|  |  |  |
| --- | --- | --- |
|  | BAN CƠ YẾU CHÍNH PHỦ  “HỌC VIỆN KỸ THUẬT MẬT MÃ” | Mẫu 2 |

BÁO CÁO CHUYÊN ĐỀ SỐ 4.1

“Xây dựng mô hình bài toán phát hiện mã độc tổng thể dựa trên học sâu”

NHIỆM VỤ: “Nghiên cứu và ứng dụng nền tảng học sâu để xây dựng hệ thống phát hiện mã độc trực tuyến”.

Mã số: 06/2022/CB.

Cơ quan chủ trì: Học viện Kỹ thuật Mật mã

Chủ nhiệm: ThS. Lê Đức Thuận

Hà Nội - 2023

|  |  |  |
| --- | --- | --- |
|  | BAN CƠ YẾU CHÍNH PHỦ  “HỌC VIỆN KỸ THUẬT MẬT MÃ” |  |

BÁO CÁO CHUYÊN ĐỀ SỐ 4.1

“Xây dựng mô hình bài toán phát hiện mã độc tổng thể dựa trên học sâu”

NHIỆM VỤ: “Nghiên cứu và ứng dụng nền tảng học sâu để xây dựng hệ thống phát hiện mã độc trực tuyến”.

Mã số: 06/2022/CB.

Cơ quan chủ trì: Học viện Kỹ thuật Mật mã

Chủ nhiệm: ThS. Lê Đức Thuận

|  |  |
| --- | --- |
| **Người thực hiện chuyên đề** | **Cơ quan chủ trì** |
| *(Họ tên và chữ ký)* | *(Họ tên và chữ ký)* |

Hà Nội - 2023

MỤC LỤC

[MỤC LỤC 1](#_Toc129761230)

[DANH MỤC HÌNH ẢNH 2](#_Toc129761231)

[1. MÔ HÌNH BÀI TOÁN PHÁT HIỆN MÃ ĐỘC TỔNG THỂ DỰA TRÊN HỌC SÂU 3](#_Toc129761232)

[1.1. Mô hình chung cho các bài toán học máy/học sâu 3](#_Toc129761233)

[1.2. Mô hình bài toán phát hiện mã độc tổng thể dựa trên học sâu 3](#_Toc129761234)

[1.3. Các giai đoạn xử lý 4](#_Toc129761235)

[1.3.1. Xây dựng hồ dữ liệu 4](#_Toc129761236)

[1.3.2. Giai đoạn huấn luyện 7](#_Toc129761237)

[1.3.3. Giai đoạn giám sát và phát hiện 10](#_Toc129761238)

[2. MÔ HÌNH PHÁT HIỆN MÃ ĐỘC ANDROID SỬ DỤNG HỌC SÂU 13](#_Toc129761239)

[2.1. Mô hình phát hiện mã độc Android sử dụng học sâu theo CNN 13](#_Toc129761240)

[2.2. Các vấn đề cần giải quyết trong bài toán phát hiện mã độc Android 13](#_Toc129761241)

[TÀI LIỆU THAM KHẢO 17](#_Toc129761242)

# DANH MỤC HÌNH ẢNH

[Hình 1. Mô hình bài toán học máy 3](#_Toc129761205)

[Hình 2. Mô hình giải pháp phát hiện mã độc tổng thẻ sử dụng học sâu 4](#_Toc129761206)

[Hình 3. Kiến trúc của hồ dữ liệu 6](#_Toc129761207)

[Hình 4. Tóm tắt các bước tiền xử lý 8](#_Toc129761208)

[Hình 5: Các yếu tố cốt lõi trong huấn luyện mạng nơ-ron 11](#_Toc129761209)

[Hình 6: Mô hình bài toán phát hiện mã độc trên nền tảng Android sử dụng CNN 13](#_Toc129761210)

# MÔ HÌNH BÀI TOÁN PHÁT HIỆN MÃ ĐỘC TỔNG THỂ DỰA TRÊN HỌC SÂU

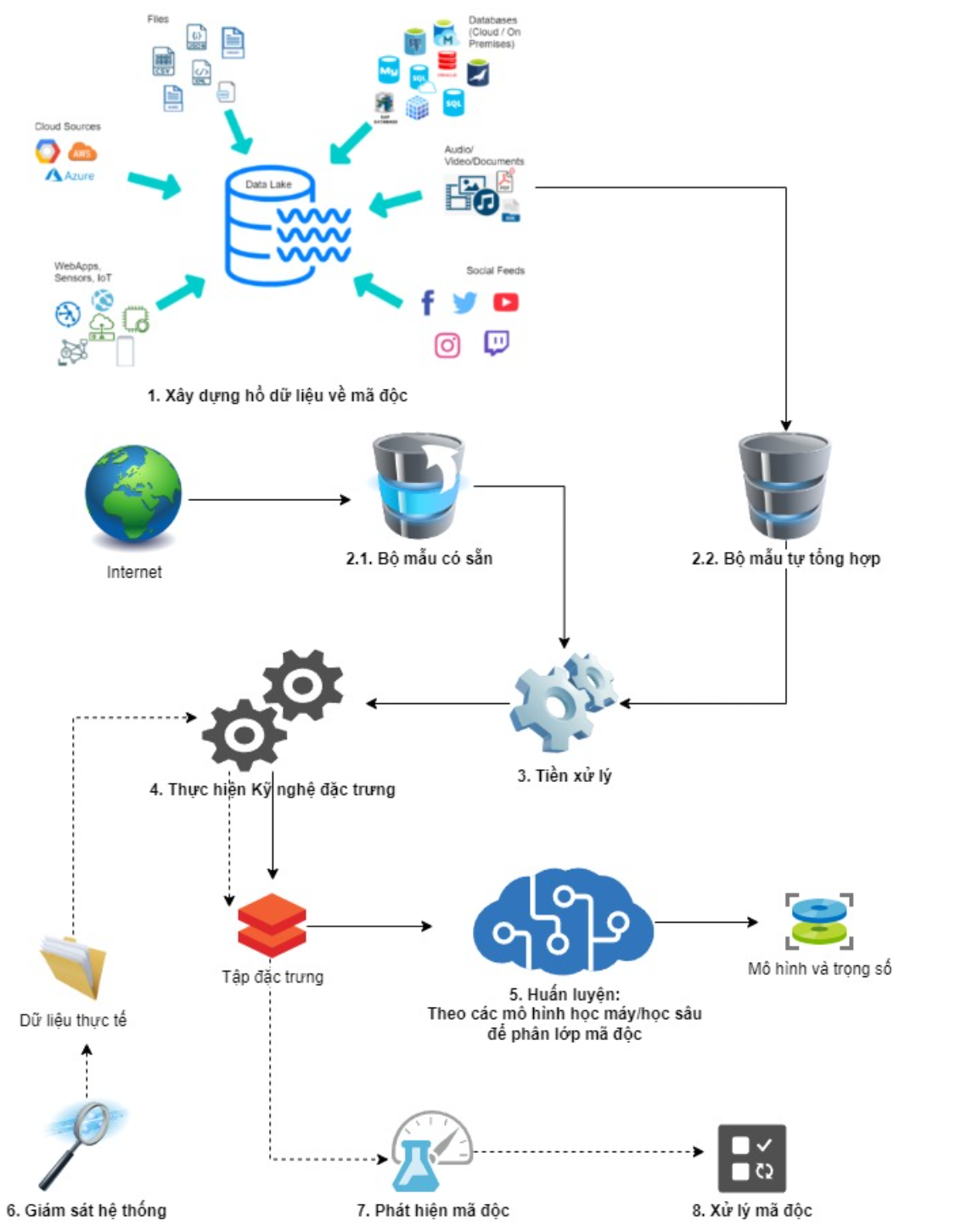
## Mô hình chung cho các bài toán học máy/học sâu

Phần lớn các bài toán học máy/học sâu có thể được mô tả như trong Hình 1, gồm 2 giai đoạn chính là giai đoạn huấn luyện (Training phase) và giai đoạn thực hiện (Testing phase). Giai đoạn thực hiện có thể là phân lớp, phát hiện, nhận diện, dự báo, v.v.



Hình 1. Mô hình bài toán học máy

## Mô hình bài toán phát hiện mã độc tổng thể dựa trên học sâu



Hình 2. Mô hình giải pháp phát hiện mã độc tổng thẻ sử dụng học sâu

## Các giai đoạn xử lý

### Xây dựng hồ dữ liệu

**Khái niệm:**

Hồ dữ liệu (Data lake) là một thuật ngữ được đưa ra bởi Pentaho CTO James Dixon vào năm 2011 để chỉ một kho dữ liệu lớn, không có cấu trúc. Dữ liệu thô được đưa vào lưu trữ trong data lake và người dùng có thể biến đổi, phân loại hay phân tích các phần dữ liệu khác nhau dựa trên nhu cầu của họ và các dữ liệu này cần được xử lý thêm khi có nhu cầu sử dụng. Trong Data lake, tất cả các loại dữ liệu (dữ liệu thô) từ các nguồn của hệ thống đều được lưu trữ. Bao gồm các nguồn dữ liệu có thể bị từ chối lưu trữ trong Data warehouse, chẳng hạn như nhật ký web server, dữ liệu cảm biến, hoạt động trên mạng xã hội, văn bản và hình ảnh, v.v.

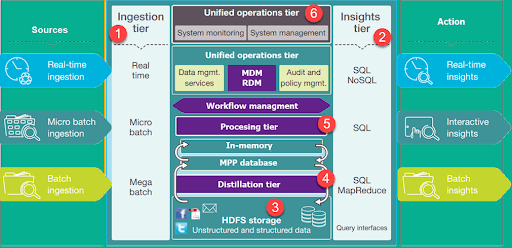
Data lake thậm chí có thể lưu trữ các dữ liệu hiện không được sử dụng nhưng lại có thể cần thiết trong tương lai mà tại thời điểm hiện tại chúng ta chưa thể nhận ra hoặc xác định. Điều này được hiện thực hóa bởi các giải pháp lưu trữ chi phí thấp như nền tảng [Hadoop](http://blog.trginternational.com/vi/thuat-ngu-business-intelligence-analytics-from-a-to-z-2) của Apache.

**Đặc điểm quan trọng của data lake:**

* Một data lake chấp nhận và hỗ trợ dữ liệu từ bất kề nguồn thông tin nào. Data lake không từ chối bất kỳ loại định dạng dữ liệu nào bởi vì theo định nghĩa một data lake tốt cho phép lưu trữ bất kỳ dữ liệu phi cấu trúc nào.
* Data lake hỗ trợ cho tất cả người dùng từ người sử dụng bao gồm cả những chuyên gia về dữ liệu, tất cả đều có thể sử dụng các dữ liệu lưu trữ trong data lake mà họ cần để phục vụ cho các nhu cầu và mục đích khác nhau như phân tích dữ liệu, khai phá dữ liệu, thu thập số liệu v.v.
* Data lake thích ứng dễ dàng với các thay đổi, điều này là một điểm khác biệt lớn so với data warehouse. Trong data lake dữ liệu được cập nhật theo thời gian thực hoặc theo lô, còn trong data warehouse nó được cập nhật theo dạng định kỳ, vì vậy nó có thể ảnh hướng đến việc ra quyết định của một doanh nghiệp.
* Data lake dựa trên các tùy chọn lưu trữ chi phí thấp để lưu trữ dữ liệu, đây cũng là một điểm khác biệt so với công nghệ data warehouse. Trong data warehouse dữ liệu phải được làm sạch và trích xuất một cách thống nhất bất kể nguồn dữ liệu từ đâu. Nhưng trong Data Lake chúng ta không cần làm việc này, chỉ cần sử dụng các công cụ lưu trữ đơn giản để lưu trữ các loại dữ liệu này.

.

**Kiến trúc hồ dữ liệu:**



Hình 3. Kiến trúc của hồ dữ liệu

**Các thành phần của hồ dữ liệu:**

1. **Ingestion Tier:** Các bậc ở bên trái mô tả các nguồn dữ liệu. Dữ liệu có thể được tải vào Data Lake hàng loạt hoặc theo thời gian thực
2. **Insights Tier:** Các bậc bên phải đại diện cho phía nghiên cứu, nơi thông tin chi tiết từ hệ thống được sử dụng. Các truy vấn SQL, NoSQL hoặc thậm chí excel có thể được sử dụng để phân tích dữ liệu.
3. **HDFS** là một giải pháp tiết kiệm chi phí cho cả dữ liệu có cấu trúc và phi cấu trúc. Nó là nơi lưu trữ cho tất cả dữ liệu đang ở trong hệ thống.
4. **Distillation tier** lấy dữ liệu từ tầng lưu trữ và chuyển nó thành dữ liệu có cấu trúc để phân tích dễ dàng hơn.
5. **Processing tier** chạy các thuật toán phân tích và người dùng truy vấn với thời gian thực khác nhau, tương tác, hàng loạt để tạo dữ liệu có cấu trúc để phân tích dễ hơn.
6. **Unified operations tier** quản lý và giám sát hệ thống. Nó bao gồm kiểm toán và quản lý thành thạo, quản lý dữ liệu, quản lý quy trình làm việc.

**Xây dựng hồ dữ liệu cho phân tích mã độc:**

Xây dựng hồ dữ liệu cho bài toán an toàn thông tin nói chung và phạm vi hẹp là phân tích, phát hiện mã độc có ý nghĩa thực tiễn quan trọng. Việc xây dựng hồ dữ liệu nhằm tạo bộ dữ liệu thực tiễn, thực hiện các kỹ thuật tiền xử lý và kỹ nghệ đặc trưng trước khi đưa vào các mô hình học máy/học sâu. Do đó, hồ dữ liệu là thành phần cốt lõi để đưa các phương pháp, thuật toán, mô hình học máy/học sâu vào các bài toán phân lớp, nhận dạng, phát hiện mã độc vào ứng dụng thực tiễn.

### Giai đoạn huấn luyện

#### Tiền xử lý

**Chất lượng dữ liệu:**

Dữ liệu có chất lượng nếu chúng đáp ứng các yêu cầu của mục đích sử dụng. Có nhiều yếu tố bao gồm chất lượng dữ liệu, bao gồm độ chính xác, đầy đủ, nhất quán, kịp thời, đáng tin cậy và khả năng diễn giải.

**Các yếu tố ảnh hưởng đến chất lượng dữ liệu:**

***Dữ liệu không đầy đủ***

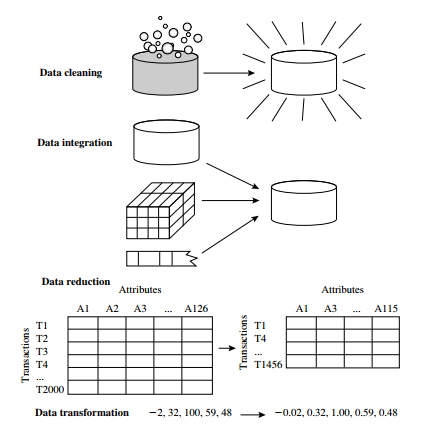
Dữ liệu không đầy đủ có thể xảy ra vì một số lý do. Các thuộc tính quan tâm có thể không phải lúc nào cũng có sẵn, chẳng hạn như thông tin khách hàng cho dữ liệu giao dịch bán hàng. Các dữ liệu khác có thể không được đưa vào đơn giản vì chúng không được coi là quan trọng tại thời điểm nhập. Dữ liệu liên quan có thể không được ghi lại do hiểu nhầm hoặc do trục trặc của thiết bị. Dữ liệu không nhất quán với dữ liệu đã ghi khác có thể đã bị xóa. Hơn nữa, việc ghi lại lịch sử dữ liệu hoặc các sửa đổi có thể đã bị bỏ qua. Dữ liệu bị thiếu, đặc biệt đối với các bộ giá trị bị thiếu cho một số thuộc tính, có thể cần được suy ra.

***Dữ liệu không cập nhật***

Tính cập cũng ảnh hưởng đến chất lượng dữ liệu. Giả sử rằng đang giám sát việc phân phối tiền thưởng doanh số hàng tháng cho các đại diện bán hàng hàng đầu tại AllElectronics. Tuy nhiên, một số đại diện bán hàng không nộp hồ sơ bán hàng của họ đúng hạn vào cuối tháng. Ngoài ra còn có một số điều chỉnh và điều chỉnh sẽ diễn ra sau cuối tháng. Trong khoảng thời gian sau mỗi tháng, dữ liệu được lưu trữ trong cơ sở dữ liệu không đầy đủ. Tuy nhiên, một khi tất cả các dữ liệu được nhận, nó là chính xác. Việc số liệu cuối tháng không được cập nhật kịp thời đã ảnh hưởng không tốt đến chất lượng dữ liệu.

***Độ tin cây và diễn giải***

Hai yếu tố khác ảnh hưởng đến chất lượng dữ liệu là độ tin cậy và khả năng diễn giải. Độ tin cậy phản ánh mức độ tin cậy của dữ liệu đối với người dùng, trong khi khả năng diễn giải phản ánh mức độ dễ hiểu của dữ liệu. Giả sử rằng một cơ sở dữ liệu, tại một thời điểm, có một số lỗi, tất cả đều đã được sửa chữa. Tuy nhiên, các lỗi trong quá khứ đã gây ra nhiều vấn đề cho người dùng bộ phận bán hàng và do đó họ không còn tin tưởng vào dữ liệu nữa. Dữ liệu cũng sử dụng nhiều mã kế toán, mà bộ phận bán hàng không biết cách diễn giải. Mặc dù cơ sở dữ liệu hiện nay chính xác, đầy đủ, nhất quán và kịp thời, nhưng người dùng bộ phận bán hàng có thể coi nó là chất lượng thấp do độ tin cậy và khả năng diễn giải kém.



Hình 4. Tóm tắt các bước tiền xử lý

#### Kỹ nghệ đặc trưng

Các mô hình học máy như ANN, Random Forest, Decision Tree, SVM, kNN,… đều là những mô hình có tính tường minh thấp, độ chính xác cao, độ phức tạp và tính linh hoạt cao. Các mô hình học máy đa dạng sẽ làm phong phú thêm sự lựa chọn của các mô hình. Tuy nhiên bên cạnh việc áp dụng các phương pháp mạnh, mô hình cần phải chuẩn hóa dữ liệu tốt, bởi dữ liệu là nguyên liệu để mô hình dựa trên đó xây dựng một phương pháp học. Nếu mô hình học trên một bộ dữ liệu không tốt, kết quả dự báo sẽ không tốt. Nếu mô hình học trên một bộ dữ liệu tốt, kết quả mô hình tự khắc sẽ được cải thiện. Chính vì thế vai trò của chuẩn hóa dữ liệu rất quan trọng. Những mô hình đơn giản nhưng được xây dựng trên biến chất lượng thường mang lại hiệu quả hơn những mô hình phức tạp như mạng nơ ron hoặc các mô hình kết hợp nhưng được xây dựng trên biến chưa được sử dụng các kỹ thuật tạo đặc trưng.

Về kỹ nghệ đặc trưng chúng ta có 3 phương pháp chính:

* **Trích xuất đặc trưng:** Không phải toàn bộ thông tin được cung cấp từ một biến dự báo hoàn toàn mang lại giá trị trong việc phân loại. Do đó chúng ta cần phải trích lọc những thông tin chính từ biến đó. Chẳng hạn như trong các mô hình chuỗi thời gian chúng ta thường sử dụng kĩ thuật phân rã thời gian để trích lọc ra các đặc trưng như Ngày thành Năm, Tháng, Quý, v.v. Các đặc trưng mới sẽ giúp phát hiện các đặc tính chu kì và mùa vụ, những đặc tính mà thường xuất hiện trong các chuỗi thời gian.
* **Biến đổi đặc trưng:** Biến đổi dữ liệu gốc thành những dữ liệu phù hợp với mô hình nghiên cứu. Những biến này thường có tương quan cao hơn đối với biến mục tiêu và do đó giúp cải thiện độ chính xác của mô hình. Các phương pháp này bao gồm:
* Chuẩn hóa và thay đổi phân phối của dữ liệu thông qua các kĩ thuật feature scaling như Minmax scaling, Mean normalization, Unit length scaling, Standardization.
* Tạo biến tương tác: Trong thống kê các bạn hẳn còn nhớ kiểm định ramsey reset test về mô hình có bỏ sót biến quan trọng? Thông qua việc thêm vào mô hình các biến bậc cao và biến tương tác để tạo ra một mô hình mới và kiểm tra hệ số các biến mới có ý nghĩa thống kê hay không. Ý tưởng của tạo biến tương tác cũng gần như thế. Tức là chúng ta sẽ tạo ra những biến mới là các biến bậc cao và biến tương tác.
* Xử lý dữ liệu thiếu (missing data): Có nhiều lý do khiến ta phải xử lý missing data. Một trong những lý do đó là dữ liệu missing cũng mang những thông tin giá trị, do đó nếu thay thế được các missing bằng những giá trị gần đúng sẽ mang lại nhiều thông tin hơn cho mô hình. Bên cạnh đó nhiều mô hình không làm việc được với dữ liệu missing dẫn tới lỗi training. Do đó ta cần giải quyết các biến missing. Đối với biến numeric, các phương pháp đơn giản nhất là thay thế bằng mean, median.
* **Lựa chọn đặc trưng:** Phương pháp này được áp dụng trong những trường hợp có rất nhiều dữ liệu mà chúng ta cần lựa chọn ra dữ liệu có ảnh hưởng lớn nhất đến sức mạnh phân loại của mô hình. Các phương pháp có thể áp dụng đó là ranking các biến theo mức độ quan trọng bằng các mô hình như Random Forest, Linear Regression, Neural Network, SVM, v.v.
* **Cải tiến tập đặc trưng:** Sinh ra các đặc trưng kết hợp từ các đặc trưng ban đầu.

#### Huấn luyện mạng

Không phải tất cả các mô hình học máy/học sâu đều sử dụng những kỹ thuật giống nhau, vì vậy việc huấn luyện các mô hình này sẽ phụ thuộc vào mục đích sử dụng. Mạng nơ-ron là một tổ hợp các thuật toán Học máy mà lấy ý tưởng từ việc mô phỏng các tế bào thần kinh trong não người.

Mạng no-ron hiện nay đang liên tục được phát triển theo các nhánh, hướng khác nhau để làm vô số các nhiệm vụ như xử lý ngôn ngữ tự nhiên, thị giác máy tính, nhận dạng tiếng nói, v.v. Hầu hết các mô hình học máy/học sâu hiện đại nhất đều dựa trên cơ sở mạng nơ-ron. Bốn yếu tố chính trong việc huấn luyện một mô hình mạng nơ-ron được tổng hợp như trong Hình 5:

### Giai đoạn giám sát và phát hiện

Sau khi thực hiện giai đoạn huấn luyện, kết quả thu được là bộ tham số cấu trúc mô hình phù hợp và bộ trọng số. Kết quả huấn luyện này sẽ được sử dụng trong giai đoạn phát hiện mã độc. Trong giai đoạn phát hiện, hệ thống luôn được giám sát và thực hiện rà quét mã độc khi xử lý file hoặc thực thi chương trình.

Các tập tin đưa vào để phân lớp cũng sẽ đi vào mạng nơ-ron tích chập, sau khi tới lớp làm phẳng ma trận, dựa vào mảng trọng số đã được huấn luyện ở trên, đưa ra kết quả phân lớp. Từ kết quả phân lớp này, so sánh với nhãn thật của các tập tin để đưa vào phát hiện từ đó đưa ra được tỉ lệ kết quả phân lớp đúng.

Việc ứng dụng mô hình học sâu vào phát hiện mã độc trong những năm gần đây được nhiều nhóm nghiên cứu quan tâm điển hình như DBN, DNN, RNN, LSTM, CNN



Hình 5: Các yếu tố cốt lõi trong huấn luyện mạng nơ-ron

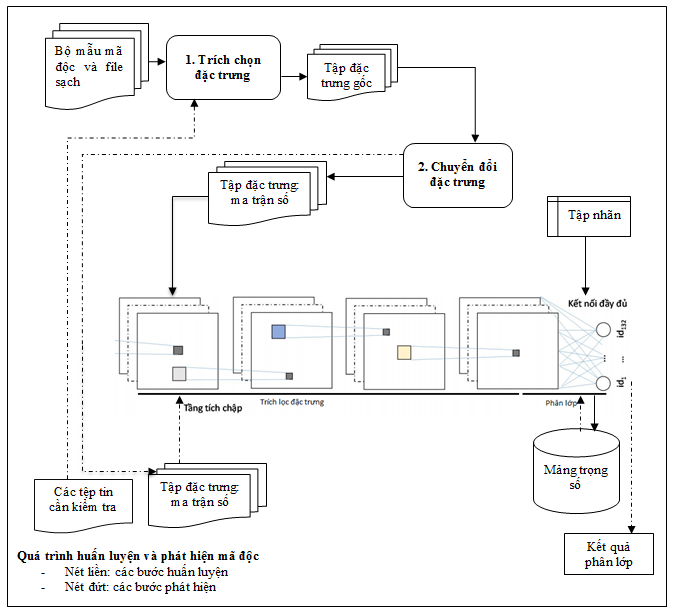
**Các bước tiền xử lý chính:**

Hình dưới tóm tắt các bước tiền xử lý dữ liệu được mô tả ở đây. Lưu ý rằng phân loại trước đó không loại trừ lẫn nhau. Ví dụ, việc loại bỏ dữ liệu thừa có thể được coi là một hình thức làm sạch dữ liệu, cũng như giảm bớt dữ liệu

# 2. MÔ HÌNH PHÁT HIỆN MÃ ĐỘC ANDROID SỬ DỤNG HỌC SÂU

## 2.1. Mô hình phát hiện mã độc Android sử dụng học sâu theo CNN

Bài toán phát hiện mã độc trên nền tảng Android sử dụng học sâu theo mạng nơ-ron tích chập (CNN) được mô tả như trong Hình 6.



Hình 6: Mô hình bài toán phát hiện mã độc trên nền tảng Android sử dụng CNN

## 2.2. Các vấn đề cần giải quyết trong bài toán phát hiện mã độc Android

Trong bài toán tổng quát, có 4 vấn đề cần giải quyết trong bài toán phát hiện mã độc trên nền tảng Android.

1. **Trích xuất đặc trưng**

File APK là một file nén, bên trong chứa các file khác như Androidmanifest.xml, classes.dex, etc. Để có thể huấn luyện dựa trên bộ dữ liệu bao gồm nhiều file APK cần phải lấy được các đặc trưng. Các đặc trưng là quan trọng, việc lấy thành phần nào là đặc trưng còn tùy thuộc vào người phát triển. Tuy nhiên, để lấy ra được các chuỗi làm đặc trưng ta có thể sử dụng hai phương pháp cơ bản là phương pháp phân tích tĩnh và phương pháp phân tích động. Các chuỗi do người phát triển lấy có thể áp dụng nhiều cách thức để thành bộ đặc trưng. Ví dụ, có thể chuyển toàn bộ mã nguồn lấy đấy trong classes.dex thành đặc trưng ảnh, hoặc lấy các nhóm đặc trưng như permission, API, intent, v.v. hay chuyển file code thành các file .smali. Nhìn chung bài toán trích xuất đặc trưng có nhiều phương pháp để tạo ra được bộ đặc trưng. Ở đây, bộ đặc trưng sau khi trích xuất để sử dụng tôi gọi là “Feature original dataset”.

1. **Lựa chọn đặc trưng phù hợp**

Các đặc trưng trong bộ dữ liệu gốc (Feature original dataset) đã được chuyển sang dạng số, các giá trị số này có thể được quy định khác nhau như:

* Ảnh: chuyển từ text sang binary hoặc Hex để lấy thành ảnh điểm.
* Thống kê số lần xuất hiện của từng đặc trưng (permission, API, v.v.) ở trong các file của APK để thành đặc trưng.
* Thống kê yes/no: nếu xuất hiện đặc trưng đó thì điền *“1”*, nếu không xuất hiện thì điền *“0”*.
* áp dụng mô hình toán cho mối liên hệ giữa các đặc trưng để đánh trọng số cho mối liên hệ đó. Trọng số chính là đặc trưng lấy (Ví dụ: mối tương quan của các API).
* Etc.

Có thể sử dụng các công thức toán học vào để đánh lại trọng số cho từng đặc trưng như các thuật toán IF-IDF (Term Frequency – Inverse Document Frequency), IG (Information Gain), PSO (Particle Swarm Optimization), GA (Genetic Algorithm), v.v. để thay đổi giá trị của từng đặc trưng trong mỗi file. Mặt khác, có thể sử dụng các thuật toán trên để đánh giá mức độ quan trọng của từng đặc trưng trong bộ dữ liệu. Nhiều nghiên cứu không sử dụng việc đánh trọng số mới mà sử dụng luôn bộ dữ liệu đặc trưng gốc.

Bộ dữ liệu có thể chứa hàng trăm, hàng nghìn, hàng chục nghìn đặc trưng. Dĩ nhiên, khi đưa càng nhiều đặc trưng vào mô hình huấn luyện thì thời gian huấn luyện càng lâu. Mặt khác, số lượng đặc trưng nhiều chưa chắc đã tốt trong bài toán phân lớp. Do đó, việc lựa chọn đặc trưng phù hợp cũng là bài toán được nghiên cứu trong việc phân lớp nói chung và phân lớp mã độc nói riêng. Việc lựa chọn đặc trưng nào, loại bỏ đặc trưng nào phụ thuộc tiêu chí của mỗi nhà nghiên cứu đưa ra, ví dụ, có thể tạo một ngưỡng của accurancy, recall, etc. để dừng việc loại bỏ đặc trưng hoặc đơn giản là ta dựa vào giá trị trọng số được đánh của mỗi đặc trưng trong bước 2 để loại bỏ đặc trưng với ngưỡng là do mình đặt.

1. **Kết hợp, sinh đặc trưng từ tập sẵn có**

Với bộ đặc trưng, có thể áp dụng một số mô hình toán học để sinh thêm các đặc trưng. Việc sinh thêm các đặc trưng có thể dựa vào mối tương quan của các đặc trưng, lai ghép giữa các đặc trưng, etc.

Một số phương pháp sinh đặc trưng như apriori, K-mean, FP-growth, etc.

Việc áp dụng sinh đặc trưng cũng rất phù hợp với các mô hình học sâu như CNN. Do đó, có nhiều nhóm nghiên cứu đã áp dụng sinh đặc trưng trong bài toán phân lớp.

1. **Phát triển mô hình huấn luyện**

Việc áp dụng mô hình mới trong huấn luyện luôn được quan tâm trong tất cả các bài toán phân lớp dữ liệu. Đặc biệt trong phân lớp ảnh, các nhà nghiên cứu đã áp dụng nhiều mô hình học máy nhằm nâng cao chất lượng phân lớp. Từ đó, sinh ra các mô hình mới mà học sâu là nhánh được sinh ra từ đó. Trong học sâu, mô hình điển hình được sử dụng là CNN, kèm theo CNN thì có rất nhiều biến thể của CNN như VGG-16, VGG19, ResNet, etc.

Từ đó thấy rằng phát triển hay áp dụng mô hình mới vào một lĩnh vực phân lớp có ý nghĩa lớn. Rất nhiều nhóm nghiên cứu đã áp dụng các mô hình ở các bài toán khác và các mô hình mình đề xuất cho bài toán phát hiện mã độc trên Android.

Tổng quát bài toán phát hiện mã độc trên Android hướng đến sẽ tạo ra một sản phẩm phát hiện mã độc thử nghiệm và sẽ áp dụng, có cải tiến ở trong năm phần phía trên. Đó cũng là đóng góp của nghiên cứu sinh trong việc áp dụng và phát triển ở bài toán phát hiện mã độc trên nền tảng Android.

# TÀI LIỆU THAM KHẢO

1. Sun, Y., Chen, Y., Pan, Y., Wu, L.: Android malware family classifica- tion based on deep learning of code images. International Journal of Computer Science 46(4) (2019). https://doi.org/10.1016/j.comnet.2020. 107138
2. Li, D., Zhao, L., Cheng, Q., Lu, N., Shi, W.: Opcode sequence analy- sis of android malware by a convolutional neural network. Concurrency and Computation: Practice and Experience 32 (2019). https://doi.org/ 10.1002/cpe.5308
3. Lê Đức Thuận, Phạm Ngọc Hưng, Phạm Văn Hưởng, Nguyễn Đức Trung, “Phát hiện mã độc Android dựa trên mạng niềm tin sâu”, pp 430-436, FAIR’11, 2018, ISBN 978-604-913-749-5
4. Le Duc Thuan, Pham van Huong, Le Thi Hong Van, Hoang Quoc Cuong, Hoang Van Hiep, Nguyen Kim Khanh, “Android malware detection based on deep learning using convolutional neural network”, pp 32-41, Tạp chí Nghiên cứu khoa học và công nghệ quân sự, 8/2019, ISSN 1859-1043.
5. L. D. Thuan, P. V. Huong, H. V. Hiep and N. K. Khanh, "Improvement of feature set based on Apriori algorithm in Android malware classification using machine learning method," 2020 RIVF International Conference on Computing and Communication Technologies (RIVF), 2020, pp. 1-7, doi: 10.1109/RIVF48685.2020.9140779.
6. Le Duc Thuan, Hoang Van Hiep, Nguyen Kim Khanh, “Android Malware classification using deep learning CNN with Co-occurrance matrix feature”, Tạp chí khoa học và công nghệ các trường đại học kỹ thuật, 2021.
7. P. V. Huong, L. D. Thuan, L. T. Hong Van and D. V. Hung, "Intrusion Detection in IoT Systems Based on Deep Learning Using Convolutional Neural Network," 2019 6th NAFOSTED Conference on Information and Computer Science (NICS), 2019, pp. 448-453, doi: 10.1109/NICS48868.2019.9023871.
8. L. T. Hong Van, P. V. Huong, L. D. Thuan and N. H. Minh, "Improving the Feature Set in IoT Intrusion Detection Problem Based on FP-Growth Algorithm," 2020 International Conference on Advanced Technologies for Communications (ATC), 2020, pp. 18-23, doi: 10.1109/ATC50776.2020.9255431.
9. Le Thi Hong Van, Pham Van Huong, Ta Quang Hua, Le Duc Thuan, Nguyen Hieu Minh, Feature Generation by K-means for Convolutional Neural Network in Detecting IoT System Attacks, RICE 2021