ron tích chập (Convolutional Neural Network - CNN)**

* **Ứng dụng:** Sử dụng để phân tích mã độc dưới dạng hình ảnh hoặc chuỗi nhị phân.
* **Ưu điểm:** Khả năng tự động chiết xuất đặc trưng từ dữ liệu thô, hiệu quả cao với dữ liệu lớn.
* **Nhược điểm:** Yêu cầu tài nguyên tính toán lớn và dữ liệu huấn luyện phong phú.

b. **Mạng nơ-ron hồi quy (Recurrent Neural Network - RNN)**

* **Ứng dụng:** Phân tích hành vi thời gian thực của mã độc.
* **Ưu điểm:** Hiệu quả với dữ liệu chuỗi và hành vi theo thời gian, khả năng học các mẫu tuần tự phức tạp.
* **Nhược điểm:** Khó huấn luyện, cần nhiều dữ liệu và tài nguyên tính toán.

c. **Mô hình học sâu hỗn hợp (Hybrid Deep Learning Models)**

* **Ứng dụng:** Kết hợp các mô hình CNN và RNN để tận dụng ưu điểm của cả hai mô hình.
* **Ưu điểm:** Khả năng phát hiện mã độc với độ chính xác cao bằng cách kết hợp chiết xuất đặc trưng và phân tích hành vi.
* **Nhược điểm:** Phức tạp trong thiết kế và huấn luyện mô hình.

3. **Phương pháp kết hợp (Ensemble Methods)**

* **Ứng dụng:** Sử dụng các kỹ thuật như Bagging, Boosting và Stacking để kết hợp nhiều mô hình học máy hoặc học sâu.
* **Ưu điểm:** Tăng cường độ chính xác và khả năng tổng quát của mô hình.
* **Nhược điểm:** Tốn nhiều tài nguyên và thời gian huấn luyện.

4. **Phát hiện mã độc sử dụng học tăng cường (Reinforcement Learning)**

* **Ứng dụng:** Sử dụng học tăng cường để phát hiện và phản ứng với các cuộc tấn công mã độc theo thời gian thực.
* **Ưu điểm:** Khả năng học và tối ưu hóa hành động phát hiện và ngăn chặn mã độc theo thời gian thực.
* **Nhược điểm:** Khó khăn trong thiết kế và huấn luyện mô hình, yêu cầu dữ liệu và tài nguyên lớn.

**Câu 11. Ảnh hưởng của tốc độ học, batch size và epoch lên quá trình huấn luyện mô hình học máy nói chung?**

*Trả lời:*

Tốc độ học (learning rate), kích thước lô (batch size), và số lần huấn luyện (epochs) là những siêu tham số quan trọng ảnh hưởng trực tiếp đến quá trình huấn luyện và hiệu suất của mô hình học máy. Dưới đây là ảnh hưởng của từng yếu tố này:

1. Tốc độ học (Learning Rate)

**Tốc độ học** xác định kích thước của bước điều chỉnh trọng số trong mỗi lần cập nhật. Nó là yếu tố quan trọng để đảm bảo mô hình học một cách hiệu quả mà không bỏ qua hoặc chệch hướng. Thường sử dụng kỹ thuật giảm tốc độ học theo thời gian.

* **Tốc độ học quá cao:**
  + Mô hình có thể dao động quanh điểm tối ưu hoặc thậm chí không hội tụ.
  + Các bước nhảy quá lớn có thể bỏ qua điểm tối ưu.
* **Tốc độ học quá thấp:**
  + Quá trình huấn luyện sẽ chậm, mất nhiều thời gian để hội tụ.
  + Có thể mắc kẹt tại các điểm cực tiểu cục bộ và không tìm ra điểm tối ưu toàn cục.

2. Kích thước lô (Batch Size)

**Kích thước lô** xác định số lượng mẫu được sử dụng trong một lần cập nhật trọng số.

* **Kích thước lô nhỏ:**
  + Cập nhật trọng số thường xuyên hơn, giúp mô hình hội tụ nhanh.
  + Tuy nhiên, có thể gây ra sự dao động mạnh và dẫn đến quá trình hội tụ không ổn định.
  + Thường đòi hỏi nhiều tài nguyên tính toán hơn.
* **Kích thước lô lớn:**
  + Cập nhật trọng số ít thường xuyên hơn, có thể dẫn đến hội tụ ổn định hơn.
  + Tuy nhiên, quá trình hội tụ có thể chậm và yêu cầu nhiều bộ nhớ.
  + Có thể bỏ qua các chi tiết nhỏ và rơi vào điểm tối ưu cục bộ.

3. Số lần huấn luyện (Epochs)

**Epochs** xác định số lần toàn bộ dữ liệu huấn luyện được đưa vào mô hình. 1 epoch là một lần duyệt qua hết các dữ liệu trong tập huấn luyện.

* **Số epochs quá ít:**
  + Mô hình chưa học đủ từ dữ liệu, dẫn đến underfitting (quá trình học kém).
  + Không đạt được độ chính xác mong muốn.
* **Số epochs quá nhiều:**
  + Mô hình có thể học quá mức (overfitting), nghĩa là mô hình quá khớp với dữ liệu huấn luyện và không tổng quát hóa tốt trên dữ liệu mới.
  + Tốn nhiều thời gian huấn luyện hơn.

**Iterations:** số lượng các Batch size mà mô hình phải duyệt trong 1 epoch.

**Câu 12. Khi xây dựng các ứng dụng dựa trên học máy, cần làm gì khi dữ liệu bị mất cân bằng? Tiêu chí nào có thể dùng để đánh giá mô hình trong trường hợp dữ liệu bị mất cân bằng?**

*Trả lời:*

Mất cân bằng dữ liệu (Imbalanced dataset) là tập dữ liệu có tỷ lệ categories khác nhau, thường là chênh nhau khá xa.

- Lấy mẫu dưới (Under-sampling) lớp đa số: Điều này có nghĩa là chúng ta sẽ giảm số lượng mẫu trong lớp chiếm đa số bằng cách loại bỏ bớt một số điểm dữ liệu, để cho số lượng mẫu của lớp này bằng với số lượng mẫu của lớp chiếm thiểu số.

- Lấy mẫu quá mức (Over-sampling) lớp thiểu số bằng cách nhân bản: Điều này có nghĩa là chúng ta sẽ tăng số lượng mẫu trong lớp chiếm thiểu số bằng cách sao chép một số điểm dữ liệu trong lớp này, để cho số lượng mẫu của nó bằng với số lượng mẫu của lớp chiếm đa số.

- Lấy mẫu quá mức lớp thiểu số bằng kỹ thuật Tổng hợp Mẫu Thiểu số (SMOTE): Điều này có nghĩa là chúng ta sẽ tạo ra các mẫu mới cho lớp chiếm thiểu số bằng cách kết hợp các điểm dữ liệu hiện có, để cho số lượng mẫu của lớp này bằng với số lượng mẫu của lớp chiếm đa số.

**Tiêu chí nào có thể dùng để đánh giá mô hình trong trường hợp dữ liệu bị mất cân bằng**

Khi dữ liệu bị mất cân bằng, các tiêu chí đánh giá truyền thống như độ chính xác (accuracy) có thể không phản ánh đúng hiệu suất của mô hình. Trong các trường hợp này, các tiêu chí sau đây thường được sử dụng để đánh giá mô hình một cách chính xác hơn:

1. Ma trận nhầm lẫn (Confusion Matrix): Ma trận nhầm lẫn cung cấp cái nhìn tổng quan về số lượng dự đoán đúng và sai cho từng lớp. Nó bao gồm các giá trị:

- True Positive (TP): Số mẫu thuộc lớp thiểu số được dự đoán đúng.

- True Negative (TN): Số mẫu thuộc lớp đa số được dự đoán đúng.

- False Positive (FP): Số mẫu thuộc lớp đa số nhưng bị dự đoán nhầm là lớp thiểu số.

- False Negative (FN): Số mẫu thuộc lớp thiểu số nhưng bị dự đoán nhầm là lớp đa số.

2. Độ chính xác (Precision), Độ nhạy (Recall), F1-score, AUC-ROC (Area Under the Receiver Operating Characteristic Curve), AUC-PR (Area Under the Precision-Recall Curve)

\* được miêu tả ở câu 2

**Câu 13. Học liên kết (Federated Learning - FL) là gì? Ưu và nhược điểm của nó? Công thức tổng hợp mô hình trong FL (thuật toán tổng hợp FedAvg)?**

*Trả lời:*

**Khái niệm**

Học liên kết (Federated learning) là một môi trường học máy mà trong đó nhiều thực thể (client) hợp tác để giải quyết một vấn đề học máy, dưới sự điều phối của một máy chủ trung tâm hoặc nhà cung cấp dịch vụ. Dữ liệu thô của mỗi khách hàng được lưu trữ cục bộ và không được trao đổi hoặc chuyển giao; thay vào đó, các khách hàng chỉ gửi các cập nhật mô hình nhỏ gọn đến máy chủ trung tâm, nơi các cập nhật này được tổng hợp ngay lập tức để đạt được mục tiêu học máy.

Client - Các nút tính toán lưu trữ dữ liệu cục bộ, thường là:

- Thiết bị IoT

- Thiết bị di động

- Kho dữ liệu (Data silos)

- Trung tâm dữ liệu ở các khu vực địa lý khác nhau

Server - Các nút tính toán bổ sung điều phối quá trình học liên kết nhưng không truy cập vào dữ liệu thô. Thường không phải là một máy vật lý duy nhất.

**Ưu điểm**

* *Bảo mật và quyền riêng tư:* dữ liệu thô không bao giờ rời khỏi thiết bị của người dùng, giảm thiểu rủi ro vi phạm quyền riêng tư và bảo mật. Người dùng kiểm soát dữ liệu của họ, giảm nguy cơ lộ thông tin nhạy cảm.
* *Giảm băng thông và tài nguyên mạng*: dữ liệu không chuyển giao, chỉ các bản cập nhật mô hình nhỏ được gửi đến máy chủ trung tâm, giảm thiểu lưu lượng mạng so với việc chuyển giao toàn bộ dữ liệu.
* *Tận dụng sức mạnh tính toán phân tán*: sử dụng khả năng tính toán của các thiết bị cục bộ, giảm tải cho các máy chủ trung tâm.
* *Khả năng mở rộng nhiều thiết bị:* dễ dàng mở rộng mô hình học máy với sự tham gia của hàng ngàn thiết bị, không cần hạ tầng máy chủ lớn.
* *Giảm độ trễ:* các thiết bị có thể cập nhật mô hình một cách độc lập và song song, giúp tăng tốc độ huấn luyện.

**Nhược điểm**

* Phức tạp trong triển khai: việc thiết lập và điều phối giữa hàng ngàn thiết bị đòi hỏi hệ thống phức tạp và quản lý tốt.
* Cập nhật không đồng bộ: thiết bị có thể có tốc độ và khả năng kết nối khác nhau, gây khó khăn trong việc đồng bộ hóa các bản cập nhật.
* Tài nguyên cục bộ giới hạn: Không phải tất cả các thiết bị đều có khả năng tính toán mạnh mẽ, ảnh hưởng đến hiệu suất huấn luyện.
* Bảo mật và tính toàn vẹn của mô hình: mô hình có thể bị tấn công nếu một thiết bị bị xâm nhập và gửi các bản cập nhật độc hại.

**Công thức tổng hợp mô hình trong FL (thuật toán tổng hợp FedAvg)**

Thuật toán FedAvg (Federated Averaging) là một phương pháp phổ biến trong học liên kết, được sử dụng để tổng hợp các mô hình cục bộ từ nhiều thiết bị (khách hàng) thành một mô hình chung.

Các bước thực hiện FedAvg:

1. **Khởi tạo mô hình toàn cục**:
   * Khởi tạo mô hình chung trên máy chủ trung tâm với các trọng số ban đầu ​.
2. **Phân phối mô hình**:
   * Máy chủ phân phối mô hình hiện tại ​ đến tất cả các thiết bị tham gia.
3. **Huấn luyện cục bộ**:
   * Mỗi thiết bị k sử dụng dữ liệu cục bộ của mình để huấn luyện mô hình và thu được các trọng số cục bộ ​.
4. **Gửi cập nhật lên máy chủ**:
   * Mỗi thiết bị gửi các trọng số cục bộ ​ của mình về máy chủ trung tâm.
5. **Tổng hợp mô hình**:
   * Máy chủ trung tâm thực hiện tổng hợp các trọng số cục bộ ​ để tạo ra mô hình toàn cục mới .

**Công thức tổng hợp trọng số trong FedAvg:**

Công thức tổng hợp các trọng số trong FedAvg được tính như sau:

A black screen with white text

Description automatically generated

**Câu 14. Xử lý đặc trưng (feature engineering) là gì? Mục đích áp dụng? Có những kỹ thuật nào để thực hiện?**

*Trả lời:*

**Khái niệm**

Feature engineering là quá trình tạo ra và chọn lọc các đặc trưng (features) từ dữ liệu gốc (raw data) để cải thiện hiệu suất của mô hình học máy. Mục tiêu của feature engineering là tối ưu hóa việc biểu diễn dữ liệu và cung cấp những thông tin quan trọng nhất cho mô hình học máy để dự đoán đúng và chính xác.

**Những kỹ thuật trong feature engineering:**

* 1. Trích lọc đặc trưng (Feature Extraction)

Trích lọc đặc trưng là một kĩ thuật giúp giảm chiều giữ liệu mà ở đó cho phép chúng ta lựa chọn hoặc kết hợp các biến đầu vào thành những đặc trưng dự báo nhưng vẫn thể hiện một cách chính xác và nguyên vẹn của dữ liệu gốc.

* Autoencoder: một kỹ thuật trong self-supervised learning, dùng để tự động mã hóa dữ liệu từ không gian cao chiều xuống không gian thấp chiều và giải mã ngược lại sao cho dữ liệu giải mã gần giống với dữ liệu đầu vào ban đầu.
* Bag-of-Words: là thuật toán xử lý ngôn ngữ tự nhiên và trích xuất thông tin, biểu diễn văn bản thành vectơ tần suất từ mà không quan tâm đến thứ tự và cấu trúc ngữ pháp của chúng.
* Image Processing: các thuật toán để phát hiện đặc trưng trên ảnh như hình dạng (shapes) và cạnh (edges), bao gồm các phương pháp trích xuất đặc trưng thủ công và sử dụng mạng nơ-ron tích chập (CNN) để tự động trích xuất đặc trưng từ ảnh.
  1. Biến đổi đặc trưng (Feature Transformation)

Biến đổi đặc trưng là những kĩ thuật giúp biến đổi dữ liệu đầu vào thành những dữ liệu phù hợp với mô hình nghiên cứu.

* Chuẩn hóa biến: Đồng nhất đơn vị của các biến đầu vào và giảm sự khác biệt về độ lớn, sử dụng các kỹ thuật như MinMax Scaling và Standardization.
* Biến đổi biến theo hàm: Biến đổi các biến đầu vào bằng logarithm, căn bậc 2, căn bậc 3 để đạt phương sai ổn định và phân phối gần với chuẩn.
* Tạo biến tương tác: Kết hợp các biến đầu vào thành các biến mới như tích của hai hoặc nhiều biến để mở rộng số lượng biến giải thích.
* Tạo biến bậc cao: Tạo biến mới từ biến đầu vào bằng cách áp dụng các phép tính bậc cao như bình phương, lập phương.
* Dữ liệu về vị trí địa lý: Cung cấp thông tin về vị trí địa lý như thành thị, nông thôn, thu nhập trung bình, giúp mô hình hiểu sâu hơn về bối cảnh địa lý.
* Dữ liệu thời gian: Biến đổi thành các đặc trưng như chu kỳ và mùa vụ (ví dụ: buổi sáng/chiều/tối, ngày/tháng/năm) để cải thiện dự đoán cho các biến mục tiêu có tính chất thời gian.
  1. Lựa chọn đặc trưng (Feature Selection)

Lựa chọn đặc trưng là một phần rất quan trọng trong Machine Learning với mục tiêu chính là loại bỏ những đặc trưng không thực sự chứa thông tin hữu ích cho bài toán phân loại hoặc dự báo.

Các kĩ thuật lựa chọn đặc trưng khá đa dạng: sử dụng hệ số tương quan với biến mục tiêu, sử dụng chỉ số AIC (Akaike Information Criterion - được tính dựa trên giá trị logarith của hàm hợp lý, AIC đánh giá chất lượng mô hình), lựa chọn đặc trưng bằng sử dụng mô hình, ….

**Câu 15. MLOps là gì? Nó giải quyết vấn đề gì trong ngữ cảnh các ứng dụng có sử dụng Học máy? Đánh giá tiềm năng và cơ hội của MLOps trong lĩnh vực An toàn thông tin?**

*Trả lời:*

ML + Dev + Ops = MLOps

MLOps là phương pháp quản lý và tự động hóa các quy trình phát triển, triển khai và vận hành mô hình học máy trong môi trường sản xuất.

MLOps giải quyết các vấn đề sau trong các ứng dụng sử dụng Học máy:

* Quản lý quy trình: Tối ưu hóa và quản lý các quy trình từ phát triển đến triển khai và vận hành mô hình học máy.
* Tự động hóa: Tự động hóa các quy trình để giảm thiểu sự phụ thuộc vào con người và tăng tính nhất quán và tin cậy của sản phẩm.
* Đảm bảo chất lượng: Đảm bảo rằng mô hình học máy hoạt động hiệu quả và đáng tin cậy trong môi trường sản xuất.
* Tích hợp liên tục: Tích hợp liên tục các cập nhật và cải tiến vào mô hình học máy một cách dễ dàng và an toàn.
* Quản lý thay đổi: Quản lý và giám sát các thay đổi và cập nhật mô hình để đảm bảo tính ổn định và hiệu quả của sản phẩm.
* Dễ dàng tích hợp và mở rộng.

MLOps có tiềm năng và cơ hội lớn trong lĩnh vực An toàn thông tin như sau:

* Tăng cường khả năng phát hiện và phòng ngừa: MLOps có thể tích hợp các mô hình học máy vào quy trình phân tích log và giám sát mạng để phát hiện và ngăn chặn các hành vi tấn công một cách nhanh chóng và hiệu quả hơn.
* Tự động hóa các quy trình bảo mật: MLOps cho phép tự động hóa việc triển khai các mô hình phân loại và phát hiện xâm nhập, giúp giảm thiểu thời gian phản ứng và cải thiện tính linh hoạt của hệ thống bảo mật.
* Xử lý dữ liệu lớn và phức tạp: MLOps hỗ trợ xử lý và phân tích các dữ liệu lớn và phức tạp từ các nguồn khác nhau như log hệ thống, dữ liệu mạng, và dữ liệu người dùng để tìm ra các hành vi bất thường và rủi ro tiềm ẩn.
* Phát triển và triển khai nhanh chóng: MLOps cho phép các nhà phát triển và nhà quản lý bảo mật triển khai các mô hình mới và cập nhật mô hình hiện tại một cách nhanh chóng và có độ tin cậy cao hơn.

=> MLOps mang đến cơ hội lớn để cải thiện và nâng cao khả năng phòng ngừa, phát hiện và ứng phó với các mối đe dọa an ninh mạng một cách hiệu quả và linh hoạt trong lĩnh vực An toàn thông tin.

**Câu 16. Reinforcement Learning có thể được áp dụng để tự động hóa các quyết định an ninh mạng như thế nào? Cho ví dụ.**

*Trả lời:*

VD: Để áp dụng Reinforcement Learning (RL) vào việc phát hiện xâm nhập và phân tích hành vi độc hại trong mạng, chúng ta cần xây dựng một hệ thống RL phù hợp và thực hiện các bước sau đây:

1. Xác định môi trường và trạng thái

- Môi trường: Là mạng máy tính hoặc hệ thống mạng cần được bảo vệ. Đây bao gồm các thành phần như thiết bị, ứng dụng, dữ liệu và các luồng lưu lượng mạng.

- Trạng thái (state): Đại diện cho tình trạng hiện tại của mạng. Trạng thái có thể bao gồm thông tin về các luồng lưu lượng mạng, các hoạt động của người dùng, các sự kiện mạng, v.v.

2. Xác định hành động và phần thưởng

- Hành động (action): Là các hành động mà hệ thống RL có thể thực hiện để bảo vệ mạng, chẳng hạn như cảnh báo, cắt kết nối thiết bị, yêu cầu xác thực bổ sung, v.v.

- Phần thưởng (reward): Là phần thưởng được cung cấp cho hệ thống RL dựa trên việc phát hiện và xử lý các hành vi độc hại.

3. Thiết lập mô hình RL

- Agent: Đại diện cho hệ thống RL, quản lý việc ra quyết định và tương tác với môi trường.

- Policy (chiến lược): Là cách mà agent quyết định hành động dựa trên trạng thái hiện tại. Policy có thể được triển khai dưới dạng một mạng nơ-ron hoặc các phương pháp học máy khác.

- Reward function: Xác định cách tính điểm thưởng dựa trên hành động của agent và trạng thái hiện tại của mạng.

4. Huấn luyện và tối ưu hóa:

- Sử dụng dữ liệu từ các cuộc tấn công trước đó hoặc mô phỏng các kịch bản tấn công để huấn luyện mô hình RL. Quá trình này có thể sử dụng các kỹ thuật như Q-learning, Deep Q-networks (DQN), hay các thuật toán RL nâng cao khác.

- Agent sẽ học từ các trải nghiệm và phản hồi, điều chỉnh chiến lược để cải thiện khả năng phát hiện và phản ứng với các hành vi độc hại.

5. Triển khai và đánh giá

- Sau khi huấn luyện, triển khai mô hình RL vào môi trường thực tế để phát hiện và phân tích hành vi độc hại. Các phản hồi từ hệ thống và môi trường sẽ được sử dụng để cải thiện và điều chỉnh mô hình RL. Đánh giá hiệu suất của mô hình bằng cách đo lường khả năng phát hiện, tỷ lệ dương tính giả, và thời gian phản ứng.

6. Tối ưu hóa liên tục và cải thiện học tập

- Cập nhật mô hình RL thường xuyên dựa trên các mối đe dọa mới và các kỹ thuật tấn công tiến bộ để đảm bảo rằng hệ thống luôn làm việc hiệu quả và hiệu quả.

**Câu 17. Qui trình kiểm thử xâm nhập hiện tại có những vấn đề nào mà các phương pháp học máy có thể giúp giải quyết? Trình bày phương pháp ứng dụng Học tăng cường trong bài toán Kiểm thử xâm nhập (Pentest/Network Exploitation)?**

*Trả lời:*

**Qui trình kiểm thử xâm nhập hiện tại có những vấn đề nào mà các phương pháp học máy có thể giúp giải quyết?**

1. Tốn kém thời gian và tài nguyên

- Vấn đề: Quy trình kiểm thử xâm nhập thường đòi hỏi nhiều thời gian và nguồn lực, vì phải thực hiện các phân tích chi tiết và kiểm tra từng thành phần của hệ thống.

- Giải pháp học máy: Học máy có thể tự động hóa một số phần của quy trình kiểm thử, chẳng hạn như quét lỗ hổng bảo mật và phân tích các mẫu hành vi tấn công, từ đó giảm thiểu thời gian và tài nguyên cần thiết.

2. Phụ thuộc vào kỹ năng của người kiểm thử

- Vấn đề: Hiệu quả của kiểm thử xâm nhập phụ thuộc nhiều vào kỹ năng và kinh nghiệm của người kiểm thử. Người kiểm thử phải có kiến thức sâu rộng về các phương pháp tấn công và các lỗ hổng bảo mật.

- Giải pháp học máy: Học máy có thể hỗ trợ người kiểm thử bằng cách cung cấp các gợi ý và dự đoán về các lỗ hổng có khả năng tồn tại, giúp tăng cường hiệu quả kiểm thử mà không đòi hỏi kỹ năng quá cao.

3. Phạm vi kiểm thử hạn chế

- Vấn đề: Kiểm thử xâm nhập thủ công thường chỉ tập trung vào một số phần nhất định của hệ thống, dẫn đến việc bỏ sót các lỗ hổng tiềm ẩn ở những phần khác.

- Giải pháp học máy: Các mô hình học máy có thể được huấn luyện trên các dữ liệu lớn để phát hiện lỗ hổng bảo mật ở nhiều phần khác nhau của hệ thống, từ đó mở rộng phạm vi kiểm thử.

4. Khả năng phát hiện lỗ hổng mới

- Vấn đề: Các phương pháp kiểm thử truyền thống có thể gặp khó khăn trong việc phát hiện các lỗ hổng mới mà chưa từng được biết đến.

- Giải pháp học máy: Học máy có thể học từ các mẫu tấn công và hành vi bất thường để phát hiện các lỗ hổng mới và chưa được biết đến, giúp cải thiện khả năng phát hiện và phòng ngừa tấn công.

5. Xử lý dữ liệu lớn

- Vấn đề: Quá trình kiểm thử xâm nhập có thể tạo ra một lượng lớn dữ liệu cần phải phân tích, gây khó khăn cho việc xử lý và đưa ra kết luận.

- Giải pháp học máy: Các kỹ thuật học máy có khả năng xử lý và phân tích dữ liệu lớn, từ đó giúp nhận diện các mẫu tấn công và lỗ hổng bảo mật một cách hiệu quả hơn.

**Trình bày phương pháp ứng dụng Học tăng cường trong bài toán Kiểm thử xâm nhập (Pentest/Network Exploitation)?**

1. Định nghĩa môi trường và các thành phần

- Môi trường (Environment): Đại diện cho mạng hoặc hệ thống cần được kiểm thử, bao gồm các thành phần như thiết bị, máy chủ, ứng dụng, và các dịch vụ mạng.

- Trạng thái (State): Thông tin về hiện trạng của mạng, chẳng hạn như cấu hình mạng, các dịch vụ đang chạy, các lỗ hổng đã biết, và các dấu hiệu bất thường.

- Hành động (Action): Các thao tác kiểm thử xâm nhập mà agent có thể thực hiện, chẳng hạn như quét cổng, thử nghiệm mật khẩu, khai thác lỗ hổng, cài đặt mã độc, v.v.

- Phần thưởng (Reward): Điểm thưởng dựa trên kết quả của hành động. Ví dụ, phát hiện thành công một lỗ hổng có thể nhận được điểm thưởng cao, trong khi hành động dẫn đến phát hiện và ngăn chặn có thể bị phạt.

2. Xây dựng mô hình RL

- Agent: Đại diện cho hệ thống RL thực hiện kiểm thử xâm nhập. Agent sẽ học cách thực hiện các hành động để tối đa hóa phần thưởng.

- Policy (Chiến lược): Chiến lược mà agent sử dụng để quyết định hành động dựa trên trạng thái hiện tại của môi trường.

- Value Function: Hàm ước lượng giá trị của một trạng thái nhất định, thường được sử dụng để giúp agent chọn hành động tối ưu.

- Model: Đại diện cho các luật và động học của môi trường, có thể được học hoặc được định nghĩa trước.

3. Huấn luyện mô hình RL

- Dữ liệu huấn luyện: Sử dụng dữ liệu từ các kịch bản kiểm thử xâm nhập trước đây hoặc mô phỏng các kịch bản tấn công để huấn luyện mô hình RL.

- Thuật toán RL: Sử dụng các thuật toán RL như Q-learning, Deep Q-Networks (DQN), hoặc các phương pháp tiên tiến hơn như Proximal Policy Optimization (PPO) để huấn luyện agent.

- Học từ phản hồi: Agent sẽ thực hiện các hành động trong môi trường, nhận phản hồi (reward), và cập nhật chiến lược của mình dựa trên các phản hồi này.

4. Triển khai và thử nghiệm

- Mô phỏng: Triển khai mô hình RL trong một môi trường mô phỏng để kiểm tra và điều chỉnh trước khi áp dụng vào môi trường thực.

- Thử nghiệm thực tế: Triển khai mô hình trong môi trường mạng thực tế và giám sát hiệu suất của agent trong việc phát hiện và khai thác lỗ hổng.

5. Tối ưu hóa và cập nhật liên tục

Thu thập phản hồi từ môi trường thực để tiếp tục cải thiện và điều chỉnh mô hình RL.

**Hoặc:**

DeepExploit là một công cụ kiểm thử xâm nhập hoàn toàn tự động sử dụng Học tăng cường (Reinforcement Learning - RL).

Giới thiệu về DeepExploit

DeepExploit là một công cụ kiểm thử xâm nhập (penetration testing) tự động dựa trên Học sâu (Deep Learning) và Học tăng cường. Nó được thiết kế để tự động hóa hoàn toàn quy trình kiểm thử xâm nhập, từ việc phát hiện các lỗ hổng đến khai thác chúng. DeepExploit sử dụng Metasploit Framework như một phần của cơ sở hạ tầng kiểm thử, giúp nó có thể thực hiện một loạt các kỹ thuật tấn công khác nhau.

Các thành phần chính của DeepExploit

* Agent RL: Agent sử dụng thuật toán học tăng cường để học cách tấn công mạng mục tiêu một cách hiệu quả nhất.
* Môi trường: Môi trường là hệ thống hoặc mạng mà DeepExploit sẽ kiểm thử. Môi trường này bao gồm các máy chủ, dịch vụ, và các thành phần mạng khác.
* Metasploit Framework: Công cụ hỗ trợ kiểm thử xâm nhập với một tập hợp lớn các khai thác (exploits) và các module tấn công.

Cách hoạt động của DeepExploit

* Quét lỗ hổng (Scanning)

- DeepExploit bắt đầu bằng việc quét hệ thống mục tiêu để thu thập thông tin về các dịch vụ đang chạy, các cổng mở, và các lỗ hổng tiềm năng.

- Nó sử dụng các công cụ quét như Nmap để thu thập dữ liệu ban đầu.

* Phân tích và xác định mục tiêu (Analysis and Targeting)

- DeepExploit phân tích thông tin thu thập được để xác định các mục tiêu tiềm năng và các lỗ hổng có thể khai thác.

- Agent sử dụng thuật toán học tăng cường để xác định hành động tốt nhất tiếp theo dựa trên trạng thái hiện tại của môi trường.

* Khai thác (Exploitation)

- Agent thực hiện các hành động khai thác dựa trên các quyết định của nó, sử dụng các module khai thác từ Metasploit.

- DeepExploit liên tục theo dõi và ghi nhận kết quả của các hành động để cải thiện chiến lược tấn công.

* Học và cải thiện (Learning and Improvement)

- DeepExploit học từ các kết quả kiểm thử và điều chỉnh chiến lược của mình để trở nên hiệu quả hơn trong các lần kiểm thử sau.

- Sử dụng các kỹ thuật học sâu và học tăng cường, agent liên tục cải thiện khả năng tấn công của mình.

**Câu 18. An ninh mạng thích ứng là gì? Các mô hình học tăng cường có thể được sử dụng để phát triển các hệ thống phòng thủ mạng thích ứng (adaptive cybersecurity) như thế nào?**

*Trả lời:*

*Định nghĩa*

An ninh mạng thích ứng (Adaptive Cybersecurity) là một phương pháp tiếp cận bảo mật mạng linh hoạt và tự điều chỉnh, giúp các hệ thống và tổ chức phản ứng nhanh chóng và hiệu quả với các mối đe dọa bảo mật ngày càng phát triển. Khác với các phương pháp bảo mật truyền thống, an ninh mạng thích ứng không chỉ dựa vào các quy tắc và chữ ký cố định, mà còn sử dụng các kỹ thuật tiên tiến như học máy, phân tích hành vi, và giám sát liên tục để phát hiện, ngăn chặn và phản ứng với các mối đe dọa theo thời gian thực.

Adaptive Security còn được gọi là bảo mật "Zero Trust" nơi không có gì được tin cậy theo mặc định.

*Thành phần*

Adaptive Security là thành phần của 4 lớp sau:

* Phòng ngừa

Phòng bệnh hơn chữa bệnh và lớp trên cùng của mô hình bảo mật Adaptive Security được thiết kế hướng tới mục đích này. Lớp này cô lập mọi sự cố trước khi chúng phát sinh và vạch ra những chính sách, thủ tục và công cụ phòng ngừa để đánh bại trước mọi mối đe dọa tiềm ẩn.

* Phát hiện

Lớp này xác định bất kỳ mối đe dọa nào mà lớp phòng ngừa không phát hiện được. Mục đích chính ở đây là giảm thời gian phản ứng đối với các mối đe dọa tiềm ẩn bằng cách ngăn chặn chúng ngay trên đường đi.

* Xem xét, phân tích

Lớp này đào sâu hơn để tìm bất kỳ mối đe dọa nào bị bỏ lỡ bởi lớp trước. Đây cũng là nơi tiến hành phân tích sự cố chi tiết với sự trợ giúp của các phương pháp phát hiện tiên tiến và phân tích mối đe dọa.

* Dự đoán

Cuối cùng nhưng không kém phần quan trọng, lớp dự đoán theo dõi các sự kiện bên ngoài. Nó cung cấp đánh giá rủi ro kỹ lưỡng và cảnh báo cho nhân viên CNTT về bất kỳ hoạt động đáng ngờ nào. Thông tin được cung cấp bởi lớp này giúp xác định các cuộc tấn công thành công, dự đoán và ngăn chặn những cuộc tấn công tương tự trong tương lai.

**Các mô hình học tăng cường có thể được sử dụng để phát triển các hệ thống phòng thủ mạng thích ứng (adaptive cybersecurity) như thế nào?**

Không biết trả lời :(( chắc làm giống câu 17.

**Câu 19. Nêu một số ưu điểm mà các mô hình ngôn ngữ lớn (Large Language Model – LLM) trong 1 bài toán điển hình của an toàn thông tin?** Gợi ý: phát hiện lỗ hổng phần mềm (vulnerability detection), sửa lỗi bảo mật phần mềm (vulnerability fixing code generation), kiểm thử fuzzing, phát hiện mã độc (malware detection).

*Trả lời:*

*Định nghĩa*

Mô hình ngôn ngữ lớn (Large Language Model - LLM) là một loại thuật toán trí tuệ nhân tạo áp dụng các kỹ thuật mạng nơ-ron với nhiều tham số để xử lý và hiểu ngôn ngữ hoặc văn bản của con người bằng cách sử dụng các kỹ thuật học tự giám sát. Có nhiều kỹ thuật đã được thử nghiệm để thực hiện các nhiệm vụ liên quan đến ngôn ngữ tự nhiên nhưng LLM hoàn toàn dựa trên các phương pháp học sâu. Các nhiệm vụ như tạo văn bản, dịch máy, viết tóm tắt, tạo hình ảnh từ văn bản, lập trình máy, chatbot, hoặc AI hội thoại là các ứng dụng của mô hình ngôn ngữ lớn.

Các ví dụ về mô hình LLM như Chat GPT của OpenAI, BERT (Bidirectional Encoder Representations from Transformers) của Google, v.v.

Ưu điểm

1. Phát hiện lỗ hổng phần mềm (Vulnerability Detection)

* **Hiểu ngữ cảnh mã nguồn**: LLM có khả năng hiểu và phân tích mã nguồn phức tạp, từ đó phát hiện các lỗ hổng bảo mật ẩn trong mã. Điều này giúp cải thiện độ chính xác và giảm thiểu các kết quả dương tính giả.
* **Phân tích tự động**: LLM có thể tự động phân tích mã nguồn từ nhiều ngôn ngữ lập trình khác nhau, giúp phát hiện lỗ hổng trong các dự án lớn mà không cần sự can thiệp thủ công nhiều.
* **Tìm kiếm mẫu lỗ hổng**: LLM có thể tìm kiếm các mẫu lỗ hổng đã biết và áp dụng chúng vào mã nguồn để phát hiện các lỗ hổng tương tự, giúp phát hiện nhanh chóng các vấn đề bảo mật phổ biến.

2. Sửa lỗi bảo mật phần mềm (Vulnerability Fixing Code Generation)

* **Đề xuất sửa lỗi**: LLM có thể tạo ra các bản vá lỗi bảo mật tự động bằng cách đề xuất các đoạn mã sửa lỗi dựa trên phân tích mã nguồn hiện tại và các mẫu sửa lỗi từ trước.
* **Cải thiện hiệu quả sửa lỗi**: Bằng cách tự động hóa việc sửa lỗi bảo mật, LLM giúp giảm thời gian và công sức cần thiết để phát triển các bản vá, đồng thời giúp nhà phát triển tập trung vào các nhiệm vụ quan trọng khác.
* **Học từ dữ liệu thực tế**: LLM có thể học từ dữ liệu sửa lỗi trước đó và áp dụng kiến thức này vào các trường hợp mới, giúp cải thiện chất lượng và hiệu quả của các bản vá lỗi bảo mật.

3. Kiểm thử fuzzing (Fuzz Testing)

* **Tạo dữ liệu đầu vào đa dạng**: LLM có khả năng tạo ra các dữ liệu đầu vào đa dạng và phức tạp cho các ứng dụng, giúp phát hiện các lỗ hổng bảo mật mà các phương pháp fuzzing truyền thống có thể bỏ sót.
* **Hiểu biết ngữ cảnh**: LLM có thể tạo ra các dữ liệu đầu vào dựa trên ngữ cảnh cụ thể của ứng dụng, giúp tăng khả năng phát hiện các lỗ hổng bảo mật liên quan đến dữ liệu đầu vào.
* **Tự động hóa quá trình kiểm thử**: Sử dụng LLM để tự động hóa quá trình kiểm thử fuzzing giúp cải thiện hiệu suất và khả năng phát hiện lỗi bảo mật trong các ứng dụng phức tạp.

4. Phát hiện mã độc (Malware Detection)

* **Phân tích mã độc**: LLM có thể phân tích mã độc phức tạp, hiểu các mẫu hành vi độc hại và phát hiện các đoạn mã đáng ngờ. Điều này giúp cải thiện khả năng phát hiện mã độc mới và chưa biết.
* **Phân tích ngữ cảnh động**: LLM có khả năng phân tích ngữ cảnh động của mã độc, giúp xác định các hành vi độc hại tiềm ẩn dựa trên các mô hình hành vi đã học từ dữ liệu trước đó.
* **Hỗ trợ phản ứng nhanh**: Bằng cách tự động phát hiện và phân loại mã độc, LLM giúp các chuyên gia an ninh mạng phản ứng nhanh chóng và hiệu quả với các mối đe dọa mới.

**Câu 20. Generative AI là gì? Phân tích tiềm năng mang lại cũng như rủi ro có thể gây ra vấn đề mất an toàn thông tin, tấn công mạng, khống chế và điều khiển các hệ thống thông tin.**

*Trả lời:*

Generative AI, viết tắt của "Generative Artificial Intelligence", là một loại hệ thống AI có thể tạo nội dung độc đáo hoặc nguyên bản như văn bản, âm thanh, video hoặc hình ảnh theo yêu cầu. Không giống như một số hệ thống AI truyền thống được thiết kế cho các nhiệm vụ như phân loại hoặc phân tích dữ liệu, các mô hình Generative AI quan tâm nhiều hơn đến việc tạo ra những kết quả đầu ra mới hoặc sáng tạo dựa trên các hướng dẫn mà chúng được cung cấp.

Cách hoạt động của Generative AI

Generative AI thường sử dụng các mô hình học sâu như mạng nơ-ron tích chập (Convolutional Neural Networks - CNNs) và mạng nơ-ron đối nghịch (Generative Adversarial Networks - GANs) để tạo ra nội dung mới. Quá trình hoạt động chung bao gồm các bước sau:

1. Thu thập và chuẩn bị dữ liệu

Thu thập một lượng lớn dữ liệu liên quan đến lĩnh vực cần tạo nội dung (ví dụ: hình ảnh, văn bản, âm thanh). Chuẩn bị và làm sạch dữ liệu để mô hình có thể học hiệu quả.

2. Huấn luyện mô hình

Sử dụng dữ liệu đã chuẩn bị để huấn luyện mô hình AI. Mô hình sẽ học các mẫu và cấu trúc từ dữ liệu đầu vào. Quá trình huấn luyện thường đòi hỏi nhiều tài nguyên tính toán và thời gian.

3. Tạo nội dung mới

Sau khi huấn luyện, mô hình có thể nhận dữ liệu đầu vào mới (ví dụ: một đoạn văn bản hoặc một mô tả) và tạo ra nội dung mới dựa trên những gì nó đã học. Quá trình này có thể bao gồm việc tạo ra văn bản, hình ảnh, âm thanh, hoặc mã lập trình.

4. Điều chỉnh và cải thiện

Nội dung tạo ra có thể được đánh giá và điều chỉnh để cải thiện chất lượng và độ chính xác. Quá trình này có thể bao gồm việc huấn luyện lại mô hình với dữ liệu mới hoặc điều chỉnh các tham số của mô hình.

Generative AI mang lại tiềm năng to lớn trong nhiều lĩnh vực, nhưng cũng đặt ra những rủi ro và thách thức đáng kể về an toàn thông tin và an ninh mạng. Dưới đây là một phân tích chi tiết về cả tiềm năng và rủi ro của Generative AI trong bối cảnh an toàn thông tin, tấn công mạng, và kiểm soát hệ thống thông tin.

**Tiềm năng của Generative AI**

* Cải thiện an ninh mạng

Generative AI có thể phân tích lưu lượng mạng để phát hiện các hoạt động bất thường và cảnh báo sớm về các cuộc tấn công mạng.

* Tăng cường phát hiện lỗ hổng

- Phát hiện lỗ hổng bảo mật: Generative AI có thể tự động quét mã nguồn và hệ thống để tìm ra các lỗ hổng bảo mật tiềm ẩn.

- Kiểm thử fuzzing tự động: Sử dụng AI để tạo ra các dữ liệu kiểm thử phức tạp, giúp phát hiện các lỗ hổng khó tìm ra bằng các phương pháp truyền thống.

Cải thiện chất lượng phần mềm

Generative AI có thể đề xuất và áp dụng các bản vá lỗi bảo mật một cách tự động và nhanh chóng. Ccó thể tối ưu hóa mã nguồn để cải thiện hiệu suất và bảo mật.

**Rủi ro và thách thức**

* Tấn công mạng sử dụng Generative AI:

Generative AI có thể tạo ra các email và tin nhắn giả mạo cực kỳ thuyết phục, đánh lừa người dùng và thu thập thông tin nhạy cảm.

* Lạm dụng Generative AI

Deepfake: Sử dụng AI để tạo ra các video hoặc âm thanh giả mạo, có thể gây ra các cuộc tấn công giả danh hoặc lan truyền thông tin sai lệch.

* Rủi ro về quyền riêng tư và đạo đức

Vi phạm quyền riêng tư: Generative AI có thể tạo ra các nội dung xâm phạm quyền riêng tư của cá nhân hoặc tổ chức. AI có thể được lạm dụng cho các mục đích không đạo đức, chẳng hạn như tạo ra các nội dung phản cảm hoặc thông tin sai lệch.

* Khống chế và điều khiển hệ thống thông tin

- Tấn công vào hệ thống điều khiển công nghiệp (ICS): AI có thể được sử dụng để phát hiện và khai thác lỗ hổng trong các hệ thống điều khiển công nghiệp, gây ra hậu quả nghiêm trọng.

- Tấn công vào cơ sở hạ tầng quan trọng: Các hệ thống cơ sở hạ tầng quan trọng như điện, nước, và giao thông có thể bị tấn công và điều khiển bởi AI, gây ra các thảm họa lớn.