HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO THỰC TẬP CƠ SỞ
Đề tài: Phát hiện URL website độc hại dựa trên mô hình học máy

Sinh viên thực hiện: Phạm Văn Tiến

Mã sinh viên: B21DCCN708

Giảng viên hướng dẫn: Phạm Văn Cường

Hà Nội 2024

MỤC LỤC

I. Giới thiệu

Trong thời đại công nghệ số phát triển mạnh mẽ, internet đã trở thành một phần không thể thiếu trong cuộc sống của con người. Tuy nhiên, bên cạnh những tiện ích mà internet mang lại thì cũng có không ít những mối đe dọa về an toàn thông tin. Một trong những mối đe dọa phổ biến nhất là các trang web độc hại, có thể gây ra các thiệt hại như chứa các mã độc có thể tự động tải xuống, giả mạo đánh cắp thông tin cá nhân, lừa đảo tài chính,...

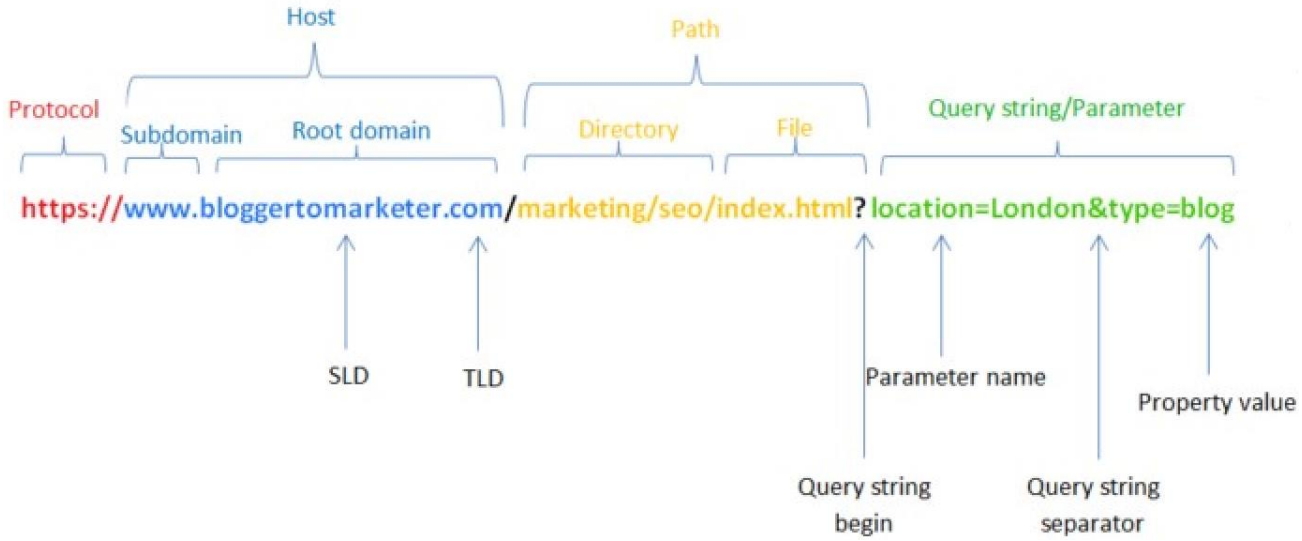
Vì vậy, việc nghiên cứu để phát hiện sớm có các kỹ thuật ngăn chặn URL độc hại trước khi người dùng truy vấn là biện pháp hiệu quả trong giải quyết vấn đề xâm nhập và phát tán các phần mềm độc hại thông qua môi trường web. Nhiệm vụ này được coi là một trong những hướng đi thu hút sự quan tâm của rất nhiều nhà nghiên cứu trong lĩnh vực bảo mật và an toàn thông tin.

Hiện nay hướng tìm hiểu, phân tích dựa vào các mô hình học máy để phát hiện các URL website độc hại là một phương pháp rất hiệu quả trong ngành công nghệ thông tin nói chung và an toàn thông tin nói riêng. Trong đề tài này, em sẽ sử dụng một số thuật toán phân loại được sử dụng trong học máy để đánh giá URL. Sau đó sử dụng mô hình đạt hiệu quả tốt nhất thiết kế một tiện ích mở rộng (extension) tích hợp vào trình duyệt web để giúp người dùng phát hiện sớm các trang web độc hại.

II. Đôi nét về URL website

Bộ định vị tài nguyên thống nhất (Uniform Resource Locator – URL) được sử dụng để tham chiếu tới tài nguyên trên Internet, mang lại khả năng siêu liên kết cho các trang web. Các tài nguyên khác nhau được tham chiếu bằng các địa chỉ mạng (hay gọi cách khác là đường dẫn liên kết mạng) khác nhau.

Cấu trúc cơ bản của một liên kết URL gồm 2 phần chính là: giao thức (protocol) và tên máy chủ (hostname), ngoài ra, trong báo cáo này sẽ sử dụng thêm các thành phần bổ sung khác của 1 URL là đường dẫn (path) và truy vấn (query):



Rất nhiều lập luận cho rằng URL là một địa chỉ web nhưng thực tế nó không hoàn toàn đơn giản như vậy. Một địa chỉ web là URL nhưng tất cả URL không phải chỉ là địa chỉ web. Một số dịch vụ có thể truy cập trên Internet như FTP hoặc thậm chí MAILTO cũng được sử dụng dưới dạng URL.

- Giao thức (Protocol)

  Phần Protocol của URL biểu thị giao thức mà ứng dụng và máy chủ giao tiếp. Các địa chỉ web là các URL phổ biến nhất với 2 giao thức phổ biến:

  - Giao thức truyền tải siêu văn bản (HTTP): Đây là giao thức cơ bản của web, xác định hành động của các máy chủ web và trình duyệt cần thực hiện để đáp ứng các lệnh nhất định.
  - Giao thức HTTP an toàn (HTTPS): Đây là một dạng HTTP hoạt động trên một lớp bảo mật, được mã hóa để truyền tải thông tin an toàn hơn.

  Trình duyệt web cũng có thể xử lý các giao thức khác, bao gồm FTP và mailto. FTP cho phép chia sẻ file giữa các máy chủ web khác nhau, cục bộ hoặc từ xa. Sau đó hướng người dùng đến một địa chỉ email cụ thể.

- Tên máy chủ (Hostname)

  Phần Hostname của URL có nhiệm vụ ánh xạ tới một địa chỉ IP của máy chủ trên Internet. Host name bao gồm các thành phần như tên miền phụ (Subdomain), tên miền cấp 2 (Second-Level Domain) và tên miền cấp cao nhất (Top-level domain). Ngoài ra, trong phần Host name có thể chứa thông tin người dùng (ví dụ

//username:password@www.example.com) hoặc chứa thông tin về cổng kết nối dịch vụ (ví dụ //www.example.com:8080).

- Đường dẫn (Path)

  Nếu như phần Host name của URL đưa trình duyệt (hoặc các ứng dụng khác) đến đúng máy chủ trên mạng, thì phần Path sẽ giúp truy vấn đến đúng thư mục hoặc tệp tin trên máy chủ đó.

  Mặc dù có thể không thấy đường dẫn này trên thanh địa chỉ, nhưng điều đó không có nghĩa là nó không có. Một vài ngôn ngữ được sử dụng để tạo trang web ẩn tên và phần mở rộng của file để người dùng dễ nhớ để gõ URL hơn.

- Truy vấn (Query)

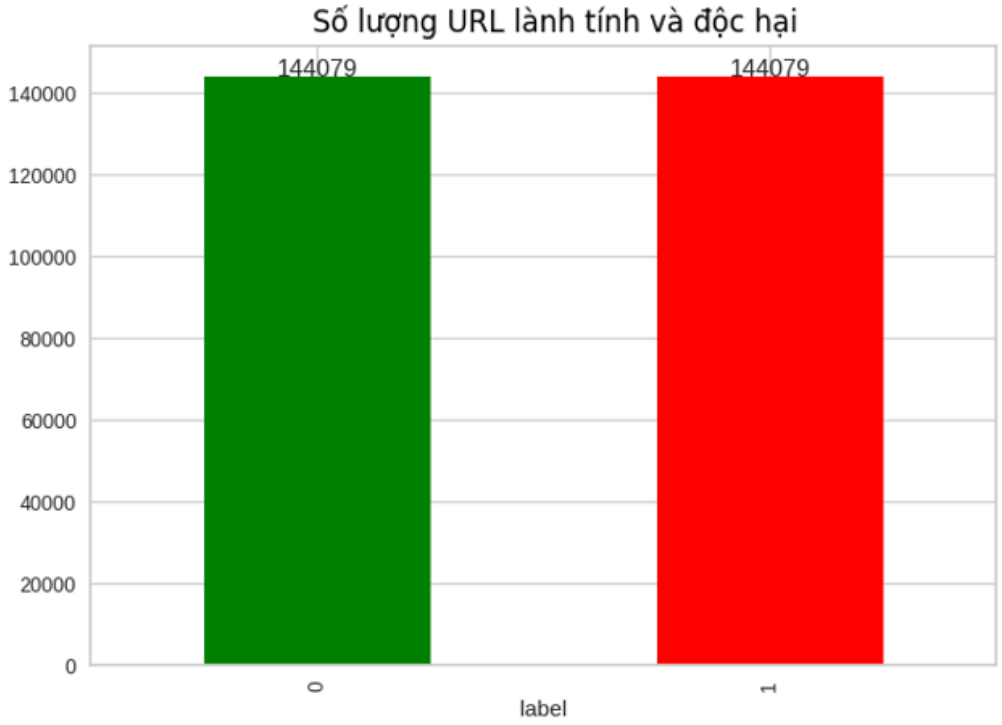
  Truyền dữ liệu thông qua các tham số cho ứng dụng web xử lý. Các tham số đó là danh sách các cặp khóa/giá trị được phân tách bằng ký hiệu &. Máy chủ web có thể sử dụng các tham số đó để thực hiện các công việc bổ sung trước khi trả lại tài nguyên.

Có thể thấy mỗi URL đều có cấu trúc và định dạng riêng. Những kẻ tấn công thường cố gắng thay đổi một hoặc nhiều thành phần cấu trúc của URL để đánh lừa người dùng phát tán URL độc hại của chúng. URL độc hại được gọi là các liên kết gây ảnh hưởng xấu đến người dùng. Các URL này sẽ chuyển hướng người dùng đến các tài nguyên hoặc trang mà trên đó kẻ tấn công có thể thực thi mã trên máy tính của người dùng, chuyển hướng người dùng đến các trang web không mong muốn, trang web độc hại, trang web lừa đảo hoặc tải xuống phần mềm độc hại. Các URL độc hại cũng có thể bị ẩn trong các liên kết tải xuống được coi là an toàn và có thể lây lan nhanh chóng thông qua việc chia sẻ tệp và tin nhắn trong mạng chia sẻ. Một số kỹ thuật tấn công sử dụng URL độc hại bao gồm: Tải xuống ổ đĩa (Drive-by Download), Lừa đảo (Phishing) và kỹ thuật xã hội (Social Engineering) và Thư rác (Spam).

III. Chuẩn bị dữ liệu

Tập dữ liệu (dataset) thu thập từ các kho dữ liệu lớn hay được các hãng bảo mật nổi tiếng trên thế giới cung cấp như Kaggle, UCI Machine Learning Repository, OpenPhish, PhishTank được tổng hợp lại và dán nhãn 1 cho các URL độc hại và 0 cho URL an toàn.

Sau khi xử lý để loại bỏ các giá trị trùng lặp và giảm sự mất cân bằng giữa 2 lớp, ta thu được một danh sách 288158 URL gồm 144079 URL an toàn và 144079 URL độc hại.



Trích chọn đặc trưng (Feature Extraction)

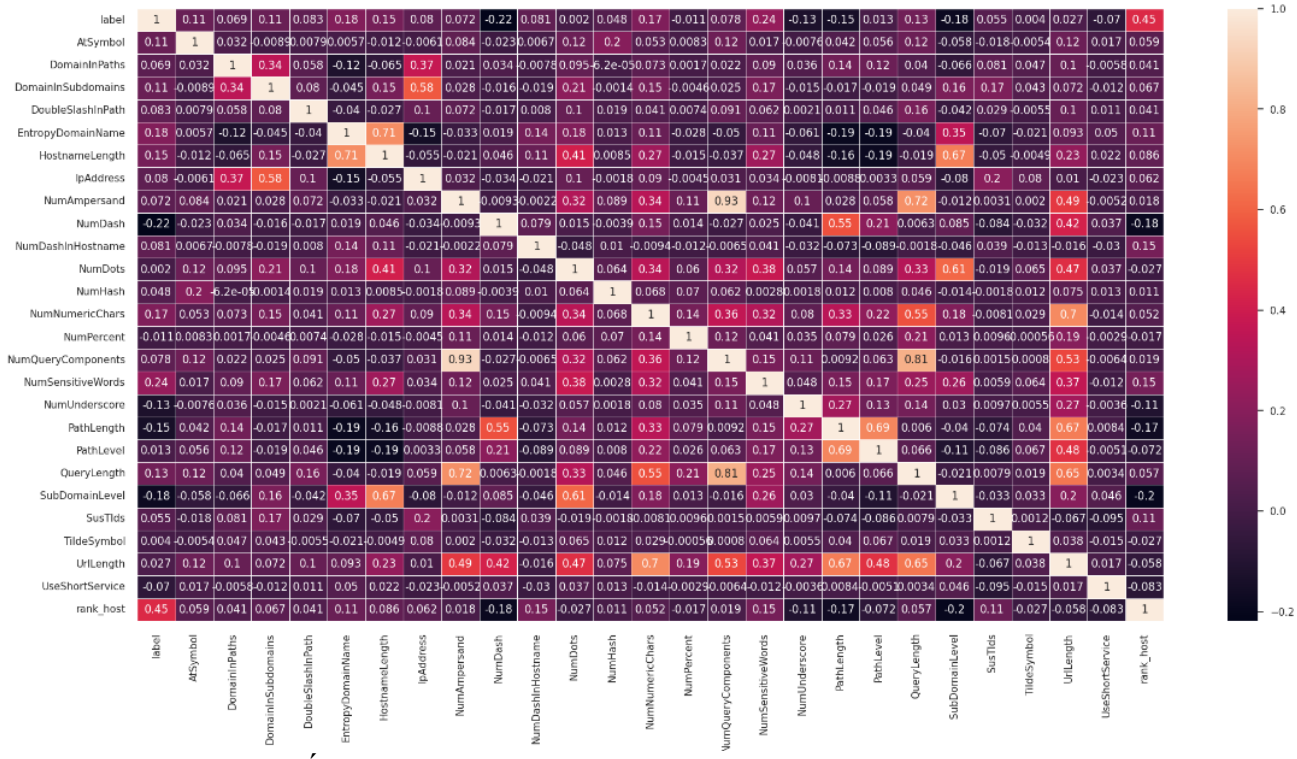
Những tính chất sau được lựa chọn là đặc trưng để phân định URL độc hại hay an toàn, trong báo cáo này mới chỉ sử dụng các đặc trưng từ vựng của URL:

STT	Đặc trưng	Mô tả	Giá trị	Thông tin thêm
1	IpAddress	Kiểm tra địa chỉ IP có được sử dụng trong phần hostname của URL không	0 hoặc 1	Nếu địa chỉ IP được sử dụng thay thế cho tên miền trong URL, người dùng có thể chắc chắn rằng ai đó đang cố lấy cắp thông tin cá nhân quan trọng của họ. Đôi khi, địa chỉ IP còn được chuyển đổi thành dạng mã thập lục phân.
2	UseShortService	Kiểm tra URL có sử dụng dịch vụ rút gọn không.	0 hoặc 1	Rút gọn URL là một phương pháp trên "World Wide Web", làm cho độ dài URL nhỏ hơn đáng kể mà vẫn dẫn đến trang web được yêu cầu. Điều này được thực hiện bằng cách "Chuyển hướng HTTP" trên một tên miền ngắn, liên

				kết đến trang web có URL dài. Với các công cụ phổ biến hỗ trợ rút ngắn độ dài URL, kẻ tấn công có thể che giấu những đặc trưng dễ nhận biết trên URL đối với người dùng và có thể đó là độc hại.
3	SusTlds	Kiểm tra tên miền cấp cao nhất (TLD) của URL có nằm trong danh sách các TLD đáng ngờ không.	0 hoặc 1	Danh sách các tên miền cấp cao nhất đáng ngờ như: 'tk', 'pw', 'biz', 'xyz', 'top', 'club', 'work', 'online', 'download', 'trade', 'cn', 'gq', 'icu', 'fun', 'buzz', 'kim', 'ga', 'cf', 'ml', 'ru',…
4	AtSymbol	Kiểm tra URL có chứa ký tự '@' hay không.	0 hoặc 1	Việc sử dụng ký hiệu '@' trong URL sẽ khiến trình duyệt bỏ qua mọi thứ trước ký hiệu này và địa chỉ thực thường theo sau ký hiệu '@'. Các URL độc hại thường chứa ký tự '@' để mạo danh một URL an toàn.
5	TildeSymbol	Kiểm tra URL có chứa ký tự '~' hay không	0 hoặc 1	
6	DoubleSlashInPath	Kiểm tra URL có chứa '//' không nằm ngay sau giao thức (cụ thể tại phần đường dẫn) hay không.	0 hoặc 1	Sự tồn tại của "//" trong liên kết URL có nghĩa là người dùng sẽ được chuyển hướng đến một trang web khác. Trong 1 URL chuẩn, "//" xuất hiện ngay sau giao thức của URL. Vì vậy, nếu URL bắt đầu bằng "http" thì "//" sẽ xuất hiện ở vị trí thứ 6 (tính từ 0) và bắt đầu bằng "https" thì sẽ xuất hiện ở vị trí thứ 7. Nếu "//" xuất hiện ở 1 vị trí khác thì có thể nghi ngờ rằng URL này là độc hại vì nó đang chuyển hướng đến 1 trang web khác.
7	DomainInSubdomains	Kiểm tra TLD hay ccTLD (tên miền cấp	0 hoặc 1	

		cao nhất theo mã quốc gia) có xuất hiện trong subdomain của URL hay không		
8	DomainInPaths	Kiểm tra TLD hay ccTLD (tên miền cấp cao nhất theo mã quốc gia) có xuất hiện trong đường dẫn URL hay không	0 hoặc 1	
9	rank_host	Kiểm tra tên miền đã đăng kí của URL có nằm trong top 1 triệu tên miền của Alexa không.	0 hoặc 1	
10	UrlLength	Độ dài URL	Số nguyên	
11	HostnameLength	Độ dài phần hostname của URL	Số nguyên	
12	PathLength	Độ dài phần path của URL	Số nguyên	
13	QueryLength	Độ dài phần query của URL	Số nguyên	
14	NumSensitiveWords	Số lượng từ nhạy cảm trong URL.	Số nguyên	Danh sách các từ nhạy cảm như: 'number', 'spoof', 'bank', 'paypal', 'credit', 'confirm', 'free', 'webscr', 'payment', 'secure', 'password', 'bonus', 'identity', 'lucky', 'social', 'money', 'account', 'transfer', 'ebayisapi', 'card', 'ssn', 'service',…
15	NumNumericChars	Số lượng ký tự chữ số trong URL	Số nguyên	
16	NumDots	Số lượng ký tự '.' trong URL	Số nguyên	
17	NumDash	Số lượng ký tự '-' trong URL	Số nguyên	
18	NumDashInHostname	Số lượng ký tự '-' trong phần hostname của URL	Số nguyên	
19	NumUnderscore	Số lượng ký tự '_' trong URL	Số nguyên	
20	NumPercent	Số lượng ký tự '%'	Số	

		trong URL	nguyên	
21	NumAmpersand	Số lượng ký tự '&' trong URL	Số nguyên	
22	NumHash	Số lượng ký tự '#' trong URL	Số nguyên	
23	NumQueryComponents	Số lượng thành phần truy vấn trong URL	Số nguyên	
24	SubDomainLevel	Số lượng cấp độ của subdomain trong 1 URL	Số nguyên	
25	PathLevel	Số lượng cấp độ của đường dẫn trong 1 URL	Số nguyên	
26	EntropyDomainName	Giá trị entropy của tên máy chủ theo công thức entropy của Shannon: $H(x) = -\sum_{i=0}^{n} p(x_i)\log_b p(x_i)$ H(x) là chỉ số Shannon Entropy của chuỗi ký tự URL, p(xi) là hàm số tính khối lượng xác suất của mỗi ký tự trong chuỗi, b = 2.	Số thực	Entropy được sử dụng để phát hiện tên miền ngẫu nhiên. Một số URL độc hại sử dụng thuật toán tạo miền (DGA) để thay đổi miền thường xuyên, do đó việc đưa các URL này vào danh sách đen là không hiệu quả. DGA là một chương trình cung cấp phần mềm độc hại với các tên miền mới theo yêu cầu nhanh chóng. Các URL có entropy của tên miền cao là những chỉ báo quan trọng về hành vi độc hại.



Mối tương quan giữa các đặc trưng trong bộ dữ liệu này

IV. Xây dựng mô hình học máy

Trong quá trình áp dụng mô hình học máy, báo cáo này áp dụng tỷ lệ 8 : 2 nhằm phân chia tập dữ liệu huấn luyện và tập kiểm tra. Trong đó, X_train và X_test là tập hợp mẫu dữ liệu lần lượt có trong tập huấn luyện và tập kiểm tra. Kế tiếp, y_train và y_test là các nhãn tương ứng cho các mẫu dữ liệu trong 2 tập trên, trong đó y_train sẽ được sử dụng để huấn luyện mô hình và sau đó y_test sẽ được dùng để so sánh kết quả phân loại, từ đó đánh giá hiệu suất phân loại của mô hình.

```
X = data.drop(["label"], axis=1)
Y = data["label"]

X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size = 0.2, random_state = 101)
X_train.shape, y_train.shape, X_test.shape, y_test.shape
```

1. Logistic Regression

Logistic Regression là một thuật toán thường được sử dụng trong phân loại nhị phân. Dù tên gọi có chứa từ "Regression", nhưng thực tế nó được sử dụng cho các bài toán phân loại. Logistic Regression là một thuật toán phân loại trong Machine Learning dựa trên nguyên tắc của Hồi quy Tuyến tính (Linear Regression). Tuy nhiên, thay vì dự đoán giá trị liên tục như Hồi quy Tuyến tính, Logistic Regression dự đoán xác suất của một sự kiện nhất

định. Điều này làm cho nó rất hữu ích cho các bài toán phân loại nhị phân, nơi mục tiêu là dự đoán một trong hai lớp có thể có.

Logistic Regression hoạt động bằng cách sử dụng hàm logit (hay hàm sigmoid) để biến đổi đầu ra của một mô hình hồi quy tuyến tính thành một giá trị xác suất từ 0 đến 1.

Cụ thể, mô hình hồi quy tuyến tính là hàm tuyến tính các đặc trưng đầu vào, với công thức:

$$h(x) = \beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n = \sum_{i=0}^{n} \beta_i x_i = \beta^T x$$

Để giới hạn đầu ra trong khoảng [0, 1], hồi quy logistic sử dụng mô hình sau:

$$h(x) = g(\beta^T x) = \frac{1}{1 + e^{-\beta^T x}}$$

Với $g(z) = \frac{1}{1+e^{-z}}$ là hàm logistic hay hàm sigmod.

Giá trị đầu ra dự đoán được xác định bằng cách so sánh $h(x)$ với 1 ngưỡng, chẳng hạn 0.5. Nếu > 0.5 thì tương đương với xác suất y = 1 lớn hơn 50% thì nhãn đầu ra nhận giá trị 1, ngược lại nhận giá trị 0. Trong 1 số trường hợp ta có thể tăng hoặc giảm ngưỡng này.

Mục tiêu của việc huấn luyện mô hình Logistic Regression là tìm các tham số mô hình sao cho hàm mất mát (Loss function) trên tập dữ liệu huấn luyện là nhỏ nhất. Hàm mất mát là một hàm số được sử dụng để đo lường mức độ lỗi mà mô hình tạo ra khi dự đoán các kết quả từ dữ liệu đầu vào. Trong hồi quy Logistic, chúng ta sử dụng hàm mất mát Cross-Entropy (còn gọi là Log Loss) để đánh giá hiệu năng của mô hình.

Hàm Cross-Entropy đo lường khoảng cách giữa hai phân phối xác suất của giá trị thực tế ứng với đầu vào thứ i (yi) và xác suất dự đoán thuộc lớp 1 của mô hình cho đầu vào thứ i (pi). Khi mô hình dự đoán chính xác, tức là nếu yi = 1 thì pi càng gần 1 và nếu yi = 0 thì pi càng gần 0, sau đó hàm mất mát sẽ tiến gần về 0.

Trong quá trình huấn luyện, chúng ta tìm cách cập nhật bộ tham số sao cho giá trị hàm mất mát Cross-Entropy đạt giá trị nhỏ nhất, dẫn đến một mô hình dự đoán tốt nhất. Để tìm giá trị tối ưu cho bộ tham số, chúng ta có thể sử dụng kỹ thuật Gradient Descent.

2. Decision Tree

Thuật toán Decision Tree, hay Cây quyết định, là một phương pháp học máy phổ biến được sử dụng trong cả bài toán phân loại và hồi quy. Cấu trúc của Decision Tree gồm có

nút gốc, nút quyết định, và nút lá. Mỗi nút quyết định trong cây tương ứng với một thuộc tính hay đặc trưng của dữ liệu, và mỗi nhánh đi xuống từ nút đó tương ứng với một giá trị hoặc một khoảng giá trị của thuộc tính đó. Nút lá của cây chứa nhãn hoặc giá trị dự đoán cho dữ liệu.

Quá trình học của Decision Tree bao gồm việc chọn thuộc tính để chia dữ liệu tại mỗi nút. Các thuật toán như ID3, C4.5, và CART được sử dụng để quyết định thuộc tính nào sẽ được chọn dựa trên các tiêu chí như Entropy, Information Gain (độ tăng thông tin) hay Gini Index. Mỗi tiêu chí này đều cung cấp một cách để đo lường chất lượng của một phép chia, giúp thuật toán quyết định cách chia dữ liệu tại mỗi nút.

3. Random Forest

Random Forest là một trong những kỹ thuật học máy tập hợp (ensemble learning) phổ biến nhất, thuộc nhóm phương pháp đóng gói (Bagging method).

Kỹ thuật tập hợp liên quan đến việc tập hợp nhiều mô hình học máy với nhau để tạo ra một mô hình tổng hợp mạnh mẽ hơn. Trong Bagging, chúng ta tạo ra nhiều mô hình dự đoán độc lập với nhau từ các tập dữ liệu con được lấy mẫu ngẫu nhiên với sự thay thế từ tập dữ liệu huấn luyện ban đầu. Sau đó, chúng ta kết hợp các dự đoán của các mô hình này để đưa ra dự đoán cuối cùng. Trong trường hợp này, mô hình Rừng ngẫu nhiên là sự kết hợp của rất nhiều mô hình học Cây quyết định (Decision Trees) để áp dụng vào các bài toán phân loại hoặc hồi quy.

Random Forest hoạt động bằng cách tạo ra một tập hợp các cây quyết định trong quá trình huấn luyện, sau đó đưa ra dự đoán dựa trên kết quả đầu ra của mỗi cây quyết định. Việc lựa chọn kết quả dự đoán cuối cùng tuân theo nguyên tắc đa số. Do đó, kết quả phân loại được lựa chọn bởi số lượng lớn các cây quyết định sẽ trở thành kết quả đầu ra cuối cùng của mô hình rừng ngẫu nhiên. Mỗi cây quyết định trong "rừng" được tạo ra từ một tập con ngẫu nhiên của dữ liệu huấn luyện, và chỉ sử dụng một tập con ngẫu nhiên của các đặc trưng để tạo ra các điểm phân chia. Điều này giúp giảm thiểu hiện tượng quá khớp (overfitting), một vấn đề thường gặp khi sử dụng cây quyết định đơn lẻ.

4. Gradient Boosting

Gradient Boosting cũng là một trong những kỹ thuật học máy tập hợp nhưng thuộc nhóm phương pháp Boosting. Khác với phương pháp đóng gói (bagging), boosting xây

dựng một lượng lớn các mô hình, mỗi mô hình sau sẽ học cách giảm thiểu lỗi của mô hình trước, tạo thành một chuỗi các mô hình mà mô hình sau sẽ tốt hơn mô hình trước bởi việc chú ý tới và cố gắng khắc phục những mẫu bị phân loại sai hơn. Chúng ta sẽ lấy kết quả của mô hình cuối cùng trong chuỗi mô hình này làm kết quả trả về.

Đối với Gradient Boosting, việc huấn luyện mỗi mô hình mới (thường là một cây quyết định) dựa trên các residuals (sự chênh lệch giữa giá trị dự đoán và giá trị thực tế), từng mô hình mới sẽ cố gắng giảm thiểu sai số của mô hình trước đó. Quá trình này sử dụng thuật toán Gradient Descent để tối ưu hóa mô hình, điều chỉnh dự đoán theo hướng giảm thiểu sai số. Quá trình bắt đầu với một mô hình đơn giản, sau đó tính toán residuals. Mỗi mô hình mới, được huấn luyện để dự đoán các residuals của mô hình trước đó, sử dụng Gradient Descent để tối ưu hóa và điều chỉnh dự đoán theo hướng giảm thiểu các sai số này. Các mô hình bổ sung này được kết hợp với mô hình hiện tại bằng cách sử dụng learning rate, một hệ số kiểm soát mức độ ảnh hưởng của mỗi mô hình mới. Quá trình này lặp lại nhiều lần, với mỗi mô hình mới dần dần cải thiện các lỗi của mô hình trước, tạo ra một mô hình tổng thể mạnh mẽ và chính xác.

5. XGBoost

XGBoost (eXtreme Gradient Boosting) là phiên bản cải tiến của Gradient Boosting. Ưu điểm của nó được chứng minh trên nhiều khía cạnh:

- XGBoost thực hiện tinh toán song song nên tốc độ xử lý có thể tăng gấp 10 lần so với GBM. Ngoài ra, XGboost còn hỗ trợ tính toán trên Hadoop.
- XGBoost áp dụng cơ chế Regularization nên hạn chế đáng kể hiện tượng Overfitting

  Regularization, một cách cơ bản, là thay đổi mô hình một chút để tránh overfitting trong khi vẫn giữ được tính tổng quát của nó. Một cách cụ thể hơn, ta sẽ tìm cách di chuyển nghiệm của bài toán tối ưu hàm mất mát tới một điểm gần nó. Hướng di chuyển sẽ là hướng làm cho mô hình ít phức tạp hơn mặc dù giá trị của hàm mất mát có tăng lên một chút.

  Kỹ thuật Regularization phổ biến nhất và được dùng trong XGBoost là thêm vào hàm mất mát 1 số hạng nữa.

$$J_{\text{reg}}(\theta) = J(\theta) + \lambda R(\theta)$$

Số hạng regularization ($R(\theta)$) này sẽ tác động đến hàm loss. Cụ thể: nếu lamda lớn thì ảnh hưởng của đại lượng thêm vào lên hàm loss cũng lớn và ngược lại. Nhưng lamda cũng không được quá lớn vì nếu quá lớn thì đại lượng thêm vào sẽ lấn át loss => mô hình xây dựng sẽ bị sai (underfitting).

XGBoost sử dụng L1 regularization (norm 1 regularization) hoặc L2 regularization.

- XGboost cho phép người dùng sử dụng hàm tối ưu và chỉ tiêu đánh giá của riêng họ, không hạn chế ở những hàm cung cấp sẵn.
- XGBoost bao gồm cơ chế tự động xử lý missing value bên trong nó. Vì thế, có thể bỏ qua bước này khi chuẩn bị dữ liệu cho XGBoost.
- Tự động cắt tỉa cây (auto pruning): tự động bỏ qua những leaves, nodes không mang giá trị tích cực trong quá trình mở rộng tree.

V. Kết quả và đánh giá

1. Các chỉ số đánh giá

Để đánh giá các mô hình được đề xuất, sử dụng ma trận nhầm lẫn và các chỉ số đánh giá phổ biến như Accuracy, Precision, Recall và F1 score:

	Nhãn dự đoán		
Nhãn thật	Âm	Dương	Tổng số
Âm	TN: âm đúng	FP: dương sai	p
Dương	FN: âm sai	TP: dương đúng	n
Tổng số	p'	n'	N

Trong bài toán này thì nhãn URL độc hại là 1 và URL an toàn là 0:

- TP (True Positive): Số lần URL độc hại được dự đoán đúng.
- TN (True Negative): Số lần URL an toàn được dự đoán đúng.
- FP (False Positive): Số lần URL an toàn bị dự đoán sai thành URL độc hại.
- FN (False Negative): Số lần URL độc hại bị dự đoán sai thành URL an toàn.

Dựa trên các thông số trên, những chỉ số đánh giá sau có thể được tính:

- Độ chính xác (Accuracy): Tỷ lệ số lần dự đoán chính xác trong tất cả các mẫu thử nghiệm: Accuracy = (TP + TN) / (TP + TN + FP + FN) = (TP + TN) / N
- Precision: Tỷ lệ số lần dự đoán đúng URL độc hại với tất cả URL được dự đoán là độc hại bởi mô hình: Precision = TP / (TP + FP)
- Recall: Tỷ lệ số lần dự đoán đúng URL độc hại với tất cả URL độc hại trong mẫu thử nghiệm: Recall = TP / (TP + FN)
- F1 score: Tổng hợp 2 chỉ số Precision và Recall, một mô hình có chỉ số F-score cao chỉ khi cả 2 chỉ số Precision và Recall đều cao. Sử dụng F1-score cũng là một thước đo đáng tin cậy về hiệu năng của mô hình trong các bài toán phân loại, đặc biệt khi dữ liệu về một lớp lớn hơn gấp nhiều lần so với dữ liệu về lớp còn lại: F1 score = (2 * Precision * Recall) / (Precision + Recall)

2. Kết quả

Trong báo cáo này sử dụng các lớp model được xây dựng sẵn trong thư viện của python là sklearn (đối với các 4 thuật toán Logistic Regression, Decision Tree, Random Forest và Gradient Boosting) và xgboost (đối với XGBoost). Bảng sau thống kê kết quả huấn luyện và thử nghiệm trên tập dữ liệu có tập train gồm 230526 URL và tập test gồm 57632 URL; các tham số của các mô hình đều đã được điều chỉnh siêu tham số bằng GridSearch (tìm kiếm lưới):

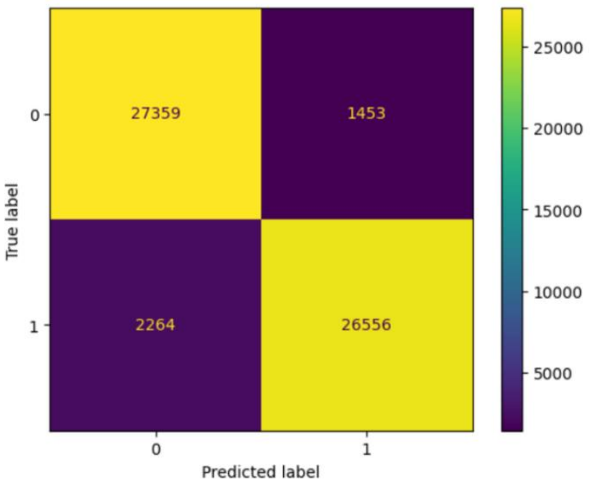
Mô hình	Độ chính xác trên tập huấn luyện	Độ chính xác trên tập kiểm tra	F1 Score	Recall Score	Precision Score
XGBoost	98.8%	95.8%	95.8%	94.8%	96.8%
Gradient Boosting	99.4%	94.6%	94.6%	93.6%	95.6%
Random Forest	98.5%	93.6%	93.5%	92.1%	94.8%
Decision Tree	96.6%	91.7%	91.6%	90.2%	92.9%
Logistic Regression	86.9%	86.8%	86.5%	84.5%	88.5%

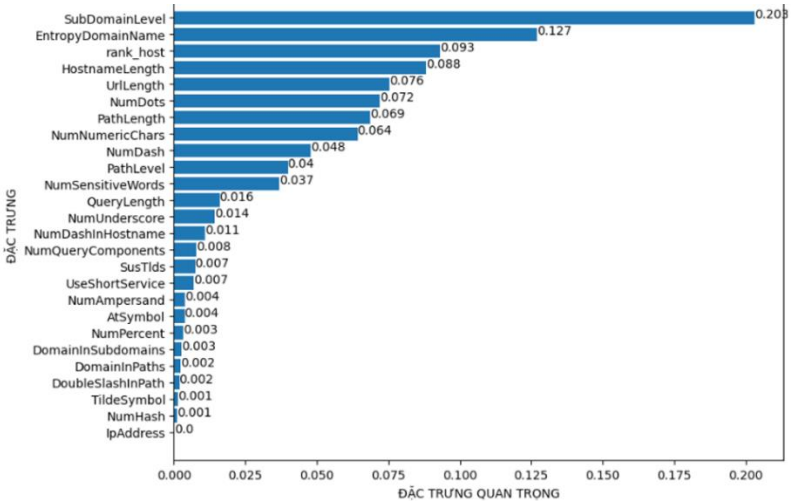
Với các bộ siêu tham số cho mỗi mô hình ứng với kết quả trên như sau, các bộ tham số này có thể chưa phải kết quả tốt nhất:

- Logistic Regression: max_iter = 8000 (số lượng tối đa các lần lặp lại thuật toán tối ưu hóa sẽ thực hiện trước khi dừng), C = 25.
- Decision Tree: criterion = ‘entropy’ (đo lường chất lượng cùa 1 phân chia bằng entropy), min_samples_leaf = 2 (số lượng mẫu tối thiểu 1 nút lá phải có), min_samples_split = 10 (số lượng mẫu tối thiểu 1 nút phải có trước khi nó có thể được phân chia).
- Random Forest: criterion = ‘entropy’, max_features = ‘sqrt’ (số lượng đặc trưng tối đa được chọn ngẫu nhiên để tạo 1 cây trong rừng), min_samples_split = 5, n_estimators = 200 (số cây trong rừng)
- Gradient Boosting: max_depth = 20 (độ sâu tối đa của 1 cây), learning_rate = 0.2 (tốc độ học của mô hình), subsample = 1 (tỷ lệ mẫu con được sử dụng cho mỗi cây), n_estimators = 200.
- XGBoost: max_depth = 20, learning_rate = 0.2, min_child_weight = 5 (tổng trọng lượng (hoặc số lượng) tối thiểu các mẫu mà một nút con cần có), subsample = 1, n_estimators = 200.

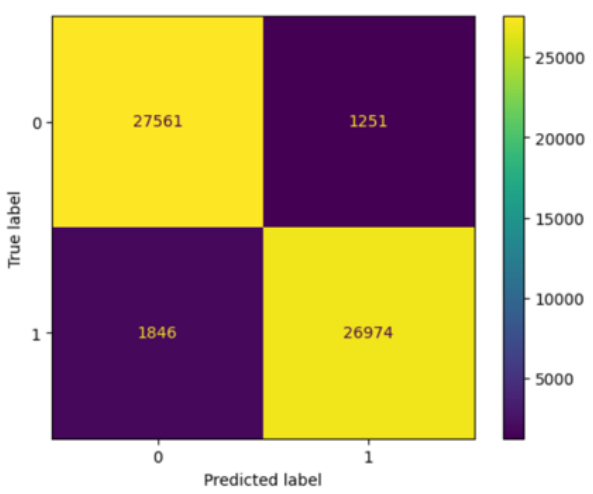
Ma trận nhầm lẫn và danh sách các đặc trưng quan trọng của 3 mô hình cho độ chính xác trên tập kiểm tra tốt nhất:

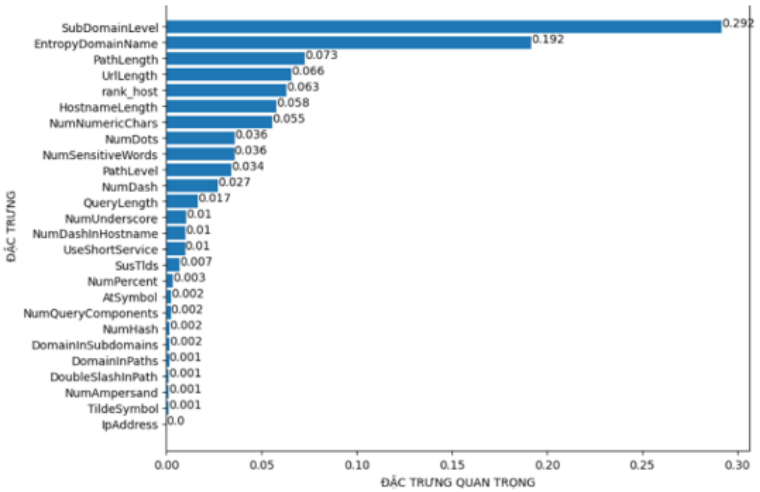
- Random Forest:



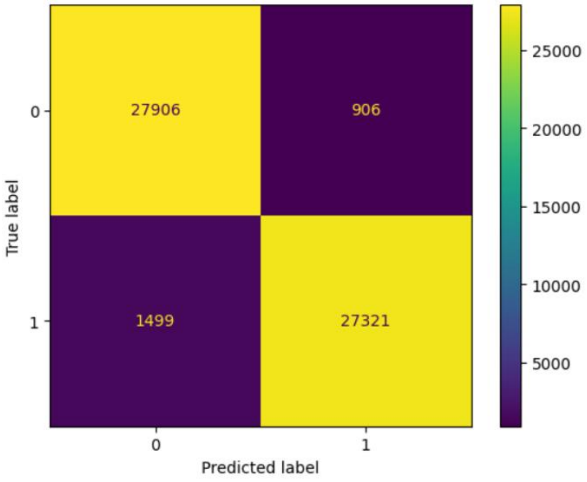


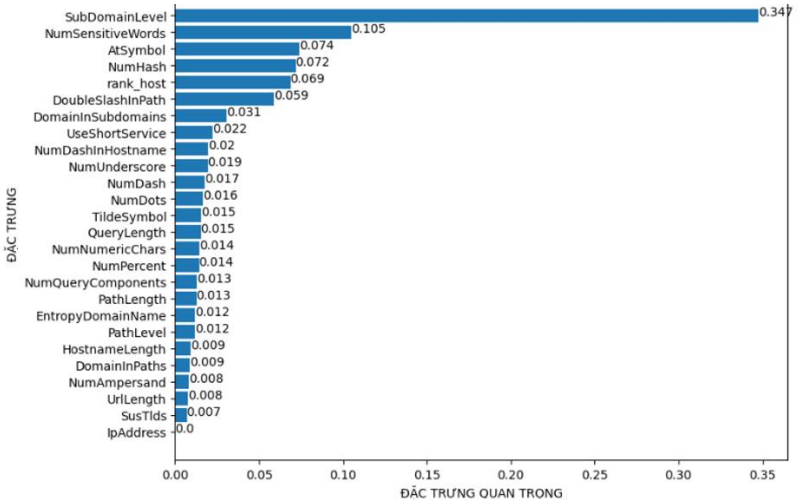
- Gradient Boosting:





- XGBoost:





Mô hình XGBoost cho độ chính xác trên tập kiểm tra cao nhất so với 2 mô hình còn lại và ở cả 3 mô hình thì đặc trưng quan trọng nhất là SubDomainLevel.

VI. Kết luận

Trong báo cáo này, em đã trình bày một phương pháp phát hiện URL độc hại đó là sử dụng machine learning, thử nghiệm với năm mô hình bao gồm Logistic Regression, Decision Trees, Random Forest, Gradient Boosting, và XGBoost . Kết quả được rút ra từ độ chính xác và ma trận nhầm lẫn của ba mô hình là Gradient Boosting, Random Forest và XGBoost cho thấy tính hiệu quả của các mô hình học máy này và các đặc trưng được trích xuất từ URL. Kết quả của báo cáo này được sử dụng để xây dựng một tiện ích đơn giản trên trình duyệt nhằm phát hiện các URL độc hại.

Tuy nhiên, em cũng nhận ra rằng việc chỉ sử dụng các đặc trưng từ vựng từ URL không đủ để đối mặt với tất cả các tình huống trong thực tế hiện nay. Có rất nhiều các đặc trưng khác có thể được rút ra từ URL hoặc dữ liệu trang web, bao gồm như các đặc trưng được trích rút từ code HTML và JavaScript của trang web, hay tuổi tên miền, số lượng truy cập và nhiều hơn nữa. Những đặc trưng này có thể cung cấp thêm nhiều thông tin quan trọng giúp cải thiện tính đúng đắn của mô hình học máy. Vì việc trích xuất các đặc trưng đó mất rất nhiều thời gian, nên trong báo cáo này em vẫn chưa thực hiện được, đó là một thiếu xót lớn.

Trong tương lai, em dự định sẽ mở rộng mô hình của mình hơn bằng cách:

- Sử dụng các bộ dữ liệu lớn hơn và mới hơn.
- Nghiên cứu và chọn ra thêm những đặc trưng khác từ URL để tăng cường độ chính xác.
- Tối ưu hoá thêm các tham số cho mô hình.
- Thử nghiệm và triển khai các mô hình học máy tiên tiến khác.

Em tin rằng việc này sẽ giúp em tạo ra 1 mô hình học máy mạnh mẽ hơn và có khả năng đối mặt với nhiều tình huống phức tạp hơn trong thực tế.

VII. Tài liệu tham khảo

[1] Xây dựng cơ sở dữ liệu huấn luyện phục vụ phát hiện URL độc hại (mic.gov.vn)
[2] PhiUSIIL Phishing URL (Website) - UCI Machine Learning Repository
[3] Malicious And Benign URLs (kaggle.com)
[4] OpenPhish - Phishing Intelligence
[5] PhishStorm - phishing / legitimate URL dataset — Aalto University's research portal
[6] shreyagopal/Phishing-Website-Detection-by-Machine-Learning-Techniques (github.com)
[7] [PDF] An assessment of features related to phishing websites using an automated technique | Semantic Scholar
[8] Detection of malicious URLs using machine learning | Wireless Networks (springer.com)
[9] Bài 6: Logistic Regression (Hồi quy Logistic) - Trí tuệ nhân tạo (trituenhantao.io)

[10] Gradient Boosting - Tất tần tật về thuật toán mạnh mẽ nhất trong Machine Learning (viblo.asia)
[11] (PDF) Malicious URL Detection based on Machine Learning (researchgate.net)
[12] Entropy | Free Full-Text | Malicious URL Detection Based on Associative Classification (mdpi.com)
[13] URL là gì? Cấu trúc của URL - QuanTriMang.com
[14] Machine Learning cơ bản (machinelearningcoban.com)