**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

---🙞🕮🙜---

A red and white logo

Description automatically generated

**BÁO CÁO PROJECT II**

*Đề tài: Phát hiện URL Phishing ứng dụng AI/ML*

**Mã lớp:** 750648

**Mã học phần:** IT3931

**Giảng viên hướng dẫn:** TS. Đinh Thị Hà Ly

Sinh viên thực hiện: Hoàng Đức Khải

Mã số sinh viên: 20225341

Ngành: Kỹ thuật máy tính – IT02

Hà Nội, tháng 5 năm 2025

**Lời nói đầu**

Ngày nay, các cuộc **tấn công lừa đảo** (phishing) ngày càng phổ biến và gây ra nhiều thiệt hại nghiêm trọng. Kẻ tấn công thường tạo ra các **trang web giả mạo** có giao diện giống với các trang web uy tín như Facebook, Amazon, hay eBay, nhằm **đánh lừa** người dùng cung cấp **thông tin cá nhân như mật khẩu hoặc thẻ tín dụng.**

Với người dùng thông thường, việc phân biệt giữa trang web thật và giả là rất khó khăn, vì các trang lừa đảo được thiết kế rất tinh vi. Nhiều người không kiểm tra kỹ địa chỉ trang web (URL) nên dễ bị mắc bẫy và vô tình để lộ thông tin quan trọng cho kẻ gian.

Để góp phần giảm thiểu rủi ro từ các cuộc tấn công này, đề tài “**Phát hiện URL lừa đảo sử dụng AI/ML**” được thực hiện nhằm xây dựng một mô hình có khả năng tự động nhận diện các URL nguy hiểm dựa trên các đặc trưng kỹ thuật được trích xuất từ đường dẫn.

Trong báo cáo này, em sẽ trình bày quy trình xây dựng mô hình phát hiện URL lừa đảo, bao gồm các bước chính như: **thu thập dữ liệu**, **trích xuất đặc trưng**, **lựa chọn mô hình học máy**, **huấn luyện – đánh giá mô hình** và **thử nghiệm**. Hy vọng rằng kết quả đạt được sẽ góp phần vào việc nâng cao hiệu quả phát hiện phishing trong thực tế.

Trong quá trình thực hiện đề tài, em xin gửi lời cảm ơn chân thành đến cô **Đinh Thị Hà Ly**, người đã tận tình hướng dẫn, hỗ trợ em cả về chuyên môn lẫn phương pháp nghiên cứu. Sự góp ý quý báu và đồng hành của cô là yếu tố quan trọng giúp em hoàn thành tốt đồ án này.

**Em xin chân thành cảm ơn!**

Mục lục

[**1.** **Giới thiệu** 4](#_Toc199349686)

[**1.1.** **Vấn đề thực tế** 4](#_Toc199349687)

[**1.2.** **Mục tiêu đề tài** 5](#_Toc199349688)

[**1.3.** **Phạm vi nghiên cứu** 5](#_Toc199349689)

[**2.** **Cơ sở lý thuyết** 5](#_Toc199349690)

[**2.1.** **Kiến thức nền tảng** 5](#_Toc199349691)

[**2.1.1. Khái niệm URL** 5](#_Toc199349692)

[a) URL là gì? 5](#_Toc199349693)

[b) Các hình thức tấn công thông qua URL 6](#_Toc199349694)

[**2.1.2. Các phương pháp phòng chống** 6](#_Toc199349695)

[a) Blacklist/Whitelist 6](#_Toc199349696)

[b) Tách đặc trưng URL và học máy (Machine Learning) 7](#_Toc199349697)

[**2.1.3. Các thuật toán học máy phổ biến** 7](#_Toc199349698)

[**2.2.** **Công nghệ, công cụ sử dụng** 11](#_Toc199349699)

[**2.2.1.** **Python** 11](#_Toc199349700)

[**2.2.2.** **Github** 11](#_Toc199349701)

[**3.** **Kết quả** 11](#_Toc199349702)

[**3.1.** **Random Forest** 11](#_Toc199349703)

[**3.2.** **XGBoost** 11](#_Toc199349704)

[**4. Thiết kế hệ thống** 11](#_Toc199349705)

[**4.1.** **Phân tích chức năng** 11](#_Toc199349706)

[**4.2.** **Sơ đồ hoạt động** 13](#_Toc199349707)

[**5.** **Cài đặt thực nghiệm** 14](#_Toc199349708)

[**5.1.** **Cấu trúc thư mục** 14](#_Toc199349709)

[**5.2.** **Giao diện hệ thống** 15](#_Toc199349710)

[**6.** **Kết luận và hướng phát triển** 17](#_Toc199349711)

# **Giới thiệu**

## **Vấn đề thực tế**

* Các cuộc tấn công lừa đảo đã trở thành mối quan ngại đáng kể do số lượng các cuộc tấn công này ngày càng tăng. Đây là một trong những cuộc tấn công được sử dụng rộng rãi, hiệu quả và mang tính phá hoại nhất, trong đó kẻ tấn công cố gắng lừa người dùng tiết lộ thông tin cá nhân nhạy cảm, chẳng hạn như mật khẩu và thông tin thẻ tín dụng của họ.
* Đối với một người sử dụng thông thường, việc phân biệt giữa các trang web lừa đảo và các trang web hợp lệ là một điều vô cùng khó khăn bởi vì các trang web lừa đảo thường trông giống với các trang web mà chúng giả mạo. Trong nhiều trường hợp, người dùng không kiểm tra toàn bộ URL của trang web và khi họ truy cập vào một trang web lừa đảo, kẻ tấn công có thể truy cập vào thông tin nhạy cảm và cá nhân.
* Với sự phát triển của lĩnh vực thương mại điện tử, các cuộc tấn công lừa đảo và tội phạm mạng đang phát triển nhanh chóng. Kẻ tấn công sử dụng các trang web, email và phần mềm độc hại để thực hiện các cuộc tấn công lừa đảo. Theo tổ chức Anti-Phishing Working Group (APWG), vào năm 2023, đã có xấp xỉ 5 triệu các cuộc tấn công lừa đảo, so với 4,7 triệu vào năm 2022 [[1].](https://apwg.org/apwg-q4-report-finds-2023-was-record-year-for-phishing/#:~:text=CAMBRIDGE%2C%20Mass.%2C%20April%2016,year%20for%20phishing%20on%20record.)

Cách phổ biến nhất để chống lại các cuộc tấn công lừa đảo là sử dụng phương pháp **blacklist/whitelist**. Phương pháp này thường đem lại kết quả đáng kinh ngạc về mặt tốc độ phát hiện các trang web lừa đảo. Tuy nhiên, bất lợi lớn của phương pháp này là tỉ lệ phát hiện phụ thuộc hoàn toàn vào số lượng các trang web có trong blacklist và whitelist. Hơn nữa, phương pháp này không chống được các **cuộc tấn công Zero-Day**.

Trong bối cảnh đó, các phương pháp truyền thống như blacklist/whitelist dần bộc lộ nhiều hạn chế và không còn đủ mạnh để chống lại các hình thức tấn công ngày càng tinh vi. Điều này đặt ra yêu cầu cấp thiết về các **giải pháp tự động, thông minh và có khả năng học hỏi từ dữ liệu.**

Một trong những hướng tiếp cận hiệu quả hiện nay là ứng dụng **trí tuệ nhân tạo (AI)** và **học máy (Machine Learning)** để xây dựng các mô hình có khả năng phân tích, nhận diện URL lừa đảo dựa trên các đặc trưng kỹ thuật và hành vi truy cập. Những mô hình này có khả năng phát hiện cả những trang web mới chưa từng xuất hiện trong danh sách đen, từ đó cải thiện đáng kể hiệu quả phòng chống tấn công lừa đảo.

## **Mục tiêu đề tài**

Mục tiêu chính của đề tài là xây dựng một mô hình có khả năng phát hiện URL lừa đảo bằng cách sử dụng trí tuệ nhân tạo (AI) và học máy (Machine Learning). Cụ thể, đề tài hướng đến:

* Thu thập và xử lý dữ liệu URL bao gồm cả URL hợp lệ và URL lừa đảo.
* Trích xuất các đặc trưng quan trọng từ URL phục vụ cho việc huấn luyện mô hình.
* Lựa chọn và huấn luyện các mô hình học máy để phát hiện các URL lừa đảo với độ chĩnh xác cao.
* Đánh giá, so sánh hiệu năng giữa các mô hình và chọn ra mô hình tối ưu.
* Xây dựng giao diện thử nghiệm đơn giản để minh họa kết quả của mô hình.
* Triển khai hệ thống trình duyệt thực tế áp dụng mô hình

## **Phạm vi nghiên cứu**

Đề tài tập trung vào các nội dung chính sau:

* Phân tích, tìm hiểu về URL và các hình thức tấn công thông qua URL.
* Tập trung vào việc phát hiện URL lừa đảo dựa trên đặc trưng của chính URL đó (như chiều dài, cấu trúc domain/subdomain, entropy,…) mà không phân tích nội dung trang web hoặc hành vi người dùng.
* Áp dụng các mô hình học máy phổ biến như Random Forest, XGBoost.
* Sử dụng tập dữ liệu đã được gán nhãn để huấn luyện và đánh giá mô hình.
* Triển khai hệ thống trình duyệt áp dụng mô hình.

# **Cơ sở lý thuyết**

## **Kiến thức nền tảng**

### **2.1.1. Khái niệm URL**

### a) URL là gì?

URL (Uniform Resource Locator): gọi tắt là địa chỉ web, là một tham chiếu đến tài nguyên web chỉ định vị trí của nó trên một mạng máy tính và cơ chế để truy xuất nó.

Cấu tạo URL về cơ bản gồm 5 phần:

A close-up of a website

AI-generated content may be incorrect.

Ảnh : URL

* **Scheme** hiển thị cho máy chủ web sử dụng giao thức nào khi truy cập một trang trên web (theo chuẩn RFC 3986, hoặc còn được gọi là **Protocol**).
* **Subdomain** là phần mở rộng của tên miền chính (domain), tạo ra để tổ chức hoặc phân chia các khu vực, chức năng trên 1 website.
* **Second-level domain** là tên của website.
* **Top-level domain** chỉ định loại tên miền đã đăng kí.
* **Subdirectory** cung cấp cho người dùng cái nhìn trực quan về phần cụ thể của trang web mà người dùng đang truy cập.

### b) Các hình thức tấn công thông qua URL

* Nếu như 1 trang web được xem là 1 ngôi nhà, thì URL chính là con đường để đi đến đúng ngôi nhà đó
* Lợi dụng điều này, kẻ tấn công hoàn toàn có thể lợi dụng URL để điều hướng người sử dụng đến các trang web giả mạo.
* URL có thể bị lợi dụng bởi 1 trong các cách phổ biến sau:
  + Sử dụng chứng chỉ HTTPS giả mạo
  + Giả mạo tên miền
  + Mã hóa kí tự
  + Tấn công thư mục con
  + URL rút gọn

### **2.1.2. Các phương pháp phòng chống**

### a) Blacklist/Whitelist

* Là phương pháp phòng chống truyền thống, hoạt động dựa trên việc so sánh URL truy cập với một danh sách các địa chỉ đã được phân loại sẵn. Cụ thể:
  + **Blacklist** (danh sách đen): chứa các URL, domain hoặc địa chỉ IP đã được xác định là lừa đảo, độc hại hoặc chứa mã độc. Khi người dùng truy cập vào một địa chỉ thuộc danh sách này, hệ thống sẽ ngay lập tức cảnh báo hoặc chặn truy cập.
  + **Whitelist** (danh sách trắng): gồm các địa chỉ được xác định là an toàn tuyệt đối và luôn được phép truy cập.
* **Ưu điểm** của phương pháp này là khả năng phát hiện nhanh chóng các URL độc hại đã biết, dễ triển khai và vận hành. Đặc biệt, whitelist giúp đảm bảo mức độ an toàn cao trong môi trường giới hạn như mạng nội bộ.
* Tuy nhiên, phương pháp này tồn tại một số hạn chế. Quan trọng nhất là **không thể phát hiện các cuộc tấn công mới (Zero-day)** – những URL chưa từng xuất hiện trong blacklist. Ngoài ra, độ hiệu quả phụ thuộc lớn vào việc **cập nhật danh sách thường xuyên**, và kẻ tấn công có thể dễ dàng vượt qua bằng cách thay đổi nhẹ nội dung URL (ví dụ thay đổi subdomain, thêm ký tự lạ).

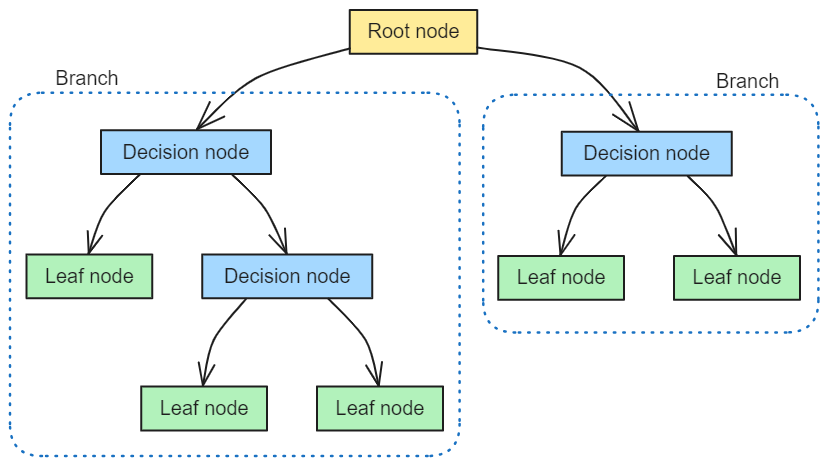
### b) Tách đặc trưng URL và học máy (Machine Learning)

* Phương pháp này sử dụng các thuật toán học máy để tự động phân biệt giữa URL hợp lệ và URL lừa đảo dựa trên các đặc trưng (features) được trích xuất từ chính cấu trúc của URL. Không giống như blacklist/whitelist chỉ xử lý các địa chỉ đã biết, học máy có khả năng **phát hiện các URL chưa từng xuất hiện trước đó** nhờ vào việc học từ dữ liệu.
* Các đặc trưng thường được trích xuất từ URL bao gồm:
  + Độ dài URL, domain, subdomain
  + Sự có mặt của địa chỉ IP thay vì domain
  + Entropy của chuỗi URL
  + Các đặc trưng thống kê khác như số lượng dấu - @ . ? /…
* Sau khi các đặc trưng này được trích xuất, chúng sẽ được đưa vào các thuật toán học máy như Decision Tree, Random Forest, hoặc XGBoost để huấn luyện mô hình phân loại.
* **Ưu điểm** của phương pháp này là khả năng **phát hiện tốt các cuộc tấn công mới (Zero-day)** và **không phụ thuộc vào danh sách có sẵn**. Ngoài ra, mô hình học máy có thể tự động cải thiện độ chính xác khi được huấn luyện với tập dữ liệu đủ lớn và chất lượng.

### **2.1.3. Các thuật toán học máy phổ biến**

**a) Decision Tree**

* Là một trong những thuật toán phổ biến trong AI/ML. Bằng việc chia nhỏ dữ liệu thành các nhóm nhỏ hơn theo 1 cấu trúc dạng cây, giúp mô hình đưa ra quyết định 1 cách trực quan.
* Cấu trúc của một Cây quyết định:



Ảnh : Cây quyết định

* **Gốc (Root Node):** Là điểm bắt đầu của cây, chứa toàn bộ dữ liệu.
* **Nút quyết định (Decision Node):** Là nơi mà dữ liệu được chia thành các nhánh dựa trên điều kiện nhất định.
* **Lá (Leaf Node):** Là các nút cuối cùng, chứa kết quả phân loại hoặc giá trị dự đoán.
* **Nhánh (Branch):** Đại diện cho các quyết định hoặc điều kiện từ một nút cha.
* Cơ sở xây dựng cây quyết định:
  + **Gini Impurity:** Là 1 chỉ số đo mức độ hỗn loạn của 1 tập dữ liệu trong Decision Tree. Nó cho biết xác suất chọn ngẫu nhiên một mẫuvà gán sai nhãn nếu ta chọn theo phân phối hiện tại. **Gini càng thấp, thì nút càng có độ tinh khiết càng cao**, tức là các mẫu nằm trong nhóm đó có xu hướng tập trung về cùng 1 lớp.
  + Thuật toán sẽ lặp lại liên tục đến khi dữ liệu tại nút con đã “thuần” (Cùng nằm trong 1 nhãn), hoặc đạt đến độ sâu tối da cho phép (**max\_depth**), hoặc số lượng mẫu tại nút con quá nhỏ (**min\_samples**).
* Đánh giá thuật toán:
  + Dễ hiểu, dễ trực quan hóa, hoạt động tốt với cả dữ liệu số và dữ liệu phân loại.
  + Có thể xử lý các quan hệ phi tuyến tính.
  + Tuy nhiên, dễ bị tình trạng **“quá khớp” (overfitting)** nếu cây quá sâu (Model chỉ cho kết quả tốt với tập dữ liệu dùng để training nhưng thể hiện kém với tập dữ liệu testing, **không nắm được quy luật tổng quát, mà chỉ “học thuộc lòng”**) và **nhạy cảm với dữ liệu nhiễu.**

**b) Random Forest**

* Là 1 thuật toán học máy dùng để giải quyết các bài toán hồi quy và phân loại, dựa trên ý tưởng học tập hợp **Bagging** – Kết hợp nhiều mô hình yếu (weak learners), cụ thể là các quyết định (decision trees) để tạo ra 1 mô hình mạnh và ổn định hơn.
* Ý tưởng của **Bagging**: Xây dựng 1 lượng lớn các model cùng loại từ những tập subsamples khác nhau từ tập training dataset, còn được gọi là **Boostrapping** (random sample trong 1 tập dataset để tạo dataset mới). Những model này sẽ **được train độc lập và song song với nhau** nhưng đầu ra của chúng sẽ được **trung bình cộng để cho ra kết quả cuối cùng (bài toán hồi quy)** hoặc **lấy phiếu số đông (bài toán phân loại)**.
* Xây dựng thuật toán:
  + Ta có L bộ dữ liệu được tạo ra thông qua boostrapping có kích thước B
  + Tương ứng với L bộ dữ liệu là L model “yếu”

**w1(.), w2(.), …, wL(.)**

* + Kết hợp các model này lại, ta được 1 model mạnh hơn. Kết quả đầu ra cho Bài toán hồi quy là

Bài toán phân loại là

* Đánh giá thuật toán:
  + Ít bị overfitting hơn cây đơn lẻ.
  + Hiệu năng tốt hơn trên nhiều loại dữ liệu khác nhau.
  + Tự động đánh giá tầm quan trọng của đặc trưng.
  + Làm việc tốt với dữ liệu lớn và nhiều chiều.

**c) Gradient Boosting**

* Là 1 kỹ thuật học máy mạnh mẽ dùng để giải quyết các bài toán hồi quy và phân loại, dựa trên ý tưởng học tập hợp **Boosting** – Kết hợp nhiều mô hình yếu (weak learners), thường là các cây quyết định nông (shallow trees), để tạo thành 1 mô hình mạnh.
* Ý tưởng của **Boosting:** Xây dựng 1 lượng lớn các model, tuy nhiên, khác với **Random Forest** sử dụng **Bagging**, mỗi model được xây dựng sau sẽ học và sửa những lỗi của model trước (dữ liệu mà model trước dự đoán sai) -> Tạo thành 1 chuỗi các model mà **model sau sẽ tốt hơn model trước** bởi **trọng số (phần dư)** được cập nhật qua mỗi model (cụ thể ở đây là **trọng số của những dữ liệu dự đoán đúng sẽ không đổi, còn trọng số của những dữ liệu dự đoán sai sẽ được tăng thêm**). Kết quả của model cuối cùng sẽ là kết quả trả về.
* Xây dựng thuật toán: Sử dụng hàm mất mát (loss function) và Gradient Descent để tối ưu.
  + Hàm mất mát:
    - yi  là nhãn thực tế (0 hoặc 1)
    - i là xác suất dự đoán, được tính bằng công thức

­i = log(odds) với odds =

* + Chạy thuật toán:
    - Input: Dữ liệu {(xi, yi)}ni=1 và hàm mất mát khả vi L(yi, F(x)).
    - B1: Khởi tạo giá trị dự đoán cho model
    - Từ m = 1 đến M:
      * Tính toán phần dư rim = yi – Fi-1(x)
      * Huấn luyện 1 cây hồi quy trên các giá trị phần dư rim và tạo các vùng lá (vùng kết thúc) Rjm với j = 1…Jm (Tổng số lá (Vùng kết thúc) mà cây tại vòng lặp m tạo ra)
      * Với j = 1…Jm tính output value tại từng vùng kết thúc
      * Cập nhật lại giá trị với là learning rate (tốc độ học) để kiểm soát mức độ điều chỉnh mô hình sau mỗi lần lặp khi cập nhật cây mới.
    - Kết quả đầu ra là FM(x)
* Đánh giá thuật toán
  + Ưu điểm: Độ chính xác cao nếu được tinh chỉnh tốt, có thể xử lý dữ liệu dạng số và dạng phân loại, linh hoạt với nhiều hàm mất mát, có thể dùng cho bài toán phân loại, đa lớp, hồi quy.
  + Nhược điểm: Dẽ bị overfitting nếu không điều chỉnh hợp lý (max\_depth lơn, learning\_rate cao, …)

**d) XGBoost**

* XGBoost (Extreme Gradient Boosting) là một phương pháp học máy mạnh mẽ được tối ưu hóa cho bài toán cây quyết định tăng cường (gradient boosting decision trees).
* Về bản chất, XGBoost chính là Gradient Boosting, tuy nhiên đã có những cải tiến so với Gradient Boosting chính là có thể chống overfitting sử dụng 2 hyperparameter **L1 (Lasso)** và **L2 (Ridge)**, hỗ trợ GPU, tính toán song song nhằm tăng tốc độ xử lý.
* L1 Regularization (Lasso - Least Absolute Shrinkage and Selection Operator) **thêm một hình phạt** **vào tổng giá trị tuyệt đối của các trọng số** **(weights)/phần dư (residuals)** trong mô hình, từ đó có thể dẫn đến việc một số **trọng số trở thành 0**, làm cho **mô hình trở nên đơn giản hơn** (có thể sử dụng cho chọn lọc tính năng - feature selection).
* L2 Regularization (Ridge Regression) **thêm một hình phạt vào tổng bình phương của các trọng số trong mô hình**, từ đó giúp **giảm thiểu các trọng số lớn mà không làm cho chúng trở thành 0, giúp mô hình ổn định hơn**.

## **Công nghệ, công cụ sử dụng**

### **Python**

Ngôn ngữ lập trình của toàn bộ hệ thống, dễ đọc, hỗ trợ mạnh mẽ cho Machine Learning, phù hợp với đồ án với các thư viện sau:

* **Scikit-learn:** Sử dụng để thao tác trên tập dataset có sẵn.
* **Xgboost:** Thư viện cho mô hình XGBoost.
* **Optuna:** Thư viện tối ưu hyperparameter mạnh mẽ, tự động.
* **Joblib:** Sử dụng để đóng gói model sau khi huấn luyện, để sử dụng lại sau này.
* **Flask:** Framework tạo giao diện web đơn giản.
* **Requests:** Sử dụng trong việc kiểm tra URL rút gọn.
* **Numpy:** Sử dụng đẻ chứa thông tin dữ liệu đầu vào để chuẩn bị đưa vào model tiến hành dự đoán.

### **Github**

Sử dụng để quản lý phiên bản của mã nguồn.

# **Kết quả**

Trong đồ án này, em sử dụng tập [dataset](https://data.mendeley.com/datasets/6tm2d6sz7p/1) để huấn luyện với các thuộc tính sau:

* Tổng cộng 247950 mẫu, với 128541 là URL lừa đảo (51.84%) và 119409 URL an toàn (48.16%) với tổng cộng 41 đặc trưng trích xuất được từ URL.

Ngoài ra, em có sử dụng 1 tập dataset khác để kiểm tra với các thuộc tính sau:

* Tổng cộng 65226 mẫu, với 32613 mẫu là URL lừa đảo (50%) và 32613 mẫu là URL an toàn (50%)

Tiêu chí đánh giá sẽ là độ chính xác và thời gian phản hồi.

## **Random Forest**

Kết quả huấn luyện và test của model là:

A screenshot of a computer program

AI-generated content may be incorrect.

Ảnh 3: Kết quả tập train (trái) và test (phải)

Thời gian phản hồi của model khi chạy thực tế là:

A screen shot of a computer program

AI-generated content may be incorrect.

Ảnh 4: Thời gian phản hồi

## **XGBoost**

Kết quả huấn luyện và test của model là:

A screenshot of a computer program

AI-generated content may be incorrect.

Ảnh 5: Kết quả tập train (trái) và test (phải)

Thời gian phản hồi của model khi chạy thực tế là:

A screen shot of a computer program

AI-generated content may be incorrect.

Ảnh 6: Thời gian phản hồi

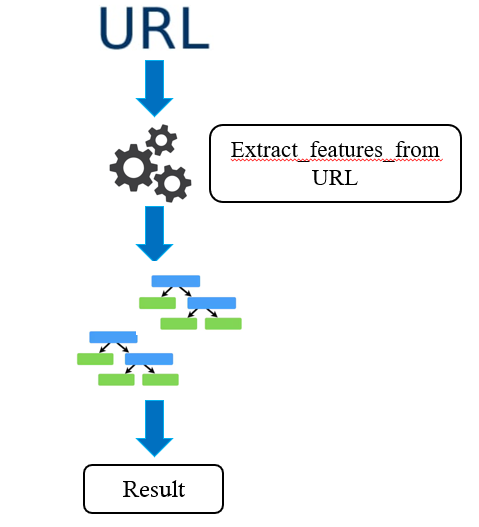
**Kết luận:** Model XGBoost cho kết quả trên tập kiểm tra tốt cũng như thời gian phản hồi thực tế tốt hơn model Random Forest, em đã quyết định chọn mô hình XGBoost cho hệ thống.

# **4. Thiết kế hệ thống**

## **Phân tích chức năng**

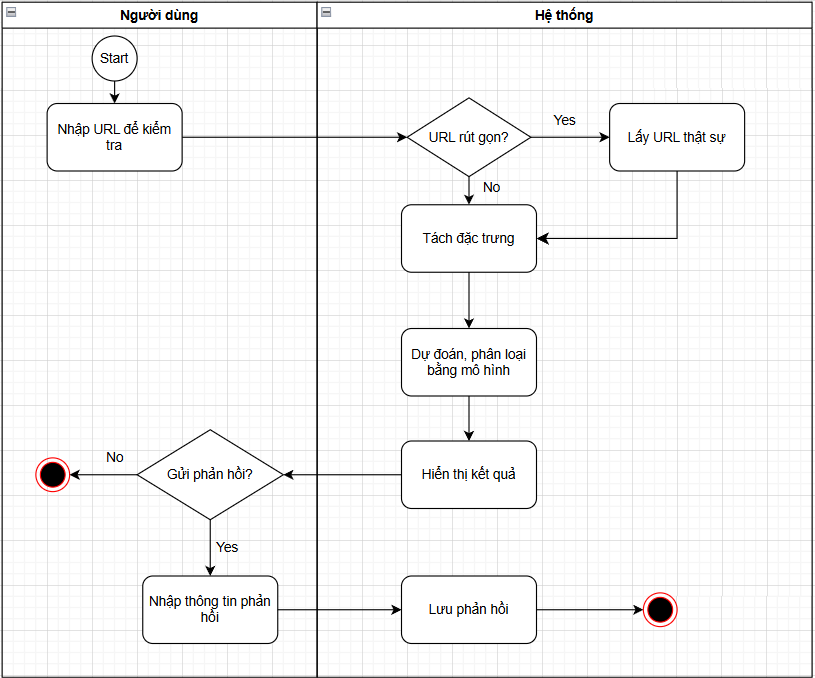
Các chức năng chính của hệ thống bao gồm:

* Phân tích các đặc trưng từ URL.
* Sử dụng mô hình XGBoost để phân loại URL dựa trên các đặc trưng đã phân tích.
* Hiển thị kết quả phân loại, cho phép người dùng gửi phản hồi nếu cần thiết.



Ảnh 7: Minh họa chức năng

## **Sơ đồ hoạt động**



Ảnh 8: Luồng hoạt động của hệ thống

Luồng hoạt động chính của chương trình bao gồm:

* Bước 1: Người dùng nhập URL để kiểm tra.
* Bước 2: Hệ thống kiểm tra URL có phải là URL rút gọn hay không.
* Bước 3: Phân tích đặc trưng của URL trích xuất được.
* Bước 4: Mô hình phân loại URL là hợp lệ/lừa đảo cùng với điểm tin cậy.
* Bước 5: Trả về kết quả cho người dùng.

# **5. Cài đặt thực nghiệm**

## **Cấu trúc thư mục**

Cấu trúc thư mục của hệ thống bao gồm:

A diagram of a computer program

AI-generated content may be incorrect.

Ảnh 9: Cấu trúc thư mục

* **WebDemov2** (Thư mục gốc): Chứa toàn bộ mã nguồn và tài nguyên của ứng dụng web.
* **app.py** (Tệp python): Logic nghiệp vụ chính của web, tương tác với các thành phần khác như Model, Utils, Template để khởi tạo ứng dụng, nhận đầu vào từ người dùng, gọi các hàm xử lý request tương ứng, gọi model tiến hành dự đoán và trả kết quả cho người dùng.
* **Model** (Thư mục): Chứa tệp **Xgb\_model.pkl**, là một mô hình XGBoost đã được huấn luyện và lưu lại. Tệp app.py sẽ tải mô hình này để thực hiện dự đoán mà không cần phải huấn luyện lại mỗi lần chạy.
* **static** (Thư mục): Chứa file **styles.css**, dùng để định nghĩa giao diện, bố cục, màu sắc, font chữ và các yếu tố trực quan khác cho trang web của ứng dụng.
* **templates** (Thư mục): Chứa file **index.html**, dùng để tạo trang HTML cho ứng dụng.
* **Utils** (Thư mục): Chưa file **extract\_features\_from\_url.py**, là file dùng dể trích xuất đặc trưng từ 1 URL đầu vào. Các đặc trưng này sẽ được sử dung làm đầu vào cho mô hình Xgb\_model.pkl để đưa ra dự doán.
* **UserLog** (Thư mục): Chứa file **user\_report.csv**, lưu trữ các phản hồi của người dùng về dự đoán của model xgb\_model.pkl, được sử dụng để cải thiện lại model sau này.

## **5.2. Giao diện hệ thống**

* Giao diện chính

A screenshot of a computer

AI-generated content may be incorrect.

Ảnh 10: Giao diện chính

* Kết quả hiển thị

A screenshot of a computer

AI-generated content may be incorrect.

Ảnh 11: Kết quả hiển thị

* Gửi phản hồi

A screenshot of a computer

AI-generated content may be incorrect.

Ảnh 12: Hiển thị kết quả

A screenshot of a computer screen

AI-generated content may be incorrect.

Ảnh 1: Gửi phản hồi thất bại

A screenshot of a web page

AI-generated content may be incorrect.

Ảnh 1: Gửi phản hồi thành công

# **6. Kết luận và hướng phát triển**

**Nhận xét mô hình phân loại URL:**

* Tốc độ phản hồi nhanh.
* Hiệu suất cao trên 1 tập kiểm thử lớn.

Tuy nhiên, kết quả chạy thực tế vẫn còn khả năng sai lệch, có thể do:

* URL lừa đảo ngày càng phát triển tinh vi hơn, cùng với vấn đề không thể tự động cập nhật mô hình.
* Chưa kết hợp các đặc trưng động của URL vào để phân loại.

**Hướng phát triển tương lai:**

* Thử nghiệm với các mô hình Deep Learning, hoặc kết hợp nhiều mô hình cùng lúc.
* Áp dụng phân tích các đặc trưng động.
* Lập lịch huấn luyện lại mô hình định kì dựa trên phản hồi từ người dùng.