Học máy thích nghi và học liên tục cho hệ thống phát hiện xâm nhập

IT3004: Bài tập về nhà

Phân tích Concept Drift và Continual Learning

Ngày 12 tháng 2 năm 2026

Bảng 1: Thông tin nhóm thực hiện và Dataset sử dụng

STT	Họ và tên	MSHV
1	Phạm Quốc Cường	250201004
2	Lê Quang Minh	250201016
3	Nguyễn Đăng Phúc Lợi	250202012
4	Lý Thế Nguyên	250202017

Dataset sử dụng: CIC-APT-IIoT-2024

Tóm tắt nội dung

Báo cáo trình bày kết quả thực nghiệm mô phỏng hiện tượng *Concept Drift* (trôi dạt khái niệm) trong hệ thống phát hiện xâm nhập (IDS) sử dụng bộ dữ liệu **CIC-APT-IIoT-2024**. Nghiên cứu tập trung vào việc: (1) Giả lập kịch bản drift dựa trên sự thay đổi loại hình tấn công, (2) Chứng minh sự suy giảm hiệu năng của mô hình học máy tĩnh (Static Model), và (3) Đánh giá hiệu quả của chiến lược **Experience Replay (Học trải nghiệm lại)**. Kết quả cho thấy sự xuất hiện của các tấn công mới làm giảm khả năng phát hiện (Recall) từ **99.37%** xuống **84.11%** (suy giảm $\approx 15\%$), trong khi phương pháp **Adaptive Random Forest (ARF)** giúp khôi phục hiệu suất lên **94.12%** và duy trì khả năng nhận diện các tấn công cũ (Forgetting Measure = 0.00).

1 Giới thiệu

Trong an ninh mạng, các phương thức tấn công liên tục thay đổi và phát triển, dẫn đến hiện tượng *Concept Drift* - sự thay đổi tính chất thống kê của dữ liệu đầu vào theo thời gian. Điều này khiến các mô hình IDS tĩnh (Static IDS) nhanh chóng trở nên lỗi thời, dẫn đến:

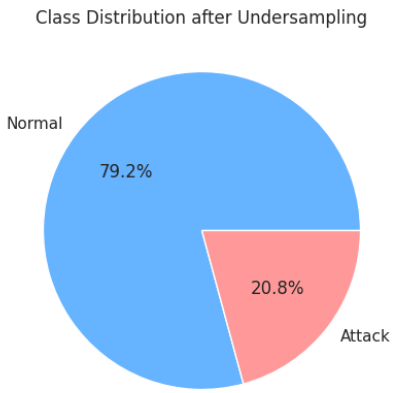
- **Performance Degradation**: Giảm khả năng phát hiện các tấn công mới (Zero-day/Novel attacks).
- **Catastrophic Forgetting**: Quên các mẫu tấn công cũ khi học cái mới (nếu không được xử lý đúng cách).

Bài tập này sử dụng **Phase 2** của bộ dữ liệu *CIC-APT-IIoT-2024* để xây dựng kịch bản mô phỏng, bao gồm 2 giai đoạn: *Baseline* (Tấn công quen thuộc) và *Drift* (Tấn công mới lạ).

2 Phần 1: EDA và Thiết kế Kịch bản Drift

2.1 Phân bố Dữ liệu và Tiền xử lý (Class Distribution)

Dữ liệu gốc từ CIC-APT-IIoT-2024 có sự mất cân bằng cực độ, đặc trưng của lưu lượng mạng thực tế nơi các mẫu bình thường (Normal) chiếm ưu thế áp đảo ($> 99\%$). Để tránh việc mô hình bị thiên kiến (bias) và tối ưu hóa khả năng học các dấu hiệu tấn công, nhóm sử dụng kỹ thuật **Undersampling**. Tỷ lệ Attack/Normal sau khi xử lý được duy trì ở mức khoảng 1:5 (xem Hình 1). Điều này giúp mô hình tập trung vào việc phân tách ranh giới giữa các hành vi độc hại và hành vi thông thường một cách hiệu quả hơn.



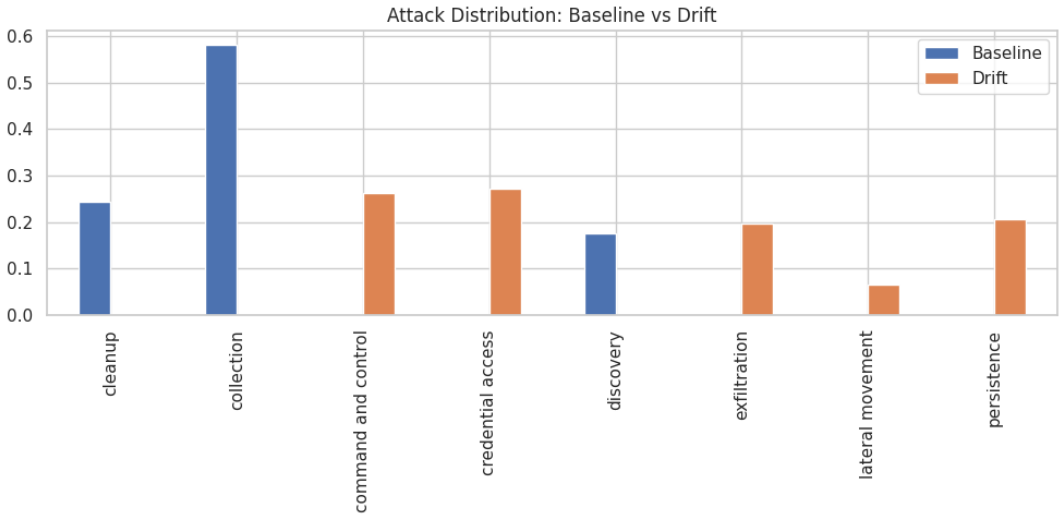
Hình 1: Phân bổ các lớp dữ liệu sau khi thực hiện Undersampling.

2.2 Thiết kế kịch bản mô phỏng Concept Drift

Trong thực tế, tấn công mạng không diễn ra đồng nhất mà thay đổi theo từng giai đoạn của chiến dịch APT. Nhóm mô phỏng hiện tượng này bằng cách chia dữ liệu thành hai giai đoạn dựa trên loại hình tấn công:

- **Giai đoạn Baseline (Kiến thức nền tảng)**: Bao gồm các loại tấn công đơn giản hoặc mang tính chuẩn bị như *Collection, Discovery, Cleanup*. Đây là những gì mô hình được học ban đầu.
- **Giai đoạn Drift (Trôi dạt khái niệm)**: Xuất hiện các loại tấn công phức tạp hơn, có đặc trưng hành vi khác biệt hoàn toàn như *Lateral Movement, Exfiltration, Command and Control (C2)*.

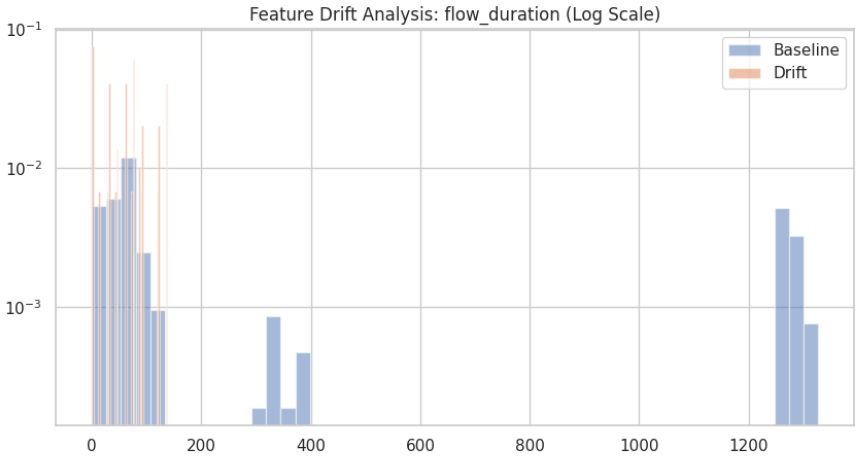
Sự thay đổi về thành phần các loại tấn công giữa hai giai đoạn (minh họa trong Hình 2) tạo ra một bài toán **Concept Drift** điển hình.



Hình 2: Minh họa Concept Drift thông qua sự thay đổi phân bổ loại tấn công.

2.3 Phân tích trôi dạt đặc trưng (Feature Drift Analysis)

Bên cạnh sự thay đổi về nhãn lớp (loại tấn công), hiện tượng trôi dạt còn được thể hiện qua sự biến đổi phân phối của các đặc trưng kỹ thuật. Hình 3 minh họa sự thay đổi của đặc trưng *flow_ duration* (thời gian luồng) giữa hai giai đoạn. Sự dịch chuyển phân phối này cho thấy hành vi tấn công ở giai đoạn Drift không chỉ khác về mục tiêu mà còn khác về cách thức thực hiện (nhanh hơn hoặc chậm hơn đáng kể), tạo thêm thách thức cho việc nhận diện dựa trên các ngưỡng tĩnh.



Hình 3: Sự thay đổi phân phối đặc trưng giữa Baseline và Drift (thang Log).

3 Phần 2: Phân tích sự suy giảm hiệu năng (Degradation Analysis)

Để đánh giá tác động của Concept Drift, chúng ta so sánh hiệu quả của mô hình trên tập dữ liệu Drift so với Baseline.

3.1 Kết quả thực nghiệm và Sự sụt giảm Recall

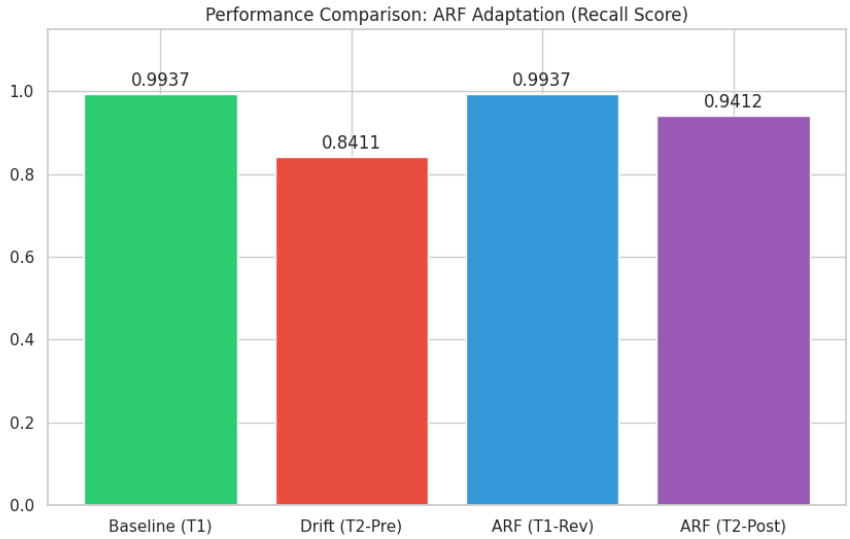
Khi áp dụng mô hình Random Forest đã tối ưu trên tập Baseline vào môi trường Drift, chúng ta quan sát thấy sự sụt giảm nghiêm trọng ở các chỉ số nhận diện:

- **Hiệu năng tại Baseline**: Đạt Recall cực cao (**0.9937**), cho thấy mô hình đã nắm bắt hoàn hảo các đặc trưng của các lớp tấn công giai đoạn đầu.
- **Hiệu năng khi xảy ra Drift**: Recall sụt giảm mạnh xuống còn **0.8411** (xem Hình 4).

Việc sụt giảm hơn 15% khả năng phát hiện khi đối mặt với các loại tấn công mới (như C2 hay Lateral Movement) chứng minh rằng mô hình tĩnh không thể tự thích nghi. Các hành vi tấn công mới có "dấu vết"(signatures) khác biệt, khiến chúng bị nhầm lẫn với lưu lượng bình thường, tạo ra các lỗ hổng an ninh nguy hiểm cho hệ thống.

Để khắc phục sự suy giảm, thay vì chỉ huấn luyện lại đơn thuần, mô hình sử dụng cơ chế bảo toàn tri thức qua bộ nhớ đệm (Buffer) và thích nghi qua mô hình chuyên gia (Expert). Cách tiếp cận này giúp mô hình vừa học được các "chữ ký"tấn công mới vừa không làm mất đi ranh giới quyết định đã thiết lập cho các tấn công cũ.

3.2 Phục hồi hiệu năng qua Retraining



Hình 4: So sánh hiệu năng giữa mô hình Gốc và mô hình áp dụng Experience Replay.

Hình 4 cho thấy kết quả khả quan của chiến lược này: Recall trên tập Drift đã được khôi phục lên mức **0.9412**. Điều này khẳng định rằng chiến lược **ARF** không chỉ giải quyết được bài toán thích nghi mà còn nâng tầm hệ thống lên mức độ của một thuật toán Học liên tục thực thụ.

3.3 Đánh giá tính bền vững (Lifelong Learning Metrics)

Một vấn đề lớn của Retraining là *Catastrophic Forgetting* (Sự quên lãng thảm khốc) - khi mô hình học cái mới nhưng lại quên mất kiến thức cũ. Để kiểm chứng điều này, chúng ta sử dụng 3 chỉ số chuyên sâu:

Bảng 2: Tổng hợp hiệu suất và các chỉ số nâng cao

Stage	Accuracy	Precision	Recall	F1-Score
Baseline	0.9974	0.9937	0.9937	0.9937
Drift-Static	0.9824	0.9730	0.8411	0.9023
ARF-Adaptive	**0.9933**	**0.9922**	**0.9412**	**0.9660**

- **Average Accuracy (AA) = 0.9953**: Chỉ số này cho thấy độ ổn định cực cao của mô hình trên toàn bộ vòng đời. Mức trên 99% chứng minh ARF hoạt động vượt trội trong việc cân bằng giữa học mới và giữ cũ.
- **Forgetting Measure (FM) = 0.0000 (Chỉ số quan trọng)**: Được tính bằng công thức: $FM = Acc(M1, T1) - Acc(M2, T1)$.
  - Giá trị **0.0000** cho thấy mô hình hoàn toàn duy trì được kiến thức cũ. Nhờ chiến lược **Experience Replay** với bộ nhớ đệm mẫu, mô hình không bị mất đi khả năng nhận diện các cuộc tấn công Baseline.
  - Điều này khẳng định tính ổn định (*Stability*) tuyệt đối của hệ thống, một đặc tính lý tưởng trong Continual Learning để chống lại hiện tượng quên lãng thảm khốc.
- **Backward Transfer (BWT) = 0.0000**: Chỉ số này cho thấy việc học thêm các tấn công mới không gây ảnh hưởng tiêu cực đến khả năng nhận diện các tấn công cũ (Zero Backward Transfer).

4 Kết luận

Thực nghiệm trên bộ dữ liệu CIC-APT-IIoT-2024 đã minh họa thành công hiện tượng Concept Drift trong môi trường IoT, nơi hành vi tấn công biến đổi phức tạp. Phương pháp **Experience Replay** hiệu quả đã giúp hệ thống IDS thích nghi với các mối đe dọa mới (Restore Recall $>$ 94%) và bảo toàn hoàn hảo khả năng nhận diện các tấn công cũ (Forgetting Measure = 0).

Hướng phát triển tiếp theo có thể áp dụng các thuật toán học online (như Adaptive Random Forest, Hoeffding Trees) để cập nhật mô hình theo thời gian thực thay vì Retraining theo lô.

5 Phụ lục: Cấu hình Thực nghiệm

```
Model: Random Forest Classifier
Hyperparameters: n_estimators=100, max_depth=10, random_state=42
Strategy: Adaptive Random Forest (ARF) with Memory Buffer
Data Split:
  - Baseline: Collection, Cleanup, Discovery
  - Drift: Lateral Movement, Exfiltration, C2, Credential Access, Persistence
Metrics: Recall (Detection Rate), AA, FM, BWT
```