HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



Nguyễn Thu Trang

PHÁT HIỆN MÃ ĐỘC NHÚNG TRONG FILE PDF

Chuyên ngành: Hệ thống thông tin
Mã số: 8.48.01.04

TÓM TẮT ĐỀ ÁN TỐT NGHIỆP THẠC SĨ

HÀ NỘI - NĂM 2025

Đề án tốt nghiệp được hoàn thành tại:

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Người hướng dẫn khoa học: TS. Phạm Hoàng Duy

(Ghi rõ học hàm, học vị)

Phản biện 1: ……

Phản biện 2: ……

Đề án tốt nghiệp sẽ được bảo vệ trước Hội đồng chấm đề án tốt nghiệp thạc sĩ tại Học viện Công nghệ Bưu chính Viễn thông

Vào lúc: ..... giờ ..... ngày ..... tháng ..... năm .....

Có thể tìm hiểu đề án tốt nghiệp tại:

- Thư viện của Học viện Công nghệ Bưu chính Viễn thông.

I. MỞ ĐẦU

1. Lý do chọn đề tài

Trong bối cảnh chuyển đổi số, việc trao đổi và lưu trữ tài liệu điện tử ngày càng phổ biến, trong đó định dạng PDF đã trở thành một trong những chuẩn tài liệu được sử dụng rộng rãi trong các tổ chức, doanh nghiệp và hệ thống thông tin.

Trên thực tế, tin tặc thường lợi dụng các đặc điểm này để nhúng mã độc vào file PDF nhằm phát tán phần mềm độc hại, đánh cắp dữ liệu hoặc khai thác lỗ hổng của phần mềm đọc PDF. Các phương pháp phát hiện truyền thống dựa trên chữ ký hoặc tập luật tĩnh chỉ hiệu quả với các mẫu mã độc đã biết, nhưng gặp nhiều khó khăn trước các biến thể mới sử dụng kỹ thuật né tránh và làm rối mã.

Xuất phát từ thực tiễn đó, đề tài "Phát hiện mã độc nhúng trong file PDF" được lựa chọn nhằm nghiên cứu và xây dựng mô hình học máy dựa trên phân tích đặc trưng tĩnh của file PDF, góp phần nâng cao khả năng phát hiện sớm mã độc và tăng cường hiệu quả phòng chống tấn công qua tài liệu.

2. Tổng quan về vấn đề nghiên cứu

File PDF có khả năng chứa mã độc dưới dạng mã JavaScript hoặc nội dung nhúng, gây khó khăn cho việc

phát hiện bằng các công cụ truyền thống. Các phương pháp học máy như Random Forest, SVM, XGBoost được đề xuất nhằm phát hiện hiệu quả các mẫu mã độc mới. Tuy nhiên, việc triển khai còn gặp hạn chế do thiếu dữ liệu thực nghiệm, hoặc đặc trưng không đủ phân biệt. Vì vậy, đề tài tập trung nghiên cứu và triển khai giải pháp học máy nhằm nâng cao hiệu quả phát hiện mã độc PDF trong bối cảnh các kỹ thuật tấn công ngày càng tinh vi.

3. Mục đích nghiên cứu

- Tìm hiểu về cấu trúc tệp PDF và các kỹ thuật mã độc thường gặp.
- Xây dựng mô hình học máy (Random Forest) để phát hiện mã độc.
- Tối ưu quy trình tiền xử lý, huấn luyện và đánh giá mô hình.

4. Đối tượng và phạm vi nghiên cứu

Đối tượng: Các file PDF lành tính và file PDF chứa mã độc.

Phạm vi: Nghiên cứu tập trung vào phân tích đặc trưng tĩnh của file PDF, sử dụng dữ liệu từ bộ CIC-Evasive-PDFMal2022.

5. Phương pháp nghiên cứu

- Nghiên cứu lý thuyết về cấu trúc file PDF, các kỹ thuật mã độc phổ biến và thuật toán học máy Random Forest.
- Thu thập và tiền xử lý dữ liệu từ bộ CIC-Evasive-PDFMal2022, trích xuất các đặc trưng tĩnh của file PDF.
- Huấn luyện mô hình Random Forest và đánh giá hiệu suất thông qua các chỉ số Accuracy, Precision, Recall và F1-score.

II. NỘI DUNG

CHƯƠNG 1 - TỔNG QUAN PHÁT HIỆN MÃ ĐỘC NHÚNG TRONG FILE

1.1. Cơ sở lý thuyết về file PDF

Tệp PDF (Portable Document Format) là một trong những định dạng tài liệu được sử dụng phổ biến trong lưu trữ và chia sẻ thông tin nhờ khả năng hiển thị nhất quán trên nhiều nền tảng và hỗ trợ đa dạng nội dung. Bên cạnh các thành phần cơ bản như văn bản và hình ảnh, file PDF còn cho phép nhúng liên kết, biểu mẫu, file đính kèm và mã JavaScript.

Chính đặc điểm này khiến PDF trở thành mục tiêu bị lợi dụng trong các cuộc tấn công mạng. Tin tặc có thể

khai thác cấu trúc phức tạp của file PDF để che giấu mã độc, vượt qua các cơ chế kiểm tra bảo mật truyền thống và thực thi hành vi độc hại khi người dùng mở file.

1.2. Khái niệm và phân loại mã độc

1.2.1. Khái niệm mã độc

Malware là phần mềm độc hại được thiết kế nhằm thực hiện các hành vi trái phép như phá hoại hệ thống, đánh cắp dữ liệu hoặc chiếm quyền kiểm soát thiết bị người dùng.

1.2.2. Các loại mã độc thường gặp trong file PDF

Các loại mã độc điển hình trong file PDF:

Mã độc JavaScript: Lợi dụng khả năng nhúng mã JavaScript để thực hiện hành vi độc hại.

Mã khai thác lỗ hổng (Exploit): Tận dụng điểm yếu phần mềm đọc PDF để thực thi mã độc mà không cần tương tác người dùng.

Trojan: Ngụy trang thành file PDF hợp lệ để lừa người dùng mở và chiếm quyền điều khiển.

Phishing: Chứa liên kết hoặc biểu mẫu giả mạo nhằm đánh cắp thông tin cá nhân.

Tấn công qua file nhúng: Nhúng các file độc hại bên trong PDF.

1.2.3. Kỹ thuật nhúng mã độc vào file PDF

Để tăng khả năng lẩn tránh bị phát hiện, tin tặc sử dụng nhiều kỹ thuật nhúng mã độc khác nhau trong file PDF như nhúng JavaScript độc hại thực thi khi mở file, khai thác lỗ hổng bảo mật của trình đọc PDF, chèn mã độc vào file đính kèm hoặc che giấu mã độc trong hình ảnh, font chữ và các đối tượng PDF.

Ngoài ra, các file PDF lừa đảo chứa liên kết hoặc biểu mẫu giả mạo cũng được sử dụng phổ biến trong các chiến dịch tấn công.

1.3. Phương pháp phát hiện mã độc

1.3.1. Phân tích tĩnh (Static Analysis)

Phân tích tĩnh là phương pháp kiểm tra file PDF mà không cần thực thi, thông qua việc phân tích cấu trúc, metadata và nội dung bên trong file. Phương pháp này cho phép phát hiện các dấu hiệu bất thường như mã JavaScript nhúng, file đính kèm, hành động tự động mở hoặc các đối tượng có entropy cao. Ưu điểm của phân tích tĩnh là an toàn, nhanh và dễ triển khai, tuy nhiên còn hạn chế trong việc phát hiện các mã độc sử dụng kỹ thuật mã hóa hoặc làm rối nâng cao.

1.3.2. Phân tích động (Dynamic Analysis)

Phân tích động được thực hiện bằng cách chạy file PDF trong môi trường giả lập (sandbox) nhằm quan sát các hành vi thực thi như tạo tiến trình hoặc thiết lập kết nối mạng. Phương pháp này có khả năng phát hiện các hành vi độc hại tinh vi, nhưng đòi hỏi nhiều tài nguyên, thời gian và có thể bị né tránh bởi các kỹ thuật chống phân tích động.

1.3.3. Sử dụng học máy (Machine Learning-based Detection)

Phương pháp phát hiện dựa trên học máy sử dụng các thuật toán phân loại để học và nhận diện file PDF độc hại dựa trên **các đặc trưng tĩnh của cấu trúc file**. Cách tiếp cận này giúp phát hiện hiệu quả các biến thể mã độc mới và khắc phục một phần hạn chế của các phương pháp phát hiện truyền thống.

1.4. Thách thức trong phát hiện mã độc

1.4.1. Đa dạng phương thức tấn công

Mã độc PDF được triển khai thông qua nhiều kỹ thuật khác nhau như JavaScript độc hại, làm rối mã, file đính kèm độc hại hoặc khai thác lỗ hổng, gây khó khăn cho việc phát hiện.

1.4.2. Kỹ thuật che giấu và vượt qua kiểm tra bảo mật

Tin tặc thường sử dụng các kỹ thuật như mã hóa, nén dữ liệu, chống phân tích động và thay đổi hành vi nhằm vượt qua các hệ thống phòng vệ như phần mềm diệt virus và IDS/IPS.

1.4.3. Hạn chế của các phương pháp phát hiện truyền thống

Phát hiện dựa trên chữ ký không hiệu quả với mã độc mới. Dựa trên quy tắc dễ bị qua mặt. Phân tích động tốn tài nguyên và có thể bị né tránh.

1.4.4. Thách thức trong ứng dụng học máy để phát hiện mã độc PDF

Cần tập dữ liệu lớn, đa dạng và được gán nhãn chính xác, đồng thời cần chọn đặc trưng phù hợp và tránh hiện tượng quá khớp (overfitting) để mô hình hoạt động hiệu quả trên dữ liệu thực tế.

1.5. Kết luận chương 1

Chương này đã trình bày tổng quan về file PDF, các loại mã độc thường gặp, kỹ thuật nhúng mã độc, các phương pháp phát hiện cũng như những thách thức trong bài toán phát hiện mã độc PDF. Đây là cơ sở lý thuyết quan trọng cho việc đề xuất và triển khai mô hình phát hiện mã độc dựa trên học máy ở các chương tiếp theo.

CHƯƠNG 2 - PHÁT HIỆN MÃ ĐỘC NHÚNG TRONG FILE PDF

2.1. Giải pháp phát hiện mã độc nhúng trong file PDF

2.1.1. Tổng quan về giải pháp sử dụng học máy

Giải pháp đề xuất sử dụng phương pháp học máy có giám sát để tự động phân loại file PDF thành hai nhóm: file lành tính và file chứa mã độc. Khác với các phương pháp truyền thống dựa trên chữ ký hoặc quy tắc cố định, học máy cho phép mô hình học được các đặc trưng tiềm ẩn trong dữ liệu và phát hiện hiệu quả các mẫu mã độc mới hoặc biến thể chưa từng xuất hiện.

Trong khuôn khổ đề tài, giải pháp tập trung vào phân tích tĩnh cấu trúc file PDF, không yêu cầu thực thi file, đảm bảo tính an toàn và phù hợp cho các hệ thống phát hiện sớm trong môi trường thực tế**.**

2.1.2. Mô hình tổng thể phát hiện mã độc nhúng

- **Giai đoạn huấn luyện**: Trích xuất các đặc trưng tĩnh từ tập dữ liệu PDF đã gán nhãn, sau đó sử dụng các đặc trưng này để huấn luyện mô hình học máy.
- **Giai đoạn phát hiện**: Trích xuất đặc trưng tĩnh từ các file PDF mới và sử dụng mô hình đã huấn luyện để phân loại file lành tính hoặc chứa mã độc.

2.2. Trích xuất đặc trưng từ file PDF

Quy trình trích xuất đặc trưng tĩnh từ file PDF gồm 4 bước chính, nhằm chuyển đổi file PDF sang dữ liệu số hóa phục vụ mô hình học máy:

2.2.1. Tập dự liệu

Đề tài sử dụng bộ dữ liệu CIC-Evasive-PDFMal2022 do Canadian Institute for Cybersecurity (CIC), Đại học New Brunswick (Canada) xây dựng. Đây là bộ dữ liệu chuyên dụng phục vụ nghiên cứu phát hiện mã độc nhúng trong file PDF dựa trên phân tích tĩnh.

Bộ dữ liệu gồm 10.025 file PDF, trong đó có 5.557 file chứa mã độc và 4.468 file lành tính. Các mẫu mã độc trong tập dữ liệu bao gồm nhiều kỹ thuật tấn công phổ biến như JavaScript độc hại, file nhúng, hành động tự động và các kỹ thuật né tránh phát hiện.

Dữ liệu được gán nhãn rõ ràng, phù hợp cho các mô hình học máy có giám sát và được sử dụng để trích xuất các đặc trưng tĩnh phục vụ huấn luyện và đánh giá mô hình.

2.2.2. Phân tích cấu trúc của PDF

Xác định các đặc trưng cơ bản như số trang, số đối tượng, kích thước file bằng công cụ phân tích tĩnh (PyPDF2, pdfminer...).

2.2.2. Trích xuất đặc trưng hành vi tiềm ẩn

Các đặc trưng mang tính bảo mật được trích xuất nhằm phát hiện các dấu hiệu tiềm ẩn của mã độc, bao gồm sự tồn tại của mã JavaScript, hành động tự động khi mở file (OpenAction), file đính kèm, các đối tượng bất thường và entropy của các vùng dữ liệu stream.

2.2.3. Chuẩn hóa và mã hóa đặc trưng

Các đặc trưng dạng nhị phân được mã hóa về giá trị 0/1, trong khi các đặc trưng số liên tục được chuẩn hóa nhằm giảm ảnh hưởng của sự chênh lệch thang đo và giúp mô hình học máy hoạt động hiệu quả hơn.

2.2.4. Tổng hợp thành tập dữ liệu đầu vào

Sau khi trích xuất và tiền xử lý, các đặc trưng được tổng hợp thành tập dữ liệu đầu vào, trong đó mỗi dòng tương ứng với một file PDF và mỗi cột là một đặc trưng. Tập dữ liệu này được sử dụng để huấn luyện và đánh giá mô hình học máy.

Bảng 2.1. Các đặc trưng trích xuất từ file PDF

Nhóm	Đặc trưng	Mô tả
Thông tin chung	FileName	Tên file PDF (định danh, không dùng để huấn luyện).
	PdfSize	Kích thước file PDF (byte).

	MetadataSize	Kích thước phần siêu dữ liệu.
	Pages	Tổng số trang trong file.
	TitleCharacters	Số ký tự trong trường tiêu đề.
	isEncrypted	Trạng thái mã hóa của file (0: Không, 1: Có).
Cấu trúc file	XrefLength	Độ dài bảng tham chiếu chéo.
	Header	Kiểm tra định dạng Header của PDF.
	Obj	Số lượng đối tượng (object) được định nghĩa.
	EndObj	Số lượng từ khóa kết thúc đối tượng *endobj*.
	Stream	Số lượng luồng dữ liệu *stream*.
	Endstream	Số lượng từ khóa kết thúc luồng *endstream*.
	Xref	Số lượng bảng tham chiếu chéo.
	Trailer	Số lượng từ khóa *trailer*.
	StartXref	Số lượng từ khóa *startxref*.

	PageNo	Số lượng từ khóa đánh dấu trang.
	Encrypt	Số lượng từ khóa *Encrypt*.
	ObjStm	Số lượng Object Stream.
Nội dung đáng ngờ	JS	Số lượng từ khóa *JS*.
	Javascript	Số lượng từ khóa *Javascript*.
	AA	Số lượng từ khóa *AA* (Additional Action).
	OpenAction	Số lượng từ khóa *OpenAction*.
	AcroForm	Số lượng từ khóa *AcroForm*.
	JBIG2Decode	Số lượng bộ lọc *JBIG2Decode*.
	RichMedia	Số lượng đối tượng đa phương tiện.
	Launch	Số lượng từ khóa *Launch*.
	EmbeddedFile	Số lượng từ khóa *EmbeddedFile*.
	EmbeddedFiles	Số lượng file được nhúng.
	XFA	Số lượng biểu mẫu XFA.
	Colors	Số lượng màu sắc sử dụng.

Dữ liệu thành phần	Images	Số lượng hình ảnh trong file.
	Text	File có chứa văn bản hay không (0/1).
Nhãn	Class	Nhãn phân loại (0: Benign, 1: Malicious).

2.3. Mô tả giải pháp

2.3.1. Lựa chọn phương pháp phát hiện mã độc

Đề tài lựa chọn phương pháp học máy có giám sát để phát hiện mã độc nhúng trong file PDF dựa trên phân tích tĩnh cấu trúc file. So với các phương pháp truyền thống dựa trên chữ ký hoặc quy tắc, cách tiếp cận này cho phép phát hiện hiệu quả các mẫu mã độc mới, kể cả khi mã nguồn đã bị làm rối, đồng thời không yêu cầu thực thi file như phân tích động.

Trong số các thuật toán học máy phổ biến, Random Forest được lựa chọn làm mô hình chính do có khả năng xử lý tốt dữ liệu nhiều chiều, kháng nhiễu, giảm hiện tượng quá khớp và phù hợp với dữ liệu dạng bảng chứa các đặc trưng cấu trúc của file PDF. Ngoài ra, Random Forest còn cung cấp cơ chế đánh giá mức độ quan trọng của đặc trưng, giúp tăng khả năng giải thích và tính minh bạch của hệ thống phát hiện mã độc.

Bảng so sánh 2.2 cho thấy so với các mô hình khác như XGBoost, Naive Bayes và SVM, Random Forest thể hiện sự cân bằng tốt giữa độ chính xác, khả năng tổng quát hóa và hiệu quả thực tế, phù hợp với mục tiêu của đề tài.

Bảng 2.2. So sánh ưu nhược điểm của các thuật toán

Thuật toán	Đặc điểm chính	Hạn chế
Decision Tree	Dễ diễn giải, cấu trúc đơn giản, huấn luyện nhanh	Dễ bị quá khớp, hiệu suất không ổn định với dữ liệu nhiều nhiễu
SVM	Khả năng phân tách tốt trong không gian nhiều chiều	Nhạy cảm với tham số, chi phí huấn luyện cao, khó mở rộng với dữ liệu lớn
KNN	Cấu trúc đơn giản, không cần huấn luyện mô hình	Hiệu suất suy giảm với dữ liệu nhiều chiều, chi phí suy luận lớn
Random Forest	Kháng nhiễu tốt, giảm overfitting, ổn định, hỗ trợ đánh giá mức độ quan trọng của đặc trưng	Thời gian huấn luyện tăng khi số lượng cây lớn

2.4.2. Quy trình phát hiện mã độc

Quy trình phát hiện mã độc nhúng trong file PDF bằng thuật toán Random Forest bao gồm các bước:



Trích xuất đặc trưng: Thu thập các đặc trưng tĩnh từ file PDF như metadata, kích thước, cấu trúc và các dấu hiệu nội dung tiềm ẩn mã độc.

Tiền xử lý dữ liệu: Làm sạch dữ liệu, mã hóa nhãn, chuẩn hóa đặc trưng và cân bằng dữ liệu.

Huấn luyện mô hình: Sử dụng thuật toán Random Forest trên tập đặc trưng đã được tiền xử lý.

Đánh giá mô hình: Đánh giá hiệu quả mô hình thông qua các chỉ số Accuracy, Precision, Recall và F1-score.

Áp dụng thực tế: Mô hình dự đoán file PDF thuộc lớp mã độc hoặc an toàn, từ đó hệ thống đưa ra cảnh báo hoặc xử lý phù hợp.

2.4. Kết luận chương 2

Chương này đã trình bày tổng quan giải pháp phát hiện mã độc nhúng trong file PDF dựa trên phương pháp học máy có giám sát, với mô hình Random Forest làm trung

tâm. Quy trình phát hiện mã độc được xây dựng gồm các bước trích xuất đặc trưng, tiền xử lý dữ liệu, huấn luyện và đánh giá mô hình, làm cơ sở cho chương tiếp theo về thực nghiệm và đánh giá kết quả.

CHƯƠNG 3 - THỬ NGHIỆM VÀ ĐÁNH GIÁ

3.1. Thiết lập môi trường thực nghiệm

3.1.1. Công cụ và framework hỗ trợ

Quá trình thử nghiệm mô hình phát hiện mã độc trong file PDF được thực hiện trên môi trường sau:

- **Hệ điều hành:** Windows 10
- **Phần cứng:** Bộ xử lý Intel Core i7, RAM 16GB
- **Ngôn ngữ lập trình:** Python 3.10
- **Thư viện xử lý file PDF:** PyPDF2, PDFMiner
- **Thư viện xử lý dữ liệu và học máy:** pandas, numpy, scikit-learn.
- **Công cụ lưu trữ mô hình:** pickle
- **Môi trường phát triển:** Visual Studio Code

3.2. Quy trình huấn luyện

3.2.1. Chuẩn bị dữ liệu huấn luyện

Bộ dữ liệu CIC-Evasive-PDFMal2022 do CIC – Đại học New Brunswick phát triển được sử dụng trong nghiên cứu. Đây là bộ dữ liệu chuyên dụng cho bài toán phát hiện mã độc PDF, tập trung vào các mẫu sử dụng kỹ thuật lẩn tránh và làm rối.

Dữ liệu được xây dựng theo phương pháp phân tích tĩnh, trích xuất trực tiếp các đặc trưng cấu trúc từ file PDF mà không cần thực thi. Tập dữ liệu ban đầu gồm 10,023

mẫu được tải từ kho dữ liệu. Sau bước tiền xử lý và loại bỏ 220 mẫu trùng lặp, dữ liệu sạch còn lại 9,803 mẫu, trong đó gồm 5,369 mẫu mã độc (Malicious) và 4,434 mẫu lành tính (Benign).

Tỷ lệ phân bố giữa hai lớp tương đối cân bằng, giúp hạn chế hiện tượng mất cân bằng dữ liệu và đảm bảo kết quả đánh giá mô hình khách quan

3.2.2. Tiền xử lý và chuẩn hóa dữ liệu

Dữ liệu gồm 33 đặc trưng tĩnh và được tiền xử lý qua các bước sau:

- Làm sạch dữ liệu, xử lý giá trị thiếu và loại bỏ các thuộc tính không có ý nghĩa phân loại.
- Mã hóa nhãn với Benign = 0 và Malicious = 1.
- Chuẩn hóa các đặc trưng số bằng StandardScaler nhằm giảm chênh lệch thang đo.

Sau đó, dữ liệu được chia thành ba tập theo phương pháp lấy mẫu phân tầng: huấn luyện (70%), kiểm tra (15%) và đánh giá (15%).

3.2.3. Huấn luyện mô hình Random Forest

Mô hình Random Forest được huấn luyện trên tập huấn luyện và tối ưu siêu tham số bằng GridSearchCV với các tham số chính gồm: n_estimators, max_depth, min_samples_split, min_samples_leaf.

Mô hình tối ưu được lựa chọn dựa trên **F1-score**, với bộ tham số tốt nhất như sau:

```
Best Parameters: {
   "n_estimators": 200,
   "max_depth": 15,
   "min_samples_split": 2,
   "min_samples_leaf": 1
}
```

Toàn bộ dữ liệu đã xử lý, mô hình tối ưu và bộ chuẩn hóa được lưu lại để phục vụ đánh giá và triển khai.

3.3. Đánh giá mô hình

3.3.1. Đánh giá hiệu suất mô hình

Mô hình Random Forest được kiểm thử trên tập kiểm tra và tập đánh giá, với kết quả:

Kết quả trên tập kiểm tra (Validation Set)

Tập kiểm tra gồm 1.470 mẫu, được sử dụng trong quá trình lựa chọn tham số và tinh chỉnh mô hình. Kết quả thực nghiệm cho thấy mô hình đạt độ chính xác 99.05%, phản ánh hiệu suất phân loại cao và ổn định. Kết quả chi tiết như sau:

Bảng 3.4. Kết quả thử nghiệm trên tập kiểm tra

Lớp dữ liệu	Precision	Recall	F1-Score
0 - Lành tính	0.9924	0.9865	0.9894

1 - Mã độc	0.9889	0.9938	0.9913
Trung bình	0.9905	0.9905	0.9905

Kết quả cho thấy mô hình duy trì sự cân bằng tốt giữa Precision và Recall đối với cả hai lớp. Đặc biệt, chỉ số Recall của lớp mã độc đạt 99.38%**,** cho thấy mô hình có khả năng phát hiện hầu hết các mẫu mã độc trong tập kiểm tra, đồng thời vẫn kiểm soát tốt tỷ lệ cảnh báo sai.

Kết quả trên tập đánh giá (Test Set)

Tập đánh giá gồm 1.471 mẫu hoàn toàn mới, không tham gia vào bất kỳ giai đoạn huấn luyện hay tối ưu tham số nào, nhằm phản ánh khả năng tổng quát hóa của mô hình trong điều kiện vận hành thực tế. Kết quả đánh giá như sau:

Bảng 3.5. Kết quả thử nghiệm trên tập đánh giá

Lớp dữ liệu	Precision	Recall	F1-Score
0 - Lành tính	0.9940	0.9925	0.9932
1 - Mã độc	0.9938	0.9950	0.9944
Trung bình	0.9939	0.9939	0.9939

Đối với lớp mã độc, chỉ số Recall đạt 99.50%, phản ánh khả năng phát hiện hiệu quả các mẫu tấn công và giảm thiểu nguy cơ bỏ sót. Đồng thời, Precision đạt 99.38%, cho thấy tỷ lệ cảnh báo sai đối với các file PDF lành tính được duy trì ở mức thấp. Tổng hợp các chỉ số cho thấy mô hình

Random Forest đạt hiệu suất cao, cân bằng tốt giữa khả năng phát hiện mã độc và hạn chế báo động giả, phù hợp để triển khai trong thực tế.

3.3.2. Trực quan hóa kết quả

Hiệu suất mô hình được trực quan hóa thông qua ma trận nhầm lẫn, đường cong ROC/AUC và biểu đồ mức độ quan trọng của đặc trưng.

Ma trận nhầm lẫn cho thấy số lượng mẫu bị phân loại sai rất thấp, với phần lớn dự đoán tập trung trên đường chéo chính. Đường cong ROC tiệm cận góc trên bên trái với giá trị AUC xấp xỉ 0.99, thể hiện khả năng phân biệt gần như hoàn hảo giữa file mã độc và file lành tính.

Biểu đồ Feature Importance cho thấy các đặc trưng như kích thước file PDF và sự hiện diện của JavaScript đóng vai trò quan trọng nhất, phù hợp với đặc điểm tấn công phổ biến của mã độc PDF hiện nay.

3.3.3. So sánh kết quả giữa các mô hình

Bảng 3.6. So sánh hiệu suất giữa các mô hình

Mô hình	Accuracy	Precision	Recall	F1-score
Random Forest	0.9912	0.9913	0.9926	0.9919
Decision Tree	0.9864	0.9925	0.9826	0.9875

KNN	0.9816	0.9839	0.9826	0.9832
SVM	0.9674	0.9859	0.9541	0.9697

Kết quả cho thấy Random Forest đạt hiệu suất cao nhất với Accuracy 99.39% và Recall 99.50%.

Decision Tree và KNN cho kết quả khá tốt nhưng kém ổn định hơn, trong khi SVM có Recall thấp hơn, tiềm ẩn nguy cơ bỏ sót mã độc. Điều này khẳng định Random Forest là lựa chọn phù hợp nhất cho bài toán phát hiện mã độc PDF.

3.3.4. Đánh giá tác động của tham số huấn luyện

Phân tích độ nhạy cho thấy độ sâu cây (max_depth) có ảnh hưởng lớn nhất đến hiệu suất mô hình. Khi độ sâu tăng từ 2 lên 15, F1-score cải thiện rõ rệt và đạt trạng thái ổn định.

Việc tăng số lượng cây (n_estimators) giúp mô hình ổn định hơn, dù mức cải thiện không quá lớn. Cấu hình tối ưu được lựa chọn là max_depth = 15 và n_estimators = 200, đảm bảo hiệu suất cao và ổn định.

3.4. Kết luận chương 3

Chương 3 trình bày quá trình huấn luyện và đánh giá mô hình Random Forest cho bài toán phát hiện mã độc PDF dựa trên phân tích tĩnh. Mô hình được tối ưu bằng GridSearchCV và đạt Accuracy 99.39%, Recall 99.50%. Kết quả cho thấy Random

Forest cho hiệu suất và độ ổn định cao hơn các mô hình so sánh. Các đặc trưng cấu trúc như kích thước file và JavaScript đóng vai trò quan trọng trong việc phát hiện mã độc.

III. KẾT LUẬN

1. Các kết quả đạt được của đề án

Đề án đã xây dựng thành công giải pháp phát hiện mã độc nhúng trong file PDF dựa trên phương pháp phân tích tĩnh kết hợp với thuật toán học máy Random Forest. Mô hình được huấn luyện và đánh giá trên bộ dữ liệu CIC-Evasive-PDFMal2022, đạt độ chính xác 99.39% và Recall 99.50% trên tập đánh giá độc lập. Kết quả cho thấy Random Forest có hiệu năng và độ ổn định vượt trội so với các thuật toán như Decision Tree, SVM và KNN.

Bên cạnh đó, đề án đã hoàn thiện quy trình trích xuất đặc trưng, tiền xử lý, huấn luyện và lưu trữ mô hình, cho thấy khả năng triển khai trong các hệ thống phát hiện mã độc thực tế.

2. Nhận xét, đề xuất, khuyến nghị

Kết quả nghiên cứu khẳng định hiệu quả của phương pháp phân tích tĩnh kết hợp học máy trong bài toán phát hiện mã độc PDF, với ưu điểm là tốc độ xử lý nhanh, không cần thực thi file và dễ tích hợp vào hệ thống thực tế. Tuy nhiên, do dựa trên đặc trưng tĩnh, mô hình có thể gặp hạn chế khi đối mặt với các mẫu mã độc mới sử dụng kỹ thuật làm rối hoặc che giấu nâng cao, do đó cần được cập nhật dữ liệu và đặc trưng thường xuyên.

3. Hướng phát triển

Trong tương lai, đề án có thể được mở rộng theo các hướng:

- Mở rộng và cập nhật tập dữ liệu huấn luyện với các mẫu PDF mới từ môi trường thực tế.
- Kết hợp phương pháp phân tích tĩnh với phân tích động theo hướng lai (hybrid analysis) để nâng cao khả năng phát hiện mã độc zero-day.
- Nghiên cứu áp dụng các mô hình học sâu nhằm tự động trích xuất đặc trưng phức tạp từ cấu trúc file PDF.
- Xây dựng ứng dụng hoặc API cho phép tích hợp mô hình vào các hệ thống email gateway và nền tảng quét tệp tin.