TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH

KHOA CÔNG NGHỆ THÔNG TIN

-----🙤🕮🙦-----

A blue book with a ring around it

Description automatically generated

BÁO CÁO MÔN HỌC

PHÂN TÍCH DỮ LIỆU

ĐỀ TÀI: Phát hiện trang web lừa đảo (Web Phishing)

bằng phương pháp học máy

**Sinh viên thực hiện:** ĐÀO NGUYÊN HƯNG

**Giảng viên hướng dẫn:** Th.S HỒ HƯỚNG THIÊN

**Khoa:** Công nghệ thông tin

**Lớp:** DH22CS01

**Khóa:** 2022

*Thành phố Hồ Chí Minh, ngày 26 tháng 4 năm 2024*

MỤC LỤC

[**LỜI MỞ ĐẦU 1**](#_Toc196773051)

[**CHƯƠNG I: TỔNG QUAN ĐỀ TÀI 2**](#_Toc196773052)

[**1. Tổng quan đề tài và lý do chọn đề tài 2**](#_Toc196773053)

[**2. Phương pháp thực hiện 2**](#_Toc196773054)

[**CHƯƠNG 2: DỮ LIỆU VÀ TIỀN XỬ LÝ 5**](#_Toc196773055)

[**1.Thu thập dữ liệu 5**](#_Toc196773056)

[**2.Tiền xử lý dữ liệu 6**](#_Toc196773057)

[**2.1.Lọc và chuẩn hóa dữ liệu hợp pháp 6**](#_Toc196773058)

[**2.2.Hợp nhất và làm sạch dữ liệu lừa đảo 6**](#_Toc196773059)

[**2.3. Kết hợp dữ liệu 7**](#_Toc196773060)

[**2.4. Kiểm tra sự cân bằng của bộ dữ liệu 7**](#_Toc196773061)

[**3. Trích xuất đặc trưng 9**](#_Toc196773062)

[**3.1. Các đặc trưng trích xuất từ URL 9**](#_Toc196773063)

[**3.2. Trích xuất đặc trưng 9**](#_Toc196773064)

[**4.Chia tập dữ liệu 11**](#_Toc196773065)

[**CHƯƠNG 3: HUẤN LUYỆN MÔ HÌNH 12**](#_Toc196773066)

[**I. Mô hình học máy 12**](#_Toc196773067)

[**1. Decision Tree (Cây quyết định) 12**](#_Toc196773068)

[**2. Random Forest 12**](#_Toc196773069)

[**3. XGBoost (eXtreme Gradient Boosting) 12**](#_Toc196773070)

[**4. CatBoost (Categorical Boosting) 13**](#_Toc196773071)

[**II. Quy trình huấn luyện 13**](#_Toc196773072)

[**III. Kết quả 13**](#_Toc196773073)

[**1. Mô hình Decision Tree 13**](#_Toc196773074)

[**2. Mô hình Random Forest 14**](#_Toc196773075)

[**3. Mô hình XGBoost 15**](#_Toc196773076)

[**4. Mô hình Cat Boost 16**](#_Toc196773077)

[**5. So sánh kết quả các mô hình 18**](#_Toc196773078)

[**6. Thử nghiệm hàm Dự đoán (Predict) với một vài đường dẫn 21**](#_Toc196773079)

[**CHƯƠNG 5: XÂY DỰNG ỨNG DỤNG NHẬN DẠNG PHISHING 24**](#_Toc196773080)

[**KẾT LUẬN 30**](#_Toc196773081)

[**TÀI LIỆU THAM KHẢO 31**](#_Toc196773082)

LỜI MỞ ĐẦU

Trong thời đại kỹ thuật số ngày nay, sự phát triển mạnh mẽ của Internet đã và đang lại nhiều tiện ích cho cuộc sống của con người - từ mua sắm trực tuyến, giao dịch ngân hàng đến các dịch vụ giải trí và giáo dục. Tuy nhiên, song song với những lợi ích đó, Internet cũng mở ra cánh cửa cho các hoạt động tội phạm mạng, trong đó lừa đảo qua trang Web (Web Phishing) là một trong những mối đe dọa nghiệm trọng nhất.

Với khả năng ngày càng *“biến hình”* khó lường, các trang web phishing không chỉ bắt chước giao diện mà còn mô phỏng hành vi của các website hợp pháp, khiến người dùng dễ dàng bị sập bẫy. Theo báo cáo của tổ chức an ninh mạng, số lượng các vụ tấn công phishing đã tăng vọt trong những năm gần đây, gây thiệt hại hàng tỷ đô la cho các cá nhân và tổ chức trên toàn thế giới. Vì vậy mà *việc phát hiện sớm và chính xác các trang web lừa đảo* đã trở thành một yêu cầu cấp thiết trong lĩnh vực an ninh mạng.

Trong bối cảnh đó, *học máy (machine learning)* đã nổi lên như một giải pháp đầy hứa hẹn cho việc phát hiện trang web lừa đảo. Không giống như các phương pháp truyền thống dựa trên quy tắc cố định, học máy cho phép hệ thống tự học hỏi từ dữ liệu, nhận diện các mẫu và đặc điểm của các trang Web Phishing một cách linh hoạt và hiệu quả. *Bằng cách sử dụng các thuật toán học máy*, chúng ta có thể xây dựng các mô hình có khả năng phân loại chính xác các trang Web là hợp pháp hay lừa đảo, từ đó nâng cao mức độ an toàn cho người dùng trong môi trường số.

Đề tài “*Phát hiện trang web lừa đảo bằng phương pháp học máy*” được thực hiện nhằm xây dựng một hệ thống phát hiện Web Phishing dựa trên việc trích xuất các đặc trưng có khả năng phân loại tốt từ URL, sau đó huấn luyện mô hình học máy để phân loại trang web một cách tối ưu và chính xác.

Không chỉ dừng lại ở việc xây dựng mô hình, đề tài còn hướng tới việc xây dựng một *giao diện Web thân thiện,* cho phép người dùng dễ dàng nhập URL và kiểm tra mức độ an toàn của liên kết, góp phần nâng cao ý thức và khả năng phòng tránh các cuộc tấn công lừa đảo trên không gian mạng.

CHƯƠNG I: TỔNG QUAN ĐỀ TÀI

1. Tổng quan đề tài và lý do chọn đề tài

Tấn công lừa đảo qua mạng (*phishing*) là một trong những mối đe dọa an ninh mạng nghiêm trọng và phổ biến nhất hiện nay. Kẻ tấn công sử dụng các kỹ thuật giả mạo tinh vi, chẳng hạn như tạo các trang web, email hoặc tin nhắn SMS giả danh các tổ chức uy tín (ngân hàng, nhà mạng, nền tảng thương mại điện tử) nhằm lừa đảo người dùng cung cấp thông tin nhạy cảm như mật khẩu, thông tin tài khoản ngân hàng hoặc dữ liệu cá nhân. Theo báo cáo của *Anti-Phishing Working Group (APWG)*, số lượng các vụ tấn công *phishing* trên toàn cầu tăng mạnh, với hàng triệu URL lừa đảo được phát hiện mỗi năm. Tại Việt Nam, tình hình cũng đáng báo động khi Cục An toàn thông tin ghi nhận khoảng 600–800 website lừa đảo mới mỗi tháng trong năm 2024.

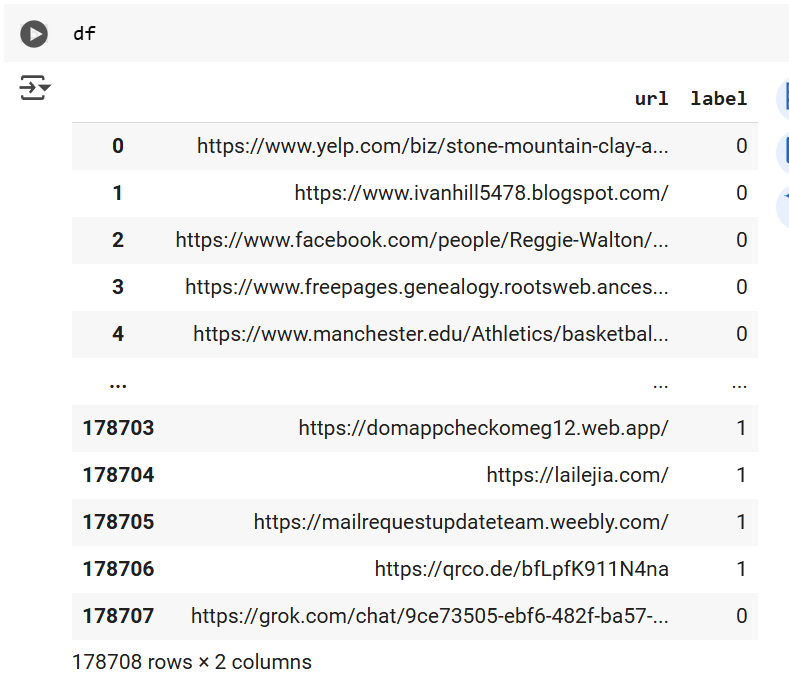
Các phương pháp truyền thống như danh sách đen (*blacklist*) hoặc phân tích thủ công đặc trưng URL tỏ ra kém hiệu quả trước sự tinh vi và tốc độ xuất hiện của các website lừa đảo mới. Trong khi đó, học máy (*machine learning*) đã chứng minh tiềm năng vượt trội nhờ khả năng phân tích dữ liệu lớn (*big data*) và nhận diện các mẫu bất thường (*patterns*). Ví dụ, Google đã áp dụng *Random Forest* đạt độ chính xác 95% trong phát hiện website lừa đảo vào năm 2021, hay trong khi Barracuda Networks sử dụng *XGBoost* để ngăn chặn 90% email lừa đảo. Những thành tựu này khẳng định vai trò của học máy trong việc đối phó với các mối đe dọa *phishing*.

Đề tài tập trung xây dựng và so sánh các mô hình học máy như Decision Tree, Random Forest, XGBoost và CatBoost để nhận diện website phishing, phân tích đặc trưng như độ dài URL hay ký tự đặc biệt nhằm tối ưu hóa hiệu suất. Nghiên cứu xuất phát từ thiệt hại hàng nghìn tỷ đồng mỗi năm tại Việt Nam và tiềm năng ứng dụng thực tiễn, như ngăn chặn vụ giả mạo website Shopee năm 2023 khiến người dùng mất tiền qua thẻ tín dụng.

Với sự quan tâm đến an ninh mạng và trí tuệ nhân tạo, nhóm mong muốn góp một giải pháp thiết thực, nâng cao nhận thức cộng đồng và kỹ năng nghiên cứu. Đề tài không chỉ có giá trị khoa học mà còn mang ý nghĩa thực tiễn trong bối cảnh số hóa ngày càng sâu rộng.

2. Phương pháp thực hiện

Đầu tiên, thu thập dữ liệu là bước quan trọng trong nghiên cứu. Dữ liệu sẽ được thu thập từ các nguồn đáng tin cậy như PhishTank, OpenPhish và bộ dataset có sẵn để lấy các nhãn an toàn. Bộ dữ liệu này sẽ bao gồm các URL của các trang web phishing (được gán nhãn 1) và các trang web hợp pháp (được gán nhãn 0). Dữ liệu thu thập được sẽ được sử dụng để xây dựng và kiểm tra các mô hình học máy.



*Hình 2. Tổng quan dataset*

Tiếp theo, nghiên cứu sẽ tiến hành trích xuất đặc trưng từ URL. Các đặc trưng quan trọng sẽ được trích xuất bao gồm độ dài URL, độ dài domain, độ dài đường dẫn, số lượng các kí tự đặc biệt trong URL và domain, số lượng tham số trong truy vấn, kiểm tra xem có là địa chỉ IP không,... Những đặc trưng này sẽ cung cấp thông tin cần thiết để cải thiện độ chính xác của mô hình phân loại.

Sau khi trích xuất đặc trưng, bước tiếp theo là tiền xử lý dữ liệu. Dữ liệu sẽ được chuẩn hóa để đảm bảo rằng tất cả các đặc trưng đều ở cùng một thang đo. Các bước tiền xử lý bao gồm xử lý các dữ liệu bị thiếu và chia dữ liệu thành hai phần: một phần để huấn luyện mô hình (60%) và một phần để kiểm tra mô hình (40%).

Nghiên cứu sẽ áp dụng các thuật toán học máy trong bước xây dựng mô hình học máy. Các thuật toán như Decision Tree, eXtreme Gradient Boosting (XGBoost), Random Forest, và Cat Boost sẽ được sử dụng để xây dựng các mô hình phát hiện phishing. Mỗi mô hình sẽ được huấn luyện trên tập dữ liệu huấn luyện đã được chuẩn bị.

Sau khi xây dựng và tối ưu hóa các mô hình, bước tiếp theo là đánh giá mô hình. Các mô hình sẽ được đánh giá dựa trên các chỉ số như độ chính xác, độ nhạy, độ đặc hiệu, và F1-score. Kết quả sẽ được so sánh để xác định mô hình nào hoạt động tốt nhất trong việc phát hiện các trang web phishing.

Cuối cùng, đề tài sẽ tiến hành phân tích kết quả để rút ra các kết luận về hiệu quả của từng phương pháp. Đề tài cũng sẽ xem xét các yếu tố ảnh hưởng đến độ chính xác của mô hình và đề xuất các cải tiến cho các nghiên cứu trong tương lai. Bên cạnh đó, đề tài sẽ đưa ra các khuyến nghị và hướng dẫn cho người dùng về cách nhận biết và phòng tránh các trang web phishing, đồng thời cung cấp hướng dẫn cho các nhà phát triển và tổ chức về cách áp dụng các phương pháp phát hiện phishing hiệu quả.

Thông qua các phương pháp nghiên cứu này, đề tài hy vọng sẽ đóng góp vào việc phát triển các giải pháp an ninh mạng hiệu quả hơn, bảo vệ người dùng khỏi các mối đe dọa từ phishing.

CHƯƠNG 2: DỮ LIỆU VÀ TIỀN XỬ LÝ

1. Thu thập dữ liệu

- Bộ dữ liệu các trang web lừa đảo được cào từ hai trang: OpenPhish và PhishTank.

Ảnh có chứa văn bản, ảnh chụp màn hình, số, tài liệu

Nội dung do AI tạo ra có thể không chính xác.

*Hình 3. Dataset URL phishing*

- Bộ dataset các URL hợp pháp được lấy từ dataset *“url\_dataset”* và chỉ lấy các URL có type là *“legitimate”*

*Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, đen và trắng

Nội dung do AI tạo ra có thể không chính xác.*

*Hình 4. Dataset URL hợp pháp*

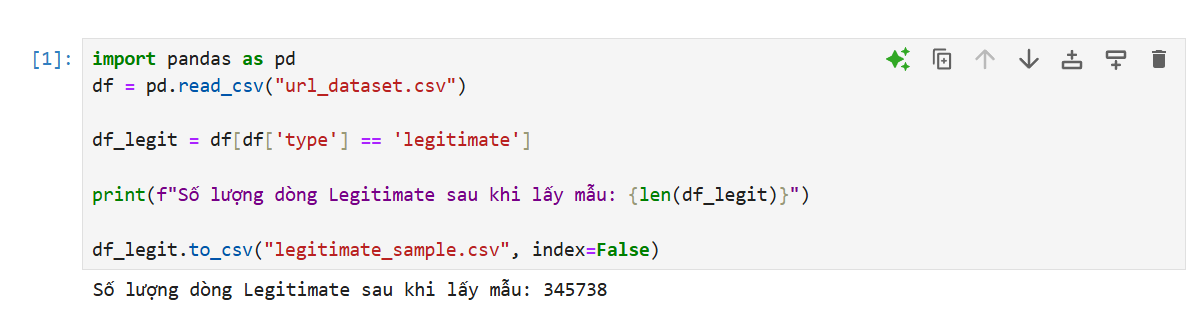
1. Tiền xử lý dữ liệu

Quá trình tiền xử lý dữ liệu được thực hiện qua các bước sau:

* 1. **Lọc và chuẩn hóa dữ liệu hợp pháp**

- Từ file “url\_dataset.csv”, các URL có nhãn *“legitimate”* được trích xuất, với tổng cộng *345.738* mẫu sau khi lấy mẫu.

- File kết quả được lưu vào *“legitimate\_sample.csv”.*



- Cột *“type”* được đổi tên thành *“label”,* và nhãn nhãn *“legitimate”* được thay thế bằng giá trị 0 để biểu thị lớp hợp pháp.

- Kết quả cuối cùng được lưu vào file *“updated\_dataset.csv”.*



* 1. **Hợp nhất và làm sạch dữ liệu lừa đảo**

- Hai file *“phishtank\_urls.csv”* và *“openphish\_urls.csv”* được hợp nhất thành file *“merged\_urls.csv”.*

- Cột URL trong *“openphish\_urls.csv”* được đổi tên thành url để đồng bộ với *“phishtank\_urls.csv”.*

- Thực hiện kiểm tra trùng lặp và loại bỏ, với *3.379 URL trùng lặp* bị xóa.

- Tất cả các URL trong *“merged\_urls.csv”* được gán nhãn 1 (phishing).



**2.3. Kết hợp dữ liệu**

- Dữ liệu hợp pháp *(“updated\_dataset.csv”)* và lừa đảo *(“merged\_urls.csv”)* được hợp nhất thành file *“web\_phishing\_dataset\_1.csv”.*

- Các URL trùng lặp giữa hai tập dữ liệu được loại bỏ để đảm bảo tính duy nhất.

- File cuối cùng chứa các cột chính: *url* (địa chỉ URL) và label (0 cho hợp pháp và 1 cho lừa đảo).

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình

Nội dung do AI tạo ra có thể không chính xác.

**2.4. Kiểm tra sự cân bằng của bộ dữ liệu**

- Sự cân bằng của bộ dữ liệu là yếu tố quan trọng để đảm bảo mô hình học máy không bị *thiên lệch* (bias) hướng về lớp chiếm ưu thế.

- Dựa vào biểu đồ cho thấy, sự phân phối nhãn của bộ dữ liệu có sự chênh lệch giữa hai nhãn và *nhãn 0* chiếm *75.71%* bộ dữ liệu trong khi đó *nhãn 1* chỉ chiếm *24.29%* bộ dữ liệu. Sự mất cân bằng này có thể khiến các mô hình học máy thiên lệch về lớp hợp pháp, dẫn đến việc bỏ sót các URL lừa đảo (*recall* thấp cho lớp 1), đặc biệt với các thuật toán nhạy cảm như *Decision Tree.*

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, số

Nội dung do AI tạo ra có thể không chính xác.

- Tiến hành cân bằng dữ liệu: Sử dụng phương pháp Undersampling lớp 0 để cân bằng với lớp 1.

Ảnh có chứa văn bản, ảnh chụp màn hình

Nội dung do AI tạo ra có thể không chính xác.

3. Trích xuất đặc trưng

**3.1. Các đặc trưng trích xuất từ URL**

|  |  |  |
| --- | --- | --- |
| STT | Tên đặc trưng | Mô tả chi tiết |
| 1 | len\_url | Độ dài toàn bộ URL. |
| 2 | len\_domain | Độ dài domain chính. |
| 3 | len\_path | Độ dài của đường dẫn (path) sau domain. |
| 4 | num\_dots | Số dấu chấm (.) trong domain, dùng để xác định số lượng subdomain. |
| 5 | num\_slashes | Số lượng dấu gạch chéo (/) trong phần path. |
| 6 | num\_query\_params | Số lượng tham số truy vấn (query parameters) |
| 7 | has\_ip | Có sử dụng địa chỉ IP trong domain không  (1: *Có*, 0: *Không*) |
| 8 | is\_https | Giao thức có phải là https không  (1: *Có*, 0: *Không*) |
| 9 | has\_keyword | URL có chứa từ khóa đáng ngờ không (như: *login, secure, account, etc.*) |
| 10 | num\_special\_chars | Số ký tự đặc biệt trong URL (-, @,%, &, =, ?, \_, ~) |

**3.2. Trích xuất đặc trưng**

- Cài đặt các thư viện cần thiết

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình, màu trắng

Nội dung do AI tạo ra có thể không chính xác.

- Trích xuất đặc trưng từ URL

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

Hàm *extract\_features* được sử dụng để phân tích và trích xuất các đặc trưng kỹ thuật từ một URL:

- *urlparse(url) ->* dùng để phân tích cú pháp URL thành các phần cơ bản như *giao thức, tên miền, đường dẫn, chuỗi truy vấn,…*

*- parsed.scheme ->* là giao thức của URL, ví dụ như *http, https ->* dùng để kiểm tra tính bảo mật (*is\_https)*

*- parsed.netloc ->* là tên miền của URL, ví dụ như [*www.facebook.com*](http://www.facebook.com) *->* dùng để *kiểm tra IP*, *độ dài domain*, *số dấu chấm*,…

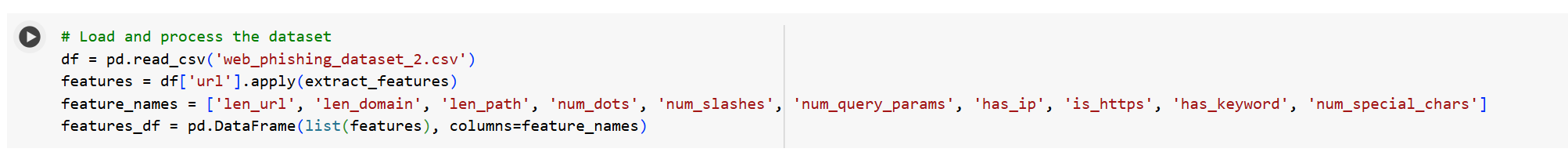
*- parsed.path ->* là phần đường dẫn phía sau domain *->* dùng để *đo độ dài* và *số lượng (/).*

- *parsed.query ->* là chuỗi truy vấn -> dùng để đếm *số tham số truy vấn*.

*Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.*

* Thực hiện trích xuất với một URL mẫu

**

* Thực hiện trích xuất với toàn bộ dataset

Ảnh có chứa văn bản, ảnh chụp màn hình, số, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

* Kết quả sau khi trích xuất

1. Chia tập dữ liệu

Sau khi kiểm tra sự cân bằng, dữ liệu được chia thành *training set* (60%) và test set (40%) bằng hàm train\_test\_split từ thư viện scikit-learn.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Nội dung do AI tạo ra có thể không chính xác.

CHƯƠNG 3: HUẤN LUYỆN MÔ HÌNH

I. Mô hình học máy

**1. Decision Tree (Cây quyết định)**

- Là một mô hình phân loại dựa trên cấu trúc cây, trong đó mỗi nút đại diện cho một điều kiện dựa trên đặc trưng đầu vào, mỗi nhánh thể hiện kết quả của điều kiện đó, và mỗi lá đại diện cho một nhãn lớp. Mô hình thường sử dụng các tiêu chí như GINI Index hoặc Information Gain để lựa chọn đặc trưng và ngưỡng phân chia tối ưu tại mỗi nút.

- Ưu điểm:

+ Dễ hiểu và trực quan, đặc biệt khi cây không quá sâu.

+ Yêu cầu ít tài nguyên tính toán so với các mô hình phức tạp.

+ Có thể xử lý cả đặc trưng số (*numerical*) và phân loại (*categorical*).

- Nhược điểm:

+ Dễ bị quá khớp (*overfitting*), đặc biệt khi cây quá sâu.

+ Nhạy cảm với dữ liệu không cân bằng dẫn đến thiên lệch về lớp chiếm ưu thế.

+ Hiệu suất không quá hiệu quả so với các mô hình ensemble trong các bài toán phức tạp.

**2. Random Forest**

- Là một mô hình ensemble kết hợp nhiều cây quyết định (Decision Tree) để cải thiện độ chính xác và giảm hiện tượng quá khớp. Mỗi cây được huấn luyện trên một tập con ngẫu nhiên của dữ liệu (bootstrap sampling), đồng thời chỉ xét một tập con ngẫu nhiên của các đặc trưng tại mỗi nút phân chia. Dự đoán cuối cùng được xác định bằng cách lấy trung bình (hoặc đa số phiếu) từ các cây.

- Ưu điểm:

+ Hiệu suất cao nhờ kết hợp nhiều cây quyết định, giảm nguy cơ quá khớp.

+ Ổn định và ít nhạy cảm hơn với dữ liệu không cân bằng so với *Decision Tree.*

+ Có thể đánh giá tầm quan trọng của đặc trưng, hữu ích để phân tích các đặc trưng quan trọng.

- Nhược điểm:

+ Yêu cầu nhiều tài nguyên tính toán hơn, đặc biệt với số lượng cây lớn.

+ Có thể bị ảnh hưởng bởi dữ liệu không cân bằng nếu không điều chỉnh trọng số lớp.

**3. XGBoost (eXtreme Gradient Boosting)**

- Là một mô hình gradient boosting mạnh mẽ, xây dựng các Decision Tree tuần tự, trong đó mỗi cây được huấn luyện để sửa lỗi các cây trước đó. Mô hình sử dụng hàm mất mát (loss function) như log loss để tối ưu hóa dự đoán, kết hợp với các kỹ thuật khác để giảm overfitting và tăng tốc độ huấn luyện.

- Ưu điểm:

+ Hiệu quả vượt trội trong các bài toán phân loại nhị phân, kể cả với dữ liệu không cân bằng.

+ Hỗ trợ điều chỉnh trọng số lớp (class weigths) để xử lý mất cân bằng.

+ Cung cấp tính năng đánh giá *feature importance* và khả năng xử lý dữ liệu thưa (*sparse data*).

- Nhược điểm:

+ Yêu cầu tinh chỉnh siêu tham số (*learning rate*, *max\_depth*, *n\_estimators*, v.v.) để đạt hiệu suất tối ưu.

+ Có thể nhạy cảm với dữ liệu không cân bằng nếu không được cấu hình đúng.

**4. CatBoost (Categorical Boosting)**

- Là một mô hình *gradient boosting* được phát triển bởi Yandex, được tối ưu hóa cho các bài toán có dữ liệu phân loại (*categorical data*). Tương tự *XGBoost*, *CatBoost* xây dựng các *Decision Tree* tuần tự, nhưng sử dụng kỹ thuật *ordered boosting* để giảm thiên lệch và *symmetric trees* để tăng tốc độ huấn luyện. Mô hình tự động xử lý các đặc trưng phân loại mà không cần mã hóa (*encoding*).

- Ưu điểm:

+ Hiệu suất cao, đặc biệt với dữ liệu có nhiều đặc trưng phân loại.

+ Tự động xử lý dữ liệu không cân bằng thông qua tham số *class\_weights* hoặc *auto\_class\_weights.*

+ Giảm nguy cơ quá khớp nhờ *ordered boosting* và *regularization*.

- Nhược điểm:

+ Cần tinh chỉnh siêu tham số (*depth*, *learning\_rate*, v.v.) để đạt hiệu quả tối ưu.

+ Yêu cầu thời gian huấn luyện lâu hơn với dữ liệu lớn nếu không sử dụng GPU.

II. Quy trình huấn luyện

- Thư viện sử dụng: Các mô hình được triển khai bằng *scikit-learn* (*Decision Tree*, *Random Forest*), *xgboost* (*XGBoost*) và *catboost* (*CatBoost*).

- Đánh giá hiệu suất:

+ Các chỉ số được sử dụng bao gồm:

1. Accuracy: Tỷ lệ dự đoán đúng trên tổng số mẫu.
2. Precision: Tỷ lệ dự đoán đúng trong số các URL được dự đoán là lừa đảo.
3. Recall: Tỷ lệ các URL lừa đảo thực sự được phát hiện.
4. F1-score: Trung bình hài hòa giữa *precision* và *recall*, phù hợp để đánh giá dữ liệu không cân bằng.
5. AUC-ROC: Diện tích dưới đường cong *Receiver Operating Characteristic*, đo lường khả năng phân biệt giữa hai lớp.

III. Kết quả

**1. Mô hình Decision Tree**

- Kết quả của mô hình Decision Tree:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Nội dung do AI tạo ra có thể không chính xác.

*Mô hình Decision Tree* đạt độ chính xác *91.67%* trên tập dữ liệu gồm *88.757 mẫu*, với thời gian huấn luyện chỉ *0.48s*, cho thấy hiệu suất cao và khả năng xử lý nhanh chóng, phù hợp với phát hiện trang Web lừa đảo theo thời gian thực. Các chỉ số hiệu suất chi tiết bao gồm:

+ *Precision (0.91 – 0.92):* Tỷ lệ dự đoán chính xác cao, giảm thiểu số lượng trang web bị phân loại sai là lừa đảo (false positive).

+ *Recall (0.91 – 0.92):* Khả năng nhận diện tốt các trang web lừa đảo thực sự, hạn chế bỏ sót các trường hợp nguy hiểm (false negative).

+ *F1-score (0.92):* Sự cân bằng giữa precision và recall khẳng định độ ổn định của mô hình trên cả hai lớp.

Các chỉ số *macro avg* và *weighted avg* đều đạt *0.92*, chứng minh mô hình hoạt động hiệu quả trên tập dữ liệu cân bằng.

**2. Mô hình Random Forest**

- Kết quả của mô hình Random Forest:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Nội dung do AI tạo ra có thể không chính xác.

*Mô hình Random Forest* đạt độ chính xác *92.18%* trên tập dữ liệu gồm *88.757 mẫu*, cao hơn đôi chút so với Decision Tree *(91,67%).* Các chỉ số hiệu suất chi tiết bao gồm:

*+ Precision (0,92):* Mô hình dự đoán chính xác 92% các trường hợp, giảm thiểu sai sót khi phân loại trang web an toàn hoặc lừa đảo.

*+ Recall (0,92):* Khả năng nhận diện đúng 92% các trang web lừa đảo, đảm bảo ít bỏ sót các mẫu nguy hiểm.

*+ F1-score (0,92):* Sự cân bằng giữa precision và recall thể hiện độ ổn định của mô hình trên cả hai lớp.

Các chỉ số *macro avg* và *weighted avg* đều đạt *0,92*, khẳng định hiệu suất đồng đều trên tập dữ liệu cân bằng. Tuy nhiên, thời gian huấn luyện của Random Forest (*13,30 giây*) dài hơn đáng kể so với Cây quyết định (*0,48 giây*), cho thấy chi phí tính toán cao hơn. Kết quả này nhấn mạnh Random Forest tận dụng tốt các đặc trưng trích xuất từ URL (như độ dài URL, ký tự đặc biệt, và thông tin tên miền) nhờ cơ chế ensemble, giúp cải thiện độ chính xác so với Decision Tree đơn lẻ.

1. Mô hình XGBoost

- Kết quả của mô hình XGBoost:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Nội dung do AI tạo ra có thể không chính xác.

*Mô hình XGBoost* đạt độ chính xác *92,02%* trên tập dữ liệu gồm *88.757 mẫu*, gần ngang ngửa với Random Forest (92,18%) và vượt trội hơn Decision Tree (91,67%). Thời gian huấn luyện chỉ *2,33 giây*, nhanh hơn đáng kể so với Random Forest (*13,30 giây*) và chỉ chậm hơn Cây quyết định (*0,48 giây*). Các chỉ số hiệu suất bao gồm:

*+ Precision (0,92-0,93)*: Tỷ lệ dự đoán chính xác cao, đặc biệt ở lớp 1 (trang web lừa đảo), giảm thiểu phân loại sai.

*+ Recall (0,91-0,93):* Khả năng nhận diện tốt các trang web lừa đảo và an toàn, hạn chế bỏ sót.

*+ F1-score (0,92):* Sự cân bằng giữa precision và recall khẳng định hiệu suất ổn định trên cả hai lớp.

Các chỉ số *macro avg* và *weighted avg* đều đạt *0,92*, thể hiện khả năng tổng quát hóa tốt trên tập dữ liệu cân bằng. XGBoost tận dụng hiệu quả các đặc trưng trích xuất từ URL (như độ dài URL, ký tự đặc biệt, thông tin tên miền) nhờ cơ chế tăng cường gradient, mang lại độ chính xác cao với chi phí tính toán thấp hơn Random Forest.

**4. Mô hình Cat Boost**

- Kết quả của mô hình Cat Boost:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Nội dung do AI tạo ra có thể không chính xác.

*Mô hình Cat Boost* đạt độ chính xác *92,16%* trên tập dữ liệu gồm *88.757 mẫu*, tương đương với Random Forest (92,18%) và XGBoost (92,02%), đồng thời vượt trội hơn Cây quyết định (91,67%). Tuy nhiên, thời gian huấn luyện của CatBoost là *36,82 giây*, chậm nhất trong các mô hình được đánh giá (Cây quyết định: 0,48 giây; XGBoost: 2,33 giây; Random Forest: 13,30 giây). Các chỉ số hiệu suất bao gồm:

*+ Precision (0,92-0,93):* Tỷ lệ dự đoán chính xác cao, đặc biệt ở lớp 1 (trang web lừa đảo), giảm thiểu phân loại sai.

*+ Recall (0,92-0,93):* Khả năng nhận diện tốt các trang web lừa đảo và an toàn, hạn chế bỏ sót.

*+ F1-score (0,92):* Sự cân bằng giữa precision và recall khẳng định hiệu suất ổn định trên cả hai lớp.

Các chỉ *số macro avg* và *weighted avg* đều đạt *0,92*, thể hiện khả năng tổng quát hóa tốt trên tập dữ liệu cân bằng. CatBoost tận dụng hiệu quả các đặc trưng trích xuất từ URL (như độ dài URL, ký tự đặc biệt, thông tin tên miền) nhờ khả năng xử lý dữ liệu phân loại và tối ưu hóa gradient, mang lại độ chính xác cao. Tuy nhiên, thời gian huấn luyện dài khiến CatBoost kém phù hợp cho các hệ thống yêu cầu phản hồi tức thời.

**5. So sánh kết quả các mô hình**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Bộ phân lớp | Precision | Recall | F1 score | Thời gian huấn luyện | Test Accuracy |
| Decision Tree | 91-92 | 91-92 | 92 | 0.48s | 91.67 |
| Random Forest | 92 | 92 | 92 | 13.30s | 92.18 |
| XGBoost | 92-93 | 91-93 | 92 | 2.33 s | 92.02 |
| Cat Boost | 92-93 | 91-93 | 92 | 36.82s | 92.16 |

*Bảng: Các tham số đánh giá các bộ phân lớp trong mô hình (%)*

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, số

Nội dung do AI tạo ra có thể không chính xác.

*Hình. Độ chính xác của các bộ phân lớp mô hình*

Ảnh có chứa văn bản, ảnh chụp màn hình, Sơ đồ, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

*Hình. Thời gian huấn luyện của các bộ phân lớp mô hình*

*Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, số

Nội dung do AI tạo ra có thể không chính xác.*

*Hình. Ma trận nhầm lẫn của các bộ phân lớp mô hình*

- Decision Tree: Dự đoán đúng 40.557 mẫu hợp pháp (Legit) và 40.807 mẫu lừa đảo (Phish), nhưng có 3.943 mẫu Legit bị phân loại sai thành Phish (false positive) và 3.450 mẫu Phish bị bỏ sót (false negative).

- Random Forest: Hiệu suất tốt hơn với 40.917 mẫu Legit và 40.903 mẫu Phish được phân loại đúng, giảm false positive (3.583) và false negative (3.354), thể hiện khả năng tổng quát hóa tốt hơn Decision Tree.

- XGBoost: Phân loại đúng 41.224 mẫu Legit và 40.452 mẫu Phish, với false positive (3.276) và false negative (3.805), cho thấy khả năng nhận diện mẫu Legit vượt trội nhưng bỏ sót nhiều mẫu Phish hơn so với Random Forest.

- CatBoost: Dự đoán đúng 41.275 mẫu Legit và 40.523 mẫu Phish, với false positive (3.225) thấp nhất, nhưng false negative (3.734) tương đương XGBoost, thể hiện khả năng nhận diện mẫu Legit tốt nhất trong các mô hình.

* Random Forest và CatBoost có hiệu suất cân bằng nhất giữa hai lớp, với số lượng sai sót thấp hơn Decision Tree. XGBoost và CatBoost nổi bật trong việc giảm false positive, rất quan trọng để tránh cảnh báo sai cho các trang web hợp pháp. Tuy nhiên, cả hai mô hình bỏ sót nhiều mẫu Phish hơn (false negative), có thể gây rủi ro trong phát hiện lừa đảo.

Ảnh có chứa văn bản, ảnh chụp màn hình, hàng, Sơ đồ

Nội dung do AI tạo ra có thể không chính xác.

*Hình. Biểu đồ ROC đánh giá hiệu suất của các bộ phân lớp mô hình*

- *CatBoost* đạt AUC cao nhất (*0,9791*), thể hiện khả năng phân biệt tốt nhất giữa trang web hợp pháp và lừa đảo, với tỷ lệ True Positive Rate (TPR) cao và False Positive Rate (FPR) thấp.

- *XGBoost* (AUC = *0,9786*) và *Random Forest* (AUC = *0,9774*) có hiệu suất gần tương đương, nhỉnh hơn *Decision Tree* (AUC *= 0,9570*), cho thấy các mô hình gradient boosting có khả năng tổng quát hóa tốt hơn trên tập dữ liệu.

- *Decision Tree* tuy có AUC thấp nhất nhưng vẫn đạt *0,9570*, chứng minh hiệu suất đáng tin cậy, đặc biệt khi xét đến tốc độ huấn luyện nhanh (0,48 giây).

* XGBoost và CatBoost nổi bật với AUC cao, phù hợp cho các hệ thống cần độ chính xác cao trong phát hiện trang web lừa đảo. XGBoost cũng là lựa chọn tốt khi cần cân bằng giữa hiệu suất (AUC cao) và tốc độ (2,33 giây).

**6. Thử nghiệm hàm Dự đoán (Predict) với một vài đường dẫn**

*Random Forest* và *XGBoost* là hai mô hình được lựa chọn để triển khai hàm dự đoán cũng như để áp dụng cho trang Web kiểm tra nhờ hiệu suất vượt trội và tính phù hợp trong bài toán phát hiện trang web lừa đảo

*+ Random Forest* đạt độ chính xác cao nhất (*92.18%)* và AUC tốt (*0.9774*), thể hiện khả năng phân biệt hiệu quả giữa trang web hợp pháp và lừa đảo. Ma trận nhầm lẫn cũng cho thấy mô hình có số lượng sai sót thấp, đảm bảo độ tin cậy cao. Dù thời gian huấn luyện dài (*13,30s)*, thời gian dự đoán nhanh, phù hợp cho hệ thống kiểm tra URL theo thời gian thực.

*+ XGBoost* có độ chính xác đương tương (*92.02%)* và AUC cao (*0.9786*), với thời gian huấn luyện nhanh hơn đáng kể (*2.33s)*, lý tưởng cho các ứng dụng cần xử lý nhanh. Ma trận nhẫm lẫn cho thấy khả năng giảm false positive tốt, quan trọng để cảnh báo sai cho trang web hợp pháp.

* So với *Decision Tree* (AUC thấp hơn: *0.9570*) và *CatBoost* (thời gian huấn luyện quá lâu: *36.82s*), Random Forest và XGBoost cân bằng giữa độ chính xác, tốc độ, và tính thực tiễn, phù hợp để triển khai trong hệ thống phát hiện phishing thời gian thực.

- Hàm Dự đoán (Predict)

Ảnh có chứa văn bản, Phông chữ, phần mềm, Trang web

Nội dung do AI tạo ra có thể không chính xác.

- Dự đoán

Ảnh có chứa văn bản, Phông chữ, số, hàng

Nội dung do AI tạo ra có thể không chính xác.

Ảnh có chứa văn bản, Phông chữ, hàng, ảnh chụp màn hình

Nội dung do AI tạo ra có thể không chính xác.

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình

Nội dung do AI tạo ra có thể không chính xác.

Ảnh có chứa văn bản, Phông chữ, hàng, ảnh chụp màn hình

Nội dung do AI tạo ra có thể không chính xác.

CHƯƠNG 5: XÂY DỰNG ỨNG DỤNG NHẬN DẠNG PHISHING

Sau khi xây dựng mô hình học máy để phân loại website lừa đảo, mô hình này được triển khai thành một ứng dụng web để cho phép người dùng sử dụng mô hình một cách trực quan, thuận tiện. Ứng dụng được xây dựng dựa trên công nghệ web sử dụng Flask cho phép người dùng nhập URL và kiểm tra xem URL đó là an toàn (Safe) hay là trang web lừa đảo (Phishing). Giao diện ứng dụng đơn giản, dễ sử dụng. Khi người dùng nhập URL của website cần kiểm tra, hệ thống có tích hợp mô hình Machine Learning (*Random Forest* và *XGBoost*) sẽ tiến hành phân tích và trả về kết quả ngay lập tức. Ứng dụng này giúp người dùng dễ dàng nhận biết các trang web đáng ngờ, qua đó nâng cao nhận thức và giảm thiểu rủi ro từ các cuộc tấn công mạng.

Giao diện của ứng dụng web được thiết kế đơn giản với 2 thành phần chính:

* Một trường nhập liệu (textbox) cho phép người dùng nhập vào URL của website cần kiểm tra.
* Một nút “Kiểm tra” để thực hiện phân tích URL.

Link GitHub: [nguyenhung18/web-phishing-detection-v2](https://github.com/nguyenhung18/web-phishing-detection-v2)

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Nội dung do AI tạo ra có thể không chính xác.

**\* Một vài ví dụ khi chạy**

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, Trang web

Nội dung do AI tạo ra có thể không chính xác.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, phần mềm

Nội dung do AI tạo ra có thể không chính xác.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

KẾT LUẬN

Mặc dù các mô hình hiện tại đã đạt được kết quả khả quan, nhưng vẫn còn một số mức độ sai sót nhất định trong việc phát hiện trang web lừa đảo, điều này chỉ ra rằng hệ thống vẫn chưa hoàn toàn chính xác. Một phần nguyên nhân có thể do bộ dữ liệu hiện tại còn hạn chế, chưa bao quát hết tất cả các kiểu trang web lừa đảo thực tế, dẫn đến việc mô hình gặp khó khăn khi xử lý các trường hợp chưa được nhìn thấy trong quá trình huấn luyện.

Bên cạnh đó, các đặc trưng mà đề tài đã trích xuất chỉ là những đặc trưng cơ bản như cấu trúc URL và các thuộc tính của domain. Những đặc trưng này, dù hữu ích, nhưng chưa đi sâu vào các đặc trưng mạnh mẽ hơn như hành vi người dùng, phân tích nội dung HTML, hay các yếu tố liên quan đến tính năng động của trang web. Việc bổ sung thêm các đặc trưng mạnh mẽ và phức tạp hơn sẽ giúp cải thiện đáng kể khả năng phân loại của mô hình.

Để cải thiện hiệu suất và giảm thiểu sai sót, có thể tích hợp thêm nhiều đặc trưng mạnh mẽ như hành vi người dùng, phân tích nội dung HTML, và áp dụng các phương pháp học sâu ***(Deep Learning)*** để nâng cao độ chính xác. Ngoài ra, việc tối ưu hóa các mô hình và cải thiện khả năng xử lý thời gian thực sẽ là yếu tố quan trọng trong việc đưa hệ thống vào ứng dụng rộng rãi.

Mặc dù còn nhiều thách thức, nhưng với sự phát triển không ngừng của công nghệ và việc tiếp tục mở rộng bộ dữ liệu, ứng dụng học máy trong phát hiện phishing vẫn là một hướng đi triển vọng, góp phần bảo vệ an toàn thông tin và nâng cao nhận thức về bảo mật cho người dùng trong môi trường trực tuyến.

TÀI LIỆU THAM KHẢO

[1] PhishTank. (2023). Phishing Database - <https://www.phishtank.com/>

[2] OpenPhish. (2023). OpenPhish Phishing Feeds -<https://openphish.com/>

[3] XGBoost Documentation. (2023) - <https://xgboost.readthedocs.io/en/stable/>

[4] [shreyagopal/Phishing-Website-Detection-by-Machine-Learning-Techniques](https://github.com/shreyagopal/Phishing-Website-Detection-by-Machine-Learning-Techniques)

[5] *CatBoost*. (2025). *CatBoost Documentation* - [CatBoost - open-source gradient boosting library](https://catboost.ai/)

[6] Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems (2nd Edition). O'Reilly Media.