**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KHOA HỌC VÀ KĨ THUẬT THÔNG TIN**



**BÁO CÁO ĐỒ ÁN**

**Đề tài:**

**Phân tích mã độc Android bằng NLP**

Giảng viên hướng dẫn: **TS.Nguyễn Tấn Cầm**

Lớp: **IE105.O21**

Sinh viên thực hiện:

**Nguyễn Công Nam Triều – 22521533**

**Nguyễn Hoàng Trọng Sơn – 22521252**

**Nguyễn Hạc Phong – 22521087**

*TP. Hồ Chí Minh, ngày 15 tháng 5 năm 2024*

**NHẬN XÉT TỪ GIẢNG VIÊN**

……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..……………………………………………………………………………………………..……………………………………………………………………………………………..……………………………………………………………………………………………..……………………………………………………………………………………………..……………………………………………………………………………………………..……………………………………………………………………………………………..……………………………………………………………………………………………..……………………………………………………………………………………………..……………………………………………………………………………………………..……………………………………………………………………………………………..……………………………………………………………………………………………..……………………………………………………………………………………………..……………………………………………………………………………………………..……………………………………………………………………………………………..……………………………………………………………………………………………..……………………………………………………………………………………………..……………………………………………………………………………………………..……………………………………………………………………………………………..……………………………………………………………………………………………..……………………………………………………………………………………………..……………………………………………………………………………………………..……………………………………………………………………………………………..……………………………………………………………………………………………..……………………………………………………………………………………………..……………………………………………………………………

MỤC LỤC

[MỤC LỤC i](#_Toc168002583)

[TÓM TẮT 1](#_Toc168002584)

[CHƯƠNG 1: TỔNG QUAN 2](#_Toc168002585)

[1.1 Giới thiệu 2](#_Toc168002586)

[1.2 Bối cảnh 2](#_Toc168002587)

[1.2.1 Bối cảnh 2](#_Toc168002588)

[1.3 Mục tiêu 3](#_Toc168002589)

[1.3.1 Mục tiêu 3](#_Toc168002590)

[1.4 Các nghiên cứu liên quan 4](#_Toc168002591)

[1.4.1 Khảo sát Tài liệu 4](#_Toc168002592)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 5](#_Toc168002593)

[2.1 Natural Processing Language (NLP) 5](#_Toc168002594)

[2.2 Các phương pháp và kỹ thuật trong Machine Learning 7](#_Toc168002595)

[2.3 Ứng dụng của NLP trong Machine Learning 10](#_Toc168002596)

[CHƯƠNG 3: PHƯƠNG PHÁP ĐỀ XUẤT 13](#_Toc168002597)

[3.1 Bộ dữ liệu 13](#_Toc168002598)

[3.2 Phân tích Dữ liệu Khám phá (EDA): 15](#_Toc168002599)

[3.3 Phương pháp luận 16](#_Toc168002600)

[3.4 Kết quả và Phân tích 18](#_Toc168002601)

[CHƯƠNG 4: KẾT LUẬN CHUNG 17](#_Toc168002602)

[4.1 Kết luận 17](#_Toc168002603)

[4.2 Hướng phát triển: 17](#_Toc168002604)

[TÀI LIỆU THAM KHẢO 18](#_Toc168002605)

[BẢNG PHÂN CÔNG 19](#_Toc168002606)

TÓM TẮT

Với sự gia tăng của phần mềm độc hại nhắm vào các thiết bị Android, việc phát hiện Android malware đã trở thành một lĩnh vực nghiên cứu quan trọng. Việc phát hiện phần mềm độc hại trên Android là cần thiết để bảo vệ quyền riêng tư của người dùng, bảo vệ an ninh dữ liệu và duy trì hiệu suất thiết bị. Tuy nhiên, nhiệm vụ phát hiện Android malware bị phức tạp bởi nhiều yếu tố, bao gồm sự đa dạng của các gia đình malware, tính đa hình của malware và các chiến thuật lẩn tránh do các tác giả malware sử dụng. Do đó, nhiều nghiên cứu đã chỉ ra những thách thức liên quan đến việc phát hiện Android malware.

Bài viết này khám phá việc ứng dụng công nghệ Xử lý Ngôn Ngữ Tự Nhiên (Natural Language Processing - NLP) để xác định các loại mã độc, qua đó giải quyết thách thức trong việc phân biệt và nhận diện nhiều dạng mã độc khác nhau. Cách tiếp cận được đề xuất liên quan đến việc phân tích cả các yếu tố tĩnh và động của mã độc, bằng cách dùng NLP để phân tích và hiểu các thuộc tính ngôn ngữ có trong mã độc.

Kết quả của nghiên cứu cho thấy cách thức áp dụng NLP trong việc nhận diện mã độc đem lại hiệu quả cao, với năng lực phân biệt chính xác các loại mã độc hại. Sự phối hợp giữa các yếu tố tĩnh và động cùng với việc sử dụng NLP mang lại một phương pháp toàn diện và hiệu quả hơn trong công cuộc phát hiện và chống lại các loại mã độc đa dạng và biến thể.

# TỔNG QUAN

Trong chương này, nhóm tác giả trình bày sơ lược về phương pháp áp dụng NLP trong việc phát hiện các loại mã độc (malware detection). Nhóm tác giả đặt mục tiêu thực hiện trên nhiều mô hình nhằm tìm ra mô hình tối ưu nhất.

## Giới thiệu

Trong thời gian gần đây, phương pháp xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP) đã được áp dụng trong lĩnh vực phát hiện mã độc (malware detection) để tăng cường khả năng phát hiện và phân loại các mẫu mã độc. NLP trong malware detection sử dụng các kỹ thuật và công cụ NLP để phân tích và hiểu các đặc điểm của mã độc dựa trên ngôn ngữ tự nhiên. Tuy nhiên, câu hỏi được đặt ra ở đây là liệu mô hình học máy nào khi kết hợp NLP sẽ cho ra kết quả tối ưu nhất trong việc phát hiện malware?

Trong đồ án này, nhóm tác giả sẽ tiến hành thực hiện kết hợp phương pháp NLP với các mô hình học máy phổ biến khác nhau, qua đó so sánh với các bài báo khác, cũng như để tìm ra mô hình tối ưu nhất.

## Bối cảnh

### Bối cảnh

Mối đe dọa từ các cuộc tấn công malware đã tăng lên đáng kể khi sử dụng thiết bị di động, đặc biệt là điện thoại thông minh Android, ngày càng gia tăng. Các ứng dụng có thể chứa phần mềm độc hại ẩn có thể làm hỏng thiết bị hoặc đánh cắp dữ liệu người dùng quan trọng. Các biện pháp bảo mật thông thường và phần mềm chống virus có hạn chế trong việc xác định và ngăn chặn những mối đe dọa này, những mối đe dọa ngày càng trở nên phức tạp hơn. Học máy đã xuất hiện như một chiến lược khả thi cho việc dự đoán phần mềm độc hại Android để đối phó với khó khăn này. Các thuật toán học máy có thể xem xét đa dạng các tính năng và đặc điểm của ứng dụng để phát hiện hoạt động có thể nguy hiểm. Điều này có thể giúp phát hiện malware mà các phương pháp truyền thống không thể làm.

Dù mối đe dọa của malware ngày càng tăng, hiện vẫn chưa có phương pháp đáng tin cậy và mạnh mẽ để phát hiện các ứng dụng độc hại. Tuy nhiên, với việc ứng dụng học máy trong nhiều lĩnh vực ngày càng tăng, vấn đề này có thể được giải quyết thông qua việc áp dụng các kỹ thuật học máy. Dự án của chúng tôi nhằm tiến hành một cuộc điều tra kỹ lưỡng và hệ thống về việc sử dụng học máy để phát hiện malware, với mục tiêu cuối cùng là phát triển một mô hình ML hiệu quả có khả năng phân loại chính xác các ứng dụng là lành tính (0) hoặc malware (1) dựa trên các quyền được yêu cầu. Nghiên cứu này đề xuất:

* Tiến hành một cuộc khảo sát sâu rộng và đánh giá về metadata và quyền của Android như là những chỉ báo của malware.
* Giới thiệu một chiến lược phát hiện malware dựa trên học máy sử dụng thông tin metadata công khai.
* Phân tích hiệu quả của mô hình này và đánh giá tiềm năng của nó như một bộ lọc giai đoạn đầu để phát hiện phần mềm độc hại Android.

## Mục tiêu

### Mục tiêu

* Cung cấp một mô hình học máy có khả năng xác định chính xác phần mềm độc hại Android dựa trên các đặc điểm hành vi, tính năng mã hóa và các đặc điểm khác.
* Thu thập và chuẩn bị một tập dữ liệu lớn về mẫu phần mềm độc hại Android và các ứng dụng tốt cho mục đích đào tạo và đánh giá mô hình học máy.
* Tiền xử lý dữ liệu và thực hiện EDA trên tập dữ liệu. Lựa chọn và trích xuất các cuộc gọi API, quyền, lưu lượng mạng, cuộc gọi hệ thống và các thông tin liên quan khác có thể phân biệt giữa các ứng dụng độc hại và lành tính.
* Sử dụng các chỉ số liên quan để đánh giá hiệu suất của mô hình học máy bao gồm độ chính xác, độ nhạy, điểm F1 và độ chính xác.
* Cải thiện độ chính xác, tốc độ và độ bền của mô hình học máy trong khi giảm thiểu sự quá khớp và thiên vị.
* Kiểm tra hiệu quả của mô hình học máy bằng cách sử dụng mẫu phần mềm độc hại Android thực tế và đánh giá hiệu suất của nó so với các phương pháp phát hiện malware tiên tiến khác.

## Các nghiên cứu liên quan

### Khảo sát Tài liệu

***[1]***: Dự đoán Phần mềm độc hại Android sử dụng các Kỹ thuật Học máy

Báo cáo này cung cấp một phân tích kỹ lưỡng về nhiều phương pháp học máy đã được áp dụng để dự đoán phần mềm độc hại Android. Các phương pháp học máy thường được sử dụng nhất để dự đoán phần mềm độc hại Android bao gồm SVM, thuật toán Decision Tree, mô hình phân lớp Naive Bayes và mô hình Random Forests. Quyền truy cập vào API, các yêu cầu hệ thống và lưu lượng mạng là một số đặc điểm đã được sử dụng để dự đoán phần mềm độc hại Android. Nghiên cứu này nhấn mạnh một số hạn chế của các nghiên cứu trước đây như sự thiếu hụt của các bộ dữ liệu đa dạng để đào tạo và kiểm tra các mô hình học máy và khó khăn trong việc xử lý phần mềm độc hại bị mã hóa và đa hình. Nó cũng chỉ ra rằng các kỹ thuật học máy có thể dự đoán hiệu quả phần mềm độc hại Android và độ chính xác của dự đoán có thể được cải thiện bằng cách sử dụng một sự kết hợp của các đặc điểm và thuật toán học máy. Các tác giả cho rằng phương pháp lựa chọn đặc trưng và giảm chiều dữ liệu có thể tăng cường hiệu quả của việc dự đoán phần mềm độc hại Android. Nghiên cứu nhấn mạnh một số vấn đề cần được giải quyết để nâng cao hiệu quả của các phương pháp này và cung cấp cái nhìn sâu sắc hữu ích vào tình trạng hiện tại của việc dự đoán phần mềm độc hại Android bằng kỹ thuật học máy.

***[2]***: Dự đoán Phần mềm độc hại Android Hiệu quả sử dụng Các thuật toán Học máy Tập hợp

Để phân loại các ứng dụng Android là độc hại hay lành tính, các tác giả kết hợp một số phương pháp phân loại như Decision Tree, K-nearest neighbors và SVM. Các tác giả đã kiểm tra bộ phân loại của họ trên một tập dữ liệu bao gồm hơn 17.000 ứng dụng Android để đánh giá chiến lược của họ. Họ phát hiện ra rằng phương pháp tập hợp của họ vượt trội hơn các thuật toán phân loại riêng lẻ được sử dụng trong tập hợp và đạt độ chính xác cao hơn 99% trong việc phát hiện phần mềm độc hại. Các tác giả cũng đánh giá các chiến lược học máy tiên tiến khác cho việc phát hiện phần mềm độc hại Android và phát hiện ra rằng chiến lược của họ cũng vượt trội hơn các kỹ thuật này. Họ kết luận rằng phương pháp tập hợp của họ có thể được sử dụng để tăng cường an ninh di động và là một phương pháp rất thành công để xác định phần mềm độc hại trong các ứng dụng Android.

# CƠ SỞ LÝ THUYẾT

## Natural Processing Language (NLP)

Tất cả các phương pháp xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP) trong malware detection nhắm vào mục tiêu là tận dụng thông tin ngôn ngữ tự nhiên trong mã độc để tăng cường khả năng phát hiện và phân loại các mẫu độc hại. Các phương pháp NLP này sử dụng các kỹ thuật và công cụ NLP để phân tích và hiểu các đặc điểm của mã độc dựa trên ngôn ngữ tự nhiên. Các loại phân tích đó gồm:

* Phân tích cú pháp (Syntax Analysis): Phương pháp này tập trung vào việc phân tích cấu trúc và quy tắc ngữ pháp của mã độc. Cú pháp được sử dụng để xác định các thành phần trong mã, bao gồm các câu lệnh, biểu thức, hàm, lớp, và các quy tắc cú pháp khác. Các công cụ như lexer và parser được sử dụng để phân tích cú pháp mã độc và xác định các đặc điểm không phù hợp hoặc đáng ngờ.
* Phân tích từ vựng (Lexical Analysis): Phương pháp này tập trung vào các từ và thuật ngữ trong mã độc. Các từ khóa đặc trưng, biểu thức chính quy, và từ điển từ khóa được sử dụng để tìm kiếm các từ và cụm từ có liên quan đến mã độc. Việc phân tích từ vựng giúp xác định các đặc điểm ngôn ngữ độc hại, như tên biến, hàm, lớp, hoặc các chuỗi ký tự đáng ngờ.
* Phân tích ngữ nghĩa (Semantic Analysis): Phương pháp này tập trung vào việc hiểu ý nghĩa của các câu và văn bản trong mã độc. Các kỹ thuật semantic parsing và semantic role labeling được sử dụng để phân tích và rút trích thông tin cần thiết từ các câu và văn bản liên quan đến mã độc. Điều này giúp xác định các hành vi, mục tiêu, và tác động tiềm năng của mã độc.
* Phân tích ngữ cảnh (Contextual Analysis): Phương pháp này đánh giá mã độc dựa trên ngữ cảnh mà nó xuất hiện. Các thuật toán học máy và mô hình ngữ cảnh được sử dụng để phân tích và xác định xem một đoạn mã có khả năng là độc hại hay không dựa trên các thông tin xung quanh nó. Ví dụ, việc phân tích ngữ cảnh có thể xác định liệu một hàm cụ thể có được gọi từ một đoạn mã độc khác hay không.
* Phân tích dựa trên mẫu (Pattern-based Analysis): Phương pháp này sử dụng các mẫu đã biết hoặc các quy tắc để phát hiện và phân loại các mẫu mã độc. Các mẫu đặc trưng và quy tắc mô tả các đặc điểm và hành vi của mã độc được sử dụng để so khớp và tìm kiếm các mẫu tương tự trong mã đích. Các kỹ thuật như regular expressions (biểu thức chính quy) và rule-based matching được sử dụng để áp dụng các mẫu này và phát hiện các mã độc. Tuy nhiên, việc áp dụng NLP trong lĩnh vực malware detection cũng đối mặt với một số thách thức. Mã độc ngày càng trở nên phức tạp và tiên tiến hơn, với sự phát triển của các kỹ thuật che dấu (obfuscation) và tấn công zero-day. Do đó, cần phải liên tục cập nhật và phát triển các kỹ thuật NLP mới nhằm đối phó với các mẫu mã độc mới.

Hơn nữa, việc sử dụng NLP trong malware detection cũng đòi hỏi kiến thức chuyên sâu về an ninh mạng và NLP. Để xử lý mã độc hiệu quả, các chuyên gia phải nắm vững kiến thức về ngôn ngữ tự nhiên, các kỹ thuật xử lý ngôn ngữ tự nhiên, các thuật toán học máy, và các công cụ NLP.

Một số ứng dụng cụ thể của NLP trong malware detection bao gồm:

* Phát hiện mã độc dựa trên phân tích ngữ nghĩa của các hàm, lớp, hoặc các đoạn mã.
* Phát hiện mã độc dựa trên phân tích từ vựng, tìm kiếm các từ khóa đặc trưng hoặc các biểu thức chính quy trong mã độc.
* Phát hiện mã độc dựa trên phân tích cú pháp, xác định các câu lệnh hoặc cấu trúc ngôn ngữ độc hại.
* Phân loại và gom nhóm các mẫu mã độc dựa trên các đặc điểm ngôn ngữ, hành vi, hoặc mục tiêu của chúng.
* Xác định các quan hệ và mối liên hệ giữa các thành phần mã độc, như các hàm gọi nhau hoặc tương tác với nhau.

Tổng quan, NLP trong malware detection đóng vai trò quan trọng trong việc nâng cao khả năng phát hiện và phân loại mã độc. Bằng cách áp dụng các phương pháp và công cụ NLP, chúng ta có thể phân tích và hiểu mã độc dựa trên thông tin ngôn ngữ tự nhiên, từ đó nâng cao hiệu suất và độ chính xác của quá trình phát hiện malware.

## Các phương pháp và kỹ thuật trong Machine Learning

Machine Learning (ML) bao gồm nhiều phương pháp và kỹ thuật khác nhau để phân tích và dự đoán từ dữ liệu. Trong phần này, chúng ta sẽ tìm hiểu chi tiết về 3 phương pháp quan trọng: Decision Trees, Logistic Regression và Gaussian Naive Bayes. Ngoài ra, chúng ta cũng sẽ tóm tắt sơ lược về một vài kĩ thuật như Principal Component Analysis (PCA), Support Vector Machine (SVM) và Multilayer Perceptron (MLP).

* Decision Trees
* Khái niệm: Decision Trees là một mô hình học máy dựa trên cấu trúc cây quyết định. Mỗi nút trong cây đại diện cho một đặc điểm của dữ liệu, mỗi nhánh đại diện cho một kết quả của quyết định tại nút đó, và mỗi lá (leaf) đại diện cho một nhãn hoặc giá trị dự đoán.
* Cách sử dụng:
* Phân loại: Decision Trees được sử dụng rộng rãi trong phân loại. Cây quyết định chia dữ liệu thành các nhóm nhỏ hơn dựa trên các đặc điểm, giúp xác định nhãn của dữ liệu đầu vào.
* Hồi quy: Ngoài phân loại, Decision Trees cũng có thể được sử dụng cho các bài toán hồi quy, dự đoán giá trị liên tục dựa trên các đặc điểm của dữ liệu.
* Ưu điểm:
* Dễ hiểu và giải thích: Các mô hình Decision Trees dễ hiểu và có thể được hình dung trực quan.
* Xử lý dữ liệu phi tuyến: Decision Trees có thể xử lý các mối quan hệ phi tuyến giữa các đặc điểm và mục tiêu.
* Nhược điểm: dễ bị overfitting: Nếu không được điều chỉnh đúng cách, Decision Trees có thể dễ bị overfitting, dẫn đến hiệu suất kém trên dữ liệu mới.
* Logistic Regression
* Khái niệm: Logistic Regression là một thuật toán học máy dùng để phân loại nhị phân. Thuật toán này dựa trên mô hình xác suất, sử dụng hàm sigmoid để dự đoán xác suất của một biến nhị phân (0 hoặc 1).
* Cách sử dụng:
* Phân loại nhị phân: Logistic Regression được sử dụng để phân loại các đối tượng vào một trong hai nhóm. Ví dụ, phân loại email thành spam hoặc không spam, hoặc xác định xem một bệnh nhân có bệnh hay không.
* Phân tích hồi quy: Mặc dù được gọi là hồi quy, Logistic Regression chủ yếu được sử dụng cho phân loại, không phải dự đoán giá trị liên tục.
* Ưu điểm:
* Hiệu quả với dữ liệu tuyến tính: Logistic Regression hoạt động tốt với các mối quan hệ tuyến tính giữa các đặc điểm và biến mục tiêu.
* Tính toán nhanh: Mô hình này tính toán nhanh và hiệu quả trên các tập dữ liệu lớn.
* Nhược điểm: hạn chế với dữ liệu phi tuyến: Logistic Regression không hiệu quả với các mối quan hệ phi tuyến, yêu cầu kỹ thuật khác như SVM hoặc neural networks.
* Gaussian Naive Bayes
* Khái niệm: Naive Bayes là một nhóm các thuật toán phân loại dựa trên định lý Bayes, giả định rằng các đặc điểm đầu vào là độc lập với nhau. Gaussian Naive Bayes, một biến thể của Naive Bayes, giả định rằng các đặc điểm tuân theo phân phối chuẩn (Gaussian distribution).
* Cách sử dụng:
* Phân loại văn bản: Gaussian Naive Bayes thường được sử dụng trong các bài toán phân loại văn bản, chẳng hạn như phân loại email spam, phân loại tài liệu, và phân tích cảm xúc.
* Phân loại nhãn: Được sử dụng để phân loại dữ liệu thành nhiều nhãn khác nhau dựa trên các đặc điểm đầu vào.
* Ưu điểm:
* Hiệu quả với dữ liệu lớn: Gaussian Naive Bayes có thể xử lý nhanh các tập dữ liệu lớn.
* Đơn giản và hiệu quả: Mô hình đơn giản nhưng hiệu quả trong nhiều bài toán phân loại.
* Nhược điểm:
* Giả định độc lập: Giả định rằng các đặc điểm đầu vào là độc lập có thể không chính xác, ảnh hưởng đến hiệu suất của mô hình.
* Các kỹ thuật khác trong Machine Learning:
* Principal Component Analysis (PCA)
* Khái niệm: PCA là một kỹ thuật giảm chiều dữ liệu dùng để chuyển đổi một tập dữ liệu có kích thước lớn thành một tập dữ liệu nhỏ hơn mà vẫn giữ lại được các thông tin quan trọng nhất.
* Cách sử dụng: giảm chiều dữ liệu: PCA được sử dụng để giảm số lượng biến trong dữ liệu, giúp tăng tốc độ xử lý và giảm bớt sự phức tạp của mô hình ML.
* Support Vector Machine (SVM)
* Khái niệm: SVM là một thuật toán học máy dùng cho các bài toán phân loại và hồi quy, tìm ra một siêu phẳng (hyperplane) tối ưu để phân chia các lớp dữ liệu trong không gian nhiều chiều.
* Cách sử dụng: Phân loại nhị phân: SVM thường được sử dụng để phân loại văn bản và phát hiện spam. Nó cũng có thể được mở rộng để sử dụng trong các bài toán phân loại đa lớp.
* Multilayer Perceptron (MLP)
* Khái niệm: MLP là một loại mạng neural nhân tạo bao gồm nhiều lớp các perceptron. Mỗi perceptron là một đơn vị tính toán đơn giản tương tự như một neuron trong não người.
* Cách sử dụng: Phân loại và hồi quy phức tạp: MLP được sử dụng cho cả các bài toán phân loại và hồi quy phức tạp, có thể học các đặc điểm phi tuyến tính từ dữ liệu và thực hiện phân loại hoặc dự đoán dựa trên dữ liệu văn bản.
* Kết luận:

Các phương pháp và kỹ thuật trong Machine Learning như Decision Trees, Logistic Regression, Gaussian Naive Bayes, PCA, SVM, và MLP đều có khả năng ứng dụng vào xử lý và phân tích dữ liệu ngôn ngữ tự nhiên (NLP). Việc kết hợp các kỹ thuật này với NLP giúp nâng cao hiệu suất và độ chính xác của các mô hình, mở ra nhiều cơ hội ứng dụng trong các lĩnh vực khác nhau.

## Ứng dụng của NLP trong Machine Learning

Ứng dụng của Xử lý Ngôn ngữ Tự nhiên (NLP) trong Học Máy (ML) đã mở ra nhiều cơ hội để phân tích và hiểu sâu hơn các tập dữ liệu ngôn ngữ tự nhiên. NLP không chỉ giúp xử lý và phân tích ngôn ngữ tự nhiên mà còn hỗ trợ ML trong việc phát hiện các mẫu, dự đoán, và ra quyết định thông qua dữ liệu văn bản. Dưới đây là một số cách mà NLP được ứng dụng vào Machine Learning:

* Logistic Regression
* Ứng dụng:
* Phân loại ứng dụng Android: Logistic Regression là một mô hình đơn giản và hiệu quả để phân loại ứng dụng Android. Ví dụ, nó có thể phân loại ứng dụng thành phần mềm an toàn hoặc phần mềm độc hại dựa trên các đặc điểm như số lượng quyền truy cập nguy hiểm, đánh giá người dùng, và mô tả ứng dụng.
* Xác định mức độ an toàn: Logistic Regression có thể xác định mức độ an toàn của một ứng dụng dựa trên các đặc điểm như quyền truy cập và các đánh giá từ người dùng, giúp người dùng có cái nhìn tổng quan về mức độ rủi ro của ứng dụng.
* Phương pháp:
* Vector hóa văn bản: Trước khi áp dụng Logistic Regression, văn bản cần được biến đổi thành các vector số thông qua các phương pháp như TF-IDF hoặc Word Embeddings (Word2Vec, GloVe).
* Xử lý dữ liệu: Các bước tiền xử lý như Tokenization, loại bỏ từ dừng, và Lemmatization giúp chuẩn bị dữ liệu văn bản để áp dụng mô hình Logistic Regression.
* Decision Trees
* Ứng dụng:
* Phân loại phần mềm độc hại: Decision Trees có thể phân loại các ứng dụng Android thành an toàn hoặc độc hại dựa trên các đặc điểm như quyền truy cập, đánh giá người dùng, và thông tin mô tả. Ví dụ, Decision Trees có thể xác định và phân loại các ứng dụng dựa trên từ khóa, quyền truy cập, và cấu trúc câu trong mô tả ứng dụng.
* Phân tích mối quan hệ: Decision Trees giúp nhận dạng các mối quan hệ giữa các đặc điểm khác nhau của ứng dụng, như quyền truy cập và loại ứng dụng, từ đó giúp hiểu rõ hơn về cách các yếu tố này tương tác và ảnh hưởng lẫn nhau.
* Phương pháp:
* Phân tích cú pháp và từ vựng: Trước khi áp dụng Decision Trees, cần thực hiện phân tích cú pháp và từ vựng để xác định các đặc điểm ngôn ngữ quan trọng.
* Tạo cây quyết định: Sử dụng các đặc điểm ngôn ngữ đã được phân tích để xây dựng cây quyết định, giúp phân loại và dự đoán dựa trên dữ liệu văn bản.
* Gaussian Naive Bayes
* Ứng dụng:
* Phân loại văn bản: Naive Bayes là một mô hình rất phổ biến và hiệu quả trong phân loại văn bản, đặc biệt là trong các bài toán có nhiều lớp nhãn như phân loại tài liệu, email spam, và tin tức.
* Phân tích cảm xúc: Naive Bayes cũng được sử dụng rộng rãi trong phân tích cảm xúc của văn bản nhờ khả năng xử lý tốt các đặc điểm ngôn ngữ.
* Phương pháp:
* Xác suất từ vựng: Naive Bayes dựa trên xác suất xuất hiện của các từ trong văn bản để phân loại. Việc tính toán xác suất này được thực hiện sau khi văn bản đã được tiền xử lý và vector hóa.
* Đào tạo mô hình: Sử dụng các tập dữ liệu văn bản đã được gán nhãn để đào tạo mô hình Naive Bayes, sau đó áp dụng mô hình này để phân loại các văn bản mới.
* Các kỹ thuật khác
* Principal Component Analysis (PCA)
* Ứng dụng: PCA được sử dụng để giảm chiều dữ liệu, giúp giảm bớt sự phức tạp và tăng tốc độ xử lý khi làm việc với các tập dữ liệu văn bản lớn.
* Phương pháp:
* Vector hóa văn bản: Biến đổi văn bản thành các vector số.
* Giảm chiều dữ liệu: Áp dụng PCA để giảm số lượng chiều của các vector, giữ lại các thông tin quan trọng nhất.
* Support Vector Machine (SVM)
* Ứng dụng: SVM thường được sử dụng để phân loại văn bản, đặc biệt là trong các bài toán phân loại nhị phân như phân loại email spam.
* Phương pháp:
* Vector hóa văn bản: Sử dụng các phương pháp như TF-IDF hoặc Word Embeddings để biến đổi văn bản thành các vector số.
* Phân loại: Áp dụng SVM để phân loại văn bản dựa trên các vector đã được tạo.
* Multilayer Perceptron (MLP)
* Ứng dụng: MLP có thể được sử dụng trong các bài toán phân loại văn bản phức tạp, nơi các mô hình đơn giản như Logistic Regression hoặc Naive Bayes không đạt hiệu quả cao.
* Phương pháp:
* Vector hóa văn bản: Biến đổi văn bản thành các vector số.
* Mạng neural: Sử dụng MLP để học các đặc điểm phức tạp từ dữ liệu văn bản và thực hiện phân loại.
* Kết luận:

Sự kết hợp giữa NLP và Machine Learning đã mang lại những tiến bộ vượt bậc trong việc xử lý và hiểu dữ liệu ngôn ngữ tự nhiên. Việc ứng dụng các kỹ thuật NLP vào ML không chỉ giúp nâng cao hiệu suất và độ chính xác của các mô hình mà còn mở ra nhiều cơ hội ứng dụng trong các lĩnh vực khác nhau như kinh doanh, y tế, giáo dục, và nhiều ngành công nghiệp khác.

# PHƯƠNG PHÁP ĐỀ XUẤT

## Bộ dữ liệu

* Bộ dữ liệu này sử dụng dữ liệu có sẵn từ Kaggle **[5]**.
* Dữ liệu chứa thông tin về quyền cho gần ba mươi ngàn ứng dụng.
* Bộ dữ liệu chứa 184 đặc điểm từ class label, app title, package name, related apps,…v.v
* "Class" là một lớp mục tiêu nhị phân duy nhất định nghĩa ứng dụng lành tính (nhị phân 0) và phần mềm độc hại (nhị phân 1).
* A screenshot of a computer

  Description automatically generatedA graph of malware and malware

  Description automatically generatedCó 29,999 bản ghi, 9,999 chương trình lành tính và 20,000 phần mềm độc hại dựa vào bộ dữ liệu này và chúng tôi đã tự kiểm chứng nó và có kết quả như dưới đây:

Quá trình **Tiền xử lý, Trực quan hóa và Phân tích dữ liệu**: Dữ liệu đầu tiên được nhập từ **tệp CSV** và tải vào dataframe để dễ dàng sử dụng. Các thuộc tính cần thiết sau đó được trích từ bộ dữ liệu. Để hiểu rõ hơn về dữ liệu, một số biểu đồ được tạo ra. Dữ liệu được kiểm tra **để tìm giá trị null hoặc các giá trị bị thiếu** và bất kỳ giá trị nào như vậy được thay thế bằng giá trị trung bình của cột tương ứng. Sự phân bố của các ứng dụng phần mềm độc hại và lành tính trên các cài đặt khác nhau được phân tích và kết quả được trực quan hóa thông qua một loạt các biểu đồ được tạo bằng **Matplotlib** và **Seaborn**.

A graph with a green rectangular bar and red rectangular bar

Description automatically generated

A screenshot of a graph

Description automatically generated

## Phân tích Dữ liệu Khám phá (EDA):

**EDA** cho tập dữ liệu quyền của Android cung cấp thông tin chi tiết có giá trị về mối quan hệ giữa các tính năng khác nhau trong tập dữ liệu và giúp chúng tôi xác định các tính năng quan trọng nhất để dự đoán xếp hạng ứng dụng. Nó cũng cung cấp một nền tảng cho hơn nữa Phân tích bằng kỹ thuật học máy.

A group of graphs and diagrams

Description automatically generated

## Phương pháp luận

Sau quá trình tiền xử lý dữ liệu, nó được chia thành các bộ dữ liệu kiểm tra và training theo tỷ lệ 8:2. Chúng tôi đã thử nghiệm cả các kỹ thuật lấy mẫu dưới và lấy mẫu quá trên bộ dữ liệu nhưng kết quả không hứa hẹn. Sau đó, chúng tôi áp dụng các bộ phân loại khác nhau bao gồm Logistic Regression, Decision Trees và Naive Bayes nhưng kết quả không thỏa mãn. Sau khi kiểm tra kỹ bộ dữ liệu, chúng tôi phát hiện ra rằng nó chứa một số bảng dữ liệu đa biến yêu cầu chúng tôi áp dụng **PCA (Principal Component Analysis)** hay còn gọi là **Phân tích thành phần chính** cho mỗi bộ dữ liệu. Chúng tôi đã vẽ biểu đồ tỷ lệ phần trăm biến thiên sau khi sử dụng PCA và chọn sử dụng biến đổi ngược. Sau đó, chúng tôi áp dụng **RF (Random Forest)** vào bộ dữ liệu, dẫn đến một sự cải thiện đáng kể về độ chính xác. Chúng tôi sau đó sử dụng phương pháp **Boosting** để tăng thêm độ chính xác dự đoán cả trên bộ dữ liệu không lấy mẫu và trên một bộ dữ liệu có các đặc điểm đáng tin cậy được chọn. Kết quả cho thấy mô hình đang được cải thiện. Cuối cùng, chúng tôi áp dụng **SVM (Support Vector Machine)** và **MLP (Multilayer Perceptron)** vào bộ dữ liệu cuối cùng và đạt được kết quả tốt nhất. Khi so sánh các kết quả thu được sau khi được **feature selection (lựa chọn đặc trưng)** và **boosting (phương pháp tăng cường)**, chúng tôi có thể thấy rằng chúng tôi đã đạt được tiến bộ đáng kể và đạt được **độ** **chính xác** cuối cùng.

A blue and orange pie chart

Description automatically generated A blue and orange pie chart

Description automatically generated

A graph of a number of dots

Description automatically generated A graph of a number of people

Description automatically generated with medium confidence

## Kết quả và Phân tích

A graph of a logistic regression confusion matrix

Description automatically generatedA screenshot of a computer

Description automatically generated A graph of a tree confusion matrix

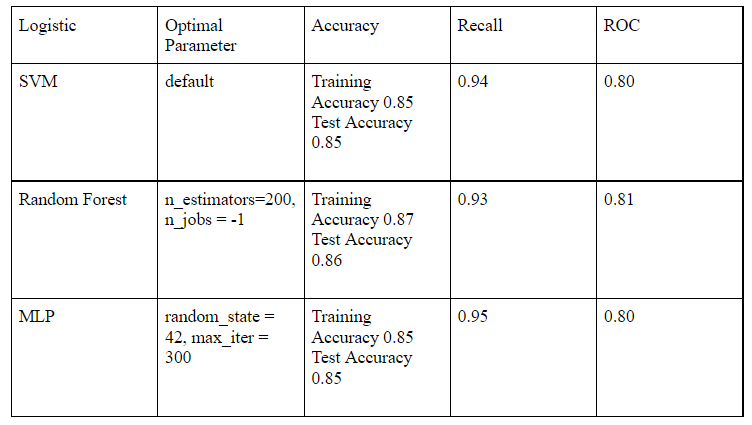
Description automatically generated A graph of a forest confusion matrix

Description automatically generated A graph of a network confusion matrix

Description automatically generated

Như chúng ta đã thấy trong Bảng xếp hạng, Độ chính xác được theo thứ tự như sau: **Decision Tree (Cây Quyết Định)** > **Logistic Regression (Hồi Quy Logistic)** > **Gaussian Naive Bayes.**

Đây là các kết quả trước khi chúng tôi thực hiện lấy mẫu và lấy mẫu giảm.



Nhìn vào kết quả, chúng ta có thể nói rằng Random Forest hoạt động tốt nhất trong tất cả các bộ phân loại với Độ chính xác là 86%. Như chúng ta đã thấy trong Bảng xếp hạng, độ chính xác được theo thứ tự như sau: **RF > MLP > SVM.**

A diagram of a number of ratings

Description automatically generated

Đối với tập dữ liệu huấn luyện, ta thu được sơ đồ ma trận tương quan có độ tương quan khá ổn như sau:

A screen shot of a computer

Description automatically generated

# KẾT LUẬN CHUNG

## Kết luận

Sau khoảng 1 tháng tìm hiểu về công nghệ, dựa trên các tài liệu chuyên môn, các tài liệu trên Internet,… nhóm đã đạt được những thành quả sau:

* + Nắm được kiến thức cơ bản về NLP.
  + Các cách khác nhau để trực quan hóa dữ liệu để hiểu rõ hơn về các đặc điểm trong các mô hình.
  + Tìm hiểu được các thông tin cơ bản của Machine Learning như các mô hình Logistic Regression,….
  + Nâng cao khả năng làm việc nhóm.
  + Khả năng đọc hiểu tài liệu tiếng Anh được cải thiện.
  + Cách sử dụng các nền tảng Kaggle, Google để phục vụ mục đích học tập.

Về kết quả, đồ án đáp ứng được những yêu cầu đặt ra, tuy nhiên bên cạnh đó cũng có những hạn chế như:

* + Chưa tìm hiểu sâu hơn, chưa lấy được nhiều thông tin để khai thác.

## Hướng phát triển:

* + Trong tương lai, nhóm sẽ tìm phương pháp cải thiện hoặc thay đổi các thuật toán dựa trên các nghiên cứu khác.
  + Thực hiện thử nghiệm nhằm chứng minh được tính hiệu quả của mô hình.
  + Tạo bộ dataset mới dựa vào nguồn khác mới hơn.

TÀI LIỆU THAM KHẢO

1. Hareram Kumar (2022) The Research Paper, [Android Malware Prediction using Machine Learning Techniques: A Review](https://www.ijrdet.com/files/Volume11Issue2/IJRDET_0222_03.pdf) (Lần cuối truy cập: 15/05/2024)
2. Neamat Al Sarah (2021) Online Paper, [An Efficient Android Malware Prediction Using Ensemble machine learning algorithms](https://www.ijrdet.com/files/Volume11Issue2/IJRDET_0222_03.pdf) (Lần cuối truy cập: 15/05/2024)
3. [Machine Learning for Android Malware Detection Using Permission and API Calls](https://link.springer.com/chapter/10.1007/978-3-319-12280-9_13) (Lần cuối truy cập: 15/05/2024)
4. [Dynamic Permissions based Android Malware Detection using Machine Learning Techniques](https://www.researchgate.net/publication/312252612_Dynamic_Permissions_based_Android_Malware_Detection_using_Machine_Learning_Techniques) (Lần cuối truy cập: 15/05/2024)
5. [Android Permission Dataset](https://www.kaggle.com/datasets/saurabhshahane/android-permission-dataset) (Lần cuối truy cập: 15/05/2024)
6. [Học máy (ML) để xử lý ngôn ngữ tự nhiên (NLP) - Lexalytics](https://www.lexalytics.com/blog/machine-learning-natural-language-processing/) (Lần cuối truy cập: 15/05/2024)

BẢNG PHÂN CÔNG

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **MSSV** | **Họ và tên** | **Phân công nhiệm vụ** | **Mức độ hoàn thành** | **Ghi chú** |
| **1** | **22521533** | **Nguyễn Công Nam Triều** | **-Phân chia công việc, tổng hợp và hoàn thiện file báo cáo.**  **-Tìm hiểu khái niệm NLP và ML**  **-Nghiên cứu cách sử dụng ML**  **-Soạn slide báo cáo**  **-Tìm kiếm Dataset** | **100%** | **Không** |
| **2** | **22521252** | **Nguyễn Hoàng Trọng Sơn** | **- Nghiên cứu nội dung liên quan đến NLP và ML**  **-Làm báo cáo**  **-Soạn slide báo cáo**  **-Thuyết trình**  **-Tìm kiếm Dataset** | **100%** | **Không** |
| **3** | **22521087** | **Nguyễn Hạc Phong** | **-Nghiên cứu nội dung liên quan đến NLP và ML**  **-Soạn slide báo cáo**  **-Demo code**  **-Tìm kiếm Dataset**  **-Thuyết trình** | **100%** | **Không** |