PHÁT HIỆN MÃ ĐỘC DỰA VÀO MÁY HỌC VÀ THÔNG TIN PE HEADER (PHẦN II)

Trần Ngọc Anh[1], Võ Khương Lĩnh[2]

[1]Bộ Tư Lệnh 86, [2]Đại học Nguyễn Huệ

Trong phần trước (số 2(060) 2021), các tác giả đã tiến hành phân tích, khảo sát thống kê 55 đặc trưng từ cấu trúc PE Header của tập dữ liệu 5000 file thực thi EXE/DLL và đã trích chọn được 14 đặc trưng quan trọng. Phần này, các tác giả nghiên cứu thử nghiệm một số mô hình máy học tiêu biểu với tập đặc trưng gốc (55 đặc trưng) và tập đặc trưng rút gọn (14 đặc trưng) cho phát hiện mã độc. Trên cơ sở đánh giá, so sánh thời gian thực hiện và độ chính xác, đồng thời so sánh với một số kết quả nghiên cứu trước nhằm chỉ ra kết quả nghiên cứu của bài báo là có giá trị.

MỘT SỐ MÔ HÌNH

Qua phân tích PE Header (trong Phần I), ta thu được hai tập đặc trưng dùng cho huấn luyện và thử nghiệm: Tập 1 có 55 đặc trưng (trong Phần I) và Tập 2 có 14 đặc trưng (Bảng 1, Phần I). Từ đây ta có thể áp dụng một số mô hình máy học để phân lớp như sau:

- Mô hình phân lớp theo xác xuất NB (Naive Bayes - NB);

- Mô hình Mạng nơ-ron nhân tạo ANN (Artificial Neural Network - ANN);

- Mô hình Cây quyết định DT (Decision Tree - DT);

- Mô hình Rừng ngẫu nhiên RF (Random Forest - RF).

Mô hình phân lớp theo xác suất NB

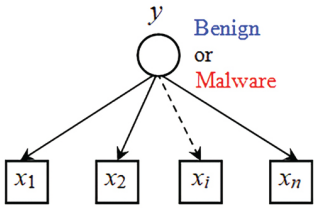
Mô hình NB còn được gọi là mô hình xác suất có điều kiện, là một phân bố xác suất $p(y|\vec{x})$ với vector đầu vào $\vec{x} = (x_1,...,x_n)$, trong đó x_i $(1 \le i \le n)$ là các đặc trưng và y là biến lớp cần được dự đoán. Xác suất đó được tính theo định lý Bayes:

$$p(y|\vec{x}) = \frac{p(y)p(\vec{x}|y)}{p(\vec{x})} \quad (1)$$

Bộ phân lớp NB sẽ thực hiện tìm lớp y_{NB} thích hợp nhất cho $\vec{x}$ bằng công thức sau:

$$y_{NB} = \arg\max_{y_i \in Y} p(y_i) \times \prod_{j=1}^{n} p(x_j|y_i) \quad (2)$$

Hình 1 minh họa mô hình NB với dãy quan sát đầu vào là tập đặc trưng $(x_1, x_2,...,x_n)$, kết quả thu được phân lớp đầu ra là $y_{NB} = y_1$ (Benign) hoặc $y_{NB} = y_2$ (Malware).



Hình 1. Mô hình NB phân lớp Benign hay Malware

Mô hình NB thường sử dụng một số hàm mật độ phân bố xác suất giả định như: phân bố Gaussian, phân bố đa thức (Multinomial), phân bố Bernoulli. Bài báo chọn phân bố Gaussian (3) để thử nghiệm.

$$p(x \mid y_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left[-\frac{(x-\mu_k)^2}{2\sigma_k^2}\right] \quad (3)$$

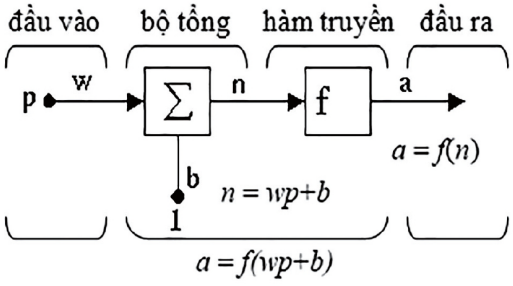
Trong đó: μ_k là trung bình của $x \in y_k$

σ_k^2 là phương sai của $x \in y_k$

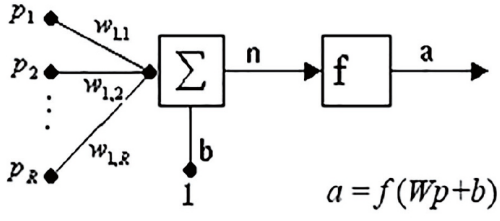
Mô hình mạng nơ-ron nhân tạo ANN

Mạng nơ-ron nhân tạo ANN là một mạng chứa các nơ-ron nhân tạo (mô phỏng nơ-ron sinh học) hoặc các nút (node) dùng để giải quyết các bài toán thuộc lĩnh vực trí tuệ nhân tạo (AI).

Mỗi nơ-ron (perceptron) được mô hình hoá như sau (Hình 2a, 2b):



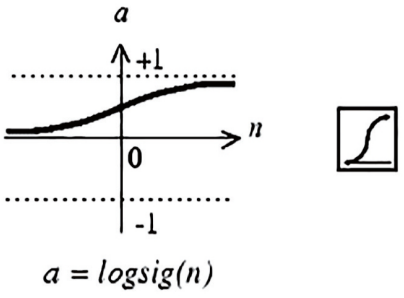
Hình 2a. Nơ-ron có 1 đầu vào



Hình 2b. Nơ-ron có R đầu vào

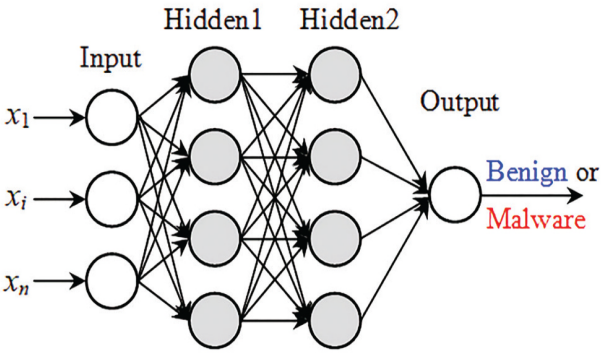
Các kết nối của nơ-ron được mô hình hóa bằng các trọng số *w*. Một trọng số dương (tích cực) phản ánh một kết nối kích thích (xúc động), ngược lại giá trị âm (tiêu cực) nghĩa là các kết nối ức chế (bị ngăn cản). Tất cả đầu vào được thay đổi bằng một giá trị trọng số và tính tổng. Hoạt động này được coi là một kết hợp tuyến tính. Ở đầu ra, sử dụng một hàm truyền, kích hoạt điều khiển biên độ. Một số hàm truyền *f*: Linear, LogSigmoid, TanSigmoid, Radial Basic,... Bài báo chọn hàm truyền *log-sigmoid* (4) để thử nghiệm.

$$a = logsig(n) = \frac{1}{1+e^{-n}} \quad (4)$$



Hình 2c. Đồ thị hàm truyền log-sigmoid

Hình 2d mô tả Mạng nơ-ron nhân tạo ANN đa lớp để phân lớp tập đặc trưng ($x_1, x_2, ..., x_n$) đầu vào và kết quả đầu ra là Benign hay Malware.

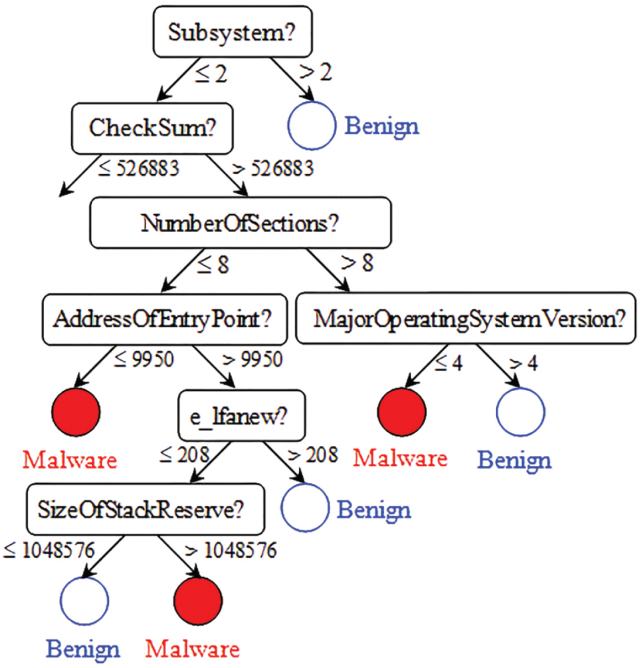


Hình 2d. Mô hình ANN phân lớp Benign hay Malware

Mô hình Cây quyết định DT

Cây quyết định DT là một trong những phương pháp máy học đơn giản thường được sử dụng cho bài toán phân lớp. Mục đích là tạo ra một mô hình dự đoán giá trị của một biến đích (kết quả phân lớp) dựa trên biến đầu vào (tập đặc trưng). Mỗi nút bên trong tương ứng với một trong các biến đầu vào (nút con), các cạnh cho nút con là giá trị của biến đầu vào. Giá trị của mỗi biến đích (nút lá) được đưa ra là kết quả phân lớp, được biểu diễn bởi đường đi từ gốc tới lá. Mỗi lá của cây

được gán nhãn như một lớp hoặc xác suất phân phối trên lớp.



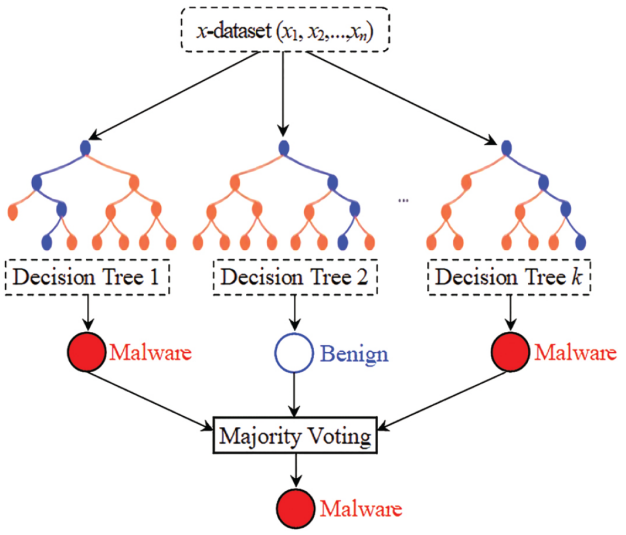
Hình 3. Cây quyết định phân lớp Benign hay Malware

Hình 3 mô tả một phần cây quyết định DT với đầu vào là tập đặc trưng rút gọn của PE Header, đầu ra là các nút lá được phân lớp là Benign hay Malware.

Mô hình Rừng ngẫu nhiên RF

Mô hình Rừng ngẫu nhiên RF là mô hình máy học có giám sát, có thể được sử dụng cho cả phân lớp và hồi quy một cách khá linh hoạt và dễ sử dụng. Khái niệm rừng (forest) nghĩa là gồm nhiều cây, càng có nhiều cây thì rừng càng mạnh. Với ý tưởng đó, mô hình RF sẽ tạo ra nhiều cây quyết định trên các mẫu dữ liệu được chọn ngẫu nhiên. Trên cơ sở kết quả dự đoán từ mỗi cây, RF chọn giải pháp tốt nhất bằng phương pháp bầu chọn (majority voting).

Trên Hình 4 minh hoạ mô hình RF phân lớp tập đặc trưng $(x_1, x_2,...,x_n)$ theo 2 giai đoạn: (1) thực hiện phân lớp dựa vào các cây quyết định được tạo ngẫu nhiên theo đặc trưng; (2) kết quả đầu ra của các cây quyết định ngẫu nhiên sẽ được bầu chọn theo đa số để có kết quả tốt nhất là Benign hay Malware.



Hình 4. Mô hình RF phân lớp Benign hay Malware

THỬ NGHIỆM VÀ ĐÁNH GIÁ

Dữ liệu cho huấn luyện và thử nghiệm

Với 5000 mẫu dữ liệu, trong đó 2500 mẫu Malware và 2500 mẫu Benign, tiến hành phân chia cho huấn luyện và thử nghiệm như sau:

Tỉ lệ 70% cho huấn luyện và 30% cho thử nghiệm được thực hiện: lấy ngẫu nhiên 30% bản ghi từ file sạch và file mã độc rồi ghép thành file thử nghiệm *TEST.CLASS.CSV*, các bản ghi còn lại được ghép thành file *TRAIN.CLASS.CSV* dùng cho huấn luyện.

Thử nghiệm và đánh giá

Đánh giá độ chính xác

a) Kí hiệu:

+ N_m là số bản ghi mẫu dùng cho đánh giá.

+ N_c là số bản ghi được mô hình gán nhãn.

+ N_d là số bản ghi được gán đúng so với mẫu.

Đánh giá độ chính xác kết quả gán nhãn bằng các mô hình máy học theo các công thức sau:

b) Đánh giá độ chính xác gán nhãn:

+ Độ hồi tưởng R (*Recall*):

$$R = N_d/N_m \quad (5)$$

+ Độ chính xác P (*Precision*):

$$P = N_d/N_c \quad (6)$$

+ Độ chính xác cân đối $F_{1\text{-}score}$:

$$F_{1-score} = \frac{2RP}{(R+P)} \quad (7)$$

Kết quả thử nghiệm

Bảng 1. Kết quả phân lớp với Tập 1 (55 đặc trưng)

Mô hình	Phân lớp	P (%)	R (%)	F_1 (%)
NB TGHL: 0,04s TGTN: 0,04s	Benign	95,65	93,73	94,68
	Malware	93,86	95,73	94,79
	trung bình	*94,73*	*94,73*	***94,73***
ANN TGHL: 329s TGTN: 0,10s	Benign	94,32	93,07	93,69
	Malware	93,16	94,40	93,73
	trung bình	*93,73*	*93,73*	***93,73***
DT TGHL: 0,16s TGTN: 0,01s	Benign	98,09	96,00	97,04
	Malware	96,08	98,13	97,10
	trung bình	*97,07*	*97,07*	***97,07***
RF TGHL: 0,71s TGTN: 0,06s	Benign	99,06	98,00	98,53
	Malware	98,02	99,07	98,54
	trung bình	*98,53*	*98,53*	***98,53***

Bảng 2. Kết quả phân lớp với Tập 2 (14 đặc trưng)

Mô hình	Phân lớp	P (%)	R (%)	F_1 (%)
NB TGHL: 0,01s TGTN: 0,01s	Benign	97,04	91,73	94,31
	Malware	92,16	97,20	94,61
	trung bình	*94,47*	*94,47*	***94,47***
ANN TGHL: 88s TGTN: 0,04s	Benign	96,61	94,93	95,76
	Malware	95,02	96,67	95,84
	trung bình	*95,80*	*95,80*	***95,80***
DT TGHL: 0,05s TGTN: 0,00s	Benign	98,9	95,60	97,22
	Malware	95,74	98,93	97,31
	trung bình	*97,27*	*97,27*	***97,27***
RF TGHL: 0,48s TGTN: 0,03s	Benign	98,92	97,87	98,39
	Malware	97,89	98,93	98,41
	trung bình	*98,40*	*98,40*	***98,40***

Nhận xét:

Trong 4 mô hình máy học NB, ANN, DT, RF, thì mô hình Rừng ngẫu nhiên RF cho kết quả tốt hơn cả. Tuy nhiên về thời gian huấn luyện (TGHL) thì NB và DT tốt hơn, thời gian huấn luyện và thời gian thử nghiệm (TGTN) thì ANN là kém nhất. Trên cơ sở kết quả so sánh lựa chọn chung cả về tập đặc trưng cũng kết quả phân lớp của các mô hình máy học, chúng tôi chọn mô hình Rừng ngẫu nhiên RF để xây dựng ứng dụng phát hiện mã độc.

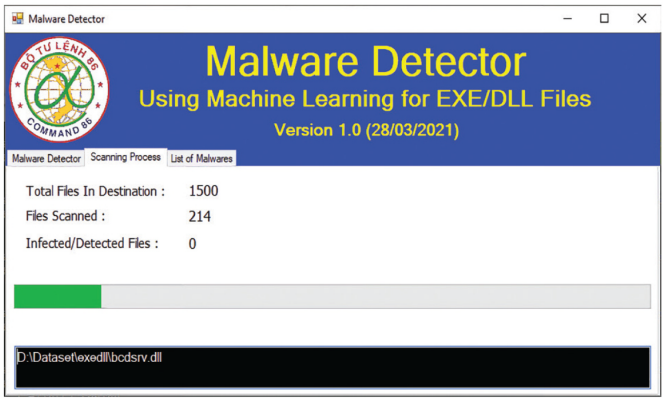
So sánh với các kết quả nghiên cứu khác

Bảng 3. So sánh với một số kết quả nghiên cứu khác

Phương pháp	Dữ liệu	Đặc trưng	F1 (%)
Schultz [9], 2001	4.266	Strings	97,11
Moskovitch [6], 2008	30.423	n-gram & TF.IDF	95,00
Tian [10], 2009	1.367	Strings	97,50
Wang [11], 2009	714	Các hàm API	93,71
Ye [12], 2010	50.000	Các hàm API	67,50
Belaoued [3], 2015	552	Optional Header	97,25
Belaoued [4], 2016	552	PE-Header & các hàm API	98,17
A. Hellal [1], 2016	2.083	Hành vi (Graph Mining)	92,00
BatErdene [2], 2017	650	Entropy	95,35
H. Xing [5], 2017	4.069	Mã Assembly (Graph)	97,00
Nahid [7], 2019	971	PE-Header&SectionHeader	98,26
Nguyen [8], 2020	11.425	PE-Header&SectionHeader	96,99
Bài báo này, 2021	**5.000**	**PE-Header (14 đặc trưng)**	**98,40**

Nhận xét Bảng 3:

Mặc dù tiếp cận của bài báo với số loại và số lượng đặc trưng là ít nhất so với [4], [7] và [8] nhưng vẫn cho kết quả tốt hơn chính là ở chỗ chọn những đặc trưng có giá trị phân loại tốt. Nếu bổ sung các đặc trưng Section Header, hàm API,... sẽ cải thiện làm tăng độ chính xác cho kết quả nghiên cứu.



Hình 5. Minh họa chương trình thử nghiệm

KẾT LUẬN

Bài báo đã nghiên cứu, đưa ra một tiếp cận mới trong việc phân tích khảo sát thống kê 55 đặc trưng từ cấu trúc PE Header của tập dữ liệu 5000 file thực thi EXE/DLL (gồm 2500 Benign và 2500 Malware) và đã trích chọn được tập 14 đặc trưng quan trọng có giá trị phân loại cao. Trên cơ sở đó đã nghiên cứu và thử nghiệm với 4 mô hình tiêu biểu NB, ANN, DT và RF. Qua đánh giá kết quả phân lớp của từng mô hình theo hai tập đặc trưng, kết quả cho thấy tập 14 đặc trưng rút gọn của bài báo là nổi trội cả về thời gian thực hiện cũng như độ chính xác so với tập 55 đặc trưng ban đầu. Kết quả nghiên cứu bài báo với độ chính xác $F_{1\text{-}score}$ là 98,40% cũng được so sánh với một số kết quả nghiên cứu khác, cho thấy hướng tiếp cận của bài báo là hiệu quả. Với những kết quả đó, cho phép xây dựng chương trình (Hình 5) dò quét phát hiện mã độc trên máy tính dựa vào một cách hiệu quả hơn so với các phần mềm antivirus chỉ dựa vào chữ ký.❖

TÀI LIỆU THAM KHẢO

1. A. Hellal, L. Romdhane (2016), "Minimal Contrast Frequent Pattern Mining for Malware Detection", Computers Security, Vol.62, pp.19-32.
2. Bat-Erdene, Munkhbayar and Park, Hyundo and Li, Hongzhe and Lee, Heejo and Choi, Mahnoo (2017), "Entropy Analysis to Classify Unknown Packing Algorithms for Malware Detection", International Journal of Information Security, Vol.16, pp.227-248.
3. Belaoued, M., Mazouzi S. (2015), "A Real-time PE-malware Detection System based on Chi-square Test and PE-file features", IFIP International Conference on Computer Science and its Applications, Springer, pp.416–425.
4. Belaoued M., Mazouzi S.(2016), "A Chi-Square-Based Decision for Real-Time Malware Detection Using PE-File Features", Journal of Information Processing Systems, Vol.12, No.4, pp.644-660.
5. H. Xing, Z. Li and Z. Jin (2017), "A Virus Detection Model Based on Artificial Immunity System", The 7th International Workshop on Computer Science and Engineering (WCSE 2017), Beijing, China.
6. Moskovitch R., Stopel D., Feher C., Nissim N., and Elovici Y. (2008), "Unknown Malcode Detection via Text Categorization and the Imbalance Problem", Proc. of 6th IEEE International Conference on Intelligence and Security Informatics, Taiwan, pp.156–161.
7. Nahid Maleki, Mehdi Bateni, Hamid Rastegari (2019), "An Improved Method for Packed Malware Detection using PE Header and Section Table Information", I. J. Computer Network and Information Security, pp.9-17.
8. Nguyen V.T., Hien V.T., Tuan L.D., Tiep M.V., Anh N.H., and Vuong P.T. (2020), "A Computer Virus Detection Method Based on Information from PE Structure of FilesCombined with Deep Learning Models", Proceedings of 7th International Conference, FDSE 2020, Quy Nhon, Vietnam, November 25–27, 2020, Springer, pp.120-129.
9. Schultz, M.G., Eskin, E., Zadok, E., Stolfo, S.J. (2001), "Data Mining Methods for Detection of New Malicious Executables", Proc. of the 2001 IEEE Symposium on Security and Privacy, pp.1-12.
10. Tian, R., Batten, L., Islam, R., and Versteeg, S. (2009), "An Automated Classification System based on the Strings of Trojan and Virus Families", In Proc. of 4th International Conference on Malicious and Unwanted Software, Montréal, Quebec, Canada, pp.23-30.
11. Wang, C., Pang, J., Zhao, R., and Liu, X. (2009), "Using API Sequence and Bayes Algorithm to Detect Suspicious Behavior", In Proceedings of International Conference on Communication Software and Networks, IEEE Computer Society, Washington, DC, USA, pp.544–548.
12. Ye, Y., Li, T., Jiang, Q., and Wang, Y. (2010), "CIMDS: Adapting Post Processing Techniques of Associative Classification for Malware Detection", IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, Vol.40, No.3, pp.298-307.