**HỌC VIỆN KỸ THUẬT MẬT MÃ**

KHOA CÔNG NGHỆ THÔNG TIN



**THỰC TẬP CƠ SỞ CHUYÊN NGÀNH**

XÂY DỰNG ỨNG DỤNG PHÁT HIỆN

PHẦM MỀM ĐỘC HẠI ANDROID

|  |  |  |
| --- | --- | --- |
| Giảng viên hướng dẫn: | **Ths. Thái Thị Thanh Vân** | |
| Sinh viên thực hiện: | *Trần Gia Lương* | *CT030433* |
| *Trương Quốc Quân* | *CT030440* |
| *Vũ Thị Thanh Vân* | *CT030358* |

**Hà Nội, 2021-2022**

**LỜI CẢM ƠN**

Trong thời gian làm đề tài thực tập cơ sở chuyên ngành, chúng em đã nhận được nhiều sự giúp đỡ, đóng góp nhiệt tình của thầy cô, gia đình và bạn bè. Đầu tiên, chúng em xin được gửi lời cảm ơn chân thành và sâu sắc tới cô **Ths. Thái Thị Thanh Vân**, người đã tận tình hướng dẫn, chỉ bảo chúng em trong suốt quá trình thực hiện đề tài thực tập cơ sở chuyên ngành, giúp chúng em có thêm kiến thức chuyên môn cũng như học hỏi được tinh thần trách nhiệm, thái độ làm việc nghiêm túc, hiệu quả từ cô.

Chúng em cũng xin chân thành cảm ơn các thầy cô giáo trong Học viện Kỹ thuật Mật Mã nói chung cùng các thầy cô trong khoa Công nghệ thông tin nói riêng đã tận tình giảng dạy, truyền đạt cho chúng em những kiến thức và kinh nghiệm quý báu trong suốt thời gian học tập, giúp chúng em có được cơ sở lý thuyết vững vàng cũng như sự quan tâm và tạo mọi điều kiện thuận lợi cho chúng em trong quá trình thực hiện đề tài thực tập cơ sở chuyên ngành. Cuối cùng, chúng em xin chân thành cảm ơn gia đình và bạn bè, đã luôn tạo điều kiện, quan tâm, giúp đỡ, động viên chúng em trong suốt quá trình học tập và hoàn thành đề tài thực tập cơ sở chuyên ngành.

Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế nên không thể tránh được những thiếu sót. Chúng em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của các thầy cô để chúng em có điều kiện bổ sung, nâng cao ý thức của mình, phục vụ tốt hơn công tác thực tế sau này.

*Chúng em xin chân thành cảm ơn!*

**MỤC LỤC**

CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI 1

1.1. Giới thiệu đề tài 1

1.2. Mục tiêu của đề tài 2

CHƯƠNG 2. MACHINE LEARNING VÀ MẠNG THÔNG TIN KHÔNG ĐỒNG NHẤT 3

CHƯƠNG 3. XÂY DỰNG ỨNG DỤNG PHÁT HIỆN PHẦM MỀM ĐỘC HẠI ANDROID 4

3.1. Quy trình triển khai 4

**DANH MỤC BẢNG**

**DANH MỤC HÌNH ẢNH**

[Hình 3.1: Xây dựng các ma trận cơ sở. 4](#_Toc91665565)

# TỔNG QUAN VỀ ĐỀ TÀI

## Giới thiệu đề tài

Theo các số liệu thống kê mới nhất, Android hiện nay vẫn là hệ điều hành chiếm lĩnh phần lớn thị phần di động trên toàn cầu ( 87,7% - theo IDC 2018) và tỷ lệ này vẫn giữ nguyên cho đến năm 2021. Vì vậy, các thiết bị sử dụng hệ điều hành Android đã trở thành đối tượng bị tấn công của các ứng dụng độc hại android, gây ra mối đe dọa nghiêm trọng đối với việc rò rỉ dữ liệu cá nhân như vị trí người dùng, thông tin liên lạc, tài khoản, ảnh, vv... Thực trạng trên khiến việc xây dựng ứng dụng phát hiện phần mềm độc hại Android ngày càng trở nên cấp thiết.

Trước đây, hướng tiếp cận để giải quyết bài toán phát hiện phần mềm độc hại chủ yếu dựa vào kỹ thuật đối sánh mẫu truyền thống. Tuy nhiên, với sự phát triển nhanh chóng của các phương pháp học máy và trí tuệ nhân tạo, hướng nghiên cứu về việc phát triển các hệ thống tự động phát hiện phần mềm độc hại bằng cách sử dụng các kỹ thuật khai phá dữ liệu và học máy đang thu hút sự quan tâm của các nhà nghiên cứu trong lĩnh vực an ninh mạng. Vấn đề lớn nhất của bài toán phát hiện phần mềm độc hại sử dụng học máy đó là giải pháp phân tích, biểu diễn dữ liệu và trích chọn đặc trưng. Có hai cách tiếp cận để giải quyết vấn đề này, một là sử dụng kỹ thuật phân tích hành vi và hai là kỹ thuật phân tích chữ ký.

Kỹ thuật phân tích chữ ký dựa trên các dấu hiệu số (Digital Footprint) để đoán nhận mã độc. Tất cả các chương trình, dù là lành hay độc, đều có những dấu hiệu số đặc trưng của riêng mình. Các phần mềm diệt virus thường kết hợp với sử dụng một cơ sở dữ liệu lưu trữ các dấu hiệu số của các mã độc. Các phần mềm ấy sẽ đối sánh dấu hiệu số của file đang được quét với các dữ liệu trong cơ sở dữ liệu, Nếu tìm thấy dấu hiệu đó trong cơ sở dữ liệu, file đó sẽ được coi là độc. Khi một loại mã độc mới được phát hiện, dấu hiệu số của nó sẽ được thêm vào cơ sở dữ liệu của công ty viết ra phần mềm virus đó, và sẽ được chia sẻ cho cơ sở dữ liệu từ phía người dùng.

Hướng nghiên cứu phân tích hành vi dựa vào việc phân tích và đánh giá mã nguồn của ứng dụng được nghi ngờ. Trong bài toán đoán nhận mã độc Android, hướng nghiên cứu này dựa vào việc phân tích lời gọi API, quyền truy cập, lời gọi hệ thống. Các thông tin trên được biểu diễn bằng các mô hình mạng thông tin đồng nhất, nghĩa là nếu biểu diễn các dữ liệu này thành đồ thị thì các đỉnh trong đồ thị có chung một kiểu, và các cạnh của đồ thị đều biểu diễn một loại quan hệ. Trên mô hình mạng đã xây dựng, các nhóm sẽ đề xuất các thuật toán trích chọn đặc trưng phù hợp với mô hình học máy sử dụng. Hiệu quả của các hệ thống tự động phát hiện mã độc sử dụng mạng thông tin đồng nhất để biểu diễn dữ liệu và trích chọn đặc trưng cho kết quả dự đoán chính xác từ 85-92%. Nguyên nhân là do dữ liệu trong các hệ thống thực tế - đặc biệt là dữ liệu về mã độc thường phong phú, đa dạng và chứa đựng nhiều ý nghĩa tiềm ẩn khác. Nếu sử dụng mạng đồng nhất để biểu diễn thì có thể sẽ làm mất đi các thông tin ngữ nghĩa quan trọng trong mạng và việc trích chọn đặc trưng trên mô hình mạng đồng nhất đã không tận dụng tối đa các thông tin thu được để phát hiện mã độc, đặc biệt đối với các mã độc thế hệ mới. Vì vậy hướng nghiên cứu ứng dụng các kỹ thuật phân tích dữ liệu phức tạp để giải quyết thách thức trên đang được rất nhiều nhà nghiên cứu quan tâm. Một trong các công cụ thường được sử dụng để phân tích, biểu diễn dữ liệu phức tạp hiện nay là sử dụng mô hình mạng thông tin không đồng nhất (Heterogeneous Information Network- HIN).

Mạng thông tin không đồng nhất gồm nhiều loại đối tượng khác nhau và các liên kết (cạnh) trong mạng thông tin không đồng nhất có thể mang nhiều ý nghĩa khác nhau. Bởi vậy, việc khai phá các mạng thông tin không đồng nhất sẽ cho chúng ta biết thêm các tri thức còn tiềm ẩn trong các cấu trúc mạng, từ đó có thể ứng dụng trong nhiều lĩnh vực khác nhau. Cho đến hiện này, HIN đã được ứng dụng trong nhiều lĩnh vực khác nhau, như khai phá dữ liệu văn bản, khai phá dữ liệu sinh học, và đặc biệt ứng dụng trong bài toán phát hiện mã độc. Điển hình như nhóm nghiên cứu của [*Yanfang Ye*](https://www.researchgate.net/scientific-contributions/69831786_Yanfang_Ye) *và đồng nghiệp* đã sử mạng thông tin không đồng nhất để biểu diễn dữ liệu mã độc Android, mạng bao gồm 5 loại đỉnh (ứng dụng, API, IMEI, nhà sản xuất, chữ ký) và 5 loại cạnh (ứng dụng – API, ứng dụng-IMEI, IMEI-Nhà sản xuất, ứng dụng-nhà sản xuất, ứng dụng – chữ ký).

## Mục tiêu của đề tài

# MACHINE LEARNING VÀ MẠNG THÔNG TIN KHÔNG ĐỒNG NHẤT

# XÂY DỰNG ỨNG DỤNG PHÁT HIỆN PHẦM MỀM ĐỘC HẠI ANDROID

## Xây dựng mô hình học máy

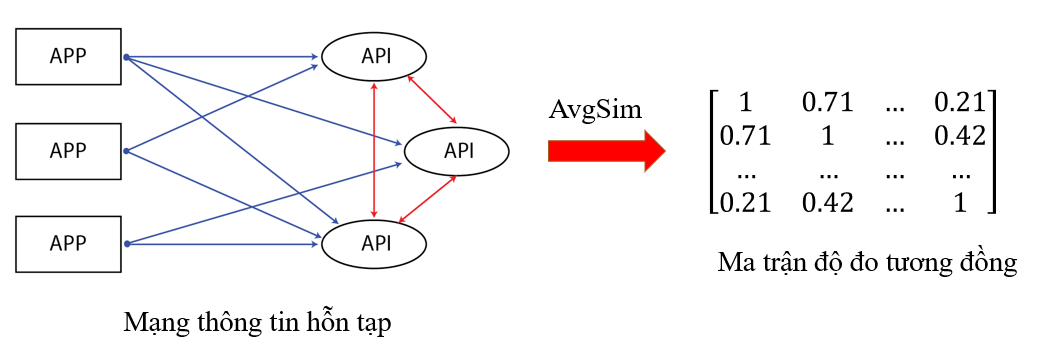
Do việc cơ sở dữ liệu của các họ mã độc còn hạn chế, nhóm sử dụng một tập dữ liệu chứa 589 ứng dụng android được chia thành sáu họ là: Benign, Adware, Botnet, Ransomware, Scareware và SMSmalware để xây dựng nên một mô hình học máy. Sau khi xây dựng lên được mô hình học máy này, nhóm sử dụng mô hình để phân chia 12000 ứng dụng android thành các họ riêng biệt để tạo nên một bộ dữ liệu học máy lớn hơn. Khi đã có bộ dữ liệu bao gồm 12000 ứng dụng đã được phân thia thành các họ, nhóm tiến hành xây dựng lại mô hình học máy dựa trên bộ dữ liệu 12000 ứng dụng này.

Vì vậy quá trình xây dựng mô hình học máy sẽ được chia thành hai giai đoạn bao gồm giai đoạn một: xây dựng mô hình semi – supervised learning để tạo cơ sở xây dựng một tập dữ liệu lớn hơn. Quy trình triển khai xây dựng mô hình được chia thành ba bước như sau:

* **Bước 1:** Thu thập dữ liệu và xây dựng các ma trận cần thiết. Nhóm đã sử dụng tập dữ liệu CICDataset và cụ thể tập dữ liệu có tên là CIC-AndMal2017, kết hợp cùng với một bộ android botnet để tạo nên bộ dữ liệu bao gồm 589 ứng dụng android, các ứng dụng android được chia thành sáu họ là: Benign, Adware, Botnet, Ransomware, Scareware và SMSmalware. Tiếp theo là sử dụng phầm mềm dịch ngược APK Tool để dịch ngược các ứng dụng android, thu được dạng mã smali và tiến hành trích xuất các API của từng ứng dụng. Từ dữ liệu đã được trích xuất, nhóm xây dựng nên các ma trận cần thiết bao gồm ma trận app – api, ma trận invoke, ma trận method và ma trận package

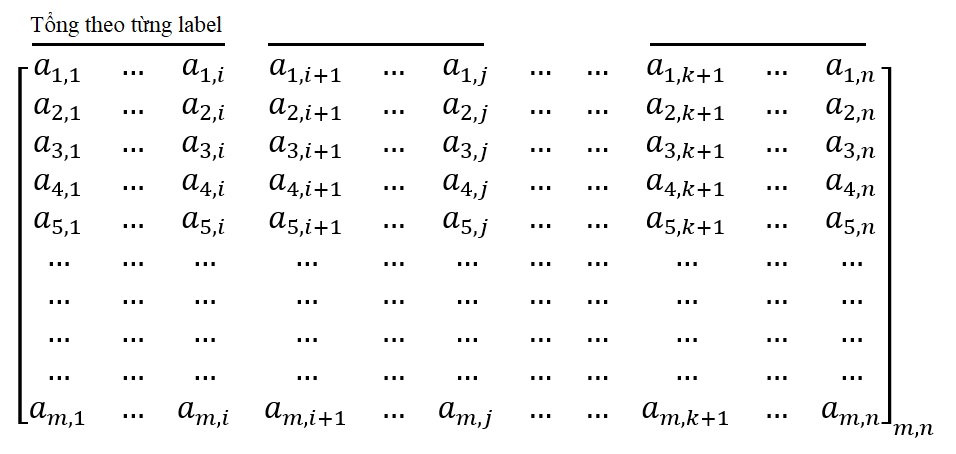
|  |  |
| --- | --- |
| Hình .: Xây dựng các ma trận cơ sở. |  |

* **Bước 2:** Xây dựng mạng thông tin không đồng nhất và trích xuất vector đặc trưng. Từ bốn ma trận cơ bản, nhóm xây dựng mười sáu ma trận độ đo tương đồng và trích xuất vector đặc trưng dựa vào các ma trận độ đo trương đồng này.



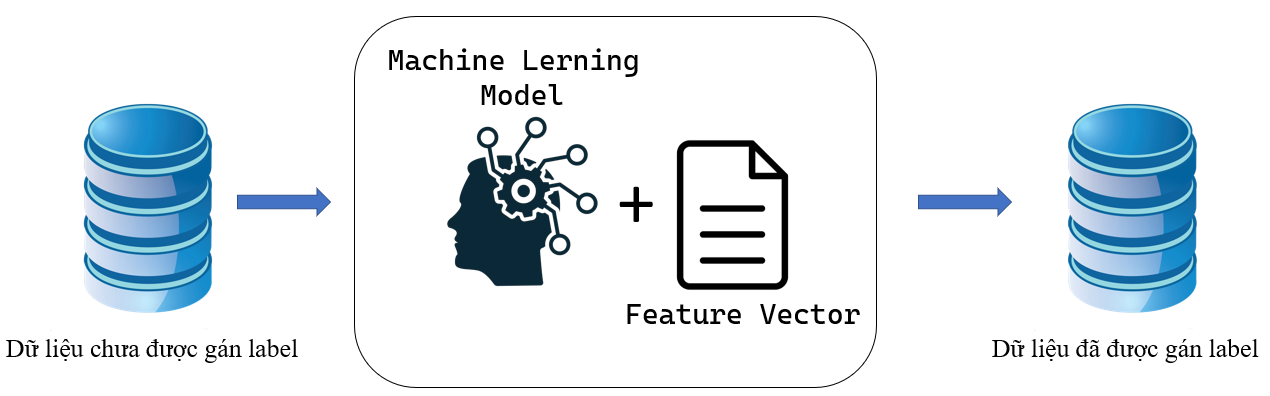
Hình .: Xây dựng ma trận độ đo tương đồng

* **Bước 3:** Mỗi ma trận độ đo tương đồng sẽ cho ra sáu đặc trưng bằng cách gộp các hàng của ma trận đô đo tương đồng lại theo số lượng từng họ ứng dụng, cùng với mười sáu ma trận độ đo tương đồng kết hợp với sáu họ android ở tập dữ liệu sẽ tạo nên một ma trận vector đặc trưng sẽ có số chiều là chín mươi sáu. Ma trận vector đặc trưng này sẽ được sử dụng để phân loại 12000 ứng dụng android vào từng họ ứng dụng riêng biệt.



Hình .: Mô tả cách tổng hợp ra các đặc trưng từ ma trận độ đo tương đồng

* **Bước 4:** Phân loại bộ 12000 ứng dụng andoid. Nhóm sử dụng mô hình vừa xây dựng được để phân loại 12000 ứng dụng android này thành 6 họ ứng dụng. với từng app android, nhóm sẽ đi trích xuất api để nhúng vào trong ma trận app – api, sau đó đưa qua AvgSim để thu được vector đặc trưng ứng với tập dữ liệu 589 ứng dụng android. Sau đó sẽ được mô hình học máy gán nhãn thành các họ ứng dụng riêng biệt. Kết thúc giai đoạn thứ nhất, ta thu được tập dữ liệu gồm 12000 ứng dụng đã được gán nhãn từng họ riêng biệt.



Hình .: Gán label cho dữ liệu

Giai đoạn thứ hai giống hệt như giai đoạn một, chỉ khác ở chỗ là việc sử dụng tập dữ liệu lớn hơn, và vì khi dữ liệu ở trên mạng thông tin không đồng nhất càng nhiều, độ chích xác càng cao nên tỉ lệ chính xác tăng lên đáng kể. Cộng thêm việc nhóm đã có các ma trận app – api, invoke, method, package của tập dữ liệu 12000 ứng dụng này rồi nên không cần phải dịch ngược và trích xuất API nữa. Giai đoạn thứ hai bao gồm các bước như sau:

* Bước 1: Xây dựng ma trận độ đo tương đồng và vector đặc trưng. Từ bốn ma trận app – api, invoke, method, package. Nhóm sử dụng thuật toán để xây dựng lên mười sáu ma trận độ đo tương đồng, mỗi ma trận độ đo tương đồng sẽ cho ra sáu đặc trưng và kết quả thu được ma trận đặc trưng có số chiều là chín mươi sáu.
* Bước 2: Xây dựng hệ thống doán nhận có đầu vào là một app không biết trước, hệ thống sẽ dịch ngược ứng dụng và trích xuất lấy vector đặc trưng, sau đó đoán nhận và trả về kết quả.