Tạp chí Tin học và Điều khiển học, T.24, S.1 (2008), 32–41

CƠ CHẾ MÁY HỌC CHẨN ĐOÁN VIRUS MÁY TÍNH

HOÀNG KIẾM[1], TRƯƠNG MINH NHẬT QUANG[2]

[1] *Trường Đại học Công nghệ Thông tin, ĐHQG TP.HCM*
[2] *Trung tâm Đào tạo Đại học Tại chức Cần Thơ*

Abstract. When computer virus wide spreads in the world nowadays, anti-virus needs to improve their identifying methods to enhance the performance. In this paper, we introduce a new method to diagnose computer virus. First, we analyse the characteristics of viral data type to define virus classes through object-oriented methods. Second, we study the machine learnning mechanism for each virus class. Finally, we apply these learning forms to a data processing stage of a machine learning anti-virus expert system. The experimentation results show that the machine learning approach is suitable for anti-virus to identify the computer virus. This approach also gives a new aspect of anti-virus technology.

Tóm tắt. Trong bối cảnh các hệ thống máy tính thường xuyên bị virus tấn công, các hệ phòng chống virus máy tính cần cải tiến phương pháp nhận dạng và tăng cường hiệu quả chẩn đoán. Trong bài viết này chúng tôi giới thiệu phương pháp mới để chẩn đoán virus máy tính. Đầu tiên, virus máy tính được định nghĩa hướng đối tượng theo đặc trưng dữ liệu. Kế tiếp, xây dựng các mô hình học thích hợp cho từng lớp virus. Cuối cùng, áp dụng các bài toán học vào giai đoạn xử lý của một hệ phòng chống virus máy tính hướng tiếp cận máy học và hệ chuyên gia. Kết quả thực nghiệm cho thấy phương pháp này thích hợp cho bài toán nhận dạng virus máy tính, mở ra hướng nghiên cứu mới trong công nghệ anti-virus ngày nay.

1. GIỚI THIỆU

Internet là môi trường thuận lợi cho virus máy tính lây lan trên diện rộng. Mặc dù các hệ phòng chống virus (AV, Anti-virus software) không ngừng cập nhật và phát triển, các hệ thống máy tính vẫn thường xuyên bị virus xâm nhập, đánh cắp và phá hủy dữ liệu. Do đó cần nghiên cứu cải tiến cơ chế nhận dạng virus máy tính, bảo vệ an toàn dữ liệu cho các hệ thống công nghệ thông tin (CNTT). Khác với các phương pháp đã biết, chúng tôi giải quyết bài toán nhận dạng virus máy tính theo hướng tiếp cận máy học. Đầu tiên chúng tôi định nghĩa hướng đối tượng 5 lớp virus cơ bản A, B, C, D và E dựa vào đặc trưng dữ liệu của chúng. Sau đó xây dựng các bài toán học cho các lớp virus dựa vào các kỹ thuật học quy nạp, học chỉ dẫn, học vẹt, học tương tự và học tình huống. Để đánh giá tiếp cận, chúng tôi tích hợp 5 bài toán học vào hệ phòng chống virus máy tính hướng tiếp cận máy học và hệ chuyên gia MAV (Machine Learning Approach to Anti-virus Expert System). Kết quả thực nghiệm cho thấy hệ nhận dạng chính xác các virus đã cập nhật và dự đoán trên 91% biến thể virus mới.

2. TỔNG QUAN

2.1. Khái niệm về virus máy tính

Virus máy tính (computer virus, trong bài này gọi tắt là virus) là loại chương trình máy được thiết kế để thực hiện các chỉ thị của nó sau chương trình khác [1]. Bí mật sao chép bản thân nó vào các hệ thống máy tính, virus lây từ máy này sang máy khác, làm suy giảm năng lực hoạt động hệ thống và xâm phạm dữ liệu người dùng. Theo Bordera [2], virus máy tính là: "*bất cứ chỉ thị, thông tin, dữ liệu hoặc chương trình làm suy giảm tính hoàn thiện của tài nguyên máy tính, làm vô hiệu, gây nguy hiểm hoặc phá hủy, hoặc ghép bản thân nó vào tài nguyên của máy tính khác và thi hành khi chương trình máy tính thi hành*".

Chúng tôi phân loại virus dựa vào đặc trưng dữ liệu theo 5 lớp như sau:

- Lớp A (stand Alone program): các loại sâu trình có định dạng ứng dụng độc lập.
- Lớp B (Boot record): các loại virus lây vào mẫu tin khởi động hệ thống.
- Lớp C (asCii text): các loại virus, sâu trình có mã nguồn dạng script.
- Lớp D (Document): các loại macro virus lây vào tư liệu Microsoft Office.
- Lớp E (Executable): các loại virus lây vào các tập tin thi hành.

2.2. Tổng quan về bài toán nhận dạng và dự báo virus máy tính

Nhận dạng virus máy tính là quá trình tìm kiếm các mô tả đặc trưng virus trong thư viện mẫu trên tập chẩn đoán [3]. Năm 1995, Lo và cộng sự [4] giới thiệu phương pháp lọc mã độc dựa vào phân tích đặc trưng và thuộc tính. Phương pháp này có ưu điểm là đơn giản nhưng khả năng dự báo virus mới còn hạn chế. Năm 1996, IBM đề xuất phương pháp thống kê để trích chọn chuỗi nhận dạng tự động [5]. Do đầu ra chỉ là các chuỗi mã trích chọn nên chưa dự báo được đối tượng có phải là mã độc hay không. Năm 1998, Spafford giới thiệu phương pháp phân tích quá trình lây lan của sâu trình Internet trên cơ sở dữ liệu (CSDL) mã thực thi, cách lây và vị trí các nút mạng bị tấn công để dự báo các tình huống tương tự trên các nút khác [6]. Phương pháp này chạy chậm, chi phí cao, dễ quá tải khi mở rộng danh sách các nút mạng và sâu trình. Năm 2000, IBM sử dụng mô hình mạng trí tuệ nhân tạo ANN (Artificial Neural Networks) phân lớp các mẫu tin khởi động (MTKĐ). Kết quả nhận dạng được 80–85% các virus lạ với sai số dưới 1% trên các mẫu dương [7]. Tuy nhiên khi áp dụng ANN cho các đối tượng thi hành Win32, các chuyên gia IBM cũng chưa đưa ra được minh chứng thuyết phục nào cho hướng nghiên cứu này [8]. Năm 2001, G. Matthew và cộng sự công bố kết quả nhận dạng mã độc Win32 bằng kỹ thuật học quy nạp Find-S (đạt 87.35%) và phân lớp Nave Bayes (đạt 96.7%) [9]. Tuy nhiên do các thuật toán chuẩn hóa dữ liệu phức tạp, cần đến 1 GB bộ nhớ cho 4266 mẫu thử (3265 mã độc và 1001 ứng dụng) nhưng hoạt động kém hiệu quả trên các đối tượng chưa được phân lớp nên phương pháp này có hạn chế về mặt thực tiễn.

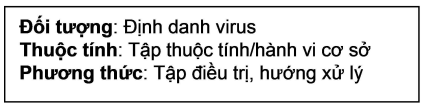
3. CƠ CHẾ MÁY HỌC CHẨN ĐOÁN VIRUS MÁY TÍNH

3.1. Tổ chức cơ sở tri thức

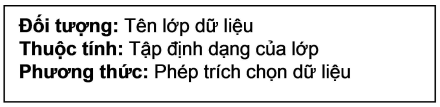
Máy học (machine learning) là lý thuyết xây dựng các hệ chương trình tự khám phá tri

thức bằng các cấu trúc dữ liệu và thuật giải đặc biệt, giúp phân tích, xử lý, trích chọn, chi tiết hóa dữ liệu và hỗ trợ quyết định liên quan đến kinh nghiệm của con người. Một số kỹ thuật học có thể sinh luật chuyên gia, thích hợp cho các trường hợp cần tham khảo ý kiến chuyên gia trong các lĩnh vực cụ thể và chuyên sâu [10]. Nguyên liệu dành cho các hệ học là cơ sở tri thức (CSTT, knowledge base), chứa các sự kiện mô tả dữ liệu và các luật nhận dạng.

Trong tiếp cận máy học, tri thức virus chứa thông tin về loại virus cần xử lý, các mô tả hành vi của virus trên đối tượng, các luật nhận dạng và dạng thức dữ liệu mà virus nhắm vào. MAV sử dụng mô hình lớp (class) chứa các virus có cùng đặc trưng dữ liệu. Mỗi lớp virus tương ứng với một lớp dữ liệu chẩn đoán [11] được định nghĩa hướng đối tượng như ở Hình 1.



Hình 1a. Biểu diễn tri thức virus



Hình 1b. Biểu diễn các lớp dữ liệu

Dạng tri thức thứ hai được mô tả trong CSTT là tập luật nhận dạng. MAV sử dụng một thư viện mô tả đặc trưng virus dưới dạng tập các vector $V_K = \{v_1, v_2, ..., v_k\}$ và áp dụng phép truy vấn các vectơ v_i trong tập dữ liệu S theo các luật dẫn xuất dạng: $R : p1 \wedge p2 \wedge ... \wedge p_n \rightarrow q$, trong đó p_i đặc trưng cho tập thuộc tính virus, q là kết luận của quá trình suy diễn.

3.2. Phân hoạch bài toán chẩn đoán virus máy tính

Dựa vào đặc trưng nhận dạng của các lớp dữ liệu, bài toán chẩn đoán virus máy tính được phân thành các bài toán con, sử dụng các kỹ thuật học từ đơn giản đến phức tạp như sau:

- Bài toán 1: chẩn đoán lớp C (asCii text files) theo cơ chế học vẹt.
- Bài toán 2: chẩn đoán lớp D (Document files) theo cơ chế học tương tự.
- Bài toán 3: chẩn đoán lớp B (Boot record) theo cơ chế học chỉ dẫn.
- Bài toán 4: chẩn đoán lớp E (Executable files) theo cơ chế học tình huống.
- Bài toán 5: chẩn đoán lớp A (stand Alone program) theo cơ chế học quy nạp.

Mỗi bài toán sử dụng cơ sở dữ liệu (CSDL) virus mẫu đặc thù tương ứng của lớp:

$$S = \{S_A, S_B, S_C, S_D, S_E\}$$

với S_A, S_B, S_C, S_D và S_E là CSDL virus mẫu của các lớp; $aObject, bObject, cObject, dObject$ và $eObject$ là các điểm dữ liệu trong không gian chẩn đoán của mỗi bài toán, theo thứ tự đó.

3.3. Các bài toán chẩn đoán virus máy tính

3.3.1. Bài toán 1: chẩn đoán lớp virus C-class

Virus lớp C lây nhiễm bằng cách chèn hoặc tạo mới câu lệnh script vào đối tượng. Gọi:

$T = \{a_i,\ c | i = 32, ..., 127;\ c \in N\}$ là đối tượng chẩn đoán.

$V = \{b_j,\ m | i = 32, ..., 127;\ n \in N\}$ là đối tượng lây nhiễm (virus).

trong đó a_i là tập ký tự của T, c là kích thước (số ký tự) của T, b_j là tập ký tự của virus V, m là kích thước của V và N là tập số nguyên dương. T nhiễm virus V khi và chỉ khi

$V \subseteq T$.

Gọi $S_C = \{V_1, V_2, ..., V_n\}$ là CSDL lớp C. Ứng với mỗi đối tượng chẩn đoán T, xác định:

• Trường hợp 1: $T \supset V_i \forall i = 1..n$, kết luận T nhiễm virus V_i (tức là $T = T_0 \cup V$):

- Xác định $T_0 = C_T(V_i) = T \backslash V_i \forall C_T(V_i)$ là phần bù của V_i trong T
- Loại bỏ virus: $V_i \leftarrow \{\phi\}$.

• Trường hợp 2: $T = V_i \forall i = 1..n$, kết luận đối tượng T là sâu trình V_i. Do sâu trình không có vật chủ ($T_0 = \{\phi\}$) nên thực hiện $V_i \leftarrow \{\phi\}$.

Bản chất của bài toán chẩn đoán C-class là học vẹt. Tri thức virus được chuyên gia cung cấp dưới dạng <<Mẫu dữ liệu, Khẳng định virus>>. Thuật giải đơn giản, có độ phức tạp $O(n)$ tỷ lệ với kích thước dữ liệu và số mẫu virus có trong S_C. Tuy nhiên thuật toán không đưa ra khẳng định dương khi có virus mới. Do virus text có tập lệnh hạn chế và ít phổ biến nên học vẹt là lựa chọn phù hợp trong giai đoạn hiện nay. Trong tương lai khi lượng virus text đủ lớn, có thể thay bằng các mô hình học dựa xác suất trên dữ liệu văn bản như Nave Bayes.

3.3.2. Bài toán 2: chẩn đoán lớp virus D-class

$D - class$ là lớp các virus macro sử dụng tập mã lệnh VBA (Visual Basic Application) để lây nhiễm trên môi trường MSOffice [12]. Khác với các macro thông thường thi hành nhờ lệnh Run, các virus macro tự thi hành bằng các thủ tục trigger (như AutoExec). Chỉ có các tư liệu nào sử dụng macro mới có nguy cơ chứa virus macro (Hình 2).

Trong mô hình học khám phá tương đồng, các hàm R nhận dạng có dạng:

$$(X_i = V_i) \wedge ... \wedge (X_k = V_k)$$

trong đó mỗi X_j là các biến, V_j là các giá trị có thể có của các biến này, các phép tuyển của những giá trị có thể có, hoặc tập của những giá trị này.

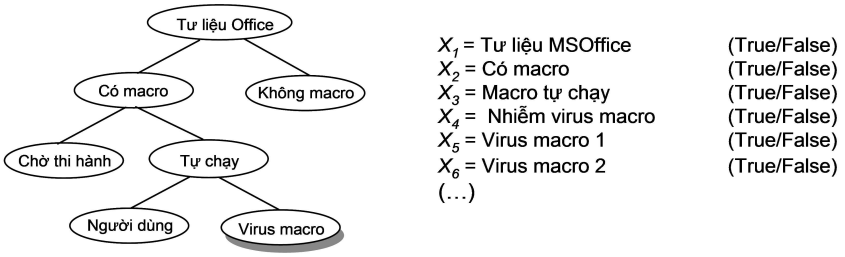
Một hàm R có trị TRUE đối với đối tượng chẩn đoán $dObject$ khi các giá trị của các biến của $dObject$ là một trong những hàm đó. Ngoài ra, hàm trả về trị FALSE. Trong không gian chẩn đoán N đối tượng, khi hàm R nhận dạng nhiều hơn một đối tượng, tập con của các giá trị mà nó nhận dạng gọi là được nhận dạng bởi R. Ngược lại, cho một tập con các đối tượng, ta có thể tạo một hàm nhận dạng được phát sinh bởi tập con này bằng cách lấy phép tuyển các giá trị của các biến của chúng [13].

Trong không gian SD, hệ sẽ xây dựng các hàm R cho mỗi đối tượng $dObject$. Nếu R nhận dạng được Vj (tương ứng với nút lá "Virus macro"), kết luận $dObject$ nhiễm virus đã biết:

$$R : (X_1 = true) \wedge (X_2 = true) \wedge (X_3 = true) \wedge (X_4 = true) \wedge (X_{4+i} = true)\ \forall i = 1..n.$$

Ngược lại, có thể kết luận $dObject$ nhiễm một loại virus macro mới.

Hình 3a và 3b mô tả các luật nhận dạng virus macro cũ và mới theo cơ chế học tương tự. Bài toán chẩn đoán $D - class$ có thể nhận dạng đến 98% các macro lạ (2% thất bại do password của người dùng). Tuy nhiên kỹ thuật này không phát hiện được các virus chen giữa các macro tự tạo. Hướng giải quyết là thiết lập bộ tinh chỉnh luật dưới dạng tùy chọn điều khiển trạng thái các mệnh đề "$dObject$ không có macro tự tạo" và "Đồng ý xóa macro."



Hình 2. Phân loại tư liệu MSOffice và các hàm R nhận dạng virus macro

Luật 1: IF *dObject* là tư liệu MSOffice
AND *dObject* có macro
AND macro thuộc loại tự chạy
THEN *dObject* là nguy hiểm
Luật 2: IF *dObject* là nguy hiểm
AND macro có tên là M[i]
THEN *dObject* nhiễm virus M[i]

Hình 3a. Luật nhận dạng virus macro

Luật 3: IF *dObject* là nguy hiểm
AND không có macro tự tạo
THEN *dObject* nhiễm virus mới
Luật 4: IF *dObject* nhiễm virus M[i]
OR *dObject* nhiễm virus mới
AND đồng ý xóa macro
THEN Loại trừ macro của *dObject*

Hình 3b. Luật nhận dạng virus macro mới

3.3.3. Bài toán 3: chẩn đoán lớp virus B-class

Lớp B chứa các boot virus lây vào các MTKĐ trên sector đầu tiên của tổ chức đĩa. Bài toán chẩn đoán $B - class$ được giải quyết theo hướng phân tích hành vi [14] như sau:

• Tổ chức 2 CSDL chứa các boot virus đã biết và các MTKĐ sạch phổ biến của các HĐH.

• Cung cấp 2 tập miền (domain theory) định nghĩa hành vi của boot virus và MTKĐ sạch. Ví dụ:

$Bootvirus \leftarrow GetMemSize, DecMemSize, SetMemSize, SetMemVi, MovViCode$

$GetMemSize \leftarrow ReadMem, GetValue$

$DecMemSize \leftarrow SetNewSize, WriteMem(...)$

• Tải *bObject* vào không gian tìm kiếm là một cây nhị phân có nút gốc đặc tả điểm vào lệnh. Nhánh biểu diễn các lệnh tuần tự. Nút con là các lệnh rẽ hướng và nhảy. Nút lá là các điểm dừng. Các lệnh lặp xử lý như lệnh tuần tự vào-ra trên cây con cục bộ (Hình 4).

• Áp dụng thuật giải tìm kiếm, thu thập các hành vi của *bObject* vào danh sách tác vụ:

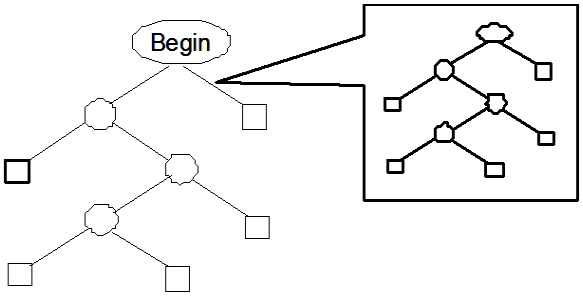
- Nếu danh sách phản ánh đầy đủ các mô tả của tập miền thứ nhất, thông báo tình trạng nhiễm virus của *bObject*, xử lý bệnh, báo cáo kết quả, kết thúc quá trình.

- Nếu danh sách phản ánh các mô tả của tập miền thứ hai, kết luận *bObject* an toàn.

- Ngoài ra, *bObject* có tình trạng bất thường (virus mới, sector hỏng, định dạng lạ...).

• Kết thúc quá trình, cập nhật thông tin đối tượng vào CSDL tương ứng.

So với mô hình mạng nơron [7], chẩn đoán boot virus theo cơ chế học chỉ dẫn có tốc độ nhanh (tương đương thời gian khởi động đĩa mềm trống) và chính xác hơn (nhận dạng 96% boot virus lạ) [15]. Tuy nhiên phương pháp này có nhược điểm là phức tạp trong cài đặt [16].



Hình 4. Cây chỉ thị nhị phân tìm kiếm

3.3.4. Bài toán 4: chẩn đoán lớp virus E-class

Lớp $E-class$ chứa các loại virus ghép mã vào tập thi hành [17]. MAV giải quyết bài toán này bằng mô hình AMKBD (Association Model of Knowledge Base and Database) [18]. Sử dụng CSDL (chứa thông tin đối tượng chẩn đoán) và CSTT (chứa tập luật nhận dạng virus), cơ chế suy luận chẩn đoán virus lớp E như sau:

- Đối với tập dữ liệu lạ, kiểm tra bệnh cũ, ghi nhận thông tin vào CSDL "hồ sơ bệnh án".
- Khi đã có thông tin, thường xuyên giám sát cộng đồng về mặt "vệ sinh dịch tể".
- Khi có cá thể lạ xuất hiện, kiểm tra đối tượng để hạn chế việc nhiễm bệnh từ bên ngoài.
- Khi có dịch virus, chỉ cần kiểm tra từng cá thể xem có mắc bệnh mới hay không.
- Khi phát hiện bệnh mới, phục hồi tình trạng cho cá thể từ CSDL hồ sơ bệnh án.

Để bảo vệ hệ thống trong thời gian thực, MAV sử dụng cơ chế đa tác tử (multi-agent mechanism) với hai tác tử. Tác tử Canh phòng (Autoprotect Agent) chạy thường trực ở mức nền sau (background) nhằm đón bắt các tình huống phát sinh trên đối tượng. Tác tử Duyệt quét (Scanning Agent) chạy ở mức nền trước (foreground) có nhiệm vụ duyệt tập dữ liệu. Cả hai tác tử sử dụng chung động cơ suy diễn, liên lạc nhau theo cơ chế truyền thông điệp [19]. 4.3.34. Trong điều kiện lý tưởng, phương pháp này có thể phát hiện đến 99% file virus lạ. Tuy nhiên khi AMKBD cảnh báo, hệ sẽ gây bối rối cho người dùng ít kinh nghiệm.

3.3.5. Bài toán 5: chẩn đoán lớp virus A-class

Lớp $A-class$ chứa các trojan horse/sâu trình như germs, dropper, injector, rootkit, intruder, zombie... Nhận dạng mã độc (malware) là vấn đề mở của các anti-virus hiện nay [20]. Nhiệm vụ của bài toán là kiểm tra đối tượng M có phải là mã độc hay không. Nếu không, hệ phải dự báo M có khả năng thuộc nhóm virus nào không, tỷ lệ mã độc là bao nhiêu.

Gọi $wRate \in (0,1]$ là tỷ lệ mã độc của M; $\lambda \in [0,1]$ là hằng số ngưỡng an toàn cho trước. Đầu tiên, tách CSDL A thành các nhóm f theo trật tự cha-con trên cấu trúc dữ liệu $V-tree$ [21]. Sau đó, áp dụng nguyên lý TF-IDF [22], biểu diễn M dưới dạng vectơ tần suất từ $F(M)$ sử dụng mô hình không gian vectơ, trong đó mỗi thành phần $F(M,w)$ đặc tả số lần từ w xuất hiện trong M. Tiếp theo, biểu diễn mỗi virus trong CSDL A dưới dạng vectơ tần suất từ $d_i = (w_{i1}, w_{i2}, ..., w_{iv})$, rồi ánh xạ các vectơ này vào ma trận 2 chiều từ - tài liệu (word-document matrix). Mỗi hàng ma trận tương ứng với bộ dữ liệu mẫu của virus đã "từ hóa" (to word), mỗi cột tương ứng với một từ duy nhất. Mục tiêu là xác định trọng số $W(f,w)$ trong từng tập f để tính độ đồng dạng dữ liệu (similarity measure) của M với các

tập f theo công thức:

$$SIM(M,f) = \frac{\sum\limits_{w\in M} F(M,w)W(f,w)}{\min(\sum\limits_{w\in M} F(M,w), \sum\limits_{w\in M} W(f,w))}.$$

Các đại lượng dùng tính toán SIM được định nghĩa trong Bảng 1.

Sau khi chọn được f (có SIM cao nhất), tính tỷ lệ mã độc của M so với các mẫu trong f:

$wRate_i(M,v_i) = FF(v_i,w)$ $\forall v_i$ là mẫu thứ i trong tập f.

Bảng 1. Các đại lượng tính toán theo nguyên lý TD-IDF

stt	Tên gọi	Thuật ngữ	Ý nghĩa	Ký hiệu/Công thức
1	Tần suất phân đoạn	Fractional frequency	Số lần xuất hiện của từ *w* trong tập *f* chia cho tổng số từ có trong *f*	$FF(f,w) = \frac{F(f,w)}{\sum_{w'\in f} F(f,w')}$
2	Tần suất từ	Term frequency	Tỷ số của tần suất phân đoạn của từ *w* trong nhóm *f* và số lần từ *w* xuất hiện trong A	$TF(f,w) = \frac{FF(f,w)}{FF(A,w)}$
3	Tần suất tài liệu	Document frequency	Thương của số nhóm *f* có từ *w* xuất hiện ít nhất 1 lần và tổng số các nhóm *f*	$DF(w)$
4	Trọng số	Weight	Trọng số của từ	$W(f,w) = \frac{TF(f,w)}{DF(w)^2}$

Cuối cùng, chọn mẫu có $wRatei$ lớn nhất. Nếu:

- $wRate = 1$, kết luận M là mã độc.
- $wRate \geq \lambda$, dự báo M chứa $(wRate \times 100)\%$ mã độc.

4. KẾT QUẢ THỰC NGHIỆM

4.1. Thử nghiệm tốc độ thực thi của MAV

Cùng với MAV, các AV thử nghiệm gồm Norton Anti-virus (NAV), Kaspersky Lab (KL) và Grisoft Anti-virus (AVG). Tập dữ liệu X có 36178 tập tin. Cách thực hiện như sau:

- Đo thời gian trung bình của các VirusFix (chỉ quét một virus) của mỗi AV.
- Đo thời gian chạy trung bình của mỗi AV hoàn chỉnh (có số virus xác định).
- Tính tốc độ quét trung bình của mỗi AV trong điều kiện chuẩn (ĐKC).

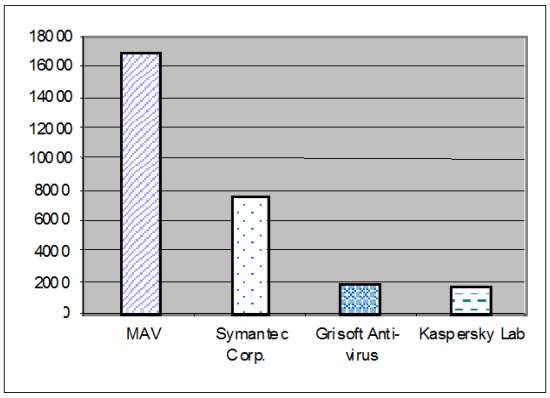
Đối với mỗi anti-virus thử nghiệm, gọi:

- V_c là số mẫu tin trong CSDL virus.
- T_0 là thời gian (giây) quét toàn bộ tập X trong trường hợp $V_c = 1$.
- T là thời gian (giây) quét toàn bộ tập X trong trường hợp $V_c > 1$.
- T_1 là thời gian trung bình (giây) chẩn đoán một virus trên tập $X : T_1 = T/V_c$.
- T_2 là thời gian trung bình (giây) chẩn đoán một mẫu tin trong CSDL: $T_2 = (T - T_0)/(V_c - 1)$
- V_e là số mẫu tin trong CSDL virus ở ĐKC.
- C_e là dung lượng (KB) dữ liệu trong ĐKC.
- T_e là thời gian (giây) chẩn đoán trong ĐKC: $T_e = T + (V_e - V_c) \times T_2$.
- S_e là tốc độ (KB/giây) đo được trong ĐKC: $S_e = C_e/T_e$.

ĐKC cho $V_e = 2.000; C_e = 10.000.000$ KB. Kết quả thực nghiệm trong Bảng 2 và Hình 5.

Bảng 2. Kết quả thử nghiệm tốc độ các AV trong điều kiện chuẩn

Anti-virus	$T_0(s)$	T_1 (s)	T_2 (s)	T (s)	T_e (s)	S_e (KB/s)
MAV	195	0.498	0.1987	324	592.245	16884.9
NAV	196	0.699	0.5748	1095	1345.038	7434.734
AVG	337	3.897	2.6259	1025	5586.188	1790.129
KL	390	5.918	2.7704	728	5928.041	1686.898



Hình 5. So sánh tốc độ các AV thử nghiệm trong điều kiện chuẩn

4.2. Thử nghiệm hiệu quả nhận dạng virus của MAV

Trong thử nghiệm này, các AV tham gia gồm NAV, VirusScan (McAfee) và Bit Defender. Không gian quan sát gồm 35178 tệp dữ liệu và 1000 mẫu virus. Kết quả MAV và BitDef phát hiện 957 và 959 virus, NAV và Scan là 907 và 906 virus (Bảng 3). Độ dự báo của các AV là tỷ số của số cảnh báo với hiệu của số virus thử nghiệm và số phát hiện chính xác:

$Proactivedetection = Proaction/(Viruses - Detections)$

Bảng 3. Kết quả thử nghiệm hiệu quả nhận dạng của các anti-virus

AV	Số virus	Phiên bản	Cảnh báo	Chính xác	Bỏ sót	Dự báo	Độ dự báo (%)
NAV	72020	9.05.15	907	889	93	18	16.22
Scan	N/A	4.0.4682	906	877	94	29	23.57
BitDef	253993	7.05450	959	925	41	34	45.33
MAV	890	N/A	957	483	43	474	91.68

Bảng 4. Hiệu quả dự báo virus lạ của MAV phụ thuộc vào hệ số λ

λ %	Dự báo	Tỷ lệ dự báo (%)	Nhầm	Tỷ lệ nhầm (%)	λ %	Dự báo	Tỷ lệ dự báo (%)	Nhầm	Tỷ lệ nhầm (%)
100	474	91.68	0	0	89	495	95.74	1	0.003
98	476	92.07	0	0	87	496	95.94	2	0.006
96	480	92.84	0	0	84	496	95.94	6	0.017
95	482	93.23	0	0	81	496	95.94	9	0.025
93	488	94.39	0	0	79	497	96.13	10	0.028
90	495	95.74	0	0	75	497	96.13	13	0.036

Khi giảm λ, độ dự báo của MAV tốt hơn nhưng cũng tăng rủi ro phát hiện nhầm (Bảng 4). Kết quả thử nghiệm cho thấy với CSDL khiêm tốn, MAV vẫn có thể phát hiện số virus tương đương với các phần mềm có số virus cập nhật nhiều hơn với tỷ lệ dự báo virus mới trên 91%. Khi $\lambda = 0,9$, tỷ lệ này là 95,74%, MAV sẽ đạt hiệu quả dự báo virus lạ tốt nhất.

5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Nhận định bản chất hoạt động của anti-virus và virus máy tính là cuộc đấu trí giữa các chuyên gia anti-virus và hacker, chúng tôi vận dụng các nguyên lý cơ bản của khoa học trí tuệ nhân tạo để xây dựng một hệ phòng chống virus máy tính hướng tiếp cận máy học. Áp dụng chiến thuật "chia để trị", bài toán nhận dạng virus máy tính được giải quyết từng phần bằng các bài toán học từ đơn giản đến phức tạp. Trong mỗi bài toán, các mô hình học được lựa chọn phù hợp với đặc điểm và tình hình lây nhiễm ở thế giới thực. Kết quả thực nghiệm chứng tỏ tiếp cận máy học khá thích hợp cho bài toán nhận dạng virus máy tính.

Sắp tới, chúng tôi sẽ áp dụng lý thuyết mờ để cải thiện độ dự báo bằng cách học các giá trị tích lũy của hằng số λ. Từ những kết quả bước đầu này, chúng tôi sẽ tiếp tục nghiên cứu các giải pháp kế thừa tri thức từ các hệ anti-virus khác, hướng đến mục tiêu phát triển MAV thành hệ tích hợp tri thức chuyên gia trong lĩnh vực nhận dạng thông minh virus máy tính.

TÀI LIỆU THAM KHẢO

[1] E. H. Spafford, Computer viruses as artificial life, *Journal of Artificial Life*, 1994.

[2] M. Bordera, "The Computer Virus War. Is The Legal System Fighting or Surrendering?" Computer and Law, University of Buffalo School of Law, 1997.

[3] Peter Szor, *The Art of Computer Virus Research and Defense,* Addison Wesley Professional, ISBN 0-321-30454-3, February 03, 2005.

[4] R. W. Lo, K. N. Levitt, R. A. Olsson, MCF: a malicious code filter, *Computer & Security* **14** (6) (1995) 541–566.

[5] Jeffrey O. Kephart and William C. Arnold, Automatic extraction of computer virus signatures, *Proceedings of the* 4^{th} *Virus Bulletin Conference*, Jersey - England, October 1994 (178–184).

[6] Eugene H. Spafford, "The Internet worm program: an analysis. Technical Report CSD-TR-823," Department of Computer Science, Purdue University, 1998.

[7] Gerald Tesauro, Jeffred O. Kephart, Gregory B. Sorkin, Neural networks for computer virus recognition, *IEEE Expert* **11** (4) (August 1996) 5–6.

[8] William Arnold, Gerald Tesauro, Automatically generated Win32 heuristic virus detection, *Proceedings of the 2000 International Virus Bulletin Conference*, Orlando-USA, September 2000.

[9] Matthew G. Schultz, Eleazar Eskin, Erez Zadok, Salvatore J. Stolfo, Data mining methods for detection of new malicious executables, *Proceedings of IEEE Symposium on Security and Privacy*, Oakland, CA. May 2001.

[10] Hoàng Kiếm, Đỗ Văn Nhơn, Đỗ Phúc, *Giáo trình các hệ cơ sở tri thức*, NXB ĐHQG Tp. Hồ Chí Minh, 2002.

[11] Hoang Kiem, Nguyen Thanh Thuy, Truong Minh Nhat Quang, A machine learning approach to anti-virus system, *Joint Workshop of Vietnamese Society of AI, SIGKBS-JSAI, ICS-IPSJ and IEICE-SIGAI on Active Mining*, Hanoi-VN, 4-7 Dec. 2004, (61–65).

[12] Vesselin Bontchev, Solving the VBA upconversion problem, *Virus Bulletin Conference*, Oxfordshire, England, 2000.

[13] Nguyễn Đình Thúc, *Trí tuệ nhân tạo - Máy học*, NXB Lao động Xã hội, 2002.

[14] Nguyễn Thanh Thủy, Trương Minh Nhật Quang, Các cơ chế chẩn đoán virus tin học thông minh dựa trên tri thức, *Tạp chí Tin học và iều khiển* **14** (2) (1998) 45–52.

[15] Nguyen Thanh Thuy, Truong Minh Nhat Quang, A global solution to anti-virus systems, *The Proceedings of the 1st International Conference on Advanced Communication Technology*, Muju-Korea, 10-12 February 1999 (374–377).

[16] Nguyễn Thanh Thủy, Trương Minh Nhật Quang, Máy ảo, công cụ hỗ trợ chẩn đoán và diệt virus tin học thông minh, *Tạp chí Tin học và iều khiển* **16** (2) (2000) 37–40.

[17] M. Pietrek, *Windows 95 System Programming Secrets*, IDG Books, 1995.

[18] Truong Minh Nhat Quang, Hoang Van Kiem, Nguyen Thanh Thuy, Association model of knowledge base and database in machine learning anti-virus system, *The Proceedings of the WMSCI 2006 Conference*, Florida-USA, July 2006 (277–282).

[19] Truong Minh Nhat Quang, Hoang Trong Nghia, A multi-agent mechanism in machine learning approach to anti-virus system, *The 2nd Symposium on Agents and Multi-Agent Systems, KES-AMSTA'08*, Korea. Springer LNAI, Vol. 4953, (743–752).

[20] Ian Waller, Controled worm replication - 'Internet-In-A-Box', *Virus Bulletin Conference*, Oxfordshire, England, 2000.

[21] Trương Minh Nhật Quang, Hoàng Kiếm, Nguyễn Thanh Thủy, Ứng dụng Máy học và Hệ chuyên gia trong phân loại và nhận dạng virus máy tính, *Tạp chí Công nghệ Thông tin và Truyền thông* (19) (2-2008) 93–101.

[22] J. A. Black, N. Ranjan, Automated event extraction from email, "Final Report of CS224N/ Ling237 Course in Stanford" (http://nlp.stanford.edu/courses/cs224n/2004).

Nhận bài ngày 15 - 10 - 2007
Nhận lại sau sửa ngày 14 - 1 -2008