CHƯƠNG 1. URL-BASED VÀ SMS SYSTEMS

Hệ thống dựa trên URL và SMS đã trở thành một phần không thể thiếu trong cuộc sống kỹ thuật số hiện đại, được sử dụng rộng rãi cho nhiều mục đích, từ thương mại điện tử và tiếp thị đến giao tiếp cá nhân và dịch vụ công. Tuy nhiên, sự phổ biến này cũng đi kèm với những rủi ro bảo mật, đặc biệt là sự gia tăng của các hoạt động lừa đảo. Phần này sẽ tập trung vào việc phân tích các hệ thống dựa trên URL và SMS, làm nổi bật các phương pháp phát hiện lừa đảo và các biện pháp bảo mật cần thiết.

1.1. Sự phát triển của các hệ thống dựa trên URL và SMS

Sự phát triển của internet và điện thoại di động đã thúc đẩy sự phổ biến của các hệ thống dựa trên URL và SMS. URL (Uniform Resource Locator) là một trong các tiêu chuẩn để định vị tài nguyên trên internet, trong khi SMS (Short Message Service) cho phép gửi tin nhắn văn bản ngắn giữa các thiết bị di động. Sự kết hợp của hai công nghệ này đã tạo ra một nền tảng mạnh mẽ cho nhiều ứng dụng, bao gồm:

- Thương mại điện tử: URL được sử dụng để dẫn người dùng đến các trang web mua sắm, trong khi SMS được sử dụng để gửi xác nhận đơn hàng, khuyến mãi và cập nhật giao hàng.
- Tiếp thị: URL trong tin nhắn SMS cho phép người dùng truy cập nhanh vào các trang web quảng cáo, chương trình khuyến mãi và thông tin sản phẩm.
- Dịch vụ tài chính: SMS được sử dụng để xác thực giao dịch, gửi thông báo số dư tài khoản và cảnh báo gian lận.
- Giao tiếp cá nhân: SMS vẫn là một phương thức giao tiếp phổ biến, cho phép người dùng gửi tin nhắn văn bản nhanh chóng và dễ dàng.
- Dịch vụ công: SMS được sử dụng để gửi thông báo khẩn cấp, thông tin y tế và các dịch vụ công cộng khác.

1.2. Rủi ro bảo mật và lừa đảo

Mặc dù mang lại nhiều lợi ích, các hệ thống dựa trên URL và SMS cũng tiềm ẩn nhiều rủi ro bảo mật. Sự gia tăng của các hoạt động lừa đảo là một mối quan tâm đáng kể. Các hình thức lừa đảo phổ biến bao gồm:

- Phishing: Tin nhắn SMS hoặc email chứa URL dẫn đến các trang web giả mạo, yêu cầu người dùng cung cấp thông tin cá nhân như tên đăng nhập, mật khẩu và thông tin tài khoản ngân hàng.
- Smishing: Tương tự như phishing, smishing sử dụng SMS để lừa đảo người dùng.
- Malware: URL độc hại có thể dẫn đến việc tải xuống phần mềm độc hại lên thiết bị của người dùng.
- Spam: Tin nhắn SMS không mong muốn, thường chứa quảng cáo hoặc nội dung lừa đảo.

CHƯƠNG 2. ĐỀ XUẤT THIẾT KẾ HỆ THỐNG

Bài này trình bày đề xuất thiết kế một hệ thống phát hiện lừa đảo trong tin nhắn SMS và URL, sử dụng các kỹ thuật học máy và xử lý ngôn ngữ tự nhiên. Hệ thống này nhằm mục đích bảo vệ người dùng khỏi các mối đe dọa trực tuyến bằng cách phân tích nội dung, ngữ cảnh và các đặc điểm khác của tin nhắn và URL để xác định các dấu hiệu lừa đảo.

2.1. Kiến trúc tổng thể

Hệ thống được thiết kế theo kiến trúc modular, bao gồm các thành phần chính sau:

- Mô-đun phân tích và trích xuất đặc trưng: Phân tích nội dung của SMS và các URL được tìm thấy trong tin nhắn SMS để phát hiện các đặc điểm đáng ngờ, chẳng hạn như tên miền đáng ngờ, URL rút gọn và các tham số bất thường. Từ đó xây dựng mô hình phù hợp cho việc phân loại.
- Mô-đun học máy và xử lý ngôn ngữ tự nhiên: Sử dụng các mô hình học máy có giám sát và các kỹ thuật NLP để phân tích nội dung tin nhắn SMS. Nhằm phân loại tin nhắn SMS và URL là hợp pháp hoặc lừa đảo. Mô-đun này được huấn luyện trên một tập dữ liệu lớn các tin nhắn và URL đã được gắn nhãn.
- Mô-đun báo cáo: Cung cấp giao diện để hiển thị kết quả phân tích và báo cáo các tin nhắn và URL đáng ngờ cho người dùng.

2.2. Quy trình thu thập và xử lý dữ liệu

2.2.1. Các phương pháp phân tích URL trong phát hiện lừa đảo

Phân tích URL là một phần quan trọng trong việc phát hiện lừa đảo, vì các URL thường được sử dụng để dẫn người dùng đến các trang web giả mạo hoặc độc hại. Các phương pháp phân tích URL bao gồm:

2.2.1.1. Phân tích cấu trúc URL

Phân tích cấu trúc URL tập trung vào việc kiểm tra các thành phần của URL để phát hiện các dấu hiệu lừa đảo. Các đặc điểm thường được xem xét bao gồm:

- Kiểm tra xem URL có sử dụng địa chỉ IP thay vì tên miền hay không.
- URL quá dài có thể là dấu hiệu của một URL độc hại.
- URL rút gọn có thể che giấu đích đến thực sự của liên kết.
- Kiểm tra xem URL có chứa ký tự @ hay không.
- Kiểm tra xem URL có chuyển hướng người dùng đến trang web khác hay không.
- Kiểm tra xem URL có chứa dấu gạch ngang (-) giữa tên miền hay không.
- Kiểm tra số lượng tên miền phụ trong URL.

2.2.1.2. Phân tích bảo mật URL

Phân tích bảo mật URL liên quan đến việc kiểm tra các yếu tố bảo mật của URL. Các đặc điểm thường được xem xét bao gồm:

- Kiểm tra xem URL có sử dụng HTTPS hay không.
- Kiểm tra xem URL có sử dụng port chuẩn hay không.
- Kiểm tra xem tên miền có khớp với chứng chỉ HTTPS hay không.

2.2.1.3. Phân tích nội dung trang web

Phân tích nội dung trang web liên quan đến việc kiểm tra nội dung của trang web mà URL dẫn đến. Các đặc điểm thường được xem xét bao gồm:

- Kiểm tra xem URL có yêu cầu tài nguyên từ các tên miền khác hay không.
- Kiểm tra các liên kết neo trong trang web.
- Kiểm tra các liên kết trong thẻ script.
- Kiểm tra xem biểu mẫu trên trang web có gửi dữ liệu đến tên miền khác hay không.
- Kiểm tra xem trang web có chứa địa chỉ email thông tin hay không.

2.2.1.4. Phân tích hành vi

Phân tích hành vi tập trung vào việc theo dõi và phân tích hành vi của người dùng khi tương tác với URL. Các đặc điểm thường được xem xét bao gồm:

- Kiểm tra xem trang web có chuyển hướng nhiều lần hay không.
- Kiểm tra xem thanh trạng thái có bị tùy chỉnh hay không.
- Kiểm tra xem trang web có vô hiệu hóa chuột phải hay không.
- Kiểm tra xem trang web có sử dụng cửa sổ popup hay không.
- Kiểm tra xem trang web có sử dụng iframe để chuyển hướng hay không.

2.2.1.5. Phân tích lịch sử và uy tín trang web

Phân tích lịch sử và uy tín trang web liên quan đến việc kiểm tra các yếu tố như tuổi đời và lưu lượng truy cập của trang web. Các đặc điểm thường được xem xét bao gồm:

- Kiểm tra tuổi đời của tên miền.
- Kiểm tra bản ghi DNS của tên miền.
- Kiểm tra lưu lượng truy cập của trang web.
- Kiểm tra xếp hạng trang của trang web.
- Kiểm tra xem trang web có được Google lập chỉ mục hay không.
- Kiểm tra số lượng liên kết trỏ đến trang web.
- Kiểm tra báo cáo thống kê của trang web.

2.2.1.6. Kết hợp các phương pháp phân tích

Việc kết hợp nhiều phương pháp phân tích URL có thể giúp cải thiện độ chính xác và hiệu quả của hệ thống phát hiện lừa đảo. Bằng cách sử dụng cả phân tích cấu trúc, bảo mật, nội dung, hành vi và lịch sử, hệ thống có thể đưa ra các đánh giá toàn diện và chính xác hơn về mức độ an toàn của URL.

2.2.2. Các phương pháp phân tích tin nhắn SMS

Phân tích tin nhắn SMS để phát hiện lừa đảo là một ứng dụng quan trọng của học máy và học sâu trong xử lý ngôn ngữ tự nhiên (NLP). Một số phương pháp phân tích tin nhắn SMS bao gồm:

2.2.3. Tiền xử lý văn bản

- Làm sạch dữ liệu: Loại bỏ các ký tự không cần thiết như dấu câu, ký tự đặc biệt, hoặc các từ dư thừa. Tin nhắn thường được chuyển thành chữ thường để đảm bảo tính đồng nhất.
- Tokenization: Phân tách văn bản thành các từ hoặc cụm từ nhỏ hơn (tokens) để phân tích.
- Loại bỏ từ dừng (Stop Words Removal): Loại bỏ các từ phổ biến không mang nhiều ý nghĩa như "là", "của", "và" trong các ngôn ngữ khác nhau.
- Stemming và Lemmatization: Rút gọn từ về gốc để giảm bớt sự khác biệt giữa các biến thể của từ.

2.2.4. Khai thác đặc trưng (Feature Extraction)

- Bag of Words (BoW): Chuyển đổi văn bản thành ma trận tần suất các từ xuất hiện, không quan tâm đến thứ tự của chúng.
- TF-IDF (Term Frequency-Inverse Document Frequency): Đánh trọng số từ dựa trên tần suất xuất hiện trong một tin nhắn so với toàn bộ tập dữ liệu.
- Word Embeddings: Sử dụng các phương pháp như Word2Vec, GloVe, hoặc FastText để chuyển đổi từ ngữ thành các vector không gian liên tục, biểu diễn ngữ nghĩa tốt hơn.

2.2.5. Kỹ thuật nâng cao

- Transfer Learning: Tận dụng các mô hình ngôn ngữ được huấn luyện trước, như BERT, để giảm thời gian huấn luyện và cải thiện độ chính xác.

• Attention Mechanisms: Áp dụng các cơ chế attention để giúp mô hình tập trung vào các phần quan trọng trong tin nhắn.

2.2.6. Một số thách thức trong việc phân tích

• Dữ liệu không cân bằng: Số lượng tin nhắn lừa đảo thường ít hơn so với tin nhắn thông thường, gây ra sự mất cân bằng dữ liệu. Các kỹ thuật như oversampling, undersampling hoặc sử dụng các thuật toán đặc biệt như XGBoost có thể giải quyết vấn đề này.

• Thay đổi hành vi lừa đảo: Các kẻ tấn công liên tục thay đổi chiến thuật, nên mô hình cần được cập nhật thường xuyên bằng cách huấn luyện lại với dữ liệu mới.

2.3. Xây dựng mô hình học máy

Quy trình huấn luyện mô hình học máy bao gồm các bước sau:

1. Chuẩn bị dữ liệu: Thu thập và chuẩn bị một tập dữ liệu lớn các tin nhắn SMS và URL đã được gắn nhãn là hợp pháp hoặc lừa đảo. Tập dữ liệu này được chia thành tập huấn luyện, tập kiểm tra và tập xác thực.
2. Lựa chọn mô hình: Lựa chọn một hoặc nhiều mô hình học máy phù hợp cho bài toán phân loại, chẳng hạn như máy vectơ hỗ trợ (SVM), cây quyết định, hoặc mạng nơ-ron.
3. Huấn luyện mô hình: Huấn luyện mô hình trên tập dữ liệu huấn luyện, sử dụng các kỹ thuật tối ưu hóa để cải thiện hiệu suất của mô hình.
4. Đánh giá mô hình: Đánh giá hiệu suất của mô hình trên tập dữ liệu kiểm tra và tập xác thực, sử dụng các chỉ số đánh giá như độ chính xác, độ phủ, F1-score.
5. Điều chỉnh siêu tham số: Điều chỉnh các siêu tham số của mô hình để tối ưu hóa hiệu suất.
6. Triển khai mô hình: Triển khai mô hình đã được huấn luyện vào hệ thống để phân loại tin nhắn SMS và URL trong thời gian thực.

Quá trình này được lặp lại và cải tiến liên tục để đảm bảo hiệu suất và độ chính xác của hệ thống. Việc sử dụng các kỹ thuật học máy tiên tiến và dữ liệu huấn luyện chất lượng cao là yếu tố quan trọng để xây dựng một hệ thống phát hiện lừa đảo hiệu quả. Hệ thống cũng được thiết kế để dễ dàng cập nhật và bảo trì, cho phép tích hợp các mô hình và thuật toán mới trong tương lai. Việc theo dõi và đánh giá hiệu suất hệ thống thường xuyên là cần thiết để đảm bảo hệ thống hoạt động hiệu quả và đáp ứng được các yêu cầu bảo mật.

2.4. Đánh giá, lựa chọn và huấn luyện mô hình

2.4.1. Support Vector Machine (SVM)

- Hiệu quả với dữ liệu có số chiều cao, giúp phân biệt tốt giữa các lớp nếu dữ liệu phân tách được. Sử dụng tốt khi có một lượng dữ liệu nhỏ hoặc trung bình, thường đạt độ chính xác cao trong phân loại văn bản. Có thể sử dụng hạt nhân (kernel) để xử lý dữ liệu phi tuyến tính.
- Nhược điểm: Không hiệu quả khi làm việc với các tập dữ liệu lớn vì yêu cầu thời gian tính toán cao. Hiệu suất giảm nếu dữ liệu không phân tách tuyến tính tốt, và việc chọn kernel phù hợp có thể phức tạp.
- Thích hợp khi dùng với các phương pháp khai thác đặc trưng như TF-IDF hoặc BoW.

2.4.2. Naive Bayes

- Ưu điểm: Nhanh, dễ triển khai, và yêu cầu ít tài nguyên tính toán, lý tưởng cho các bài toán phân loại văn bản. Hiệu quả đặc biệt trong bài toán phân loại văn bản hoặc với dữ liệu có nhiều đặc trưng không liên quan. Hoạt động tốt với dữ liệu có sự phân phối xác suất rõ ràng.
- Nhược điểm: Giả định độc lập giữa các đặc trưng (giả định của Naive Bayes) có thể không đúng trong thực tế, làm giảm độ chính xác. Không hiệu quả khi xử lý các đặc trưng có mối quan hệ phức tạp.

- Các biến thể phổ biến như Multinomial Naive Bayes hoặc Bernoulli Naive Bayes được dùng nhiều cho phân loại văn bản.

2.4.3. Random Forest

- Ưu điểm: Khả năng tổng quát hóa tốt và hiệu quả với các tập dữ liệu có nhiều đặc trưng. Chống overfitting do sử dụng nhiều cây quyết định với kết quả trung bình. Có thể xử lý dữ liệu không cân bằng và cung cấp thông tin về tầm quan trọng của từng đặc trưng.
- Nhược điểm: Yêu cầu nhiều tài nguyên tính toán và bộ nhớ khi làm việc với các tập dữ liệu lớn. Khó diễn giải các quyết định của mô hình, vì Random Forest là một "hộp đen".
- Thích hợp để thử nghiệm ban đầu vì có hiệu suất tốt mà không cần nhiều tinh chỉnh.

2.4.4. Logistic Regression

- Ưu điểm: Dễ hiểu, dễ triển khai, và diễn giải, với hiệu quả tốt trong bài toán phân loại nhị phân. Hoạt động tốt khi các đặc trưng có mối quan hệ tuyến tính với đầu ra. Yêu cầu ít tài nguyên tính toán, thích hợp với các bộ dữ liệu cỡ vừa và nhỏ.
- Nhược điểm: Không xử lý tốt dữ liệu phi tuyến tính hoặc có mối quan hệ phức tạp giữa các đặc trưng. Nhạy cảm với dữ liệu không cân bằng, cần sử dụng thêm các kỹ thuật như điều chỉnh trọng số hoặc oversampling.
- Có thể mở rộng để xử lý đa lớp bằng phương pháp One-vs-Rest.

2.4.5. Long Short-Term Memory (LSTM)

- Ưu điểm: Khả năng ghi nhớ thông tin dài hạn, giúp xử lý tốt các chuỗi văn bản dài và có ngữ cảnh phức tạp. Phù hợp để phát hiện các mẫu ngữ nghĩa trong dữ liệu tuần tự. Thích hợp cho dữ liệu mà ngữ cảnh trước đó có ảnh hưởng quan trọng đến quyết định.

• Nhược điểm: Yêu cầu nhiều tài nguyên tính toán và thời gian huấn luyện lâu. Cần một lượng lớn dữ liệu để hoạt động hiệu quả.

• LSTM thường được sử dụng trong các bài toán xử lý ngôn ngữ tự nhiên phức tạp hơn, như phân tích cảm xúc hoặc nhận diện thực thể.

2.4.6. Bidirectional Encoder Representations from Transformers (BERT)

• Ưu điểm: Mô hình mạnh mẽ với khả năng hiểu ngữ cảnh hai chiều, giúp nắm bắt ý nghĩa của từ trong văn bản tốt hơn so với các mô hình trước đó. Hiệu quả cao trong các bài toán xử lý ngôn ngữ tự nhiên phức tạp, có thể tinh chỉnh (fine-tune) cho các nhiệm vụ cụ thể như phân loại spam. Được tiền huấn luyện trên tập dữ liệu lớn, giúp giảm thiểu thời gian và tài nguyên cần thiết để huấn luyện từ đầu.

• Nhược điểm: Yêu cầu rất nhiều tài nguyên phần cứng, như GPU hoặc TPU, để huấn luyện và suy luận. Triển khai phức tạp và cần nhiều điều chỉnh để có kết quả tối ưu.

• Thích hợp cho các bài toán mà ngữ cảnh sâu của từ hoặc cụm từ cần được hiểu rõ.

2.4.7. Recurrent Neural Network (RNN)

• Ưu điểm: Phù hợp để xử lý dữ liệu tuần tự, như văn bản, với khả năng mô hình hóa ngữ cảnh ngắn hạn. Dễ dàng xử lý các chuỗi dữ liệu có độ dài khác nhau.

• Nhược điểm: RNN truyền thống có vấn đề với việc ghi nhớ thông tin dài hạn, do hiện tượng "vấn đề biến mất hoặc bùng nổ của gradient". Hiệu quả không cao so với các mô hình học sâu tiên tiến như LSTM hoặc BERT.

• Thường được thay thế bằng LSTM hoặc GRU (Gated Recurrent Unit) trong các bài toán yêu cầu ngữ cảnh dài hạn.

2.4.8. Đánh giá lựa chọn mô hình

- Mô hình đơn giản: Naive Bayes, Logistic Regression, và SVM là lựa chọn tốt nếu dữ liệu không quá phức tạp và muốn mô hình dễ hiểu.
- Mô hình trung cấp: Random Forest thích hợp khi cần một mô hình mạnh mẽ với khả năng xử lý dữ liệu phức tạp mà không phải mất nhiều thời gian tinh chỉnh.
- Mô hình nâng cao: LSTM và BERT là những lựa chọn tốt khi ngữ cảnh của văn bản đóng vai trò quan trọng, và có đủ tài liệu và tài nguyên tính toán.

2.5. Thiết kế giao diện người dùng

Giao diện người dùng được thiết kế đơn giản và dễ sử dụng, cho phép người dùng dễ dàng tương tác với hệ thống:

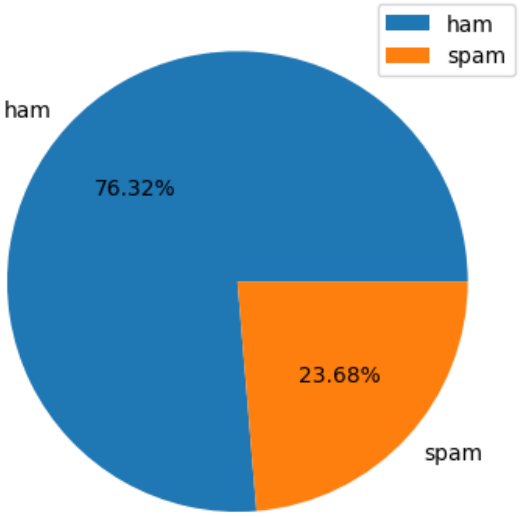
- Hiển thị form nhập văn bản để người dùng nhập tin nhắn SMS hoặc URL cần phân tích.
- Hiển thị kết quả phân tích, bao gồm xác định tin nhắn hoặc URL là hợp pháp hoặc lừa đảo, và các thông tin chi tiết về các đặc điểm đáng ngờ.

CHƯƠNG 3. TRIỂN KHAI HỆ THỐNG PHÂN LOẠI

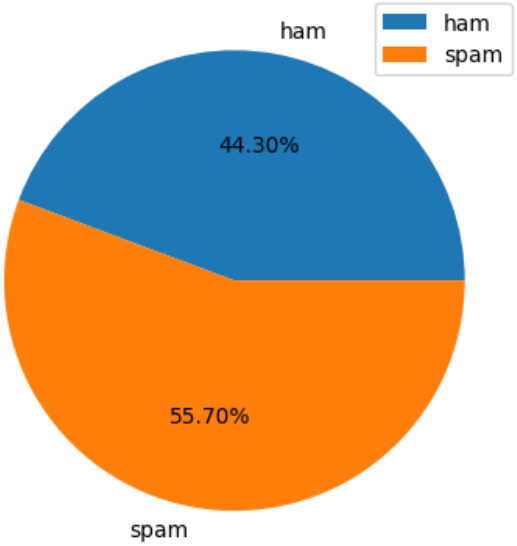
3.1. Thu thập và xử lý dữ liệu huấn luyện

Dữ liệu trong bài viết này tôi sử dụng nguồn từ Kaggle và Corpus. Bao gồm:

- 425 tin nhắn SMS rác đã được trích xuất thủ công từ Grumbletext. [1]
- 3,375 tin nhắn SMS hợp lệ được chọn ngẫu nhiên từ NUS SMS Corpus (NSC), là một tập dữ liệu gồm khoảng 10.000 tin nhắn hợp pháp được thu thập để nghiên cứu tại Khoa Khoa học Máy tính của Đại học Quốc gia Singapore. Phần lớn các tin nhắn này đến từ người Singapore, chủ yếu là sinh viên của trường. [2]
- 450 tin nhắn SMS hợp lệ được thu thập từ Luận án Tiến sĩ của Caroline Tag. [3]
- 1.002 tin nhắn SMS hợp lệ và 322 tin nhắn rác của tập dữ liệu SMS Spam Corpus v.0.1 Big. [4]
- Tập dữ liệu URL của hơn 11.000 trang web. Mỗi mẫu có 30 tham số và một tham số xác định trang web đó có phải là lừa đảo / spam hay không. [5]



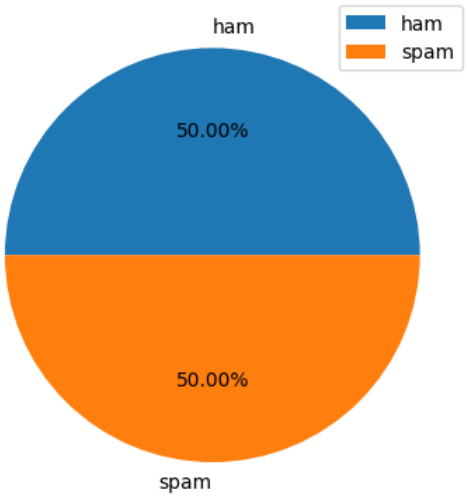
Hình 1: Phân phối tin nhắn SMS hợp pháp (ham) và rác (spam)



Hình 2: Phân phối URL hợp pháp (ham) và lừa đảo / spam (spam)

3.1.1. Tiền xử lý dữ liệu và trích xuất đặc trưng

Do số lượng mẫu trong mỗi lớp của dữ liệu SMS bị chênh lệch khá lớn (23.68% ham / 76.32% spam) nên tôi sử dụng phương pháp undersampling để cân bằng dữ liệu. Phương pháp này thực hiện bằng cách chọn ngẫu nhiên một số mẫu từ lớp chiếm ưu thế sao cho số lượng mẫu của hai lớp trở nên cân bằng hoặc gần cân bằng. Ngoài ra, ta cũng có thể sử dụng kỹ thuật tăng cường dữ liệu văn bản (data augmentation) để tạo ra thêm các mẫu huấn luyện.

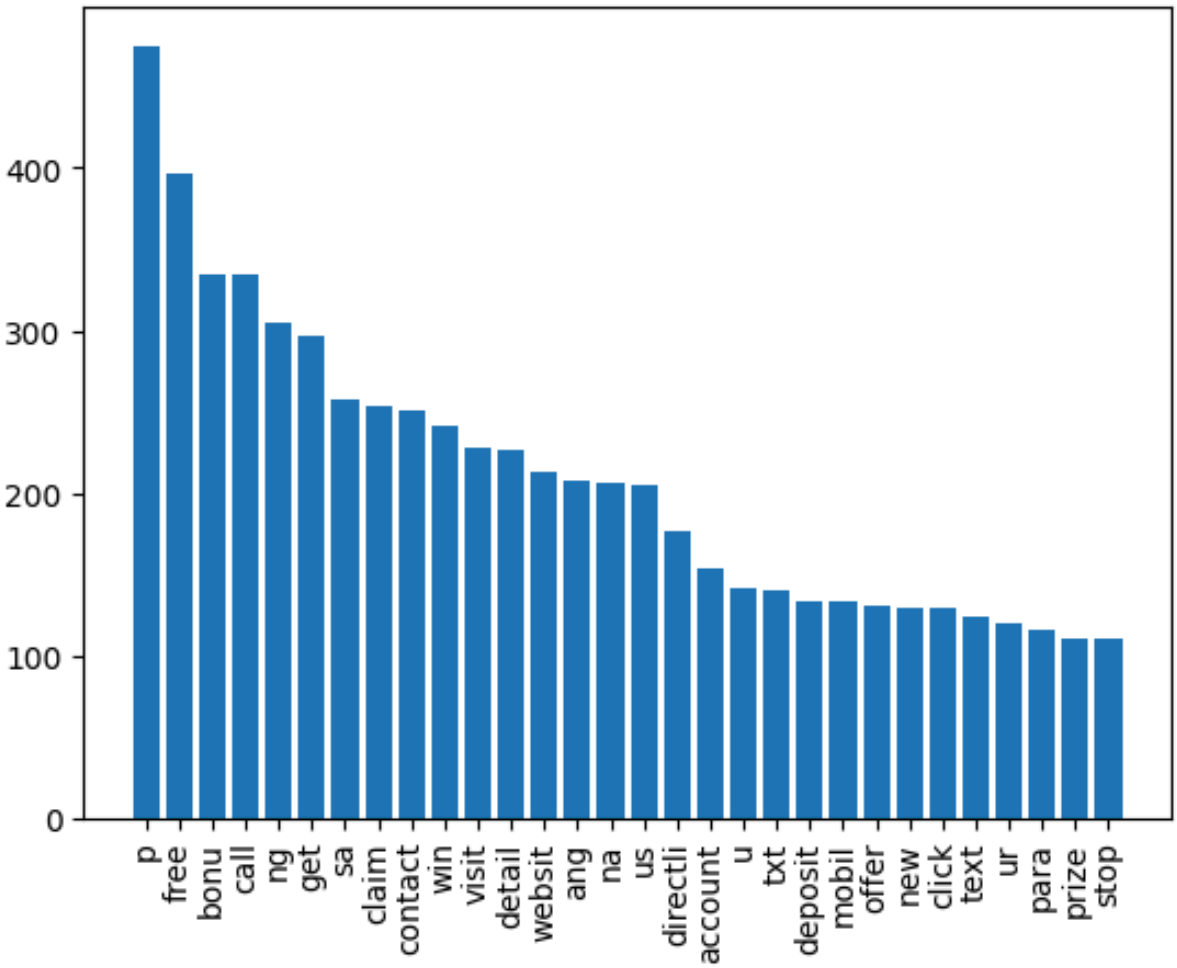


Hình 3: Phân phối tin nhắn SMS hợp pháp (ham) và rác (spam) sau khi cân bằng

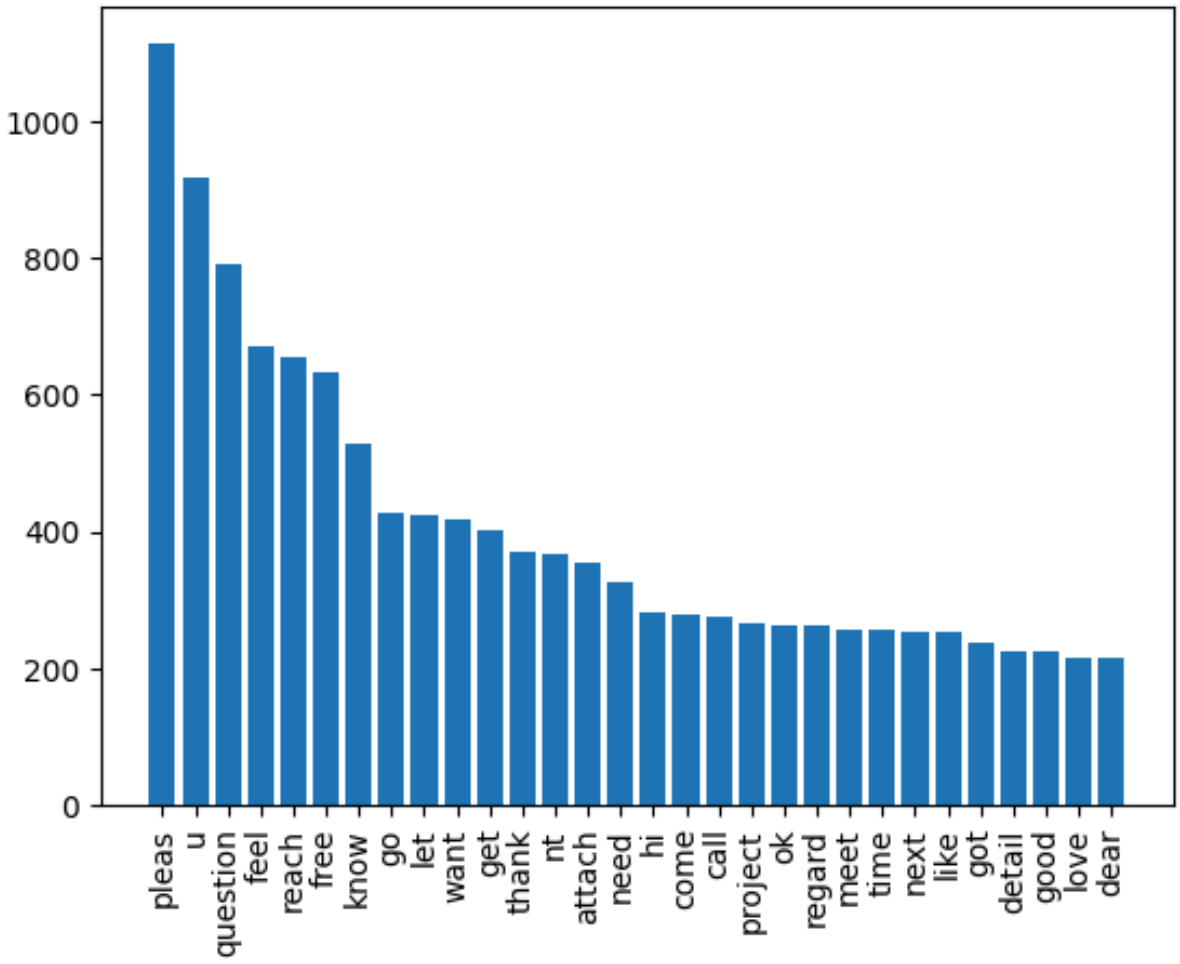
Dữ liệu SMS sau đó được tiền xử lý để loại bỏ các thông tin nhiễu khác như stopwords, các ký tự đặc biệt, dấu câu không cần thiết, chuyển đổi văn bản thành chữ thường,…

	content	clean_content
0	Sorry chikku, my cell got some problem thts y ...	sorri chikku cell got problem tht nt abl repli...
1	Yes ammae....life takes lot of turns you can o...	ye amma life take lot turn sit tri hold steer
2	Maglaro sa pinakamalaking platform at makuha a...	maglaro sa pinakamalak platform makuha ang p n...
3	I'm used to it. I just hope my agents don't dr...	use hope agent nt drop sinc book thing year wh...
4	Have you seen who's back at Holby?!	seen back holbi

Hình 4: Dữ liệu SMS trước và sau khi được làm sạch

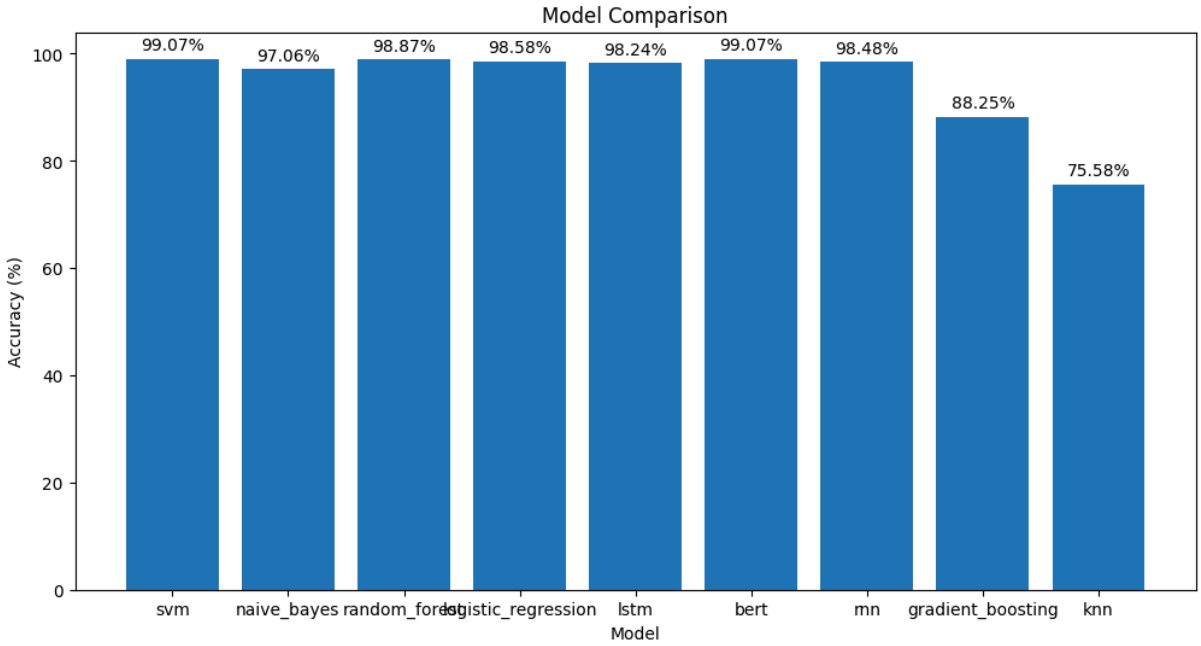


Hình 5: Các keyword phổ biến trong tin nhắn SMS rác



Hình 6: Các keyword phổ biến trong tin nhắn SMS hợp lệ

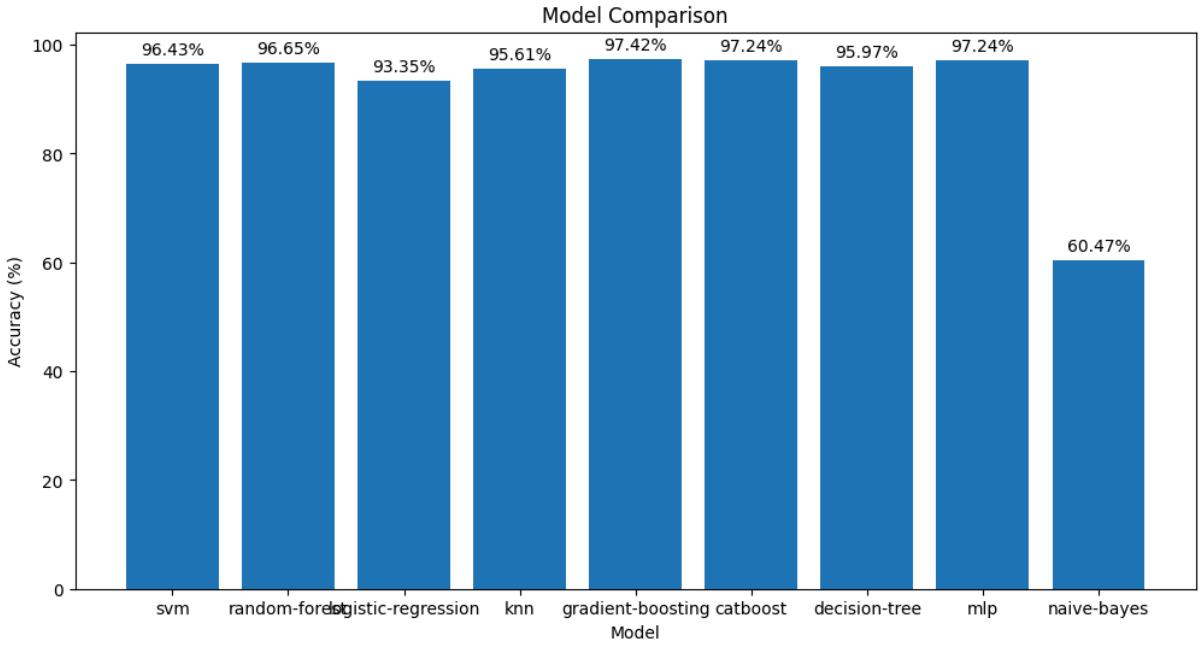
3.2. Huấn luyện mô hình



Hình 7: So sánh hiệu suất của các mô hình học máy trên dữ liệu SMS

	Model	Accuracy (%)	F1 Score (%)	Recall (%)	Precision (%)
0	svm	99.069995	99.069960	99.069995	99.077950
1	naive_bayes	97.063142	97.063134	97.063142	97.063870
2	random_forest	98.874205	98.874068	98.874205	98.899007
3	logistic_regression	98.580519	98.580403	98.580519	98.597344
4	lstm	98.237885	98.237883	98.237885	98.238069
5	bert	99.069995	99.069975	99.069995	99.073801
6	rnn	98.482624	98.482591	98.482624	98.486381
7	gradient_boosting	88.252570	88.091648	88.252570	90.449179
8	knn	75.575135	74.029065	75.575135	83.593561

Hình 8: Kết quả chi tiết huấn luyện mô hình phân loại tin nhắn SMS



Hình 9: So sánh hiệu suất của các mô hình học máy trên dữ liệu URL

	Model	Accuracy (%)	F1 Score (%)	Recall (%)	Precision (%)
0	svm	96.426956	96.420746	96.426956	96.448365
1	random-forest	96.653098	96.651614	96.653098	96.652698
2	logistic-regression	93.351425	93.338984	93.351425	93.363938
3	knn	95.612845	95.612129	95.612845	95.611771
4	gradient-boosting	97.421981	97.417500	97.421981	97.447211
5	catboost	97.241067	97.238377	97.241067	97.247380
6	decision-tree	95.974672	95.974455	95.974672	95.974278
7	mlp	97.241067	97.236993	97.241067	97.258254
8	naive-bayes	60.470375	55.816592	60.470375	78.842023

Hình 10: Kết quả chi tiết huấn luyện mô hình phân loại URL

3.2.1. Đánh giá hiệu quả của các mô hình

- SVM: Hiệu suất rất cao với cả Accuracy (99.07%), F1 Score (99.07%), Recall (99.07%), và Precision (99.08%). Điều này cho thấy mô hình SVM hoạt động rất tốt trong phân loại SMS và đạt sự cân bằng giữa Recall và Precision.

- Naive Bayes: Cũng có hiệu suất cao, với Accuracy, F1 Score, Recall, và Precision đều xấp xỉ 97.06%. Naive Bayes là một mô hình đơn giản nhưng hoạt động rất tốt với dữ liệu văn bản, vì vậy kết quả này là phù hợp.
- Random Forest: Cũng có kết quả tốt với Accuracy 98.87% và F1 Score 98.87%. Mô hình này có thể hơi phức tạp hơn nhưng cung cấp hiệu suất mạnh mẽ và khả năng tổng quát hóa cao.
- Logistic Regression: Có Accuracy và F1 Score khoảng 98.58%, đây là một mô hình tuyến tính hiệu quả và dễ diễn giải, nhưng hiệu suất thấp hơn một chút so với SVM và Random Forest.
- LSTM: Đạt Accuracy và F1 Score là 98.24%, cho thấy LSTM hoạt động tốt khi xử lý chuỗi dữ liệu văn bản. Tuy nhiên, hiệu suất này không cao hơn nhiều so với các mô hình truyền thống như SVM hoặc Random Forest.
- BERT: Hiệu suất của BERT (99.07%) rất cao, tương đương với SVM, cho thấy mô hình này nắm bắt tốt ngữ cảnh trong dữ liệu văn bản. Tuy nhiên, việc sử dụng BERT yêu cầu nhiều tài nguyên tính toán.
- RNN: Hiệu suất của RNN là 98.48%, thấp hơn so với LSTM và BERT, điều này là do RNN gặp vấn đề khi xử lý các chuỗi dài và mất ngữ cảnh trong một số trường hợp.
- Gradient Boosting: Hiệu suất thấp hơn đáng kể, với Accuracy 88.25% và F1 Score 88.09%. Đây có thể là do mô hình không phù hợp lắm với kiểu dữ liệu văn bản hoặc chưa được tối ưu đúng cách.
- KNN: Kết quả thấp nhất, với Accuracy 75.58% và F1 Score 74.03%, cho thấy mô hình KNN không phù hợp với bài toán này, đặc biệt khi dữ liệu văn bản có không gian đặc trưng cao.

3.2.2. Đề xuất cải tiến mô hình

Với SVM, ta có thể thử điều chỉnh các siêu tham số như loại kernel (Radial Basis Function, Polynomial) hoặc hệ số phạt (C) để tối ưu hóa thêm. Thực tế, sau khi sử dụng mô hình GridSearchCV để tinh chỉnh siêu tham số nhằm tìm ra tham số tối ưu, mô hình SVM có thể đạt hiệu suất cao hơn so với trước khi tinh chỉnh:

```
Fitting 5 folds for each of 30 candidates, totalling 150 fits
[CV] END .................C=0.01, gamma=scale, kernel=linear; total time=  19.5s
[CV] END .................C=0.01, gamma=scale, kernel=linear; total time=  19.3s
[CV] END .................C=0.01, gamma=scale, kernel=linear; total time=  20.1s
[CV] END .................C=0.01, gamma=scale, kernel=linear; total time=  18.8s
[CV] END .................C=0.01, gamma=scale, kernel=linear; total time=  19.4s
[CV] END .....................C=0.01, gamma=scale, kernel=rbf; total time=  21.5s
[CV] END .....................C=0.01, gamma=scale, kernel=rbf; total time=  20.7s
[CV] END .....................C=0.01, gamma=scale, kernel=rbf; total time=  20.4s
[CV] END .....................C=0.01, gamma=scale, kernel=rbf; total time=  20.3s
[CV] END .....................C=0.01, gamma=scale, kernel=rbf; total time=  20.5s
[CV] END ...................C=0.01, gamma=scale, kernel=poly; total time=  20.8s
[CV] END ...................C=0.01, gamma=scale, kernel=poly; total time=  21.6s
[CV] END ...................C=0.01, gamma=scale, kernel=poly; total time=  18.9s
[CV] END ...................C=0.01, gamma=scale, kernel=poly; total time=  18.8s
[CV] END ...................C=0.01, gamma=scale, kernel=poly; total time=  18.7s
[CV] END ..................C=0.01, gamma=auto, kernel=linear; total time=  18.7s
[CV] END ..................C=0.01, gamma=auto, kernel=linear; total time=  19.4s
[CV] END ..................C=0.01, gamma=auto, kernel=linear; total time=  18.6s
[CV] END ..................C=0.01, gamma=auto, kernel=linear; total time=  18.4s
[CV] END ..................C=0.01, gamma=auto, kernel=linear; total time=  19.2s
[CV] END ......................C=0.01, gamma=auto, kernel=rbf; total time=  20.8s
[CV] END ......................C=0.01, gamma=auto, kernel=rbf; total time=  20.6s
[CV] END ......................C=0.01, gamma=auto, kernel=rbf; total time=  20.7s
[CV] END ......................C=0.01, gamma=auto, kernel=rbf; total time=  21.2s
...
[CV] END ......................C=100, gamma=auto, kernel=poly; total time=  19.2s
[CV] END ......................C=100, gamma=auto, kernel=poly; total time=  19.4s
[CV] END ......................C=100, gamma=auto, kernel=poly; total time=  19.4s
Best parameters found:  {'C': 10, 'gamma': 'scale', 'kernel': 'rbf'}
```

Hình 11: Tinh chỉnh siêu tham số của mô hình SVM với GridSearchCV

```
Evaluating svm...
Accuracy: 0.9951052373959863
F1 Score: 0.9951052233232975
Recall: 0.9951052373959863
Precision: 0.9951128385971205
```

	Model	Accuracy (%)	F1 Score (%)	Recall (%)	Precision (%)
0	svm	99.510524	99.510522	99.510524	99.511284

Hình 12: Hiệu suất của mô hình SVM tăng mạnh sau khi tinh chỉnh siêu tham số. Accuracy tăng từ 99.07% lên 99.51%

Với BERT, ta có thể thử điều chỉnh quá trình fine-tuning hoặc dùng phiên bản nhỏ hơn của BERT như DistilBERT để giảm thời gian huấn luyện.

Cải tiến kỹ thuật tiền xử lý:

- Mô hình cần cải thiện tiền xử lý văn bản, chẳng hạn như sử dụng stemming, lemmatization, hoặc loại bỏ các từ dư thừa để giúp mô hình phân biệt rõ hơn giữa spam và ham.
- Cần thử nghiệm với các biểu diễn văn bản khác, như sử dụng từ nhúng (word embeddings) như Word2Vec hoặc GloVe.

Ta cũng có thể kết hợp các mô hình khác nhau (như SVM và Naive Bayes) để tận dụng ưu điểm của từng mô hình và cải thiện độ chính xác.

Ngoài ra, do hạn chế của nguồn dữ liệu và số lượng mẫu, việc tăng cường dữ liệu bằng cách thu thập thêm dữ liệu hoặc tạo dữ liệu tổng hợp có thể giúp cải thiện hiệu suất của mô hình.

3.2.3. Kết luận

- Hiệu suất hàng đầu: SVM và BERT có hiệu suất cao nhất. Tuy nhiên, SVM đơn giản hơn, có kết quả tốt hơn và yêu cầu ít tài nguyên hơn so với BERT. Do đó tôi chọn SVM là mô hình chính cho hệ thống phân loại tin nhắn SMS và URL.
- Mô hình học sâu: LSTM và RNN hoạt động tốt nhưng không vượt trội so với các mô hình truyền thống như SVM hoặc Random Forest. Điều này có thể là do kích thước tập dữ liệu chưa đủ lớn để phát huy toàn bộ tiềm năng của các mô hình học sâu.

3.3. Xây dựng chương trình phân loại

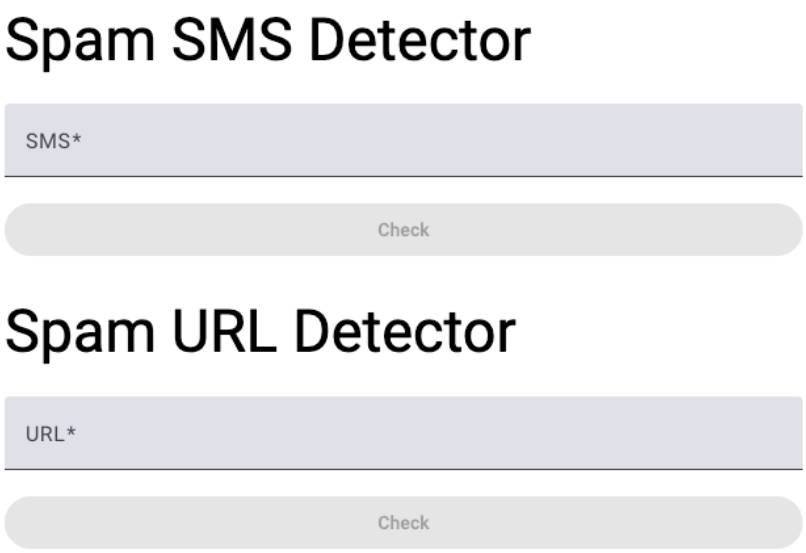
Sử dụng thư viện scikit-learn để xây dựng mô hình học máy SVM:

```
class SMSMLClassifier(SMSClassifier):
  def train(self, X: np.ndarray, Y: np.ndarray) -> None:
    # Nếu vectorizer chưa được khởi tạo, khởi tạo nó bằng TfidfVectorizerFactory
    if self.vectorizer is None:
      self.vectorizer = TfidfVectorizerFactory().vectorizer
```

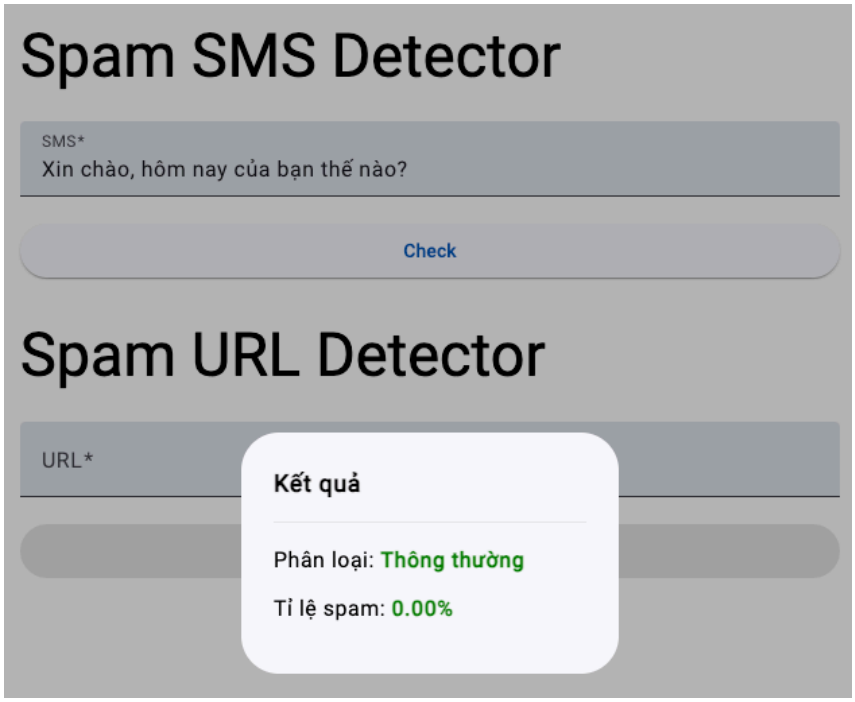
```
    # Huấn luyện vectorizer trên dữ liệu X
    self.vectorizer.fit(X.copy())
    # Chuyển đổi dữ liệu X thành dạng TF-IDF
    X_tfidf_transformed = self.vectorizer.transform(X.copy())

    # Huấn luyện mô hình trên dữ liệu đã được chuyển đổi
    self.model.fit(X_tfidf_transformed, Y)

  def predict(self, X: np.ndarray) -> np.ndarray[int]:
    # Chuyển đổi dữ liệu X thành dạng TF-IDF
    X_tfidf_transformed = self.vectorizer.transform(X.copy())
    # Dự đoán nhãn cho dữ liệu đã được chuyển đổi
    return self.model.predict(X_tfidf_transformed)

class SMSMLSVMClassifier(SMSMLClassifier):
    def __init__(self, model_dir):
        super().__init__("svm", model_dir)
        self.model = SVC(kernel="rbf", C=10, gamma="scale", random_state=42,
probability=True)
```

3.4. Phát triển giao diện người dùng

Giao diện người dùng được phát triển bằng framework Angular và FastAPI cho phép người dùng tương tác với hệ thống, xem kết quả phân tích, và báo cáo tin nhắn đáng ngờ.



Hình 13: Giao diện người dùng của ứng dụng



Hình 14: Thông báo kết quả phân loại tin nhắn SMS

CHƯƠNG 4. KẾT LUẬN

4.1. Tóm tắt kết quả

4.1.1. Tóm tắt các thành tựu của dự án và những điểm nổi bật của ứng dụng

Trong quá trình thực hiện dự án này, tôi đã đạt được nhiều thành tựu quan trọng. Hệ thống phát hiện lừa đảo trong tin nhắn SMS và URL đã được phát triển và triển khai thành công, sử dụng các kỹ thuật học máy và xử lý ngôn ngữ tự nhiên tiên tiến. Các điểm nổi bật của hệ thống bao gồm:

- Hiệu suất cao: Hệ thống đạt được độ chính xác cao trong việc phân loại tin nhắn SMS và URL là hợp pháp hoặc lừa đảo, với mô hình SVM và BERT đạt hiệu suất hàng đầu.
- Khả năng mở rộng: Hệ thống được thiết kế theo kiến trúc modular, cho phép dễ dàng mở rộng và tích hợp các mô hình và thuật toán mới trong tương lai.
- Giao diện người dùng thân thiện: Giao diện người dùng được phát triển đơn giản và dễ sử dụng, cho phép người dùng dễ dàng tương tác với hệ thống và xem kết quả phân tích.

4.1.2. Thách thức và hướng phát triển

Việc phát hiện lừa đảo trong tin nhắn SMS vẫn còn nhiều thách thức, bao gồm:

- Sự tinh vi của các kỹ thuật lừa đảo: Những kẻ lừa đảo liên tục phát triển các kỹ thuật mới để vượt qua các biện pháp bảo mật, đòi hỏi hệ thống phải được cập nhật thường xuyên.
- Sự đa dạng của ngôn ngữ và nội dung: Tin nhắn SMS có thể được viết bằng nhiều ngôn ngữ khác nhau và có nhiều nội dung khác nhau, khiến việc phân tích trở nên khó khăn và phức tạp.

- Bảo vệ quyền riêng tư của người dùng: Việc phân tích tin nhắn SMS cần phải được thực hiện theo cách bảo vệ quyền riêng tư của người dùng, đảm bảo rằng dữ liệu cá nhân không bị lạm dụng.

Các hướng phát triển trong tương lai bao gồm:

- Phát triển các thuật toán học máy mạnh mẽ hơn để phát hiện lừa đảo, bao gồm việc sử dụng các mô hình học sâu tiên tiến và các kỹ thuật học không giám sát.
- Sử dụng phân tích hành vi nâng cao để xác định các mẫu hoạt động đáng ngờ, giúp cải thiện độ chính xác và hiệu quả của hệ thống.
- Hợp tác giữa các nhà cung cấp dịch vụ, các cơ quan chính phủ và các nhà nghiên cứu để chia sẻ thông tin và phát triển các biện pháp bảo mật hiệu quả hơn, tạo ra một môi trường an toàn hơn cho người dùng.

4.1.3. Đề xuất các phương pháp nâng cao độ chính xác của mô hình

Để nâng cao độ chính xác của mô hình, tôi đề xuất nghiên cứu và áp dụng các kỹ thuật học máy tiên tiến, bao gồm:

- Sử dụng các mô hình học sâu như Transformer và các biến thể của nó để cải thiện khả năng hiểu ngữ cảnh và ngữ nghĩa của tin nhắn.
- Áp dụng các kỹ thuật tăng cường dữ liệu để tạo ra thêm các mẫu huấn luyện, giúp mô hình học tốt hơn từ dữ liệu đa dạng.

4.1.4. Ứng dụng cho các nền tảng khác

Ngoài việc phát hiện lừa đảo trong tin nhắn SMS và URL, hệ thống có thể được mở rộng để ứng dụng cho các nền tảng khác như email, mạng xã hội và các ứng dụng nhắn tin khác. Điều này sẽ giúp bảo vệ người dùng trên nhiều nền tảng khác nhau, đảm bảo an toàn và bảo mật thông tin cá nhân.

Việc mở rộng ứng dụng cho các nền tảng khác cũng đòi hỏi nghiên cứu và phát triển thêm các kỹ thuật phân tích và mô hình học máy phù hợp với từng nền tảng, đảm bảo hiệu suất và độ chính xác cao trong việc phát hiện lừa đảo.

4.2. Cảm ơn

4.2.1. Lời cảm ơn đến những người đã hỗ trợ trong quá trình thực hiện đồ án

TÀI LIỆU THAM KHẢO

[1] “Grumbletext.” [Online]. Available: https://www.grumbletext.co.uk/

[2] “National University of Singapore.” [Online]. Available: https://www.comp.nus.edu.sg/

[3] “A Corpus Linguistics Study of SMS Text Messaging.” [Online]. Available: https://etheses.bham.ac.uk/id/eprint/253/1/Tagg09PhD.pdf

[4] “SMS Spam Collection Dataset.” [Online]. Available: https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset

[5] “Phishing Website Detector.” [Online]. Available: https://www.kaggle.com/datasets/eswarchandt/phishing-website-detector