VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY

HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY

FACULTY OF COMPUTER SCIENCE AND ENGINEERING

BÁO CÁO
ĐỒ ÁN CHUYÊN NGÀNH

NÂNG CAO PHÁT HIỆN TẤN CÔNG TRÊN HỆ THỐNG BẢO MẬT ỨNG DỤNG WEB DỰA VÀO PHƯƠNG PHÁP HỌC MÁY

CHUYÊN NGÀNH: KHOA HỌC MÁY TÍNH

HỘI ĐỒNG : HỘI ĐỒNG 15L ĐỒ ÁN CHUYÊN NGÀNH

GVHD : TS. NGUYỄN LÊ DUY LAI

————o0o————

SINH VIÊN : DƯƠNG THUẬN ĐÔNG - 2210762

HO CHI MINH CITY, DECEMBER 2025

Lời cam đoan

Tôi xin cam đoan đồ án này là công trình nghiên cứu của cá nhân tôi, được thực hiện dưới sự hướng dẫn của TS. Nguyễn Lê Duy Lai.

Các kết quả trình bày trong báo cáo đảm bảo tính trung thực và chưa từng được công bố trong bất kỳ ấn phẩm nào trước đây. Mọi tài liệu và dữ liệu phục vụ nghiên cứu được thu thập từ nhiều nguồn khác nhau và đã được liệt kê chi tiết trong danh mục tài liệu tham khảo.

Đối với các nội dung kế thừa từ kết quả nghiên cứu của các tác giả hoặc tổ chức khác, tôi đã thực hiện trích dẫn nguồn gốc rõ ràng theo đúng quy định.

Tôi xin chịu hoàn toàn trách nhiệm về tính xác thực của đồ án này trước Hội đồng. Trường Đại học Bách Khoa không chịu trách nhiệm đối với các vi phạm về tác quyền (nếu có) phát sinh từ nội dung của nghiên cứu này.

Lời cảm ơn

Lời đầu tiên, tôi xin bày tỏ lòng biết ơn chân thành và sâu sắc nhất đến TS. Nguyễn Lê Duy Lai, người đã tận tình hướng dẫn, định hướng nghiên cứu và truyền đạt những kinh nghiệm quý báu trong suốt quá trình thực hiện đồ án. Sự hỗ trợ nhiệt tình của Thầy là động lực to lớn giúp tôi hoàn thành công việc đúng tiến độ.

Bên cạnh đó, tôi xin gửi lời tri ân đến các thầy cô thuộc Khoa Khoa học và Kỹ thuật Máy tính, Trường Đại học Bách Khoa – ĐHQG TP.HCM. Những kiến thức nền tảng và chuyên sâu nhận được trong suốt những năm tháng học tập chính là hành trang vững chắc để tôi có thể hoàn thành đề tài này.

Mặc dù đã nỗ lực hết mình, nhưng do kiến thức và thời gian có hạn nên bài báo cáo khó tránh khỏi những thiếu sót. Tôi rất mong nhận được những ý kiến đóng góp và chỉ dẫn từ quý Thầy Cô để đồ án được hoàn thiện hơn.

Xin chân thành cảm ơn!

Tóm tắt

Tấn công giả mạo (phishing) hiện là một trong những mối đe dọa an ninh mạng nghiêm trọng nhất, gây thiệt hại lớn về tài chính và dữ liệu cá nhân. Đồ án đề xuất giải pháp xây dựng và triển khai hệ thống phát hiện website độc hại toàn diện, tự động hóa dựa trên phương pháp học máy.

Hệ thống triển khai theo kiến trúc microservices với các thành phần độc lập, đảm bảo tính linh hoạt và khả năng mở rộng. Quy trình thu thập dữ liệu vận hành thông qua các crawler chuyên biệt, thực hiện trích xuất đa dạng các nhóm đặc trưng từ URL đầu vào. Dữ liệu thô sau đó được chuẩn hóa và chuyển đổi qua đường ống ETL (Extract, Transform, Load) để phục vụ huấn luyện mô hình.

Thực nghiệm cho thấy mô hình đạt hiệu năng ấn tượng với độ chính xác (Accuracy) lên đến 98% và chỉ số F1-score 0.93, khẳng định khả năng phân loại hiệu quả giữa URL giả mạo và hợp pháp. Sản phẩm cuối cùng là một tiện ích mở rộng trình duyệt (Browser Extension), hỗ trợ người dùng kiểm tra độ an toàn của trang web theo thời gian thực.

Toàn bộ quy trình từ tiếp nhận URL đến đưa ra dự đoán đều được tự động hóa hoàn toàn, cung cấp một công cụ bảo vệ mạnh mẽ, góp phần nâng cao an toàn cho người dùng trên không gian mạng.

Mục lục

Danh mục bảng biểu

Chương 1

Giới thiệu

1.1 Bối cảnh và lý do chọn đề tài

1.1.1 Bối cảnh

Trong kỷ nguyên số, Internet đã trở thành một phần không thể thiếu trong đời sống mỗi cá nhân. Tuy nhiên, song song với lợi ích to lớn, môi trường Internet cũng tồn tại nhiều rủi ro an ninh, trong đó Phishing nổi lên như một trong những mối đe dọa nghiêm trọng và phổ biến nhất. Phishing là hành vi giả mạo trang web, email, tin nhắn, cuộc gọi thoại nhằm lừa người dùng tiết lộ các thông tin nhạy cảm. Các cuộc tấn công này ngày càng trở nên tinh vi, gây ra những hậu quả nặng nề, điển hình là:

- **Đối với cá nhân:** Thiệt hại về tài chính, bị đánh cắp thông tin và mất quyền kiểm soát tài khoản.
- **Đối với tổ chức, doanh nghiệp:** Rò rỉ dữ liệu nội bộ, mất uy tín thương hiệu, gián đoạn hoạt động kinh doanh và đối mặt với các vấn đề pháp lý.

Các phương pháp phòng chống truyền thống như sử dụng danh sách đen (black-list) hoặc bộ lọc dựa trên quy tắc (rule-based) dần bộc lộ nhiều hạn chế. Những giải pháp này mang tính bị động, dễ dàng bị vượt mặt bởi các tên miền mới đăng

ký liên tục. Do đó, việc phát triển một công cụ phát hiện thông minh, có khả năng thích ứng cao là yêu cầu cấp thiết.

1.1.2 Lý do chọn đề tài

Để giải quyết các thách thức trên, đề tài "Nâng cao phát hiện tấn công trên hệ thống bảo mật ứng dụng web dựa vào phương pháp học máy" được lựa chọn với các lý do chính sau:

- **Khắc phục nhược điểm của phương pháp truyền thống:** Học máy (Machine Learning) cho phép nhận diện các mẫu hình ẩn (hidden patterns) mà con người khó phát hiện thủ công. Giải pháp này có thể chủ động phân loại website ngay khi chúng vừa xuất hiện mà không cần phụ thuộc vào các danh sách đen có sẵn.

- **Tiếp cận đa đặc trưng:** Thay vì chỉ dựa vào cấu trúc URL đơn thuần vốn dễ bị làm giả, đề tài khai thác sâu các đặc trưng hạ tầng như mạng, chứng chỉ số, bản ghi DNS và hành vi chuyển hướng HTTP. Cách tiếp cận này giúp tăng cường độ tin cậy trong việc nhận diện các cuộc tấn công tinh vi sử dụng HTTPS hợp lệ.

- **Khả năng ứng dụng thực tế:** Đồ án không chỉ dừng lại ở nghiên cứu lý thuyết mà hiện thực hóa thành một giải pháp hoàn chỉnh, từ quy trình xử lý dữ liệu tự động đến tiện ích mở rộng trên trình duyệt (Browser Extension) hỗ trợ người dùng cuối.

Với những lý do trên, đề tài này hứa hẹn sẽ mang lại một giải pháp hiệu quả, linh

hoạt và toàn diện trong cuộc chiến chống lại các cuộc tấn công phishing, góp phần bảo vệ người dùng và doanh nghiệp trên không gian mạng.

1.2 Mục tiêu nghiên cứu

1.2.1 Mục tiêu tổng quát:

Xây dựng và đánh giá một hệ thống toàn diện, hiệu quả để phát hiện các trang web lừa đảo trong thời gian thực, sử dụng phương pháp tiếp cận dựa trên học máy và kiến trúc microservices để thu thập và xử lý đa dạng các loại đặc trưng.

1.2.2 Mục tiêu cụ thể:

- **Triển khai hạ tầng thu thập phân tán:** Vận hành các crawlers độc lập dưới dạng microservices để tự động trích xuất dữ liệu từ URL.
- **Xây dựng bộ dữ liệu chuẩn:** Tổng hợp và gán nhãn dữ liệu quy mô lớn từ các nguồn uy tín như PhishTank và Majestic Million.
- **Tối ưu hóa quy trình ETL:** Xây dựng đường ống xử lý dữ liệu tự động, đảm bảo tính nhất quán và sẵn sàng cho giai đoạn huấn luyện.
- **Huấn luyện và thực nghiệm mô hình:** So sánh hiệu năng giữa Random Forest và LightGBM dựa trên các độ đo tiêu chuẩn.
- **Hoàn thiện sản phẩm đầu-cuối:** Tích hợp mô hình vào backend và phát triển tiện ích trình duyệt để cung cấp cảnh báo thời gian thực cho người dùng.

1.3 Phạm vi nghiên cứu

Phạm vi của nghiên cứu này được xác định rõ ràng để tập trung nguồn lực vào các mục tiêu đã đề ra, đảm bảo tính khả thi và chiều sâu của dự án.

1.3.1 Phạm vi bao gồm (In-Scope):

- **Đối tượng nghiên cứu:** Nghiên cứu tập trung vào việc phát hiện các trang web lừa đảo có thể truy cập công khai thông qua một URL.
- **Đặc trưng được sử dụng:** Quá trình phát hiện sẽ dựa trên việc phân tích và tổng hợp một tập hợp đa dạng các đặc trưng có thể thu thập được một cách tự động, bao gồm: Đặc trưng từ vựng của URL (URL Lexical), Đặc trưng mạng (Network), Đặc trưng chứng chỉ số (Certificate), Đặc trưng chuyển hướng HTTP (HTTP Redirection), Đặc trưng tiêu đề HTTP (HTTP Header), Đặc trưng hệ thống phân giải tên miền (DNS).
- **Phương pháp luận:** Nghiên cứu áp dụng phương pháp học máy có giám sát (Supervised Machine Learning) để xây dựng mô hình phân loại. Phạm vi bao gồm việc so sánh và đánh giá hiệu quả của một số thuật toán phổ biến để tìm ra mô hình tối ưu.
- **Sản phẩm cuối cùng:** Kết quả của nghiên cứu là một mô hình học máy đã được huấn luyện, hệ thống backend xử lý và một tiện ích mở rộng trình duyệt có khả năng cảnh báo người dùng khi truy cập trang web lừa đảo.

1.3.2 Phạm vi không bao gồm (Out-of-Scope):

- **Phân tích nội dung trang:** Mặc dù việc phân tích nội dung trực quan và cấu trúc DOM của trang web là một phương pháp mang lại hiệu quả cực kỳ cao trong nhận diện phishing, tuy nhiên kỹ thuật này đòi hỏi tài nguyên lớn và thời gian xử lý kéo dài. Do những hạn chế về tài nguyên tính toán hiện tại và yêu cầu tối ưu hóa tốc độ xử lý thời gian thực, nghiên cứu này chưa triển khai phần phân tích nội dung trang.
- **Các hình thức tấn công khác:** Nghiên cứu tập trung chuyên sâu vào bài toán phát hiện website giả mạo dựa trên các đặc trưng hạ tầng và định danh. Do đó, phạm vi nghiên cứu không mở rộng sang các hình thức lừa

đảo qua tin nhắn SMS (Smishing), lừa đảo qua giọng nói (Vishing), hoặc các kỹ thuật tấn công khai thác lỗ hổng trực tiếp trên ứng dụng web như SQL Injection (SQLi), Cross-Site Scripting (XSS), hay Cross-Site Request Forgery (CSRF).

- **Chức năng ngăn chặn thời gian thực:** Hệ thống được xây dựng với mục tiêu phát hiện và đưa ra cảnh báo, không bao gồm chức năng ngăn chặn truy cập của người dùng trong thời gian thực.

1.4 Phương pháp tiếp cận

Nghiên cứu này được xây dựng và phát triển dựa trên nền tảng lý thuyết và phương pháp luận được đề xuất trong luận văn "Effective Phishing Detection using Machine Learning Approach" của Yaokai Yang[6]. Đồ án kế thừa kiến trúc hệ thống, bộ đặc trưng và quy trình thực nghiệm từ nghiên cứu gốc, đồng thời thực hiện các tinh chỉnh và cải tiến quan trọng về mặt kiến trúc phân tán, xử lý dữ liệu mất cân bằng, tối ưu hóa thuật toán và triển khai thực tế. Các cải tiến chi tiết được trình bày tại Chương 4 (Mục 4.2).

Để hiện thực hóa các mục tiêu, dự án áp dụng phương pháp tiếp cận dựa trên công nghệ container hóa và kiến trúc hướng dịch vụ:

- **Kiến trúc Microservices:** Sử dụng Docker và Docker Compose để module hóa các thành phần thu thập đặc trưng, giúp hệ thống vận hành song song và dễ dàng mở rộng.

- **Công nghệ lưu trữ và xử lý:** Kết hợp MongoDB cho dữ liệu thô và PostgreSQL cho dữ liệu đã qua xử lý ETL, tối ưu hóa tốc độ truy vấn và huấn luyện.

- **Chiến lược học máy:** Tập trung vào các thuật toán dạng cây (Tree-based models) như Random Forest và LightGBM, kết hợp với kỹ thuật SMOTE để xử lý vấn đề mất cân bằng dữ liệu đặc thù trong an ninh mạng.

- **Triển khai thực tế:** Xây dựng API backend để phục vụ dự đoán và tích hợp trực tiếp vào trình duyệt qua Browser Extension, đảm bảo quy trình từ thu thập đến cảnh báo diễn ra khép kín.

1.5 Cấu trúc báo cáo

Nội dung của báo cáo được tổ chức thành 6 chương chính như sau:

- **Chương 1: Giới thiệu** – Trình bày tính cấp thiết của vấn đề tấn công phishing, lý do lựa chọn đề tài, xác định mục tiêu nghiên cứu, phạm vi thực hiện và phương pháp tiếp cận tổng thể của dự án.
- **Chương 2: Tổng quan về bảo mật ứng dụng web và học máy** – Giới thiệu các khái niệm cơ bản về hệ thống IDS/IPS, phân tích cơ chế của các loại tấn công Web phổ biến và sơ lược các kỹ thuật học máy, học sâu cũng như các bộ dữ liệu chuẩn trong an ninh mạng.
- **Chương 3: Phân tích và đánh giá các phương pháp hiện tại** – Đi sâu vào phân tích các mô hình học máy truyền thống và học sâu hiện đại, tổng quan các nghiên cứu mới nhất và đánh giá những hạn chế còn tồn tại của các phương pháp này.
- **Chương 4: Đề xuất mô hình cải tiến** – Mô tả chi tiết kiến trúc hệ thống, quy trình tiền xử lý dữ liệu và các thuật toán được sử dụng cùng các kỹ thuật tối ưu hóa.
- **Chương 5: Thực nghiệm và đánh giá** – Thiết lập môi trường thử nghiệm, mô tả tập dữ liệu, trình bày kết quả so sánh hiệu năng giữa các mô hình thông qua các độ đo Precision, Recall, F1-Score, AUC và phân tích hiệu suất thực tế.
- **Chương 6: Kết luận và hướng phát triển** – Tổng kết các kết quả đã đạt được, đóng góp của nghiên cứu về mặt thực tiễn và kỹ thuật, đồng thời nhìn nhận các hạn chế và đề xuất các hướng nghiên cứu mở rộng trong tương lai.

Chương 2

Tổng quan về bảo mật ứng dụng web và học máy

2.1 Từ IDS/IPS đến bảo mật ứng dụng web hiện đại

Để xác định cơ sở lý thuyết cho đề tài, chương này phân tích sự chuyển dịch từ các mô hình bảo mật mạng tổng quát (IDS/IPS) sang các giải pháp chuyên biệt cho tầng ứng dụng (WAF), đồng thời làm rõ vai trò của học máy trong việc nhận diện các hành vi lừa đảo hiện đại.

2.1.1 Hệ thống phát hiện và ngăn chặn xâm nhập (IDS/IPS)

Hệ thống Phát hiện Xâm nhập (IDS) và **Hệ thống Ngăn chặn Xâm nhập (IPS)** đóng vai trò là lớp rào chắn cơ bản trong hạ tầng an ninh mạng.

- **IDS (Intrusion Detection System):** Thực hiện giám sát lưu lượng mạng nhằm phát hiện các dấu hiệu bất thường. Công cụ này vận hành theo cơ chế thụ động, tập trung vào việc ghi nhật ký và gửi cảnh báo tới quản trị viên.
- **IPS (Intrusion Prevention System):** Là phiên bản nâng cao với khả năng

can thiệp chủ động, cho phép ngăn chặn các gói tin độc hại hoặc ngắt kết nối ngay khi nhận diện nguy cơ.

Các giải pháp này thường vận hành dựa trên hai nguyên lý cốt lõi:

- *Dựa trên chữ ký (Signature-based):* So sánh với cơ sở dữ liệu các mẫu tấn công đã biết.
- *Dựa trên bất thường (Anomaly-based):* Phát hiện các hành vi lệch chuẩn so với trạng thái bình thường.

Tuy nhiên, các mô hình dựa trên hạ tầng mạng (Network-based) thường bộc lộ hạn chế khi đối phó với các mối đe dọa tại tầng ứng dụng (Layer 7), đặc biệt trong bối cảnh lưu lượng được mã hóa qua giao thức HTTPS.

2.1.2 Tường lửa ứng dụng web (Web Application Firewall - WAF)

Khi các hành vi xâm nhập chuyển dịch trọng tâm sang tầng ứng dụng, **Web Application Firewall (WAF)** trở thành thành phần bảo mật thiết yếu. Khác với các giải pháp mạng thông thường, WAF có khả năng bóc tách giao thức HTTP/HTTPS và phân tích sâu nội dung của các yêu cầu (requests). WAF thường được sử dụng để bảo vệ máy chủ trước các lỗ hổng như SQL Injection hay XSS. Tuy nhiên, việc phụ thuộc vào các tập luật (ruleset) tĩnh dẫn đến hai thách thức lớn:

1. Khả năng nhận diện hạn chế đối với các biến thể tấn công mới (Zero-day) chưa được cập nhật trong cơ sở dữ liệu luật.
2. Sự kém hiệu quả trong việc đối phó với Phishing, nơi các yêu cầu HTTP hoàn toàn hợp lệ về mặt cú pháp nhưng lại dẫn người dùng tới các địa chỉ giả mạo.

2.1.3 Phương pháp tiếp cận của dự án: Học máy trong phát hiện Phishing

Trong khuôn khổ nghiên cứu này, kỹ thuật **Học máy (Machine Learning)** được ứng dụng nhằm khắc phục các "điểm mù" của bộ lọc tĩnh. Thay vì đối soát URL với danh sách đen (Blacklist) cố định, giải pháp tập trung vào việc trích xuất và phân tích các đặc trưng động như cấu trúc từ vựng URL, thông tin chứng chỉ SSL, các bản ghi DNS, v.v. Về mặt lý thuyết, đây là phương pháp tiếp cận chủ động (Proactive), tương đồng với cơ chế *Anomaly-based Detection*, cho phép nhận diện các website độc hại ngay khi chúng vừa được khởi tạo, vượt xa khả năng của các danh sách đen truyền thống.

2.2 Các loại tấn công phổ biến trên ứng dụng web

Để xây dựng hệ thống bảo mật hiệu quả, việc hiểu rõ cơ chế hoạt động của các phương thức tấn công là điều kiện tiên quyết. Dưới đây là tóm tắt các kỹ thuật phổ biến nhất nhắm vào ứng dụng web.

2.2.1 SQL Injection (SQLi)

Kẻ tấn công chèn các câu lệnh SQL độc hại vào các trường nhập liệu để thao túng câu truy vấn cơ sở dữ liệu. Cơ chế này cho phép vượt qua xác thực, truy cập trái phép hoặc đánh cắp dữ liệu nhạy cảm từ máy chủ.

2.2.2 Cross-Site Scripting (XSS)

XSS xảy ra khi mã kịch bản độc hại (thường là JavaScript) được chèn vào trang web và thực thi trên trình duyệt của người dùng khác. Tấn công này thường dùng để đánh cắp cookie, phiên làm việc hoặc điều hướng người dùng sang trang web giả mạo.

2.2.3 Cross-Site Request Forgery (CSRF)

CSRF lừa trình duyệt của nạn nhân thực hiện các hành động không mong muốn trên một ứng dụng web mà họ đã đăng nhập. Tấn công này lợi dụng sự tin tưởng của ứng dụng vào phiên làm việc hiện tại của người dùng để thực hiện các giao dịch trái phép.

2.2.4 Lỗ hổng File Inclusion (LFI và RFI)

Xảy ra khi ứng dụng nạp các tệp tin mà không kiểm duyệt kỹ. **LFI** cho phép đọc tệp cục bộ trên máy chủ, trong khi **RFI** cho phép nạp và thực thi mã từ một máy chủ từ xa, dẫn đến nguy cơ chiếm quyền kiểm soát hệ thống.

2.2.5 Tấn công Brute Force

Đây là phương pháp thử sai liên tục các tổ hợp ký tự hoặc mật khẩu phổ biến để chiếm đoạt tài khoản. Với sức mạnh tính toán hiện nay, các mật khẩu đơn giản có thể bị bẻ khóa rất nhanh chóng.

Mối liên hệ với Phishing: Phishing thường là bước khởi đầu trong chuỗi tấn công để đánh cắp thông tin xác thực. Việc phát hiện Phishing ngay từ lớp URL đóng vai trò là chốt chặn quan trọng để ngăn chặn các bước khai thác kỹ thuật sâu hơn như chiếm đoạt tài khoản hay thực thi mã độc.

2.3 Các kỹ thuật học máy trong phát hiện tấn công

Việc tích hợp Học máy cho phép hệ thống tự động học các mẫu hành vi từ dữ liệu, từ đó phát hiện các biến thể tấn công mới mà không cần cập nhật thủ công. Trong bảo mật ứng dụng web, các kỹ thuật này được chia thành ba nhóm chính (chi tiết về các thuật toán cụ thể được phân tích tại Chương 3).

2.3.1 Học có giám sát (Supervised Learning)

Mô hình được huấn luyện trên tập dữ liệu đã gán nhãn (ví dụ: URL lừa đảo và URL hợp pháp). Đây là phương pháp chủ đạo được sử dụng trong đồ án này để phân loại website dựa trên các đặc trưng hạ tầng.

2.3.2 Học không giám sát (Unsupervised Learning)

Xử lý dữ liệu chưa gán nhãn nhằm tìm ra các cấu trúc ẩn hoặc điểm bất thường. Kỹ thuật này thường dùng để phát hiện các hành vi lệch chuẩn trong luồng dữ liệu mạng mà chưa có mẫu nhận diện sẵn.

2.3.3 Học sâu (Deep Learning)

Sử dụng mạng nơ-ron đa tầng để tự động trích xuất đặc trưng từ dữ liệu thô (như chuỗi byte hoặc mã nguồn). Học sâu mang lại hiệu quả cao trong việc hiểu ngữ cảnh phức tạp của các cuộc tấn công tinh vi nhưng yêu cầu tài nguyên tính toán lớn.

2.4 Thách thức hiện tại

Việc triển khai học máy vào hệ thống bảo mật thực tế đối mặt với các rào cản đặc thù của môi trường mạng.

2.4.1 Vấn đề mất cân bằng dữ liệu (Imbalanced Data)

Lưu lượng hợp pháp luôn áp đảo lưu lượng tấn công, khiến mô hình dễ bị thiên kiến (bias) về lớp đa số. Điều này dẫn đến việc mô hình có độ chính xác tổng thể cao nhưng lại bỏ sót các cuộc tấn công thực sự (Recall thấp).

2.4.2 Phát hiện các cuộc tấn công Zero-day và biến thể mới

Các mô hình học máy thường chỉ nhận diện được các mẫu đã xuất hiện trong tập huấn luyện. Sự thay đổi liên tục của phương thức tấn công (Concept Drift) đòi hỏi hệ thống phải có khả năng thích ứng và cập nhật liên tục.

2.4.3 Yêu cầu về tốc độ xử lý và thời gian thực

Hệ thống bảo mật cần đưa ra quyết định trong mili giây để không gây trễ mạng. Việc cân bằng giữa độ phức tạp của mô hình (để đạt độ chính xác cao) và tốc độ suy diễn (inference speed) là một bài toán tối ưu quan trọng.

2.4.4 Sự phổ biến của dữ liệu mã hóa và "điểm mù" của hệ thống phòng thủ

Trong những năm gần đây, giao thức HTTPS đã trở thành tiêu chuẩn bắt buộc cho hầu hết các ứng dụng web nhằm đảm bảo tính riêng tư và toàn vẹn dữ liệu. Tuy nhiên, sự phổ biến của mã hóa cũng tạo ra một thách thức lớn cho các hệ thống bảo mật truyền thống:

- **Điểm mù payload:** Các tường lửa thế hệ cũ và IDS dựa trên chữ ký thường chỉ có thể kiểm tra các thông tin ở lớp Header mà không thể nhìn thấy nội dung bên trong gói tin khi đã được mã hóa. Kẻ tấn công lợi dụng điều này để giấu các mã độc XSS, SQL Injection hoặc các lệnh điều khiển botnet bên trong luồng dữ liệu HTTPS hợp lệ.
- **Thách thức của việc giải mã:** Để kiểm tra dữ liệu mã hóa, các hệ thống IPS cần thực hiện kỹ thuật giải mã và mã hóa lại. Quá trình này không chỉ tiêu tốn tài nguyên phần cứng cực lớn, vừa làm tăng đáng kể độ trễ mạng mà còn gây ra các rủi ro về quyền riêng tư và vi phạm tính tuân thủ bảo mật dữ liệu.

- **Giả mạo và lạm dụng chứng chỉ số :** Kẻ tấn công hiện nay có thể dễ dàng sở hữu các chứng chỉ SSL/TLS miễn phí để làm cho trang web phishing có vẻ giống như một trang web hợp lệ. Điều này khiến việc chỉ dựa vào biểu tượng khóa an toàn trên trình duyệt là không đủ để phân biệt trang web thật và giả.

Do đó, thách thức đặt ra cho mô hình nghiên cứu là phải tìm ra các phương pháp phát hiện tấn công thông qua các dấu hiệu gián tiếp như các mẫu hành vi mạng mà không cần can thiệp sâu vào nội dung đã mã hóa của người dùng.

2.5 Các bộ dữ liệu chuẩn dùng trong đánh giá (Benchmark Datasets)

Việc lựa chọn bộ dữ liệu phù hợp đảm bảo tính khách quan khi so sánh hiệu năng giữa các mô hình. Dưới đây là các bộ dữ liệu tiêu chuẩn hiện đại thường được sử dụng trong nghiên cứu an ninh mạng.

2.5.1 UNSW-NB15

Bộ dữ liệu hiện đại thay thế cho KDD99, bao gồm 9 loại tấn công phổ biến và 49 đặc trưng được trích xuất từ lưu lượng mạng thực tế, giúp phản ánh tốt hơn các mối đe dọa mới.

2.5.2 CIC-IDS2017

Mô phỏng mạng doanh nghiệp với đầy đủ các giao thức (HTTP, HTTPS,...) và các kịch bản tấn công web cụ thể như SQL Injection, XSS và Brute Force trong môi trường thực tế.

2.5.3 CSE-CIC-IDS2018

Phiên bản nâng cấp triển khai trên hạ tầng đám mây AWS, tập trung vào bài toán dữ liệu lớn (Big Data) và các biến thể tấn công tinh vi trên quy mô mạng lưới rộng lớn.

2.5.4 CSIC 2010 (HTTP Dataset)

Tập trung chuyên sâu vào giao thức HTTP, cung cấp hàng ngàn yêu cầu (requests) để đánh giá hiệu quả của các hệ thống WAF và phát hiện bất thường trên tầng ứng dụng web.

Table 2.1: Bảng so sánh tóm tắt các bộ dữ liệu phổ biến

Bộ dữ liệu	**Năm**	**Môi trường**	**Trọng tâm nghiên cứu**
UNSW-NB15	2015	Mạng giả lập (Lab)	Tấn công mạng tổng quát
CIC-IDS2017	2017	Mạng doanh nghiệp	Web Attacks, DoS, Botnet
CSE-CIC-IDS2018	2018	Điện toán đám mây (AWS)	Big Data, DDoS, Web Attacks
CSIC 2010	2010	Web Server (E-commerce)	**Chuyên sâu Web (SQLi, XSS)**

Nhận xét và lựa chọn dữ liệu cho đề tài: Mặc dù các bộ dữ liệu trên là chuẩn mực cho IDS/WAF, chúng chủ yếu chứa lưu lượng mạng tĩnh và thiếu các đặc trưng hạ tầng thời gian thực (DNS, SSL/TLS) cần thiết cho bài toán Phishing. Do đó, nghiên cứu này sẽ xây dựng bộ dữ liệu mới từ các nguồn URL uy tín (PhishTank và Majestic Million) để đảm bảo tính cập nhật và thực tiễn.

Chương 3

Phân tích và đánh giá các phương pháp hiện tại

3.1 Các mô hình học máy truyền thống

Dựa trên các khái niệm về học có giám sát đã được giới thiệu tại Chương 2, phần này đi sâu vào phân tích các thuật toán học máy truyền thống đóng vai trò nền tảng trong việc phát triển các hệ thống bảo mật. Mặc dù sự trỗi dậy của học sâu mang lại khả năng xử lý dữ liệu quy mô lớn, các mô hình truyền thống vẫn giữ vững vị thế nhờ tính minh bạch và chi phí vận hành thấp, đặc biệt phù hợp với yêu cầu thời gian thực đã nêu ở Mục 2.4.3.

3.1.1 Support Vector Machine (SVM)

Nguyên lý cơ bản: Support Vector Machine (SVM) là phương pháp học có giám sát tối ưu cho các bài toán phân lớp nhị phân với biên phân cách rõ rệt.

Phương thức vận hành: Trọng tâm của SVM là xác định một siêu phẳng (hyperplane) trong không gian đa chiều nhằm phân tách tối đa các lớp dữ liệu. Khoảng cách từ siêu phẳng đến các điểm dữ liệu gần nhất (support vectors) được gọi là lề (margin), và mục tiêu của thuật toán là cực đại hóa khoảng cách

này để tăng khả năng tổng quát hóa.
Giả sử tập dữ liệu huấn luyện bao gồm các cặp (x_i, y_i) với $x_i \in \mathbb{R}^n$ và $y_i \in \{-1, 1\}$. Siêu phẳng phân cách được định nghĩa bởi phương trình:

$$w^T x + b = 0 \tag{3.1}$$

Trong đó w là vectơ pháp tuyến và b là hệ số chệch (bias). Bài toán tối ưu hóa của SVM là tìm cực tiểu của $\|w\|$ dưới điều kiện ràng buộc:

$$y_i(w^T x_i + b) \geq 1, \quad \forall i \tag{3.2}$$

Trong thực tế, dữ liệu mạng thường không thể phân tách tuyến tính. Để giải quyết vấn đề này, SVM sử dụng kỹ thuật **Kernel Trick** (điển hình là Radial Basis Function - RBF) để ánh xạ dữ liệu từ không gian ban đầu sang không gian nhiều chiều hơn, nơi mà việc phân tách tuyến tính trở nên khả thi.
Đánh giá trong ngữ cảnh bảo mật:

- *Thế mạnh:* Duy trì độ chính xác ổn định ngay cả khi số lượng đặc trưng lớn hơn số lượng mẫu.
- *Hạn chế:* Hiệu năng giảm đáng kể khi tập dữ liệu có độ nhiễu cao và chi phí tính toán tăng lũy thừa theo quy mô dữ liệu.

3.1.2 Random Forest (RF)

Nguyên lý cơ bản: Random Forest là thuật toán học kết hợp (Ensemble Learning) dựa trên kỹ thuật Bagging, tạo ra một tập hợp các cây quyết định để giảm thiểu phương sai và tránh hiện tượng quá khớp (overfitting).
Cơ chế hoạt động: Thay vì dựa vào một cây quyết định duy nhất, RF tạo ra nhiều cây con bằng cách:

1. Lấy mẫu ngẫu nhiên có hoàn lại (bootstrapping) từ tập dữ liệu gốc để tạo ra các tập con khác nhau cho từng cây.

2. Tại mỗi nút phân chia của cây, chỉ xem xét một tập hợp ngẫu nhiên các đặc trưng thay vì toàn bộ đặc trưng.

Kết quả dự đoán cuối cùng được quyết định bằng cơ chế bỏ phiếu số đông từ tất cả các cây trong rừng. Đối với bài toán phân loại tấn công:

$$\hat{y} = \text{mode}\{T_1(x), T_2(x), ..., T_n(x)\} \tag{3.3}$$

Trong đó $T_i(x)$ đại diện cho đầu ra của cây thành phần thứ i.

Đánh giá trong ngữ cảnh bảo mật:

- *Thế mạnh:* Khả năng xử lý tốt dữ liệu mất cân bằng (thường gặp trong tấn công mạng) và cung cấp chỉ số "Feature Importance" giúp chuyên gia hiểu rõ đặc trưng nào dẫn đến hành vi lừa đảo.
- *Hạn chế:* Tốc độ dự đoán thời gian thực có thể bị ảnh hưởng nếu số lượng cây trong rừng quá lớn, gây trễ cho các hệ thống IPS.

3.1.3 Gradient Boosting và LightGBM

Khái niệm: Gradient Boosting là một kỹ thuật học máy mạnh mẽ, xây dựng mô hình dự đoán dưới dạng một tập hợp các mô hình dự đoán yếu (thường là cây quyết định). Khác với Random Forest (xây dựng các cây song song), Gradient Boosting xây dựng các cây một cách tuần tự, trong đó mỗi cây mới được tạo ra để sửa chữa các sai số (residual errors) của các cây trước đó.

LightGBM (Light Gradient Boosting Machine): Là một framework triển khai Gradient Boosting hiệu quả cao do Microsoft phát triển. LightGBM giới thiệu hai kỹ thuật mới để giải quyết vấn đề hiệu năng trên dữ liệu lớn:

- **Gradient-based One-Side Sampling (GOSS):** Giữ lại các mẫu dữ liệu có gradient lớn (những mẫu bị dự đoán sai nhiều) và lấy mẫu ngẫu nhiên các mẫu có gradient nhỏ. Điều này giúp thuật toán tập trung học vào các phần dữ liệu khó.

- **Exclusive Feature Bundling (EFB):** Gom nhóm các đặc trưng thưa (sparse features) ít khi nhận giá trị khác 0 cùng lúc để giảm số chiều dữ liệu mà không làm mất thông tin.

Ưu/Nhược điểm trong IDS:

- *Ưu điểm:* Tốc độ huấn luyện nhanh hơn nhiều so với các triển khai Boosting truyền thống (như XGBoost), tiêu tốn ít bộ nhớ, và thường đạt độ chính xác cao hơn Random Forest trên dữ liệu dạng bảng có cấu trúc.
- *Nhược điểm:* Dễ bị overfitting trên các tập dữ liệu quá nhỏ (dưới 10.000 mẫu) và nhạy cảm với nhiễu nếu các siêu tham số không được tinh chỉnh kỹ lưỡng.

3.1.4 K-Nearest Neighbors (KNN)

Khái niệm: K-Nearest Neighbors là một thuật toán lazy learning, nghĩa là nó không thực sự học một mô hình trong giai đoạn huấn luyện mà chỉ lưu trữ toàn bộ dữ liệu. Việc tính toán chỉ diễn ra khi cần dự đoán một điểm dữ liệu mới.
Cơ chế hoạt động: Để phân loại một mẫu dữ liệu mới, KNN thực hiện các bước sau:

1. Tính khoảng cách từ điểm mới đến tất cả các điểm trong tập dữ liệu đã lưu. Thước đo khoảng cách phổ biến nhất là khoảng cách Euclidean:

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \tag{3.4}$$

2. Chọn ra K điểm có khoảng cách gần nhất.
3. Gán nhãn cho điểm mới dựa trên nhãn phổ biến nhất trong số K láng giềng này.

Ưu/Nhược điểm trong IDS:

- *Ưu điểm:* Rất đơn giản để cài đặt, không cần giả định về phân phối của dữ liệu, hiệu quả với các tập dữ liệu nhỏ và ít nhiễu.
- *Nhược điểm:* Chi phí tính toán cực lớn trong giai đoạn dự đoán do phải quét toàn bộ dữ liệu, không phù hợp cho các hệ thống IPS thời gian thực yêu cầu độ trễ thấp. Ngoài ra, KNN rất nhạy cảm với việc chọn giá trị K và các đặc trưng không quan trọng.

3.2 Các mô hình học sâu

Khác với các mô hình học máy truyền thống phụ thuộc nhiều vào quá trình trích xuất đặc trưng thủ công, các mô hình học sâu có khả năng tự động học các biểu diễn đặc trưng phức tạp từ dữ liệu thô thông qua các lớp mạng nơ-ron đa tầng. Trong lĩnh vực phát hiện xâm nhập, ba kiến trúc phổ biến và hiệu quả nhất hiện nay bao gồm CNN, LSTM và Transformer.

3.2.1 Convolutional Neural Networks (CNN)

Khái niệm: Mạng nơ-ron tích chập (CNN) ban đầu được thiết kế cho bài toán thị giác máy tính, nhưng đã chứng minh được hiệu quả vượt trội trong an ninh mạng. Trong ngữ cảnh IDS, CNN thường được sử dụng để trích xuất các đặc trưng không gian từ lưu lượng mạng, bằng cách coi các gói tin hoặc chuỗi byte như một hình ảnh hoặc một mảng một chiều.

Cơ chế hoạt động: Kiến trúc CNN bao gồm ba thành phần chính: lớp tích chập (Convolutional Layer), lớp gộp (Pooling Layer) và lớp kết nối đầy đủ (Fully Connected Layer).

- **Lớp tích chập:** Thực hiện phép tính tích chập giữa dữ liệu đầu vào và các bộ lọc (filters/kernels) để tạo ra các bản đồ đặc trưng. Giả sử đầu vào x là một chuỗi 1 chiều và w là bộ lọc có kích thước K, giá trị đầu ra y tại vị trí i được tính như sau:

$$y_i = \sum_{k=0}^{K-1} w_k \cdot x_{i+k} + b \tag{3.5}$$

- **Lớp gộp (Pooling):** Giảm chiều dữ liệu để giảm chi phí tính toán và hạn chế overfitting. Phổ biến nhất là Max-Pooling (lấy giá trị lớn nhất trong một vùng có kích thước xác định).

Ưu/Nhược điểm trong IDS:

- *Ưu điểm:* Khả năng tự động trích xuất các mẫu tấn công tiềm ẩn trong payload của gói tin mà con người khó nhận biết, tốc độ dự đoán nhanh nhờ khả năng tính toán song song.
- *Nhược điểm:* Bị hạn chế trong việc nắm bắt các mối quan hệ phụ thuộc dài hạn trong chuỗi thời gian của các phiên kết nối mạng.

3.2.2 Long Short-Term Memory (LSTM)

Khái niệm: LSTM là một biến thể đặc biệt của mạng nơ-ron hồi quy (RNN), được thiết kế để giải quyết vấn đề vanishing gradient khi huấn luyện trên các chuỗi dữ liệu dài. Đây là mô hình lý tưởng để phân tích lưu lượng mạng dựa trên luồng hoặc nhật ký hệ thống theo thời gian.

Cơ chế hoạt động: Mỗi đơn vị LSTM (cell) duy trì một trạng thái (cell state) C_t chạy dọc theo chuỗi thông tin. LSTM sử dụng các cổng (gates) để điều chỉnh luồng thông tin:

1. **Cổng quên (Forget Gate):** Quyết định thông tin nào từ trạng thái trước đó h_{t-1} sẽ bị loại bỏ.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{3.6}$$

2. **Cổng vào (Input Gate):** Quyết định thông tin mới nào sẽ được lưu vào cell state.

3. **Cổng ra (Output Gate):** Tính toán trạng thái ẩn h_t cho bước tiếp theo.

Trong đó σ là hàm kích hoạt Sigmoid đưa giá trị về khoảng $(0, 1)$.

Ưu/Nhược điểm trong IDS:

- *Ưu điểm:* Hiệu quả cao trong việc phát hiện các cuộc tấn công phức tạp diễn ra qua nhiều bước hoặc kéo dài theo thời gian (tấn công APT hoặc DDoS chậm).

- *Nhược điểm:* Thời gian huấn luyện lâu do tính tuần tự, tiêu tốn nhiều tài nguyên bộ nhớ.

3.2.3 Transformer

Khái niệm: Transformer là kiến trúc hiện đại nhất, ban đầu tạo ra bước đột phá trong xử lý ngôn ngữ tự nhiên (NLP) và gần đây được áp dụng mạnh mẽ vào an ninh mạng (BERT trong phân tích log). Transformer loại bỏ hoàn toàn cấu trúc hồi quy của RNN/LSTM và thay thế bằng cơ chế **Attention**.

Cơ chế hoạt động: Trái tim của Transformer là cơ chế *Self-Attention*, cho phép mô hình xem xét mối quan hệ giữa một phần tử trong chuỗi với tất cả các phần tử khác, bất kể khoảng cách giữa chúng. Công thức tính Attention được định nghĩa:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3.7}$$

Trong đó Q (Query), K (Key), và V (Value) là các ma trận biểu diễn dữ liệu đầu vào, và d_k là chiều của vector Key. Cơ chế này giúp mô hình tập trung vào các đặc trưng quan trọng nhất của lưu lượng mạng trong bối cảnh toàn cục.

Ưu/Nhược điểm trong IDS:

- *Ưu điểm:* Khả năng xử lý song song vượt trội so với LSTM, nắm bắt tốt các mối quan hệ ngữ nghĩa phức tạp trong các payload dạng văn bản (HTTP request, SQL queries).

- *Nhược điểm:* Yêu cầu lượng dữ liệu huấn luyện khổng lồ và tài nguyên tính toán rất lớn để vận hành hiệu quả.

3.3 Tổng quan các nghiên cứu mới nhất về phát hiện tấn công Web

Trong giai đoạn từ năm 2020 đến nay, trọng tâm nghiên cứu trong lĩnh vực phát hiện xâm nhập đã chuyển dịch mạnh mẽ từ việc sử dụng các mô hình đơn lẻ sang các kiến trúc lai (Hybrid Models) và ứng dụng các kỹ thuật Xử lý ngôn ngữ tự nhiên (NLP) tiên tiến để giải mã các payload phức tạp của tấn công Web (SQLi, XSS).

3.3.1 Các mô hình lai (Hybrid Deep Learning Models)

Các nghiên cứu gần đây chỉ ra rằng việc kết hợp ưu điểm của nhiều kiến trúc mạng khác nhau mang lại hiệu quả vượt trội so với các mô hình đơn lẻ. Xu hướng phổ biến nhất là kết hợp CNN (để trích xuất đặc trưng không gian) với LSTM/GRU (để nắm bắt sự phụ thuộc thời gian).

- **Kiến trúc lai CNN-LSTM:** Một nghiên cứu được công bố trên IEEE Access (2022) đã đề xuất mô hình lai tận dụng khả năng trích xuất các đặc trưng không gian từ dữ liệu thô của CNN và khả năng ghi nhớ các phụ thuộc dài hạn của mạng LSTM. Cụ thể, mô hình này tích hợp thêm các lớp Batch Normalization và lớp Dropout để giải quyết vấn đề overfitting và tăng tốc độ hội tụ. Các thực nghiệm trên các bộ dữ liệu tiêu chuẩn như CIC-IDS 2017, UNSW-NB15 và WSN-DS cho thấy kiến trúc này không chỉ đạt tỷ lệ phát hiện cao đối với cả phân loại nhị phân và đa lớp, mà còn giảm thiểu đáng kể tỷ lệ báo động giả so với việc sử dụng CNN hoặc LSTM riêng lẻ[2].

- **Kết hợp Autoencoder và Bộ phân lớp (Autoencoder-Classifier Ensemble):** Để giải quyết thách thức về dữ liệu mất cân bằng và các dạng tấn công chưa biết, các kiến trúc lai sử dụng Autoencoder (AE) làm bộ lọc nhiễu đang trở thành xu hướng. Một nghiên cứu tiêu biểu trên tạp chí *Symmetry*

(2023) đã đề xuất mô hình hai giai đoạn: sử dụng Random Forest để phân loại sơ bộ, sau đó kết hợp với Autoencoder để thẩm định lại các mẫu nghi ngờ dựa trên Reconstruction Error. Phương pháp này tận dụng khả năng học biểu diễn dữ liệu bình thường của AE để loại bỏ các cảnh báo sai (FP) mà các bộ phân lớp giám sát thường mắc phải. Thực nghiệm trên bộ dữ liệu NF-CSE-CIC-IDS2018-v2 và NF-BoT-IoT-v2 cho thấy mô hình này vượt trội trong việc giảm thiểu tỷ lệ báo động giả trong khi vẫn duy trì độ chính xác cao[4].

3.3.2 Ứng dụng NLP và Transformer trong phát hiện tấn công Web

Tấn công ứng dụng web (như SQL Injection, XSS) bản chất là các chuỗi văn bản độc hại được chèn vào các câu lệnh hợp lệ. Do đó, các kỹ thuật NLP đang đóng vai trò chủ đạo trong payload analysis.

- **Word Embeddings (Word2Vec, FastText):** Thay vì sử dụng One-hot encoding truyền thống, các nghiên cứu mới chuyển đổi các từ khóa trong URL hoặc HTTP Request thành các vector không gian (embeddings) để bảo toàn ngữ nghĩa. Ví dụ: từ khóa `SELECT` sẽ có vector gần với `UNION` hơn là `admin`.
- **BERT và RoBERTa:** Sự ra đời của các mô hình ngôn ngữ lớn (LLM) đã mở ra hướng đi mới. Các nghiên cứu năm 2023-2024 đã áp dụng BERT (Bidirectional Encoder Representations from Transformers) để đọc hiểu ngữ cảnh của một câu truy vấn SQL. Mô hình có thể phân biệt được `OR 1=1` trong ngữ cảnh tấn công và trong ngữ cảnh văn bản bình thường nhờ cơ chế Attention hai chiều. Một nghiên cứu năm 2023 đã chứng minh hiệu quả của việc kết hợp các kỹ thuật trên. Tác giả đề xuất mô hình lai ITCBL tích hợp cả *Word2Vec* (để vector hóa đặc trưng cục bộ) và *BERT* (để học chuyển giao ngữ cảnh). Kết quả thực nghiệm cho thấy sự kết hợp này giúp mô hình hiểu sâu hơn về cấu trúc ngữ nghĩa của câu lệnh SQL, đạt điểm

F1-Score lên tới 99.57% và giảm tỷ lệ báo động giả xuống mức rất thấp (0.39%), vượt trội hoàn toàn so với việc chỉ sử dụng các phương pháp đơn lẻ[3].

3.3.3 Học chuyển giao (Transfer Learning) và Học liên kết (Federated Learning)

- **Transfer Learning (Học chuyển giao):** Kỹ thuật này cho phép tái sử dụng tri thức từ các mô hình đã huấn luyện trên các bộ dữ liệu lớn (Base Dataset) để tinh chỉnh cho các tập dữ liệu nhỏ hơn hoặc đặc thù (Target Dataset). Phương pháp này đặc biệt hữu ích để giải quyết vấn đề thiếu hụt dữ liệu gán nhãn. Một nghiên cứu vào năm 2019 đã đề xuất mô hình **TL-ConvNet** hoạt động theo quy trình học hai giai đoạn. Mô hình học các đặc trưng nền tảng từ bộ dữ liệu UNSW-NB15 trước khi chuyển giao kiến thức sang bộ dữ liệu NSL-KDD. Kết quả thực nghiệm cho thấy phương pháp này giúp cải thiện tới **22.02%** độ chính xác đối với các dạng tấn công mới so với mô hình ConvNet truyền thống, chứng minh hiệu quả vượt trội trong việc phát hiện các mối đe dọa chưa từng biết đến[5].

- **Federated Learning (Học liên kết):** Đây là xu hướng nổi bật để bảo vệ quyền riêng tư. Thay vì gửi dữ liệu log nhạy cảm về một máy chủ trung tâm để huấn luyện, mô hình được huấn luyện cục bộ tại từng thiết bị (edge devices) và chỉ gửi tham số cập nhật (gradients) về máy chủ. Một nghiên cứu thực nghiệm năm 2023 đã triển khai mô hình học sâu liên kết trên bộ dữ liệu TON-IoT mô phỏng môi trường thực tế (dữ liệu Non-IID). Nhóm tác giả phát hiện rằng sự không đồng nhất của dữ liệu có thể làm giảm hiệu suất phát hiện tới 50% so với mô hình tập trung. Tuy nhiên, họ đã đề xuất giải pháp khởi tạo bằng *mô hình được huấn luyện trước (Pre-trained Global Model)*, giúp cải thiện điểm F1-Score lên hơn **20%**. Đồng thời, nghiên cứu cũng chỉ ra rằng thuật toán tổng hợp **FedProx** hoạt động hiệu quả hơn FedAvg truyền thống trong các mạng IoT phân tán[1].

3.4 Đánh giá hiệu suất

Việc lựa chọn các thước đo đánh giá phù hợp là yếu tố then chốt để xác định hiệu quả thực sự của một hệ thống phát hiện xâm nhập. Đặc biệt trong bối cảnh dữ liệu an ninh mạng thường bị mất cân bằng nghiêm trọng, việc chỉ dựa vào độ chính xác đơn thuần có thể dẫn đến những kết luận sai lệch. Dưới đây là các chỉ số tiêu chuẩn được sử dụng trong luận văn này.

3.4.1 Ma trận nhầm lẫn (Confusion Matrix)

Ma trận nhầm lẫn là nền tảng để tính toán tất cả các chỉ số hiệu suất khác. Đối với bài toán phân lớp nhị phân (Bình thường - 0 và Tấn công - 1), ma trận này được cấu thành từ 4 thành phần:

- **True Positive (TP):** Số lượng mẫu tấn công được hệ thống dự đoán *chính xác* là tấn công. Đây là chỉ số quan trọng nhất thể hiện khả năng xác định được mối đe dọa.
- **True Negative (TN):** Số lượng mẫu bình thường được hệ thống dự đoán *chính xác* là bình thường.
- **False Positive (FP - Báo động giả):** Số lượng mẫu bình thường bị hệ thống *nhầm lẫn* là tấn công.
- **False Negative (FN - Bỏ sót):** Số lượng mẫu tấn công bị hệ thống *nhầm lẫn* là bình thường. Đây là loại lỗi nguy hiểm nhất vì nó cho phép kẻ tấn công xâm nhập mà không bị phát hiện.

3.4.2 Các chỉ số cơ bản

3.4.2.1 Accuracy (Độ chính xác toàn cục)

Là tỷ lệ tổng số các dự đoán đúng trên tổng số mẫu dữ liệu.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.8}$$

Lưu ý: Trong IDS, Accuracy thường không phản ánh đúng hiệu năng. Ví dụ, nếu 99% lưu lượng là bình thường, một mô hình ngây thơ dự đoán tất cả là bình thường sẽ có Accuracy 99%, nhưng thất bại hoàn toàn trong việc phát hiện tấn công.

3.4.2.2 Precision (Độ chính xác của cảnh báo)

Precision là tỷ lệ số lượng tấn công thật trên tổng số các cảnh báo tấn công mà hệ thống đưa ra.

$$Precision = \frac{TP}{TP + FP} \tag{3.9}$$

Precision cao đồng nghĩa với tỷ lệ báo động giả (FP) thấp.

3.4.2.3 Recall (Độ nhạy - Sensitivity)

Recall là tỷ lệ số cuộc tấn công thực tế mà hệ thống phát hiện được.

$$Recall = \frac{TP}{TP + FN} \tag{3.10}$$

Trong an ninh mạng, Recall thường được ưu tiên hơn Precision vì hậu quả của việc bỏ sót tấn công (FN) nghiêm trọng hơn nhiều so với báo động giả.

3.4.2.4 F1-Score

F1-Score là trung bình điều hòa của Precision và Recall. Đây là chỉ số tổng quát tốt nhất khi dữ liệu bị mất cân bằng, vì nó yêu cầu cả Precision và Recall đều

phải cao.

$$F1\text{-}score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3.11}$$

3.4.3 Đường cong ROC và chỉ số AUC

Đường cong ROC cung cấp cái nhìn toàn diện hơn về khả năng phân tách của mô hình tại mọi ngưỡng quyết định.

- **Trục tung (Y-axis):** True Positive Rate (TPR), chính là Recall.
- **Trục hoành (X-axis):** False Positive Rate (FPR), được tính bằng $\frac{FP}{FP+TN}$.

Area Under the Curve (AUC): Là diện tích nằm dưới đường cong ROC, có giá trị từ 0 đến 1.

- $AUC = 0.5$: Mô hình dự đoán ngẫu nhiên (tương đương tung đồng xu).
- $AUC = 1.0$: Mô hình phân loại hoàn hảo.
- $AUC > 0.9$: Mô hình xuất sắc.

Trong nghiên cứu này, AUC được sử dụng như một thước đo chính để so sánh sự ổn định của các mô hình đề xuất so với các phương pháp truyền thống.

3.5 Phân tích hạn chế của các phương pháp hiện tại

Mặc dù các nghiên cứu đã đạt được những kết quả khả quan, việc áp dụng các mô hình học máy và học sâu vào môi trường thực tế vẫn tồn tại những rào cản đáng kể.

3.5.1 Tỷ lệ cảnh báo sai (False Positive) vẫn còn cao

Như đã thảo luận về cơ chế Anomaly-based Detection tại Mục 2.1.1, hạn chế lớn nhất của các mô hình này là khó khăn trong việc phân biệt giữa hành vi người dùng hợp lệ nhưng khác thường và hành vi tấn công thực sự.

- Các thuật toán học sâu như LSTM hay Autoencoder thường rất nhạy (Recall cao) nhưng độ chính xác (Precision) lại không ổn định khi môi trường mạng thay đổi.
- Việc tạo ra quá nhiều cảnh báo sai dẫn đến tình trạng ngó lơ cảnh báo.

3.5.2 Thiếu khả năng giải thích (Lack of Explainability)

Đây là vấn đề Hộp đen (Black-box) điển hình của các mô hình học sâu như CNN hay Transformer.

- Khi hệ thống đưa ra cảnh báo, nó thường không chỉ ra được tại sao nó lại kết luận như vậy.
- Trong an ninh mạng, khả năng giải thích (Explainable AI - XAI) là bắt buộc để các chuyên gia phân tích có thể truy vết nguồn gốc tấn công và vá lỗ hổng. Các mô hình truyền thống như Decision Tree làm tốt việc này, nhưng độ chính xác lại thấp hơn Deep Learning.

3.5.3 Tính dễ bị tổn thương trước Tấn công Đối kháng (Adversarial Attacks)

Một vấn đề mới nổi nhưng cực kỳ nghiêm trọng là các mô hình học máy có thể bị đánh lừa bởi các mẫu đối kháng (Adversarial Examples).
Kẻ tấn công có thể thêm một lượng noise nhỏ vào các gói tin độc hại - lượng nhiễu này con người không thể nhận ra nhưng đủ để làm thay đổi quyết định của mô hình AI từ "Tấn công" sang "Bình thường". Hầu hết các hệ thống IDS học máy hiện nay chưa được trang bị cơ chế phòng thủ mạnh mẽ để chống lại loại hình tấn công này.

3.5.4 Yêu cầu tài nguyên và khả năng triển khai thực tế

- **Độ trễ (Latency):** Các mô hình phức tạp như BERT hay quy trình trích xuất đặc trưng của CNN tốn nhiều thời gian tính toán. Trong các hệ thống

mạng cao tốc, độ trễ xử lý phải ở mức mili-giây để không làm gián đoạn trải nghiệm người dùng.

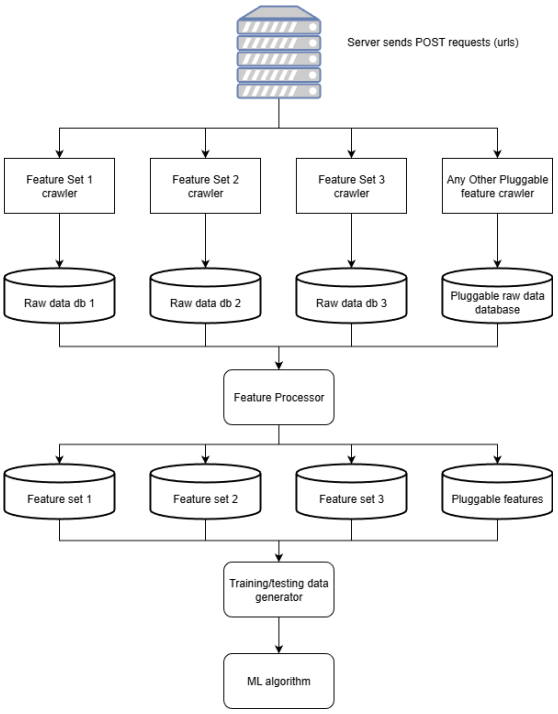
- **Chi phí phần cứng:** Việc triển khai các mô hình Deep Learning đòi hỏi các máy chủ có GPU mạnh mẽ, gây khó khăn cho việc tích hợp vào các thiết bị biên hoặc tường lửa công suất thấp.

Chương 4

Đề xuất mô hình cải tiến

4.1 Ý tưởng và kiến trúc mô hình

Mô hình phát hiện website lừa đảo được đề xuất dựa trên việc coi nhiệm vụ phát hiện phishing là một bài toán phân loại nhị phân. Ý tưởng cốt lõi của hệ thống là khai thác một tập hợp phong phú các đặc trưng tĩnh, phản ánh đa dạng các khía cạnh của trang web từ nội dung mã nguồn đến cơ sở hạ tầng lưu trữ. Đặc biệt, hệ thống tận dụng các điểm thu thập dữ liệu phân tán để trích xuất các đặc trưng vật lý và liên quan đến hệ thống mạng, thứ mà khó bị kẻ tấn công giả mạo.



Kiến trúc đề xuất tuân theo nguyên tắc module hóa cao với các thành phần tách biệt nhằm đảm bảo tính linh hoạt và khả năng mở rộng, đáp ứng yêu cầu về tốc độ xử lý thời gian thực đã đặt ra trong Mục tiêu nghiên cứu (Chương 1). Luồng vận hành bao gồm các bước chính sau:

1. **Các luồng thu thập và trích xuất đặc trưng:** Dữ liệu được xử lý qua các luồng độc lập tương ứng với từng nhóm đặc trưng, bao gồm:

    - *Crawler:* Chịu trách nhiệm tải dữ liệu thô từ các URL mục tiêu. Các crawler này hoạt động theo cơ chế phân tán tại nhiều khu vực địa lý khác nhau để đảm bảo tính khách quan của các đặc trưng mạng và chứng chỉ số (chi tiết về môi trường triển khai được trình bày tại Mục 4.4).
    - *Raw Data Database:* Lưu trữ dữ liệu thô thu thập được để phục vụ việc phân tích sau này.
    - *Processor:* Truy xuất dữ liệu từ cơ sở dữ liệu thô và chuyển đổi chúng thành các giá trị đặc trưng cụ thể.
    - *Feature Database:* Lưu trữ các đặc trưng đã được trích xuất.

2. **Tổng hợp dữ liệu (Data Generator):** Tạo một dataframe chứa dữ liệu huấn luyện/kiểm thử từ các đặc trưng trong các cơ sở dữ liệu thành phần phục vụ cho quá trình huấn luyện mô hình học máy.

3. **Phân loại (Classifier):** Dataframe chứa các vector đặc trưng được đưa vào thuật toán học máy để huấn luyện mô hình. Trong nghiên cứu này, thuật toán **Random Forests và LightGBM** được lựa chọn nhờ khả năng chống nhiễu tốt, dễ dàng song song hóa và cung cấp đánh giá về độ quan trọng của đặc trưng.

4.2 Tiền xử lý dữ liệu

Tiền xử lý dữ liệu là một bước tối quan trọng trong bất kỳ quy trình học máy nào, quyết định trực tiếp đến hiệu năng và độ chính xác của mô hình. Trong bài toán phát hiện phishing, bước này càng trở nên quan trọng do tính đa dạng của dữ liệu đầu vào và sự mất cân bằng nghiêm trọng giữa các lớp. Quy trình tiền xử lý của mô hình đề xuất tập trung vào hai công đoạn chính: trích xuất, lựa chọn đặc trưng và cân bằng dữ liệu.

4.2.1 Trích xuất và Lựa chọn Đặc trưng

Mô hình áp dụng phương pháp tiếp cận toàn diện, trích xuất các nhóm đặc trưng chính phản ánh từ tầng ứng dụng đến tầng mạng và vật lý. Cách tiếp cận này giúp khắc phục "điểm mù" của việc phân tích nội dung trên giao thức HTTPS (đã phân tích tại Mục 2.4.4) bằng cách tập trung vào các dấu hiệu hạ tầng khó giả mạo:

- **Đặc trưng chứng chỉ số (Certificate Features):** Khai thác sâu thông tin từ chứng chỉ X.509 thu thập được từ các điểm phân tán.
  - *Phần mở rộng (Extensions):* Kiểm tra sự tồn tại của các trường như `KeyUsage`, `CertificatePolicies`. Việc vắng mặt hoặc cấu hình sai

các trường quy định chính sách này thường xuất hiện ở các chứng chỉ được tạo sơ sài, là một dấu hiệu nghi vấn của các URL lừa đảo.

  - *Thông tin Issuer/Subject:* Kiểm tra danh sách tin cậy (Mozilla Trusted Store) và các chuỗi đáng ngờ ("localhost", "default"). Thông thường, các tổ chức hợp pháp thường sử dụng CA uy tín, trong khi kẻ tấn công thường dùng chứng chỉ tự ký hoặc từ CA kém uy tín. Tỷ lệ trùng khớp cao giữa Issuer và Subject cũng là dấu hiệu của chứng chỉ tự ký.
  - *Thông tin thời gian:* Các chứng chỉ có thời hạn quá dài thường được xem là kém bảo mật, và chứng chỉ mới được cấp thường có rủi ro cao hơn là các chứng chỉ đã tồn tại lâu năm.
  - *Đặc trưng phân tán:* Số lượng chứng chỉ khác nhau thu được từ các điểm đo. Việc thu được nhiều chứng chỉ khác nhau cho cùng một tên miền từ các vị trí địa lý khác nhau cho thấy một hạ tầng phân tán quy mô lớn (như CDN), điều mà các trang web lừa đảo thường ít khi đầu tư do chi phí vận hành cao.

- **Đặc trưng HTTP Header:** Phân tích tổng số trường header, sự tồn tại của `cache-control`, và độ dài trường `server`. Đa số các website hợp pháp thường được cấu hình kỹ lưỡng để tối ưu trải nghiệm người dùng, ngược lại website lừa đảo thường sử dụng cấu hình mặc định của web server với ít trường thông tin hơn.

- **Đặc trưng mạng (Network Features):** Trích xuất từ gói tin TCP (TTL, Window Size, RTT, Retransmission). Các đặc trưng này phản ánh chất lượng kết nối và ngăn xếp mạng của máy chủ vật lý. Máy chủ lừa đảo thường là các botnet hoặc các máy chủ bị xâm nhập với cấu hình mạng kém ổn định và hành vi TCP khác biệt so với các máy chủ thương mại chuyên dụng của các trang web uy tín.

- **Đặc trưng từ vựng URL (URL Lexical Features):** Kiểm tra việc sử dụng IP làm tên miền, số lượng dấu chấm hoặc gạch ngang. Sở dĩ có đặc trưng

này là vì việc sử dụng IP trực tiếp là hành vi hiếm gặp ở website hợp pháp nhằm tối ưu trải nghiệm người dùng, nhưng lại phổ biến ở website lừa đảo để tiết kiệm chi phí và thời gian đăng ký tên miền.

- **Đặc trưng DNS:** Truy vấn các bản ghi A, NS, PTR, AAAA và TTL. Số lượng bản ghi A/NS phong phú và sự tồn tại của AAAA (IPv6) cho thấy sự đầu tư kỹ lưỡng vào hạ tầng. Ngoài ra, tỷ lệ trùng khớp trong bản ghi PTR (Reverse DNS lookup) cao chứng tỏ địa chỉ IP được dành riêng cho tên miền đó, một dấu hiệu hiếm khi xuất hiện ở các trang web lừa đảo.
- **Đặc trưng chuyển hướng (HTTP Redirection):** Theo dõi chuỗi chuyển hướng và số lượng tên miền trung gian. Trong khi website hợp pháp chuyển hướng để bảo mật (HTTP sang HTTPS), kẻ tấn công thường lạm dụng chuyển hướng qua nhiều tên miền khác nhau hoặc các dịch vụ rút gọn link để che giấu URL đích thực và tránh các danh sách đen.

Sau khi trích xuất, một bước lựa chọn đặc trưng có thể được thực hiện. Bằng cách sử dụng các mô hình cây quyết định như Random Forest hoặc LightGBM, chúng ta có thể tính toán độ quan trọng của từng đặc trưng. Dựa trên điểm số này, các đặc trưng không mang lại nhiều giá trị thông tin có thể được loại bỏ để giảm độ phức tạp của mô hình và tăng tốc độ huấn luyện mà không làm ảnh hưởng độ chính xác.

4.2.2 Cân bằng dữ liệu

Vấn đề: Như đã phân tích tại Mục 2.4.1 (Thách thức hiện tại), sự mất cân bằng dữ liệu nghiêm trọng giữa lớp hợp pháp và lừa đảo là nguyên nhân chính khiến các mô hình truyền thống có độ chính xác (Accuracy) cao nhưng độ phủ (Recall) thấp. Nếu huấn luyện trực tiếp trên dữ liệu thô, mô hình sẽ có xu hướng thiên vị lớp đa số.

Giải pháp: Để giải quyết vấn đề này, kỹ thuật **SMOTE (Synthetic Minority Over-sampling Technique)** sẽ được sử dụng. SMOTE là một phương pháp over-

sampling thông minh, thay vì chỉ đơn thuần sao chép các mẫu thuộc lớp thiểu số, nó tạo ra các mẫu dữ liệu *tổng hợp* mới.
Cụ thể, SMOTE hoạt động như sau:

1. Chọn một mẫu ngẫu nhiên từ lớp thiểu số.
2. Tìm k hàng xóm gần nhất của mẫu đó (cũng thuộc lớp thiểu số).
3. Chọn ngẫu nhiên một trong số k hàng xóm này.
4. Tạo một mẫu tổng hợp mới tại một điểm ngẫu nhiên trên đoạn thẳng nối giữa mẫu ban đầu và hàng xóm đã chọn trong không gian đặc trưng.

4.3 Thuật toán và kỹ thuật tối ưu

Việc lựa chọn thuật toán phù hợp chỉ là bước khởi đầu. Để xây dựng một mô hình dự báo mạnh mẽ và đáng tin cậy, cần phải áp dụng các kỹ thuật tối ưu hóa và phương pháp đánh giá hiệu năng một cách khoa học. Phần này sẽ trình bày về thuật toán được đề xuất và các kỹ thuật được sử dụng để tinh chỉnh và kiểm định mô hình.

4.3.1 Các thuật toán đề xuất

Dựa trên kết quả phân tích ưu nhược điểm tại Chương 3, nghiên cứu lựa chọn các thuật toán thuộc họ cây quyết định (Tree-based) thay vì các mô hình học sâu để đảm bảo khả năng giải thích và hiệu suất trên dữ liệu dạng bảng. Cụ thể, nghiên cứu tập trung cấu hình và tinh chỉnh hai thuật toán: **Random Forest** (làm mô hình cơ sở) và **LightGBM** (mô hình đề xuất chính nhờ ưu điểm về tốc độ huấn luyện).

4.3.2 Tối ưu hóa siêu tham số và kiểm định chéo

Siêu tham số là các cấu hình được thiết lập trước khi huấn luyện. Việc lựa chọn thủ công thường tốn kém thời gian và không đảm bảo hiệu quả. Thay vào đó,

nghiên cứu sử dụng kỹ thuật **RandomizedSearchCV (Tìm kiếm ngẫu nhiên có kiểm định chéo)**.
Quy trình tối ưu hóa được thực hiện như sau:

1. **Định nghĩa không gian tham số (Hyperparameter Space):**
    - **Random Forest:** Số lượng cây (`n_estimators`) từ 100-300, độ sâu tối đa (`max_depth`) trong tập [10, 20, 30, None], số mẫu tối thiểu để chia nút (`min_samples_split`) từ 2-10 và tại nút lá (`min_samples_leaf`) từ 1-4.
    - **LightGBM:** Số lượng cây (`n_estimators`) từ 100-300, tốc độ học (`learning_rate`) từ 0.01-0.2, số lượng lá (`num_leaves`) từ 20-50 và tỷ lệ lấy mẫu dữ liệu (`subsample`) từ 0.6-1.0.
2. **Chiến lược tìm kiếm:** Sử dụng `RandomizedSearchCV` với số lần lặp $n\_iter = 10$ và đánh giá bằng Stratified K-Fold ($k = 3$). Metric tối ưu hóa là **F1-Score** để đảm bảo hiệu năng trên cả hai lớp dữ liệu.

4.4 Kiến trúc triển khai và Môi trường thực nghiệm

Để đảm bảo tính nhất quán, dễ dàng triển khai và khả năng tái lập của các kết quả nghiên cứu, toàn bộ hệ thống từ thu thập dữ liệu, huấn luyện mô hình cho đến dự đoán đều được xây dựng và vận hành trong một môi trường container hóa. Phần này mô tả chi tiết về kiến trúc hệ thống, các công nghệ phần mềm được sử dụng và cấu hình môi trường thử nghiệm.

4.4.1 Kiến trúc hệ thống dựa trên Microservices

Hệ thống được thiết kế theo kiến trúc Microservices và triển khai hoàn toàn trên nền tảng điện toán đám mây Microsoft Azure. Kiến trúc này mang lại sự linh hoạt, dễ dàng bảo trì và khả năng mở rộng theo nhu cầu. Các thành phần chính bao gồm:

- **Hệ thống Crawlers Thu thập dữ liệu:** Được triển khai lai (hybrid) giữa các dịch vụ PaaS và IaaS của Azure để tối ưu hóa hiệu năng và chi phí:
  - *Distributed Crawlers:* Bao gồm 4 node thu thập phân tán đặt tại các khu vực: **Japan East, East Asia, Southeast Asia, và Central India**. Các node này chạy trên **Azure App Service** để đảm bảo tính sẵn sàng cao và khả năng vượt qua các rào cản địa lý.
  - *App Service Crawlers:* Các crawler xử lý thông tin tầng ứng dụng (Certificate, HTTP Header, HTTP Redirection, URL Lexical) được triển khai dưới dạng Web Apps (Docker Container).
  - *VM Crawlers:* Các crawler yêu cầu quyền truy cập mạng cấp thấp (*Network Crawler*) được triển khai trên **Azure Virtual Machines**. Các container tại đây được cấu hình với quyền `NET_ADMIN` và chế độ mạng `host` để thực hiện các đo đạc TCP/IP chính xác.
- **Cơ sở dữ liệu (Data Persistence):** Sử dụng **Azure Cosmos DB for MongoDB (vCore)** làm kho lưu trữ trung tâm. Dữ liệu thô từ mỗi crawler được lưu trữ trong các database riêng biệt và được tự động đánh chỉ mục (indexing) trên trường `url` sau khi đã mã hóa để tối ưu hóa tốc độ truy vấn.
- **ETL Pipeline Service:** Một dịch vụ chuyên biệt có nhiệm vụ truy vấn dữ liệu từ các cơ sở dữ liệu, sau đó thực hiện hợp nhất, tính toán thành các nhóm đặc trưng riêng biệt và lưu vào các cơ sở dữ liệu phục vụ cho quá trình huấn luyện mô hình.
- **Model Trainer Service:** Thành phần này nhận dữ liệu đã qua xử lý từ ETL pipeline để thực hiện các công đoạn cân bằng dữ liệu (SMOTE), tối ưu hóa siêu tham số (Random Search CV), huấn luyện và lưu lại mô hình cuối cùng.
- **API dự đoán và tiện ích mở rộng trình duyệt:** Một API service nhận đầu vào là một URL từ tiện ích mở rộng trình duyệt, thực hiện quy trình trích

xuất đặc trưng và sử dụng mô hình đã huấn luyện để trả về kết quả. Tiện ích mở rộng trình duyệt sẽ hiển thị cảnh báo đỏ nếu phát hiện lừa đảo.

Việc quản lý cấu hình và triển khai được thực hiện thông qua các tập lệnh tự động hóa, đảm bảo tính nhất quán giữa môi trường phát triển và môi trường thực tế (production).

4.4.2 Môi trường phần mềm và Tự động hóa (DevOps)

Môi trường phát triển và vận hành được chuẩn hóa bằng các công nghệ và quy trình tự động hóa sau:

- **Tự động hóa hạ tầng (Infrastructure Automation):** Sử dụng `PowerShell` kết hợp với **Azure CLI** để quản lý vòng đời ứng dụng:
  - *Cấu hình dịch vụ:* Các script tự động thiết lập biến môi trường, kết nối cơ sở dữ liệu và triển khai Docker Image mới nhất từ Docker Hub lên Azure App Service và VM.
  - *Quản lý Database:* Script tự động kết nối đến Cosmos DB để khởi tạo collection và xây dựng index tối ưu.
  - *Giám sát:* Công cụ theo dõi log tập trung cho phép truy cập thời gian thực vào Log Stream của App Service và kết nối SSH an toàn vào VM.
- **Container hóa:** `Docker` là nền tảng chạy chính cho tất cả các microservices.
- **Ngôn ngữ lập trình:** `Python 3.x`.
- **Các thư viện Khoa học dữ liệu chính:**
  - `Pandas` & `NumPy`: Để xử lý và thao tác với dữ liệu dạng bảng.
  - `Scikit-learn`: Cung cấp các công cụ nền tảng cho học máy như `Pipeline` và các hàm đánh giá (metrics).

  - `LGBMClassifier và RandomForestClassifier`: Các thuật toán được sử dụng để xây dựng mô hình.
  - `imbalanced-learn`: Thư viện cung cấp kỹ thuật `SMOTE` để xử lý mất cân bằng dữ liệu.
  - `SQLAlchemy`: Để tương tác với các cơ sở dữ liệu PostgreSQL từ Python.
  - `Joblib`: Để lưu và tải lại mô hình đã huấn luyện.
- **Cơ sở dữ liệu:** `Azure Cosmos DB for MongoDB (vCore)` cho dữ liệu thô và `PostgreSQL` cho dữ liệu đã xử lý.

4.4.3 Môi trường phần cứng (Cấu hình đề xuất)

Để đảm bảo quá trình huấn luyện và xử lý dữ liệu diễn ra hiệu quả, môi trường thử nghiệm được đề xuất với cấu hình phần cứng tối thiểu như sau:

- **CPU:** Vi xử lý đa lõi (khuyến nghị từ 8 lõi trở lên) để tận dụng khả năng xử lý song song của các thư viện khoa học dữ liệu.
- **RAM:** Tối thiểu 16GB, khuyến nghị 32GB trở lên để chứa và xử lý các bộ dữ liệu lớn trong bộ nhớ mà không gặp phải tình trạng tràn bộ nhớ.
- **Lưu trữ:** Ổ cứng SSD (Solid State Drive) để tăng tốc độ đọc/ghi dữ liệu, đặc biệt là các thao tác với cơ sở dữ liệu và các tệp dữ liệu lớn.
- **GPU:** Mặc dù LightGBM và Random Forest chủ yếu tối ưu cho CPU, việc có GPU không phải là yêu cầu bắt buộc nhưng có thể được tận dụng để tăng tốc quá trình huấn luyện nếu sử dụng các phiên bản thư viện hỗ trợ GPU trên các bộ dữ liệu cực lớn.

4.5 Tiện ích mở rộng trình duyệt

Để đưa mô hình vào ứng dụng thực tế bảo vệ người dùng cuối, nghiên cứu đã phát triển một tiện ích mở rộng trên trình duyệt (Browser Extension). Tiện ích

này đóng vai trò là cầu nối giữa người dùng và hệ thống phân tích phía backend, cung cấp phản hồi trực quan theo thời gian thực.

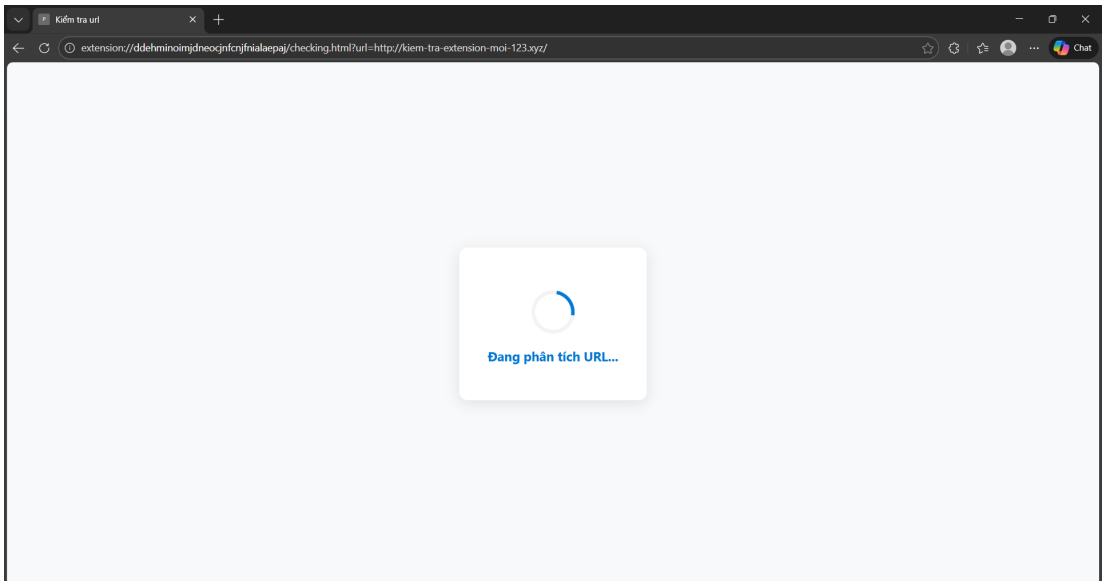


Figure 4.1: Giao diện tiện ích khi đang gửi yêu cầu và phân tích URL

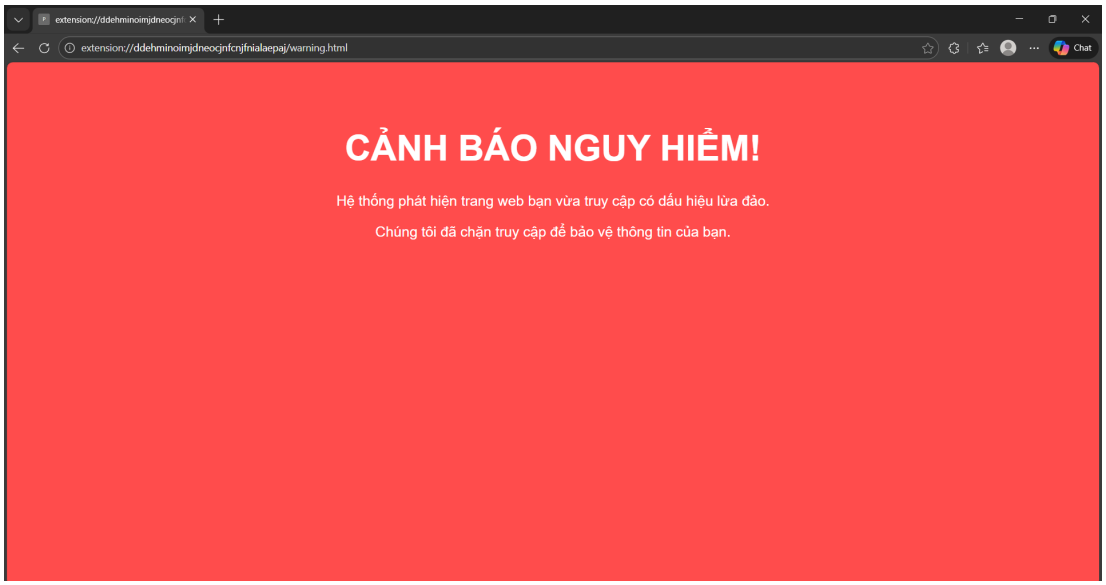


Figure 4.2: Giao diện cảnh báo đỏ khi hệ thống phát hiện URL lừa đảo

Tiện ích mở rộng hoạt động dựa trên cơ chế chặn và kiểm tra tự động mọi yêu cầu điều hướng của người dùng. Để tối ưu hiệu năng, tiện ích sử dụng một danh sách tĩnh chứa khoảng 100.000 domain phổ biến nhất từ bộ dữ liệu Majestic Million. Các domain trong danh sách này được tự động cho phép truy cập mà không cần kiểm tra qua backend, giảm thiểu độ trễ và tải cho hệ thống. Đối với các URL không nằm trong danh sách tĩnh, tiện ích sử dụng API

`declarativeNetRequest` của Chrome để chặn yêu cầu điều hướng và chuyển hướng đến trang kiểm tra trung gian. Tại đây, URL được gửi đến API backend để phân tích bằng mô hình học máy. Nếu kết quả dự đoán là lừa đảo, hệ thống sẽ hiển thị trang cảnh báo màu đỏ và chặn truy cập. Ngược lại, nếu URL được xác định là an toàn, tiện ích sẽ tự động thêm quy tắc ngoại lệ động cho domain đó và cho phép người dùng truy cập bình thường, đồng thời lưu trữ thông tin này để tránh kiểm tra lặp lại cho các lần truy cập sau.

Chương 5

Thực nghiệm và đánh giá

5.1 Thiết lập thử nghiệm

Mục tiêu của chương này là tiến hành các thử nghiệm một cách có hệ thống để đánh giá hiệu năng của mô hình đề xuất và so sánh trực tiếp với mô hình cơ sở. Phần này sẽ trình bày chi tiết về cách thức chuẩn bị dữ liệu, các độ đo được sử dụng để đánh giá, các kịch bản so sánh và quy trình thực nghiệm tổng thể.

5.1.1 Tập dữ liệu

Tập dữ liệu được sử dụng trong thực nghiệm được xây dựng bằng cách tổng hợp từ các nguồn đã được mô tả ở các chương trước.

- **Nguồn dữ liệu:**
  - **Lớp hợp pháp (Legitimate, label=0):** Gồm các tên miền từ danh sách *Majestic Million*, là tập hợp các trang web phổ biến và được tin cậy nhất trên toàn cầu.
  - **Lớp lừa đảo (Phishing, label=1):** Gồm các URL được thu thập và xác minh từ các nguồn công khai uy tín như *PhishTank*.
- **Trích xuất đặc trưng:** Toàn bộ các URL thô từ hai nguồn trên được đưa qua hệ thống thu thập dữ liệu để trích xuất bộ đặc trưng đa dạng như đã mô

tả chi tiết tại **Chương 4 (Mục 4.2.1)**.

- **Phân chia dữ liệu:** Toàn bộ bộ dữ liệu sau xử lý sẽ được phân chia thành hai tập riêng biệt theo tỉ lệ 80/20:
  - **Tập Huấn luyện (Training Set - 80%):** Tập dữ liệu này được sử dụng cho mọi hoạt động liên quan đến việc xây dựng mô hình, bao gồm việc áp dụng SMOTE, chạy xác thực chéo (Cross-Validation) và tìm kiếm siêu tham số (Hyperparameter Tuning).
  - **Tập Kiểm tra (Test Set - 20%):** Tập dữ liệu này được **giữ riêng hoàn toàn** và không được sử dụng trong bất kỳ công đoạn huấn luyện hay tinh chỉnh nào. Nó chỉ được dùng một lần duy nhất ở bước cuối cùng để đánh giá hiệu năng trên dữ liệu mới của các mô hình đã được huấn luyện xong.

  Phép chia này được thực hiện bằng kỹ thuật **chia có phân tầng (Stratified Splitting)**. Kỹ thuật này đảm bảo rằng tỉ lệ mẫu giữa lớp phishing và legitimate trong cả tập huấn luyện và tập kiểm tra là tương đồng với tỉ lệ trong tập dữ liệu gốc. Điều này cực kỳ quan trọng đối với các bài toán có dữ liệu mất cân bằng.

5.1.2 Các độ đo đánh giá (Evaluation Metrics)

Các độ đo hiệu năng chi tiết bao gồm Accuracy, Precision, Recall, F1-Score và diện tích dưới đường cong ROC (AUC) đã được trình bày chi tiết về mặt lý thuyết và công thức toán học tại **Chương 3 (Mục 3.4)**.

Trong khuôn khổ thực nghiệm này, do tính chất mất cân bằng của dữ liệu (lớp hợp pháp chiếm đa số), việc chỉ sử dụng độ chính xác tổng thể (Accuracy) sẽ không phản ánh đúng hiệu quả của mô hình. Do đó, nghiên cứu sẽ tập trung vào các chỉ số sau:

- **Recall (Độ phủ):** Được ưu tiên hàng đầu để đảm bảo giảm thiểu tối đa việc bỏ sót các trang web lừa đảo (False Negative).

- **F1-Score:** Được sử dụng làm hàm mục tiêu trong quá trình tối ưu hóa siêu tham số để cân bằng giữa độ chính xác và độ phủ.
- **Ma trận nhầm lẫn (Confusion Matrix):** Để trực quan hóa chi tiết các trường hợp dự đoán đúng và sai của mô hình.

5.1.3 Các kịch bản thực nghiệm

Thực nghiệm được thiết kế để so sánh hiệu năng giữa hai thuật toán học máy mạnh mẽ là Random Forest và LightGBM. Để đảm bảo tính công bằng và hiệu quả trên tập dữ liệu mất cân bằng, cả hai kịch bản đều được tích hợp trong một quy trình xử lý thống nhất bao gồm tiền xử lý, sinh dữ liệu nhân tạo và tối ưu hóa siêu tham số.

1. **Kịch bản 1: Mô hình Random Forest**
    - **Thuật toán:** `RandomForestClassifier` từ thư viện Scikit-learn.
    - **Xử lý dữ liệu:** Áp dụng kỹ thuật `SMOTE` (Synthetic Minority Over-sampling Technique) trong quy trình `ImbPipeline` để cân bằng lại nhãn dữ liệu trước khi huấn luyện.
    - **Tối ưu hóa siêu tham số:** Sử dụng kỹ thuật tìm kiếm ngẫu nhiên `RandomizedSearchCV` với kiểm định chéo 3 lần (3-fold cross-validation).
    - **Không gian tham số:**
        - `n_estimators`: 100 đến 300.
        - `max_depth`: [10, 20, 30, None].
        - `min_samples_split`: 2 đến 10.
        - `min_samples_leaf`: 1 đến 4.
2. **Kịch bản 2: Mô hình LightGBM**
    - **Thuật toán:** `LGBMClassifier` (Light Gradient Boosting Machine).

- **Xử lý dữ liệu:** Tương tự kịch bản 1, sử dụng `SMOTE` để xử lý mất cân bằng dữ liệu, đảm bảo mô hình học được các đặc trưng của lớp thiểu số (phishing).
- **Tối ưu hóa siêu tham số:** Sử dụng `RandomizedSearchCV` để tìm bộ tham số tốt nhất tối đa hóa điểm số F1-score.
- **Không gian tham số:**
  - `n_estimators`: 100 đến 300.
  - `learning_rate`: 0.01 đến 0.2.
  - `num_leaves`: 20 đến 50.
  - `subsample`: 0.6 đến 1.0.

5.1.4 Quy trình thực nghiệm

1. **Chuẩn bị dữ liệu:** Tổng hợp dữ liệu từ các cơ sở dữ liệu PostgreSQL và gán nhãn dựa trên danh sách URL hợp lệ (từ Majestic Million) và URL lừa đảo (Phishtank). Dữ liệu sau khi làm sạch và tích hợp được hợp nhất thành một bộ dữ liệu hoàn chỉnh.

2. **Phân chia dữ liệu:** Tập dữ liệu được chia thành tập Huấn luyện (80%) và tập Kiểm tra (20%) sử dụng phương pháp lấy mẫu phân tầng (stratified sampling) dựa trên nhãn mục tiêu để đảm bảo tỷ lệ các lớp dữ liệu được giữ nguyên ở cả hai tập.

3. **Xây dựng Pipeline xử lý và cân bằng dữ liệu:** Thiết lập một quy trình xử lý (pipeline) tự động bao gồm các bước:
   - **Tiền xử lý:** Mã hóa các biến phân loại (categorical variables) sử dụng `OrdinalEncoder` và giữ nguyên các biến số.
   - **Cân bằng dữ liệu:** Áp dụng kỹ thuật SMOTE (Synthetic Minority Over-sampling Technique) để sinh dữ liệu nhân tạo cho lớp thiểu số (lớp lừa đảo), giải quyết vấn đề mất cân bằng dữ liệu ngay trong quá trình huấn luyện.

- **Mô hình phân lớp:** Tích hợp bộ phân lớp (Random Forest hoặc LightGBM) vào cuối pipeline.

4. **Tinh chỉnh siêu tham số (Hyperparameter Tuning):** Thực hiện tối ưu hóa tham số cho cả hai mô hình ứng viên: Random Forest và LightGBM.
   - Sử dụng phương pháp tìm kiếm ngẫu nhiên (`RandomizedSearchCV`) với số lượng vòng lặp là 10 ($n\_iter = 10$).
   - Quá trình đánh giá sử dụng kiểm chứng chéo 3 lần (3-fold cross-validation). Trong mỗi fold, dữ liệu huấn luyện được cân bằng lại bằng SMOTE trước khi đưa vào mô hình để đảm bảo tính khách quan và tránh rò rỉ dữ liệu.
   - Hàm mục tiêu để tối ưu hóa là chỉ số F1-score.
5. **Huấn luyện và Đánh giá trên tập kiểm tra:** Đối với từng mô hình (Random Forest và LightGBM), sau khi tìm được bộ siêu tham số tối ưu thông qua quá trình tìm kiếm ngẫu nhiên (RandomizedSearchCV), mô hình đó sẽ được sử dụng để đưa ra dự đoán trên tập Kiểm tra (20% dữ liệu độc lập). Kết quả dự đoán này sẽ được dùng để tính toán các chỉ số hiệu năng cho từng mô hình riêng biệt.
6. **Phân tích kết quả:**
   - Tính toán các độ đo hiệu năng chi tiết: Precision, Recall, F1-score và Accuracy thông qua `classification_report`.
   - Trực quan hóa Ma trận nhầm lẫn (Confusion Matrix) để phân tích các trường hợp dự đoán sai.
   - Vẽ đường cong ROC và tính chỉ số AUC để so sánh khả năng phân tách giữa các lớp của hai mô hình.
7. **Lựa chọn mô hình tối ưu:** So sánh F1-score trên tập kiểm tra giữa Random Forest và LightGBM, từ đó chọn ra mô hình có hiệu năng cao nhất để lưu lại (file .pkl) và triển khai.

5.2 Kết quả so sánh các mô hình

Sau khi tiến hành thực nghiệm theo các kịch bản đã thiết lập, kết quả hiệu năng của hai mô hình trên tập Kiểm tra đã được ghi nhận. Phần này trình bày các kết quả đó, đi kèm với phân tích và bàn luận chi tiết nhằm làm nổi bật hiệu quả của các phương pháp cải tiến.

5.2.1 Bảng tổng hợp kết quả

```
--- Classification Report (RandomForest) ---
              precision    recall  f1-score   support

           0       0.99      0.98      0.99     19984
           1       0.91      0.93      0.92      3598

    accuracy                           0.98     23582
   macro avg       0.95      0.96      0.95     23582
weighted avg       0.98      0.98      0.98     23582
```

```
--- Classification Report (LightGBM) ---
              precision    recall  f1-score   support

           0       0.99      0.98      0.99     19984
           1       0.91      0.94      0.93      3598

    accuracy                           0.98     23582
   macro avg       0.95      0.96      0.96     23582
weighted avg       0.98      0.98      0.98     23582
```

Bảng dưới đây tổng hợp các giá trị độ đo chính được tính toán trên tập kiểm tra cho cả hai mô hình sau khi đã áp dụng kỹ thuật SMOTE và tinh chỉnh siêu tham số. Dữ liệu cho thấy LightGBM đạt hiệu năng nhỉnh hơn so với Random Forest.

Table 5.1: Bảng so sánh hiệu năng giữa các mô hình trên tập kiểm tra

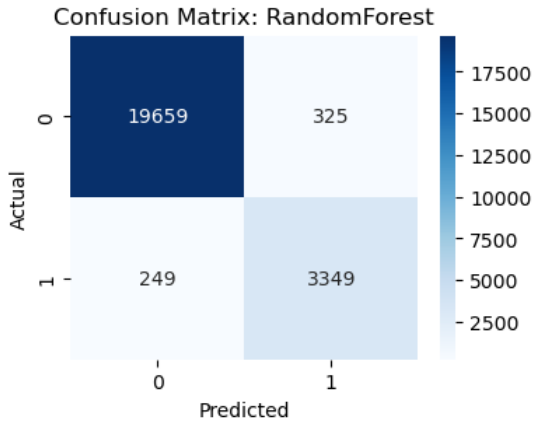
Độ đo (Metric)	**Mô hình cơ sở (Random Forest)**	**Mô hình đề xuất (LightGBM)**
Accuracy	0.98	0.98
Precision (lớp 1)	0.91	0.91
Recall (lớp 1)	0.93	**0.94**
F1-Score (lớp 1)	0.92	**0.93**

5.2.2 Phân tích Ma trận nhầm lẫn

Để đánh giá chi tiết hiệu năng của từng mô hình, kết quả dự đoán trên tập kiểm tra gồm 23,582 mẫu (với 3,598 mẫu phishing và 19,984 mẫu legitimate) được

phân tích cụ thể như sau.

5.2.2.1 Mô hình cơ sở (Random Forest)

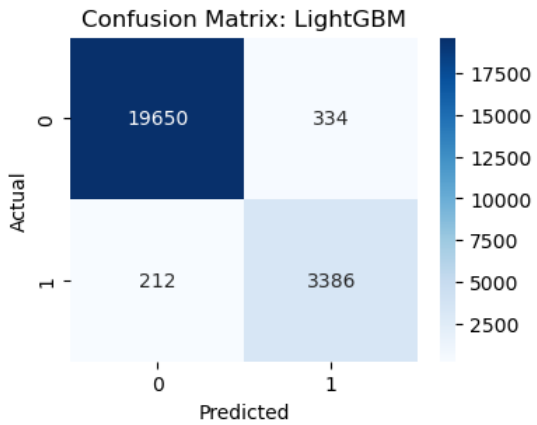


Ma trận nhầm lẫn cho thấy:

- **True Positives:** 3349
- **False Negatives:** 249
- **True Negatives:** 19659
- **False Positives:** 325

Mô hình này đã **bỏ sót 249 trang phishing**, một con số rất đáng kể, rủi ro cao cho người dùng.

5.2.2.2 Mô hình đề xuất (LightGBM)



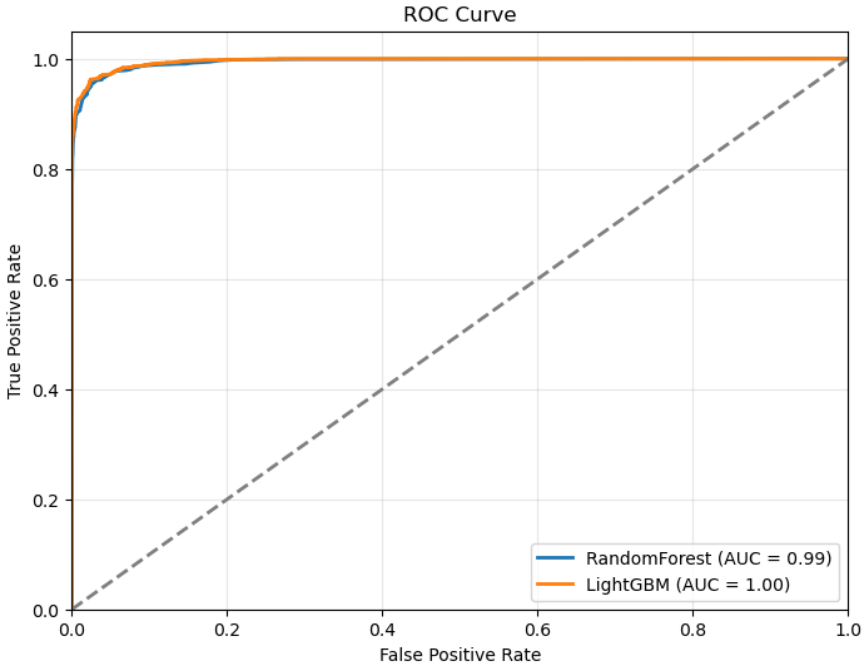
Ma trận nhầm lẫn cho thấy:

- **True Positives:** 3386
- **False Negatives:** 212
- **True Negatives:** 19650
- **False Positives:** 334

So với Random Forest, LightGBM đã **giảm số lượng trang phishing bị bỏ sót từ 249 xuống còn 212**. Đồng thời, số lượng các trang web hợp lệ bị nhận diện nhầm (False Positives) chỉ tăng nhẹ không đáng kể (từ 325 lên 334). Điều này cho thấy LightGBM nhạy hơn trong việc phát hiện các mối đe dọa thực sự mà không làm ảnh hưởng nhiều đến trải nghiệm người dùng, khẳng định đây là mô hình tối ưu hơn cho bài toán này.

5.2.3 Phân tích biểu đồ ROC và chỉ số AUC

Bên cạnh các chỉ số độ chính xác (Accuracy, Precision, Recall), biểu đồ ROC (Receiver Operating Characteristic) và chỉ số AUC (Area Under the Curve) được sử dụng để đánh giá khả năng phân tách giữa hai lớp (phishing và legitimate) của các mô hình tại các ngưỡng phân loại khác nhau.



Quan sát biểu đồ ROC, có thể thấy cả hai mô hình đều thể hiện hiệu năng rất cao, với đường cong bám sát góc trên bên trái của biểu đồ. Cụ thể:

- **Mô hình Random Forest:** Đạt chỉ số AUC = **0.99**. Kết quả này cho thấy mô hình cơ sở có khả năng phân loại rất tốt, duy trì tỷ lệ dương tính thật (True Positive Rate) cao ngay cả khi tỷ lệ dương tính giả (False Positive Rate) thấp.
- **Mô hình LightGBM:** Đạt chỉ số AUC = **1.00** (làm tròn). Đường cong của LightGBM (màu cam) nằm đè lên và bao phủ đường cong của Random Forest (màu xanh), cho thấy khả năng phân tách các lớp dữ liệu gần như hoàn hảo.

Kết luận: Việc LightGBM đạt AUC tối đa (1.00) so với 0.99 của Random Forest khẳng định tính ổn định và độ tin cậy vượt trội của mô hình đề xuất. Điều này đặc biệt quan trọng trong bài toán phát hiện lừa đảo, nơi mô hình cần phải cực kỳ nhạy bén để phát hiện các mối đe dọa nhưng vẫn phải đảm bảo không làm phiền người dùng bằng các cảnh báo sai. Kết quả phân tích ROC/AUC hoàn toàn thống nhất với sự cải thiện về F1-Score đã trình bày ở Bảng 5.1.

5.3 Phân tích hiệu suất và độ tin cậy

```
=================== Đang xử lý RandomForest ===================
Best Params: {'classifier__max_depth': None, 'classifier__min_samples_leaf': 1, 'classifier__min_samples_split': 2, 'classifier__n_estimators': 157}
Thời gian huấn luyện: 521.8033 giây
Thời gian suy luận trên tập Test (23582 mẫu): 0.7647 giây
Tốc độ trung bình: 0.0324 ms/mẫu
```

```
=================== Đang xử lý LightGBM ===================
Best Params: {'classifier__learning_rate': 0.1516145155592091, 'classifier__n_estimators': 249, 'classifier__num_leaves': 40, 'classifier__subsample': 0.9879
Thời gian huấn luyện: 67.3944 giây
Thời gian suy luận trên tập Test (23582 mẫu): 0.2592 giây
Tốc độ trung bình: 0.0110 ms/mẫu
```

Bên cạnh các chỉ số chất lượng dự đoán, yếu tố quyết định khả năng ứng dụng thực tế của hệ thống là hiệu suất tính toán.

5.3.1 Thời gian suy diễn và độ trễ (Inference Latency)

Kết quả đo đạc trên tập kiểm tra (23,582 mẫu) cho thấy sự chênh lệch rõ rệt:

- **Đối với mô hình Random Forest:**
  - **Tổng thời gian dự đoán: 0.7647 giây** cho toàn bộ tập dữ liệu.
  - **Tốc độ xử lý trung bình: 0.0324 ms/mẫu**.
  - **Nguyên nhân độ trễ:**

    Mô hình phải duyệt qua 157 cây quyết định độc lập nên tốn nhiều thời gian tính toán hơn.

- **Đối với mô hình LightGBM (Đề xuất):**
  - **Tổng thời gian dự đoán:** Chỉ **0.2592 giây** cho cùng lượng dữ liệu.
  - **Tốc độ xử lý trung bình: 0.0110 ms/mẫu**.
  - **Đánh giá hiệu năng:**

    Tốc độ suy diễn nhanh gấp gần **3 lần** so với Random Forest, hoàn toàn đáp ứng được yêu cầu xử lý thời gian thực đã đặt ra tại **Mục 2.4.3 (Thách thức về tốc độ)**.

5.3.2 Hiệu quả huấn luyện (Training Efficiency)

Chi phí thời gian cho việc huấn luyện và cập nhật mô hình cũng được ghi nhận sự khác biệt lớn:

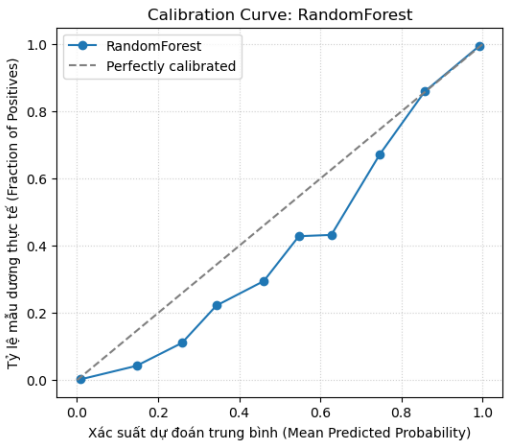
- **Mô hình Random Forest:**
  - **Thời gian thực hiện:** Quá trình tìm kiếm tham số và huấn luyện tiêu tốn **521.8033 giây**.
  - **Nhược điểm:** Tiêu tốn tài nguyên tính toán, gây khó khăn nếu cần cập nhật mô hình thường xuyên.

- **Mô hình LightGBM:**
  - **Thời gian thực hiện:** Hoàn thành quy trình tương tự chỉ trong **67.3944 giây**.
  - **Ưu điểm vượt trội:** Tốc độ huấn luyện nhanh gấp gần **8 lần** so với Random Forest. Điều này cho phép hệ thống tái huấn luyện hàng ngày với chi phí thấp để cập nhật các mẫu tấn công mới.

5.3.3 Phân tích độ tin cậy xác suất (Calibration Curve Analysis)

Bên cạnh khả năng phân lớp nhị phân (0 hoặc 1),chất lượng của xác suất dự đoán còn được đánh giá thông qua đường cong hiệu chỉnh. Điều này đặc biệt quan trọng nếu hệ thống sử dụng ngưỡng xác suất để đưa ra các mức cảnh báo khác nhau.

- **Mô hình Random Forest:**



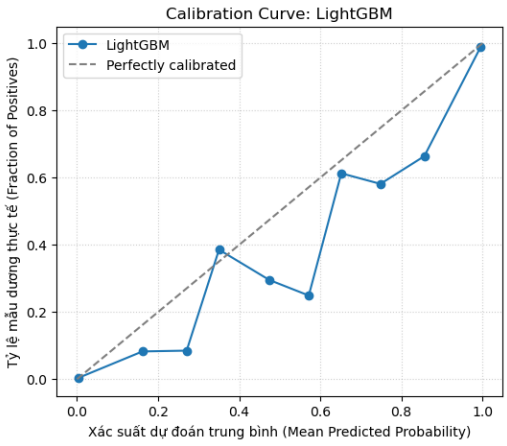
– *Đặc điểm:*

Đường cong hiệu chỉnh có xu hướng bám sát đường chéo lý tưởng ($y = x$).

– *Phân tích:*

Mô hình có độ tin cậy xác suất tốt. Tức là xác suất dự đoán phản ánh khá trung thực tỷ lệ lừa đảo thực tế.

- **Mô hình LightGBM:**



– *Đặc điểm:*

Đường cong thường có hình chữ S nhẹ, thể hiện xu hướng đẩy xác suất về hai cực 0 và 1.

– *Phân tích:*

Mặc dù độ hiệu chỉnh có thể thấp hơn Random Forest một chút, nhưng khả năng phân tách lại tốt hơn (như đã chứng minh qua AUC ≈ 1.0).

– *Ý nghĩa triển khai:*

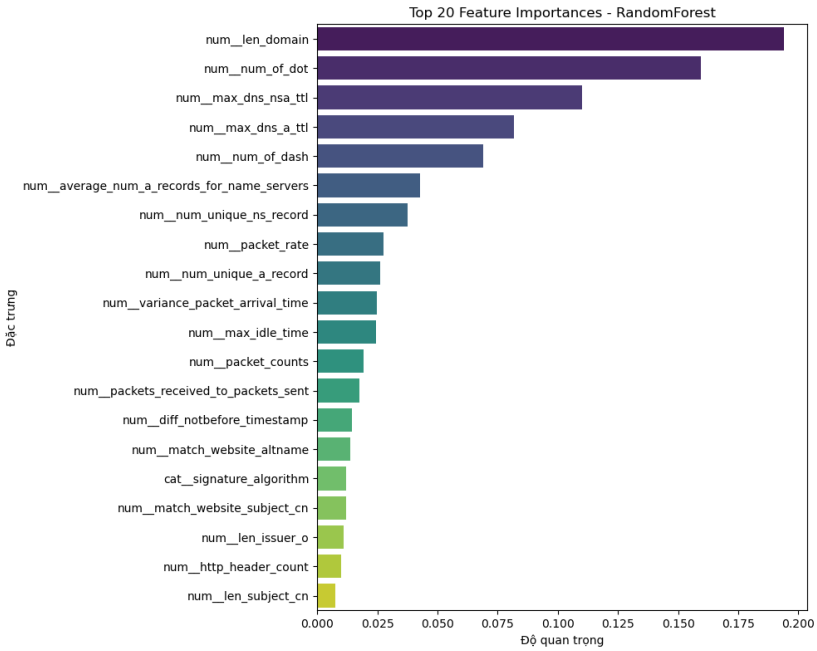
Trong bài toán an ninh mạng cần quyết định dứt khoát (Chặn/Cho phép), tính chất quyết đoán này của LightGBM lại là một ưu điểm, giúp giảm thiểu các trường hợp lửng lơ ở ngưỡng 0.5.

Kết luận: Dù Random Forest có đường hiệu chỉnh mượt hơn, nhưng LightGBM vẫn được ưu tiên lựa chọn vì mục tiêu tối thượng của hệ thống là khả năng phát hiện chính xác chứ không phải là ước lượng xác suất chính xác.

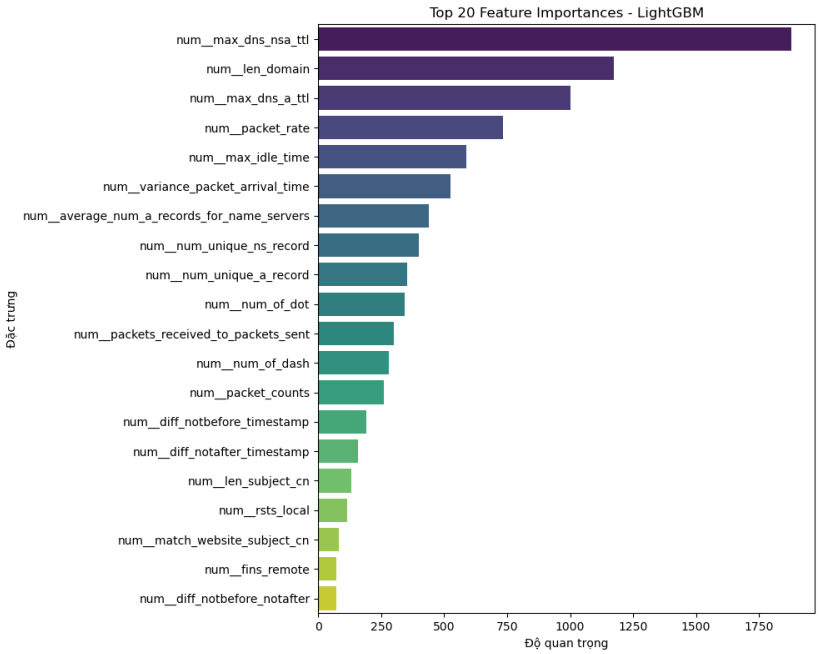
5.3.4 Phân tích độ quan trọng của đặc trưng (Feature Importance)

Việc phân tích độ quan trọng của đặc trưng giúp làm rõ các yếu tố đóng vai trò quyết định trong việc nhận diện hành vi lừa đảo của các mô hình.

- **Đối với mô hình Random Forest:** Các đặc trưng về cấu trúc tên miền chiếm ưu thế rõ rệt. `len_domain` (độ dài tên miền) và `num_of_dot` (số lượng dấu chấm) là hai yếu tố quan trọng nhất. Điều này hợp lý vì các trang web lừa đảo thường sử dụng các tên miền phụ (subdomains) dài và phức tạp để giả mạo các thương hiệu lớn. Ngoài ra, các thông tin về bản ghi DNS (`max_dns_nsa_ttl`) cũng đóng vai trò then chốt.



- **Đối với mô hình LightGBM:** Mô hình này tập trung mạnh mẽ vào các đặc trưng liên quan đến giao thức DNS. Đặc trưng `max_dns_nsa_ttl` có điểm quan trọng vượt trội so với phần còn lại. Điều này cho thấy LightGBM đã khai thác hiệu quả các dấu hiệu bất thường trong thời gian tồn tại (TTL) của các bản ghi DNS - một đặc trưng mà tội phạm mạng thường xuyên thay đổi để tránh bị phát hiện bởi các danh sách đen truyền thống.



- **Sự tương đồng giữa hai mô hình:** Cả hai mô hình đều đánh giá cao nhóm đặc trưng về DNS (`dns_nsa_ttl`, `dns_a_ttl`) và đặc trưng mạng (`packet_rate`, `max_idle_time`). Việc cả hai thuật toán khác nhau đều hội tụ vào các nhóm đặc trưng này khẳng định tính đúng đắn và hiệu quả của quy trình trích xuất đặc trưng đa dạng đã đề xuất tại **Chương 4**.

Nhận xét chung: LightGBM cho thấy khả năng tập trung vào các đặc trưng mang tính hành vi sâu (như thông số DNS và thời gian nhàn rỗi của gói tin) tốt hơn so với việc chỉ dựa vào các đặc trưng bề mặt như cấu trúc URL. Đây chính là lý do giúp LightGBM đạt được chỉ số Recall và F1-score cao hơn, giúp phát hiện các cuộc tấn công phishing tinh vi có tên miền trông giống như thật.

Chương 6

Kết luận và hướng phát triển

6.1 Tóm tắt kết quả đạt được

Nghiên cứu này đã tập trung giải quyết bài toán phát hiện tấn công trên hệ thống bảo mật ứng dụng web (Phishing URL) thông qua việc áp dụng và tối ưu hóa các thuật toán học máy. Quá trình thực hiện đã đi từ việc thu thập, xử lý dữ liệu, trích xuất đặc trưng đến việc xây dựng và tinh chỉnh mô hình. Các kết quả chính đạt được bao gồm:

1. **Xây dựng quy trình xử lý dữ liệu toàn diện:** Đề tài đã thiết lập thành công một pipeline tự động hóa bao gồm các bước: làm sạch dữ liệu, mã hóa đặc trưng và đặc biệt là xử lý mất cân bằng dữ liệu bằng kỹ thuật SMOTE. Điều này giúp mô hình không bị thiên lệch về lớp đa số, đảm bảo độ tin cậy khi phát hiện các mẫu tấn công lừa đảo.

2. **Hiệu năng phân loại vượt trội:** Thông qua thực nghiệm so sánh, mô hình đề xuất sử dụng thuật toán **LightGBM** đã đạt được các chỉ số độ chính xác và F1-Score cao trên tập dữ liệu kiểm tra, khẳng định tính đúng đắn của phương pháp tiếp cận.

3. **Tối ưu hóa khả năng triển khai thực tế:** Kết quả đo đạc cho thấy LightGBM là lựa chọn tối ưu cho các hệ thống thời gian thực với tốc độ dự đoán

và thời gian huấn luyện nhanh hơn so với Random Forest.

6.2 Đóng góp của nghiên cứu

Đồ án đóng góp những giá trị cụ thể về mặt khoa học và thực tiễn trong lĩnh vực an toàn thông tin:

- **Về mặt thực tiễn:** Cung cấp một giải pháp phát hiện lừa đảo đầu cuối có khả năng tự động hóa hoàn toàn từ khâu nhận URL đến khi đưa ra dự đoán gần như tức thời.
- **Về mặt kỹ thuật:** Chứng minh tính hiệu quả vượt trội của thuật toán LightGBM trong việc xử lý các bộ đặc trưng đa dạng và dữ liệu mất cân bằng so với các phương pháp truyền thống.
- **Sản phẩm ứng dụng:** Mô hình được tích hợp vào một tiện ích mở rộng trình duyệt (Browser Add-on), giúp bảo vệ người dùng cuối một cách trực quan và hiệu quả.

6.3 Hạn chế và hướng nghiên cứu tiếp theo

Mặc dù đã đạt được những kết quả khả quan, nghiên cứu vẫn còn tồn tại một số hạn chế cần khắc phục:

6.3.1 Hạn chế

- **Dữ liệu tĩnh:** Mô hình hiện tại có thể bị suy giảm hiệu năng theo thời gian nếu không được tái huấn luyện thường xuyên để thích nghi với các mẫu tấn công mới.
- **Hạn chế về phạm vi phát hiện:** Giải pháp hiện tại tập trung chủ yếu vào việc phân loại website giả mạo dựa trên metadata và hạ tầng mạng. Do đó, khả năng kiểm tra sâu vào payload để phát hiện các mã độc hoặc các câu lệnh tấn công kỹ thuật chèn vào gói tin vẫn còn hạn chế.

- **Chưa phân tích nội dung trang:** Do hạn chế về tài nguyên, đồ án chưa triển khai phân tích cấu trúc DOM và hình ảnh trực quan, vốn là những đặc trưng quan trọng để phát hiện các trang giả mạo tinh vi.

6.3.2 Hướng phát triển

Để nâng cao hơn nữa hiệu quả bảo mật, các hướng nghiên cứu tiếp theo được đề xuất bao gồm:

1. **Giải quyết các tấn công khai thác lỗ hổng trực tiếp:** Đây là hướng đi trọng tâm nhằm mở rộng khả năng của hệ thống. Nghiên cứu sẽ tập trung vào việc phân tích payload để phát hiện các kỹ thuật tấn công như SQL Injection (SQLi), Cross-Site Scripting (XSS), Local File Inclusion (LFI) và Remote File Inclusion (RFI). Hướng tiếp cận này yêu cầu tích hợp thêm các kỹ thuật NLP tiên tiến để hiểu sâu hơn về cấu trúc ngữ nghĩa của các câu lệnh độc hại.

2. **Phân tích đa phương thức (Multimodal Analysis):** Kết hợp phân tích đặc trưng tĩnh hiện tại với phân tích cấu trúc DOM và nhận diện hình ảnh để tăng cường khả năng chống giả mạo giao diện.

3. **Tích hợp Học sâu (Deep Learning):** Sử dụng các mạng nơ-ron như CNN hoặc LSTM để tự động trích xuất các mẫu tấn công phức tạp từ luồng dữ liệu thô mà không cần can thiệp thủ công.

4. **Cơ chế học trực tuyến (Online Learning):** Xây dựng hệ thống tự động cập nhật mô hình định kỳ nhằm đối phó hiệu quả với hiện tượng trôi dạt khái niệm (Concept Drift).

Tài liệu tham khảo

[1] Othmane Belarbi, Theodoros Spyridopoulos, Eirini Anthi, Ioannis Mavromatis, Pietro Carnelli, and Aftab Khan. Federated deep learning for intrusion detection in iot networks. *arXiv preprint arXiv:2306.02715*, 2023.

[2] Asmaa Halbouni, Teddy Surya Gunawan, Mohamed Hadi Habaebi, Murad Halbouni, Mira Kartiwi, and Robiah Ahmad. Cnn-lstm: Hybrid deep neural network for network intrusion detection system. *IEEE Access*, 10:99837–99849, 2022.

[3] Hao Sun, Yuejin Du, and Qi Li. Deep learning-based detection technology for sql injection research and implementation. *Applied Sciences*, 13(16), 2023.

[4] Chao Wanga, Yunxiao Sun, Wenting Wang, Hongri Liu, and Bailing Wang. Hybrid intrusion detection system based on combination of random forest and autoencoder. *Symmetry*, 15(3), 2023.

[5] Peilun Wu, Hui Guo, and Richard Buckland. A transfer learning approach for network intrusion detection. *arXiv preprint arXiv:1909.02352*, 2019.

[6] Yaokai Yang. Effective phishing detection using machine learning approach. Master's thesis, Case Western Reserve University, January 2019.