TẠP CHÍ KHOA HỌC & CÔNG NGHỆ ĐẠI HỌC DUY TÂN
DTU Journal of Science and Technology 5(48) (2021) 8-13

TRƯỜNG ĐẠI HỌC DUY TÂN
DUYTAN UNIVERSITY

Phát hiện malware dựa trên header của tập tin Portable Executable sử dụng Machine Learning

Malware detection based on Portable Executable file header using Machine Learning

Nguyễn Kim Tuấn[a,b*], Nguyễn Hoàng Hà[c], Trần Trương Thiện Nguyên[a,b]
Nguyen Kim Tuan[a,b*], Nguyen Hoang Ha[c], Tran Truong Thien Nguyen[a,b]

[a]*Khoa Công nghệ thông tin, Trường Khoa học máy tính, Đại học Duy Tân, Đà Nẵng, Việt Nam*
[a]*Faculty of Information Technology, School of Computer Sciences, Duy Tan University, 55000, Da Nang, Vietnam*
[b]*Viện Nghiên cứu và Phát triển Công nghệ Cao, Đại học Duy Tân, Đà Nẵng, Việt Nam*
[b]*Institute of Research and Development, Duy Tan University, Da Nang, 550000, Vietnam*
[c]*Trường Đại học Khoa học, Đại học Huế, Việt Nam*
[c]*University Sciences, Hue University, Vietnam*

(Ngày nhận bài: 18/5/2021, ngày phản biện xong: 02/6/2021, ngày chấp nhận đăng: 30/9/2021)

Tóm tắt

Trong bài báo này, chúng tôi dựa vào cấu trúc phần Portable Executable header của các tập tin Portable Executable để đề xuất một hướng tiếp cận khác trong việc sử dụng Machine learning để phân loại các tập tin này, là tập tin mã độc hay tập tin lành tính. Kết quả thực nghiệm cho thấy, tiếp cận đề xuất vẫn sử dụng thuật toán Random Forest cho bài toán phân loại nhưng độ chính xác và thời gian thực thi được cải thiện so với một số công bố gần đây (độ chính xác đạt 99.71%).

Từ khóa: Tập tin PE; Trường; Đặc trưng; Mã độc; Thuật toán Random Forest;

Abstract

In this paper, we rely on the Portable Executable header structure of Portable Executable files to propose another approach in using Machine learning to classify these files, as malware files or benign files. Experimental results show that the proposed approach still uses Random Forest algorithm for the classification problem but the accuracy and execution time are improved compared to some recent publications (accuracy reaches 99.71%).

Keywords: PE header, Field; Feature, Malware, Random Forest Algorithm.

1. Giới thiệu

Trong những năm trở lại đây, mã độc (malware) đã trở thành mối đe dọa đáng kể đối với vấn đề bảo mật trên không gian mạng. Malware có thể tồn tại trong các thiết bị đầu cuối, có thể truyền đi trên đường truyền mạng và có thể đính kèm/ẩn trong các tập tin có thể thực thi, đặc biệt là trong các tập tin Portable

[*]*Corresponding Author:* Nguyen Kim Tuan; Faculty of Information Technology, School of Computer Sciences, Duy Tan University, 55000, Da Nang; Institute of Research and Development, Duy Tan University, Da Nang, 550000, Vietnam
Email: nguyenkimtuan@duytan.edu.vn

Executable (PE) của hệ điều hành Windows. Hiện có 2 kỹ thuật được sử dụng để phát hiện malware [6]: i) Kỹ thuật dựa trên chữ ký (Signature based detection) tuy cho độ chính xác cao nhưng gặp nhiều khó khăn trước sự đa dạng và khả năng biến hình của các loại malware hiện nay; ii) Kỹ thuật không dựa trên chữ ký (Non-signature based detection) có thể giải quyết khó khăn này, nó thường được sử dụng để phát hiện được các loại malware "chưa được biết đến" (unknown), các loại malware có khả năng biến dạng cao xuất hiện gần đây… Kỹ thuật (ii) giúp việc phân loại, phát hiện malware hiện nay đạt hiệu quả cao khi được triển khai theo hướng tiếp cận Machine learning.

Tập tin PE là các tập tin hoạt động trên môi trường hệ điều hành Windows, nó có thể là các tập tin thực thi (executable files) hoặc là các tập tin chứa mã nhị phân được sử dụng bởi các tập tin thực thi khác. Vùng thông tin định dạng (format information) của tập tin PE [1] chứa những thông tin cần thiết mà hệ điều hành sử dụng để điều khiển việc thực thi của tập tin khi chúng được nạp vào main memory (bộ nhớ chính). Tất cả các tập tin PE đều có cùng cấu trúc và cùng số lượng field (trường) trong PE header, nên chúng ta có thể trích xuất các trường này, để làm tập đặc trưng (feature) đầu vào cho quá trình xây dựng mô hình phân loại malware của các tập tin này theo cách sử dụng các thuật toán Machine learning.

Chúng ta đều biết, thông tin chứa trong PE header của các tập tin PE lành tính (benign) đều ở dạng đã được chuẩn hóa bởi hệ điều hành Windows. Nếu một tập tin PE nào đó mà dữ liệu chứa trong các trường trong PE header của nó có sự "sai khác" so với các tập tin PE lành tính thì nhiều khả năng đó là tập tin malware. Như vậy, chúng ta có thể phân loại một tập tin PE, là tập tin malware hay tập tin lành tính, bằng cách xem xét dữ liệu chứa trong các trường của PE header của nó. Vì số lượng trường trong PE header là lớn, dữ liệu tại các trường lại có quan hệ với nhau, hầu hết các trường đều có thể bị làm "sai khác", ở những mức độ khác nhau, nên bài toán phát hiện malware ở đây cần tiếp cận theo hướng sử dụng các thuật toán Machine learning thì mới đạt được độ chính xác cao nhất có thể [1], [2], [4], [5], [6].

Chúng ta có thể thu thập một lượng lớn các mẫu PE header của các tập tin lành tính và tập tin malware, sau đó trích xuất các đặc trưng của mỗi trường, rồi so sánh để tìm ra sự khác biệt đáng kể nhất giữa tập tin lành tính và tập tin malware, làm cơ sở cho việc phân loại về sau. Đây là hướng tiếp cận mà chúng tôi thực nghiệm và đề xuất trong bài bào này.

2. Các nghiên cứu liên quan

Hiện có khá nhiều hướng tiếp cận cho bài toán phân loại malware sử dụng kỹ thuật Machine learning [7-9]. Trong phần này, chúng tôi điểm lại những kết quả, về độ chính xác, về tỉ lệ phát hiện và tốc độ huấn luyện, mà một số tiếp cận được công bố gần đây đạt được.

• Trong [5], Rushabh Vyah và cộng sự đã đề xuất một quy trình phát hiện malware trong tập tin PE trên môi trường mạng. Họ áp dụng 4 thuật toán học có giám sát khác nhau, Decision Tree, K-NN, SVMs và Random Forest, trên cùng một tập dữ liệu, chỉ với 28 đặc trưng (feature) tĩnh. Random Forest là mô hình mà Vyas chọn, nó đạt tỉ lệ phát hiện malware - backdoor, virus, trojan và worn - trung bình là 98.7%, tỷ lệ phát hiện dương tính là 1.8%.

• Tiếp cận được đề xuất bởi Hellal và Lotfi Ben Romdhane [2] là sự kết hợp giữa 2 kỹ thuật, phân tích tĩnh và khai phá đồ thị (static analysis – graph mining). Họ đề xuất một thuật toán mới có thể tự động trích xuất các mẫu hành vi malware có tính phổ biến và khác biệt, nhưng lặp lại, từ các tập tin nghi ngờ. Đề xuất này quan tâm đến việc tiết kiệm dung lượng bộ

nhớ và giảm thời gian quét bằng cách tạo ra một lượng chữ ký (signature) hạn chế, điều này không như các phương pháp hiện có. Tiếp cận trong [2] đạt tỷ lệ nhận dạng cao và tỷ lệ dương tính giả thấp với độ chính xác 92%.

• Jinrong Bai và cộng sự đề xuất một hướng tiếp cận cho việc phát hiện malware trong các tập tin PE bằng cách khai phá thông tin định dạng của các tập tin này [1]. Kỹ thuật "in-depth analysis" được nhóm tác giả chọn để phân tích vùng thông tin định dạng của các tập tin PE. Đầu tiên, họ cho trích xuất ra 197 đặc trưng từ vùng thông tin định dạng này, sau đó thực hiện việc chọn đặc trưng để giảm số lượng xuống còn 19 hoặc 20 đặc trưng. Tập đặc trưng được chọn sẽ được training bởi 4 thuật toán phân lớp J48, Random Forest, Bagging và Adaboost. Kết quả thực nghiệm cho thấy, tiếp cận này đạt độ chính xác cao nhất, 99.1%, ở thuật toán phân loại Random Forest.

• Yibin Liao khai thác cấu trúc của các tập tin PE theo một hướng tiếp cận khác [4]. Ông trích xuất đặc trưng của mỗi trường trong header, rồi so sánh để tìm ra sự khác biệt có ý nghĩa nhất giữa các tập tin malware và tập tin lành tính. Và trích xuất các icon trong tập tin PE để tìm ra các icon phổ biến (prevalent) nhất, có tính lừa bịp nhất (misleading) từ các tập tin malware. Yibin Liao thực nghiệm tiếp cận đề xuất trên một tập dữ liệu có 6875 mẫu, trong đó gồm 5598 mẫu header của tệp tin độc hại và 1237 mẫu header của tệp tin thực thi lành tính. Kết quả cho thấy tiếp cận này đạt tỷ lệ phát hiện hơn 99% với ít hơn 0,2% dương tính giả trong vòng chưa đầy 20 phút. Theo tác giả, có thể phát hiện malware bằng cách chỉ xem xét một vài đặc trưng/trường chính trong PE header của các tập tin PE hoặc xem xét các prevalent icon, các misleading icon được nhúng trong các tập tin này. Điều này giúp rút ngắn được thời gian phát hiện malware trên các tập tin PE.

Hiện chúng tôi chưa tìm thấy một phương pháp, một cách tiếp cận hay một mô hình được nào cho là chung nhất, là tối ưu nhất để phát hiện và phân loại malware sử dụng Machine learning đạt độ chính xác cao nhất. Vì thế, chúng tôi đề xuất một cách tiếp cận khác, đó là, tập trung vào các trường có ảnh hưởng cao trong phần PE header của các tập tin PE, như là sự đóng góp nhỏ cho hướng nghiên cứu này.

3. Tiếp cận đề xuất

Tiếp cận của chúng tôi được thực nghiệm trên tập dữ liệu (dataset) khá lớn, gồm 140.297 mẫu PE header của tập tin PE, trong đó có 44.214 mẫu malware và 96.083 mẫu lành tính. Dataset này được chúng tôi thu thập từ website virusshare.com và các tập tin PE lành tính trên môi trường hệ điều hành Windows.

Chúng tôi sử dụng các thuật toán Machine learning như: AdaBoost, Gradient Boosting, Decision Tree, Extra Tree, Random Forest, để xây dựng các mô hình phân loại tập tin PE - tập tin malware hay tập tin lành tính - từ dataset này theo hướng chỉ dựa vào đa số các trường trong phần PE header của các tập tin này. Mục tiêu thực nghiệm là để chọn ra một mô hình phân loại Machine learning sao cho có độ chính xác cao với thời gian huấn luyện chấp nhận được.

Với những thông tin có được từ việc khảo sát các trường trong phần PE Header của các tập tin này, chúng tôi tiến hành loại bỏ các trường ít bị tác động bởi malware nhất, như *LoaderFlags, NumberOfRvaAndSizes, SizeOfHeapCommit, SizeOfHeapReserve*… ra khỏi dataset, chỉ giữ lại 44 trường. Điều này hoàn toàn trùng hợp với kết quả mà chúng tôi có được khi sử dụng lần lượt thuật toán Random Forest và Extra Tree để đánh giá mức độ ảnh hưởng của các trường, chính xác là các feature, trong PE header của 140.297 mẫu PE header trong dataset. Bảng sau cho thấy mức độ ảnh hưởng của các trường theo Random Forest:

Bảng 1: Mức độ ảnh hưởng các trường trong PE header của các tập tin PE theo thuật toán Random Forest

TT	Trường trong PE header	Mức độ ảnh hưởng
1	ImageBase	0.193689
2	SizeOfStackReserve	0.103419
3	VersionInformationSize	0.075304
4	MinorImageVersion	0.065888
5	ResourcesMinSize	0.058338
6	Characteristics	0.052923
7	ExportNb	0.052831
8	Subsystem	0.049870
9	MajorOSVersion	0.045429
10	ResourcesNb	0.037733
…	…	…
41	SectionsMeanVirtualsize	0.001965
42	SectionMaxRawsize	0.001697
43	SectionsMeanRawsize	0.001697
44	ImportsNbOrdinal	0.001600
45	LoadConfigurationSize	0.001275
46	FileAlignment	0.001175
47	SectionAlignment	0.001167
48	SizeOfHeaders	0.001088
49	SizeOfUninitializedData	0.001036
50	BaseOfCode	0.000832
51	SizeOfHeapReserve	0.000401
52	SizeOfHeapCommit	0.000225
53	NumberOfRvaAndSizes	0.000008
54	LoaderFlags	0.000002
…	…	…

Việc giảm bớt một số trường của mỗi mẫu PE header không chỉ giúp làm giảm kích thước của dataset, dẫn đến giảm được tài nguyên của hệ thống dùng trong quá trình thực hiện chương trình xây dựng mô hình phân loại, mà còn giảm thời gian huấn luyện mô hình, với 54 feature là 13.04s, với 44 feature là 12.52s.

Phần còn lại trong hướng tiếp cận của chúng tôi được thực hiện theo đúng trình tự 4 thực nghiệm ở phần sau đây.

4. Kết quả thực nghiệm

4.1. Thực nghiệm 1

Chúng tôi chia ngẫu nhiên dataset thành 2 phần, 80% là tập huấn luyện (Training set) và 20% là tập kiểm thử (Test set). Hai tập dữ liệu này được sử dụng để đánh giá độ chính xác (accuracy) và thời gian huấn luyện (training time) của các mô hình Machine learning theo 5 thuật toán khác nhau. Kết quả nhận được cho ở Bảng 2.

Bảng 2. Độ chính xác và thời gian huấn luyện của của các mô hình Machine learning

Thuật toán	Độ chính xác	Thời gian huấn luyện
AdaBoost	99.12%	12.83s
GradientBoosting	99.30%	30.76s
DecisionTree	99.34%	0.98s
ExtraTree	99.69%	9.74s
RandomForest	99.71%	13.17s

Thực nghiệm này cho thấy, mô hình được xây dựng bởi thuật toán Random Forest cho độ chính xác cao nhất, lên đến hơn 99.71%, với thời gian huấn luyện đạt mức trung bình, so sánh với 4 thuật toán còn lại. Mô hình theo Extra Trees thì có thời gian huấn luyện nhanh hơn, nhưng độ chính xác thấp hơn so với Random Forest. Thuật toán Decision Tree cho mô hình có tốc độ huấn luyện rất cao, nhưng độ chính xác không như mong muốn.

4.2. Thực nghiệm 2

Cách chia tập dữ liệu thành 2 phần một cách ngẫu nhiên như ở thực nghiệm 1, tuy đơn giản nhưng độ chính xác của mô hình có thể bị ảnh hưởng nếu xảy ra hiện tượng overfit. Trong thực nghiệm này, chúng tôi sử dụng thuật toán k-fold [3], với K= 10, để giải quyết vấn đề overfit/unoverfit. Kết quả nhận được cho ở Bảng 3.

Bảng 3: Độ chính xác của các mô hình machine learning theo k-fold với K = 10

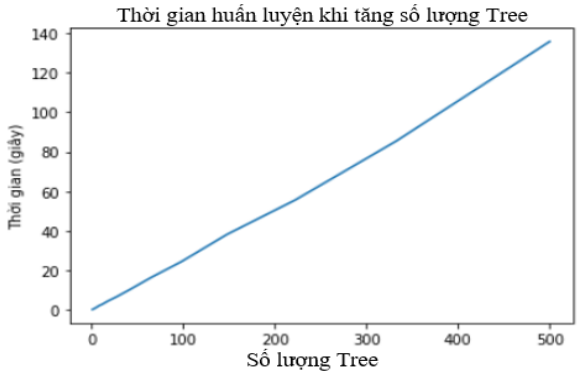
Thuật toán	Độ chính xác trung bình	Độ chính xác thấp nhất	Độ chính xác cao nhất
AdaBoost	99.11%	99.05%	99.17%
Gradient-Boosting	99.31%	99.24%	99.37%
Decision-Tree	99.34%	99.26%	99.42%
ExtraTree	99.71%	99.67%	99.75%
Random-Forest	99.72%	99.66%	99.76%

Từ kết quả có được ở thực nghiệm 1 và thực nghiệm 2, chúng tôi chọn thuật toán Random Forest để xây dựng mô hình phân loại cho đề xuất của mình, vì độ chính xác mà nó cung cấp là cao nhất (99.71% và 99.72%) và với thời gian huấn luyện hợp lý.

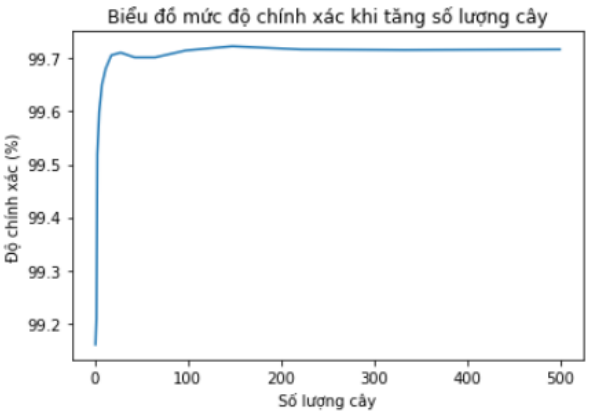
4.3. Thực nghiệm 3

Trong thực nghiệm này, chúng tôi sẽ tìm hiểu xem liệu tăng số lượng Tree trong mô hình theo Random Forest có làm cho độ chính xác tăng hay không, từ đó tìm ra số lượng Tree vừa đủ để mô hình có thể làm việc nhanh hơn với độ chính xác cao hơn.

Đầu tiên chúng tôi thử tạo ra 10 mô hình Random Forest chỉ có một tree sau đó tăng dần lên đến 500 tree, cứ mỗi lần tăng chúng tôi sẽ tính trung bình độ chính xác và thời gian huấn luyện của 10 mô hình. Kết quả cho ở 2 biểu đồ như hình bên dưới (H.1a và H.1b):



Hình 1a: Biểu đồ Thời gian huấn luyện của mô hình khi tăng số lượng Tree



Hình 1b: Biểu đồ Độ chính xác của mô hình khi tăng số lượng Tree

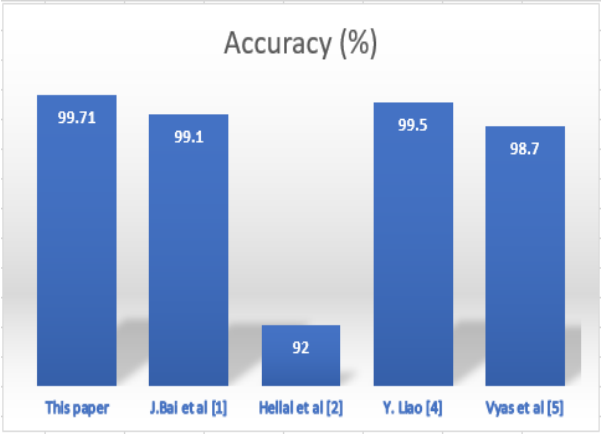
Độ chính xác khi số Tree ít hơn 20 là rất thấp, sau 50 bắt đầu tăng dần và độ chính xác bắt đầu đạt ngưỡng ở mức 100 Tree trở đi, thời gian huấn luyện tăng dần theo số lượng Tree. Điều này cho thấy, chúng ta chỉ cần một số lượng tree vừa đủ (trong trường hợp này là 100) thì mô hình cũng đã có thể đạt được độ chính xác cao. Giảm số lượng Tree giúp làm giảm thời gian huấn luyện và tiết kiệm được tài nguyên hệ thống. Đây là điều cần ghi nhận.

4.4. Thực nghiệm 4

Với việc chỉ chọn 44 đặc trưng, tương đương 44 trường trong PE header của các tập tin PE, mô hình phân loại Machine learning theo Ramdom Forest của chúng tôi có được tỉ lệ chính xác trung bình và thời gian huấn luyện lần lượt là 99.72% và 13.17s. Chúng tôi thực nghiệm việc tiếp tục giảm số lượng đặc trưng được chọn, để xem tỉ lệ chính xác và thời gian huấn luyện mô hình có bị thay đổi hay không. Kết quả như sau, khi số đặc trưng được chọn trong khoảng từ 13 đến 15 thì tỉ lệ chính xác trung bình đạt là 99.63% và thời gian huấn luyện là 3.88s.

Thực nghiệm này cho thấy, khi giảm số lượng đặc trưng đến mức có thể thì tỉ lệ chính xác trung bình chỉ giảm một lượng không đáng kể, 0.09%, nhưng độ giảm của thời gian huấn luyện giảm là đáng ghi nhận, 9.29s (70%), so với ban đầu. Việc giảm số lượng đặc trưng còn

giúp giảm kích thước tập dữ liệu, giảm thời gian cho việc phân tách các trường từ PE header của các tập tin PE, giúp tăng tốc độ phát hiện malware và tăng hiệu suất của hệ thống.



Hình 2: So sánh độ chính xác của đề xuất này so với một số công bố gần đây.

Như vậy, mô hình phân loại malware dựa vào E header của các tập tin PE theo hướng tiếp cận của chúng tôi đạt độ chính xác được ghi nhận so với một số công bố gần đây (H.2).

5. Kết luận

Qua bài báo này, chúng tôi đề xuất một hướng tiếp cận khác cho việc phát hiện malware trên các tập tin PE. Đề xuất của chúng tôi được thực nghiệm trên dataset rất lớn, gồm header của 149.297 tập tin PE, trong đó có 44.214 tập tin malware và 96.083 tập tin lành tính. Kết quả thực nghiệm cho thấy: Không cần xem xét tất cả các trường trong header, loại bỏ các trường ít ảnh hưởng nhất, thuật toán Random Forest vẫn cho độ chính xác khá cao, lên đến 99,71%, với thời gian huấn luyện đạt mức trung bình, 13.17s, so với 4 thuật toán khác; Độ chính xác của Random Forest phụ thuộc vào việc chọn số lượng Tree sao cho phù hợp chứ không cần phải chọn càng nhiều Tree càng tốt; Việc giảm số lượng Tree và việc loại bỏ các trường ít quan trọng đã cải thiện được tốc độ huấn luyện mô hình - giảm 70%, cải thiện tốc độ phát hiện malware và giảm tài nguyên hệ thống.

Tài liệu tham khảo

[1] J. Bai, J. Wang, G. Zou, (2014) "*A Malware Detection Scheme Based on Mining Format Information*", The Scientific World Journal, vol. 14, Article ID 260905, p. 1-11.

[2] A. Hellal, L. B. Romdhane, (2016) "*Minimal Contrast Frequent Pattern Mining for Malware Detection*", Computers & Security, vol. 62, p. 19-32.

[3] Davide Anguita, Luca Ghelardoni, Alessandro Ghio, Luca Oneto and Sandro Ridella, (2012) *"The 'K' in K-fold Cross Validation"*, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), p. 25-27.

[4] Y. Liao, (2012) "*Pe-Header-Based Malware Study and Detection*", Security & Privacy Workshop, San Francisco, CA, U.S.A.

[5] Vyas, R. Luo, X. McFarland, N. Justice, (2017) "*Investigation of malicious portable executable file detection on the network using supervised learning techniques*", IFIP/IEEE Symposium on Integrated Network and Service Management (IM), pp. 941–946.

[6] Ajit Kumara, K. S. Kuppusamya, G.Aghilab, (2019) *"A learning model to detect maliciousness of portable executable using integrated feature set"*, Journal of King Saud University - Computer and Information Sciences, vol. 31, iss. 2, p. 252-265.

[7] Puranik P. A., (2019) *"Static malware detection using deep neural networks on portable executables"*, *UNLV Theses, Dissertations, Professional Papers, and Capstones*. 3744.

[8] Kim, S., (2018) *"PE header analysis for malware detection"*, Master's Projects, San Jose State University.

[9] Azeez, N. A., Odufuwa, O. E., Misra, S., Oluranti, J., & Damaševičius. R, (2012) *"Windows PE Malware Detection Using Ensemble Learning"*, In: Informatics, vol. 8, no. 1, p. 10. Multidisciplinary Digital Publishing Institute.