PHÂN LOẠI MÃ ĐỘC DỰA TRÊN CÁC KỸ THUẬT N-GRAM VÀ HỌC MÁY

Nguyễn Thị Thu Trang, Nguyễn Đại Thọ, Vũ Duy Lợi
Khoa Công nghệ thông tin, Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội

Tóm tắt: Mã độc đang là mối đe dọa lớn đến an ninh của các hệ thống máy tính. Vì vậy phân loại mã độc để có những biện pháp đối phó thích hợp là một phần quan trọng trong lĩnh vực an toàn thông tin. Trong bài báo này chúng tôi cải tiến giải thuật trích rút điểm mẫu trong quy trình phân loại mở mã độc dựa trên điểm mẫu được đề xuất bởi Rieck và các công sự [1]. Chúng tôi áp dụng tư tưởng của hai phương pháp phân loại dựa trên điểm đặc trưng là Dendroid [2] và phương pháp được đề xuất bởi P. Shrestha và các cộng sự [3], trích rút các điểm mẫu trên từng họ mã độc thay vì dựa trên toàn bộ các họ mã độc. Thực nghiệm cho thấy phương pháp của chúng tôi cho kết quả phân loại tốt với khả năng nhận biết mã độc là 0.981% và phát hiện mã độc mới là 0.988% cao hơn phiên bản gốc của phương pháp dựa trên điểm mẫu [1].

Từ khóa: Phân loại mã độc, n-gram, điểm mẫu.

I. GIỚI THIỆU

Phần mềm độc hại (hay mã độc) là một chương trình được chèn vào hệ thống, thường là cố tình, với mục đích xâm phạm tính bảo mật, toàn vẹn, hoặc tính khả dụng của dữ liệu, ứng dụng, hệ điều hành của nạn nhân hoặc gây phiền nhiễu, làm gián đoạn nạn nhân [4]. Phần mềm độc hại có nhiều loại như virus, Trojan, worm, phần mềm gián điệp, rootkit, v.v. Cùng với sự phát triển nhanh chóng của Internet, phần mềm độc hại cũng đang ngày một gia tăng. Theo thống kê của Kaspersky, 21.643.947 đối tượng phần mềm độc hại đã được phát hiện vào năm 2018 gấp hơn 5 lần so với năm 2015 [5]. Mặc dù có sự cải thiện đáng kể của các cơ chế an ninh, nhưng các phần mềm độc hại đang ngày một tinh vi và có các cơ chế lẩn trốn nên chúng vẫn đang là một trong những mối đe dọa lớn nhất đối với các hệ thống máy tính. Vì vậy phát hiện và phân tích hành vi của các phần mềm độc hại là một nhiệm vụ rất quan trọng để làm giảm tối đa những thiệt hại do chúng gây ra.

Phân tích mã độc đề cập đến quá trình xác định mục đích, hành vi, phương pháp tấn công và cách thức lan truyền của chúng. Phân tích mã độc được chia làm hai loại là phân tích tĩnh và phân tích động.

Phân tích tĩnh hay còn được gọi là phân tích mã tĩnh để cố gắng suy ra các hành vi của phần mềm, là quá trình phân tích phần mềm mà không cần thực thi mã hoặc chương trình. Các mẫu phát hiện có thể được trích xuất trong phân tích tĩnh như: các Lời gọi hệ thống, API, signature, biểu đồ điều khiển, opcode, bytecode, các file DLL được gọi...Ưu điểm phân tích tĩnh là an toàn và chúng ta có thể quan sát hết các phần của chương trình. Nhược điểm của phân tích tĩnh là khó phân tích với các phần mềm sử dụng kỹ thuật che giấu, mã hóa và đóng gói.

Phân tích động tiến hành thực thi các phần mềm độc hại trong môi trường sandbox được giám sát để thu thập các hành vi của mã độc. Vì vậy sandbox cần phải an toàn. Trong loại phân tích này, có thể thu thập được tất cả các thuộc tính của hành vi, chẳng hạn như các tệp tin đc được mở, tạo mutexes, các hàm chính xác được gọi, các đối số của hàm. Ưu điểm của phân tích động là nó nhanh hơn nhiều phân tích tĩnh. Nhược điểm của phân tích động, chúng ta chỉ nhìn thấy một kịch bản có liên quan đến hiện tại của hệ thống và không phải hành vi nào cũng được phân tích (ví dụ như virus chờ đến một thời điểm nào đó mới hoạt động).

Theo [6], phân tích tự động mã độc hướng tới một trong ba mục tiêu là phát hiện, phân tích độ tương tự và phân loại. Chúng tôi chú tâm nghiên cứu đến vấn đề phân loại mã độc. Trong nghiên cứu, người ta thường sử dụng các phương pháp học máy có giám sát để giải quyết bài toán phân loại một cách tự động. Theo [7] có hai cách tiếp cận nổi bật là dựa trên mô hình (model-based learning) và dựa trên thể hiện (instance-based learning). Với học máy dựa trên mô hình (SVM, cây quyết định, Naive Bayes v.v.), các giải thuật này sẽ tạo ra mô hình khái quát hoá dữ liệu huấn luyện vì vậy không thích hợp với các bài toán có dữ liệu phức tạp. Mặt khác, các phương pháp học máy dựa trên thể hiện (k-NN,..) không khái quát hoá dữ liệu mà sử dụng luôn dữ liệu để phân loại bằng cách so sánh dữ liệu cần phân loại với dữ liệu huấn luyện, vì vậy có thể tối ưu hoá những trường hợp cụ thể và thích hợp hơn với các bài toán phức tạp như phân loại. Với học máy dựa trên thể hiện có thể sử dụng toàn bộ tập dữ liệu huấn luyện để phân loại, nhưng nhược điểm là thời gian phân loại lâu. Vì vậy người ta sử dụng một phương pháp khác của học máy dựa trên thể hiện là phương pháp sử dụng các điểm mẫu để đại diện cho tập dữ liệu huấn luyện và phân loại dựa trên các điểm mẫu này thay vì sử dụng toàn bộ tập dữ liệu huấn luyện. Vì những lý do trên, chúng tôi chọn sử dụng phương pháp học máy dựa trên thể hiện có sử dụng các điểm mẫu để phân loại mã độc"

Ba công trình nghiên cứu về phân loại mã độc sử dụng điểm mẫu được chúng tôi quan tâm đến là: phương pháp là phân loại mở dựa trên điểm mẫu của Rieck cùng các cộng sự [1], hai phương pháp dựa trên điểm đặc trưng là phương pháp Dendroid- áp dụng với các mã độc trên Android [2] và phương pháp được đề suất bởi P. Shrestha và các cộng sự [3] . Phương pháp thứ nhất sử dụng thông tin về các n-gram của chuỗi các lời gọi hệ thống, đặc trưng là sự xuất hiện hay không của các n-gram và sử dụng điểm mẫu (prototype) để đại diện cho các cụm mã độc. Phương pháp này có ba thành

Tác giả liên hệ: Nguyễn Đại Thọ,
Email: nguyendaitho@vnu.edu.vn

phần chính là: trích rút điểm mẫu giúp tìm ra các điểm mẫu đại diện cho các cụm, phân cụm sử dụng điểm mẫu giúp gộp nhóm các cụm tương tự nhau thành một cụm lớn hơn, phân lớp sử dụng điểm mẫu để dự báo nhãn lớp cho mã độc chưa biết và phát hiện ra những mẫu mã độc mới. Phương pháp thứ hai – Dendroid [2] là một phương pháp phân loại dựa trên điểm đặc trưng, áp dụng các kỹ thuật của lĩnh vực phân loại văn bản. Điểm đặc biệt của phương pháp này là sử dụng một điểm đặc trưng được tạo ra từ các mã độc trong cùng một họ để đại diện cho toàn bộ họ đó. Kết quả của quá trình trích rút đặc trưng là một vector đại diện chung cho một họ mã độc thay vì đại diện cho từng mã độc cụ thể. Các điểm đặc trưng được dùng kết hợp với thuật toán 1NN (One Nearest Neighbor) để phân loại mã độc. Mã độc mới được phân vào họ của điểm đặc trưng gần nó nhất. Phương pháp này là phân loại đóng. Một điểm dữ liệu khi được cho vào phân loại sẽ chắc chắn được phân vào một lớp đã biết. Phương pháp thứ 3 cũng xây dựng các điểm đặc trưng từ các mã độc trong cùng một họ và mỗi họ mã độc được đại diện bởi một điểm điểm đặc trưng giống như phương pháp Dendroid. Sau đó mẫu mã độc cần phân loại được tính độ tương tự với các điểm đặc trưng của các họ và được phân vào họ có độ tương tự lớn nhất.

Chúng tôi vận dụng tư tưởng phương pháp thứ hai và ba để cải tiến giải thuật trích rút điểm mẫu của phương pháp thứ nhất. Trong quá trình trích rút điểm mẫu, thay vì trích rút điểm mẫu dựa trên toàn bộ dữ liệu huấn luyện, chúng tôi sẽ trích rút điểm mẫu trên dữ liệu của từng họ. Khi đó một họ có thể có một hoặc nhiều điểm mẫu. Chúng tôi vẫn giữ quy trình tổng quát chung như phương pháp dựa trên điểm mẫu [1] để phân loại mở các mã độc, phân loại các mã độc đã biết và nhận biết được các mã độc có hành vi mới. Chúng tôi đánh giá hiệu quả của phương pháp đề xuất theo cả hai khả năng phân loại đúng những mã độc đã biết và nhận biết nhưng mã độc mới sử dụng cùng độ đo $F1_{micro}$ (tổng hợp của hai thông số phổ biến là độ chính xác và độ hồi tưởng). Kết quả thực nghiệm cho thấy phương pháp của chúng tôi đạt $F1_{micro}$ là 98.1% đối với các mã độc đã biết và 98.8% đối với các mã độc mới, đều cao hơn các độ đo tương ứng của phiên bản gốc của phương pháp thứ nhất.

II. CÁC CÔNG TRÌNH LIÊN QUAN

A. Phương pháp dựa trên điểm mẫu

Theo phương pháp dựa trên điểm mẫu [1], đầu tiên mẫu mã độc cần phân loại được đưa vào sandbox (một môi trường thực thi giả lập) để thu thập các hành vi. Thông tin của hành vi mã độc được nhúng vào không gian vector đặc trưng sau đó được đưa vào thành phần phân loại sử dụng điểm mẫu. Nếu phân loại thành công thì mã độc sẽ được gán nhãn về một họ mã độc đã biết, nếu không nó sẽ được đưa vào tập mã độc chưa biết để làm đầu vào cho giai đoạn trích rút điểm mẫu để tìm ra điểm mẫu đại diện cho các mã độc. Thành phần phân cụm sử dụng các điểm mẫu thu được để phân cụm các điểm mẫu. Tập điểm mẫu của các cụm còn được sử dụng cho quá trình phân loại tiếp theo. Quy trình tổng thể của phương pháp được mô tả trong Hình 1.

Cụ thể, ta có quy trình như sau:

1) Giai đoạn 1: Chạy mã độc trong môi trường sandbox

- *Đầu vào*: File thực thi mã độc
- *Đầu ra:* Chuỗi các lời gọi hệ thống

Đầu tiên mã độc được chạy trong môi trường sandbox để giám sát các hành vi và thu thập các chuỗi lời gọi hệ thống đặc trưng cho các hành vi của chúng.

2) Giai đoạn 2: Nhúng các hành vi vào không gian vector

- *Đầu vào:* Chuỗi các lời gọi hệ thống
- *Đầu ra:* Vector đặc trưng đại diện cho mã độc

Chuỗi lời gọi hệ thống sau đó được nhúng vào không gian vector sử dụng n-gram. Mỗi thành phần của một vector đặc trưng thể hiện sự xuất hiện hay không của n-gram tương ứng trong chuỗi lời gọi hệ thống.

- Giả sử tập $\boldsymbol{S}$= {(x_1, x_2, x_3, x_4, ..., x_n)| $x_i \in A$ *với* $1 \leq i \leq n$} là tập tất cả n-gram có thể có
- $\boldsymbol{A}$ là tập tất cả các lời gọi hệ thống khác nhau.

Với mỗi báo cáo *x*, hành vi mã độc có thể nhúng vào một không gian vector có |*S*| chiều. Mỗi chiều sẽ tương ứng với một n-gram. Giá trị các chiều của vector được tính như sau*:*

$$\varphi(x) = (\varphi_s(x))_{sCS} \quad (1)$$

Trong đó: $\varphi_s(x) = 1$ nếu mẫu báo cáo hành vi *x* chứa n-gram *s*, ngược lại $\varphi_s(x) = 0$

Ví dụ: A = {a_1, a_2}

→ S= {a_1a_1, a_1a_2, a_2a_1, a_2a_2}

Mẫu báo cáo $x = a_1a_2a_1a_1a_2$

Bảng I. Sự xuất hiện của các 2-gram

2-gram	a_1a_1	a_1a_2	a_2a_1	a_2a_2
xuất hiện	1	1	1	0

→Vector đặc trưng cho mẫu báo cáo x là:

x = (1, 1, 1, 0)

Chuẩn hóa vector: Sau khi thu được vector đặc trưng của mẫu báo cáo x, chúng ta chuẩn hóa vector để đưa nó về vector có độ dài bằng 1 bằng cách chia cho độ dài Euclid của vector đó:

$$|x| = \sqrt{1^2 + 1^2 + 1^2 + 0^2} = \sqrt{3}$$

$$x = \left(\frac{1}{|x|}, \frac{1}{|x|}, \frac{1}{|x|}, \frac{0}{|x|}\right) = \left(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, 0\right)$$

Sau bước 2, ta thu được các vector đặc trưng cho mỗi mã độc.

3) Giai đoạn 3: Phân loại sử dụng điểm mẫu

- *Đầu vào:* Vector đặc trưng đại diện cho mã độc
- *Đầu ra:* Nhãn lớp dự đoán cho mã độc

Sau bước 2, vector đặc trưng cho mã độc được đưa vào mô hình phân lớp để dự đoán nhãn lớp cho mã độc đó. Thuật toán được sử dụng để phân lớp là 1NN và một ngưỡng d_r để loại ra những mẫu mã độc mới. Khi một điểm dữ liệu được đưa vào, chúng ta sẽ xem khoảng cách của nó tới các điểm mẫu. Nếu khoảng cách của nó đến điểm mẫu gần nhất nhỏ hơn ngưỡng d_r thì điểm dữ

liệu mới đó sẽ được phân vào lớp của điểm mẫu gần nó nhất và đưa ra báo cáo lớp dự đoán được, ngược lại sẽ đưa nó vào tập mã độc chưa biết và đưa vào giai đoạn trích rút điểm mẫu. Tại thời điểm đầu tiên, chưa có điểm mẫu nào nên giai đoạn này không được thực hiện.

4) Giai đoạn 4: Trích rút điểm mẫu

- *Đầu vào:* Tập các mã độc chưa biết nhãn lớp
- *Đầu ra:* Tập các điểm mẫu đại diện cho các mã độc

Rieck và các cộng sự sử dụng giải thuật được đề xuất bởi Gonzalez trong công trình [9] để trích rút các điểm mẫu từ tập các mã độc chưa biết nhãn lớp. Bằng cách tham chiếu khoảng cách tới điểm mẫu gần nó nhất, ta tìm ra được các điểm mẫu đại diện cho các mã độc đó. Phương pháp dựa trên ý tưởng mã độc có khoảng các càng gần nhau thì khả năng thuộc một cùng họ càng cao. Do chúng tôi tập trung vào cải tiến giải thuật trích rút điểm mẫu nên chúng tôi trình bày giải thuật này ở Hình 2 và giải thích bên dưới.

5) Giai đoạn 5: Phân cụm sử dụng điểm mẫu

- *Đầu vào:* Tập các điểm mẫu đại diện cho các mã độc
- *Đầu ra:* Tập các cụm điểm mẫu

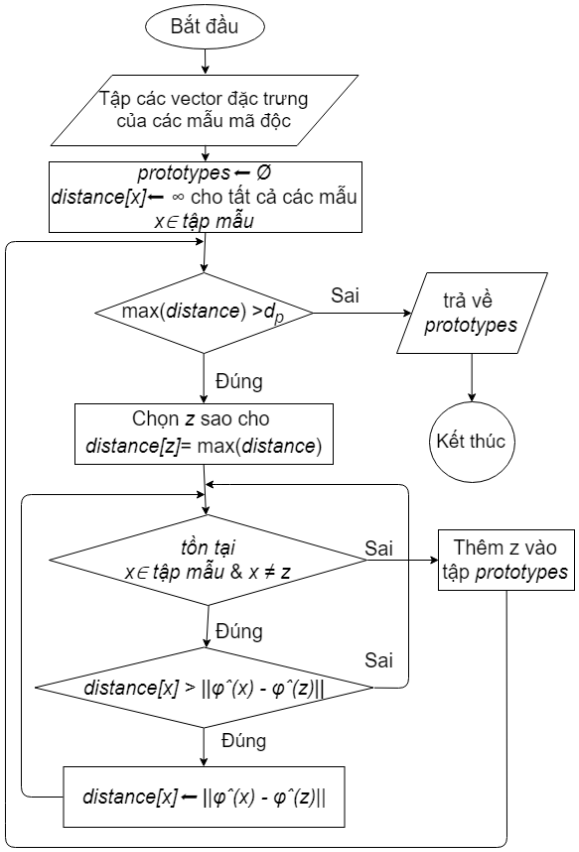
Phân cụm trên các điểm mẫu thay vì trên toàn bộ tập dữ liệu để gom nhóm các cụm thành một cụm lớn hơn sử dụng phân cụm phân cấp (hierarchical clustering) [8]. Kết quả được sử dụng trong quá trình phân lớp tiếp theo.

Giải thích giải thuật trích rút điểm mẫu của Gonzalez [9]:

Bước 1: Khởi tạo tập *prototypes* = Ø, biến *prototypes* lưu các điểm mẫu và mảng *distance* lưu khoảng cách có giá trị bằng ∞ lưu lại khoảng cách của điểm dữ liệu đến điểm mẫu gần nó nhất.

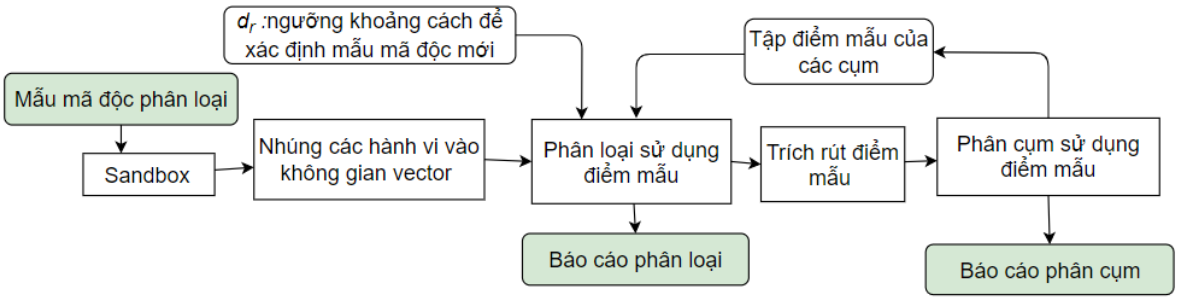
Bước 2: Kiểm tra khoảng cách lớn nhất có lớn hơn ngưỡng d_p không và lặp đến khi điều kiện đó không được thỏa mãn thì kết thúc. Tại vòng lặp đầu tiên, các khoảng cách là ∞ nên ta chọn ngẫu nhiên một điểm *z* làm điểm mẫu. Với các vòng lặp tiếp theo ta chọn điểm mẫu z là điểm dữ liệu có khoảng cách lớn nhất.

Bước 3: Với các điểm dữ liệu khác điểm mẫu lưu trong biến *prototypes*, tính khoảng cách của tất cả các điểm dữ liệu đó so với điểm mẫu mới được tìm. Nếu khoảng cách vừa tính được nhỏ hơn khoảng cách với điểm mẫu trước đó, chúng ta cập nhật lại khoảng cách của điểm dữ liệu đó và thêm *z* vào tập các *prototypes* và quay lại bước 2. Sau mỗi lần lặp chúng ta sẽ thu được một điểm mẫu đại diện cho một cụm dữ liệu.



Hình 1. Giải thuật trích rút điểm mẫu của Gonzalez

Nhận xét phương pháp: Phương pháp phân loại dựa trên điểm mẫu [1] là phương pháp phân loại mở, giúp chúng ta có thể phân loại và phát hiện ra những mẫu mã độc mới. Bên cạnh đó phương pháp này sử dụng điểm mẫu (tương tự như nén dữ liệu) làm giảm dữ liệu phải xử lý nên giảm thời gian phân loại. Phương pháp này cũng có tính năng học tăng cường cho phép cập nhật mô hình phân loại khi có thêm dữ liệu mới mà không cần huấn luyện lại. Nhược điểm của nó là các điểm mẫu được trích rút trên toàn bộ tập dữ liệu có thể dẫn đến những dữ liệu không cùng một họ mã độc có thể thuộc chung một cụm, hoặc điểm mẫu chưa chắc đã cùng lớp với đại đa số các điểm dữ liệu trong cụm nhưng lại được dùng làm đại diện cho cụm đó, điều đó là không nên.

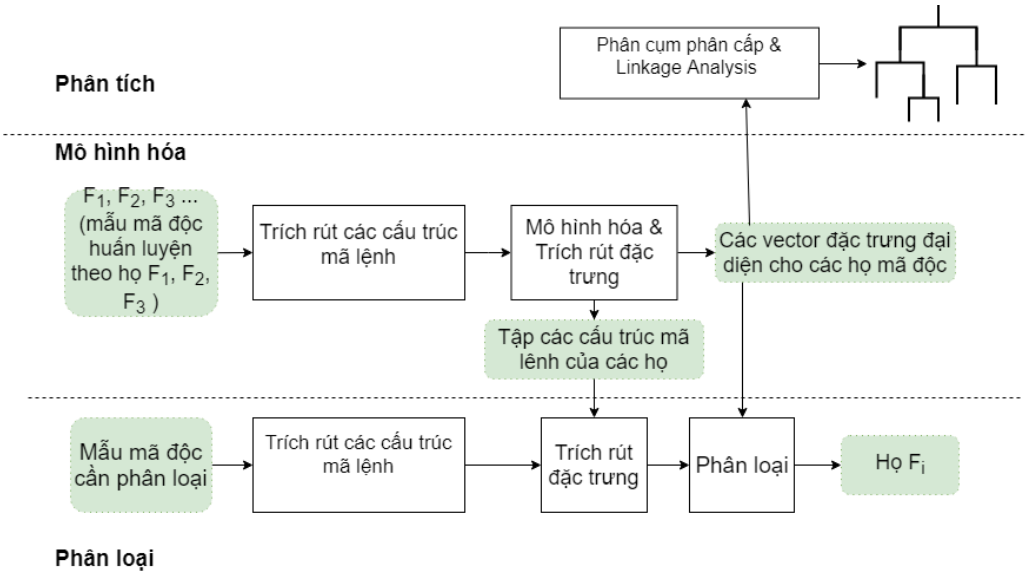


Hình 2. Quy trình của phương pháp sử dụng điểm mẫu

B. Các phương pháp dựa trên điểm đặc trưng

a, Phương pháp Dendroid

Phương pháp dựa trên điểm đặc trưng Dendroid [2] là một phương pháp dựa trên kỹ thuật khai thác văn bản và truy xuất thông tin trên nền tảng Android. Từ tất cả các mẫu mã độc trong một họ, phương pháp tổng hợp và tính ra một vector đặc trưng đại diện cho họ đó thay vì từng vector đặc trưng đại diện cho mỗi mẫu mã độc (ví dụ có 6 họ mã độc sẽ có 6 vector đặc trưng). Vector đặc trưng này có thể là một điểm ảo hoặc có thể trùng với điểm thật, nó được tính dựa trên tiếp cận khai thác văn bản. Sau đó sử dụng vector đặc trưng đại diện cho họ để phân loại. Điểm dữ liệu mới gần với vector đặc trưng của họ nào nhất sẽ được dự đoán thuộc họ đó. Quy trình phương pháp được trình bày trong Hình 3 và được diễn giải như sau:



Hình 3. Quy trình phương pháp sử dụng điểm đặc trưng - Dendroid

1) *Giai đoạn mô hình hóa*

Bước 1: Trích rút các cấu trúc mã lệnh

- *Đầu vào:* Tập các mẫu mã độc huấn luyện (được gán nhãn theo họ)
- *Đầu ra:* Tập các cấu trúc mã lệnh theo từng mã độc

Đầu tiên tất cả mẫu dữ liệu huấn luyện của họ phần mềm độc hại được đưa vào giai đoạn trích rút các cấu trúc mã lệnh. Trong bước này phương pháp trích rút ra các cấu trúc mã lệnh của từng mẫu mã độc.

Bước 2: Mô hình hóa và trích rút đặc trưng

- *Đầu vào*: Tập các cấu trúc mã lệnh theo từng mã độc
- *Đầu ra*: Các vector đặc trưng đại diện cho các họ mã độc (không phải cho từng mã độc) và tập cấu trúc mã lệnh của các họ.

Trong giai đoạn này, từ các mẫu mã độc đơn lẻ thuộc cùng một họ, chúng ta tổng hợp các cấu trúc mã lệnh của cả họ để tính vector đặc trưng cho họ đó. Các thành phần của vector đặc trưng được tính theo công thức tf-idf trong khai phá văn bản và truy xuất thông tin áp dụng với các cấu trúc mã lệnh.

2) *Giai đoạn phân tích*

- *Đầu vào*: Các vector đặc trưng cho các họ mã độc
- *Đầu ra*: Cây phân cấp của các họ mã độc

Trong giai đoạn này sử dụng thuật toán phân cụm phân cấp và phân tích liên kết để tìm ra mối quan hệ giữa các họ mã độc.

3) *Giai đoạn phân loại*

Bước 1: Trích rút các cấu trúc mã lệnh (giống trong giai đoạn mô hình hóa) nhưng đầu vào chỉ là một mã độc cần phân loại

Bước 2: Trích rút đặc trưng

- *Đầu vào*:
  - Cấu trúc mã lệnh của mã độc cần phân loại
  - Tập các cấu trúc mã lệnh của các họ
- *Đầu ra:* Vector đặc trưng cho mẫu mã độc cần phân loại

Sử dụng độ đo tf-idf trong khai phá văn bản và truy xuất thông tin để tạo ra vector đặc trưng cho mẫu mã độc.

Bước 3: Phân loại

- *Đầu vào*:
  - Vector đặc trưng cho mẫu mã độc cần phân loại
  - Các vector đặc trưng đại diện cho các họ mã độc
- *Đầu ra*: Nhãn lớp dự đoán được

Trong bước này sử dụng thuật toán 1-NN để dự đoán nhãn lớp của mẫu mã độc mới.

b, Phương pháp được đề xuất bởi P. Shrestha và các cộng sự[3]

Phương pháp này cũng sử dụng điểm đặc trưng để đại diện cho các họ, mỗi họ mã độc sẽ được đại diện bởi một vector đặc trưng giống như phương pháp Dendroid.

1) Giai đoạn xây dựng điểm mẫu

- *Đầu vào:*Tập các mẫu mã độc huấn luyện
- *Đầu ra:* Các vector đặc trưng đại diện cho từng họ mã độc

Cũng tương tự như phương pháp Dendroid, chúng ta gộp tất cả những file mã độc thuộc cùng một họ, trích rút ra các chuỗi có thể in được (printable string) trong các file mã độc của cả họ, sau đó tính trọng số của các chuỗi bằng giá trị tf-idf và xây dựng vector đặc trưng đại diện cho họ với mỗi chiều là giá trị trọng số của chuỗi tương tứng. Trong phương pháp này, người ta xây dựng hai loại điểm đặc trưng. Điểm đặc trưng thứ nhất được tập hợp từ tất cả các chuỗi có thể có, điểm đặc trưng thứ hai được xây dựng từ những chuỗi nổi bật trong từng họ mã độc (k chuỗi có trọng số cao nhất), các chuỗi nổi bật trong các họ mã độc khác nhau có thể khác nhau.

2) Giai đoạn phân loại

- *Đầu vào:* Mã độc cần phân loại
- *Đầu ra:* Nhãn lớp dự đoán được của mã độc đó

Đầu tiên, người ta tính danh sách tf-idf của từng chuỗi có thể in được trong mẫu mã độc cần phân loại.

Với mỗi chuỗi xuất hiện trong điểm đặc trưng, chúng ta chọn các giá trị tf-idf của chuỗi trong danh sách trên để tạo ra một vector đặc trưng đại diện cho mã độc chưa biết. Nếu một chuỗi có trong điểm mẫu nhưng không có trong mã độc cần phân loại thì giá trị của chiều đó trong vector đặc trưng tương ứng bằng 0. Mỗi điểm đặc trưng, chúng ta sẽ tìm được một vector khác nhau đại diện cho mã độc chưa biết. Cuối cùng, tính độ tương tự cosin giữa vector đó với vector đặc trưng. Mã độc được phân vào lớp của điểm đặc trưng mà nó có độ tương tự cao nhất.

Nhận xét các phương pháp dựa trên điểm đặc trưng: Dendroid [2] và phương pháp đề xuất bởi P. Shrestha và các cộng sự[3] đưa ra phương pháp huấn luyện khá khác biệt so với thông thường. Xây dựng mô hình từ tất cả các điểm dữ liệu huấn luyện cùng một lúc (xử lý theo lô) thay vì huấn luyện dần dần với mỗi dữ liệu tại một thời điểm (xử lý theo luồng). Ưu điểm của phương pháp này là vector đặc trưng được xây dựng từ tất cả các tập dữ liệu trong một họ vì vậy nó có tính đại diện riêng cho họ đó. Và mỗi vector đại diện cho một họ mã độc thay vì đại diện cho một mã độc như phương pháp thông thường giúp giảm thời gian xử lý trong quá trình phân loại. Nhược điểm của phương pháp này là chỉ sử dụng duy nhất một điểm để đại diện cho tất cả dữ liệu của một họ khi đó sẽ bị mất mát nhiều thông tin có thể làm cho quá trình phân loại không được chính xác.

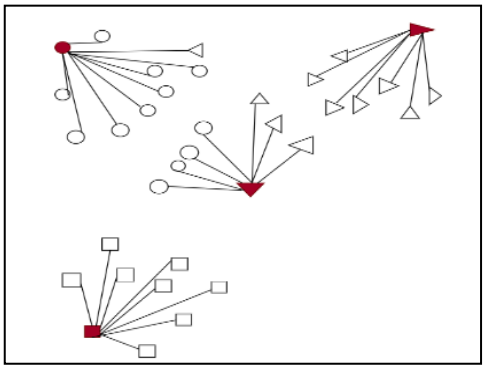
III. ĐỀ XUẤT PHƯƠNG PHÁP TRÍCH RÚT ĐIỂM MẪU CẢI TIẾN

Với phương pháp phân loại dựa trên điểm mẫu [1], các điểm mẫu được trích rút ra trên toàn bộ tập dữ liệu thì có thể có những sai sót vì khi đó có những dữ liệu không cùng một họ có thể thuộc chung một cụm, hoặc điểm mẫu chưa chắc đã cùng lớp với đại đa số các điểm dữ liệu trong nhóm nhưng lại được dùng làm đại diện cho nhóm đó. Trong khi đó, ý tưởng của phương pháp phân loại dựa trên điểm đặc trưng là lấy ra điểm đặc trưng trong tập dữ liệu thuộc cùng một lớp. Vì vậy, chúng tôi muốn theo tư tưởng của phương pháp Dendroid [2] và phương pháp [3] áp dụng và cải tiến phương pháp dựa trên điểm mẫu [1]. Chúng tôi muốn lấy ra những điểm mẫu từ những điểm trong cùng một họ mã độc. Khi đó một điểm mẫu chỉ đại diện cho một họ mà nó thuộc vào, không đại diện cho họ khác. Từ đó sẽ khắc phục được nhược điểm của phương pháp dựa trên điểm mẫu [1] nói trên. Ngoài ra sau quá trình trích rút chúng ta sẽ thu được một hay nhiều điểm mẫu để đại diện cho một họ mã độc, khác với phương pháp Dendroid [2] và phương pháp được đề xuất bởi P. Shrestha [3] là với mỗi họ chỉ trích rút được một điểm đại diện, giúp giảm sự mất mát thông tin.

Sau đây là hình minh họa điểm mẫu, điểm đặc trưng của các phương pháp để phân biệt được sự khác nhau của các phương pháp dựa trên điểm mẫu [1], các phương pháp dựa trên điểm đặc trưng Dendroid [2], phương pháp [3] và phương pháp cải tiến.

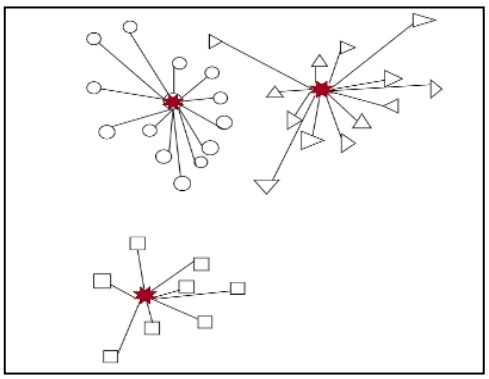
Chú thích: Δ: lớp 1,O: lớp 2, □: lớp 3

Các điểm được tô đậm là những điểm mẫu hoặc điểm đặc trưng của các lớp trong tập dữ liệu huấn luyện.



Hình 4. Minh họa điểm mẫu của phương pháp trích rút điểm mẫu [1]

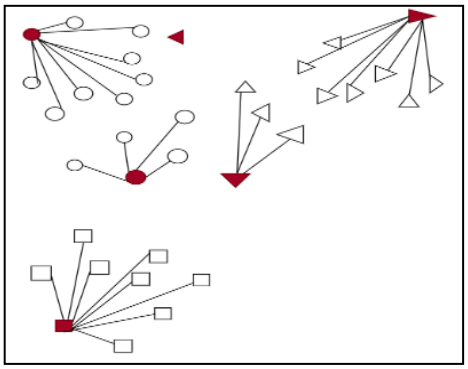
Hình 4 mô tả trường hợp có thể xảy ra là điểm thuộc lớp Δ có thể bị phân vào nhóm có điểm mẫu là lớp O, hoặc điểm thuộc lớp Δ là đại diện cho nhiều điểm thuộc lớp O.



Hình 5. Minh họa điểm đặc trưng của phương pháp Dendroid và phương pháp đề xuất bởi P. Shrestha và các cộng sự[3]

Chú thích: ✷ là điểm đặc trưng cho 1 lớp.

Từ Hình 5, một điểm đặc trưng được tổng hợp một điểm đặc trưng được tổng hợp từ tất cả các điểm dữ liệu trong một lớp. Điểm đặc trưng đó có thể là một điểm đã tồn tại hoặc một điểm ảo không tồn tại trong các điểm dữ liệu đã biết. Và một lớp chỉ có một điểm đặc trưng đại diện cho nó.



Hình 6. Minh họa điểm mẫu của phương pháp cải tiến

Hình 6 cho thấy tất cả các điểm mẫu đại diện của các lớp ☐ là điểm dữ liệu thuộc lớp ☐. Các điểm mẫu của lớp O là điểm dữ liệu thuộc lớp O và các điểm mẫu của lớp Δ là điểm dữ liệu thuộc lớp Δ. Mặc dù có một điểm Δ một mình, nó tự đại diện cho chính nó, không bị các điểm dữ liệu của lớp khác đại diện nhầm. Vì vậy trích rút đặc trưng trong phương pháp cải tiến luôn đạt trường hợp tốt nhất, tất cả các điểm trong cụm được đặc trưng bởi điểm mẫu thuộc chính lớp đó.

Từ tư tưởng trên, cải tiến của chúng tôi sẽ can thiệp vào giai đoạn trích rút điểm mẫu trong quá trình huấn luyện, còn quá trình dự đoán vẫn được thực hiện theo phương thức truyền thống sử dụng độ đo khoảng cách. Chúng tôi dựa trên quy trình tổng quát chung của phương pháp phân loại dựa trên điểm mẫu và bổ xung thêm cải tiến trong giai đoạn huấn luyện để thu được quy trình cải tiến được trình bày trong Hình 7.

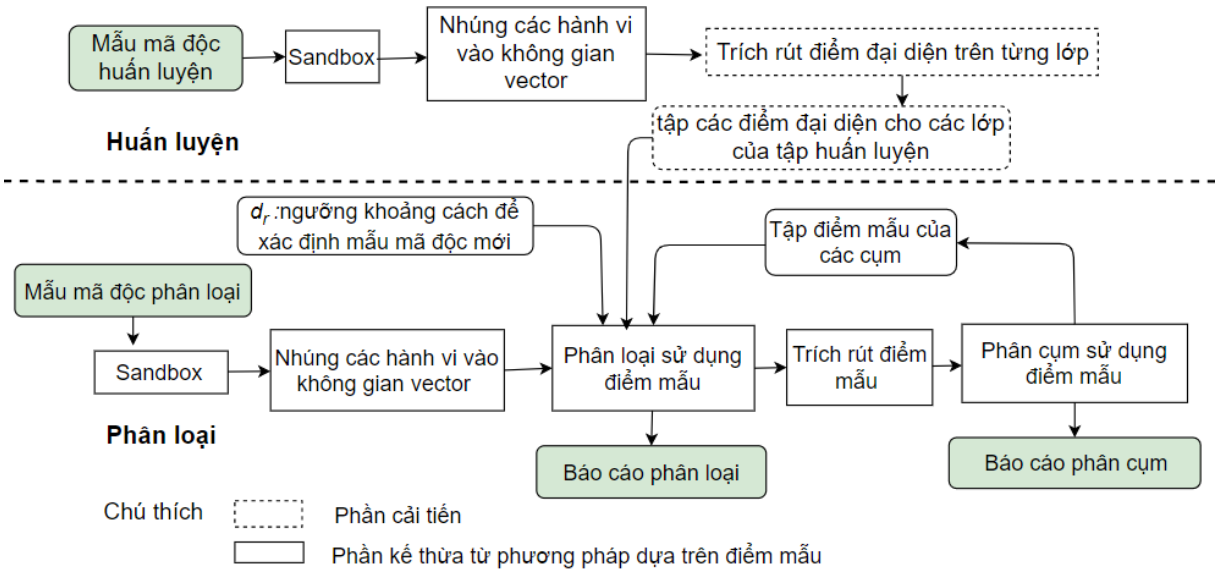
Quy trình gồm 2 giai đoạn là giai đoạn huấn luyện và giai đoạn dự đoán.

Trong giai đoạn huấn luyện:

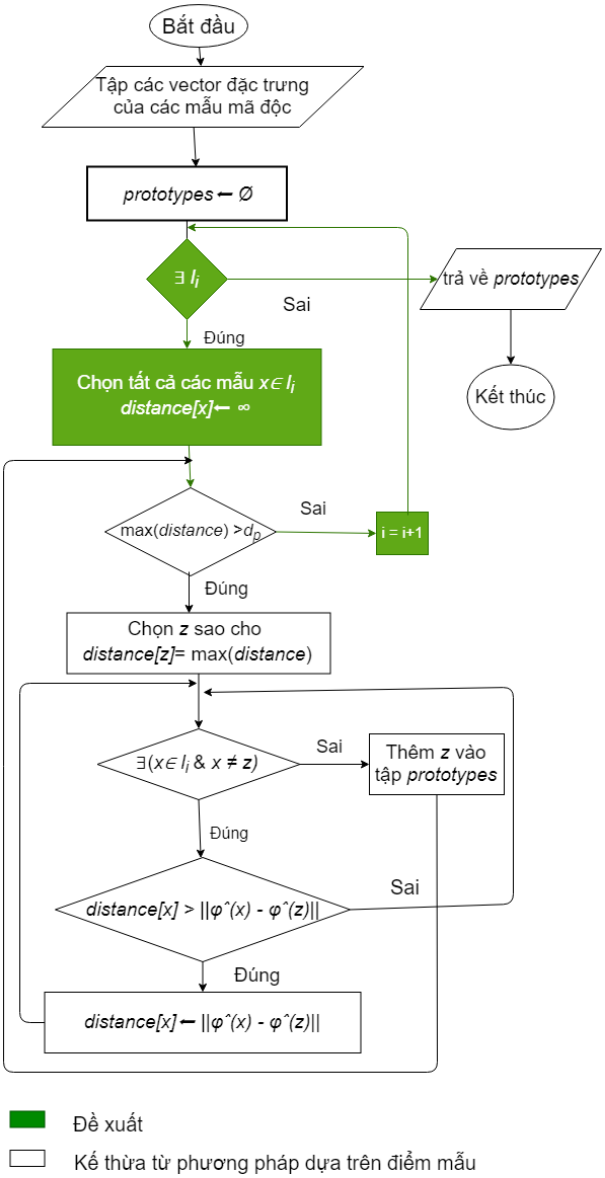
- Giữ nguyên giải thuật phân loại sử dụng điểm mẫu
- Thay đổi giải thuật trích rút điểm mẫu có áp dụng ý tưởng của phương pháp sử dụng điểm đặc trưng – Dendroid [2]. Chúng tôi sẽ trích rút các điểm mẫu trên tập dữ liệu của mỗi họ mã độc riêng biệt. Một họ mã độc chỉ được đại diện bởi một hay nhiều điểm dữ liệu thuộc họ mã độc đó. Giải thuật được trình bày trong sơ đồ khối ở Hình 8 và được giải thích bên dưới.
- Không sử dụng giải thuật phân cụm trong giai đoạn huấn luyện.

Trong giai đoạn dự đoán:

- Giữ nguyên giải thuật trích rút điểm mẫu tạo ra điểm mẫu trên tập dữ liệu chưa phân loại được để tìm ra các cụm mã độc mới.
- Sử dụng giải thuật phân cụm sử dụng các điểm mẫu được trích rút ở trên trong quá trình phân tích gia tăng
- Giải thuật phân cụm chỉ áp dụng trên các điểm mẫu được xây dựng từ dữ liệu chưa phân loại được vào các lớp đã biết, không áp dụng phân cụm sử dụng điểm mẫu cho các điểm mẫu của tập huấn luyện do các điểm mẫu này đã thuộc đúng các họ mã độc, không cần phân cụm nữa.



Hình 7. Quy trình cải tiến



Hình 8. Giải thuật đề xuất

Giải thích giải thuật đề xuất:

Bước 1: Khởi tạo tập *prototypes* = Ø, mảng *distance* có giá trị bằng ∞ để lưu khoảng cách của các điểm dữ liệu trong cùng một họ đến điểm mẫu gần nó nhất ở thời điểm hiện tại.

Bước 2: (cải tiến).

- *Bước 2.1:* Chọn tất cả các vector đặc trưng của một lớp để tiến hành trích rút điểm mẫu trên họ mã độc đó
- *Bước 2.2:* Kiểm tra khoảng cách lớn nhất trong mảng *distance* có nhỏ hơn d_p hay không, nếu không nghĩa là tất cả các điểm trong họ mã độc đã được một điểm mẫu đại diện cho chúng thì ta tiến hành thực hiện bước 2 với các họ mã độ còn lại, nếu có ta tiến hành tìm điểm mẫu mới.

Bước 3: Chọn điểm có khoảng cách lớn nhất đến các điểm mẫu tìm được trước đó làm điểm mẫu tiếp theo và thêm nó vào tập *prototypes*. Sau đó cập nhật lại khoảng cách của các điểm dữ liệu trong họ đó với điểm mẫu gần nhất. Lặp lại bước 2.2 đến khi điều kiện không thỏa mãn.

IV. THỰC NGHIỆM VÀ ĐÁNH GIÁ

A. *Chuẩn bị dữ liệu*

Chúng tôi sử dụng bộ dữ liệu reference dataset của phương pháp dựa trên điểm mẫu [1] cho công trình của mình. Tập dữ liệu này gồm toàn mã độc trích xuất từ cơ sở dữ liệu lớn về phần mềm độc hại được duy trì tại CWSandbox website và được gán nhãn bởi 6 sản phẩm Antivirus khác nhau và loại bỏ các lớp có ít hơn 20 mẫu và lấy trên một lớp tối đa 300 mẫu thực thi nhị phân. Phần mã nhị phân được thực thi và giám sát bằng CWSandbox thu được 3133 mẫu báo cáo hành vi thỏa mãn chuẩn MIST với 24 mẫu mã độc và 85 lời gọi hệ thống. Vì phương pháp của chúng tôi và phương pháp [1] đều tập trung vào phân loại mã độc thay vì phát hiện nên bộ dữ liệu được sử dụng chỉ chứa những mẫu mã độc, không có mã sạch.

	Malware class	#		Malware class	#
a	ADULTBROWSER	262	*m*	PORNDIALER	98
b	ALLAPLE*	300	*n*	RBOT	101
c	BANCOS	48	*o*	ROTATOR*	300
d	CASINO	140	*p*	SALITY	85
e	DORFDO	65	*q*	SPYGAMES	139
f	EJIK	168	*r*	SWIZZOR	78
g	FLYSTUDIO	33	*s*	VAPSUP	45
h	LDPINCH	43	*t*	VIKINGDLL	158
i	LOOPER	209	*u*	VIKINGDZ	68
j	MAGICCASINO	174	*v*	VIRUT	202
k	PODNUHA*	300	*w*	WOIKOINER	50
l	POSION	26	*x*	ZHELATIN	41

Hình 9. Mô tả tập dữ liệu

B. *Trích rút đặc trưng*

Trên tập dữ liệu, chúng tôi thực hiện trích xuất các chuỗi lời gọi hệ thống theo chuẩn MIST lever 1 (chỉ có tên của các lời gọi hệ thống, không có thông tin đối số) và thu được có 85 lời gọi hệ thống khác nhau trong tập dữ liệu. Sau khi thu được các chuỗi lời gọi hệ thống, chúng tôi tiến hành trích xuất vector theo 2-gram các lời gọi hệ thống (2 lời gọi hệ thống liên tiếp trong báo cáo). Sau khi thực nghiệm, chúng tôi thấy kết quả trên trích rút đặc trưng dựa trên sự xuất hiện của các lời gọi hệ thốngs đạt hiệu quả cao hơn trích rút đặc trưng dựa trên tần suất xuất hiện của các lời gọi hệ thống. Do đó, chúng tôi trích rút đặc trưng dựa trên sự xuất hiện hay không của các lời gọi hệ thống. Trong tập dữ liệu có 85 các lời gọi hệ thống khác nhau nên không gian của 1 vector là 85*85. Nhưng vì có nhiều chiều bằng 0 nên có thể khai thác để trích xuất đặc trưng và so sánh các vector trong thời gian tuyến tính. Thảo luận chi tiết của phương pháp thời gian tuyến tính cho trích xuất đặc trưng được cung cấp bởi Rieck và Laskov [10].

C. *Đánh giá và so sánh*

Chúng tôi đánh giá giai đoạn phân loại sử dụng điểm mẫu được trích rút theo phương pháp cải tiến của chúng tôi mà không đánh giá các giai đoạn trích rút điểm mẫu và phân cụm như trong bài báo [1] vì với giai đoạn trích rút điểm mẫu, theo phương pháp cải tiến thì

độ đo chính xác (precision) của các cụm luôn đạt giá trị tốt nhất là 1 do chúng tôi trích rút điểm mẫu từ dữ liệu trong cùng một cụm thay vì trên toàn bộ tập dữ liệu.

Do sử dụng bộ dữ liệu của phương pháp dựa trên điểm mẫu [1] nên chúng tôi sử dụng ngưỡng d_p (được trình bày trong giải thuật trích rút điểm mẫu ở Hình 2) bằng 0.65 (là ngưỡng tốt nhất để chọn ra các điểm mẫu đã được thực nghiệm và nêu ra trong bài báo [1]) để trích rút ra các điểm mẫu trong một lớp trong phương pháp cải tiến của chúng tôi.

Sau đó, chúng tôi tiến hành phân loại và đánh giá trên khả năng phân loại những lớp đã biết và khả năng nhận biết những lớp mới.

Để đánh giá khả năng phân lớp, chúng tôi sử dụng độ đo phân lớp $F1_{micro}$ là độ đo tổng hợp từ hai độ đo là độ đo chính xác P (precision) và độ hồi tưởng R (recall).

Định nghĩa các độ đo:

- TP_i: Số mẫu thuộc lớp i và được phân đúng vào lớp i
- FP_i: Số mẫu không thuộc lớp i nhưng bị phân sai vào lớp i
- TN_i: Số mẫu không thuộc lớp i và được phân đúng không thuộc lớp i
- FN_i: Số mẫu thuộc lớp i nhưng bị phân sai vào lớp không phải i.

Độ chính xác trung bình:

$$P^{\mu} = \frac{\sum TP_i}{\sum (TP_i + FP_i)} \quad (2)$$

Độ hồi tưởng trung bình:

$$R^{\mu} = \frac{\sum TP_i}{\sum (TP_i + FN_i)} \quad (3)$$

Độ đo $F1_{micro}$:

$$F^{\mu} = \frac{2 * P^{\mu} * R^{\mu}}{P^{\mu} + R^{\mu}} \quad (4)$$

$F1_{micro}$ nằm trong khoảng [0 ;1] và giá trị càng cao thì thể hiện độ phân lớp càng tốt.

Độ đo được chúng tôi lựa chọn để đánh giá là:

F_k: là $F1_{micro}$ trên tập dữ liệu đã biết nhãn để đánh giá khả năng phân lớp của thuật toán.

F_u: là $F1_{micro}$ trên tập dữ liệu chưa biết nhãn nhằm đánh giá khả năng nhận biết những lớp mới chưa xuất hiện trong tập huấn luyện.

Khả năng nhận biết các mã độc mới (F_u) và khả năng phân loại các mã độc đã biết (F_k) đều phụ thuộc vào việc chọn ngưỡng khoảng cách để xác định một mẫu mã độc mới (d_r). Nếu chúng ta chọn d_r càng lớn, độ loại mã độc ra càng thấp nên độ đo F_u càng thấp, còn độ đo F_k càng cao, ngược lại d_r càng nhỏ thì Fu càng cao và F_k càng thấp. Mục tiêu của chúng tôi là chọn ngưỡng d_r sao cho cả hai độ đo đều cao.

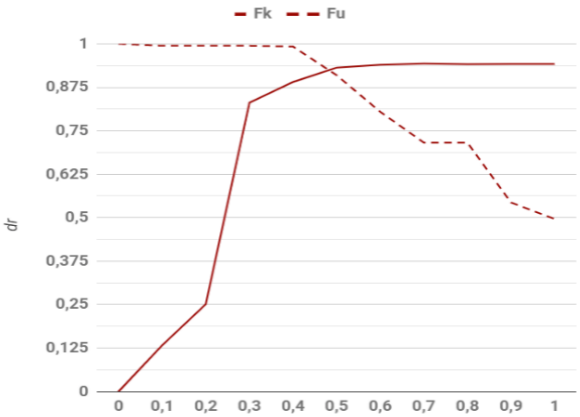
Tập dữ liệu thử nghiệm có 24 lớp. Chúng tôi sẽ sử dụng tập dữ liệu của 18 lớp để huấn luyện và đánh giá khả năng phân lớp với độ đo F_k. Dữ liệu của 6 lớp còn lại sẽ được đưa vào để đánh giá khả năng nhận biết lớp mới với độ đo F_u. Chúng tôi không chia tập dữ liệu trong bài báo [1], do sự xuất hiện của các mã độc mới thường ít hơn so với những mẫu mã độc đã biết nên chúng tôi đã chia như trên. Những thể hiện của 6 lớp để đánh giá khả năng nhận biết lớp mới chỉ được dùng để đánh giá không được cho vào giai đoạn huấn luyện. Còn tập 18 lớp chúng tôi sẽ chia tiếp theo tỷ lệ 70 :30 với 70% dùng để huấn luyện và 30% là dùng để đánh giá khả năng phân lớp. Chúng tôi thực hiện 10 lần và lấy kết quả trung bình với cả phương pháp cải tiến và phương pháp [1].

Kết quả của phương pháp dựa trên điểm mẫu [1] :

Bảng II. Kết quả phương pháp dựa trên điểm mẫu [1]

d_r	F_k	F_u
0.3	0.830	0.994
0.4	0.890	0.992
0.5	0.932	0.901
0.6	0.94	0.80
0.7	0.943	0.716

Do mục tiêu của chúng tôi là chọn ngưỡng dr sao cho cả hai độ đo Fk, Fu đạt giá trị cao. Với d_r = 0.7 thì độ đo F_u giảm mạnh trong khi Fk chỉ tăng được ít nên chúng tôi dừng thực nghiệm ở ngưỡng d_r = 0.7. Từ bảng II, ta thấy giá trị d_r để cả F_k và F_u đều đạt giá trị cao là d_r = 0.5, F_k = *0.932*, F_u = *0.901*.



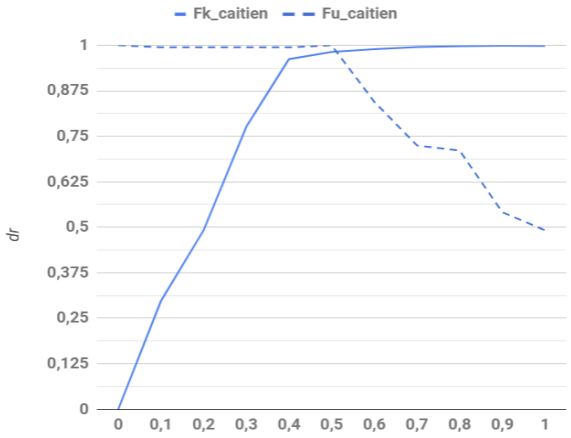
Hình 10. Biểu đồ kết quả phương pháp [1]

Kết quả phương pháp đề xuất

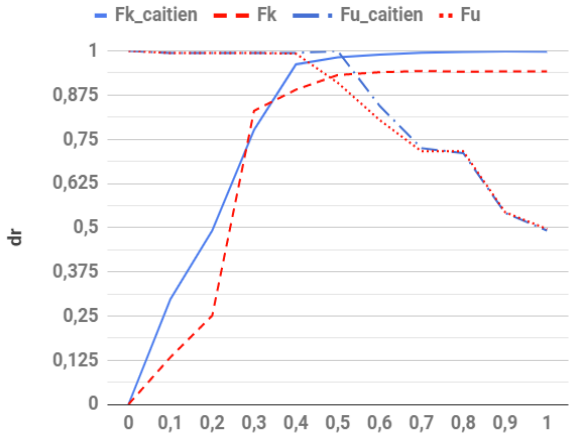
Bảng III. Kết quả phương pháp cải tiến

d_r	F_k	F_u
0.3	0.777	0.999
0.4	0.962	0.994
0.5	0.981	0.988
0.6	0.989	0.854
0.7	0.995	0.728

Tương tự, chúng tôi dừng ở ngưỡng d_r = 0.7 do độ đo F_u giảm mạnh.Từ bảng III, trường hợp tốt nhất với d_r = 0.5, F_k = 0.981, F_u = 0.988 vì cả hai F_k và F_u đều cao.



Hình 11. Biểu đồ kết quả phương pháp đề xuất



Hình 12. Biểu đồ so sánh kết quả hai phương pháp

So sánh hai trường hợp tốt nhất, ta thấy giải thuật cải tiến cho kết quả tốt hơn với $F_k = 0.981$, $F_u = 0.988$ (tương đương $F_k = 98.1\%$, $F_u = 98.8\%$).

V. KẾT LUẬN

Bài báo cáo trên của chúng tôi trình bày về một cải tiến trong giai đoạn trích rút điểm mẫu của quy trình phân loại mở mã độc dựa trên điểm mẫu được đề xuất bởi Rieck và các cộng sự vào năm 2011. Kết quả thực nghiệm cho thấy phương pháp cải tiến của chúng tôi cho hiệu quả khá tốt - đạt độ đo $F1_{micro}$ bằng 98.1% với khả năng phân loại và khả năng nhận biết mã độc mới là 98.8%, khắc phục được nhược điểm của phương pháp sử dụng điểm mẫu [1]. Tuy nhiên, phương pháp của chúng tôi vẫn còn hạn chế do chọn các điểm mẫu hoàn toàn theo khoảng cách có thể dẫn thiên lệch về hướng trong không gian nhiều chiều. Để khắc phục vấn đề này, chúng tôi sẽ nghiên cứu sử dụng mô hình siêu lập phương và chọn các điểm mẫu theo các hướng của siêu lập phương để hoàn thiện hơn quy trình cải tiến được đề xuất trong báo cáo này. Ngoài ra, vấn đề an ninh của các điện thoại thông minh cũng là một mối quan tâm lớn. Vì vậy, trong tương lai chúng tôi sẽ nghiên cứu và áp dụng phương pháp này cho các thiết bị điện thoại thông minh sử dụng hệ điều hành Android.

TÀI LIỆU THAM KHẢO

[1] "Automatic Analysis of Malware Behavior using Machine Learning" KonradRieck, Philipp Trinius, Carsten Willems, and Thorsten Holz Journal of ComputerSecurity (JCS), 19 (4) 639-668, 2011.

[2] Suarez-Tangil, Guillermo et al. "Dendroid: A text mining approach to analyzing and classifying code structures in Android malware families." Expert Syst. Appl. 41 (2014): 1104-1117.

[3] Prasha Shrestha,Suraj Maharajan,Gabriela Ramirez de la Rosa, Alan Sprague, Thamar Solorio and Gracy Warner, "Using String Information for Malware Family Identification" @Springer International Publishing Switzerland 2014,A.L.C.Bazzan and K.Pichara(Eds.): IBERAMIA 2014, LNAI 8864,pp.686- 697, 2014.DOI:10.1007/978-3-319-12027-0_55

[4] Souppaya, M., and Scarfone, K. Guide to Malware Incident Prevention and Handling for Desktops and Laptops. NIST Special Publication SP 800-83, July 2013.

[5] https://securelist.com/it-threat-evolution-q3-2018-statistics/88689/

[6] Daniele Ucci, Leonardo Aniello, Roberto Baldoni: Survey of machine learning techniques for malware analysis. Computers & Security 81: 123-147 (2019).

[7] Quinlan, J. Ross. "Combining Instance-Based and Model-Based Learning." *ICML* (1993).

[8] R. Duda, P.E.Hart, and D.G.Stork. Pattern classification. John Wiley & Sons, second edition, 2001.

[9] T. Gonzalez. Clustering to minimize the maximum intercluster distance. Theoretical Computer Science 38, pages 293–306, 1985.

[10] K. Rieck and P. Laskov. Linear-time computation of similarity measures for sequential data. Journal of Machine Learning Research, 9(Jan):23–48, 2008.