**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**
**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**VŨ XUÂN HẠNH**

# NGHIÊN CỨU CÁC KỸ THUẬT PHÁT HIỆN DGA BOTNET

**LUẬN ÁN TIẾN SĨ KỸ THUẬT**

**HÀ NỘI - 2022**

**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**
**HỌC VIỆN CÔNG NGHỆ CƯU CHÍNH VIỄN THÔNG**

**VŨ XUÂN HẠNH**

# NGHIÊN CỨU CÁC KỸ THUẬT PHÁT HIỆN DGA BOTNET

**CHUYÊN NGÀNH: HỆ THỐNG THÔNG TIN**
**MÃ SỐ: 9.48.01.04**

**LUẬN ÁN TIẾN SĨ KỸ THUẬT**

**NGƯỜI HƯỚNG DẪN KHOA HỌC:**
**1.PGS.TS. HOÀNG XUÂN DẬU**
**2.TS. NGÔ QUỐC DŨNG**

**HÀ NỘI - 2022**

Công trình được hoàn thành tại:…….....

.....

Người hướng dẫn khoa học: 1. PGS.TS. Hoàng Xuân Dậu

2. TS. Ngô Quốc Dũng

Phản biện 1:……

……..

Phản biện 2:……

……..

Phản biện 3……

……

Luận án được bảo vệ trước Hội đồng chấm luận án cấp Học viện họp tại:……..

……

Vào hồi giờ ngày tháng năm

Có thể tìm hiểu luận án tại thư viện:……..

(*ghi tên các thư viện nộp luận án*)

# LỜI CAM ĐOAN

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi. Các kết quả viết chung với các tác giả khác đều được sự đồng ý của đồng tác giả trước khi đưa vào luận án. Các kết quả nêu trong luận án là trung thực và chưa từng được công bố trong các công trình nào khác.

**Tác giả**

**Vũ Xuân Hạnh**

# LỜI CẢM ƠN

Thực hiện luận án tiến sĩ là một thách thức rất lớn, một quá trình nghiên cứu đòi hỏi sự tập trung và kiên trì. Hoàn thành chương trình nghiên cứu sinh và được công bố những kết quả đạt trong quá trình nghiên cứu tôi thực sự thấy hạnh phúc. Đây không chỉ là nỗ lực cá nhân, mà còn là sự hỗ trợ và giúp đỡ nhiệt tình của các Thầy hướng dẫn, Học viện, bộ môn, các đơn vị hỗ trợ đào tạo, đồng nghiệp và gia đình. Tôi muốn bày tỏ sự biết ơn tới họ.

Trước hết, tôi xin gửi lời cảm ơn chân thành và sâu sắc tới PGS. TS. Hoàng Xuân Dậu và TS. Ngô Quốc Dũng đã quan tâm hướng dẫn và giúp đỡ tôi trong suốt quá trình thực hiện và hoàn thành luận án.

Tôi xin chân thành cảm ơn Lãnh đạo Học viện Công nghệ Bưu chính viễn thông, Khoa Công nghệ Thông tin 1, Khoa Quốc tế và Đào tạo Sau Đại học đã tạo điều kiện thuận lợi cho tôi trong thời gian nghiên cứu và hoàn thành luận án. Tôi cũng xin cảm ơn Lãnh đạo trường Đại học Mở Hà Nội và khoa Công nghệ Thông tin và đồng nghiệp đã hỗ trợ, động viên tôi trong quá trình nghiên cứu và thực hiện luận án.

Cuối cùng, tôi xin gửi lời cảm ơn vô hạn tới gia đình và bạn bè đã luôn ở bên cạnh, chia sẻ, động viên tôi những lúc khó khăn, hỗ trợ cả về vật chất lẫn tinh thần trong suốt quá trình nghiên cứu.

# MỤC LỤC

# DANH MỤC BẢNG BIỂU

# DANH MỤC HÌNH VẼ

# DANH MỤC CÔNG THỨC

# DANH MỤC CÁC TỪ VIẾT TẮT

| KÝ HIỆU | DIỄN GIẢI | |
|---|---|---|
| | TIẾNG ANH | TIẾNG VIỆT |
| ACC | Accuracy | Độ chính xác |
| AI | Artificial Intelligence | Trí tuệ nhân tạo |
| ANN | Artificial Neural Network | Mạng nơ ron nhân tạo |
| APT | Advanced Persistent Threat | Các cuộc tấn công có chủ đích |
| CNN | Convolutional Neural Netword | Mạng nơ ron tích hợp |
| CnC | Command and Control | Máy chủ lệnh và điều khiển |
| DGA | Domain Generation Algorithms | Thuật toán sinh tên miền |
| DL | Deep Learning | Học sâu |
| DNS | Domain Name System | Hệ thống phân giải tên miền |
| DDNS | Dynamic DNS | Hệ thống tên miền động |
| DDoS | Distributed Denial of Service | Tấn công từ chối dịch vụ phân tán |
| FNR | False Negative Rate | Tỷ lệ âm tính giả (bỏ sót) |
| FPR | False Positive Rate | Tỷ lệ dương tính giả (nhầm lẫn) |
| IDS | Intrusion Detection System | Hệ thống phát hiện xâm nhập |
| IPS | Intrusion Prevention System | Hệ thống phòng chống xâm nhập |
| ISP | Internet Service Provider | Nhà cung cấp dịch vụ Internet |
| IRC | Internet Relay Chat | Chat chuyển tiếp Internet |
| kNN | k Nearest Neighbor | k láng giềng gần nhất |
| LSTM | Long Short Term Memory | Bộ nhớ ngắn-dài hạn |
| ML | Machine Learning | Học máy |
| UDP | User Datagram Protocol | Giao thức dữ liệu người dùng |
| RF | Random Forest | Thuật toán Rừng ngẫu nhiên |
| SVM | Support Vector Machine | Thuật toán Máy hỗ trợ véc tơ |
| SLD | Second-Level Domain | Tên miền cấp 2 |
| TLD | Top-Level Domain | Tên miền cấp cao |
| TCP | Transmission Control Protocol | Giao thức điều khiển giao vận |
| TTL | Time to Live | Thời gian tồn tại |
| VPN | Virtual Private Network | Mạng riêng ảo |
| VPS | Virtual Private Server | Máy chủ riêng ảo |

# PHẦN MỞ ĐẦU

## 1. GIỚI THIỆU

*Bot* là một dạng phần mềm độc hại cho phép các nhóm kẻ tấn công, hay tin tặc kiểm soát từ xa các máy tính hoặc các hệ thống tính toán (*gọi chung là máy tính*) có kết nối Internet. Khi một máy tính bị lây nhiễm bot, nó được gọi là *máy tính ma*, hay *zombie*. Tập hợp các máy bot do một nhóm tin tặc kiểm soát (*botmaster*) được gọi là *botnet* - hay mạng của các bot. Botmaster thường điều khiển các bot trong botnet do mình kiểm soát thông qua hệ thống các máy chủ chỉ huy và kiểm soát (*Command and Control, hoặc C&C, hoặc CnC*). Khác với các phần mềm độc hại thông thường, các bot trong một botnet có khả năng tương tác với nhau và kết nối đến máy chủ CnC của botnet để nhận lệnh và mã cập nhật từ botmaster. Hơn nữa, các bot cũng được trang bị các kỹ thuật ẩn mình tiên tiến, như đóng gói, xáo trộn mã, mã hóa, nâng cấp, cập nhật mã nhị phân... giúp cho chúng có khả năng tồn tại lâu dài trên hệ thống nạn nhân. Quy mô của các botnet có thể rất khác nhau, từ hàng hàng chục ngàn đến hàng trăm ngàn bot phân tán ở mọi vị trí địa lý trên mạng Internet. Đặc biệt, một số botnet như Conficker theo ước tính có hơn 10.5 triệu bot [5].

Trong những năm gần đây, các botnet được xem là một trong những mối đe dọa an ninh chủ yếu đối với các hệ thống thông tin, các thiết bị có kết nối và người dùng Internet [38] [43] [86]. Điều này là do các botnet có liên hệ trực tiếp đến nhiều dạng tấn công và lạm dụng trên mạng Internet, như các cuộc tấn công từ chối dịch vụ (DDoS) trên qui mô lớn và rất lớn, gửi thư rác, truyền tải và phát tán các loại mã độc, sinh click và like ảo và đánh cắp các thông tin nhạy cảm. Chẳng hạn, mạng xã hội Telegram đã phải chịu một cuộc tấn công DDoS với qui mô rất lớn được cho là khởi phát từ Trung Quốc và có liên hệ với các cuộc biểu tình ở Hồng Kông vào năm 2019 [86]. Cũng trong năm 2019, một cuộc tấn công DDoS có qui mô lớn khác vào hệ thống dịch vụ lưu kết quả bầu cử Quốc hội của Phần Lan được ghi nhận [86]. Theo báo cáo của hãng bảo mật Symantec [90], khoảng 95% lượng email gửi trên mạng Internet vào năm 2010 là thư rác. Hơn nữa, các dạng tấn công nguy hiểm do botnet

hỗ trợ thực hiện còn bao gồm giả mạo địa chỉ URL, giả mạo hệ thống tên miền (*DNS*), tấn công chèn mã độc trên các ứng dụng web và thu thập các thông tin nhạy cảm từ người dùng. Các tổ chức tài chính và các cơ quan chính phủ thường là các mục tiêu chính của các dạng tấn công do botnet hỗ trợ thực hiện [38] [43] [86]. Một vấn đề khác khiến cho các mối đe dọa từ botnet các trở lên nghiêm trọng, khó bị phát hiện và loại bỏ là do trong quá trình phát triển của mình các botnet liên tục tiến hóa trên mạng Internet về cả qui mô và mức độ tinh vi của các kỹ thuật điều khiển [36].

Do tính chất nguy hiểm của botnet và các dạng mã độc mà botnet hỗ trợ truyền tải và phát tán, nhiều giải pháp đã được nghiên cứu, phát triển và triển khai trên thực tế cho giám sát, phát hiện và loại bỏ botnet. Có thể chia các giải pháp giám sát, phát hiện botnet thành 2 nhóm: (1) các giải pháp dựa trên honeynet và (2) các giải pháp dựa trên hệ thống phát hiện xâm nhập (IDS) [24] [36]. Các giải pháp thuộc nhóm (1) xây dựng các honeynet - là các mạng bẫy để thu thập các thông tin về các botnet đang hoạt động và sau đó sử dụng các thông tin thu thập được để phân tích các đặc tính và hành vi của botnet. Nhìn chung, các giải pháp dựa trên honeynet có ưu điểm là dễ xây dựng và không yêu cầu lớn về tài nguyên tính toán. Tuy vậy, các giải pháp này thường bị hạn chế về khả năng mở rộng và khả năng tương tác với mã độc botnet. Các giải pháp thuộc nhóm (2) sử dụng các kỹ thuật giám sát, phát hiện của IDS để giám sát, phát hiện botnet. Dựa trên kỹ thuật phát hiện, các giải pháp dựa trên IDS lại có thể được chia thành (i) phát hiện dựa trên dấu hiệu, chữ ký và (2) phát hiện dựa trên bất thường. Trong các hướng phát hiện dựa trên bất thường, hướng phát hiện botnet dựa trên giám sát lưu lượng mạng, giám sát các truy vấn hệ thống DNS sử dụng học máy được quan tâm nghiên cứu, phát triển và cho nhiều kết quả khả quan [24] [36].

Luận án này tập trung nghiên cứu các phương pháp, kỹ thuật phát hiện các dấu hiệu hoạt động của các botnet sử dụng dữ liệu truy vấn hệ thống DNS dựa trên học máy. Trước hết, luận án sẽ thực hiện khảo sát về botnet, kiến trúc và hoạt động của botnet, và khảo sát, hệ thống hóa các giải pháp giám sát, phát hiện botnet. Sau đó, luận án phát triển và thử nghiệm một số mô hình phát hiện DGA botnet dựa trên các kỹ thuật học máy sử dụng dữ liệu truy vấn hệ thống DNS.

Phần tiếp theo sẽ trình bày về tính cấp thiết, mục tiêu, các đóng góp và bố cục của luận án.

## 2. TÍNH CẤP THIẾT CỦA LUẬN ÁN

Như đã đề cập trong mục Giới thiệu, các botnet đã thực sự trở thành một trong các mối đe dọa lớn nhất đối với mạng Internet toàn cầu do chúng đã và đang phát triển rất mạnh về cả quy mô, mức độ phân tán, kỹ thuật điều khiển và trực tiếp thực hiện, hoặc có liên quan chặt chẽ đến nhiều hoạt động độc hại, như tấn công DDoS, phát tán thư rác, quảng bá, phát tán các loại phần mềm độc hại, phần mềm gián điệp, quảng cáo, giả mạo địa chỉ URL, giả mạo hệ thống DNS, tấn công chèn mã độc trên các ứng dụng web và đánh cắp các thông tin nhạy cảm trên các hệ thống máy chủ cũng trên hệ thống máy người dùng cuối [38] [43] [86]. Một số họ mã độc tống tiền (*ransomware*) được phát hiện gần đây có khả năng tự quảng bá, truyền thông qua mạng botnet và thậm chí các cuộc tấn công có chủ đích (APT) cũng đã bắt đầu sử dụng các botnet để triển khai thực hiện. Trong vài năm qua, một xu hướng mới của mạng botnet như một dịch vụ (*Botnet as a Service - BaaS*) đã hình thành, làm giảm chi phí của tội phạm mạng khi thực hiện các cuộc tấn công liên tục với qui mô rất lớn và mặt khác, giúp chúng kiểm soát botnet dễ dàng hơn. Cùng với xu hướng này, ngày càng có nhiều mạng botnet với quy mô ngày càng tăng với mức độ phân tán rất cao, tạo ra mối đe dọa nghiêm trọng đối với hệ sinh thái Internet [23].

Do mối đe dọa của các botnet đối với mạng Internet toàn cầu, các hệ thống, dịch vụ và người dùng Internet ngày càng lớn, việc nghiên cứu, phát triển và ứng dụng các giải pháp giám sát, phát hiện và loại trừ botnet là rất cấp thiết. Tuy vậy, do các bot trong botnet thường có tính phân tán, khả năng giấu mình và tính tự động (*autonomy*) rất cao, nên việc giám sát, phát hiện và loại trừ botnet gặp rất nhiều thách thức [24] [36]. Giải pháp tổng thể để khắc chế mối đe dọa từ botnet cần sự phối hợp hành động từ nhiều bên có liên quan, bao gồm các cơ quan chính quyền, các nhà cung cấp dịch vụ Internet (*ISP*), các tổ chức, doanh nghiệp và cả người dùng Internet. Chẳng hạn, cần có khung pháp lý về an toàn thông tin mạng từ các cơ quan chính quyền; cần có

các hệ thống giám sát, phát hiện hoạt động của mã độc, các bot, botnet trên các cổng dịch vụ của các ISP, các cơ quan, tổ chức, doanh nghiệp; và ý thức cảnh giác của người dùng Internet. Trong đó, các giải pháp, kỹ thuật giám sát, phát hiện hoạt động và loại trừ các bot, botnet đóng vai trò trọng yếu và đây cũng là hướng nghiên cứu của đề tài luận án này - tập trung nghiên cứu phát hiện botnet sử dụng kỹ thuật phát hiện xâm nhập dựa trên bất thường.

Luận án sử dụng kỹ thuật phát hiện xâm nhập dựa trên bất thường cho phát hiện botnet do kỹ thuật này có ưu điểm nổi bật là có khả năng phát hiện các dạng bot, botnet mới mà không đòi hỏi phải có trước các thông tin về chúng như kỹ thuật phát hiện dựa trên dấu hiệu, chữ ký. Hơn nữa, phát hiện dựa trên bất thường cho phép tự động hóa quá trình xây dựng mô hình phát hiện botnet từ tập dữ liệu huấn luyện, nhờ đó giảm thiểu việc sử dụng nhân lực chuyên gia cho xây dựng thủ công các tập luật phát hiện. Nhược điểm chính của phát hiện botnet dựa trên bất thường là tỷ lệ cảnh báo sai (*gồm tỷ lệ dương tính giả và và tỷ lệ âm tính giả*) còn tương đối cao so với kỹ thuật phát hiện dựa trên dấu hiệu, chữ ký [24] [36].

Trong nhóm các kỹ thuật phát hiện botnet dựa trên bất thường, các hướng (1) phát hiện botnet dựa trên giám sát lưu lượng mạng và (2) phát hiện dựa trên giám sát và phân tích truy vấn DNS thu hút được sự quan tâm lớn của cộng đồng nghiên cứu và các hãng bảo mật. Nổi bật trong hướng (1) là các hệ thống giám sát, phát hiện botnet đã được phát triển và triển khai, như BotHunter [20], BotSniffer [19], BotTrack [32], BotMiner [21], BotFinder [92] và BotProbe [18]. Các hệ thống trên đã được triển khai và đã giám sát, thu thập được một lượng lớn dữ liệu lưu lượng mạng có liên quan đến hoạt động của các bot và botnet phục vụ cho phân tích. Nhằm hỗ trợ cho các nhóm nghiên cứu, Garcia và cộng sự [16] đã xây dựng bộ dữ liệu thu thập lưu lượng mạng botnet với nhiều kịch bản khác nhau với tên là CTU-13. Nhược điểm chính của các hệ thống dạng này là yêu cầu rất cao về năng lực bắt, xử lý và lưu trữ một lượng rất lớn các gói tin lưu thông qua các cổng mạng. Điều này có thể làm giảm khả năng triển khai và vận hành hiệu quả các giải pháp dạng này trên thực tế, đặc biệt là trên các cổng mạng có lưu lượng lớn.

Hướng (2) phát hiện botnet dựa trên giám sát và phân tích các truy vấn DNS được đông đảo cộng đồng nghiên cứu quan tâm trong những năm gần đây, đặc biệt với sự phát triển vượt trội của các họ DGA botnet. DGA botnet gồm các họ botnet sử dụng các thuật toán để tự động sinh và đăng ký tên miền cho các máy chủ CnC của chúng [24] [36]. Đây là kỹ thuật mà các botnet sử dụng để thay thế cho các tên miền và địa chỉ IP cố định cho các máy chủ CnC của chúng nhằm lẩn tránh các kỹ thuật rà quét và chặn lọc. Trong quá trình hoạt động của botnet, botmaster tự động định kỳ sinh các tên miền sử dụng kỹ thuật DGA cho các máy chủ CnC của botnet và đăng ký với hệ thống DNS động. Trong khi đó, các bot trong botnet được lập trình để tự động kết nối máy chủ máy chủ CnC của botnet để tải các lệnh và mã cập nhật. Để thực hiện kết nối, các bot định kỳ tự sinh tên miền của máy chủ CnC sử dụng cùng kỹ thuật DGA và gửi tên miền này lên hệ thống DNS cục bộ để tìm địa chỉ IP của máy chủ CnC. Nếu bot nhận được địa chỉ IP từ hệ thống DNS, nó tạo kết nối đến máy chủ CnC để tải các lệnh và mã cập nhật. Nếu tên miền truy vấn không tồn tại, bot lại sinh một tên miền mới và thực hiện lại quá trình truy vấn hệ thống DNS ở chu kỳ kế tiếp. Mỗi họ DGA botnet sử dụng các thuật toán DGA sinh tên miền khác nhau và số lượng, tần suất sinh tên miền mới cũng khác nhau. Một số họ botnet sử dụng thuật toán DGA sinh tên miền dựa trên thời gian, hoặc dựa trên việc tổ hợp ngẫu nhiên các ký tự (*character-based DGA*), hoặc dựa trên việc tổ hợp các từ lấy trong từ điển (*word-based DGA*), hoặc dựa trên sự kết hợp giữa tổ hợp ngẫu nhiên các ký tự và tổ hợp các từ lấy trong từ điển (*mixed DGA*). Về số lượng tên miền sinh, một số botnet chỉ sinh vài chục tên miền trong cả vòng đời hoạt động, trong khi đó cũng có những botnet sinh hàng chục, thậm chí hàng trăm ngàn tên miền trong vòng đời hoạt động của chúng.

Như vậy, do hoạt động của các DGA botnet gắn liền với việc truy vấn hệ thống DNS, nên có thể giám sát và phân tích các truy vấn các máy chủ DNS có thể tìm được các bằng chứng về sự tồn tại các bot và hoạt động của botnet [24]. Có nhiều giải pháp, kỹ thuật được sử dụng cho giám sát, phân tích lưu lượng truy vấn DNS và nhận dạng, phân loại các tên miền được sử dụng bởi botnet và các tên miền hợp lệ. Các đề xuất

tiêu biểu cho giám sát, phân tích lưu lượng truy vấn DNS để phát hiện botnet bao gồm[33] [53] [56] [68] [83] [85] [98]. Trong thời gian gần đây, các phương pháp học máy được sử dụng rộng rãi trong nhận dạng, phân loại các tên miền được sử dụng bởi botnet và các tên miền hợp lệ nhờ đạt độ chính xác cao và khả năng tự động hóa xây dựng mô hình phát hiện từ tập dữ liệu huấn luyện. Các đề xuất tiêu biểu cho hướng phát hiện botnet dựa trên học máy bao gồm [24] [26] [70] [87] [96] [103]. Ưu điểm của các đề xuất nêu trên là độ chính xác tương đối cao khi thử nghiệm với từng tập dữ liệu cụ thể và khả năng tự động hóa việc xây dựng mô hình phát hiện. Tuy vậy, tỷ lệ cảnh báo sai của các đề xuất này còn khá cao, đến hơn 10% với [24], ảnh hưởng đến khả năng triển khai thực tế. Lý do cho vấn đề này là tập đặc trưng, hoặc phương pháp phân loại sử dụng trong các đề xuất đã có chưa thực sự phù hợp để nhận dạng sự khác biệt giữa các tên miền DGA và các tên miền hợp lệ. Ngoài ra, do một số họ DGA botnet liên tục sử dụng các thuật toán sinh tên miền mới, như các họ word-based và mixed DGA cho phép sinh các tên miền DGA rất giống với các tên miền hợp lệ và do vậy một số đề xuất đã có không có khả năng phát hiện các họ DGA botnet này [24] [96].

Đề tài "Nghiên cứu các kỹ thuật phát hiện DGA botnet" được thực hiện trong phạm vi luận án tiến sĩ chuyên ngành hệ thống thông tin nhằm góp phần giải quyết một số vấn đề còn tồn tại trong các kỹ thuật, giải pháp phát hiện các dạng DGA botnet, bao gồm: (1) lựa chọn, trích xuất tập đặc trưng mới phù hợp hơn để phân biệt tốt hơn các tên miền DGA và tên miền hợp lệ, nhằm tăng độ chính xác phát hiện, giảm tỷ lệ cảnh báo sai và (2) phát triển mô hình kết hợp có khả năng phát hiện đồng thời nhiều họ DGA botnet.

## 3. MỤC TIÊU CỦA LUẬN ÁN

Mục tiêu của luận án là nghiên cứu, đề xuất một số mô hình phát hiện DGA botnet dựa trên các kỹ thuật học máy. Cụ thể, luận án tập trung vào các mục tiêu sau:

- Nghiên cứu, đánh giá các phương pháp, kỹ thuật, giải pháp, công cụ phát hiện botnet hiện có;

- Nghiên cứu, đề xuất các mô hình phát hiện botnet dựa trên học máy có giám sát và học kết hợp sử dụng các tập đặc trưng phân loại tên miền mới nhằm nâng cao độ chính xác, giảm cảnh báo sai, đồng thời cho phép phát hiện nhiều dạng DGA botnet;
- Cài đặt, thử nghiệm và đánh giá các mô hình phát hiện botnet đã đề xuất sử dụng các tệp dữ liệu thực tế.

## 4. ĐỐI TƯỢNG NGHIÊN CỨU VÀ PHẠM VI NGHIÊN CỨU

- Đối tượng nghiên cứu là botnet và đặc biệt là các họ DGA botnet.
- Phạm vi nghiên cứu giới hạn trong các kỹ thuật, giải pháp phát hiện DGA botnet sử dụng dữ liệu truy vấn DNS.

## 5. PHƯƠNG PHÁP NGHIÊN CỨU

Luận án sử dụng phương pháp nghiên cứu lý thuyết kết hợp với phương pháp thực nghiệm. Trong đó, phương pháp nghiên cứu lý thuyết được sử dụng để thực hiện các phần việc sau:

- Nghiên cứu nền tảng lý thuyết về botnet cho luận án, bao gồm khái quát về botnet, bot, phương thức hoạt động của botnet, vấn đề botnet khai thác hệ thống DNS trong quá trình hoạt động;
- Nghiên cứu nền tảng lý thuyết về học máy cho luận án, bao gồm khái quát về học máy, một số giải thuật học máy có giám sát, phương pháp đánh giá và các độ đo đánh giá mô hình phát hiện dựa trên học máy;
- Khảo sát, đánh giá các đề xuất, giải pháp đã có cho phát hiện botnet, DGA botnet, trên cơ sở đó tổng hợp các ưu điểm, nhược điểm làm cơ sở cho đề xuất của luận án;
- Lựa chọn, đề xuất các đặc trưng mới, xây dựng các mô hình phát hiện DGA botnet dựa trên phân loại trên miền DGA với tên miền hợp lệ.

Phương pháp thực nghiệm được sử dụng trong luận án để thực hiện các phần việc sau:

- Khảo sát các tập dữ liệu về botnet, DGA botnet và lựa chọn tập dữ liệu phù hợp cho thực nghiệm;
- Thử nghiệm các mô hình phát hiện DGA botnet đề xuất trong luận án, đánh giá, so sánh các mô hình đề xuất với các mô hình, đề xuất đã có.

## 6. CÁC ĐÓNG GÓP CỦA LUẬN ÁN

**Đóng góp thứ nhất của luận án** là đề xuất mô hình phát hiện DGA botnet dựa trên học máy sử dụng các đặc trưng ký tự và các đặc trưng từ. Mô hình sử dụng các đặc trưng ký tự có khả năng phát hiện hiệu quả các character-based DGA botnet - là các botnet tự sinh tên miền sử dụng thuật toán ghép ngẫu nhiên các ký tự. Mô hình sử dụng các đặc trưng từ có khả năng phát hiện hiệu quả các word-based DGA botnet - là các botnet tự sinh tên miền sử dụng thuật toán ghép các từ theo từ điển.

**Đóng góp thứ hai của luận án** là đề xuất mô hình phát hiện DGA botnet dựa trên học kết hợp (*ensemble learning*). Mô hình này cho phép phát hiện hiệu quả cả character-based và word-based DGA botnet sử dụng thuật toán học kết hợp.

## 7. BỐ CỤC CỦA LUẬN ÁN

Luận án được bố cục thành ba chương với nội dung chính như sau:

**Chương 1** giới thiệu tổng quan về botnet, khái quát về phát hiện botnet, các kỹ thuật phát hiện botnet và một số giải pháp, công cụ phát hiện botnet. Chương 1 cũng giới thiệu khái quát về học máy và mô tả một số giải thuật học máy có giám sát sử dụng trong các mô hình phát hiện botnet đề xuất trong các chương 2 và 3. Phần tiếp theo của chương mô tả các tập dữ liệu liên quan đến botnet được sử dụng trong luận án. Phần cuối của chương chỉ ra 2 vấn đề sẽ được giải quyết trong luận án.

**Chương 2** trình bày khái quát về DGA botnet và cơ chế DGA botnet khai thác hệ thống DNS để duy trì hoạt động. Chương này cũng khảo sát các phương pháp, đề xuất hiện có cho phát hiện botnet nói chung và DGA botnet nói riêng. Phần tiếp theo

của chương mô tả, thử nghiệm và đánh giá mô hình phát hiện character-based DGA botnet dựa trên học máy sử dụng các đặc trưng ký tự. Phần cuối của chương mô tả, thử nghiệm và đánh giá mô hình phát hiện character-based DGA botnet dựa trên học máy sử dụng các đặc trưng ký tự.

**Chương 3** giới thiệu khái quát về học kết hợp (*ensemble learning*), khảo sát các kỹ thuật phát hiện DGA botnet dựa trên học kết hợp (*ensemble learning*). Phần cuối của chương mô tả, thử nghiệm và đánh giá mô hình phát hiện DGA botnet đề xuất dựa trên học kết hợp.

Cuối cùng là Kết luận của luận án.

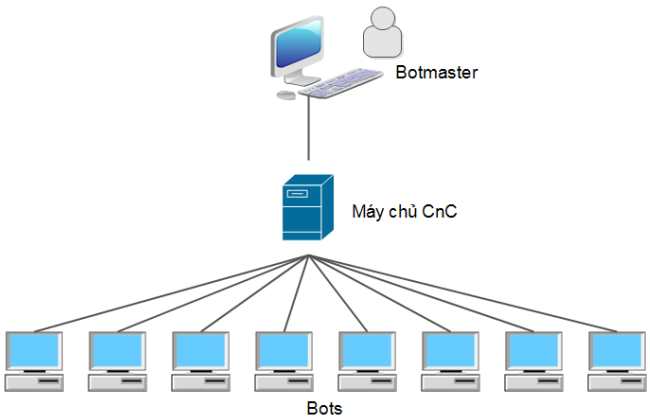
# CHƯƠNG 1: TỔNG QUAN VỀ BOTNET VÀ PHÁT HIỆN BOTNET

## 1.1. TỔNG QUAN VỀ BOTNET

### 1.1.1. Khái quát về botnet và phương thức hoạt động

#### *1.1.1.1. Giới thiệu về bot, botnet*

*Bot* là một loại phần mềm độc hại cho phép kẻ tấn công giành quyền kiểm soát máy tính, hoặc thiết bị tính toán bị lây nhiễm. Máy tính bị nhiễm bot thường được gọi là *zombie* hay là *máy tính ma*. Trên thực tế có hàng ngàn, hàng trăm ngàn máy tính và thiết bị tính toán có kết nối Internet bị nhiễm một số loại bot mà người dùng không biết và không nhận ra chúng. Kẻ tấn công có thể truy cập các *zombie* và kích hoạt chúng thực thi các cuộc tấn công từ chối dịch vụ, hoặc gửi hàng loạt thư rác. Khi thực hiện truy vết ngược lại nguồn khởi phát các cuộc tấn công, người ta thường tìm thấy các *zombie* - cũng là nạn nhân chứ không phải là kẻ tấn công thực sự. Các bot do một hoặc một nhóm kẻ tấn công thông qua một hoặc một số máy tính (*gọi là botmaster*) kiểm soát và chúng được liên kết tạo thành một mạng lưới các máy bị kiểm soát được gọi là *botnet*. Botmaster thường điều khiển các bot trong botnet do mình kiểm soát thông qua hệ thống các máy chủ chỉ huy và kiểm soát (*Command and Control, hoặc C&C, hoặc CnC*), như minh họa trên Hình 1.1. Kênh giao tiếp giữa các bot và các máy chủ CnC trong một botnet có thể là IRC, HTTP hoặc giao thức truyền thông khác.



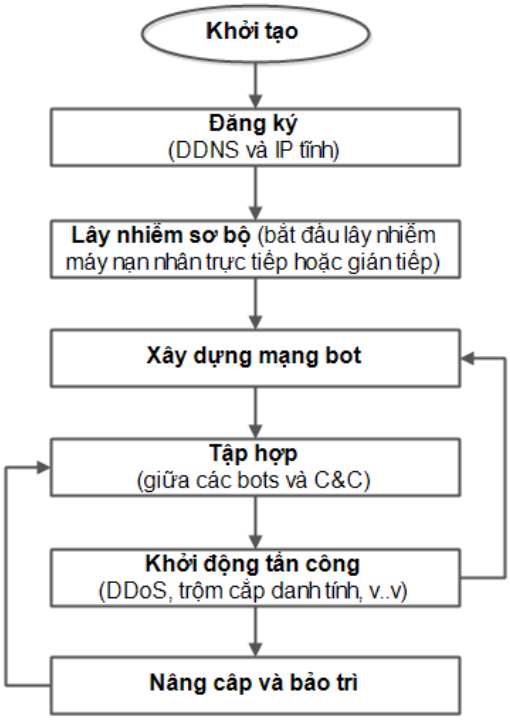

*Hình 1.1: Mô hình botmaster kiểm soát các bot thông qua các máy chủ CnC*

Botnet là mối đe dọa an ninh hàng đầu đối với mạng Internet toàn cầu, một phần là do các bot có thể lây nhiễm vào mọi hệ thống có kết nối Internet ở bất cứ nơi nào. Hơn nữa, so với các dạng phần mềm độc hại khác, các bot có phần vượt trội vì chúng có khả năng tương tác, phối hợp hành động với các bot khác trong botnet. Ngoài ra, sự phát triển của Internet và sự gia tăng của băng thông, đường truyền mạng đã làm tăng thêm đáng kể sức mạnh của các botnet. Mỗi botnet có thể có hàng ngàn, hàng chục ngàn, thậm chí hàng trăm ngàn thành viên là các máy tính, hoặc các thiết bị tính toán bị lây nhiễm bot. Botnet rất khó bị phát hiện bởi chúng có khả năng thích ứng nhanh để lẩn tránh các hệ thống an ninh phổ biến hiện nay. Botnet đã trở thành một mối đe dọa thường trực trên mạng Internet do chúng thường trực tiếp hoặc gián tiếp liên quan đến các hành vi độc hại khác nhau bao gồm: gửi thư rác, tấn công từ chối dịch vụ phân tán và tham gia thực thi nhiều hành vi nguy hiểm và độc hại khác, như lan truyền, lây nhiễm các phần mềm gián điệp và mã độc đến hàng triệu máy tính, thực hiện đánh cắp dữ liệu nhận dạng và tham gia vào các hành vi gian lận, hăm dọa và tống tiền khác [66]. Nghiên cứu về botnet và phát hiện, ngăn chặn botnet thu hút được sự quan tâm lớn của cộng đồng mạng bởi sự nguy hiểm của botnet và sự cấp thiết tìm ra các phương pháp hiệu quả cho phát hiện và ngăn chặn botnet.

#### *1.1.1.2. Phương thức hoạt động, vòng đời*

Hình 1.2 biểu diễn vòng đời của một mạng botnet [59] [62] [105]. Theo đó, các bước trong vòng đời botnet  bao gồm Khởi tạo, Đăng ký, Lây nhiễm sơ bộ, Xây dựng mạng bot, Tập hợp, Khởi động tấn công, và Nâng cấp và Bảo trì. "*Khởi tạo*" là bước cơ bản trong vòng đời của botnet, ở đó botmaster thiết lập các thông số của các bot để bắt đầu truyền thông. Sau giai đoạn đầu, botmaster thực hiện "*Đăng ký*" tên miền và địa chỉ IP tĩnh cho các máy chủ CnC với hệ thống tên miền động (DDNS). Ở giai đoạn "*Lây nhiễm sơ bộ*", thủ thuật lây nhiễm mã bot thường được tiến hành dưới nhiều hình thức khác nhau, như thông qua việc tải nội dung không mong muốn và chạy các tệp đính kèm độc hại từ thư điện tử, hoặc thông qua ổ đĩa di động bị nhiễm mã độc [3] [59] [74] [105]. Trong bước "*Xây dựng mạng bot*", các bot tiếp tục quá trình lây nhiễm và cài đặt mã độc sang các hệ thống mới. Máy bị nhiễm bot tìm kiếm

các nạn nhân mới và thực hiện cài đặt mã độc tải từ các máy chủ CnC. Sau khi được tải và cài đặt vào hệ thống, máy bị nhiễm mã độc hoạt động như một bot thực thụ. Quá trình tải mã độc về hệ thống thường được thực hiện thông qua các giao thức HTTP, FTP, hay P2P.




*Hình 1.2: Vòng đời của botnet*

Trong bước "*Tập hợp*" tiếp theo, các kết nối được thiết lập giữa các bot và máy chủ CnC của chúng. Một số nhà nghiên cứu đặt tên cho bước này là "*giai đoạn kết nối*" [59]. Trong thực tế, bước này luôn xảy ra bất cứ khi nào các bot khởi động lại và duy trì trạng thái kết nối với máy chủ CnC để có thể nhận được các lệnh thực thi các hành động từ botmaster. Do đó, giai đoạn *tập hợp* là một quá trình có tính chu kỳ trong toàn bộ vòng đời của botnet [48]. Sau khi thiết lập thành công kết nối đến máy chủ CnC, trong giai đoạn "*Khởi động tấn công*", các bot nhận lệnh từ máy chủ CnC và bắt đầu thực hiện các hoạt động độc hại. Mục đích cuối cùng của các mạng botnet là thực hiện các hoạt động độc hại, bao gồm tấn công DDoS, phân tích lưu lượng mạng, trộm cắp tài nguyên máy tính hoặc mạng, lây lan các loại mã độc khác, tìm kiếm lỗ hổng trong hệ thống máy tính, đánh cắp dữ liệu danh tính, khai thác các tài

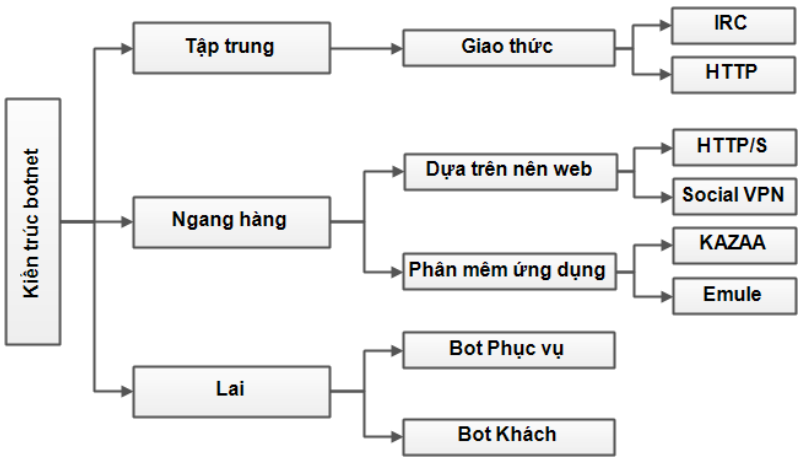
liệu cá nhân và thao túng các trò chơi trực tuyến [27] [71] [105]. Giai đoạn cuối cùng của vòng đời botnet là "*Nâng cấp và bảo trì*". Bảo trì là một việc cần thiết để giữ mối liên hệ giữa botmaster với các bot cho các cuộc tấn công tiếp theo. Hơn nữa, có nhiều lý do cho việc cập nhật mã nhị phân cho các bot, như lẩn tránh các kỹ thuật rà quét, phát hiện và bổ sung thêm các tính năng mới cho các bot [3] [59] [105]. Đây được xem là bước botnet dễ bị tổn thương do nó có thể bị phát hiện trong giai đoạn này bằng cách quan sát, phân tích các hành vi mạng.

### 1.1.2. Phân loại botnet

Các loại botnet có thể được phân loại theo 2 tiêu chí: (i) theo kiến trúc mạng và (ii) theo giao thức truyền thông. Botnet có thể được tổ chức theo nhiều mô hình mạng, chủ yếu theo mô hình tổ chức hệ thống các máy chủ CnC là trung gian giữa botmaster và các bot. Các giao thức truyền thông là các giao thức hỗ trợ giao tiếp giữa các máy chủ CnC và các bot trong botnet.

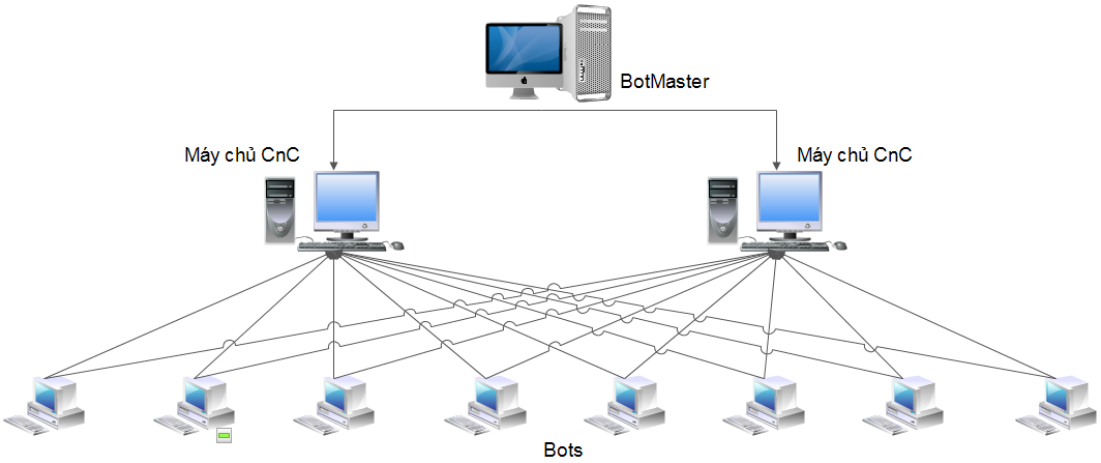
#### *1.1.2.1. Phân loại botnet theo kiến trúc mạng*

Sức mạnh của các botnet nằm ở khả năng hình thành và điều khiển một mạng lưới linh hoạt các máy bị điều khiển có kết nối Internet. Do đó, các cách tiếp cận khác nhau được sử dụng để giải quyết các vấn đề truyền thông giữa các thực thể trong mạng botnet. Hình 1.3 biểu diễn các kiến trúc mạng botnet [99] [21] [47] [48] [62], bao gồm kiến trúc tập trung, kiến trúc ngang hàng và kiến trúc lai.




*Hình 1.3: Phân loại botnet theo kiến trúc mạng*

**Kiến trúc tập trung**: Botnet với kiến trúc hệ thống CnC tập trung tương tự như mô hình khách- chủ (*client-server*) truyền thống, như biểu diễn trên Hình 1.4. Giao thức IRC là một giao thức điển hình được sử dụng trong kiến trúc CnC hệ thống tập trung [62], trong đó các bot thiết lập kênh truyền thông với một hoặc nhiều điểm kết nối. Các máy chủ CnC được triển khai trên các điểm kết nối có trách nhiệm gửi các lệnh và mã cập nhật đến các bot. IRC và HTTP thường được sử dụng là các giao thức chính trong kiến trúc tập trung.

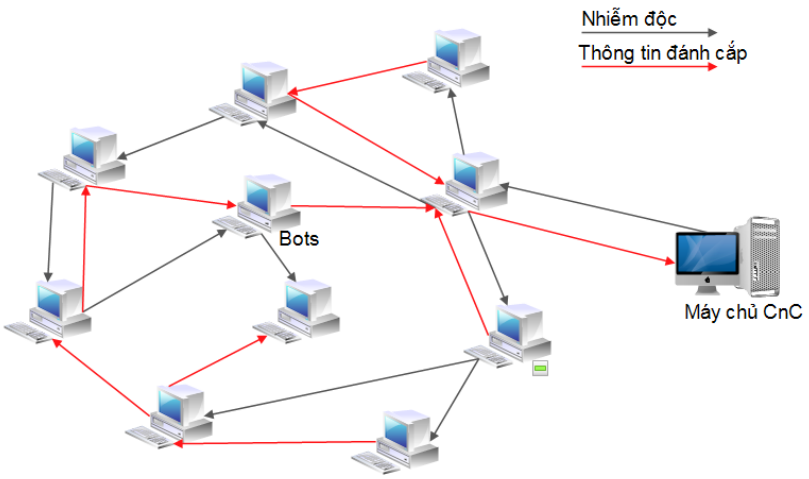



*Hình 1.4: Kiến trúc CnC tập trung*

Các ưu điểm của kiến trúc tập trung bao gồm: (i) triển khai dễ dàng, không đòi hỏi các phần cứng chuyên dụng; (ii) phản ứng nhanh do các máy chủ CnC trực tiếp điều phối các bot trong mạng mà không bị can thiệp bởi bên thứ ba; (iii) khả năng tiếp cận tốt do có sự phối hợp trực tiếp giữa botmaster và các bot; (iv) cập nhật kịp thời thông tin từ botmaster; và (v) khả năng mở rộng tốt. Hạn chế của kiến trúc tập trung là máy chủ CnC được xem như là điểm yếu duy nhất trong hệ thống botnet [99] [62] và dễ dàng bị vô hiệu hóa.

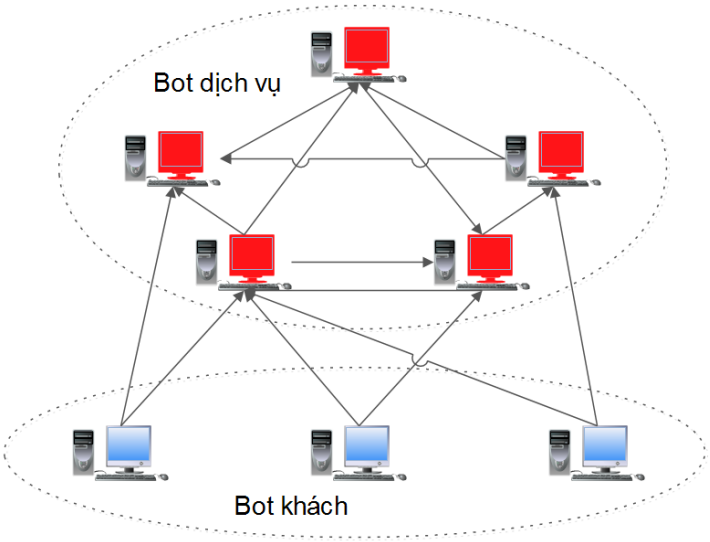
**Kiến trúc ngang hàng**: Với kiến trúc ngang hàng hoặc phi tập trung, các botnet có tính linh hoạt cao hơn [47], hỗ trợ một số lượng lớn các bot và đạt được hiệu quả tối đa, như biểu diễn trên Hình 1.5. Nhìn chung, khó đối phó với các botnet phi tập trung vì những lý do sau: (i) Việc vô hiệu hóa botnet phụ thuộc vào việc phát hiện từng nhóm bot hoạt động như botnet đơn lẻ, do đó rất khó để thống kê được tổng số

lượng bot trong botnet sử dụng kiến trúc ngang hàng để cuối cùng có thể vô hiệu hóa toàn bộ botnet; (ii) Botnet theo kiến trúc ngang hàng không có một mạng lưới máy chủ CnC tập trung do đó rất khó để chuẩn đoán phạm vi bị ảnh hưởng bởi botnet; và (iii) Rất khó để tạm dừng một botnet vì sự liên kết lỏng lẻo giữa các bot. Ưu điểm của kiến trúc ngang hàng là khó bị vô hiệu hóa do botnet có thể duy trì hoạt động mà không cần sử dụng các máy chủ CnC tập trung. Tuy nhiên, các mạng botnet ngang hàng chậm hội tụ và phản ứng, khó quản lý và khả năng mở rộng kém.




*Hình 1.5: Kiến trúc botnet ngang hàng*

**Kiến trúc lai**: Kiến trúc lai kế thừa các thuộc tính của kiến trúc tập trung và kiến trúc ngang hàng, như mô tả trên Hình 1.6. Trong botnet với kiến trúc lai, có 2 loại bot hoạt động [99]: bot phục vụ (servant bot) và bot khách (client bot). Bot phục vụ hoạt động đồng thời như một máy khách và một máy chủ, được cấu hình với các địa chỉ IP định tuyến (*IP tĩnh*); Ngược lại, bot khách chỉ hoạt động như một máy khách và được cấu hình với các địa chỉ IP không định tuyến (*IP động*). Bot phục vụ gửi thông tin địa chỉ IP của mình đến danh sách bot và ở chế độ lắng nghe chờ các kết nối đến từ các bot khách. Ở chiều ngược lại, các bot khách nhận địa chỉ IP từ các bot phục vụ, tạo kết nối để nhận các lệnh và mã cập nhật. Botnet với kiến trúc lai sử dụng mật mã khóa đối xứng để bảo mật giao tiếp giữa bot phục vụ và bot khách.




*Hình 1.6: Kiến trúc botnet lai*

#### *1.1.2.2. Phân loại botnet theo giao thức truyền thông*

Các botnet khác nhau sử dụng các giao thức khác nhau cho truyền thông [59] và điều này làm cho việc nghiên cứu, phát hiện và phòng ngừa botnet khó khăn hơn, đặc biệt là khi botmaster có xu hướng sử dụng các giao thức truyền thông, ứng dụng phổ biến. Các giao thức, hoặc ứng dụng phổ biến được sử dụng trong truyền thông của botnet bao gồm IRC, HTTP, DNS và P2P. Các giao thức, hoặc ứng dụng truyền thông khác cũng đã được sử dụng nhưng không phổ biến, như IM hoặc Skype.

**IRC**: Internet Relay Chat (*IRC*) là một trong những giao thức đầu tiên được sử dụng cho truyền thông trong mạng botnet. IRC hoạt động theo mô hình client-server, trong đó các bot sẽ đăng ký với máy chủ CnC do botmaster kiểm soát và chờ các lệnh. Sự đơn giản, linh hoạt và tính khả dụng của phần mềm máy chủ và máy khách IRC với mã nguồn mở hoặc miễn phí làm cho giao thức này rất hấp dẫn đối với những kẻ tấn công. Tuy nhiên, các botmaster bắt đầu chuyển sang sử dụng các giao thức khác khi các IRC botnet ngày càng trở nên nổi tiếng hơn và do giao tiếp dựa trên kênh IRC có thể dễ dàng bị chặn trên thực tế.

**HTTP**: Giao thức truyền siêu văn bản (*HTTP*) là một giao thức truyền thông phổ biến được sử dụng bởi các botnet. Các máy khách (*bot*) liên lạc với máy chủ HTTP do botmaster kiểm soát để nhận lệnh và thực hiện. Các lệnh được phát hành từ

máy chủ HTTP có thể được lồng trong lưu lượng HTTP thông thường để không gây ra sự nghi ngờ. Gần đây, các phương pháp sử dụng HTTP cải tiến đã được phát hiện, trong đó botmaster xuất bản các lệnh trên các trang web công khai cho phép người dùng tải lên một số dạng nội dung. Sau đó, khi các bot riêng lẻ truy cập các trang web trên, chúng kiểm tra và tải các lệnh được xuất bản gần đây [22].

**DNS**: Hệ thống phân giải tên miền (*DNS*) cũng có thể được sử dụng để truyền các lệnh điều khiển từ botmaster đến các bot trong botnet. Botmaster đạt được điều này bằng cách ẩn các lệnh bên trong lưu lượng DNS thông thường. Các máy chủ DNS theo đặc điểm kỹ thuật sẽ không nhận thấy bất kỳ điều gì bất thường với các yêu cầu DNS, trong khi botmaster kiểm soát một máy chủ DNS độc hại sẽ biết cách trích xuất thông điệp bí mật nhúng trong lưu lượng DNS.

**P2P**: Một số botnet sử dụng các giao thức truyền thông ngang hàng (*P2P*) để truyền các lệnh điều khiển tới các bot, ví dụ như Storm Worm botnet [25]. Cách thức các giao thức này hoạt động có thể khác nhau, vì một số botnet sử dụng các ứng dụng P2P nguồn mở, hoặc có thể tạo ra các giao thức P2P riêng của chúng.

Điều quan trọng cần lưu ý là ngay cả khi các giao thức truyền thông được đề cập ở trên không sử dụng, các botmaster có thể mã hóa lưu lượng lệnh điều khiển của chúng bằng cách sử dụng HTTP, hoặc bất kỳ một giao truyền thông khác. Hơn nữa, không giống như IRC có thể dễ dàng bị chặn, lưu lượng truy cập HTTP và DNS không thể bị chặn hoàn toàn vì chúng rất quan trọng đối với hoạt động của mạng Internet [51].

### 1.1.3. Lịch sử phát triển của botnet

Khái niệm 'botnet' đã được biết đến vào năm 1993 thông qua sự ra đời của botnet đầu tiên có tên 'Eggdrop' [99]. Lịch sử của các botnet được nêu trong Bảng 1.1. Cột “Năm” cho biết năm khởi đầu của mỗi botnet, "Số lượng bot ước tính" là số bot dự đoán tham gia cuộc tấn công do botnet thực hiện, "Khả năng spam" cung cấp số lượng thư rác ước tính botnet gửi đi mỗi ngày. Cột “bí danh” dùng để chỉ các quy ước đặt tên khác nhau được sử dụng bởi mỗi botnet. Cột “Cách tiếp cận phát hiện” chỉ các

phương pháp, công cụ phát hiện và giảm nhẹ tác tại của botnet có thể sử dụng và cột "Kiểu" chỉ giao thức truyền thông sử dụng trong botnet (*IRC, P2P, SMTP, HTTP, v...v.*) [37] [67].

*Bảng 1.1: Lịch sử phát triển botnet*

| Năm | Tên | Số bot ước tính | Khả năng spam (tỷ/ngày) | Bí danh | Cách tiếp cận phát hiện | Kiểu | Tham khảo |
|---|---|---|---|---|---|---|---|
| 1993 | Eggdrop | - | - | Valis | - | IRC | Wang (2003) |
| 1998 | GTBot | - | - | Aristotles | - | mIRC | Janssen (2011) |
| | NetBus | - | - | NetPrank | AV software | HTTP | Wikipedia (1998) |
| 1999 | !A | 1 tỷ | - | - | - | - | Wikipedia (2013b) |
| 2002 | Sdbot/Rbbot | - | - | IRC-SDBot | Data mining, SVM | IRC | Sevcenco (2012) |
| | Agobot | - | - | W32.HLLW.Gaobot, Gaobot | Expert system | IRC | Podrezov (2013) |
| 2003 | Spybot | - | - | - | - | P2P, IRC | Schiller & Binkley (2007) |
| | Sinit | - | - | W32.Sinit, Troj/BDSinit | Network flow analysis | P2P | Wang và cộng sự (2007) |
| 2004 | Bolax | 100,000 | 27 | - | - | - | Kassner (2003) |
| | Bagle | 230,000 | 5.7 | Beagle, Mitglieder | Symantec | SMTP | Symantic (2010) |
| | Marina Botnet | 6,215,000 | 92 | Damon Briant, BOB.dc, Cotmonger, Hacktool.Spammer, Kraken | | | |
| | Torpig | 180,000 | | Sinowal, Anserin | | | |
| | Storm | 160,000 | 3 | Nuwar, Peacomm, Zhelatin | | | |
| 2006 | Rustock | 150,000 | 30 | RKRustock, Costrat | Operation b107 | IRC | Miller (2008) |
| 2007 | Akbot | 1,300,000 | - | - | Operation: not roast | IRC | The H Sercurity (2007) |
| | Cutwail | 1,500,000 | 74 | Pandex, Mutant | - | SMTP | Marry (2010) |
| | Srizbi | 450,000 | 60 | Cbeplay, Exchanger | Symantec | IRC | BBC (2008) |
| | Storm | 160,000 | 3 | Nuwar, Peacomm, Zhelatin | Fast flux | P2P | Francia (2007) |
| 2008 | Conficker | 10,500,000+ | 10 | DownAndUp, Kido | AV soft | HTTP/P2P | Schmudlach (2009) |
| | Mariposa | 12,000,000 | - | - | Manual | IRC/HTTP | McMillan (2010) |
| | Sality | 1,000,000 | - | Sector, Kuku, Kookoo | Manual | P2P | Falliere (2011) |
| | Asprox | 15,000 | - | Danmec, Hydraflux | Symantec | HTTP | Goodin (2008) |
| | Gumblar | n/a | - | - | Manual | HTTP | Mills (2009) |
| | Waledac | 80,000 | 1.5 | Waled, Waledpak | Kaspersky | SMTP/P2P | Goodin (2010) |

| Năm | Tên | Số bot ước tính | Khả năng spam (tỷ/ngày) | Bí danh | Cách tiếp cận phát hiện | Kiểu | Tham khảo |
|---|---|---|---|---|---|---|---|
| 2008 | Onewordsub | 40,000 | 1.8 | N/A | - | SMTP | Keizer (2008) |
| | Mega-D | 509,000 | 10 | Ozdok | Manual | HTTP | Warner (2010) |
| | Torpig | 180,000 | - | Sinowal, Anserin | ESET | HTTP/IRC | Miller (2009) |
| | Bobax | 185,000 | 9 | Bobic, Oderoor, Cotmonger | Manual/BitDefender | HTTP | Symantic (2010) |
| | Maazben | 50,000 | 0.5 | - | - | SMTP | Symantic (2010) |
| 2009 | Grum | 560,000 | 39.9 | Tedroo | FireEye researchers | SMTP | Danchev (2009) |
| | BredoLab | 30,000,000 | 3.6 | Oficla | Symantec | HTTP/SMTP | Crowfoot (2012) |
| | Zeus | 3,600,000 | n/a | Zbot, PRG, Wsnpoem | - | - | Messer (2009) |
| | Mega-D | 509,000 | 10 | Ozdok | | | |
| 2010 | Kelihos | 300,000 + | 4 | Hlux | Kaspersky | P2P | Stefan (2013) |
| | TDL4 | 4,500,000 | n/a | TDSS, Alureon | Kaspersky's TDSS killer | IRC | Kespersky (2011) |
| | LowSec | 11,000 + | 0.5 | LowSecurity, FreeMoney | Symantec | HTTP | Symantic (2010) |
| | Gheg | 30,000 | 0.24 | Tofsee, Mondera | Manual | DoS | Symantic (2010) |
| 2011 | Flashback | 600,000 | n/a | BacDoor.Flashback.39 | Java program | P2P | Musil (2012) |
| | Ramnit | 3,000,000 | | | | | |
| 2012 | Chameleon | 120,000 | - | - | - | HTML | Spider (2013) |
| 2013 | Boatnet | 500 + server computers | 0.01 | YOLOBotnet | - | | Wikipedia (2013b) |
| | Zer0n3t | 200+ server computers | 4 | Fib3rl0g1c, Zer0n3t, Zer0Log1x | | | |
| 2016 | Mirai | 380,000 | | Malware | | TCP, FTP | CloudFlare |
| 2017 | Star Wars | 350,000 | | Twitter Botnet | | | Penta Security (2017) |
| | Reaper | 100,000 | | IoT Botnet | | HTML, DNS | |
| 2018 | Cridex<br>Danabot | *60,000* | | Trojan | | DNS, URL | Secure List (2018) |
| 2019 | Emotet | 250,000 | | Trojan, Ransomwares | | SMTP | Zdnet |
| | TrickBot | 125,000 | | Trojan: Ryuk, Conti | | Windows | Security Boulevard |

### 1.1.4. Tác hại và các dạng khai thác botnet

Botnet có thể được sử dụng cho một loạt các hành động nguy hiểm, bao gồm tấn công DDoS, tạo và gửi thư rác (*spam*), lừa đảo, lây lan phần mềm độc hại, quảng bá phần mềm quảng cáo, gián điệp, lưu trữ các trang web hoặc nội dung độc hại [51]:

**DDoS**: Các botnet được sử dụng thường xuyên trong các cuộc tấn công DDoS. Một kẻ tấn công có thể điều khiển số lượng lớn các máy bị chiếm quyền điều khiển tại một trạm từ xa, khai thác băng thông và gửi yêu cầu tấn công tới máy đích. Nhiều hệ thống mạng đã bị tê liệt khi hứng chịu các cuộc tấn công DDoS với qui mô lớn sử dụng các botnet với hàng chục, thậm chí hàng trăm ngàn máy bot tham gia.

**Spam**: Botnet là một công cụ lý tưởng cho những kẻ phát tán thư rác do các máy bot trong botnet có số lượng rất lớn và chúng sử dụng địa chỉ IP động nên giảm thiểu khả năng bị chặn, hoặc vô hiệu hóa. Thư rác từ botmaster được chuyển tới các bot thông qua hệ thống máy chủ CnC và từ đó phát tán tới người nhận.

**Lây lan phần mềm độc hại**: Botnet cũng là một công cụ hiệu quả cho tội phạm mạng trong vận chuyển và lan truyền các phần mềm độc hại nói chung và các bot nói riêng nhờ tính phân tán cao và qui mô lớn. Ngoài ra, việc phán tán và lan truyền bot trên mạng Internet tăng cường việc mở rộng qui mô của botnet.

**Các hoạt động gián điệp**: Các bot có thể lấy cắp thông tin trên máy người dùng cuối và cả trên các máy chủ. Điều này có nghĩa là các botmaster có thể thu thập thông tin quan trọng và nhạy cảm như mật khẩu, tài khoản người dùng hoặc bất kỳ loại thông tin nhạy cảm nào khác có thể truy cập trên máy bot. Các thông tin nhạy cảm sau đó có thể được bán, được sử dụng để tạo ra lợi nhuận tăng thêm. Các bot cũng có thể được sử dụng hiệu quả trong hoạt động nghe lén (*sniffing*) các hoạt động của người dùng trên máy bot.

**Lưu trữ các nội dung độc hại**: Nhờ tính phân tán và khả năng phục hồi, botnet có thể được sử dụng để lưu trữ các nội dung độc hại, do đó các cơ quan bảo vệ pháp luật khó tìm và xóa nội dung này hơn. Các botmaster không chỉ có thể thu lợi từ việc lưu trữ nội dung mà còn từ quảng cáo liên quan đến nội dung độc hại được lưu trữ.

## 1.2. PHÁT HIỆN BOTNET

### 1.2.1. Khát quát về phát hiện botnet

Chính vì mối đe dọa từ botnet ngày một gia tăng, các nghiên cứu về botnet và các xu hướng, cách tiếp cận để phát hiện botnet ngày càng được nhiều nhà nghiên cứu và các tổ chức quan tâm. Phát hiện botnet đề cập đến việc phát hiện các hoạt động nguy hiểm, hoặc bất thường được thực hiện trong môi trường mạng được kiểm soát. Phát hiện botnet hiện đang là một thách thức lớn đối với các nhà nghiên cứu và các tổ chức do botnet được xem là mục tiêu di động nhờ tính phân tán cao và khả năng ẩn mình của các bot. Như vậy, tất cả các khía cạnh có liên quan đến phát hiện botnet bao gồm phát hiện, giảm thiểu và phản ứng phải luôn thay đổi theo thời gian. Để có thể phòng chống botnet hiệu quả cần sự phối hợp của nhiều bên liên quan. Các bên liên quan khác nhau, ví dụ như các cơ quan chính phủ, các doanh nghiệp, các nhà mạng và các nhà cung cấp dịch vụ Internet (ISP) có nhiều cách tiếp cận khác nhau để xử lý vấn đề botnet.

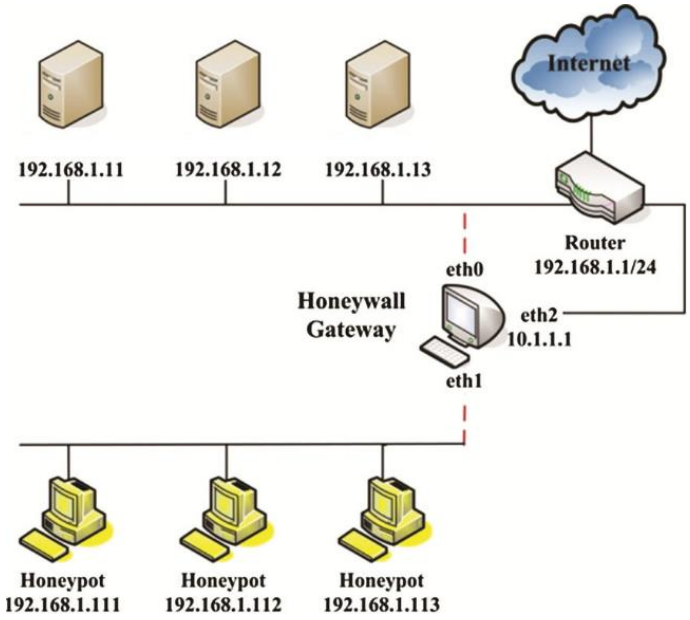
Feily và cộng sự  đã phân loại các kỹ thuật phát hiện botnet thành bốn loại chính: dựa trên chữ ký, dựa trên DNS, dựa trên khai phá dữ liệu, và dựa trên dị thường [59]. Hơn nữa, họ cung cấp một biểu đồ so sánh để làm nổi bật tầm quan trọng của các kỹ thuật này đối với các phương pháp phát hiện, như phát hiện bot chưa biết, phát hiện giao thức và cấu trúc độc lập, phát hiện botnet sử dụng mã hóa, phát hiện theo thời gian thực, cũng như sai số phát hiện. Ngoài ra, Feily và cộng sự cũng phân loại kỹ thuật phát hiện botnet theo các cách tiếp cận khác nhau, chẳng hạn như phát hiện dựa trên hành vi tạo lập, chữ ký và hành vi tấn công [61]. Li và cộng sự thảo luận về botnet và các nghiên cứu liên quan dựa trên mô hình CnC, cơ chế lây nhiễm, các giao thức truyền thông, các hành vi độc hại và các cơ chế phòng thủ [46]. Jing và cộng sự trình bày kiến trúc cơ bản của các cuộc tấn công botnet dựa trên IRC, trong đó các hoạt động nguy hiểm đã được phát hiện bằng cách giám sát trực tiếp mô hình truyền thông của IRC [47]. Các nghiên cứu [58] [71] đã thảo luận các kiến trúc CnC khác nhau (*CnC tập trung, P2P, và CnC lai*) cho các botnet. Hơn nữa, nghiên cứu [71] đã

phân loại kỹ thuật phát hiện botnet thành 2 loại, bao gồm phát hiện dựa trên honeynet và phát hiện dựa trên hệ thống phát hiện xâm nhập (*IDS*).

### 1.2.2. Các kỹ thuật phát hiện botnet

Có nhiều kỹ thuật phát hiện botnet đã được đề xuất và ứng dụng trong những năm qua. Mục này trình bày 4 nhóm kỹ thuật phát hiện botnet được sử dụng phổ biến, bao gồm (i) phát hiện dựa trên honeynet, (ii) phát hiện dựa trên luật, dấu hiệu và (iii) phát hiện dựa trên bất thường [59] [71].

#### *1.2.2.1. Phát hiện dựa trên Honeynet*



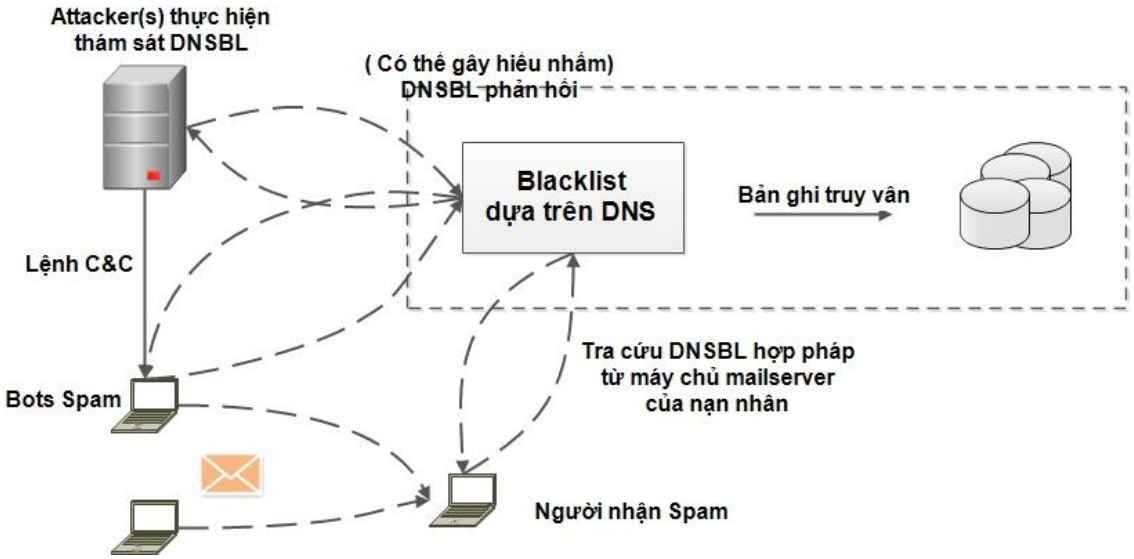

*Hình 1.7: Kiến trúc Honeynet*

Honeynet là hệ thống mạng được mô phỏng giống mạng đích, được sử dụng để thu thập các bot và thâm nhập vào các botnet [74] [89]. Hình 1.7 minh họa kiến trúc honeynet, bao gồm một honeywall và các honeypot. Có nhiều kỹ thuật khác nhau để thu thập các bot trong honeypot [46] [71]. Thành phần quan trọng của honeynet là honeywall, được sử dụng để phân chia ranh giới giữa honeypot với bên ngoài. Các honeywall là một thiết bị ở lớp L2/L3 đóng vai trò như một cửa ngõ để lưu lượng mạng đi qua. Ý nghĩa thực tế của một honeynet bao gồm sự đơn giản trong triển khai, ít yêu cầu về nguồn lực, chi phí triển khai tối thiểu và hữu dụng với dữ liệu mã hoá. Tuy nhiên, các hạn chế của honeynet bao gồm: (i) khả năng mở rộng là có hạn vì nó

đòi hỏi thiết bị phần cứng chuyên dụng được triển khai; (ii) Honeypot không thể lường trước được các cuộc tấn công mà nó chỉ có thể theo dõi các hoạt động độc hại khi tương tác với nó; (iii) Phát hiện các hệ thống bị nhiễm bệnh, được đặt như một cái bẫy cũng là một thách thức; (iv) Trong một số trường hợp, những kẻ tấn công có thể tiếp quản và điều khiển các honeypot để làm hại hệ thống hoặc các máy khác ngoài honeynet.

#### *1.2.2.2. Phát hiện dựa trên luật, dấu hiệu*

Dấu hiệu, hay chữ ký (*signature*) sử dụng trong phát hiện botnet là các mẫu hoặc đặc trưng của các botnet đã biết. Kỹ thuật này dựa trên việc so sánh các thông tin thu thập được với các chữ ký đã được xác định từ trước của các botnet, từ đó có thể phân biệt và phát hiện ra các hành vi độc hại so với các hành vi bình thường khác. Ưu điểm của kỹ thuật dựa trên dấu hiệu, chữ ký là có khả năng phát hiện nhanh và chính xác những botnet đã biết. Tuy vậy, kỹ thuật này không thể phát hiện các bot, botnet mới. Ngoài ra, cần thường xuyên cập nhật cơ sở dữ liệu dấu hiệu, chữ ký để có thể đảm bảo khả năng phát hiện.

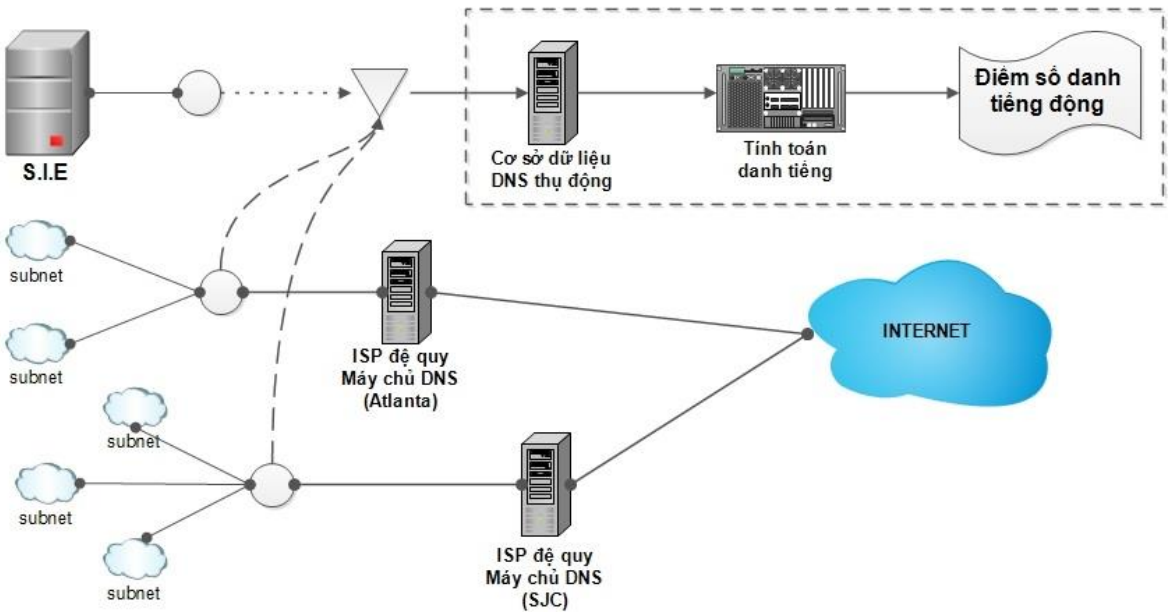



*Hình 1.8: Kiến trúc giảm spam dựa trên DNSBL*

Phát hiện dựa trên danh sách đen dựa trên DNS (*DNSBL*) được đề xuất bởi Ramachandran và cộng sự  là một ví dụ về hệ thống phát hiện botnet dựa trên chữ ký [75]. Hướng tiếp cận dựa trên DNSBL tìm kiếm chữ ký của các bot đã biết trong quá

trình giám sát lưu lượng DNS. Hướng tiếp cận dựa trên DNSBL cũng chỉ ra những hành động spam và độc hại thông qua thu thập địa chỉ IP của các máy chủ hoặc mạng nào đó có liên quan đến những hành động trên. Hướng tiếp cận dựa trên DNSBL cố gắng ghi nhận địa chỉ IP và xác định vị trí của botmaster, như biểu diễn trên Hình 1.8. Tuy nhiên, hạn chế của hướng tiếp cận dựa trên DNSBL là phải thường xuyên duy trì cập nhật cơ sở dữ liệu địa chỉ độc hại đã biết.

Tương tự, Antonakakis và cộng sự [55] đã xây dựng một hệ thống danh tiếng động DNS gọi là "Notos" sử dụng dữ liệu truy vấn DNS thụ động, phân tích mạng và đặc trưng vùng của một tên miền, như biểu diễn ở Hình 1.9. Hệ thống Notos giả định rằng truy vấn DNS độc hại có đặc điểm đặc biệt và có thể phân biệt với những truy vấn DNS lành tính. Do đó, việc quan sát các truy vấn DNS và xây dựng những mô hình của những tên miền độc hại và lành tính là khả thi và có thể dẫn đến một kết quả tốt. Điểm danh tiếng của một tên miền được tính toán bằng mô hình và thông thường sẽ cho điểm thấp đối với tên miền độc hại và điểm cao với tên miền lành tính.

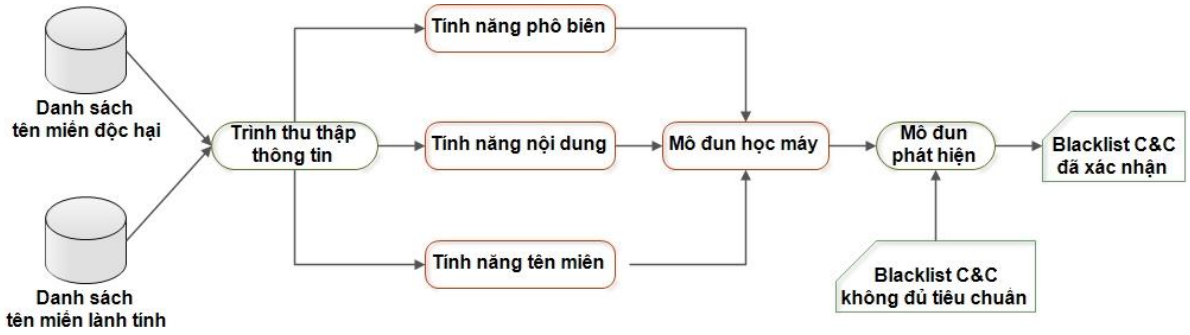



*Hình 1.9: Hệ thống danh tiếng động DNS (Notos)*

Hệ thống Notos đạt được độ chính xác cao và tỷ lệ sai thấp và nó có thể ghi nhận những tên miền mới trước khi chúng được đưa vào danh sách đen. Tuy nhiên, hệ thống cần nhiều hoạt động trong quá khứ với một tên miền nhất định để đạt tới

một điểm danh tiếng chính xác. Hệ thống cho kết quả không chính xác khi tên miền máy chủ CnC trong các botnet thường xuyên thay đổi.

Hệ thống Mentor được đề xuất bởi Kheir và cộng sự [40] thực hiện loại bỏ các tên miền hợp pháp ra khỏi danh sách trên miền botnet CnC để giảm tỷ lệ sai trong quá trình phát hiện, như thể hiện ở Hình 1.10. Hệ thống Mentor thu thập, thống kê đặc trưng của các tên miền bị nghi ngờ như thuộc tính của DNS, nội dung trang web để xây dựng một mô hình DNS áp dụng kỹ thuật học máy có giám sát vào bộ tên miền lành tính và độc hại đã biết. Mentor cho tỷ lệ sai rất thấp khi thử nghiệm trên danh sách đen công khai nhờ loại bỏ các tên miền lành tính khỏi danh sách đen.




*Hình 1.10: Tổng quan hệ thống Mentor*

Yadav và cộng sự [102] đề xuất một hướng tiếp cận để phát hiện "domain fluxes" trong lưu lượng DNS thông qua việc tìm kiếm các mẫu được tạo bởi thuật toán và sự phân bố các ký tự trong tên miền sử dụng bởi botnet khác biệt so với tên miền do con người tạo ra. Tuy nhiên, hệ thống này bị giới hạn trong việc phát hiện tên miền CnC sử dụng bởi những phần mềm độc hại đã biết.

Bảng 1.2 tổng kết lại một số kỹ thuật phát hiện botnet dựa trên chữ ký sử dụng dữ liệu truy vấn DNS.

*Bảng 1.2: Tổng hợp các kỹ thuật phát hiện botnet dựa trên chữ ký*

| Đề xuất | Cơ chế | Hạn chế |
|---|---|---|
| DNSBL [75] | Thu thập những địa chỉ IP công khai của các máy chủ. | Cần thường xuyên cập nhật danh sách đen dựa trên DNS. |
| Notos [55] | Hệ thống DNS danh tiếng động, sử dụng dữ liệu truy vấn DNS thụ động để | Cần nhiều lịch sử hoạt động cho một tên miền nhất định để tạo |

| | | |
|---|---|---|
| | phân tích đặc trưng mạng của một tên miền. | điểm số danh tiếng, không đáng tin với các botnet lai. |
| Mentor [40] | Loại bỏ các trên miền hợp pháp từ danh sách đen các tên miền botnet CnC. | Cần thường xuyên cập nhật thông tin cho hệ thống. |
| Domain fluxes [102] | Phát hiện thông lượng tên miền trong lưu lượng DNS, Tìm kiếm các mẫu được tạo ra bởi thuật toán vốn có với tên miền. | Giới hạn với các botnet đã biết, các cuộc tấn công lẩn tránh trong quá trình phân tích. |

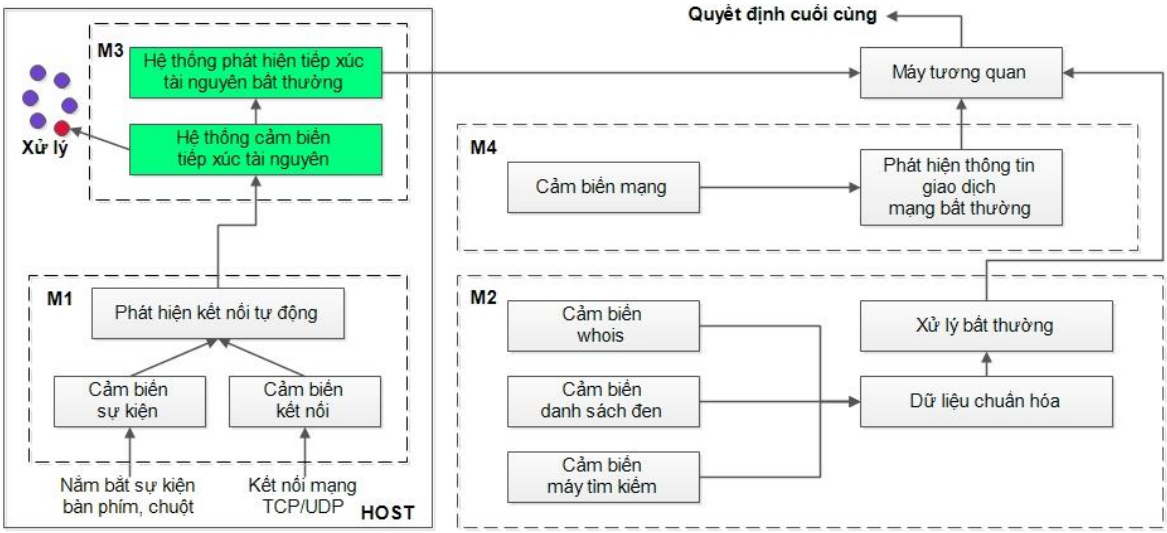
#### *1.2.2.3. Phát hiện dựa trên bất thường*

Phát hiện botnet dựa trên bất thường, hay dị thường là quá trình giám sát, phát hiện các hành vi bất thường trong lưu lượng mạng, hoặc các hành vi trong các máy (*host*) nghi ngờ có liên quan đến hoạt động của botnet và các bot. Thông thường, quá trình phát hiện botnet dựa trên bất thường gồm 2 giai đoạn: (1) xây dựng hồ sơ phát hiện - là tập hành vi mạng, hoặc hành vi máy trong chế độ làm việc bình thường và (2) giám sát phát hiện các hành vi mạng, hoặc hành vi máy khác biệt đủ lớn so với tập hành vi bình thường đã lưu trong hồ sơ. Ưu điểm của phát hiện botnet dựa trên bất thường là có tiềm năng phát hiện các bot, botnet mới mà không cần biết trước thông tin về chúng. Tuy vậy, phát hiện botnet dựa trên bất thường thường có tỷ lệ cảnh báo sai cao hơn so với phát hiện botnet dựa trên chữ ký. Ngoài ra, việc xây dựng hồ sơ phát hiện cũng đòi hỏi nhiều tài nguyên tính toán. Mặc dù vậy, phát hiện botnet dựa trên bất thường vẫn thu hút được sự quan tâm của cộng đồng nghiên cứu do 2 lý do: (i) có thể tự động hóa quá trình xây dựng hồ sơ phát hiện từ dữ liệu huấn luyện và (ii) có thể ứng dụng các kỹ thuật xử lý dữ liệu tiên tiến, như khai phá dữ liệu, học máy và học sâu để xây dựng các mô hình phát hiện phù hợp, nhằm tăng tỷ lệ phát hiện đúng và giảm cảnh báo sai.

Theo vị trí giám sát thu thập dữ liệu, kỹ thuật phát hiện botnet dựa trên bất thường có thể được chia thành 2 hướng: phát hiện dựa trên máy (*host-based*) và dựa trên mạng (*network-based*). Phần tiếp theo trình bày chi tiết về 2 hướng này.

**a. Host-based**: Trong hướng tiếp cận dựa trên máy (*host*), quá trình giám sát và phân tích mang tính cục bộ xảy ra tại mỗi host cụ thể để phát hiện những hoạt động

độc hại thông qua giám sát xử lý hệ thống, tiếp cận tới mức sử dụng nhân hệ điều hành và các lời gọi hàm tới hệ thống [37]. Một ví dụ của kỹ thuật dựa trên host là BotSwat được đề xuất bởi Stinson và Mitchell [89]. BotSwat tập trung vào cách các bot trả lời dữ liệu nhận được qua mạng thông qua việc giám sát thực thi các mã nhị phân trên nền tảng Win32. Hạn chế chính của kỹ thuật phát hiện botnet dựa trên host là nó không có khả năng mở rộng do chỉ gói gọn việc giám sát bot và hoạt động của nó trong host. Hơn nữa, để có cái nhìn bao quát trong mạng, mỗi host phải được trang bị những công cụ giám sát mạnh mẽ và có hợp tác với các host khác [27].




*Hình 1.11: Tổng quan hệ thống EFFORT*

Một trong những kỹ thuật quan trọng tập trung giám sát lưu lượng DNS tại host hoặc mạng là nền tảng EFFORT được đề xuất bởi Shin và cộng sự [83]. Nền tảng này nhằm mục đích phát hiện botnet sử dụng cách tiếp cận đa mô đun và mối liên hệ tương quan từ các host khác nhau, như minh họa trong Hình 1.11. EFFORT sử dụng một thuật toán học máy có giám sát để phân loại tên miền truy vấn là tên miền lành tính hoặc độc hại. EFFORT không phụ thuộc vào topo mạng, giao thức truyền thông triển khai và có khả năng phát hiện các botnet sử dụng các giao thức truyền thông được mã hóa. Tuy nhiên, EFFORT bị giới hạn trong phạm vi của nó với các botnet dựa trên dịch vụ DNS để xác định địa chỉ của các máy chủ CnC. Ngoài ra, EFFORT cũng bị tổn thương khi botnet sử dụng các kỹ thuật lẩn tránh khác nhau.

*Bảng 1.3: Kỹ thuật phát hiện botnet dựa trên host*

| Đề xuất | Cơ chế | Thuật toán | Hạn chế |
|---|---|---|---|
| EFFORT [85] | Áp dụng đa mô đun để tương quan các chỉ dẫn liên quan đến bot từ các client/networks. | SVM | Dễ bị tổn thương với các kỹ thuật lẩn tránh khác nhau. Giới hạn truy cập botnet dựa trên DNS. |

**b. Network-based**: Nhóm kỹ thuật phát hiện bất thường thứ hai là kỹ thuật phát hiện bất thường dựa trên mạng, thực hiện giám sát lưu lượng mạng để nhận biết sự tồn tại của botnet [73] và có thể được phân loại thành phương pháp *giám sát chủ động* và *giám sát thụ động* [27] [37] [72] [82].

***Giám sát chủ động***: Trong giám sát lưu lượng DNS chủ động, người ta chèn vào mạng/máy chủ/ứng dụng các gói tin thử nghiệm để kiểm tra phản hồi của mạng, từ đó có thể xác định được liệu một người dùng hợp pháp hay một bot đang thực hiện phiên kết nối đó. Kỹ thuật này đem lại hiệu quả trong phát hiện các IRC botnet do các IRC bot sẽ định kỳ kết nối đến máy chủ CnC để nhận lệnh và mã cập nhật. Ma, J. và các cộng sự [52] đã đề xuất một phương pháp giám sát chủ động lưu lượng DNS. Cách tiếp cận này trích xuất và phân tích các tên miền từ thư rác dựa trên việc xây dựng cơ sở từ vựng và thông tin host, sử dụng các phương pháp thống kê và học máy để phân loại các trang web độc hại. Tuy nhiên, nhược điểm chính của phương pháp này là chỉ nhắm mục tiêu đến các tên miền trích xuất từ hoạt động gửi thư rác mà không thể phát hiện các miền độc hại có liên quan đến các hoạt động như máy chủ CnC botnet [36]. Theo hướng tương tự, Ma, X. và các cộng sự [53] giám sát chủ động lưu lượng DNS trên quy mô lớn để đánh giá các đặc trưng truy vấn DNS dựa vào hoạt động của bộ đệm DNS. Tuy nhiên, phương pháp này có nhược điểm là sinh thêm nhiều lưu lượng mạng. Mặt khác, việc dò tìm và phân tích DNS chủ động có khả năng cao bị phát hiện bởi kẻ tấn công kiểm soát các miền được phân tích.

***Giám sát thụ động***: Phát hiện dựa trên phân tích lưu lượng DNS thụ động nhằm mục đích chặn bắt các thông điệp DNS giữa máy chủ và máy khách DNS và chuyển tiếp chúng đến điểm phân tích tập trung. Các thông điệp DNS và lịch sử truy vấn

DNS được thu thập giúp theo dõi và giám sát các tên miền độc hại ngay cả khi chúng đã bị xóa hoặc hết hạn. Cranor và cộng sự [12] áp dụng đồ thị với các nút là địa chỉ IP của các máy chủ DNS và các cạnh là các truy vấn được tạo bởi các máy khách để xác định các máy khách và máy chủ DNS. Tuy nhiên, một trong những vấn đề của phương pháp này là không có khả năng xử lý các tập dữ liệu lớn [37]. Dagon và cộng sự [14] đã đề ra một số chỉ số quan trọng để xác định đặc trưng của lưu lượng botnet trong các kiến trúc botnet khác nhau được sử dụng trong giai đoạn tấn công. Tuy nhiên, phương pháp này gặp khó khăn khi botmaster có thể tạo ra các lượng lớn các truy vấn DNS giả mạo để làm gián đoạn kỹ thuật này, do đó tạo ra tỷ lệ cảnh báo sai cao. Villamarín-Salomón và Brustoloni [98] đã đánh giá hai phương pháp tiếp cận xác định các máy chủ CnC botnet dựa trên truy vấn DNS bất thường. Cách tiếp cận đầu tiên dựa trên việc theo dõi tên miền với tỷ lệ truy vấn tập trung cao hoặc bất thường, trong khi phương pháp thứ hai dựa trên việc giám sát các câu trả lời DNS lặp lại bất thường [36]. Cách tiếp cận đầu tiên dựa trên giả định rằng các botmaster thường xuyên thay đổi các máy chủ CnC của họ để tránh bị phát hiện và bị liệt vào danh sách đen, việc thay đổi này có thể dẫn đến tỷ lệ truy vấn DNS cao. Mặt khác, dựa trên các phản hồi của máy chủ DNS với truy vấn tên miền không còn tồn tại có thể suy ra việc các bot đã cố gắng truy vấn hệ thống DNS để tìm địa chỉ IP nhằm kết nối đến các máy chủ CnC nhưng tên miền CnC đã bị thay đổi hoặc ngừng hoạt động. Các kỹ thuật phát hiện botnet dựa trên giám sát thụ động lưu lượng DNS có thể phân loại thành các hướng nghiên cứu dựa trên thống kê, đồ thị, phân cụm, cây quyết định, entropy, và mạng nơ-ron.

*Hướng tiếp cận dựa trên thống kê*: Marko và Vilhan [56] đề xuất phương pháp giám sát lưu lượng DNS trong mạng cục bộ và thông qua các kỹ thuật phân tích thống kê để kết luận sự tồn tại của các botnet trong phạm vi giám sát. Tuy nhiên, kỹ thuật này chỉ có thể xác định botnet sau khi bot nhận được lệnh từ máy chủ CnC [37]. Hu và cộng sự [28] đã đề xuất hệ thống RB-Seeker (*Redirection Botnet Seeker*) cho phép phát hiện botnet với giả thiết một số lượng lớn các máy tính bị xâm nhập và kiểm soát bởi botmaster được sử dụng như một proxy hoặc là thiết bị chuyển hướng. Do

đó, hệ thống RB-Seeker thu thập được các thông tin liên quan đến hoạt động chuyển hướng của botnet và dựa trên phân tích thống kê để phân biệt các tên miền độc hại và không độc hại [36].

Sanchez và cộng sự [77] đã nghiên cứu một phương pháp tiếp cận dựa trên nền tảng máy vector hỗ trợ để loại bỏ các máy tính người dùng độc hại khỏi máy chủ email hợp pháp thông qua thống kê tập các đặc trưng không thể chối bỏ của máy tính độc hại. Cách tiếp cận này có độ chính xác phát hiện cao, tuy nhiên nó chỉ có thể ứng dụng trong các mạng có qui mô nhỏ. Ngoài ra, phương pháp này có thể bị vượt qua bởi một thủ thuật đơn giản là thay đổi tên của máy tính [36]. Antonakakis và các cộng sự [55] đã đề xuất hệ thống Kopis cho phát hiện tên miền độc hại thông qua việc giám sát các yêu cầu và phản hồi từ hệ thống máy chủ DNS. Hệ thống Kopis phát hiện tên miền độc hại dựa trên một số đặc trưng được trích xuất từ thông tin thu được từ hệ thống DNS. Kopis có thể phát hiện các miền liên quan đến phần mềm độc hại một cách độc lập ngay cả khi không có địa chỉ IP.

*Hướng tiếp cận dựa trên đồ thị*: Kỹ thuật phát hiện botnet dựa trên đồ thị cố gắng mô hình hóa các truy vấn DNS thất bại đối với một số tên miền nhất định dưới dạng đồ thị. Jiang và các cộng sự [33] đã đề xuất một cách tiếp cận nhằm phát hiện các hành vi độc hại thông qua biểu đồ các truy vấn DNS thất bại. Cách tiếp cận này sử dụng thuật toán phân tích đồ thị dựa trên hệ số ma trận tri-nonnegative.

*Hướng tiếp cận dựa trên phân cụm*: Các kỹ thuật phát hiện botnet dựa trên phân cụm là việc nhóm các phần tử theo quan điểm: các thành phần được nhóm trong cùng một cụm thì giống nhau hơn so với các thành phần thuộc các cụm khác. Các cụm có các đặc điểm khác biệt so với phần còn lại đến mức có thể dẫn đến kết luận về sự tồn tại của botnet, hoặc các hoạt động của botnet [36]. Perdisci và cộng sự [68] đã đề xuất một phương pháp tiếp cận để theo dõi và phát hiện các mạng fast-flux độc hại. Phương pháp này giả định rằng botmaster thường vận hành botnet sử dụng một số tên miền fast-flux, tất cả đều trỏ tới các flux agent liên quan đến cùng flux service để tránh bị đưa vào danh sách đen tên miền độc hại. Họ nhóm các tên miền đáng ngờ dựa trên

các điểm tương đồng trong bộ các IP được phân giải từ các tên miền này. Choi và cộng sự đề xuất hệ thống BotGAD [30] thực hiện trích xuất các đặc trưng từ luồng lưu lượng DNS được giám sát để phân biệt các truy vấn hợp pháp và bất thường, từ đó nhóm các thiết bị có cùng hành vi đáng ngờ, ví dụ các thiết bị cố gắng truy vấn một máy chủ CnC hoặc địa chỉ nạn nhân. Phương pháp này có thể phát hiện mạng botnet sử dụng kênh truyền thông được mã hóa. Tuy nhiên, nhược điểm của phương pháp này là không có khả năng xác định các botnet sử dụng thuật toán fast-flux hoặc DGA [59]. Hơn nữa, cách tiếp cận này hạn chế với quy mô mạng lớn do cần nhiều thời gian xử lý.

*Hướng tiếp cận dựa trên Entropy*: Huang và cộng sự [29] đã đề xuất hệ thống Spatial Snapshot Fast-Flux Detection (*SSFD*). SSFD sử dụng các múi giờ để phân biệt botnet trong các không gian hệ thống địa lý khác nhau. Các không gian địa lý kết hợp với entropy thông tin để tính toán cách các thiết bị được phân phối tương đương trong mỗi múi giờ. SSFD được phát triển dựa trên quan sát các tên miền không độc hại có khuynh hướng phân bố trong cùng một múi giờ, trong khi các tên miền độc hại được phân bố nhanh chóng rộng rãi trên nhiều múi giờ. Huang và các cộng sự cũng lưu ý rằng nếu tất cả các bot của một mạng botnet được đặt trong cùng một múi giờ, entropy dựa trên múi giờ sẽ không hiệu quả cho quá trình phát hiện.

*Hướng tiếp cận dựa trên cây quyết định*: Stalmans và Irwin [87] đề xuất sử dụng cây quyết định C5.0 và phương pháp thống kê Bayes để phát hiện các tên miền fast-flux. Theo hướng tương tự, Bilge và cộng sự [44] đề xuất hệ thống EXPOSURE cho phát hiện các tên miền độc hại sử dụng các kỹ thuật phân tích DNS thụ động trên quy mô lớn. EXPOSURE sử dụng 15 đặc trưng được trích xuất từ truy vấn DNS và thông qua một mô đun học tập dựa trên thuật toán cây quyết định để huấn luyện bộ dữ liệu tên miền được gán nhãn, với tên miền độc hại được gán nhãn dương tính và tên miền lành tính được gán nhãn âm tính. Bộ phân loại sau huấn luyện được sử dụng để phân loại các tên miền trích xuất từ truy vấn hệ thống DNS.

*Hướng tiếp cận dựa trên mạng nơ ron*: Mạng nơron nhân tạo đã chứng minh tính hiệu quả của nó trong giải quyết nhiều lớp các bài toán trong các lĩnh vực khác nhau, như xử lý văn bản, xử lý ngôn ngữ tự nhiên và đặc biệt là phân loại và nhận dạng ảnh. Ngoài ra, mạng nơron cũng đã được chứng minh ứng dụng thành công trong phát hiện xâm nhập nói chung và phát hiện botnet nói riêng [36]. Wang và các cộng sự [42] đã đề xuất một hệ thống phát hiện bot dựa trên thuật toán nhận dạng mẫu fuzzy. Họ đã phát triển một mô đun giảm luồng lưu lượng để giảm lượng dữ liệu lưu lượng mạng phải xử lý. Từ lưu lượng mạng đã giảm, nghiên cứu thực hiện trích xuất một số đặc trưng liên quan đến hành vi của bot, như truy vấn DNS không thành công, khoảng thời gian truy vấn DNS, kết nối mạng không thành công và kích thước tải trọng cho kết nối mạng. Cuối cùng, một mô đun nhận dạng được sử dụng để xác định các bot thông qua tính toán khả năng có hoạt động của các bot trong luồng lưu lượng đã giảm [36].

*Hướng tiếp cận dựa trên học máy*: Giả thiết chính của các kỹ thuật phát hiện botnet dựa trên học máy là tạo ra các mẫu (*pattern*) có thể được sử dụng để phân biệt lưu lượng botnet trong luồng lưu lượng mạng. Các mẫu này được tạo một cách hiệu quả bằng cách sử dụng các thuật toán học máy (*MLA – Machine Learning Algorithm*) [24]. Trong phát hiện botnet, phương pháp học không giám sát thường được sử dụng cho việc phân cụm các dữ liệu liên quan đến bot. Đặc trưng chính của MLA không giám sát là chúng không cần phải được huấn luyện trước. Phương pháp học có giám sát thường được sử dụng để phát triển các bộ phân loại có khả năng phân loại những lưu lượng độc hại với các luồng lưu lượng không độc hại, hoặc xác định lưu lượng truy cập của các mạng botnet khác nhau. Đặc trưng chính của MLA có giám sát là chúng cần phải được huấn luyện trước [42].

Theo hướng này, Stevanovic và cộng sự [88] đề xuất mô hình phát hiện botnet sử dụng dữ liệu truy vấn DNS dựa trên một số thuật toán học máy có giám sát, như k-Nearest Neighbor (*kNN*), cây quyết định, rừng ngẫu nhiên, và Naive Bayes. Mô hình được đề xuất dựa trên việc các bot của các botnet thường xuyên gửi các truy vấn tới hệ thống DNS để tìm địa chỉ IP của các máy chủ CnC, trong đó các máy chủ CnC

sử dụng các tên miền được tạo tự động. Kết quả thử nghiệm cho thấy hầu hết các thuật toán học máy đều cho độ chính xác phát hiện cao, trong đó thuật toán rừng ngẫu nhiên cho kết quả tốt nhất.

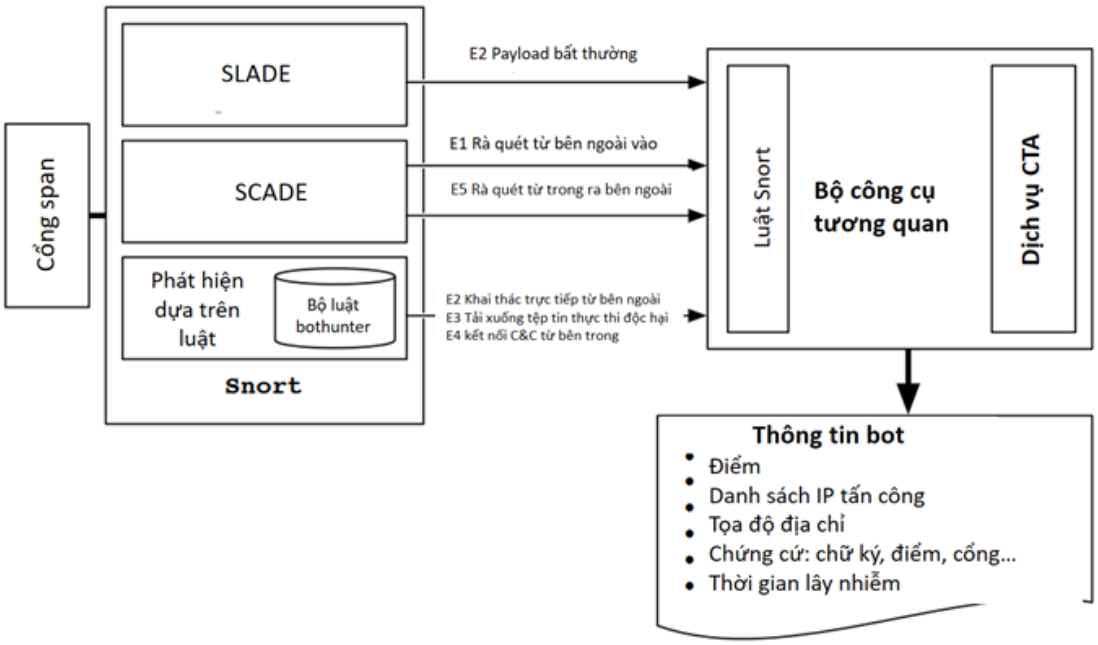
### 1.2.3. Một số giải pháp, công cụ phát hiện botnet

Có nhiều giải pháp, công cụ phát hiện botnet đã được phát triển và triển khai ứng dụng trên thực tế như BotHunter [20], BotSniffer [19], BotTrack [32], BotMiner [21], BotFinder [92] và BotProbe [18]. Mục này mô tả 3 công cụ giám sát, phát hiện botnet điển hình, gồm BotHunter [20], BotSniffer [19] và BotTrack [32].

#### *1.2.3.1. BotHunter*

BotHunter là một hệ thống phát hiện botnet thụ động dựa trên chữ ký sử dụng kỹ thuật phân tích gói tin. Hệ thống bao gồm 3 thành phần IDS để giám sát luồng lưu lượng vào/ra và một công cụ tương quan để tạo ra hồ sơ các bot, như biểu diễn trên Hình 1.12 [20]. BotHunter sử dụng hệ thống phát hiện xâm nhập Snort với bộ luật phát hiện có sẵn và các bộ luật được phát triển nội bộ và tham khảo từ cộng đồng. Snort trong hệ thống sử dụng SCADE (*Statistical sCan Anomaly Detection Engine*) là một plugin tiền xử lý với 2 mô đun: (1) Phát hiện rà quét nội bộ và (2) Phát hiện rà quét bên ngoài. Trong phát hiện rà quét nội bộ, SCADE đặc biệt quan tâm đến phát hiện các hành vi quét các cổng thường được mã độc sử dụng nhắm đến các máy chủ nội bộ. Ngoài ra, nó theo dõi các nỗ lực kết nối không thành công. Có 2 loại cổng được định nghĩa, bao gồm: cổng HS (*high severity*) đại diện cho các cổng dịch vụ thường bị khai thác (*80/HTTP, 445/NetBIOS, 5000/UPNP, 3127/MyDoom*) và cổng LS (*low severity*) đại diện cho các cổng dịch vụ ít bị khai thác. Tiếp theo, hệ thống tính điểm đối với các nỗ lực quét đến các loại cổng khác nhau. Với phát hiện rà quét bên ngoài, hệ thống theo dõi tất cả các kết nối ra bên ngoài của mỗi thiết bị nội bộ và sử dụng 3 mô đun con phát hiện bất thường song song đưa ra các cảnh báo: (i) tỷ lệ quét ra bên ngoài = $s_1$; (ii) tỷ lệ kết nối ra bên ngoài không thành công = $s_2$; (iii) giá trị entropy được chuẩn hóa = $s_3$.

Mỗi mô đun con sẽ đưa ra những cảnh báo con nếu kết quả tương ứng $s_i > t_i$, với $t_i$ là ngưỡng đã xác định. Sau đó SCADE sử dụng một lược đồ quyết định (*ví dụ AND, OR, MAJORITY*…) để đưa ra cảnh báo chính. Ví dụ sử dụng AND, SCADE sẽ đưa ra cảnh báo nếu cả 3 mô đun con đều đưa ra cảnh báo.




*Hình 1.12: Kiến trúc BotHunter*

Bên cạnh SCADE, hệ thống còn sử dụng SLADE (*Statistical PayLoad Anomaly Detection Engine*) để phát hiện các cuộc tấn công dựa trên phân tích tải trọng. Nó kiểm tra tải trọng của mỗi gói tin được gửi đến các dịch vụ được giám sát và đưa ra cảnh báo nếu phát hiện những tải trọng bất thường [20].

Bộ luật phát hiện đóng vai trò quan trọng trong phát hiện các khai thác trực tiếp (E2), tải xuống tệp tin nhị phân (E3), lưu lượng C&C (E4). Các luật BotHunter sử dụng được chia thành 4 tệp luật riêng biệt được tham khảo từ các tổ chức chia sẻ: (i) 1046 luật E2 tập trung vào các tấn công chèn mã từ bên ngoài vào bên trong; (ii) 71 luật E3 tập trung vào các sự kiện tải xuống tệp tin thực thi độc hại và các tệp tin trong bộ luật có sẵn của Snort từ bên ngoài; (iii) 246 luật E4 tập trung vào các kết nối C&C, các lệnh botnet phổ biến; (iv) 20 luật E5 bao gồm các backdoor đã biết trong khi SCADE giúp phát hiện các quét ra các cổng bên ngoài.

Bộ công cụ tương quan trong BotHunter kết hợp với một mô đun cho phép người dùng gửi báo cáo về bot lên một kho lưu trữ Cyber-TA từ xa để thu thập và đánh giá hoạt động của bot trên toàn cầu. Những báo cáo này sẽ được cung cấp cho các nhà cung cấp và nhà nghiên cứu để giúp đánh giá quy mô về hành vi của bot, nguồn lây nhiễm, các biến thể… để tìm ra máy chủ C&C và botmaster. Hồ sơ bot bao gồm: (i) Điểm đánh giá bot (*confidence score*); (ii) Danh sách IP xếp theo mức độ tấn công (*attacker IP list*); (iii) Tọa độ địa chỉ IP (*Coodination Center IP*); (iv) Tập chứng cứ (*full evidence trail*); (v) Thời gian lây nhiễm (*Infection time range*).

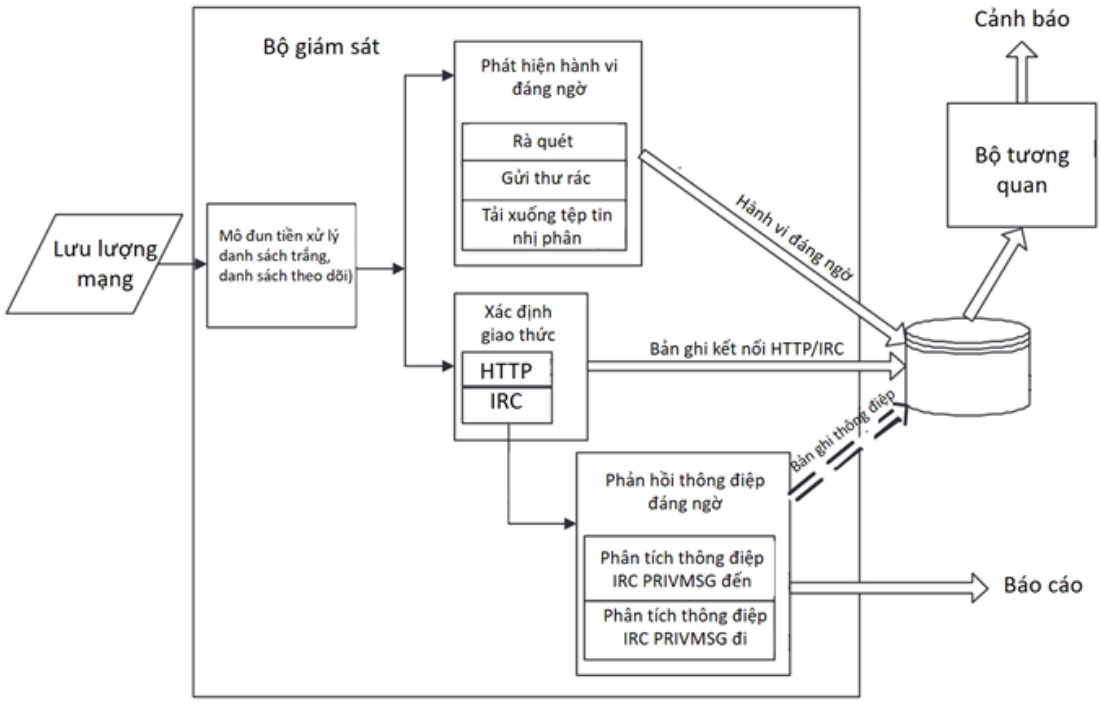
BotHunter là một hệ thống phân tán, có khả năng mở rộng cao, có thể được triển khai trên rìa của mạng, giúp giám sát được các kết nối giữa các thiết bị trong mạng và Internet [20]. Nó có khả năng phát hiện các mạng botnet bất kể sử dụng kiến trúc và giao thức truyền thông nào nếu botnet tuân theo một mô hình vòng đời đã biết [6].

#### *1.2.3.2. BotSniffer*

BotSniffer [19] là hệ thống phát hiện botnet dựa trên bất thường. Ý tưởng của BotSniffer là các bot trong cùng một mạng botnet sẽ thực thi các mã độc như nhau, có các phản hồi lại lệnh máy chủ CnC và có các hành vi tấn công tương tự nhau. Hệ thống BotSniffer bao gồm 2 thành phần: bộ giám sát (*monitor*) và bộ tương quan (*correlation*), như biểu diễn trên Hình 1.13. Trong đó, bộ giám sát được triển khai trên mạng được giám sát và thực hiện các nhiệm vụ: (i) giám sát lưu lượng mạng để sinh ra bản ghi kết nối của các giao thức CnC, (ii) phát hiện các hành vi đáng ngờ của botnet (*như rà quét, gửi thư rác*…) và (iii) phản hồi thông điệp mà botnet hay sử dụng (*như IRC PRIVMSG: phản hồi riêng cho một ai đó*). Các sự kiện từ bộ giám sát sẽ được bộ tương quan phân tích theo nhóm các hành vi, hoặc phản hồi thông điệp tương đồng và giống nhau của thiết bị kết nối đến cùng máy chủ IRC hoặc máy chủ HTTP về mặt không gian và thời gian [19].

Bộ giám sát lại gồm 3 thành phần: mô đun tiền xử lý, mô đun xác định giao thức và mô đun phát hiện và phản hồi hành vi, thông điệp đáng ngờ. Mô đun tiền xử lý (*Preprocessing*) thực hiện lọc lưu lượng để giảm khối lượng luồng cần xử lý. Một

danh sách trắng được tạo ra giúp lọc các giao thức không được sử dụng cho mạng botnet như ICMP, UDP cũng như lọc ra các lưu lượng liên quan đến các máy chủ của các hãng dịch vụ nổi tiếng như Google, Yahoo hay các địa chỉ IP được coi là lành tính. Ngoài danh sách trắng, danh sách theo dõi bao gồm các máy chủ cục bộ đang sử dụng các giao thức mà botnet sử dụng cũng được xây dựng. Mô đun xác định giao thức (*Protocol Matcher*) trong BotSniffer hỗ trợ giám sát 2 giao thức là IRC và HTTP. Việc xác định giao thức được sử dụng là khá dễ dàng. Chẳng hạn, một phiên IRC thường bắt đầu với việc đăng ký kết nối thường sử dụng 3 tin nhắn nhanh là PASS, NICK và USER, hoặc kiểm tra một số byte đầu tiên của payload khi bắt đầu kết nối.; Mô đun phát hiện các hành vi/phản hồi thông điệp đáng ngờ (*Activity/Message Response Detection*) giám sát các hành vi của các máy để phát hiện sự xuất hiện của một phản hồi botnet. Với các hành vi phản hồi thông điệp, BotSniffer giám sát các thông điệp IRC PRIVMSG; đối với các hành vi rà quét, gửi thư rác, tải xuống tệp tin nhị phân, BotSniffer sử dụng hệ thống SCADE như của BotHunter [19].



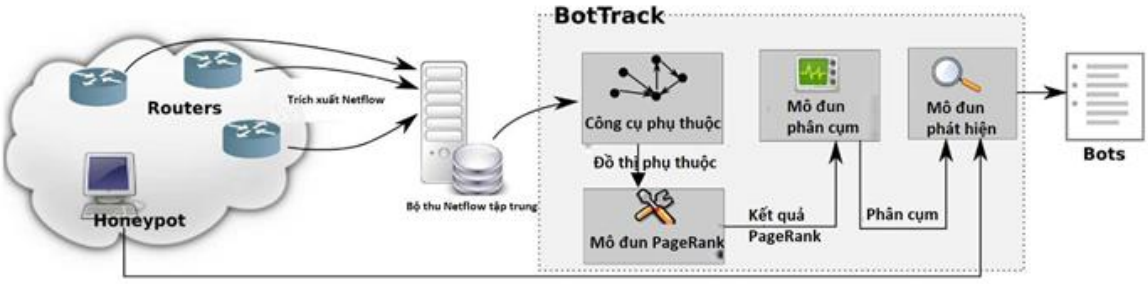

*Hình 1.13: Kiến trúc BotSniffer*

Bộ tương quan trong BotSniffer sẽ nhóm các máy trong mạng theo địa chỉ IP đích truy cập và cổng tương ứng. Khi đó các máy kết nối đến cùng một máy chủ sẽ được nhóm cùng nhau. BotSniffer sử dụng 2 thuật toán: thuật toán Response Crowd Density Check để phân tích nhóm các hành vi đáng ngờ và sử dụng thuật toán Response Crowd Homogeneity Check để phân tích nhóm phản hồi thông điệp. Sau phân tích nếu phát hiện ra dấu hiệu của một kết nối CnC, BotSniffer sẽ đưa ra cảnh báo [19].

BotSniffer có khả năng phát hiện IRC và HTTP botnet mà không đòi hỏi thông tin về chữ ký hoặc máy chủ CnC. Hệ thống cũng có thể phát hiện các bot sử dụng giao tiếp CnC được mã hóa và có thể phát hiện từng bot, hoặc nhóm bot riêng lẻ trong mạng giám sát.

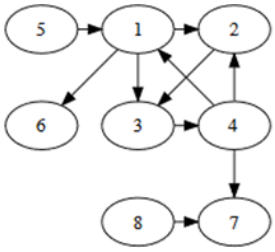
#### *1.2.3.3. BotTrack*

BotTrack [32] là hệ thống giám sát, phát hiện các botnet dựa trên phân tích luồng mạng, như biểu diễn trên Hình 1.14. Các bộ định tuyến giám sát luồng lưu lượng và trích xuất luồng gửi đến bộ thu, luồng lưu lượng tập trung và được đưa vào phân tích bởi BotTrack.




*Hình 1.14: Kiến trúc BotTrack*

Trong bước xử lý đầu tiên, mô đun Công cụ phụ thuộc xây dựng một đồ thị phụ thuộc giữa các thiết bị để xác định sự tương tác giữa các hệ thống *("ai giao tiếp với ai")*. Hình 1.15 mô tả ví dụ đồ thị phụ thuộc giữa 8 nút, mỗi nút đại diện cho một thiết bị và đường liên kết từ nút A đến nút B cho thấy có ít nhất một luồng IP từ thiết bị A đến thiết bị B. Có thể thấy nút 1, 2, 3, 4 có thể là một bot bởi có nhiều các luồng tương tác giữa chúng hơn so với các nút khác.




*Hình 1.15: Đồ thị phụ thuộc 8 nút - BotTrack*

Đồ thị phụ thuộc này sau đó được phân tích tự động sử dụng thuật toán PageRank (*PageRank module*). Thuật toán PageRank là thuật toán phân tích đường dẫn được sử dụng bởi Google để cân nhắc tầm quan trọng tương đối của các trang web trên Internet. Các kết quả đầu ra của mô đun PageRank có 2 giá trị: *authority* và *hub* được xử lý và phân cụm bởi thuật toán DBSCAN [57] do nó phù hợp cho các cơ sở dữ liệu lớn như cơ sở dữ liệu netflow trong môi trường mạng thực tế.

Phát hiện botnet ban đầu chỉ phụ thuộc vào việc so sánh các giá trị authority và hub với các ngưỡng xác định. Các ngưỡng này có thể được tinh chỉnh để có thể phát hiện và có tỷ lệ dương tính giả tốt hơn, nhưng cách tiếp cận này vẫn tạo ra tỉ lệ âm tính giả cao. Như một giải pháp, phân cụm sau đó được áp dụng để trích xuất các cụm địa chỉ IP. Để xác định một cụm địa chỉ IP có liên quan các botnet hay không, BotTrack sử dụng thêm nguồn thông tin bên ngoài - cung cấp bởi hệ thống phát hiện xâm nhập hoặc honeypot [20].

## 1.3. KHÁI QUÁT VỀ HỌC MÁY VÀ CÁC THUẬT TOÁN SỬ DỤNG

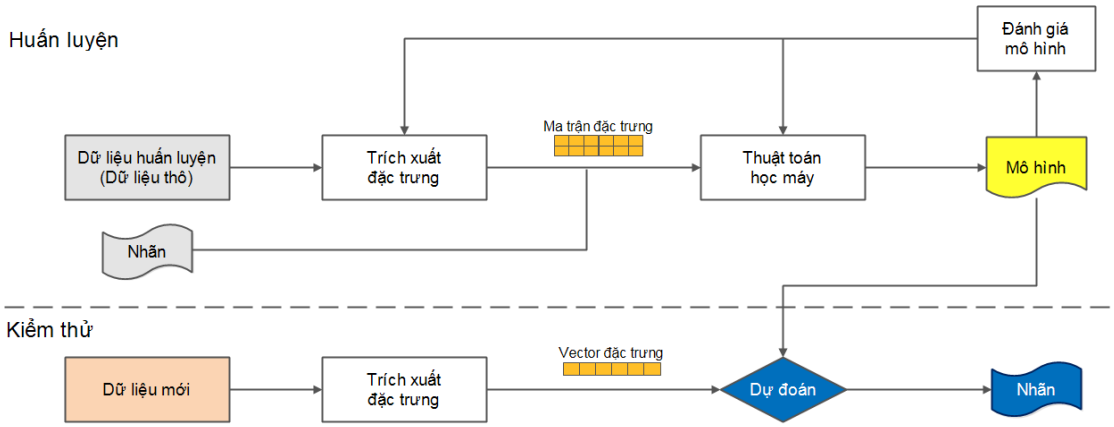
Phân loại nhị phân (*binary classification*) là nhiệm vụ phân loại các phần tử của một tập hợp các đối tượng ra thành 2 nhóm dựa trên cơ sở là một số thuộc tính nào đó (*hay còn gọi là đặc trưng*) của chúng. Đây là kỹ thuật rất phù hợp đối với các vấn đề phát hiện truy cập bất hợp pháp, tấn công mạng,..

### 1.3.1. Giới thiệu về học máy

Theo Tom M.Mitchell [63], "*Một chương trình máy tính được gọi là 'học tập' từ kinh nghiệm E để hoàn thành nhiệm vụ T với hiệu quả được đo bằng phép đánh giá P, nếu hiệu quả của nó khi thực hiện nhiệm vụ T, khi được đánh giá bởi P, cải*

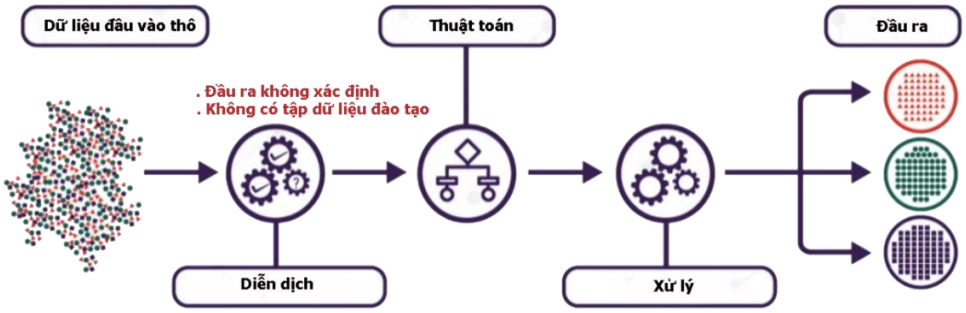
*thiện theo kinh nghiệm E*". Học máy dựa trên khái niệm và kết quả thu được từ một số lĩnh vực, bao gồm: thống kê, trí tuệ nhân tạo, lý thuyết thông tin, triết học, khoa học nhận dạng, lý thuyết điều khiển và sinh học. Phát triển thuật toán học máy là cơ sở ứng dụng trải rộng từ tầm nhìn tính toán đến xử lý ngôn ngữ, dự báo, nhận dạng, trò chơi, khai phá dữ liệu, hệ chuyên gia, robot. Đồng thời, những tiến bộ quan trọng trong lý thuyết và thuật toán học máy thúc đẩy việc học máy trở thành phương tiện chính để khám phá tri thức từ sự phong phú của dữ liệu hiện có trong các ứng dụng. Một trong những ứng dụng của học máy được quan tâm nhiều trong thời gian gần đây là phát hiện xâm nhập, mã độc nói chung và phát hiện botnet nói riêng. Nhiều bài toán phức tạp có thể được giải quyết hiệu quả bằng học máy, như: phân loại, hồi quy, dịch máy, phân cụm và khai phá dữ liệu. Dựa trên tính chất của tập dữ liệu, các thuật toán học máy có thể được phân thành hai nhóm chính là học có giám sát và học không giám sát. Ngoài ra, có hai nhóm thuật toán khác thu hút được nhiều chú ý trong thời gian gần đây là học bán giám sát và học tăng cường [93].

**Học có giám sát**: Một thuật toán học máy được gọi là học có giám sát nếu việc xây dựng mô hình dự đoán mối quan hệ giữa đầu vào và đầu ra được thực hiện dựa trên các cặp (*đầu vào, đầu ra*) đã biết trong tập huấn luyện. Đây là nhóm thuật toán phổ biến nhất trong các thuật toán học máy. Các bài toán phân loại và hồi quy là hai ví dụ điển hình trong nhóm này. Diễn giải theo toán học, học có giám sát xảy ra khi việc dự đoán quan hệ giữa đầu ra y và dữ liệu đầu vào x được thực hiện dựa trên các cặp $\{(x_1, y_2),(x_1, y_2),...,(x_n, y_n)\}$ trong tập huấn luyện. Việc huấn luyện là việc xây dựng một hàm *f* sao cho với mọi $i = 1, 2, \ldots, n$, $f(x_i)$ gần với $y_i$ nhất có thể. Hơn thế nữa, khi có một điểm dữ liệu *x* nằm ngoài tập huấn luyện, đầu ra dự đoán $f(x)$ cũng gần với đầu ra thực sự *y*. Hình 1.16 mô tả các bước của quá trình học và dựa đoán trong mô hình học máy có giám sát.




*Hình 1.16: Mô hình học máy có giám sát*

**Học không giám sát**: Một thuật toán học máy không giám sát là thuật toán có khả năng "học" từ dữ liệu huấn luyện chỉ bao gồm các dữ liệu đầu vào *x* mà không có đầu ra tương ứng. Các thuật toán học máy không giám sát có thể không dự đoán được đầu ra nhưng vẫn trích xuất được những thông tin quan trọng dựa trên mối liên quan giữa các điểm dữ liệu đầu vào. Các thuật toán giải quyết bài toán phân cụm và giảm chiều dữ liệu là các ví dụ điển hình của nhóm các thuật toán học máy không giám sát. Trong bài toán phân cụm, có thể mô hình không trực tiếp dự đoán được đầu ra của dữ liệu nhưng vẫn có khả năng phân các điểm dữ liệu có đặc tính gần giống nhau vào từng nhóm, hay cụm. Hình 1.17 biểu diễn mô hình học máy không giám sát.




*Hình 1.17: Mô hình học máy không giám sát*

**Học bán giám sát**: Một thuật toán học máy bán giám sát là thuật toán có khả năng "học" từ dữ liệu huấn luyện bao gồm một phần dữ liệu là các cặp (*đầu vào, đầu ra*) và một phần dữ liệu khác chỉ có đầu vào. Thực tế cho thấy ngày càng có nhiều

bài toán cần được giải quyết bởi các thuật toán học máy bán giám sát vì việc thu thập và gán nhãn cho dữ liệu có chi phí cao và tốn thời gian. Chẳng hạn, chỉ một phần nhỏ trong các bức ảnh y học trong tập dữ liệu huấn luyện có nhãn vì quá trình gán nhãn tốn thời gian và cần sự can thiệp của các chuyên gia.

**Học tăng cường**: Học tăng cường là một kỹ thuật Học máy dựa trên phản hồi, trong đó tác nhân học cách cư xử trong môi trường bằng cách thực hiện các hành động và xem kết quả của các hành động. Đối với mỗi hành động tốt, tác nhân nhận được phản hồi tích cực và đối với mỗi hành động xấu, tác nhân nhận được phản hồi tiêu cực. Trong học tập tăng cường, tác nhân học tự động bằng cách sử dụng phản hồi mà không có bất kỳ dữ liệu được gắn nhãn nào, không giống như học có giám sát. Vì không có dữ liệu được gắn nhãn, vì vậy tác nhân bị ràng buộc chỉ học hỏi bằng kinh nghiệm bản thân. Tác nhân tương tác với môi trường và tự khám phá nó. Mục tiêu chính của một tác nhân trong việc học tăng cường là cải thiện hiệu suất bằng cách nhận được kết quả tích cực tối đa.

### 1.3.2. Một số thuật toán học máy có giám sát

Mục này trình bày một số thuật toán học máy có giám sát truyền thống được sử dụng trong các mô hình phát hiện botnet đề xuất trong Chương 2 và Chương 3 của luận án, bao gồm: Naïve Bayes, Cây quyết định, Rừng ngẫu nhiên, SVM và Hồi quy Logistic.

#### *1.3.2.1. Naïve Bayes*

Naïve Bayes là một thuật toán dựa trên định lý Bayes về lý thuyết xác suất  để đưa ra các phán đoán cũng như phân loại dữ liệu dựa trên các dữ liệu được quan sát và thống kê. Naïve Bayes là một trong những thuật toán được ứng dụng nhiều trong các lĩnh vực học máy dùng để đưa các dự đoán chính xác nhất dựa trên một tập dữ liệu đã được thu thập, vì nó tương đối đơn giản và cho độ chính xác cao. Naïve Bayes thuộc vào nhóm các thuật toán học có giám sát, tức là học từ các dữ liệu đã được gán nhãn [64] [93].

Áp dụng trong bài toán phân loại, các dữ liệu gồm: *D* là tập dữ liệu huấn luyện đã được véc tơ hóa dưới dạng $\vec{x} = (x_1, x_2, ..., x_n)$, $C_i$ là phân lớp *i*, với $i = \{1, 2, ..., m\}$. Giả thiết, các thuộc tính độc lập điều kiện đôi một với nhau, theo định lý Bayes [65]:

$$P(C_i \mid X) = \frac{P(X \mid C_i) P(C_i)}{P(X)} \quad (1.1)$$

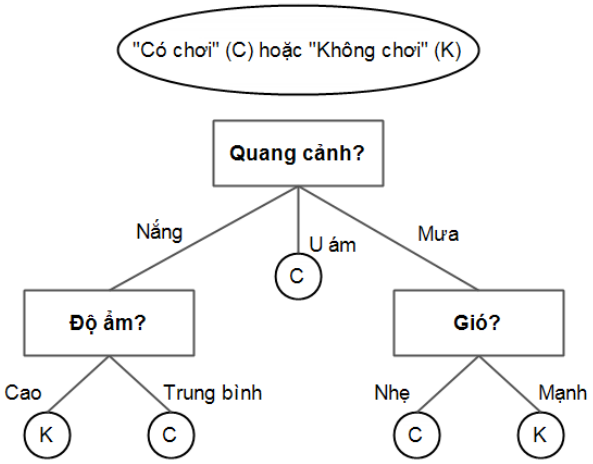
Theo tính chất độc lập điều kiện:

$$P(X \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i) \quad (1.2)$$

Trong đó, $P(C_i \mid X)$ là xác xuất thuộc phân lớp *i* khi biết trước mẫu $X$, $P(C_i)$ là xác xuất phân lớp *i* và $P(x_k \mid C_i)$ là xác xuất thuộc tính *k* mang giá trị $x_k$ khi đã biết *X* thuộc phân lớp *i*.

#### *1.3.2.2. Cây quyết định*

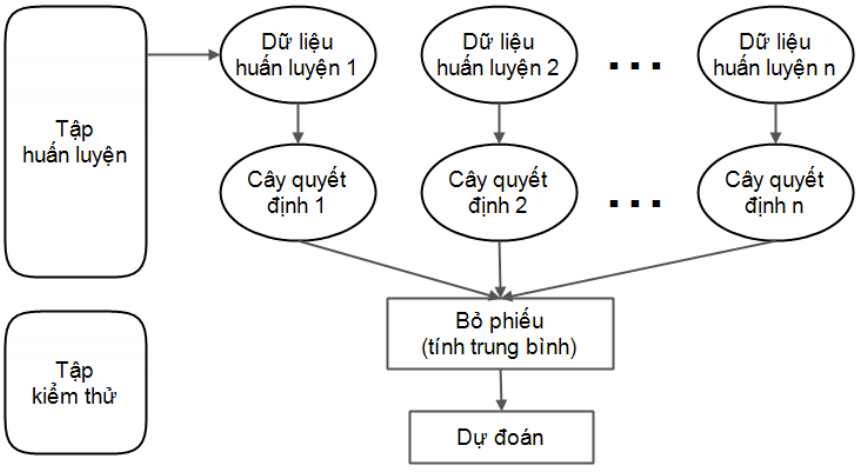
Cây quyết định là một thuật toán học máy có giám sát có thể được áp dụng vào cả hai bài toán phân loại và hồi quy. Việc xây dựng một cây quyết định trên dữ liệu huấn luyện cho trước là việc xác định các câu hỏi và thứ tự của chúng. Một điểm đáng lưu ý của cây quyết định là nó có thể làm việc với các đặc trưng (*các đặc trưng thường được gọi là thuộc tính – attribute*), thường là rời rạc và không có thứ tự. Ví dụ, mưa, nắng hay xanh, đỏ, v..v. Cây quyết định cũng làm việc với dữ liệu có vector đặc trưng bao gồm cả thuộc tính dạng rời rạc và liên tục. Một điểm đáng lưu ý nữa là cây quyết định ít yêu cầu việc chuẩn hoá dữ liệu. Hầu hết các thuật toán cây quyết định đã được phát triển là các biến thể của thuật toán cốt lõi sử dụng tìm kiếm tham lam từ trên xuống trong không gian của các cây quyết định có thể có. Cách tiếp cận này được minh họa bởi thuật toán ID3 (*Quinlan 1986*) và thuật toán kế nhiệm C4.5 (*Quinlan 1993*).




*Hình 1.18: Ví dụ cây ID3*

### *1.3.2.3. Rừng ngẫu nhiên*

Rừng ngẫu nhiên (*Random Forest*) là một thuật toán học máy phổ biến thuộc về nhóm học máy có giám sát. Tương tự cây quyết định, rừng ngẫu nhiên có thể được sử dụng giải quyết hai vấn đề phân loại và hồi quy trong học máy. Rừng ngẫu nhiên dựa trên khái niệm học tập theo nhóm, là một quá trình kết hợp nhiều bộ phân loại để giải quyết một vấn đề phức tạp và để cải thiện hiệu suất của mô hình. Thay vì dựa vào một cây quyết định, rừng ngẫu nhiên lấy dự đoán từ mỗi cây và dựa trên đa số phiếu, và dự đoán kết quả cuối cùng. Số lượng cây lớn hơn trong rừng dẫn đến độ chính xác cao hơn và ngăn ngừa vấn đề quá vừa. Hình 1.19 minh họa hoạt động của thuật toán Rừng ngẫu nhiên [31].



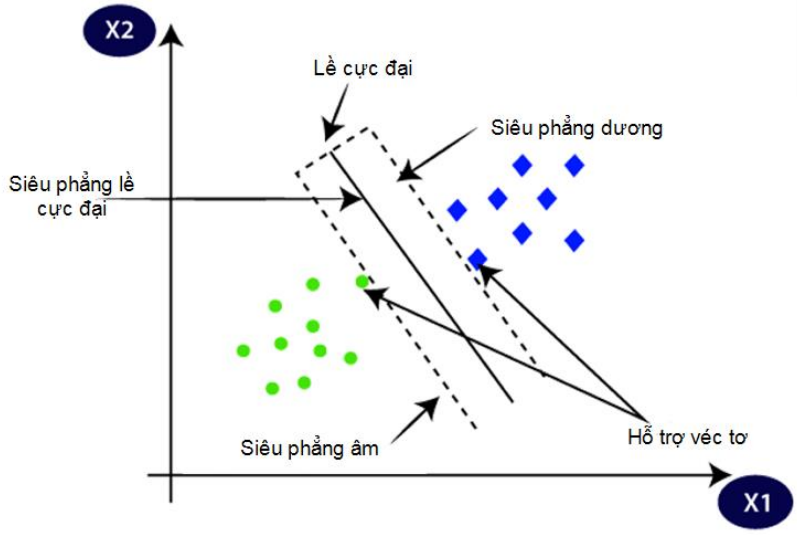

*Hình 1.19: Mô hình thuật toán rừng ngẫu nhiên*

Rừng ngẫu nhiên hoạt động trong hai giai đoạn đầu tiên là tạo ra khu rừng ngẫu nhiên bằng cách kết hợp n cây quyết định, và thứ hai là đưa ra dự đoán cho mỗi cây được tạo ra trong giai đoạn đầu tiên. Quá trình làm việc có thể được giải thích trong các bước sau đây: (i) bước 1: Chọn điểm dữ liệu k ngẫu nhiên từ tập huấn luyện; (ii) bước 2: Xây dựng cây quyết định liên kết với các điểm dữ liệu đã chọn; (iii) bước 3: Chọn số n cho cây quyết định muốn xây dựng; (iv) bước 4: Lặp lại bước 1 và bước 2; (v) bước 5: Đối với các điểm dữ liệu mới, tìm các dự đoán của từng cây quyết định và gán các điểm dữ liệu mới cho danh mục giành được đa số phiếu bầu.

Sử dụng thuật toán Rừng ngẫu nhiên có các ưu điểm sau: (i) Dự đoán đầu ra với độ chính xác cao, ngay cả đối với tập dữ liệu lớn, nó chạy hiệu quả và (ii) Có thể duy trì độ chính xác khi một phần lớn dữ liệu bị khuyết, thiếu.

#### *1.3.2.4. SVM*

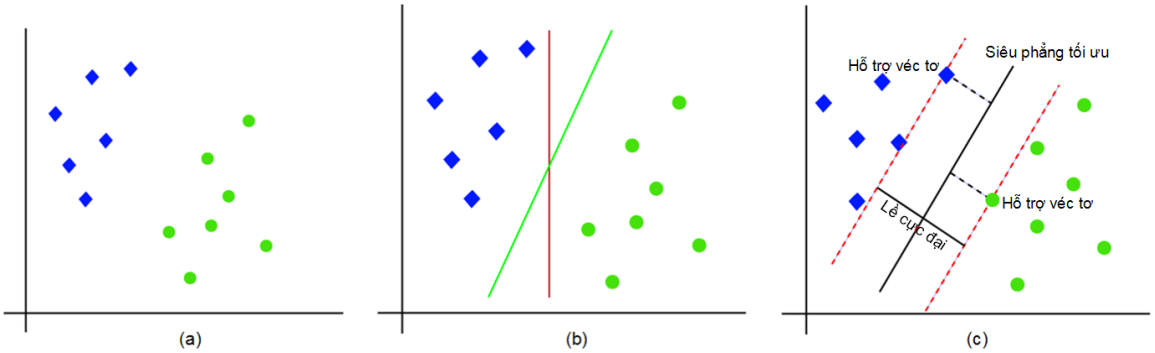
Máy véc tơ hỗ trợ (*SVM*) là một trong những thuật toán học máy có giám sát phổ biến nhất, được sử dụng cho các bài toán phân loại cũng như hồi quy. Tuy nhiên, SVM chủ yếu được sử dụng để giải quyết các bài toán phân loại. Mục tiêu của thuật toán SVM là tạo đường hoặc mặt ranh giới quyết định tốt nhất có thể tách không gian n chiều thành các lớp để có thể dễ dàng đặt điểm dữ liệu mới vào đúng danh mục trong tương lai. Ranh giới quyết định tốt nhất này được gọi là siêu phẳng. Thuật toán SVM chọn các điểm hoặc vectơ cực trị giúp tạo siêu phẳng. Những trường hợp cực đoan này được gọi là véc tơ hỗ trợ. Xem xét **Error! Reference source not found.**, trong đó có hai danh mục khác nhau được phân loại bằng cách sử dụng ranh giới quyết định hoặc siêu phẳng [31]. SVM có thể có hai loại: SVM tuyến tính và SVM phi tuyến tính dựa trên loại ranh giới sử dụng để phân tách các lớp dữ liệu.




*Hình 1.20: Phân loại sử dụng ranh giới trong SVM*

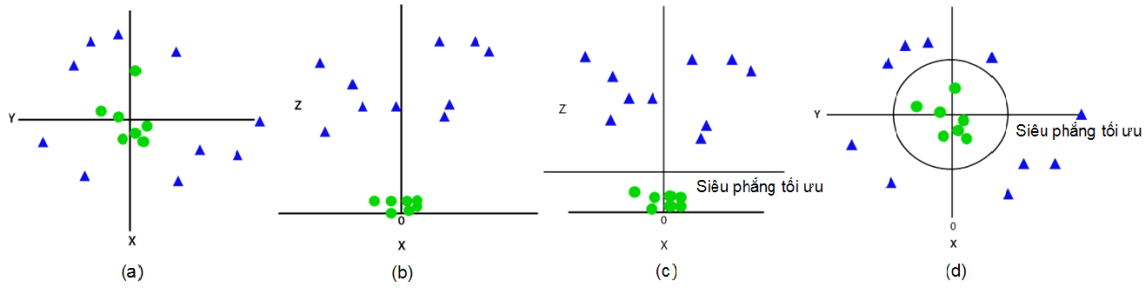
**SVM tuyến tính** được sử dụng cho dữ liệu có thể phân tách tuyến tính, có nghĩa là nếu một tập dữ liệu có thể được phân loại thành hai lớp bằng cách sử dụng một đường thẳng duy nhất, thì dữ liệu đó được gọi là dữ liệu có thể phân tách tuyến tính và bộ phân loại được sử dụng gọi là bộ phân loại SVM tuyến tính.

Hoạt động của thuật toán SVM tuyến tính có thể được hiểu bằng cách sử dụng một ví dụ. Giả sử có một tập dữ liệu có hai thẻ (*xanh lá cây và xanh lam*), và tập dữ liệu có hai đặc điểm $x_1$ và $x_2$. Muốn có một bộ phân loại có thể phân loại cặp tọa độ ($x_1$, $x_2$) theo màu xanh lục hoặc xanh lam Hình 1.21 *(a)*. Vì là không gian 2 chiều nên chỉ cần sử dụng một đoạn thẳng, có thể dễ dàng tách hai lớp này. Nhưng có thể có nhiều dòng cũng có thể phân tách các lớp này Hình 1.21 *(b)*. Do đó, thuật toán SVM giúp tìm đường hoặc ranh giới quyết định tốt nhất; ranh giới hoặc vùng tốt nhất này được gọi là siêu phẳng. Thuật toán SVM tìm điểm gần nhất của các dòng từ cả hai lớp. Những điểm này được gọi là vectơ hỗ trợ. Khoảng cách giữa các vectơ và siêu phẳng được gọi là lề. Và mục tiêu của SVM là tối đa hóa margin. Siêu phẳng với margin tối đa được gọi là siêu phẳng tối ưu Hình 1.21 *(c)*.




*Hình 1.21: Hoạt động của SVM tuyến tính*

**SVM phi tuyến tính** được sử dụng cho dữ liệu được phân tách không theo tuyến tính, có nghĩa là nếu tập dữ liệu không thể được phân loại bằng cách sử dụng một đường thẳng, thì dữ liệu đó được gọi là dữ liệu phi tuyến tính và bộ phân loại được sử dụng được gọi là bộ phân loại SVM phi tuyến tính.

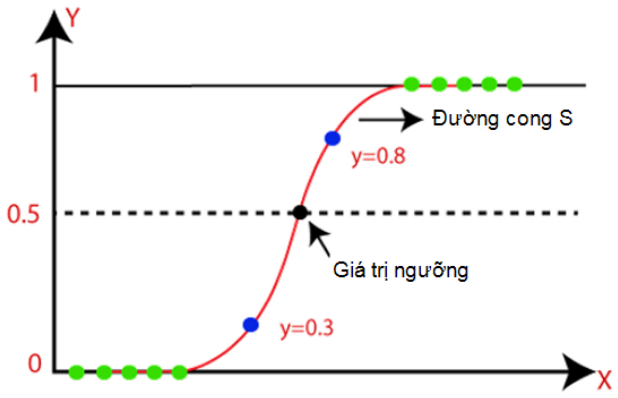



*Hình 1.22: Hoạt động của SVM phi tuyến tính*

Với dữ liệu phi tuyến tính, không thể vẽ một đường thẳng duy nhất để phân tách 2 lớp dữ liệu như trên Hình 1.22 *(a)*. Vì vậy, để tách các điểm dữ liệu này, cần thêm một thứ nguyên nữa. Đối với dữ liệu tuyến tính, sử dụng hai thứ nguyên x và y, vì vậy đối với dữ liệu phi tuyến tính, sẽ thêm thứ nguyên thứ ba z, được tính như sau: $z = x^2 + y^2$. Bằng cách thêm kích thước thứ ba, không gian mẫu sẽ trở thành như Hình 1.22 *(b)*. Và do vậy, SVM sẽ chia các tập dữ liệu thành các lớp theo cách như biểu diễn trên Hình 1.22 *(c)*. Vì đang ở trong không gian 3-D, nó trông giống như một mặt phẳng song song với trục x. Nếu chuyển đổi trong không gian 2-D với z = 1, thì sẽ trở thành Hình 1.22 *(d)*.

### *1.3.2.5. Hồi quy Logistic*

Hồi quy logistic là một trong những thuật toán học máy phổ biến, thuộc nhóm học có giám sát, được sử dụng để dự đoán biến phụ thuộc phân loại bằng cách sử dụng một tập hợp các biến độc lập nhất định. Hồi quy logistic dự đoán đầu ra của một biến phụ thuộc phân loại. Thông thường, kết quả dự đoán là một giá trị phân loại hoặc rời rạc, có thể là có hoặc không, 0 hoặc 1, đúng hoặc sai, v...v. Tuy nhiên, thay vì đưa ra giá trị chính xác là 0 và 1, nó cung cấp các giá trị xác suất nằm giữa 0 và 1. Hồi quy logistic gần giống với hồi quy tuyến tính ngoại trừ cách được sử dụng. Hồi quy tuyến tính được sử dụng để giải các bài toán hồi quy, trong khi hồi quy Logistic được sử dụng để giải các bài toán phân loại. Trong hồi quy logistic, thay vì điều chỉnh một đường hồi quy, điều chỉnh một hàm logistic hình chữ "S", hàm này dự đoán hai giá trị lớn nhất (*0 hoặc 1*). Đường cong từ hàm logistic cho biết khả năng xảy ra một số vấn đề. Hồi quy logistic là một thuật toán học máy quan trọng vì có khả năng cung cấp xác suất và phân loại dữ liệu mới bằng cách sử dụng các bộ dữ liệu liên tục và rời rạc. Hồi quy logistic có thể được sử dụng để phân loại các quan sát bằng cách sử dụng các loại dữ liệu khác nhau và có thể dễ dàng xác định các biến hiệu quả nhất được sử dụng để phân loại [31]. Hình 1.23 minh họa hàm logistic.




*Hình 1.23: Minh họa hàm logistic*

### 1.3.3. Các độ đo đánh giá

Để đánh giá khả năng phát hiện của các mô hình đề xuất trong các Chương 2 và Chương 3, luận án sử dụng sáu độ đo bao gồm: PPV, TPR, FPR, FNR, F1 và ACC. Các độ đo này được định nghĩa như sau:

- Giá trị dự đoán dương tính (*PPV-Positive Predictive Value, hay Precision*), còn gọi là độ chính xác là thước đo trong tất cả các mẫu tên miền DGA botnet dự đoán được đưa ra, có bao nhiêu mẫu dự đoán là chính xác, được tính theo công thức:

$$PPV = \frac{TP}{TP + FP} \quad (1.3)$$

- Tỷ lệ dương tính thật (*TPR-True Positive Rate, hay Recall*) còn gọi là độ nhạy, hay độ bao phủ là thước đo trong tất cả các mẫu tên miền được dự đoán là DGA botnet, có bao nhiêu mẫu đã được dự đoán chính xác, được tính theo công thức:

$$TPR = \frac{TP}{TP + FN} \quad (1.4)$$

- Tỷ lệ dương tính giả (*FPR-False Positive Rate*), hay còn gọi là "nhầm lẫn" là thước đo trong tất cả các mẫu tên miền lành tính, có bao nhiêu mẫu được dự đoán sai là tên miền DGA botnet, được tính theo công thức:

$$FPR = \frac{FP}{FP + TN} \quad (1.5)$$

- Tỷ lệ âm tính giả (*FPR- False Negative Rate*), hay còn gọi "bỏ sót", là thước đo trong tất cả các mẫu tên miền DGA botnet, có bao nhiêu mẫu được dự đoán sai là tên miền lành tính, được tính theo công thức:

$$FNR = \frac{FN}{FN + TP} \quad (1.6)$$

- Độ đo F1 là trung bình điều hòa giữa Precision và Recall. F1 có khuynh hướng lấy giá trị nhỏ hơn giữa Precision và Recall. Nếu chỉ dùng Precision, mô hình chỉ đưa ra dự đoán cho 1 điểm mà nó chắc chắn nhất, khi đó Precision = 1, tuy nhiên không thể nói đây là mô hình tốt. Mặt khác, nếu chỉ dùng Recall, mô hình dự đoán tất cả các điểm đều là dương tính, khi đó Recall = 1, tuy nhiên cũng

không thể nói đây là một mô hình tốt. Do đó, sử dụng F1 để đánh giá sẽ khách quan hơn, được tính theo công thức:

$$F1 = \frac{2TP}{2TP + FP + FN} \quad (1.7)$$

- Độ chính xác toàn cục (ACC) hay độ chính xác chung, là thước đo trong tất cả các mẫu tên miền được đưa vào dự đoán (bao gồm cả tên miền lành tính và tên miền DGA botnet), có bao nhiêu mẫu tên miền lành tính và tên miền DGA botnet được dự đoán chính xác, được tính theo công thức:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (1.8)$$

trong đó, TP là số lượng các tên miền DGA botnet được dự đoán đúng, TN là số lượng các tên miền lành tính được dự đoán đúng, FP là số lượng tên miền lành tính được dự đoán sai thành tên miền DGA botnet và FN là số lượng các tên miền DGA botnet được dự đoán sai thành tên miền lành tính.

Ngoài ra, luận án sử dụng tỷ lệ phát hiện (*DR-Detection Rate*) để đo lường hiệu quả của mô hình phát hiện đề xuất khi dự đoán tên miền của các DGA botnet khác nhau trong quá trình kiểm thử mô hình trong giai đoạn phát hiện. DR cho mỗi loại botnet được tính như sau:

$$DR = \frac{NoDB}{NoTest} \quad (1.9)$$

trong đó, NoDB là số tên miền của một DGA botnet được dự đoán đúng và NoTest là tổng số tên miền của DGA botnet đó khi đưa vào kiểm tra.

## 1.4. CÁC TẬP DỮ LIỆU CHO PHÁT HIỆN BOTNET SỬ DỤNG

### 1.4.1. Tập dữ liệu Netlab360

Netlab 360 [11] là tập dữ liệu chủ yếu được sử dụng trong luận án. Đây là bộ dữ liệu do Network Security Research Lab at 360 cung cấp công khai với hàng triệu mẫu từ nhiều họ DGA được thu thập từ các hệ thống mạng thực tế. Hệ thống phát hiện DGA botnet của Netlab 360 sàng lọc lượng dữ liệu khổng lồ và các mẫu phần

mềm độc hại để tìm các DGA botnet đáng ngờ, mới nhất theo thời gian thực [11]. Nguồn dữ liệu về các họ DGA botnet liên tục được cập nhật từ các cá nhân cũng như các tổ chức nghiên cứu về DGA botnet.

Có thể chia 53 họ DGA botnet do Netlab 360 công bố thành 3 nhóm: nhóm (1) gồm 39 họ botnet sinh tên miền sử dụng tổ hợp các ký tự trong bảng chữ cái, các chữ số và một số ký tự đặc biệt như ".", "-", ".", nhóm (2) gồm 8 họ botnet sinh tên miền sử dụng tổ hợp ký tự hexa (*gồm các ký từ từ 0-9 và A-F*), và nhóm (3) gồm 6 họ botnet sinh tên miền sử dụng tổ hợp các từ tiếng Anh lấy từ từ điển. Các botnet thuộc nhóm (1) và nhóm (2) được gọi chung là *character-based DGA botnet* và các botnet thuộc nhóm (3) được gọi là *word-based DGA botnet*.

Bảng 1.4 cung cấp danh sách 39 character-based DGA botnet sinh tên miền sử dụng các ký tự thông thường. Trong đó, xét về độ dài, có 17 họ tên miền có độ dài từ 6 ký tự đến 34 ký tự, 3 tên miền có 2 mức độ dài cố định, 18 tên miền có độ dài thuộc một khoảng nào đó và có 1 tên độ dài phụ thuộc vào nhân. Xét về số lượng tên miền được sinh ra bởi các họ botnet, một số họ botnet sinh tên miền theo chu kỳ ngày, số khác sinh theo chu kỳ tuần, tháng… Một số botnet khác sinh tên miền 1 lần duy nhất. Đặc biệt, họ *virut* có số lượng lớn nhất với 10,000 tên miền sinh ra một ngày và 5 họ botnet sinh hơn 1,000 tên miền một ngày. Có 26 họ botnet sử dụng các biến thời gian để sinh tên miền, ngược lại 13 họ botnet khác sử dụng thuật toán sinh không phụ thuộc vào thời gian. Xét về việc sử dụng ký tự trong quá trình sinh tên miền, có 19 họ botnet sinh tên miền sử dụng các ký tự từ a đến z, 10 họ botnet sinh tên miền sử dụng các ký tự từ a đến y, 6 họ botnet sinh tên miền sử dụng các ký tự từ a đến z kết hợp với các ký tự số từ 0 đến 9 và 2 họ botnet sinh tên miền sử dụng tập con của bảng chữ cái abc (cụ thể là *mydoom* sử dụng tập ký tự [aehmnpqrsw] và *padcrypt* sử dụng tập ký tự [abcdefklmno]).

Bảng 1.5 liệt kê các character-based DGA botnet sinh tên miền sử dụng các ký tự hexa. Tất cả các tên miền đều có độ dài cố định từ 8 đến 32 ký tự, trong đó có 2 họ botnet sinh tên miền với độ dài lớn nhất là *bamital* và *omexo*, đặc biệt *bamital* được

sinh ra như là giá trị băm MD5. Lượng tên miền sinh ra trong ngày của nhóm (2) là không nhiều, tuy nhiên họ botnet *moneromine*r sinh 2,495 tên miền một ngày. Có 5 họ botnet sử dụng các biến thời gian để sinh tên miền và 3 họ botnet khác sử dụng thuật toán sinh tên miền không phụ thuộc vào thời gian.

Bảng 1.6 liệt kê các word-based DGA botnet. Theo đó, các botnet thuộc nhóm này lấy ngẫu nhiên trong các từ điển tiếng Anh của các họ botnet để tổ hợp sinh các tên miền. Độ dài tên miền sinh phụ thuộc vào độ dài của từ, thường là sự kết hợp ngẫu nhiên của 2-3 từ, có hoặc không có ký tự nối *("-" hoặc ".")*. Trong word-based DGA botnet, có 3 họ botnet sử dụng thuật toán sinh tên miền không phụ thuộc vào thời gian và 3 họ còn lại sử dụng các biến thời gian để sinh tên miền.

Các tập dữ liệu tên miền DGA botnet được trích xuất từ nguồn Netlab 360 kết hợp với tập dữ liệu tên miền lành tính trích xuất từ 1,000,000 tên miền theo bảng xếp hạng của Alexa [17] được sử dụng để xây dựng và kiểm thử các mô hình phát hiện DGA botnet đề xuất trong luận án.

*Bảng 1.4: Các họ botnet sinh tên miền sử dụng ký tự a-z, 0..9 (character-based DGA botnet) [11]*

| STT | Họ botnet | Độ dài | Số lượng | TD | Ký tự sử dụng | Ví dụ |
|---|---|---|---|---|---|---|
| 1 | abcbot | 9 | 27 /tháng | x | a..z | nerjyzkup.com |
| 2 | blackhole | 16 | 2 /ngày | x | a..z | mkjdkbwuxcnuxtqd.ru |
| 3 | chinad | 16 | 1,000 /ngày | x | a..z và 0..9 | qowhi81jvoid4j0m.biz |
| 4 | conficker | 5 - 11 | 250 /ngày | x | a..z | ydqtkptuwsa.org |
| 5 | cryptolocker | 12 - 15 | 1,000 /tuần | x | a..y | nvjwoofansjbh.ru |
| 6 | dircrypt | 8 - 20 | 30 | | a..z | mycojenxktsmozzthdv.com |
| 7 | dyre | 34 | 1,000 /ngày | x | 1 [a..z] + 33 ký tự SHA256 | l54c2e21e80ba5471be7a8402cffb98768.so |
| 8 | emotet | 16 | 96 /ngày | x | a..y | grdawgrcwegpjaoo.eu |
| 9 | feodo | 16\|18 | 97\|65 | | a..z | dvyfuaopltfxjzsp.ru |
| 10 | flubot | 15 | 5,000 /tháng | x | a..y | bsgejiagbavgavk.cn |
| 11 | fobber | 17\|10 | 300 | | a..z | zzwzzqmihkfdevymi.net |
| 12 | gameover | 20 - 28 | 1,000 /ngày | x | a..z và 0..9 | 14dtuor1aubbmjhgup7915tlinc.net |
| 13 | locky | 5-17 | 6\|12 / 2 ngày | x | a..y | lpfpdovapot.ru |
| 14 | madmax | 10 | 1 / tuần | x | a..z và 0..9 | s82r4luxrw.com |
| 15 | mirai | 12 | 1 /ngày | x | a..y | xvrvdsuhphjg.online |
| 16 | murofet | 8 - 16 | 1,020 /ngày | x | a..z | uqiqvqylwlhutwvh.info |
| 17 | mydoom | 10 | 51 /ngày | x | ["aehmnpqrsw"] | wmhmqsqsqa.in |
| 18 | necro | 16 | 2,048 | | a..z | aveixucyimxwcmph.xyz |
| 19 | necurs | 7 - 21 | 2,048 /3 ngày | x | a..y | otenbmgbpuskiasvehxm.ki |

| STT | Họ botnet | Độ dài | Số lượng | TD | Ký tự sử dụng | Ví dụ |
|---|---|---|---|---|---|---|
| 20 | nymaim | 5 - 12 | 30\|128 /ngày | x | a..z | zzayzoabsi.net |
| 21 | padcrypt | 16 | 24\|72 /ngày | x | [abcdefklmno] | adbbfbdnddbodacd.online |
| 22 | proslikefan | 6 - 13 | 100 /ngày | x | a..z | nuipkjqarq.in |
| 23 | pykspa | 6 - 15 | 5,000 /2 ngày | x | a..z | ynrvwgfqbex.org |
| 24 | qadars | 12 | 200 /tuần | x | a..z và 0..9 | lmjwd6fs9ur4.com |
| 25 | ramnit | 8 - 19 | 1,000 | | a..y | jrkaxdlkvhgsiyknhw.com |
| 26 | ranbyus | 14\|17 | 40 /ngày | x | a..y | nslxbdyiofityx.com |
| 27 | rovnix | 18 | 10,000 | | a..z và 1..8 | aby71fqwc3ai12wseh.com |
| 28 | shifu | 7 | 1,000 | | a..y | nqqxqdg.info |
| 29 | shiotob | 10 - 15 | 2,000 | | a..z và 1..5,9 | wwdnioqno3p9k2x.com |
| 30 | simda | | 1,500 – 2,500 | | a..z | digivehusyd.eu |
| 31 | symmi | 8 - 15 | 64 / 15 ngày | x | a..z | ukbounapimusamx.ddns.net |
| 32 | tempedreve | 7 - 11 | 204 | | a..z | ahskjnrhueg.net |
| 33 | tinba | 12 | 100\|200\|1,000 | | a..y | nvfowikhevmy.com |
| 34 | tofsee | 7 | 20 / tuần | x | a..z | dqhdqhd.biz |
| 36 | vawtrak | 7 - 11 | 150 | | a..z | kdcbwvehop.top |
| 37 | vidro | 7 - 12 | 100 / tuần | x | a..z | ckdypldcxi.com |
| 38 | virut | 6 | 10,000 /ngày | x | a..z | yvvioe.com |
| 39 | xshellghost | 10 - 15 | 1 / tháng | x | a..z | huxerorebmzir.com |

*Bảng 1.5: Các họ botnet sinh tên miền sử dụng ký tự Hexa*

| STT | Họ botnet | Độ dài | Số lượng | TD | Ký tự sử dụng | Ví dụ |
|---|---|---|---|---|---|---|
| 1 | antavnu | 8 | 16 /ngày | x | ký tự hexa | 2c2b3749.com |
| 2 | bamital | 32 | 26 /ngày | x | như MD5 | cd8f66549913a78c5a8004c82bcf6b01.info |
| 3 | copperstealer | 16 | 11 /tháng | x | ký tự hexa | 1cd81defbab5fc17.xyz |
| 4 | enviserv | 10 | 500 | | ký tự hexa | 9dcd84b090.net |
| 5 | gspy | 16 | 50 | | ký tự hexa | 484b072f94637588.net |
| 6 | monerominer | 13 | 2,495/ ngày | x | ký tự hexa | 5c95f79304b49.org |
| 7 | omexo | 32 | 20 | | ký tự hexa | eef795a4eddaf1e7bd79212acc9dde16.net |
| 8 | tordwm | 8 | 510 /ngày | x | ký tự hexa | f0fe6744.top |

*Bảng 1.6: Các họ botnet word-based DGA*

| STT | Họ botnet | Độ dài | Số lượng | TD | Ký tự sử dụng | Ví dụ |
|---|---|---|---|---|---|---|
| 1 | banjori | | 2,196\|15,372 | | Thay đổi 4 ký tự từ đầu kết hợp với domain từ nhân | **earn**estnessbiophysicalohax.com<br>**kwto**estnessbiophysicalohax.com |
| 2 | bigviktor | | 1,000/ tháng | x | Kết hợp 3-4 từ danh sách định nghĩa trước. Có hoặc không có dấu "-" và "." | support.showremote-conclusion.fans<br>turntruebreakfast.futbol |
| 3 | kfos | | 135 | | 2 từ nối với nhau bằng dấu "-" | service-goole.tw<br>mails-googles.ml |
| 4 | matsnu | | 10/ ngày | x | Kết hợp 2-3 từ các danh sách định nghĩa trước. Có hoặc không có dấu "-". | world-bite-care.com<br>activitypossess.com |
| 5 | ngioweb | | 300\|1024 | | Kết hợp từ từ 3 danh sách định nghĩa trước. Có hoặc không có dấu "-". | overecobism-revacidom-ultrasaxeship.org<br>rexarurission.biz |
| 6 | suppobox | | 254\|782 /ngày | x | Kết hợp 2 từ từ danh sách định nghĩa trước. | sharmainewestbrook.net<br>stephaniebernadine.ru |

### 1.4.2. Các tập dữ liệu khác được sử dụng

Ngoài các dữ liệu về botnet từ hai tập dữ liệu đã trình bày ở trên, dữ liệu DGA botnet được bổ sung từ bộ sưu tập 33 DGA botnet của tác giả Johannes Bader (*bao gồm cả mã nguồn các thuật toán sinh*) [2]. Trong tập dữ liệu này có 1 số họ botnet giống trong tệp dữ liệu của Netlab360, tuy nhiên luận án có sử dụng tập tên miền DGA từ bộ dữ liệu này để tăng số lượng tên miền của một số họ botnet sử dụng trong các mô hình huấn luyện.

Để có được kết quả đánh giá một cách tổng quát, trong luận án sử dụng bộ dữ liệu UMUDGA của Universidad de Murcia [60]. Bộ dữ liệu có hơn 30 triệu tên miền được tạo theo thuật toán được gắn nhãn thủ công sẵn sàng sử dụng cho phân tích học máy. Tập dữ liệu được đề xuất này cho phép các nhà nghiên cứu tiến tới các giai đoạn thu thập, tổ chức và tiền xử lý dữ liệu, cuối cùng cho phép tập trung vào việc phân tích và sản xuất các giải pháp hỗ trợ học máy để phát hiện xâm nhập mạng. Từ bộ dữ liệu này sẽ chọn ra một số họ botnet chưa được công bố trên Netlab360 để thử nghiệm phát hiện dựa ra các mô hình sẽ được đề xuất ở chương 2 và chương 3.

Tập dữ liệu các tên miền lành tính được lấy top 1 triệu tên miền của Alexa [17]. Các tên miền được lược bỏ TLD, chỉ lấy phần SLD và loại bỏ các tên miền trùng nhau (*có TLD khác nhau*). Luận án sử dụng 110,000 tên miền đầu tiên có thứ hạng cao nhất trong tệp dữ liệu này để xây dựng và kiểm thử các mô hình phát hiện DGA botnet đề xuất.

## 1.5. HƯỚNG NGHIÊN CỨU CỦA LUẬN ÁN

### 1.5.1. Ưu điểm và nhược điểm của các kỹ thuật phát hiện botnet

Bảng 1.7 tổng hợp các ưu điểm và nhược điểm của các kỹ thuật phát hiện botnet.

*Bảng 1.7: Ưu nhược điểm của các kỹ thuật phát hiện botnet*

| Kỹ thuật | Ưu điểm | Nhược điểm |
|---|---|---|
| Phát hiện dựa trên Honeynet | Đơn giản trong triển khai, ít yêu cầu về nguồn lực, chi phí | Khó mở rộng, nhiều thách thức khi giám sát các dạng botnet và các dạng tấn công có liên quan, có khả năng bị vô hiệu hóa. |

| | | |
|---|---|---|
| | triển khai tối thiểu và hữu dụng với dữ liệu mã hoá. | |
| Phát hiện dựa trên luật, dấu hiệu | Có khả năng phát hiện nhanh và chính xác các bot và botnet đã biết. | Không có khả năng phát hiện các bot và botnet mới, cần thường xuyên cập nhật cơ sở dữ liệu dấu hiệu, chữ ký. |
| Phát hiện dựa trên bất thường | Có khả năng phát hiện các dạng bot, botnet mới, có khả năng tự động hóa việc xây dựng mô hình phát hiện. | Tỷ lệ cảnh báo sai thường cao hơn so với phát hiện dựa trên luật, dấu hiệu; đòi hỏi tài nguyên tính toán lớn hơn cho xây dựng mô hình và giám sát phát hiện. |

### 1.5.2. Các vấn đề giải quyết trong luận án

Từ việc phân tích mô hình hoạt động và các tác hại của các dạng botnet nói chung và DGA botnet nói riêng, việc nghiên cứu các giải pháp, kỹ thuật phát hiện botnet, DGA botnet là rất cấp thiết. Luận án cũng đã nghiên cứu, khảo sát các kỹ thuật phát hiện botnet dựa trên Honeynet, dựa trên dấu hiệu, luật, dựa trên bất thường và một số giải pháp, công cụ cho giám sát và phát hiện các dạng botnet. Mỗi phương pháp, giải pháp có các ưu điểm và nhược điểm riêng như đã chỉ ra trong mục 1.5.1.

Hướng nghiên cứu của luận án là sử dụng phương pháp phát hiện bot, botnet dựa trên bất thường do phương pháp này có khả năng phát hiện các dạng bot, botnet mới, đồng thời có khả năng tự động hóa việc xây dựng mô hình phát hiện. Trên cơ sở khảo sát, phân tích các ưu điểm và hạn chế của các đề xuất đã có, luận án tập trung nghiên cứu, giải quyết các vấn đề sau: (1) nghiên cứu, đề xuất tập đặc trưng phân loại tên miền mới phù hợp hơn cho xây dựng các mô hình phát hiện DGA botnet, nhằm tăng tỷ lệ phát hiện đúng và giảm tỷ lệ cảnh báo sai và (2) nghiên cứu, lựa chọn sử dụng phương pháp học máy phù hợp cho xây dựng các mô hình phát hiện DGA botnet, nhằm xây dựng một mô hình phát hiện thống nhất cho phép phát hiện hiệu quả nhiều dạng DGA botnet. Vấn đề (1) là do tập đặc trưng phân loại tên miền sử dụng trong các đề xuất đã có chưa thực sự phù hợp để phân biệt các tên miền DGA với các tên miền lành tính dẫn đến tỷ lệ cảnh báo sai còn tương đối cao. Vấn đề (2) xuất phát từ thực tế là mỗi đề xuất đã có chỉ có khả năng phát hiện hiệu quả một số

họ DGA botnet, hoặc trên một tập dữ liệu cụ thể, mà không thể phát hiện hiệu quả nhiều dạng DGA botnet.

## 1.6. KẾT LUẬN CHƯƠNG

Botnet đã và đang trở thành một trong những mối đe dọa an ninh chính cho các cơ quan, tổ chức, doanh nghiệp và người dùng Internet. Do vậy, nghiên cứu phát triển các kỹ thuật và giải pháp hiệu quả cho giám sát, phát hiện botnet là việc cấp thiết. Chương 1 giới thiệu tổng quan về botnet, vấn đề phát hiện botnet, khái quát về học máy và các giải thuật học máy sử dụng cho phát hiện botnet và các tập dữ liệu sử dụng trong luận án. Cụ thể, trong phần đầu chương trình bày khái quát về botnet và phương thức hoạt động của chúng, phân loại botnet dựa trên kiến trúc mạng và giao thức truyền thông, vắn tắt về lịch sử phát triển của botnet và tác hại cũng như các dạng khai thác botnet.

Một trong các nội dung chính được trình bày trong chương này là vấn đề phát hiện botnet. Luận án phân tích 3 hướng phát hiện botnet được sử dụng phổ biến bao gồm: phát hiện dựa trên honeynet, phát hiện dựa trên luật, dấu hiệu và phát hiện dựa trên dựa trên bất thường, đồng thời tổng hợp các ưu và nhược điểm của 3 hướng trên làm cơ sở cho hướng nghiên cứu của luận án. Tiếp theo luận án mô tả 3 giải pháp giám sát và phát hiện botnet đã được triển khai trên thực tế, bao gồm BotHunter, BotSniffer và BotTrack.

Trong hướng phát hiện botnet dựa trên dựa trên bất thường, việc ứng dụng học máy trong xây dựng các mô hình và giải pháp phát hiện botnet ngày càng được quan tâm do học máy có thể ứng dụng để tự động hóa việc xây dựng mô hình hoặc hồ sơ phát hiện. Điều này giúp giảm đáng kể yêu cầu nhân lực chuyên gia cho xây dựng tập luật, dấu hiệu theo phương pháp thủ công. Để phục vụ cho việc ứng dụng học máy trong các mô hình phát hiện botnet đề xuất trong chương 2 và chương 3, chương này trình bày khái quát về học máy, tập trung mô tả các thuật toán học máy có giám sát truyền thống, bao gồm Naive Bayes, cây quyết định, rừng ngẫu nhiên, SVM và hồi

qui logistic. Chương cũng mô tả các độ đo đánh giá các mô hình phát hiện DGA botnet dựa trên học máy đề xuất trong luận án.

Phần tiếp theo của chương 1 trình bày về các tập dữ liệu sử dụng trong luận án, bao gồm tập dữ liệu Netlab 360, CTU-13 và tập dữ liệu tên miền lành tính từ nguồn Alexa. Đây là các tập dữ liệu tên miền do các DGA botnet sinh ra, được thu thập từ nhiều nguồn. Từ các tập dữ liệu gốc, luận án xây dựng tập dữ liệu chung, gồm tập tên miền DGA và tập tên miền lành tính sử dụng trong các mô hình phát hiện botnet dựa trên học máy đề xuất trong luận án ở các chương 2 và chương 3.

Phần cuối của chương 1 nêu 2 vấn đề chính được tập trung giải quyết trong các chương 2 và 3 của luận án.

# CHƯƠNG 2: PHÁT HIỆN DGA BOTNET DỰA TRÊN HỌC MÁY SỬ DỤNG CÁC ĐẶC TRƯNG KÝ TỰ VÀ TỪ

## 2.1. DGA BOTNET VÀ CƠ CHẾ KHAI THÁC HỆ THỐNG DNS

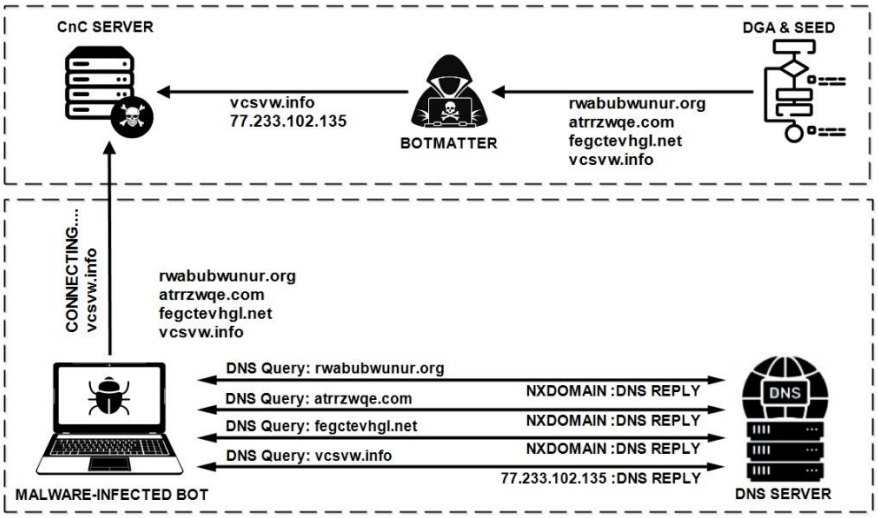
### 2.1.1. Khái quát về DGA botnet

#### *2.1.1.1. Giới thiệu về DGA botnet*

DGA botnet là các họ botnet sử dụng kỹ thuật DGA (*Domain Generation Algorithm*) để sinh và đăng ký nhiều tên miền ngẫu nhiên khác nhau cho máy chủ chỉ huy và điều khiển CnC của chúng nhằm chống lại việc bị kiểm soát và đưa vào danh sách đen [15] [36]. Các botnet dạng này còn được gọi là DGA-based botnet, hay ngắn gọn là DGA botnet. Các DGA botnet sử dụng thuật toán DGA để định kỳ sinh và đăng ký một lượng lớn tên miền giả ngẫu nhiên mà chúng được phân giải thành địa chỉ IP của máy chủ CnC của botnet. Lý do chính của việc sử dụng DGA là làm phức tạp việc kiểm soát và thu hồi tên miền. Nếu botnet sử dụng một tên miền tĩnh cho máy chủ CnC của nó, việc kiểm soát và thu hồi tên miền có thể được thực hiện dễ dàng thông qua việc phối hợp với bên quản lý tên miền gốc để chỉnh sửa các bản ghi tên miền trên máy chủ DNS. Tuy nhiên, khi DGA được sử dụng để sinh các tên miền động, việc kiểm soát và thu hồi các tên miền sẽ trở nên rất khó khăn. Do các bot sử dụng một tên miền mới được tạo ra sau một giai đoạn để kết nối đến máy chủ CnC, việc kiểm soát các tên miền đã hết hạn sử dụng không có ý nghĩa.

Hình 2.1 minh họa cơ chế botnet sử dụng DGA để tự động sinh và đăng ký các tên miền cho máy chủ CnC. Theo đó, botmaster và các bot cùng sử dụng một thuật toán DGA với cùng một nhân (*seed*) nên chúng có thể sinh ra cùng một tập các tên miền. DGA botnet thường sử dụng ngày giờ như là nhân để khởi tạo thuật toán sinh tên miền và như vậy DGA botnet tạo một tập các tên miền trong mỗi ngày nó hoạt động. Ở phía máy chủ, botmaster định kỳ sinh và đăng ký các tên miền cho máy chủ CnC của botnet thông qua hệ thống DNS động sử dụng một thuật toán DGA. Để khởi tạo kết nối đến máy chủ CnC ở phía các bot, mỗi bot sử dụng cùng thuật toán DGA để sinh một tên miền và truy vấn hệ thống DNS cục bộ để phân giải tên miền thành

địa chỉ IP của máy chủ CnC. Nếu quá trình phân giải tên miền thành công, bot sử dụng địa chỉ IP để kết nối đến máy chủ CnC để nhận các lệnh và mã cập nhật. Nếu quá trình phân giải tên miền không thành công, bot lại sử dụng thuật toán DGA để sinh một tên miền mới và lặp lại yêu cầu phân giải địa chỉ.




*Hình 2.1: Cơ chế botnet sử dụng DGA để sinh và đăng ký cho máy chủ CnC*

Có nhiều thuật toán DGA được các botnet sử dụng để sinh các tên miền. Một só thuật toán DGA sinh tên miền bằng cách ghép ngẫu nhiên các ký tự, một số khác lại sử dụng phương pháp ghép các từ theo một qui luật nào đó. Có một số botnet sử dụng thuật toán DGA sinh các tên miền phụ thuộc thời gian. DGA có thể sử dụng các phép toán kết hợp với các biến luôn thay đổi, chẳng hạn như năm, tháng, ngày, giờ, phút để sinh tên miền ngẫu nhiên. Ví dụ, một dạng của thuật toán DGA được thực hiện bởi 1 hàm có chứa 16 vòng lặp. Mỗi vòng lặp sinh ngẫu nhiên 1 ký tự trong tên miền như sau [15]:

```
-year = ((year ^ 8 * year) >> 11) ^ ((year & 0xFFFFFFF0) << 17)
-month = ((month ^ 4 * month) >> 25) ^ 16 * (month & 0xFFFFFFF8)
-day = ((day ^ (day << 13)) >> 19) ^ ((day & 0xFFFFFFFE) << 12)
-domain += chr(((year ^ month ^ day) % 25) + 97)
```

Trong đó*: (i) toán tử '^': sao chép bit nếu nó được đặt (chỉ bit 1) chỉ trong một toán hạng; (ii) toán tử '*': phép nhân và gán; (iii) toán tử '>>': dịch phải bit; (iv) toán tử '&': phép AND; (v) toán tử '<<': dịch trái bit.*

Ngoài ra, DGA hỗ trợ DNS fluxing là kỹ thuật cho phép tên miền có thể được đăng ký và hủy đăng ký nhanh chóng. Nhờ vậy, địa chỉ IP của nhiều tên miền sinh tự động được thường xuyên thay đổi bởi kẻ tấn công để tránh bị nằm trong IP Blacklist.

#### *2.1.1.2. Các loại DGA botnet*

Theo tập ký tự được sử dụng để tự động sinh tên miền, các DGA botnet có thể được chia thành 3 dạng chính [11]: character-based DGA botnet, word-based DGA botnet và mixed DGA botnet. Character-based DGA botnet là các DGA botnet được biết đến nhiều nhất. Phương pháp sinh tên miền của character-based DGA botnet là kết hợp các ký tự được chọn ngẫu nhiên thông qua các thuật toán. Các tên miền của họ botnet này thường được sinh bằng cách ghép ngẫu nhiên các ký tự trong bảng chữ cái tiếng Anh (a-z), các số (0-9) và các chữ số trong hệ Hexa-decimal (0-9, a-f). Mỗi họ character-based DGA botnet lại sinh tên miền với độ dài khác nhau. Bảng 2.1 liệt kê một số họ character-based DGA botnet và mẫu tên miền tự sinh.

*Bảng 2.1: Một số họ character-based DGA botnet*

| STT | Họ botnet | Ký tự sử dụng | Ví dụ |
|---|---|---|---|
| 1 | blackhole | “a..z” | mkjdkbwuxcnuxtqd.ru |
| 2 | bamital | ký tự Hexa | cd8f66549913a78c5a8004c82bcf6b01.info |
| 3 | ccleaner | ký tự Hexa | ab1145b758c30.com |
| 4 | cryptocloker | “a..y” | eqmbcmgemghxbcj.co.uk |
| 5 | chinad | “a..z” và “0..9” | 29cqdf6obnq462yv.com |
| 6 | dyre | ký tự Hexa | jaa12148a5831a5af92aa1d8fe6059e276.ws |
| 7 | gameover | “a..z” và “0..9” | 2id0lapmam6w1799w7315zaqj5.com |
| 8 | mirai | “a..y” | xvrvdsuhphjg.online |
| 9 | mydoom | "aehmnpqrsw" | swaqwnhnhs.biz |
| 10 | rovnix | “a..z” và “0..9” | rc7thuhy8agn43zzgi.biz |
| … | … | … | … |

Word-based DGA botnet là các DGA botnet sử dụng kỹ thuật sinh tên miền dựa trên tổ hợp các từ tiếng Anh lấy từ các từ điển. Điều này là do các tên miền sinh bởi các character-based DGA botnet thường dễ bị phát hiện bởi vì đa số các tên miền sinh bởi các botnet dạng này là tổ hợp ngẫu nhiên các ký tự, thường không có nghĩa và dễ phân biệt so với các tên miền lành tính do con người lựa chọn. Các word-based DGA

botnet thường xây dựng các từ điển riêng biệt để sử dụng, có thể là từ điển danh từ, động từ hoặc tính từ. Mỗi họ word-based DGA botnet thường sử dụng phương pháp riêng để tổ hợp các từ từ các từ điển để sinh tên miền. Nhờ vậy, các tên miền sinh bởi các word-based DGA botnet là tổ hợp của các từ có nghĩa nên trông giống các tên miền lành tính hơn.

*Bảng 2.2: Một số họ word-based DGA botnet*

| STT | Họ botnet | Từ sử dụng | Ví dụ |
|---|---|---|---|
| 1 | bigviktor | Nối 3-4 từ trong 4 từ điển | showremote-conclusion.fans<br>turntruebreakfast.futbol |
| 2 | matsnu | Nối 2-3 từ trong 2 từ điển | world-bite-care.com<br>activitypossess.com |
| 3 | ngioweb | Nối từ trong 3 từ điển | overecobism-revacidom-ultrasaxeship.org<br>rexarurission.biz |
| 4 | suppobox | Nối 2 từ trong 1 từ điển (*có 3 từ điển*) | sharmainewestbrook.net (*từ điển 3*)<br>tablethirteen.net (*từ điển 2*)<br>childrencatch.net (*từ điển 1*) |

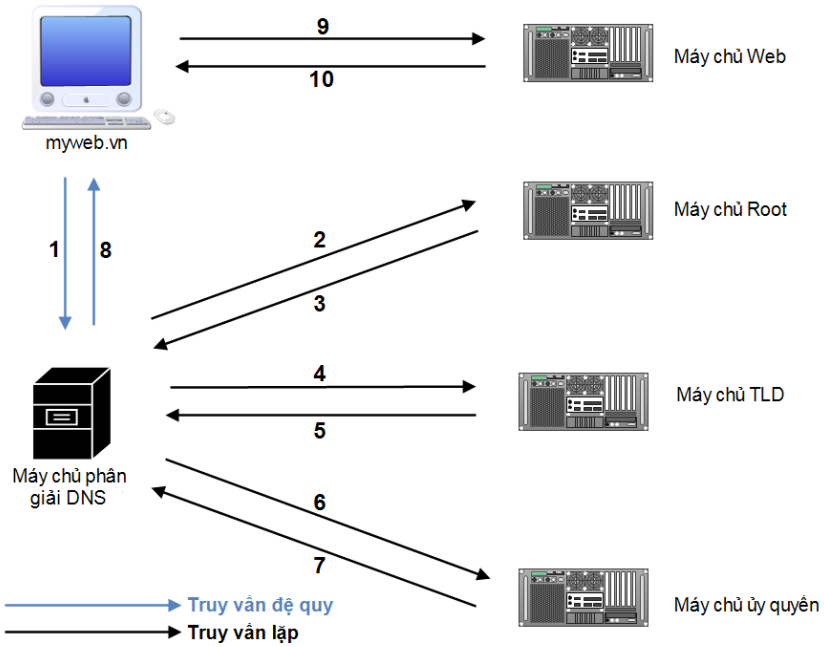
Mixed DGA botnet là dạng botnet sử dụng kỹ thuật lai giữa character-based DGA botnet và word-based DGA botnet để tăng độ khó cho việc phân biệt giữa tên miền sinh bởi botnet và tên miền lành tính. Mỗi tên miền của mixed DGA botnet thường gồm 2 phần: một phần là tổ hợp ngẫu nhiên các ký tự (*character-based*) và phần còn lại là tổ hợp các từ có nghĩa (*word-based*). Điển hình là họ *banjori* botnet, sử dụng một domain lõi sau đó lấy ngẫu nhiên 4 ký tự làm tiền tố cho domain lõi, như "**earn**estnessbiophysicalohax.com", "**kwto**estnessbiophysicalohax.com".

### 2.1.2. Cơ chế DGA botnet khai thác hệ thống DNS

#### *2.1.2.1. Giới thiệu hệ thống DNS*

Hệ thống tên miền (*DNS – Domain Name System*) có thể được xem như một danh bạ khổng lồ của mạng Internet cho truy cập thông tin trực tuyến thông qua các tên miền. Để truy cập 1 trang web, trình duyệt web trước hết truy vấn hệ thống DNS để tìm địa chỉ IP của máy của web. DNS dịch tên miền thành địa chỉ IP và chuyển lại cho trình duyệt để trình duyệt kết nối và tải trang web cho người dùng.

Quá trình phân giải DNS bao gồm việc chuyển đổi tên máy chủ (*chẳng hạn như example.com*) thành địa chỉ IP của máy tính (*chẳng hạn như 200.168.32.16*). Một địa chỉ IP được cung cấp cho mỗi thiết bị trên Internet và địa chỉ đó là cần thiết để tìm thiết bị Internet phù hợp - giống như một địa chỉ đường phố được sử dụng để tìm một ngôi nhà cụ thể. Khi người dùng muốn tải một trang web, hệ thống sẽ chuyển đổi những gì người dùng nhập vào trình duyệt web của họ (*example.com*) thành địa chỉ IP của máy chủ vận hành trang web example.com. Đối với trình duyệt web, việc tra cứu DNS xảy ra trong hậu trường và không yêu cầu sự tương tác bổ sung nào từ máy tính của người dùng, ngoài yêu cầu ban đầu. Hình 2.2 mô tả quá trình truy cập một trang web, bao gồm việc phân giải một tên miền thành địa chỉ IP của máy chủ web.

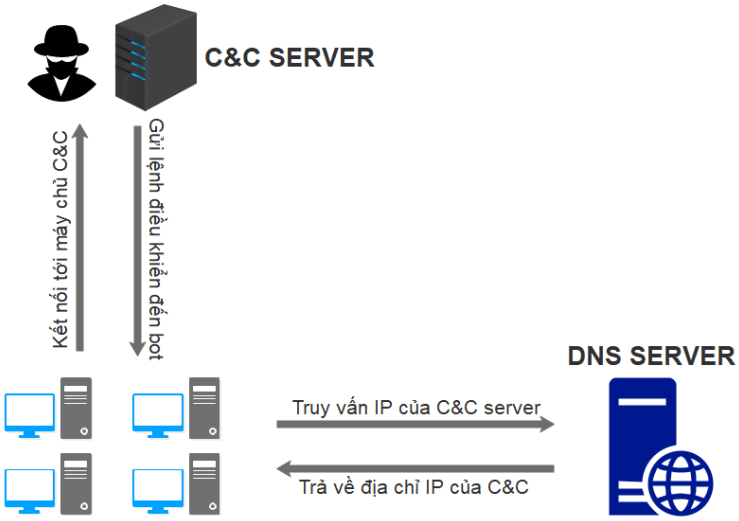



*Hình 2.2: Quá trình phân giải tên miền*

Trong hầu hết các tình huống, DNS liên quan đến một tên miền được dịch sang địa chỉ IP thích hợp. Tìm hiểu cách thức hoạt động của quy trình này, giúp theo dõi quá trình tra cứu DNS khi di chuyển từ trình duyệt web, thông qua quy trình tra cứu DNS và trả về. Lưu ý rằng,  thông thường thông tin tra cứu DNS sẽ được lưu trong bộ nhớ cache bên trong máy tính truy vấn hoặc từ xa trong cơ sở hạ tầng DNS. Khi thông tin DNS được lưu trong bộ nhớ cache, các bước được bỏ qua khỏi quy trình tra cứu DNS giúp việc này nhanh hơn.

### *2.1.2.2. Cơ chế DGA botnet khai thác hệ thống DNS*

Trong hầu hết các dạng kiến trúc botnet sử dụng, botmaster liên lạc và kiểm soát các bot sử dụng một mạng lưới các máy chủ trung gian, hay các máy chủ CnC. Các bot được cấu hình để định kỳ liên lạc với máy chủ CnC thông qua các giao thức truyền thông như IRC, hoặc HTTP để nhận lệnh và mã cập nhật. Do vậy, các máy chủ CnC cần có địa chỉ IP công cộng (*public IP*), hoặc tên miền để phân giải sang một địa chỉ IP nhất định [24] để có thể thực hiện việc truyền thông với các bot.




*Hình 2.3: DGA botnet khai thác hệ thống DNS*

Nếu như các máy chủ CnC sử dụng các địa chỉ IP công cộng thì việc bị phát hiện là khá đơn giản. Người quản trị có thể dễ dàng phát hiện ra những địa chỉ IP đáng ngờ liên quan đến lượng lớn lưu lượng mạng nhưng không xuất hiện trong truy vấn DNS. Điều này là đáng ngờ do người dùng bình thường không có khả năng hoặc rất khi nhớ địa chỉ IP để truy cập trực tiếp. Để lẩn tránh việc rà quét, phát hiện các máy chủ CnC, botmaster liên tục thay đổi tên và địa chỉ IP của các máy chủ CnC theo các kỹ thuật xác định trước, như DGA, hoặc FF (*Fast Flux*) [15] [45]. Các thay đổi về tên và IP của các máy chủ CnC cũng liên tục được đẩy lên hệ thống DNS. Các bot cũng được trang bị khả năng sinh tự động tên máy chủ CnC theo các kỹ thuật này. Nhờ vậy, các bot vẫn có thể tìm được địa chỉ IP của máy chủ CnC bằng cách tự động sinh tên miền và truy vấn dịch vụ DNS, như minh họa trên Hình 2.3. Do vậy, việc

giám sát và phân tích dữ liệu truy vấn DNS có thể tiết lộ thông tin liên quan đến sự hiện diện của các máy chủ CnC và botnet trong hệ thống mạng được giám sát, do một phần dữ liệu truy vấn DNS có thể do botnet tạo ra [45].

## 2.2. PHÁT HIỆN CHARACTER-BASED DGA BOTNET SỬ DỤNG CÁC ĐẶC TRƯNG KÝ TỰ

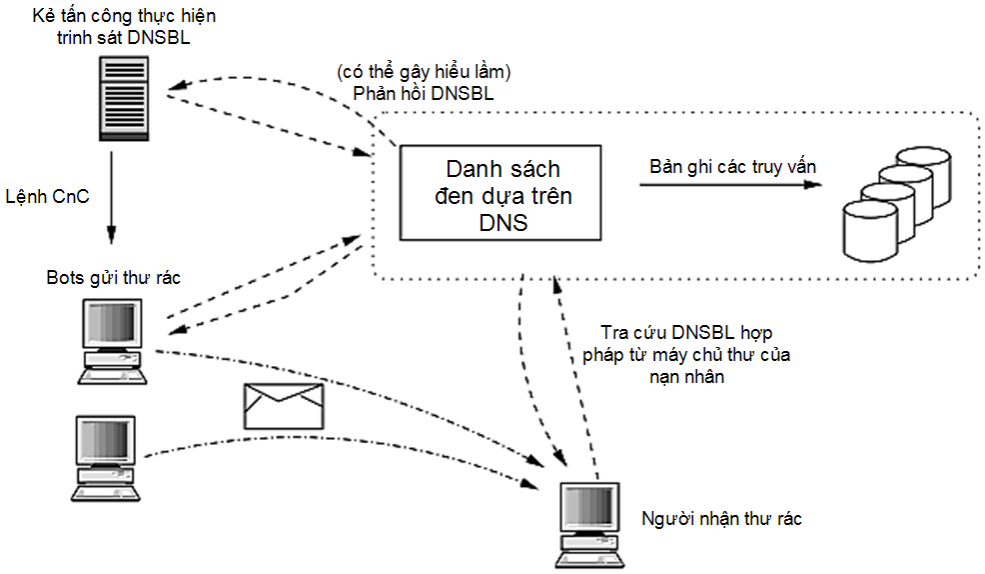
### 2.2.1. Các phương pháp phát hiện DGA botnet

Có thể thấy các họ DGA botnet chiếm tỷ lệ lớn trong số các họ botnet đã và đang hoạt động và do vậy các phương phát hiện DGA botnet đã được đề xuất trong thời gian qua cũng rất phong phú. Có thể chia các phương phát hiện DGA botnet đã được đề xuất trong những năm qua thành 3 nhóm: phát hiện dựa trên phân tích truy vấn DNS, phát hiện dựa trên thống kê và phát hiện dựa trên học máy. Mục này phân tích các đề xuất nổi bật của từng nhóm và các ưu, nhược điểm của chúng.

#### *2.2.1.1. Phát hiện DGA botnet dựa trên phân tích truy vấn DNS*

Kỹ thuật phát hiện botnet dựa trên truy vấn DNS sử dụng phân tích lưu lượng DNS để xác định bất kỳ sự bất thường nào. Kỹ thuật này bao gồm bốn cách tiếp cận: (i) Giám sát các yêu cầu DNS thất bại, (ii) Theo dõi các miền độc hại, (iii) Theo dõi các tên miền có TTL thấp và (iv) Theo dõi lưu lượng truy cập bất thường của DNS. ***Giám sát các yêu cầu DNS thất bại*** là phương pháp phát hiện sự hiện diện của các bot trong một mạng dựa trên quan sát thực tế là các bot thường xuyên sinh tên miền và truy vấn hệ thống DNS để tìm địa chỉ IP của máy chủ CnC. Tuy nhiên, do các tên miền CnC có thời gian sống khá ngắn, nên bot thường phải sinh và truy vấn nhiều tên miền và đa số chúng là các tên miền không tồn tại (*có thể chưa được đăng ký hoặc đã hết hạn*). Việc truy vấn các tên miền không tồn tại sẽ sinh ra lỗi truy vấn DNS thất bại và có thể thấy các bot thường sinh ra nhiều yêu cầu DNS thất bại hơn người dùng thông thường. ***Theo dõi các miền độc hại*** là kỹ thuật liên quan đến việc kiểm tra tất cả các yêu cầu gửi đến máy chủ DNS để đảm bảo rằng không có tên miền nào được xử lý thuộc cơ sở dữ liệu danh sách đen, như DNSBL. ***Kỹ thuật theo dõi các tên miền có TTL thấp*** dựa trên thực tế là các botnet sử dụng một kỹ thuật thông lượng nhanh để cản trở việc phát hiện bằng cách sửa đổi địa chỉ IP được liên kết với một tên miền.

Tuy nhiên, các tên miền như vậy có TTL rất thấp, và điều này có nghĩa là hệ thống DNS cần liên tục làm mới bộ đệm phân giải của IP liên quan đến tên miền. ***Kỹ thuật theo dõi lưu lượng truy cập bất thường của DNS*** cố gắng phát hiện các botnet dựa trên phân tích lưu lượng để tìm bất kỳ sự bất thường nào, như thay đổi đột ngột về lưu lượng truy cập, lưu lượng truy cập đến các cổng bất thường và độ trễ mạng. Tất cả những điều này có thể chỉ ra sự tồn tại của một botnet. Phần tiếp theo của mục này mô tả một số đề xuất phát hiện DGA botnet dựa trên phân tích truy vấn DNS theo các hướng tiếp cận nêu trên.

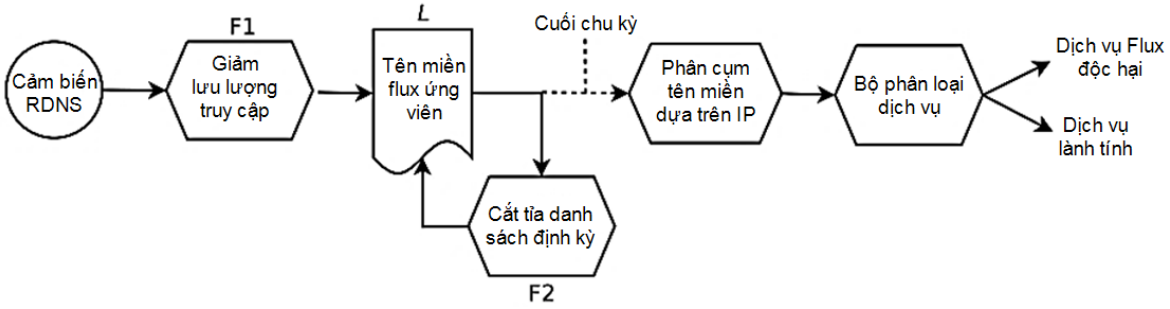



*Hình 2.4: Mô hình Botmatter truy vấn DNSBL*

Ramachandran và cộng sự [75] đề xuất các kỹ thuật bộ đếm thông minh để phát hiện các thành viên của botnet thông qua việc giám sát các truy vấn cơ sở dữ liệu DNSBL (*danh sách đen truy vấn DNS*) từ các botmaster để xác định xem các bot thành viên có trong danh sách đen gửi thư rác hay không. Đề xuất này dựa trên quan sát thực tế là botmaster và cả các bot thường thực hiện các truy vấn trinh sát các cơ sở dữ liệu DNSBL để kiểm tra xem các bot của mạng có tồn tại trong danh sách đen gửi thư rác hay không và trên cơ sở đó có điều chỉnh để vượt qua hệ thống lọc thư rác, như biểu diễn trên Hình 2.4. Thông qua việc giám sát và nhận diện các truy vấn trinh sát đến cơ sở dữ liệu DNSBL, phương pháp đề xuất có khả năng nhận dạng các bot đã biết và phát hiện các bot mới là thành viên tham gia gửi thư rác của mạng

botnet. Ngoài ra, các tác giả cũng đề xuất một nhóm kỹ thuật "đầu độc trinh sát (*reconnaissance poisoning*)", cho phép gửi phản hồi giả, hoặc thông tin bẫy khi phát hiện các truy vấn trinh sát từ botmaster và các bot, nhằm giảm thiểu thư rác.

Một hướng nghiên cứu phát hiện các botnet được quan tâm là phát hiện botnet thông qua việc giám sát các truy vấn từ các bot gửi hệ thống DNS để tìm địa chỉ IP của máy chủ CnC. Bằng việc giám sát các yêu cầu truy vấn hệ thống DNS, có thể phát hiện sự tồn tại của các bot và mạng botnet. Theo hướng này, Villamari-Salomo và cộng sự [98] đề xuất phương pháp nhận dạng các máy chủ CnC của botnet dựa trên phát hiện bất thường thông qua việc giám sát các truy vấn hệ thống DNS động. Hai hướng tiếp cận được thử nghiệm gồm (1) giám sát phát hiện các tên miền có tần suất truy nhập cao bất thường và tập trung tạm thời và (2) giám sát các hồi đáp lặp lại cho các truy vấn các tên miền không tồn tại. Đề xuất này dựa trên thực tế là botmaster thường xuyên thay đổi địa chỉ IP và tên máy chủ CnC, có một số lượng rất lớn các bot đồng thời truy vấn dịch vụ DNS để tìm địa chỉ IP của máy chủ CnC và trong số đó nhiều bot chưa được cập nhật vẫn gửi truy vấn các tên miền đã bị hủy.




*Hình 2.5: Kiến trúc hệ thống phát hiện dịch vụ độc hại*

Mở rộng đề xuất của Villamari-Salomo, Perdisci và cộng sự [68] đề xuất phương pháp giám sát các truy vấn DNS đệ quy (Recursive DNS - RDNS) để phát hiện các dịch vụ độc hại (malicious flux services) có liên quan đến các máy tính bị điều khiển của botnet, như blog rác, tin nhắn rác và thư rác. Hình 2.5 biểu diễn kiến trúc hệ thống phát hiện dịch vụ độc hại đề xuất. Theo đó, các tác giả sử dụng phương pháp phân cụm để xây dựng bộ phân loại các tên miền hợp lệ và các tên miền sử dụng cho các dịch vụ độc hại. Các kết quả thử nghiệm trên 2 mạng ISP thực tế có lưu lượng

lớn trong 45 ngày cho thấy phương pháp đề xuất phát hiện chính xác tên miền sử dụng cho các dịch vụ độc hại.

Jiang và cộng sự [33] đề xuất phương pháp nhận dạng các hành vi đáng ngờ dựa trên phân tích các truy vấn DNS thất bại sử dụng công cụ DNS Failure Graphs. Các truy vấn DNS thất bại là những truy vấn với tên miền không tồn tại, tên miền hết hạn, hoặc do lỗi trong hệ thống DNS. Đề xuất này xuất phát từ thực tế, các hành vi đáng ngờ khởi phát từ hoạt động gửi thư rác, hoạt động của các trojan, đặc biệt là hoạt động của các bot bao gồm việc truy vấn hệ thống DNS để tìm địa chỉ IP của các máy chủ gửi thư, hoặc máy chủ CnC. Một số lượng đáng kể trong số truy vấn dạng này là truy vấn DNS thất bại do tên miền truy vấn không tồn tại, tên miền hết hạn. Kết quả thử nghiệm trên tập dữ liệu truy vấn DNS thu thập trong 3 tháng ở một hệ thống mạng cỡ lớn cho thấy phương pháp đề xuất có thể nhận dạng các hành vi bất thường mới với xác xuất cao khởi phát từ các bot chưa biết.

Các nhà nghiên cứu từ Đại học Texas A&M và Narus.com đã phát triển một phương pháp để phát hiện dòng chảy tên miền trong lưu lượng DNS [102]. Dựa trên quan sát, các tác giả nhận thấy rằng các tên miền được tạo theo thuật toán thể hiện các đặc điểm khác biệt rất lớn so với các tên miền hợp pháp. Một số thước đo khoảng cách, bao gồm KL-Distance, Edit Distance và thước đo Jaccard được sử dụng để xem xét sự phân bố của các ký tự và chữ số [102]. Ngoài ra, theo quan sát là các bot từ cùng một botnet sử dụng cùng một thuật toán DGA sẽ tạo ra lưu lượng truy cập NXDomain tương tự nhau. NXDomain là lưu lượng truy vấn các tên miền không tồn tại, hoặc hết hạn. Yong-lin Zhou và cộng sự [103] đã đề xuất một phương pháp phát hiện DGA-botnet dựa trên lưu lượng DNS NXDomain được thu thập tại các RDNS thử nghiệm. Nhóm nghiên cứu trích xuất thời gian hoạt động của từng tên miền, sau đó nhóm các tên miền theo cấp độ, đồng thời tính toán mức độ tương tự về quyền truy cập tên miền cho mỗi nhóm để có được danh sách tên miền DGA đáng ngờ.

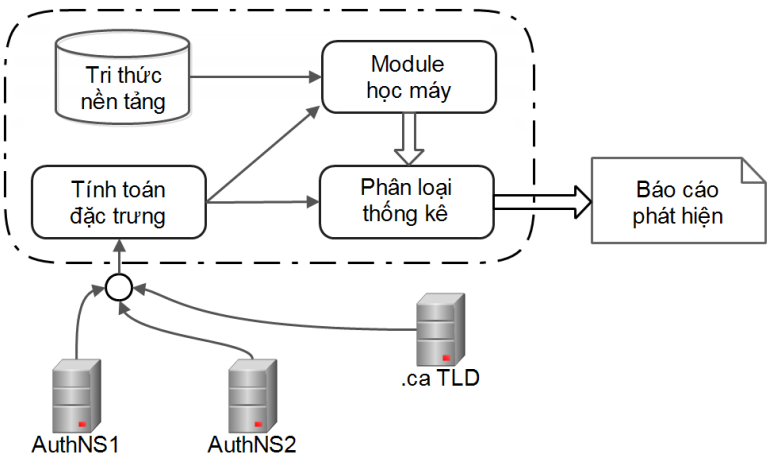
Monika và cộng sự [100] đề xuất một hệ thống phát hiện botnet nguyên mẫu dựa trên phân tích lưu lượng DNS thụ động để phát hiện sự hiện diện của botnet trong

mạng cục bộ. Phương pháp được đề xuất dựa trên phân tích lưu lượng DNS nên có khả năng cho phép phát hiện sớm các bot trên mạng. Ngoài ra, phương pháp này không phụ thuộc vào số lượng bot hoạt động trong mạng cục bộ và hiệu quả khi chỉ có một số lượng nhỏ máy bị nhiễm mã độc. Các đặc trưng được trích xuất bao gồm: Số lượng ASN (*Autonomous System Number*) tối đa được trả về cho một miền trong một phản hồi DNS duy nhất; Số lượng ASN tối đa được trả về cho một tên miền trên tất cả các phản hồi DNS; Tần suất phản hồi NXDomain; và Tần suất sử dụng DNS mở rộng.

#### *2.2.1.2. Phát hiện DGA botnet dựa trên thống kê*

Bên cạnh kỹ thuật phân tích truy vấn DNS, thống kê cũng là phương pháp được sử dụng rộng rãi trong phát hiện các dạng DGA botnet. Tiêu biểu, Yadav và cộng sự [102] đề xuất phương pháp phân biệt các tên miền sinh tự động bằng thuật toán thường được sử dụng trong các botnet với tên miền hợp lệ dựa trên phân tích sự phân bố các nhóm ký tự (*1-gram, hoặc 2-gram liền kề*) trong tên miền. Các tác giả sử dụng độ đo phân kỳ Kullback-Leibler (*K-L*) để tính toán khoảng cách giữa các tên miền hợp lệ và các tên miền độc hại được sinh tự động. Bài báo cũng đề xuất một số kỹ thuật nhóm các tên miền hỗ trợ cho việc tính toán độ đo K-L, bao gồm kỹ thuật phân tích theo tên miền, phân tích theo địa chỉ IP và phân tích thành phần có kết nối trong tên miền. Các kết quả thử nghiệm trên tập dữ liệu thu thập từ một ISP khẳng định phương pháp đề xuất có thể phát hiện Conflicker botnet với tỷ lệ dương tính giả thấp.

Cũng với mục đích tương tự, Stalmans và cộng sự [87] sử dụng phương pháp cây quyết định C5.0 và thống kê Baysian để phân loại các tên miền sinh tự động với các tên hợp lệ. Hai nhóm đặc trưng được sử dụng để huấn luyện các bộ phân loại, bao gồm các đặc trưng DNS (*địa chỉ IP, địa chỉ mạng, TLD, TTL,...*) và các đặc trưng văn bản (*sự phân bố các ký tự trong tên miền*) của tên miền. Các kết quả thử nghiệm cho thấy bộ phân loại Baysian cho kết quả phân loại tốt nhất và có khả năng phát hiện chính xác cả hai loại tên miền độc hại, bao gồm tên miền fast-flux và tên miền sinh theo thuật toán với tỷ lệ cảnh báo giả thấp.




*Hình 2.6: Hệ thống phát hiện Kopis*

Nhằm mục đích phát hiện các tên miền có liên quan đến mã độc botnet, Antonakakis và cộng sự đề xuất hệ thống phát hiện có tên là Kopis [55]. Kopis thực hiện giám sát các truy vấn DNS ở mức cao trong hệ thống phân cấp DNS và phân tích các mẫu phân giải các câu truy vấn DNS toàn cục để phát hiện các tên miền có liên quan đến mã độc. Hình 2.6 mô tả hệ thống phát hiện Kopis. Theo đó, các truy vấn DNS ở mức toàn cục được giám sát, thu thập và đưa về xử lý. Tiếp theo, các đặc trưng thống kê của tên miền được trích chọn và đưa vào huấn luyện bộ phân loại. Bộ phân loại thống kê sau đó được sử dụng để phân loại một tên miền là hợp lệ hay có liên quan đến mã độc botnet. Các kết quả thử nghiệm Kopis cho thấy hệ thống đạt độ chính xác đến hơn 98% và tỷ lệ phát hiện sai dưới 0.5%. Kopis cũng có khả năng phát hiện sớm các các tên miền có liên quan đến mã độc nhiều ngày trước khi chúng được đưa vào các danh sách đen.

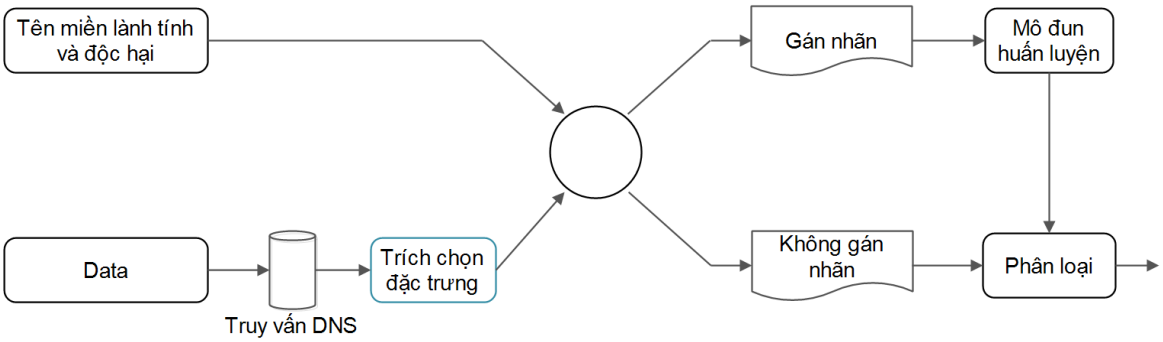
Theo một cách tiếp cận khác, Zhao và cộng sự [26] đề xuất một phương pháp dựa trên thống kê cho phát hiện các tên miền độc hại sử dụng kỹ thuật n-gram. Theo đó, mỗi tên miền trong tập huấn luyện gồm các tên miền hợp pháp trước hết được tách thành các chuỗi ngắn sử dụng kỹ thuật 3, 4, 5, 6 và 7-gram. Các giá trị thống kê và trọng số của mỗi chuỗi ngắn của tất cả các tên miền trong tập huấn luyện được tính toán để tạo thành một ‘hồ sơ phát hiện’. Với mỗi tên miền giám sát, các giá trị thống kê của tên miền được tính toán và chúng được sử dụng để tính toán ‘giá trị danh tiếng’ của tên miền dựa trên ‘hồ sơ phát hiện’. Một ngưỡng ‘danh tiếng’ tên

miền được tạo ra cho mỗi nhóm tên miền độc hại sử dụng ‘hồ sơ phát hiện’. Nếu ‘giá trị danh tiếng’ của tên miền lớn hơn ngưỡng ‘danh tiếng’, tên miền là hợp pháp và ngược lại tên miền là độc hại. Các thử nghiệm cho thấy, phương pháp đề xuất đạt độ chính xác phát hiện 94.04%. Tuy vậy, hiệu suất phát hiện của phương pháp đề xuất phụ thuộc nhiều vào việc lựa chọn ngưỡng ‘danh tiếng’ và giá trị này được sinh và lựa chọn thủ công. Hơn nữa, tỷ lệ dương tính giả (FPR) và âm tính giả (FNR) của phương pháp đề xuất còn tương đối cao, lần lượt là 6.14% và 7.42%.

#### *2.2.1.3. Phát hiện DGA botnet dựa trên học máy*

Bên cạnh thống kê, các kỹ thuật học máy đã và đang được ứng dụng rộng rãi trong phát hiện tấn công, xâm nhập nói chung và phát hiện botnet nói riêng. Ưu điểm của phát hiện DGA botnet dựa trên học máy là độ chính xác ngày càng được cải thiện cao và khả năng tự động xây dựng mô hình phát hiện từ tập dữ liệu huấn luyện. Mục này khảo sát một số đề xuất tiêu biểu cho phát hiện DGA botnet dựa trên các kỹ thuật học máy truyền thống, các kỹ thuật học sâu, cũng như phương pháp kết hợp giữa học máy truyền thống và học sâu.

Bilge và cộng sự giới thiệu hệ thống EXPOSURE [44], như minh họa trên Hình 2.7 cho phép giám sát lưu lượng truy vấn DNS trên diện rộng để phát hiện các tên miền có liên quan đến các hành vi độc hại dựa trên học máy có giám sát truyền thống. EXPOSURE sử dụng 4 nhóm gồm 15 đặc trưng của tên miền để phân biệt các tên miền đáng ngờ với tên miền hợp lệ. Bốn nhóm đặc trưng của tên miền được sử dụng trong huấn luyện bộ phân loại bao gồm: các đặc trưng thời gian (*tần suất truy nhập, mẫu lặp,...*), các đặc trưng hồi đáp truy vấn DNS (*IP phân biệt, TLD, số tên miền chia sẻ địa chỉ IP,...*), các đặc trưng TTL (*TTL trung bình, độ lệch chuẩn TTL,...*) và các đặc trưng văn bản của tên miền (*tỷ lệ ký tự số, tỷ lệ độ dài các chuỗi dài nhất có nghĩa trong tên miền*). Các kết quả thử nghiệm trên tập dữ liệu 100 triệu truy vấn DNS cho thấy hệ thống có khả năng mở rộng tốt và nhận dạng được các tên miền mới có liên quan đến các hành vi độc hại, như được sử dụng cho máy chủ CnC, gửi thư rác và sử dụng cho website lừa đảo.

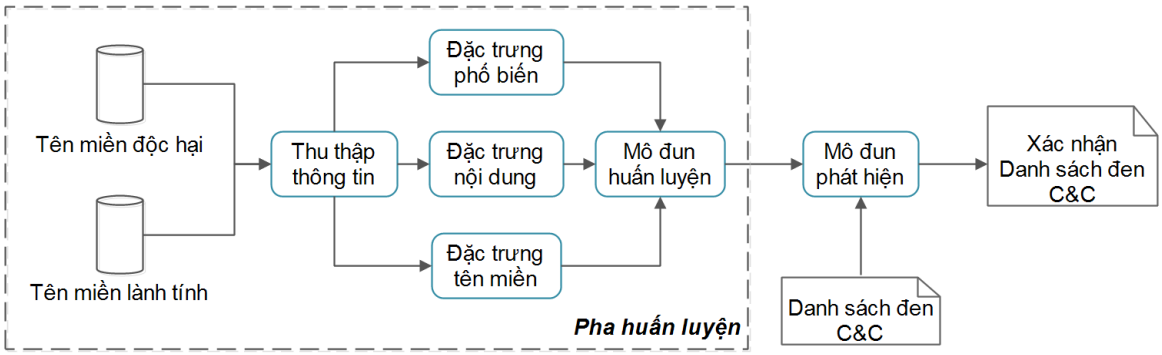



*Hình 2.7: Mô hình kiến trúc của EXPOSURE*

Theo một cách tiếp cần khác, Kheir và cộng sự đề xuất hệ thống Mentor [40] cho phép loại bỏ các tên miền hợp lệ ra khỏi danh sách đen các tên miền của các máy chủ CnC dựa trên hệ thống dựa trên danh tiếng DNS tích cực (*Positive DNS reputation system*). Mentor sử dụng một hệ thống thu thập thông tin và các đặc tính thống kê về nội dung trang web và tên miền để phân loại tên miền hợp lệ và tên miền đáng ngờ dựa trên học máy có giám sát. Hình 2.8 mô tả kiến trúc và lưu đồ xử lý của Mentor, trong đó bộ thu thập thông tin được sử dụng để chủ động truy vấn và thu thập các thông tin về các tên miền sử dụng cho quá trình huấn luyện. 16 đặc trưng được trích xuất phân thành 3 nhóm bao gồm: nhóm các đặc trưng đăng ký, quản lý tên miền, nhóm các đặc trưng nội dung của website gắn với tên miền và nhóm các đặc trưng độ phổ biến của tên miền. Các thuật toán học máy có giám sát được sử dụng để xây dựng bộ phân loại bao gồm phân loại Baysian, SVM, cây quyết định J48 và C4.5. Các kết quả thực nghiệm trên một số danh sách đen công cộng chứa các tên miền botnet cho thấy hệ thống có khả năng nhận dạng chính xác các tên miền hợp lệ với tỷ lệ sai rất thấp.

Theo hướng kết hợp giữa các kỹ thuật học máy truyền thống và học sâu, Hieu Mac và cộng sự [54] đã thử nghiệm nhiều kỹ thuật học máy cho phát hiện DGA botnet. Các phương pháp được thử nghiệm bao gồm: HMM, ELM, C4.5, SVM, LSTM, R-SVM, CNN+LSTM và Bi-LSTM. Thử nghiệm trên tập dữ liệu DGA được thu thập trong thế giới thực liên quan đến 38 họ botnet với 168.900 tên miền cho thấy, các phương pháp dựa trên học sâu và kết hợp (*LSTM, R-SVM, CNN+LSTM và Bi-*

*LSTM*) cho kết quả tốt hơn các phương pháp dựa trên học truyền thống (*HMM, ELM, C4.5, SVM*).




*Hình 2.8: Kiến trúc và lưu đồ xử lý của Mentor*

Hoàng và cộng sự [24] đề xuất một phương pháp phát hiện botnet dựa trên học máy sử dụng phân tích truy vấn DNS. Bài báo sử dụng các kỹ thuật học máy có giám sát, bao gồm Naive Bayes, kNN, cây quyết định và rừng ngẫu nhiên để xây dựng các mô hình phát hiện cho phân loại các tên miền sinh và sử dụng bởi botnet và các tên miền bình thường. Mô hình đề xuất trích xuất 18 đặc trưng phân loại cho mỗi tên miền, bao gồm 16 đặc trưng thống kê n-gram, 1 đặc trưng phân bố các nguyên âm trong tên miền và 1 đặc trưng entropy của các ký tự trong tên miền. Các kết quả thử nghiệm cho thấy hầu hết các kỹ thuật học máy đều cho độ chính xác phát hiện khả quan, trong đó thuật toán rừng ngẫu nhiên cho độ chính xác cao nhất (*đạt trên 90%*) và tỷ lệ cảnh báo sai thấp nhất. Các hạn chế của mô hình đề xuất là (1) chỉ có khả năng phát hiện các character-based DGA, (2) tỷ lệ cảnh báo sai còn khá cao (đến 9.30%) và (3) tập dữ liệu thử nghiệm tương đối nhỏ, có thể ảnh hưởng đến độ tin cậy của kết quả.

Cũng nhằm phát hiện các character-based DGA, Truong và cộng sự [96] đề xuất một phương pháp phát hiện các botnet tự động sinh các tên miền dựa trên các đặc trưng lưu lượng DNS. Nghiên cứu sử dụng các đặc trưng tên miền, bao gồm độ dài và giá trị mong đợi của tên miền để phân biệt các tên miền sinh tự động (PDN) và tên miền bình thường. Giá trị mong đợi của tên miền được tính toán dựa trên phân bố ký tự của 100,000 tên miền thông dụng nhất trên bảng xếp hạng của Alexa [17]. Năm

thuật toán học máy, bao gồm Naive Bayes, kNN, SVM, cây quyết định và rừng ngẫu nhiên được sử dụng để xây dựng các bộ phân loại. Kết quả thử nghiệm trên tập dữ liệu 100,000 tên miền bình thường và 20,000 tên miền PDN cho thấy, phương pháp đề xuất đạt độ chính xác phát hiện chung cao nhất là 92.30% với thuật toán cây quyết định. Mặc dù độ chính xác phát hiện của phương pháp đề xuất khá cao, nhưng tỷ lệ cảnh báo sai tổng cũng tương đối cao, khoảng 7.70% trong trường hợp tốt nhất. Ngoài ra, phương pháp đề xuất cũng không thể phát hiện các họ word-based, hoặc mixed DGA botnet.

Theo hướng sử dụng học sâu, Qiao và cộng sự [70] đề xuất phương pháp phân loại các tên miền DGA sử dụng kỹ thuật học sâu LSTM với cơ chế chú ý. Trong phương pháp đề xuất, mỗi tên miền được đưa qua quá trình tiền xử lý gồm các bước: tách, đệm và nhúng các chuỗi ký tự. Tên miền sau đó được chuyển đổi thành một ma trận 54x128 cho huấn luyện và kiểm thử. Các thử nghiệm trên tập dữ liệu gồm 1 triệu tên miền thông dụng nhất theo xếp hạng của Alexa [17] và 1,675,404 tên miền DGA cho thấy phương pháp đề xuất đạt độ đo F1 trung bình là 94.58%. Ưu điểm của phương pháp là đạt độ chính xác cao và loại bỏ được quá trình trích chọn các đặc trưng. Tuy nhiên, phương pháp đề xuất cũng chỉ phát hiện tốt các tên miền character-based DGA. Ngoài ra, tỷ lệ cảnh báo sai tổng cũng còn tương đối cao, khoảng 5% tính theo độ đo F1.

#### *2.2.1.4. Ưu điểm và hạn chế của các đề xuất phát hiện DGA botnet*

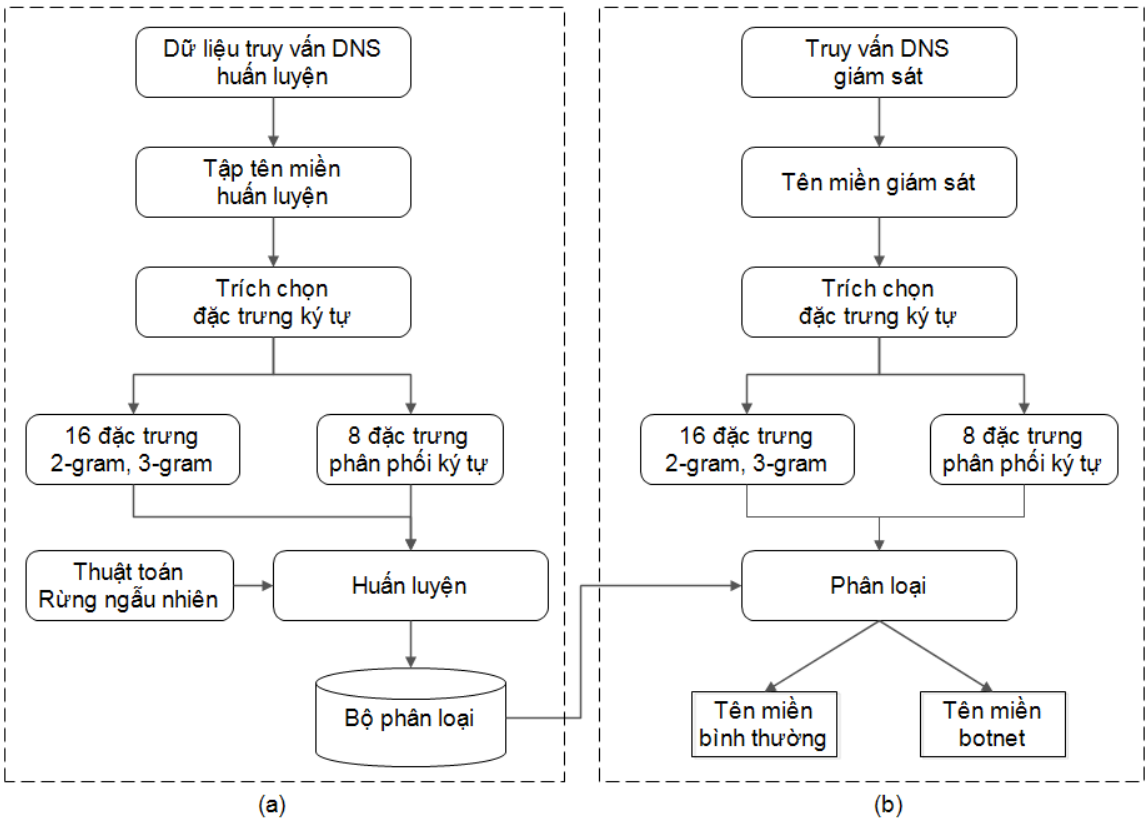
Kết quả khảo sát cho thấy, các giải pháp đề xuất phát hiện botnet dựa trên phân loại các tên miền hợp lệ và các tên miền được sinh tự động cho các máy chủ CnC của botnet là một trong các hướng đi hiệu quả trong phát hiện botnet do độ chính xác phát hiện cao, tỷ lệ cảnh báo sai thấp và yêu cầu chi phí tính toán không quá lớn. Đặc biệt, các đề xuất phát hiện DGA botnet dựa trên học máy là nhánh nghiên cứu rất có triển vọng do mô hình phát hiện có thể được xây dựng tự động từ dữ liệu huấn luyện. Đây cũng chính là nhánh nghiên cứu mà luận án lựa chọn thực hiện.

Mặc dù các nghiên cứu đã có cho phát hiện DGA botnet đã đạt được nhiều kết quả hứa hẹn, vẫn còn một số vấn đề cần tiếp tục nghiên cứu như: (1) do các tập đặc trưng lựa chọn cho phân loại tên miền và/hoặc giải thuật học máy sử dụng chưa thực sự phù hợp nên độ chính xác phát hiện chung đạt khoảng 90% và do vậy tỷ lệ phát hiện sai khoảng 10% vẫn là tương đối cao, điều này làm giảm khả năng triển khai ứng dụng trên thực tế; (2) một số đề xuất chỉ cho độ chính xác phát hiện cao với một tập dữ liệu cụ thể, hoặc một số họ DGA botnet, nhưng có khả năng phát hiện kém, hoặc không thể phát hiện một số họ DGA botnet khác; và (3) một số đề xuất đòi hỏi chi phí tính toán rất lớn cho quá trình xây dựng mô hình, cũng như quá trình giám sát phát hiện và điều này làm giảm khả năng triển khai ứng dụng trên các hệ thống mạng có lưu lượng lớn. Mô hình đề xuất phát hiện CDM trong mục này nhằm giải quyết vấn đề (1) nêu trên.

### 2.2.2. Giới thiệu mô hình phát hiện CDM

Hình 2.9 biểu diễn mô hình phát hiện character-based DGA botnet đề xuất (*CDM – Character-based DGA Botnet Detection Model*) dựa trên phân loại tên miền trích xuất từ dữ liệu truy vấn DNS. Mô hình CDM sử dụng tập đặc trưng ký tự có khả năng phân biệt hiệu quả giữa các tên miền lành tính và các tên miền character-based DGA - là các tên miền được sinh tự động sử dụng tổ hợp ngẫu nhiên các ký tự. Các đặc trưng ký tự gồm các đặc trưng được được trích xuất dựa trên quan sát, phân tích sự khác biệt giữa các tên miền lành tính và các tên miền character-based DGA ở mức ký tự mà không xem xét đến ngữ nghĩa của các cụm ký tự trong tên miền. Mô hình CDM được xây dựng trên cơ sở phân tích hoạt động của các botnet: botmaster sử dụng thuật toán DGA để sinh và đăng ký tự động các tên miền cho các máy chủ CnC nhằm lẩn trách bị phát hiện và đưa vào danh sách đen, đồng thời các bot định kỳ sử dụng thuật toán DGA để sinh tự động tên miền và sau đó truy vấn hệ thống DNS để tìm địa chỉ IP của máy chủ CnC theo tên miền sinh tự động. Từ dữ liệu truy vấn DNS các tên miền truy vấn được bóc tách và phân loại nhằm phát hiện các hoạt động của botnet trong hệ thống.

Mô hình phát hiện CDM được thực hiện thành 2 giai đoạn: (a) giai đoạn huấn luyện và (b) giai đoạn phát hiện. Trong giai đoạn huấn luyện, dữ liệu truy vấn hệ thống DNS được thu thập, sau đó trích xuất các tên miền được truy vấn. Tiếp theo, tập tên miền được tiền xử lý nhằm trích xuất các đặc trưng cho bước huấn luyện. Có 24 đặc trưng được trích xuất, bao gồm 16 đặc trưng n-gram và 8 đặc trưng thống kê, phân phối các dạng ký tự. Trong đó, 18 đặc trưng được kế thừa từ [24] và 6 đặc trưng bổ sung mới. Trong bước huấn luyện, thuật toán học máy Rừng ngẫu nhiên được áp dụng để học ra Bộ phân loại, hay Mô hình phát hiện. Thuật toán học máy Rừng ngẫu nhiên được lựa chọn do đây là thuật toán học máy có tốc độ xử lý nhanh và độ chính xác phát hiện cao trong nhiều đánh giá [24] [26] [70] [96]. Trong giai đoạn phát hiện của mô hình, các truy vấn DNS được giám sát và qua quá trình trích xuất tên miền, tiền xử lý tương tự như quá trình huấn luyện và đến bước phân loại sử dụng Bộ phân loại từ giai đoạn huấn luyện để xác định một tên miền lành tính hay tên miền DGA botnet.




*Hình 2.9: Mô hình phát hiện Character-based DGA botnet*

### 2.2.3. Tập dữ liệu huấn luyện và kiểm thử

Để đánh giá hiệu năng của mô hình CDM, luận án sử dụng các tập dữ liệu tên miền đã được bóc tách và gán nhãn, bao gồm tập các tên miền lành tính và tập các tên miền độc hại do DGA botnet sinh và sử dụng. Danh sách các tên miền lành tính gồm 100,000 tên miền có thứ hạng cao nhất trong xếp hạng của Alexa [17]. Danh sách các tên miền độc hại được thu thập tại [11], bao gồm 171,393 tên miền được sinh và sử dụng bởi 39 họ DGA botnet. Từ tập dữ liệu ban đầu, các tên miền sẽ được xử lý để bỏ đi phần tên miền mức cao nhất (TLD) chỉ lấy phần tên miền thứ cấp (SLD). Ví dụ, với tên miền "example.com", sau khi qua xử lý, dữ liệu thu được sẽ là "example". Bảng 2.3 là danh sách 39 họ DGA botnet được lựa chọn để huấn luyện và kiểm thử, trong đó: (1) tập huấn luyện: bao gồm 100,000 tên miền lành tính và 100.000 tên miền của 13 họ DGA botnet; (2) tệp kiểm thử: bao gồm 71.393 tên miền của 39 họ DGA botnet. Ngoài ra, để đánh giá khả năng phát hiện botnet mới của mô hình CDM, luận án sử dụng tập dữ liệu UMUDGA gồm 7 họ DGA botnet liệt kê tại Bảng 2.4, không xuất hiện trong tập huấn luyện.

*Bảng 2.3: Tệp huấn luyện và kiểm thử cho mô hình CDM [11]*

| STT | Họ DGA botnet | Kiểu DGA | Tập huấn luyện | Tập kiểm thử |
|---|---|---|---|---|
| 1 | emotet | character-based | 10,000 | 4,000 |
| 2 | gameover | character-based | 8,000 | 4,000 |
| 3 | murofet | character-based | 4,000 | 4,000 |
| 4 | necurs | character-based | 4,000 | 4,000 |
| 5 | pykspa_v1 | character-based | 15,000 | 4,000 |
| 6 | ramnit | character-based | 10,000 | 4,000 |
| 7 | ranbyus | character-based | 4,000 | 4,000 |
| 8 | rovnix | character-based | 10,000 | 4,000 |
| 9 | shiotob | character-based | 4,000 | 4,000 |
| 10 | symmi | character-based | 3,000 | 1,200 |
| 11 | tinba | character-based | 10,000 | 4,000 |
| 12 | simda | character-based | 14,000 | 4,000 |
| 13 | virut | character-based | 4,000 | 4,000 |
| 14 | proslikefan | character-based | - | 100 |
| 15 | tempedreve | character-based | - | 195 |
| 16 | tinynuke | character-based | - | 32 |

| | | | | |
|---|---|---|---|---|
| 17 | vidro | character-based | - | 100 |
| 18 | pykspa_v2_real | character-based | - | 199 |
| 19 | pykspa_v2_fake | character-based | - | 799 |
| 20 | padcrypt | character-based | - | 168 |
| 21 | nymaim | character-based | - | 480 |
| 22 | vawtrak | character-based | - | 827 |
| 23 | shifu | character-based | - | 2,546 |
| 24 | fobber_v1 | character-based | - | 298 |
| 25 | fobber_v2 | character-based | - | 299 |
| 26 | dircrypt | character-based | - | 762 |
| 27 | cryptolocker | character-based | - | 1,000 |
| 28 | locky | chracter-based | - | 1,158 |
| 29 | chinad | chracter-based | - | 1,000 |
| 30 | qadars | chracter-based | - | 2,000 |
| 31 | dyre | chracter-based | - | 1,000 |
| 32 | mydoom | chracter-based | - | 50 |
| 33 | gspy | chracter-based | - | 100 |
| 34 | enviserv | chracter-based | - | 500 |
| 35 | conflicker | chracter-based | - | 495 |
| 36 | banjori | mixed-based | - | 4,000 |
| 37 | matsnu | word-based | - | 881 |
| 38 | bigviktor | word-based | - | 999 |
| 39 | suppobox | word-based | - | 2,205 |
| 40 | Tên miền lành tính | | 100,000 | - |
| **Tổng** | | | **200,000** | **71,393** |

*Bảng 2.4: Tập kiểm thử UMUDGA*

| STT | Họ DGA botnet | Kiểu DGA | Tập huấn luyện | Tập kiểm thử |
|---|---|---|---|---|
| 1 | alureon | chracter-based | - | 5,000 |
| 2 | bedep | chracter-based | - | 5,000 |
| 3 | corebot | chracter-based | - | 5,000 |
| 4 | kraken | chracter-based | - | 2,000 |
| 5 | pushdo | chracter-based | - | 5,000 |
| 6 | zeus | chracter-based | - | 5,000 |
| 7 | pizd | word-based | - | 4,000 |
| **Tổng** | | | **-** | **31,000** |

### 2.2.4. Tiền xử lý dữ liệu

#### *2.2.4.1. Giới thiệu*

Theo [51] [87] [102], các tên miền do botnet sinh tự động thường có các đặc trưng DNS, đặc trưng mạng và đặc trưng ngữ nghĩa khác biệt so với các tên miền thông thường. Nghiên cứu [102] đề xuất sử dụng kỹ thuật phân tích phân bố các nguyên âm, chữ số và các ký tự khác để phân biệt các tên miền hợp lệ và các tên miền sinh bằng thuật toán của botnet. Mở rộng hơn, [87] đề xuất sử dụng 2 nhóm đặc trưng của tên miền, gồm các đặc trưng DNS (địa chỉ IP, địa chỉ mạng, quốc gia, TTL,...) và các đặc trưng từ vựng (*phân bố các ký tự của tên miền*). Trong khi đó, [51] đề xuất sử dụng 36 đặc trưng trong 2 nhóm, gồm 18 đặc trưng từ vựng (*trung bình, phương sai và độ lệch chuẩn của 1-gram, 2-gram, 3-gram và 4-gram, entropy, các đặc trưng ký tự, số, nguyên âm, phụ âm*) và 18 đặc trưng mạng (*TTL, số lượng địa chỉ mạng,...*). Kế thừa từ công bố trước đây của nhóm nghiên cứu [24], luận án tập trung khai thác các đặc trưng thống kê từ vựng dựa trên các cụm 2-gram và 3-gram, và các đặc trưng phân bố các dạng ký tự trong tên miền. Cụ thể, mô hình CDM đề xuất sử dụng 24 đặc trưng mức ký tự cho mỗi tên miền, bao gồm:

- Các đặc trưng n-gram gồm 16 đặc trưng thống kê cho các cụm 2-gram (bi-gram) và 3-gram (tri-gram);
- Các đặc trưng loại ký tự gồm 6 đặc trưng phân bố nguyên âm, ký tự, chữ số;
- Các đặc trưng thống kê gồm 2 đặc trưng entropy theo ký tự và giá trị kỳ vọng của tên miền.

#### *2.2.4.2. Các đặc trưng n-gram*

Bi-gram là một cụm gồm 2 ký tự kề nhau được trích ra từ một chuỗi ký tự. Ví dụ, tên miền "example" (*đã loại bỏ TLD*) gồm các bi-gram: ex, xa, am, mp, pl, le. Một tên miền có thể chứa các ký tự trong tập 26 ký tự chữ cái (a-z), các ký tự số (0-9), ký tự "." và "-", do đó tổng số bi-gram có thể có là TS(bi-gram) = $38^2$ =1,444. Từ tập hợp các tên miền lành tính trích rút ra danh sách gồm N cụm bi-gram thường xuyên xuất hiện nhất, ký hiệu là DS(bi-gram). DS(bi-gram) được sử dụng cho việc tính toán 8 đặc trưng liên quan đến bi-gram cho từng tên miền.

Tri-gram là một cụm gồm 3 ký tự kề nhau được trích ra từ một chuỗi ký tự. Ví dụ, tên miền “example ” gồm các tri-gram: exa, xam, amp, mpl, ple. Tương tự cách tính tổng số bi-gram, tổng số tri-gram có thể có TS(tri-gram) = $38^3$ = 54,872. Từ tập hợp các tên miền lành tính trích rút ra danh sách gồm M cụm tri-gram có tần suất xuất hiện cao nhất, ký hiệu DS(tri-gram). DS(tri-gram) được sử dụng cho việc tính toán 8 đặc trưng liên quan đến tri-gram cho từng tên miền.

*Bảng 2.5: 100 bi-gram có tần suất cao nhất của tên miền lành tính và DGA*

| STT | LEGIT | DGA | STT | LEGIT | DGA | STT | LEGIT | DGA | STT | LEGIT | DGA |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | in | us | 26 | la | gm | 51 | ha | ko | 76 | ce | sa |
| 2 | er | so | 27 | as | qe | 52 | ec | mq | 77 | ot | ks |
| 3 | an | ci | 28 | ic | um | 53 | il | uk | 78 | ai | ak |
| 4 | ar | ae | 29 | et | ma | 54 | ol | im | 79 | un | ms |
| 5 | re | qk | 30 | co | oq | 55 | ni | sy | 80 | mi | yo |
| 6 | on | iu | 31 | it | qo | 56 | sh | se | 81 | ns | is |
| 7 | es | iq | 32 | ng | go | 57 | po | aq | 82 | oo | qm |
| 8 | or | ku | 33 | ch | ow | 58 | ve | ke | 83 | hi | ii |
| 9 | st | ya | 34 | el | yk | 59 | ac | wy | 84 | ge | gi |
| 10 | te | em | 35 | nt | yi | 60 | os | am | 85 | ga | mg |
| 11 | al | ky | 36 | se | ok | 61 | io | eg | 86 | ie | ac |
| 12 | en | aa | 37 | is | mu | 62 | rt | qu | 87 | em | wm |
| 13 | ra | sk | 38 | ca | om | 63 | ou | ys | 88 | pe | ew |
| 14 | ne | uo | 39 | am | cu | 64 | ur | mi | 89 | no | qc |
| 15 | ma | ce | 40 | ea | ou | 65 | th | si | 90 | so | qy |
| 16 | li | my | 41 | di | mo | 66 | ed | oo | 91 | rs | aw |
| 17 | at | oc | 42 | nd | wo | 67 | sa | ca | 92 | ir | ia |
| 18 | le | mc | 43 | na | ue | 68 | mo | uy | 93 | be | mm |
| 19 | ta | og | 44 | ia | ga | 69 | om | as | 94 | op | uw |
| 20 | ti | yw | 45 | he | eq | 70 | pa | gu | 95 | pr | ym |
| 21 | to | qg | 46 | ho | ec | 71 | vi | os | 96 | do | ui |
| 22 | de | ws | 47 | lo | we | 72 | ba | oe | 97 | ee | km |
| 23 | ri | gq | 48 | tr | ug | 73 | us | ka | 98 | ss | qs |
| 24 | ro | ei | 49 | ad | ig | 74 | ll | gy | 99 | ap | oy |
| 25 | me | ww | 50 | si | uq | 75 | da | cq | 100 | tu | ek |

*Bảng 2.6: 100 tri-gram có tần suất cao nhất của tên miền lành tính và DGA*

| STT | LEGIT | DGA | STT | LEGIT | DGA | STT | LEGIT | DGA | STT | LEGIT | DGA |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | ing | aaq | 26 | mar | myc | 51 | new | kuy | 76 | ite | kym |
| 2 | ine | aeo | 27 | ste | skm | 52 | car | iuy | 77 | ect | aaa |
| 3 | ion | usq | 28 | ame | cis | 53 | ind | ogu | 78 | ian | kyo |
| 4 | lin | iue | 29 | all | qkw | 54 | par | kuc | 79 | ost | iqo |
| 5 | ter | kui | 30 | net | gmu | 55 | com | ema | 80 | der | uso |
| 6 | ent | aak | 31 | one | oca | 56 | int | sow | 81 | ara | sko |
| 7 | and | usk | 32 | per | emm | 57 | wor | aeu | 82 | tic | eii |
| 8 | the | ius | 33 | sho | emo | 58 | edi | ogo | 83 | ade | gqi |
| 9 | por | ium | 34 | nli | iug | 59 | eri | qkg | 84 | min | usi |
| 10 | ers | usa | 35 | are | emg | 60 | ica | yau | 85 | ari | uok |
| 11 | tra | mck | 36 | lan | aey | 61 | chi | myw | 86 | ear | qko |
| 12 | est | emi | 37 | rea | qgq | 62 | tin | mcm | 87 | ren | ciu |
| 13 | tor | yak | 38 | onl | qka | 63 | str | wsk | 88 | ard | ciq |
| 14 | sta | aec | 39 | ist | gqs | 64 | ant | cii | 89 | sti | wss |
| 15 | res | wsa | 40 | sto | skg | 65 | che | uoi | 90 | ama | sks |
| 16 | ati | wwy | 41 | con | ecu | 66 | cha | ska | 91 | les | owa |
| 17 | art | mca | 42 | ide | iqy | 67 | log | ocu | 92 | ews | ciw |
| 18 | tio | oga | 43 | nte | mys | 68 | our | kus | 93 | ech | usu |
| 19 | ort | qks | 44 | ive | ywo | 69 | ast | skc | 94 | ina | soe |
| 20 | pro | myy | 45 | ess | use | 70 | ana | mcg | 95 | nes | yam |
| 21 | for | soq | 46 | ore | gqa | 71 | age | iqa | 96 | lle | yai |
| 22 | ran | qgy | 47 | man | ceq | 72 | orn | iui | 97 | mes | qgk |
| 23 | ate | ogg | 48 | dia | cic | 73 | ita | yae | 98 | ani | eim |
| 24 | ver | yac | 49 | ang | qkq | 74 | ons | mcu | 99 | tec | aee |
| 25 | tal | emy | 50 | ree | som | 75 | pla | sou | 100 | ser | gmi |

*Bảng 2.5 liệt kê 100 bi-gram có tần suất xuất hiện nhiều nhất trong trong 1,000 bi-gram được lấy ra từ 100,000 tên miền lành tính và 100,000 tên miền DGA botnet. Thống kê cho thấy chỉ có 22 bi-gram (22%) của tên miền DGA có trong 100 bi-gram của tên miền lành tính. Mặt khác, trong 1,000 bi-gram có tần suất xuất hiện nhiều nhất thì có tới 147 bi-gram không tồn tại trong 1,000 bi-gram tên miền lành tính.*

Bảng 2.6 mô tả 100 tri-gram có tần suất cao nhất của tên miền lành tính và tên miền DGA. Trong 200 tri-gram có tần suất cao nhất của tên miền DGA thì chỉ có 20 tri-gram (10%) trùng với tri-gram của tên miền lành tính, trong đó chỉ có 2 tri-gram

nằm trong top 200. Mặt khác, 1,000 tri-gram có tần suất cao nhất của tên miền DGA thì chỉ có 82 tri-gram (8.2%) trùng với tri-gram của tên miền lành tính.

Các số liệu thống kê và phân tích nêu trên cho thấy rằng, có thể sử dụng các đặc trưng n-gram để phân loại được tên miền hợp pháp và tên miền DGA. 16 đặc trưng bi-gram và tri-gram (*n-gram*) cho từng tên miền *d* được đánh số từ *f1* đến *f16* bao gồm:

- *f1* và *f9*: *count(d)* là số lượng n-gram của tên miền *d* có trong *DS(n-gram)*.
- *f2* và *f10*: *m(d)* là phân bố tần suất chung của các n-gram trong tên miền *d*, được tính theo công thức:

$$m(d) = \sum_{i=1}^{count(d)} f(i) * index(i) \qquad (2.1)$$

trong đó *f(i)* là tổng số lần suất hiện của n-gram *i* trong *DS(n-gram)* và *index(i)* là thứ hạng của n-gram *i* trong *TS(n-gram)*

- *f3* và *f11*: *s(d)* là trọng số n-gram, được tính theo công thức:

$$s(d) = \frac{\sum_{i=1}^{count(d)} f(i) * vt(i)}{count(d)} \qquad (2.2)$$

trong đó, *vt(i)* là thứ hạng của n-gram *i* trong *DS(n-gram)*.

- *f4* và *f12*: *ma(d)* là trung bình phân bố tần suất chung của các n-gram của tên miền *d*, được tính theo công thức:

$$ma(d) = \frac{m(d)}{len(d)} \qquad (2.3)$$

trong đó, *len(d)* là tổng số các n-gram có trong tên miền *d*.

- *f5* và *f13*: *sa(d)* là trung bình trọng số n-gram của tên miền *d*, được tính theo công thức:

$$sa(d) = \frac{s(d)}{len(d)} \qquad (2.4)$$

- *f6* và *f14*: *tan(d)* là trung bình số lượng n-gram phổ biến của tên miền *d*, được tính theo công thức:

$$tan(d) = \frac{count(d)}{len(d)} \qquad (2.5)$$

- *f7* và *f15*: *taf(d)* là trung bình tần suất n-gram phổ biến của tên miền *d*, được tính theo công thức:

$$taf(d) = \frac{\sum_{i=1}^{count(d)} f(i)}{len(d)} \quad (2.6)$$

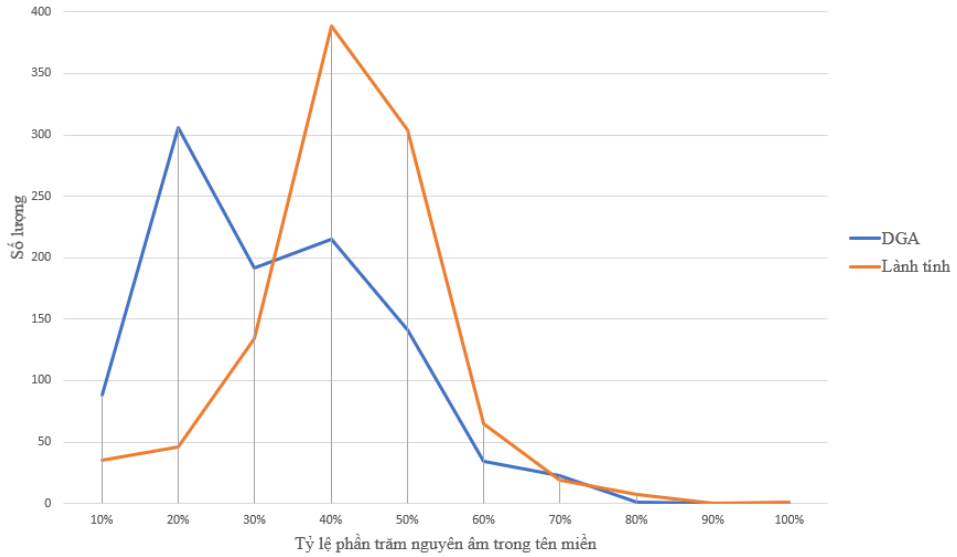
- *f8* và *f16*: *ent(d)* là entropy theo n-gram của tên miền d, tính theo công thức:

$$ent(d) = -\sum_{i=1}^{count(d)} \frac{vt(i)}{L} * log(\frac{vt(i)}{L}) \quad (2.7)$$

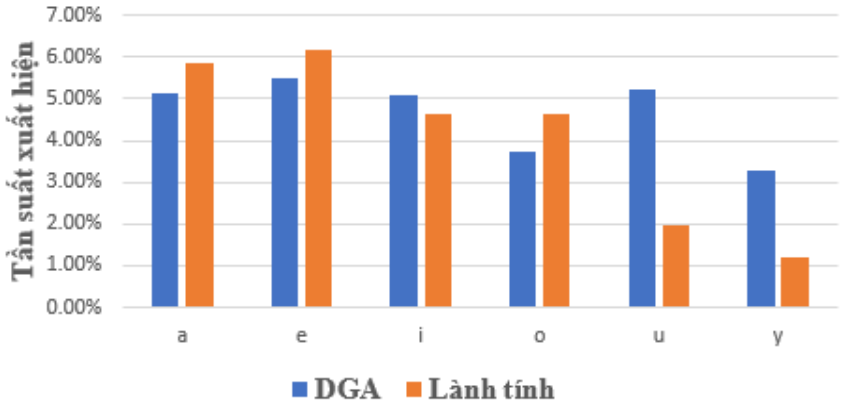
trong đó, *L* là số cụm n-gram phổ biến trong tệp tên miền lành tính, *L=N* với bi-gram và *L=M* với tri-gram.

### *2.2.4.3. Các đặc trưng loại ký tự*

Theo [87] [102], các tên miền lành tính thường có số lượng nguyên âm cao hơn so với các tên miền sinh tự động bởi botnet, do các tên miền lành tính thường được xây dựng dựa trên các đặc trưng ngữ pháp của ngôn ngữ tự nhiên, trong khi đó các tên miền do botnet sinh tự động thường là các ký tự ngẫu nhiên và không có ý nghĩa. Thống kê trên 100,000 tên miền lành tính và 100,000 tên miền DGA botnet thấy rằng đối với các tên miền lành tính thì tỷ lệ nguyên âm trong tên miền đa số chiếm từ 40% đến 50%, đối với tên miền DGA botnet thì tỷ lệ này chiếm đa số ở mức 20%, như minh họa trên Hình 2.10.

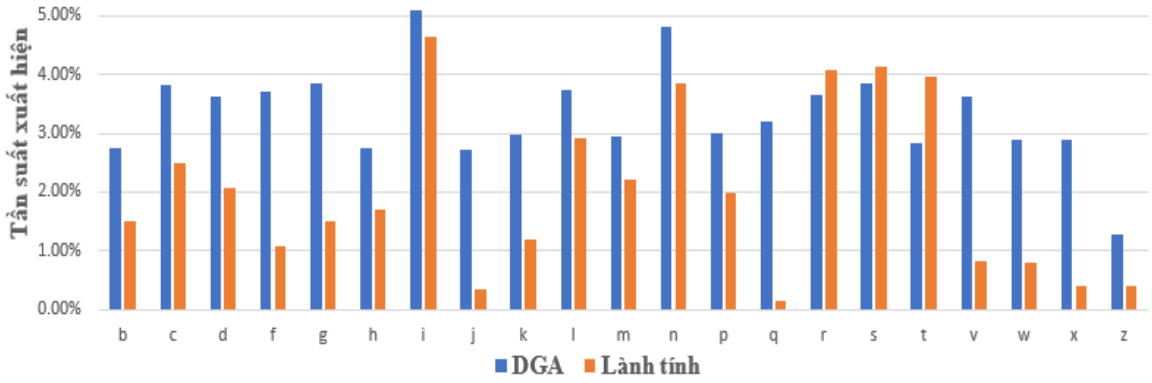



*Hình 2.10: Biểu đồ phân bố tần suất xuất hiện nguyên âm trong tên miền*




*Hình 2.11: Tần suất xuất hiện các nguyên âm*

Biểu đồ trong Hình 2.11 mô tả tần suất xuất hiện của các nguyên âm đã được thống kê từ tập dữ liệu huấn luyện. Rất dễ nhận ra rằng, tần suất xuất hiện của các nguyên âm "u" và "y" ở tệp tên miền lành tính là khá thấp lần lượt là 1.99% và 1.19%, trong khi đó tỷ lệ này là 5.20% và 3.28% đối với tệp các tên miền DGA botnet. Đối với các nguyên âm "a", "e" và "o" vẫn có sự chênh lệch giữa tên miền lành tính và tên miền DGA botnet, tuy nhiên không nhiều.




*Hình 2.12: Tần suất xuất hiện các phụ âm*

Tần suất xuất hiện của các phụ âm được minh họa tại Hình 2.12. Sự chênh lệch về tần suất xuất hiện của các phụ âm giữa tên miền lành tính và tên miền DGA botnet là tương đối lớn, điển hình đối với các phụ âm "f", "g", "j", "q", "v", "w", "x" và "z". Tần suất xuất hiện của các phụ âm và nguyên âm đối với tên miền DGA botnet là khá đều nhau do việc sinh ngẫu nhiên, nhưng với tên miền lành tính thì tần suất xuất hiện của phụ âm và nguyên âm là không đều nhau do tính chất đặc trưng của ngôn ngữ.

Với những thống kê và phân tích ở trên, nhận thấy rằng phân bố nguyên âm, phụ âm trong tên miền lành tính và tên miền DGA botnet có sự khác biệt. Do đó, đề xuất 2 đặc trưng f17 và f18:

- *f17*: *tanv(d)* là phân bố nguyên âm của tên miền *d*. Với *countnv(d)* là số nguyên âm trong miền *d* , *tanv(d)* được tính theo công thức:

$$tanv(d) = \frac{countnv(d)}{len(d)} \quad (2.8)$$

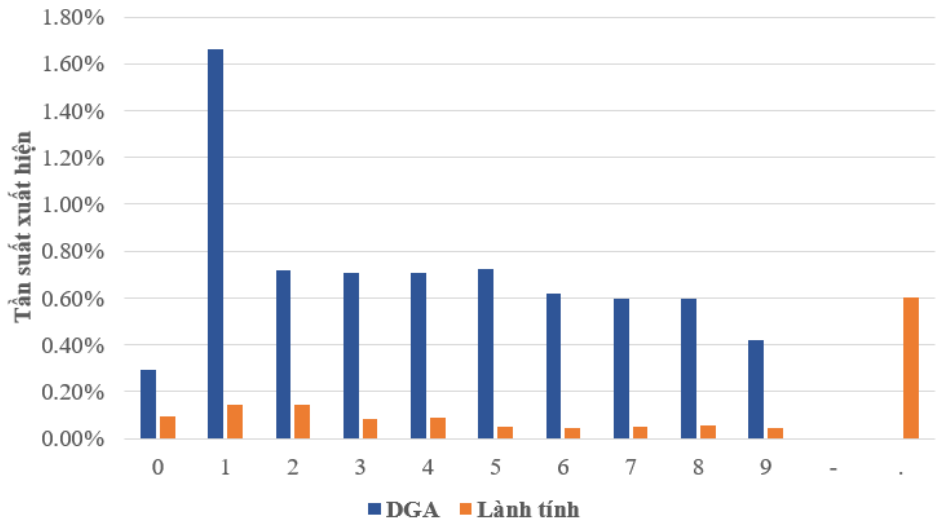
- *f18*: *tanco(d)* là phân bố phụ âm của tên miền *d*. Với *countco(d)* là số phụ âm trong miền *d*, *tanco(d)* được tính theo công thức:

$$tanco(d) = \frac{countco(d)}{len(d)} \quad (2.9)$$

*Bảng 2.7: Thống kê tên miền có ký tự số, "-" và "."*

| Chỉ số thống kê | DGA | | Lành tính | |
|---|---|---|---|---|
| | Số lượng | Tỷ lệ | Số lượng | Tỷ lệ |
| Tên miền có chứa ký tự “-” và “.” | 0 | 0.00% | 8126 | 8.23% |
| Tên miền có chứa ký tự là số | 18,617 | 18.62% | 6055 | 6.06% |
| Tên miền có ký tự đầu tiên là số | 7,010 | 7.01% | 1866 | 1.87% |

Từ dữ liệu thống kê được mô tả tại Bảng 2.7 và Hình 2.13 cho thấy, với 100,000 tên miền DGA botnet thì có 18.62% tên miền có chứa ký tự là số, chỉ số này với tên miền lành tính là 6.06%. Số lượng các tên miền có ký tự đầu tiên là số của các tên miền DGA botnet cũng chiếm tỷ trọng cao hơn so với các tên miền lành tính, cụ thể là 7.01% và 1.87% tương ứng với tên miền DGA botnet và tên miền lành tính. Trong số 200,000 tên miền DGA botnet và tên miền lành tính được thống kê cho thấy có 8126 tên miền có chứa ký tự “-” và “.” chiếm 8.23% đều là các tên miền lành tính, chỉ số này đối với tên miền DGA botnet là 0.00%. Từ các phân tích trên, các đặc trưng f19, f20, f22 được trích chọn:




*Hình 2.13: Tần suất xuất hiện các ký tự số, "-" và "."*

- f19: *tandi(d)* là phân bố chữ số của tên miền *d*. Với *countdi(d)* là số các chữ số trong miền *d*, *tandi(d)* được tính theo công thức:

$$tandi(d) = \frac{countdi(d)}{len(d)} \quad (2.10)$$

- *f20*: *tansc(d)* là phân bố ký tự đặc biệt của tên miền *d*. Với *countsc(d)* là số ký tự đặc biệt trong miền *d*, *tansc(d)* được tính theo công thức:

$$tansc(d) = \frac{countsc(d)}{len(d)} \quad (2.11)$$

Bảng 1.5 cho thấy trong 53 họ DGA botnet thống kê trong Netlab 360 thì có 8 họ chỉ sử dụng ký tự hexa để sinh tên miền, trong đó: 02 họ tên miền có độ dài là 32 ký tự, 02 họ có độ dài là 16 ký tự, 02 họ có độ dài là 8 ký tự. Số tên miền DGA sinh ra khoảng 3000 tên miền/ ngày. Đây là sự khác biệt rất lớn giữa tên miền lành tính và tên miền DGA, do đó đặc trưng phân bố ký tự trong hệ đếm hexa của tên miền được trích chọn để sử dụng trong huấn luyện và kiểm thử.

- *f21*: *tanhe(d)* là phân bố ký tự trong hệ đếm hexa của tên miền *d*. Với *counthe(d)* là số ký tự hexa trong miền *d*, *tanhe(d)* được tính theo công thức:

$$tanhe(d) = \frac{counthe(d)}{len(d)} \quad (2.12)$$

- *f22*: *is_digit* trả về giá trị 1 khi ký tự đầu tiên của tên miền d là số, 0 khi ký tự đầu là ký tự.

### *2.2.4.4. Các đặc trưng thống kê*

Về nguyên tắc, các thuật toán sinh tên miền tự động DGA được thiết kế để sinh các tên miền tránh bị trùng lặp với các tên miền lành tính hiện có. Trên thực tế, thông qua phân tích thống kê, thấy rằng phân phối tần suất của 38 ký tự (*bao gồm 26 chữ cái, 10 chữ số, dấu chấm và dấu gạch ngang*) có sự khác biệt đáng kể giữa tên miền lành tính và tên miền DGA. Tên miền DGA có phân phối rất ổn định và đều nhau, trong khi phân phối của tên miền lành tính thường không ổn định và không đồng đều. Do đó, dựa trên đặc điểm này, đề xuất sử dụng 2 đặc trưng thống kê: đặc trưng entropy của ký tự x trong tên miền d để tính toán mức độ ngẫu nhiên của các ký tự trong tên miền (f23) và đặc trưng giá trị dự kiến, hay giá trị kỳ vọng cho mỗi tên miền (f24).

- *f23*: *ent_char(d)* là entropy theo ký tự của miền *d*. Với *D(x)* là phân phối xác suất của ký tự *x* trong miền *d*, *ent_char(d)* được tính bởi công thức:

$$ent\_char(d) = -\frac{\sum_x D(x)\log(D(x))}{\log(len(d))} \qquad (2.13)$$

- f24: *EOD(d)* là giá trị kỳ vọng của tên miền *d [96]*. Giả thiết tên miền *d* bao gồm *k* ký tự $\{x_1, x_2,...,x_k\}$, $n(x_i)$ là tần suất xuất hiện của ký tự $x_i$ và $p(x_i)$ là phân phối xác suất của ký tự $x_i$. Giá trị $p(x_i)$ được tính bằng cách sử dụng 100.000 tên miền hàng đầu được liệt kê bởi Alexa, cho như trên Bảng 2.8. *EOD(d)* được tính bởi công thức:

$$EOD(d) = \frac{\sum_{i=1}^{k} n(x_i)\,p(x_i)}{\sum_{i=1}^{k} n(x_i)} \qquad (2.14)$$

*Bảng 2.8: Xác suất của 38 ký tự xuất hiện trong 100.000 tên miền lành tính*

| C | P(C) | C | P(C) | C | P(C) | C | P(C) | C | P(C) | C | P(C) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| a | 9.35 | g | 2.40 | m | 3.37 | s | 6.48 | y | 1.67 | 5 | 0.10 |
| b | 2.27 | h | 2.56 | n | 6.12 | t | 6.13 | x | 0.68 | 6 | 0.09 |
| c | 3.87 | i | 7.40 | o | 7.28 | u | 3.23 | 0 | 0.18 | 7 | 0.09 |
| d | 3.26 | j | 0.55 | p | 2.91 | v | 1.37 | 1 | 0.24 | 8 | 0.10 |
| e | 9.69 | k | 1.90 | q | 0.21 | w | 1.20 | 2 | 0.23 | 9 | 0.08 |
| f | 1.67 | l | 4.56 | r | 6.44 | x | 0.67 | 3 | 0.15 | . | 0.00 |
| | | | | | | | | 4 | 0.16 | - | 1.26 |

### 2.2.5. Thử nghiệm và kết quả

#### *2.2.5.1. Kịch bản thử nghiệm*

Tập dữ liệu huấn luyện gồm 200,000 tên miền tại Bảng 2.3 được sử dụng để xây dựng và kiểm tra hiệu suất của mô hình CDM sử dụng thuật toán máy học rừng ngẫu nhiên (*37-trees*). Luận án sử dụng phương pháp kiểm tra chéo 10 lần (*10-fold cross-validation*) với 80% tập dữ liệu lấy ngẫu nhiên cho huấn luyện và 20% còn lại cho kiểm tra để tính kết quả trung bình hiệu suất phát hiện của mô hình đề xuất. Kết quả sẽ được so sánh với các đề xuất trước đây để đánh giá hiệu suất của mô hình.

Để so sánh và chứng minh hiệu quả của mô hình CDM với 24 đặc trưng ký tự, tập huấn luyện sẽ được sử dụng để kiểm tra hiệu suất của mô hình sử dụng 18 đặc trưng được đề xuất bởi Hoang và cộng sự [24].

Mô hình CDM sau huấn luyện được sử dụng cho thử nghiệm phát hiện sử dụng tập 71,393 tên miền DGA boetnet sinh bởi 39 họ lấy tại Netlab360 [11] và 31,000 tên miền DGA botnet sinh ra bởi 7 họ tại UMUDGA [60] (7 họ này không được công bố trong Netlab 360 - Bảng 2.4) để tính toán tỷ lệ phát hiện (DR) cho mỗi họ botnet cũng như tỷ lệ phát hiện chung trên toàn tập kiểm thử. Ngoài giá trị DR của mỗi họ tên miền, giá trị DR chung là tỷ lệ của tổng số tên miền phát hiện chính xác và tổng số tên miền thử nghiệm.

#### *2.2.5.2. Kết quả thử nghiệm*

Bảng 2.9 cho thấy, hiệu suất của CDM tốt hơn so với hiệu suất của Hoang và cộng sự [24] khi sử dụng cùng tập huấn luyện với F1 và ACC lần lượt là 99.60% và 99.60% so với 94.60% và 94.61%. Tỷ lệ dương tính giả và âm tính giả của CDM cũng giảm đáng kể, cụ thể là tỷ lệ dương tính giả và âm tính giả của CDM lần lượt là 0.43% và 0.38% so với 5.13% và 5.67% của mô hình đề xuất bởi Hoang và cộng sự [24]. Như vậy, có thể khẳng định tập 24 đặc trưng ký tự sử dụng trong CDM có khả năng phân loại các tên miền DGA tốt hơn so với tập 18 đặc trưng trong Hoang và cộng sự [24].

*Bảng 2.9: Hiệu suất của mô hình CDM so với Hoang và cộng sự [24]*

| Mô hình phát hiện | PPV | TPR | FPR | FNR | ACC | F1 |
|---|---|---|---|---|---|---|
| Hoang và cộng sự [24] | 94.87 | 94.33 | 5.13 | 5.67 | 94.60 | 94.61 |
| CDM | 99.57 | 99.62 | 0.43 | 0.38 | 99.60 | 99.60 |

Bảng 2.10 so sánh hiệu suất của mô hình CDM với hiệu suất của các mô hình phát hiện đã có. Có thể thấy mô hình phát hiện CDM cho hiệu suất tốt hơn đáng kể so với các đề xuất của Truong và cộng sự [96], Hoang và cộng sự [24], Qiao và cộng sự [70], Zhao và cộng sự [26].

Các Bảng 2.11, Bảng 2.12 và Bảng 2.13 cung cấp tỷ lệ phát hiện (DR) của mô hình CDM trong giai đoạn phát hiện trên tập tên miền DGA botnet sinh bởi 39 họ DGA botnet chia tương ứng thành 3 nhóm: nhóm có DR>=90%, nhóm có 90% > DR>=50% và nhóm có DR < 50%.

*Bảng 2.10: Hiệu suất của mô hình CDM so với các mô hình trước đó*

| Mô hình phát hiện | PPV | TPR | FPR | FNR | ACC | F1 |
|---|---|---|---|---|---|---|
| Truong và cộng sự [96] | 94.70 | | 4.80 | | 92.30 | |
| Hoang và cộng sự [24] | 90.70 | 91.00 | 9.30 | | 90.90 | 90.90 |
| Qiao và cộng sự [70] | 95.05 | 85.14 | | | | 94.58 |
| Zhao và cộng sự [26] | | | 6.14 | 7.42 | 94.04 | |
| Mô hình đề xuất CDM | 99.57 | 99.62 | 0.43 | 0.38 | 99.60 | 99.60 |

*Bảng 2.11: Các họ botnet có tỷ lệ phát hiện (DR) lớn hơn 90%*

| STT | Họ DGA botnet | Tổng số tên miền | Phát hiện chính xác | DR% |
|---|---|---|---|---|
| 1 | emotet | 4000 | 3987 | 99.68 |
| 2 | gameover | 4000 | 4000 | 100.00 |
| 3 | murofet | 4000 | 3992 | 99.80 |
| 4 | necurs | 4000 | 3974 | 99.35 |
| 5 | pykspa_v1 | 4000 | 3988 | 99.70 |
| 6 | ramnit | 4000 | 3982 | 99.55 |
| 7 | ranbyus | 4000 | 3983 | 99.58 |
| 8 | rovnix | 4000 | 4000 | 100.00 |
| 9 | shiotob | 4000 | 3987 | 99.68 |
| 10 | symmi | 1200 | 1159 | 96.58 |
| 11 | tinba | 4000 | 3999 | 99.98 |
| 12 | simda | 4000 | 3986 | 99.65 |
| 13 | virut | 4000 | 3990 | 99.75 |

| | | | | |
|---|---|---|---|---|
| 14 | proslikefan | 100 | 98 | 98.00 |
| 15 | tempedreve | 195 | 190 | 97.44 |
| 16 | tinynuke | 32 | 32 | 100.00 |
| 17 | vidro | 100 | 100 | 100.00 |
| 18 | pykspa_v2_real | 199 | 197 | 98.99 |
| 19 | pykspa_v2_fake | 799 | 790 | 98.87 |
| 20 | padcrypt | 168 | 165 | 98.21 |
| 21 | nymaim | 480 | 455 | 94.79 |
| 22 | vawtrak | 827 | 799 | 96.61 |
| 23 | shifu | 2546 | 2510 | 98.59 |
| 24 | fobber_v1 | 298 | 298 | 100.00 |
| 25 | fobber_v2 | 299 | 299 | 100.00 |
| 26 | dircrypt | 762 | 757 | 99.34 |
| 27 | cryptolocker | 1000 | 997 | 99.70 |
| 28 | locky | 1158 | 1147 | 99.05 |
| 29 | chinad | 1000 | 1000 | 100.00 |
| 30 | qadars | 2000 | 1981 | 99.05 |
| 31 | dyre | 1000 | 1000 | 100.00 |
| **Tổng** | | **62163** | **61842** | **99.48** |

*Bảng 2.12: Các họ botnet có tỷ lệ phát hiện (DR) từ 50%-90%*

| STT | Họ DGA botnet | Tổng số tên miền | Phát hiện chính xác | DR% |
|---|---|---|---|---|
| 1 | mydoom | 50 | 44 | 88.00 |
| 2 | gspy | 100 | 76 | 76.00 |
| 3 | enviserv | 500 | 252 | 50.40 |
| 4 | conficker | 495 | 442 | 89.29 |
| | **Tổng cộng** | **1145** | **814** | **71.09** |

*Bảng 2.13: Các họ botnet có tỷ lệ phát hiện thấp*

| STT | Họ DGA botnet | Tổng số tên miền | Phát hiện chính xác | DR% |
|---|---|---|---|---|
| 1 | banjori | 4000 | 0 | 0 |
| 2 | matsnu | 881 | 107 | 12.15 |
| 3 | bigviktor | 999 | 111 | 11.11 |
| 4 | suppobox | 2205 | 425 | 19.27 |
| | **Tổng cộng** | **8085** | **643** | **7.95** |

*Bảng 2.14: Tỷ lệ phát hiện của CDM trên tập dữ liệu UMUDGA*

| STT | Họ DGA botnet | Tổng số tên miền | Phát hiện chính xác | DR% |
|---|---|---|---|---|
| 1 | alureon | 5000 | 4911 | 98.22 |
| 2 | bedep | 5000 | 4991 | 99.82 |
| 3 | corebot | 5000 | 4988 | 99.76 |
| 4 | kraken | 2000 | 1968 | 98.40 |
| 5 | pushdo | 5000 | 4718 | 94.40 |
| 6 | zeus | 5000 | 5000 | 100.00 |
| | | **27000** | **26576** | **98.43** |
| 7 | pizd | 4000 | 642 | 16.05 |
| | **Tổng cộng** | **31000** | **27218** | **87.85** |

### 2.2.6. Đánh giá

Dựa vào kết quả thử nghiệm ở các Bảng 2.10, Bảng 2.11, Bảng 2.12 và Bảng 2.13, có thể rút ra những nhận xét sau: Mô hình phát hiện CDM hoạt động tốt hơn các đề xuất trước đó với tất cả các độ đo, trong đó mô hình đề xuất cho độ chính xác và độ đo F1 cao hơn đáng kể so với các mô hình trước đó. Chẳng hạn, độ đo F1 của Hoang và cộng sự [24], Qiao và cộng sự [70] và mô hình phát hiện CDM đề xuất tương ứng là 90.90%, 94.58% và 99.59%. Ngoài ra, tỷ lệ dương tính giả (FPR) và tỷ lệ âm tính giả (FNR) của mô hình phát hiện CDM đề xuất cũng thấp hơn đáng kể so với các mô hình trước đó, như thể hiện trên Bảng 2.10.

Thông qua phát hiện thử nghiệm trên 39 họ botnet cho thấy, mô hình CDM có khả năng phát hiện hiệu quả hầu hết các họ DGA botnet. Trong số 39 DGA botnet, 31 họ DGA botnet được phát hiện với tỷ lệ phát hiện trên 90%, như trình bày trong Bảng 2.11. Tỷ lệ phát hiện trung bình của nhóm DGA botnet này là 99.48%. Bốn DGA botnet trong nhóm thứ hai, như trong Bảng 2.12 cũng có DR trung bình tương đối cao là 71.09%. Lý do mà mô hình CDM đề xuất hoạt động tốt trong phát hiện tên miền DGA botnet thuộc các nhóm này là bởi tập 24 đặc trưng ký tự đề xuất trong mô hình là phù hợp cho phân biệt các tên miền character-based DGA và các tên miền lành tính.

Bảng 2.14 thể hiện tỷ lệ phát hiện của mô hình CDM với bộ dữ liệu UMUDGA. Theo đó, có thể thấy rằng với 06 họ character-based DGA botnet, tỷ lệ phát hiện đạt 98.43%. Đây là những botnet không được công bố trong tập dữ liệu Netlab 360, không được sử dụng để huấn luyện mô hình, điều này khẳng định CDM có thể phát hiện hiệu quả các character-based DGA botnet mới.

*Như thể hiện trong các Bảng 2.13 và*

Bảng 2.14, mô hình CDM không phát hiện được các tên miền được sử dụng bởi 4 họ DGA botnet, bao gồm ‘banjori’, ‘matsnu’, ‘bigviktor’ và ‘suppobox’ thuộc tập dữ liệu Netlab 360 và 1 họ botnet ‘pizd’ thuộc tập dữ liệu UMUDGA. Cụ thể, mô hình không thể phát hiện bất kỳ tên miền nào được tạo bởi botnet ‘banjori’ và chỉ có thể phát hiện một số tên miền được tạo bởi botnet ‘matsnu’, ‘bigviktor’, ‘suppobox’ và ‘pizd’. Điều này là do các DGA botnet này sử dụng các thuật toán DGA có thể tạo ra các tên miền rất giống với các tên miền lành tính. Đây cũng là vấn đề sẽ được giải quyết trong các mô hình phát hiện trong mục tiếp theo của luận án.

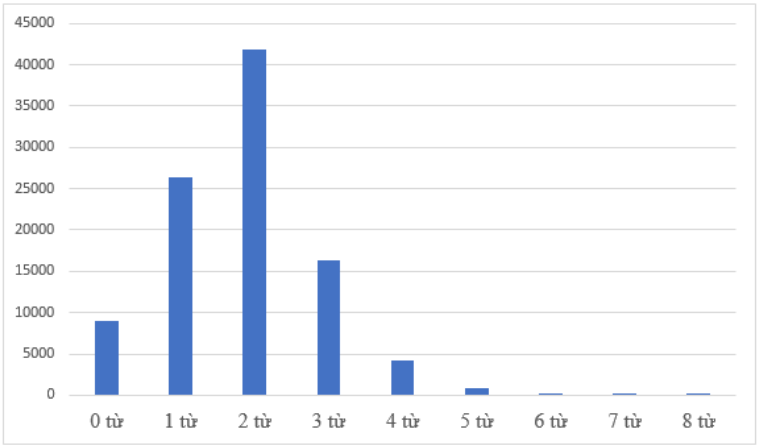
## 2.3. PHÁT HIỆN WORD-BASED DGA BOTNET SỬ DỤNG CÁC ĐẶC TRƯNG TỪ

### 2.3.1. Đặt vấn đề

Mục 2.1.1.2 đã trình bày sơ lược về các dạng DGA botnet, trong đó có word-based và mixed DGA botnet (từ đây luận án gọi chung là word-based DGA botnet). Mục này đi sâu phân tích các đặc điểm của hai dạng botnet này. Khác với character-based DGA botnet, word-based DGA botnet sinh các tên miền bằng cách tổ hợp các từ tiếng Anh lấy từ các danh sách các từ được lập sẵn. Các tên miền DGA dạng này thường chứa hai hoặc ba từ được lấy các danh sách từ khác nhau được chọn và nối ngẫu nhiên. Cuối cùng, một TLD được thêm vào cuối giống như một tên miền thông thường, ví dụ như tên miền *crossmentioncare.com*. Rovnix là một trong những họ DGA botnet tạo các tên miền CnC bằng cách sử dụng các từ trong Tuyên ngôn Độc lập của Hoa Kỳ và một số tài liệu khác. Theo các nhà nghiên cứu bảo mật, các họ

DGA botnet tiến hóa, như Matsnu sử dụng một kỹ thuật thông minh để tránh các cơ chế kiểm tra thông thường. Cụ thể, Matsnu tạo các tên miền dựa trên sự pha trộn của các danh từ và động từ. Các danh từ và động từ có thể được nhập vào thủ công, hoặc lấy từ danh sách xác định trước, như một danh sách chứa 878 danh từ và một danh sách sách chứa 444 động từ. Matsnu cho phép cấu hình thiết lập số lượng tên miền muốn tạo trong một ngày một cách dễ dàng. Tên miền do Matsnu tạo ra có thể sử dụng các ký tự như "-" để nối các từ (*như world-bite-care.com*), hoặc không dùng ký tự nối (*như activitypossess.com*).

Khó khăn lớn nhất cho phát hiện các word-based DGA botnet là chúng có khả năng sinh các tên miền được tổ hợp từ các từ tiếng Anh có nghĩa và *các tên miền này rất giống so với các tên miền lành tính đang được sử dụng rộng rãi*. Điều này được chứng minh trong Yang và cộng sự [50] khi phân tích một triệu tên miền hàng đầu trong danh mục Cisco Umbrella [97], đã phát hiện ra rằng hơn 67% tên miền chứa ít nhất một từ tiếng Anh và gần 30% tên miền hoàn toàn bao gồm các từ tiếng Anh. Và theo thống kê trong dữ liệu thực nghiệm với 98,866 tên miền lành tính nguyên tố [17] có thứ hạng cao nhất, thì số tên miền lành tính không có từ tiếng Anh chỉ chiếm 9.05%, có 1 từ chiếm 26.70% và có 2 từ chiếm tới 42.34%, như minh họa trên Hình 2.14.




*Hình 2.14: Biểu đồ phân bố các tên miền với số lượng từ tương ứng*

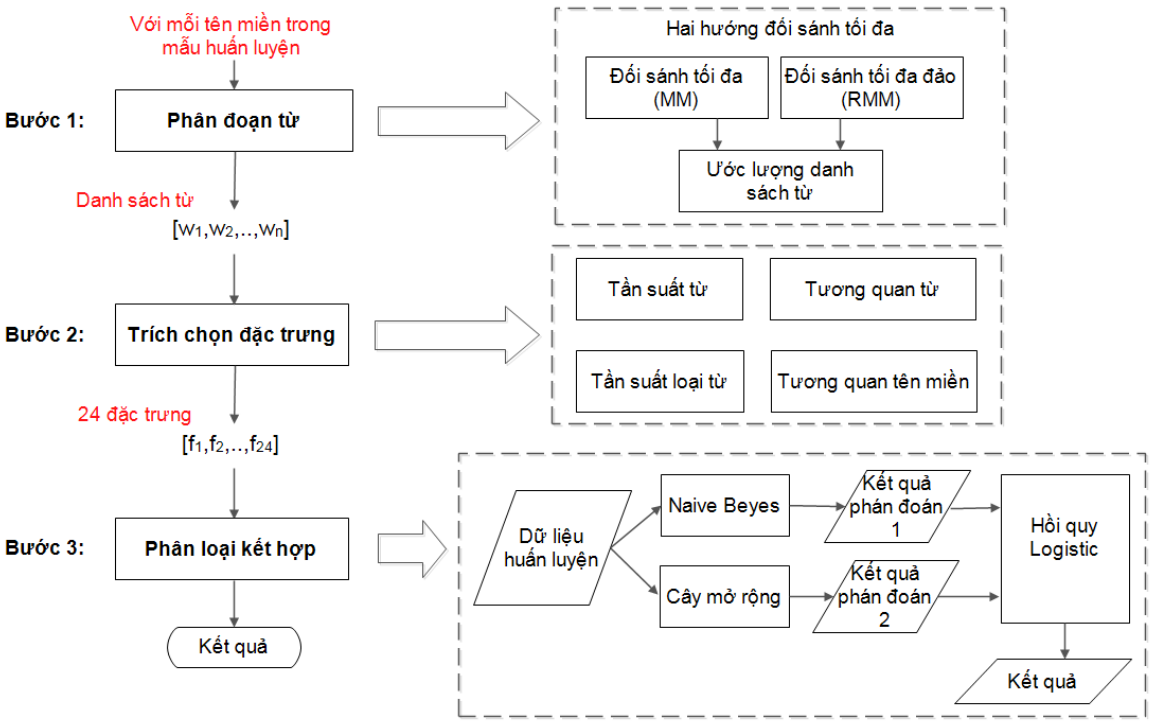
Do các tên miền do các word-based DGA botnet, như bigviktor, matsnu, ngioweb và suppobox sinh ra rất giống các tên miền lành tính, nhiều phương pháp phát hiện DGA botnet dựa trên phân loại tên miền [35], như Hoang và cộng sự [24] và mô hình CDM đề xuất ở mục 2.2 của luận án hầu như không thể phát hiện các DGA botnet này. Cụ thể như, mô hình CDM đề xuất sử dụng tập đặc trưng ký tự có khả năng phát hiện các character-based DGA botnet với tỷ lệ phát hiện đúng bình quân là 99.48%, nhưng không thể phát hiện tên miền nào của banjori và chỉ có thể phát hiện một số tên miền được tạo bởi matsnu, bigviktor và suppobox. Như vậy, có thể thấy cần nghiên cứu phát triển các mô hình sử dụng các tập đặc trưng phù hợp hơn cho phép  phát hiện hiệu quả các word-based DGA botnet. Đây cũng là vấn đề được giải quyết trong phần này của luận án.

### 2.3.2. Các phương pháp phát hiện word-based DGA botnet

Mục này khảo sát một số nghiên cứu phát hiện word-based và mixed DGA botnet, bao gồm các đề xuất phát hiện dựa trên học máy có giám sát truyền thống và các đề xuất phát hiện dựa trên học sâu.

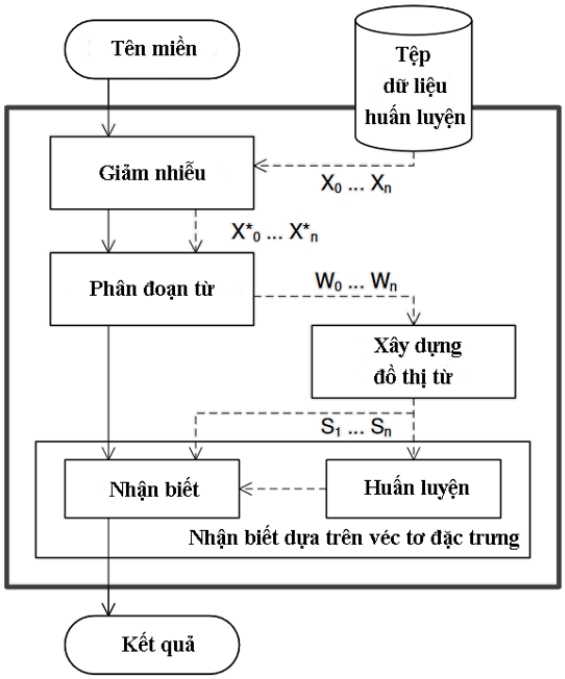
#### *2.3.2.1. Phát hiện dựa trên học máy có giám sát truyền thống*

Yang và cộng sự [50] đề xuất một phương pháp phát hiện mới cho các word-based DGA botnet bằng cách phân tích các đặc điểm ngữ nghĩa của tên miền, bao gồm phân phối theo từ, phân bố theo ký tự và mối tương quan của chúng. Phương pháp đề xuất gồm 3 bước: phân đoạn từ, trích chọn đặc trưng và phân loại kết hợp, như biểu diễn trên Hình 2.15. Các nhóm đặc trưng gồm tần xuất từ, tương quan từ, tần suất loại từ và tương quan tên miền được trích xuất để đưa vào bước phân loại kết hợp. Bộ phân loại kết hợp được xây dựng sử dụng các thuật toán học máy, bao gồm Naive Bayes, Cây mở rộng (*Extra-Trees*) và Hồi qui Logistic. Các kết quả đánh giá dựa trên các mẫu tên miền độc hại và hợp pháp được trích xuất từ các tập dữ liệu công khai cho thấy phương pháp đề xuất có khả năng phát hiện hiệu quả các word-based DGA botnet. Kết quả phát hiện đạt được với WB-DGA, Matsnu và Suppobox lần lượt là: 92.21%, 88.09% và 83.64%.




*Hình 2.15: Nền tảng phát hiện word-based DGA botnet [50]*

Ở một cách tiếp cận khác, Satoh và cộng sự [79] đề xuất một phương pháp xác định tên miền độc hại bằng cách phân tích các chuỗi ký tự ở cấp độ từ của tên miền trích xuất từ truy vấn DNS. Ý tưởng của phương pháp này dựa trên 2 khía cạnh: (i) sử dụng các kỹ thuật trong lý thuyết đồ thị chung để biểu diễn mối quan hệ giữa các từ trong tên miền và (ii) sử dụng trọng tâm để định lượng tầm quan trọng của mỗi từ trong đồ thị đã xây dựng. Các bước xử lý của phương pháp đề xuất bao gồm: Giảm nhiễu, Phân đoạn từ, Xây dựng đồ thị từ, Nhận biết và Huấn luyện, như biểu diễn trên Hình 2.16. Kết quả của các thí nghiệm được tiến hành như một phần của nghiên cứu này có thể đạt được ACC, TPR, PPV lần lượt là 99.89%, 99.77% và 98.69%.




*Hình 2.16: Tổng quan về hướng tiếp cận theo đề xuất của Satoh [79]*

#### *2.3.2.2. Phát hiện dựa trên học sâu*

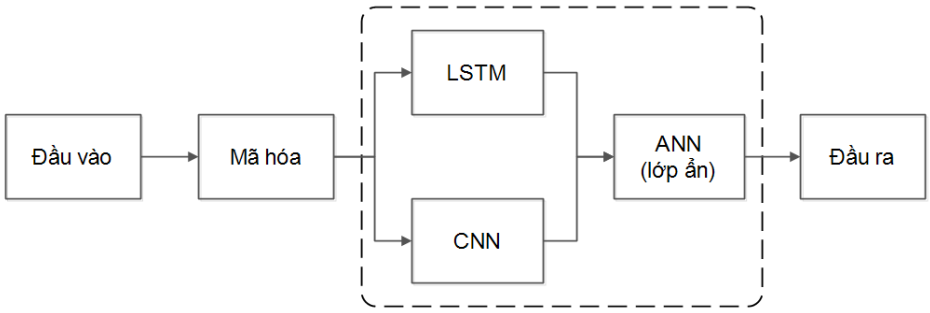
Học sâu đã được ứng dụng rộng rãi trong những năm gần đây trong nhiều lĩnh vực và đặc biệt đạt nhiều thành công trong lĩnh vực xử lý ảnh và xử ngôn ngữ tự nhiên. Học sâu cũng đã và đang được sử dụng trong các mô hình phát hiện tấn công, xâm nhập, cũng như trong phát hiện botnet. Woodbridge và cộng sự [35] là một trong những nhóm đầu tiên sử dụng phương pháp học sâu LSTM để phân loại các tên miền DGA. Mô hình chấp nhận các chuỗi ký tự có độ dài thay đổi làm đầu vào, do đó không có yêu cầu phụ trợ nào cho việc trích xuất đặc trưng. Một ưu điểm khác của phương pháp đề xuất là mô hình sử dụng nhỏ gọn, chỉ bao gồm một lớp nhúng, một lớp mạng LSTM và một lớp đầu ra được kết nối đầy đủ. Mặc dù việc huấn luyện trên một tập dữ liệu có kích thước lớn yêu cầu nhiều tài nguyên tính toán, nhưng cấu trúc nhỏ gọn cho phép thời gian xử lý tương đối nhanh. Tuy nhiên, phương pháp đề xuất dựa trên LSTM không đạt độ chính xác cao với các tên miền của '*suppobox*' hoặc '*matsnu*' - là các họ word-based DGA botnet. Các kiến trúc học sâu khác sau đó cũng đã được áp dụng để xây dựng bộ phân loại tên miền, chẳng hạn như các biến thể của LSTM, mạng CNN và mô hình kết hợp CNN-LSTM. Mặc dù thành công đối với các

tên miền character-based DGA, các bộ phân loại này phần lớn không hiệu quả trong phân loại các tên miền word-based DGA. Các mô hình phân loại đề xuất cũng hoạt động tốt trên các bộ dữ liệu thử nghiệm khác nhau, nhưng hiệu suất phát hiện có thể bị ảnh hưởng khi cố gắng tổng quát hóa thành các họ DGA mới hoặc các phiên bản mới của các họ đã thấy trước đó [39].

Theo một hướng khác, Tran và cộng sự [94] đã tính đến sự mất cân bằng lớp nền của dữ liệu tên miền DGA. Một số nhóm nghiên cứu khác thực hiện cập nhật dữ liệu huấn luyện với các bộ dữ liệu DGA đã biết, hoặc bổ sung thêm thông tin ngữ cảnh vào dữ liệu huấn luyện. Hơn nữa, một số đề xuất thay đổi kiến trúc ban đầu của LSTM thành Bi-LSTM, thể hiện những cải tiến tiềm năng của việc thay đổi kiến trúc của mô hình. Khi CNN được áp dụng để phân loại văn bản [101] và cho thấy sự thành công so với LSTM trong một số nhiệm vụ [84], nó được áp dụng một cách tự nhiên vào phân tích URL độc hại [80]. Các đánh giá gần đây chỉ ra rằng học sâu duy trì thành công hơn so với các mô hình rừng ngẫu nhiên được huấn luyện bằng cách sử dụng các đặc trưng được lựa chọn thủ công, nhưng không xem xét bối cảnh lớn hơn của môi trường triển khai hoặc triển khai của mô hình . Tuy nhiên, đánh giá một cách có hệ thống hầu hết các hệ thống phát hiện DGA luôn hoạt động kém hiệu quả trên các họ word-based DGA botnet [7] [8].

Koh và cộng sự là một trong những nhóm nghiên cứu đầu tiên xây dựng mô hình phân loại dựa trên học sâu nhằm phân loại các tên miền word-based DGA [41]. Bằng cách sử dụng phương pháp nhúng được huấn luyện trước cho các từ trong tên miền, họ đã huấn luyện LSTM trên cả tập dữ liệu DGA botnet đơn và nhiều DGA botnet. Mặc dù mục tiêu của mô hình đề xuất là phát hiện word-based DGA botnet, nhưng mô hình đề xuất có những hạn chế nghiêm trọng từ việc nhúng từ nhạy cảm theo ngữ cảnh vào những gì mô hình có thể học đến việc mô hình không sử dụng tất cả dữ liệu có sẵn trong quá trình huấn luyện và thử nghiệm. Một công trình liên quan khác về phát hiện word-based DGA botnet là WordGraph của Pereira và cộng sự [69]. Nghiên cứu sử dụng một tập hợp tên miền truy vấn không tồn tại (*NXDomain*) và trích xuất chuỗi con chung (*LCS*) dài nhất của mọi cặp tên miền trong tập hợp, kết

nối bất kỳ LCS nào tồn tại trong các cặp tên miền để tạo WordGraph. Các tên miền word-based DGA botnet được hiển thị để phân cụm trong khi các miền lành tính không có mẫu rõ ràng và được hiển thị để tổng quát hóa các thay đổi đối với word-based DGA botnet. Một bộ phân loại rừng ngẫu nhiên được huấn luyện để xác định các mẫu word-based DGA botnet. Phương pháp này cho thấy hứa hẹn trong việc thích ứng với các DGA khác nhau.




*Hình 2.17: Kiến trúc Billo [39]*

Highnam và cộng sự [39] giới thiệu một mô hình học sâu lai Bilbo cho phân loại các tên miền word-based DGA botnet theo thời gian thực. Hình 2.17 minh họa kiến trúc cấp cao của Bilbo. Các tên miền thô được nhập và mã hóa thành các chuỗi trước khi được chuyển đến các khối LSTM và CNN riêng biệt. Các đặc trưng được trích xuất bởi mỗi khối thành phần này được gửi đến một lớp ANN duy nhất hoặc một lớp ẩn, sau đó được bổ sung để tạo ra kết quả. Do các tên miền là chuỗi ký tự nên các mô hình LSTM là sự phù hợp tự nhiên để phân loại các tên miền DGA botnet. Các nút LSTM đưa ra quyết định về một phần tử trong chuỗi dựa trên những gì nó đã thấy trước đó trong chuỗi. Do đó, các nút LSTM học các tham số được chia sẻ trên các phần tử của chuỗi. Việc chia sẻ tham số cho phép các LSTM mở rộng quy mô để xử lý các chuỗi dài hơn nhiều so với thực tế đối với các mạng nơ-ron truyền thẳng truyền thống. Bilbo xử lý tên miền qua lớp LSTM và lớp CNN song song. Đầu ra của hai khối này được tổng hợp hoặc "đóng gói" bởi ANN một lớp. Việc "đóng gói" này là một cơ hội quan trọng để mô hình này phân biệt phần nào của thông tin thu được từ LSTM và CNN hỗ trợ tốt nhất khi gắn nhãn các tên miền word-based DGA botnet và các tên miền lành tính. Việc chèn một ANN thay vì một hàm ra duy nhất sẽ làm

tăng khả năng tối ưu hóa tiềm năng của việc "đóng gói". Không giống như quần thể tối ưu hóa các thành phần của nó trước khi ghép nối, mô hình lai tối ưu hóa trên tất cả các thành phần. Bilbo kết hợp thành công các lớp LSTM, CNN và ANN để phát hiện word-based DGA botnet và được đánh giá là công cụ tốt nhất cho phân loại các word-based DGA botnet trong số các mô hình học sâu hiện đại.

#### *2.3.2.3. Ưu điểm và hạn chế của các đề xuất phát hiện word-based DGA botnet*

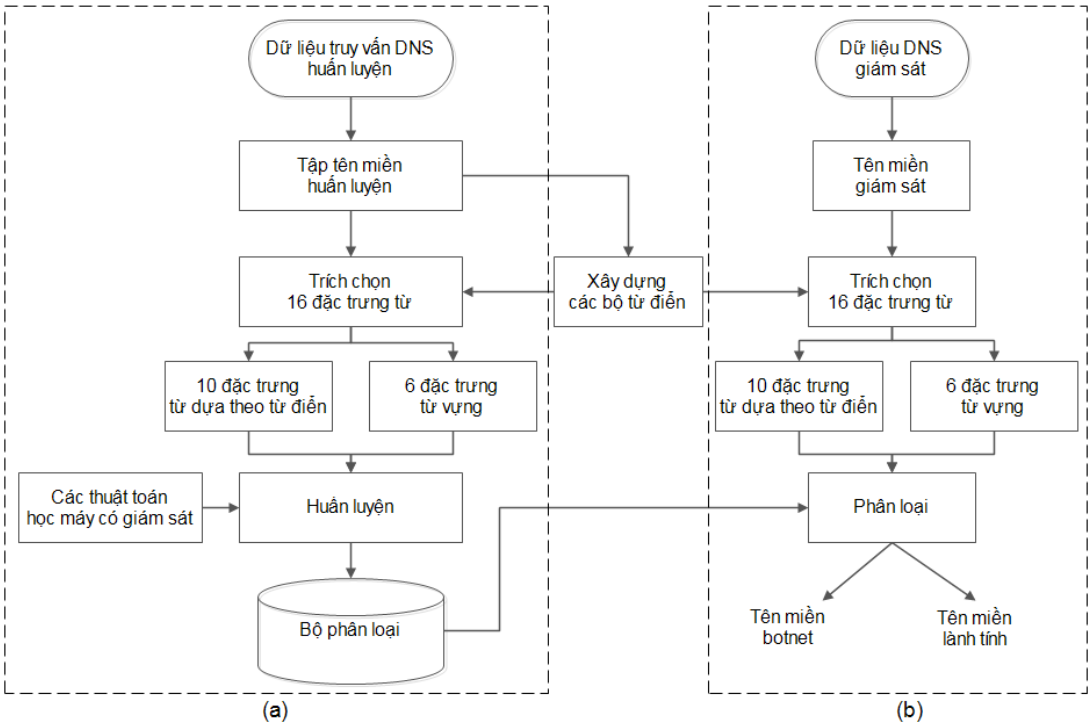
Có thể thấy, hầu hết các đề xuất phát hiện DGA botnet dựa trên các kỹ thuật học máy truyền thống chủ yếu tập trung phát hiện các họ character-based DGA botnet. Chúng không có khả năng phát hiện hoặc không hiệu quả trong phát hiện các họ word-based DGA botnet.

Với các hướng phát hiện DGA botnet dựa trên học sâu, ưu điểm chính của các nghiên cứu theo hướng này là có độ chính xác phát hiện DGA botnet và tính linh hoạt cao. Một số nghiên cứu đi sâu phân tích các đặc điểm ngữ nghĩa của tên miền, bao gồm phân phối theo từ, phân bố theo ký tự và mối tương quan đã cải thiện đáng kể kết quả phân loại các tên miền thuộc họ word-based DGA. Hạn chế chủ yếu của các bộ phân loại dựa trên học sâu là phần lớn không thực sự hiệu quả trong phân biệt các tên miền word-based DGA, mặc dù các đề xuất này khá thành công đối với các tên miền character-based DGA. Một số mô hình có khả năng hoạt động tốt trên các bộ dữ liệu thử nghiệm khác nhau, nhưng hiệu suất có thể bị ảnh hưởng khi cố gắng tổng quát hóa thành các họ DGA mới hoặc các phiên bản mới của các họ đã biết. Ngoài ra, các đề xuất dựa trên học sâu thường yêu cầu lớn hơn về tài nguyên tính toán, thời gian huấn luyện và phát hiện đều dài hơn so với các mô hình dựa trên học máy truyền thống. Các mục tiếp theo trình bày mô hình phát hiện WDM sử dụng tập 16 đặc trưng từ mới, cho phép phát hiện hiệu quả các họ word-based DGA botnet.

### 2.3.3. Giới thiệu mô hình WDM

Mô hình phát hiện word-based DGA botnet (*WDM*), như biểu diễn trên Hình 2.18 bao gồm hai giai đoạn: (a) giai đoạn huấn luyện và (b) giai đoạn phát hiện. Trong giai đoạn huấn luyện, mô hình được xây dựng từ dữ liệu huấn luyện. Trong giai đoạn

phát hiện, mô hình đã xây dựng được sử dụng để phân loại từng tên miền giám sát nếu nó là tên miền lành tính hoặc tên miền DGA botnet.




*Hình 2.18:Mô hình phát hiện word-based DGA botnet*

Trong giai đoạn huấn luyện được mô tả tại Hình 2.18 (a) gồm hai bước chính sau: (i) Trích chọn đặc trưng và (ii) Huấn luyện. Trong bước Trích chọn đặc trưng, 16 đặc trưng từ được trích xuất cho mỗi tên miền của tập dữ liệu huấn luyện. Mỗi tên miền được chuyển đổi thành một vector gồm 16 đặc trưng và một nhãn lớp. Nhãn lớp gồm 2 giá trị là 0 cho tên miền lành tính và 1 cho tên miền botnet. Kết quả của việc trích xuất các đặc trưng là một ma trận dữ liệu huấn luyện gồm M hàng và 17 cột, trong đó M là số lượng tên miền từ tập huấn luyện.

Trong bước Huấn luyện, dữ liệu huấn luyện được sử dụng để xây dựng mô hình phát hiện hoặc ‘Bộ phân loại’ sử dụng các kỹ thuật học máy có giám sát truyền thống. Một số thuật toán học máy có giám sát, bao gồm như Naïve Bayes, cây quyết định, rừng ngẫu nhiên, hồi quy logistic và SVM đã được sử dụng để xây dựng mô hình phát hiện. Các thuật toán học máy này được lựa chọn vì chúng có tốc độ thực thi nhanh và do đó phù hợp để xử lý lượng lớn dữ liệu ở chế độ trực tuyến. Ngoài ra, chúng đã

được sử dụng rộng rãi trong nhiều lĩnh vực và đạt hiệu quả tốt [13] [78]. Mô hình được đánh giá bằng phương pháp kiểm tra chéo 10 lần để tính các độ đo hiệu suất.

Trong giai đoạn phát hiện được mô tả tại Hình 2.18 (b) cũng gồm hai bước sau: (i) Trích xuất đặc trưng và (ii) Phân loại. Trong bước Trích xuất đặc trưng, mỗi tên miền giám sát được xử lý theo quy trình tương tự như được thực hiện trong giai đoạn huấn luyện ở bước này. Kết quả là mỗi tên miền được chuyển đổi thành một vector gồm 16 đặc trưng; Trong bước Phân loại, vector của mỗi tên miền được phân loại sử dụng 'Bộ phân loại' được xây dựng trong giai đoạn huấn luyện. Kết quả của bước này là nhãn dự đoán tên miền là botnet hay lành tính.

### 2.3.4. Tập dữ liệu thử nghiệm

Tập dữ liệu được sử dụng bao gồm ba tập con như sau: (i) tập con gồm 48,000 tên miền lành tính được trích xuất từ một triệu tên miền top Alexa [17]; (ii) Tập con gồm 64,000 tên miền word-based DGA botnet được tạo bằng tập lệnh DGA [34] cho 4 họ word-based DGA botnet điển hình, bao gồm bigviktor, matsnu, suppobox và pizd. Trong đó 48,000 tên miền của tập con này được sử dụng để huấn luyện và kiểm tra chéo các mô hình phát hiện và 16,000 tên miền được sử dụng để kiểm thử phát hiện; và (iii) Tập con gồm 63,905 tên miền DGA được tạo bởi 16 họ botnet DGA thu thập từ Netlab360 [11]. Trong đó 48,000 tên miền của tập con này được sử dụng để huấn luyện và kiểm tra chéo các mô hình phát hiện và 15,905 tên miền được sử dụng để kiểm thử phát hiện. Ngoài ra, sử dụng thêm tập dữ liệu gồm 31,000 tên miền DGA botnet thu thập từ UMUDGA để kiểm thử tỷ lệ phát hiện của mô hình WDM đối với các DGA botnet ở các tập dữ liệu khác nhau.

Từ 3 tập dữ liệu con trên, tạo 2 tập dữ liệu DATASET-01 và DATASET-02 cho các kịch bản thử nghiệm khác nhau. Tập DATASET-01 được sử dụng để đánh giá khả năng phát hiện các word-based DGA botnet của mô hình WDM. Tập DATASET-01 bao gồm (i) tập huấn luyện gồm 48,000 tên miền lành tính và 48,000 tên miền word-based DGA, và (ii) tập kiểm thử phát hiện gồm 16,000 tên miền word-based DGA. Bảng 2.15 hiển thị các thành phần chi tiết của DATASET-01.

*Bảng 2.15: Thành phần DATASET-01*

| Họ DGA botnet | Kiểu tên miền | Tập huấn luyện | Tập kiểm thử |
|---|---|---|---|
| Bigviktor | word-based | 12,000 | 4,000 |
| Matsnu | word-based | 12,000 | 4,000 |
| Suppobox | word-based | 12,000 | 4,000 |
| Pizd | word-based | 12,000 | 4,000 |
| Benign | lành tính | 48,000 | 16,000 |
| **Tổng cộng** | | **96,000** | **16,000** |

DATASET-02 được sử dụng để đánh giá khả năng phát hiện các loại DGA botnet, bao gồm cả word-based DGA botnet và character-based DGA botnet của mô hình WDM. Tập DATASET-02 bao gồm (i) tập huấn luyện gồm 48,000 tên miền lành tính và 48,000 tên miền DGA và (ii) tập kiểm thử phát hiện gồm 15,905 tên miền DGA. Các tên miền DGA được tạo bởi cả word-based DGA botnet và character-based DGA botnet. Bảng 2.16 trình bày các thành phần chi tiết của DATASET-02.

*Bảng 2.16: Thành phần DATASET-02*

| Họ DGA botnet | Kiểu tên miền | Tập huấn luyện | Tập kiểm thử |
|---|---|---|---|
| Bigviktor | word-based | 3,000 | 1,000 |
| Matsnu | word-based | 3,000 | 905 |
| Suppobox | word-based | 3,000 | 1,000 |
| Pizd | word-based | 3,000 | 1,000 |
| Flubot | character-based | 3,000 | 1,000 |
| Necurs | character-based | 3,000 | 1,000 |
| Ramnit | character-based | 3,000 | 1,000 |
| Ranbyus | character-based | 3,000 | 1,000 |
| Rovnix | character-based | 3,000 | 1,000 |
| Tinba | character-based | 3,000 | 1,000 |
| Cryptolocker | character-based | 3,000 | 1,000 |
| Dyre | character-based | 3,000 | 1,000 |
| Emotet | character-based | 3,000 | 1,000 |
| Murofet | character-based | 3,000 | 1,000 |
| Shiotob | character-based | 3,000 | 1,000 |
| Benign | lành tính | 48,000 | |
| **Tổng cộng** | | **96,000** | **15,905** |

### 2.3.5. Tiền xử lý dữ liệu

#### *2.3.5.1. Giới thiệu*

Dựa vào các đặc điểm của các họ word-based DGA botnet và mixed DGA botnet, 16 đặc trưng từ được trích xuất cho mỗi tên miền trong cả giai đoạn huấn luyện và phát hiện. Các đặc trưng này được đặt tên là f1, f2, f3, ..., f16. Trong số đó, các đặc trưng f1, f3, f4, f5, f6 được định nghĩa trong [24] [96]. Các đặc trưng còn lại được đề xuất mới trong mô hình WDM.

Đối với các tên miền lành tính có sử dụng các từ, theo thói quen và để tăng tính phổ cập của tên miền, các từ sẽ được lấy gần với tên, các lĩnh vực hoạt động của tổ chức, cá nhân sở hữu tên miền đó. Và đương nhiên, các từ được sử dụng sẽ nằm trong từ điển tiếng Anh thông dụng. Một bộ từ điển tiếng Anh chứa 58,000 từ [95] được sử dụng để làm cơ sở đối chiếu với các từ được sử dụng trong tên miền. Đây là một đặc trưng được trích chọn bởi vì có một số họ botnet sử dụng các danh sách từ riêng biệt không có trong từ điển tiếng Anh thông dụng. Từ điển này được đặt tên là '*english_dict*'.

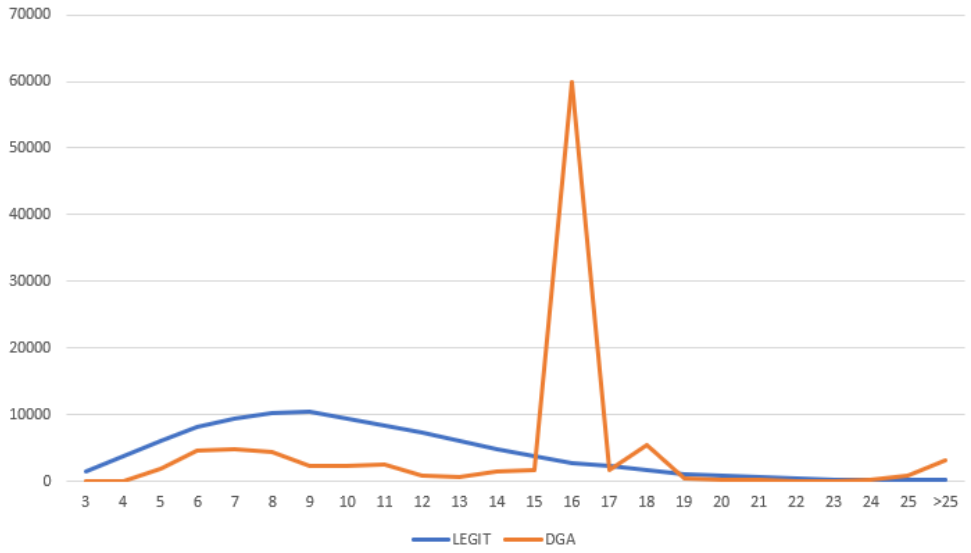
*Bảng 2.17: Thống kê các từ điển được sử dụng trong 4 word-based DGA botnet*

| Họ botnet | Tính từ | Danh từ | Động từ | Từ điển DGA | Từ điển riêng |
|---|---|---|---|---|---|
| Bigviktor | 252 | 1508 | 100 | | |
| Mastnu | | 879 | 1007 | | |
| Suppobox | | | | 1152 | |
| Pizd | | | | 384 | |
| Tổng hợp | 586 | 1527 | 1123 | 2636 | 433 |

Danh sách các danh từ, động từ và tính từ tiếng Anh thông dụng được xây dựng bằng cách sử dụng các từ thường dùng được liệt kê trong [91] được thống kê tại Bảng 2.17. Lý do tạo danh sách các từ này là một số botnet, chẳng hạn như bigviktor và matsnu sử dụng danh sách danh từ, động từ và tính từ chung để tạo tên miền cho máy chủ CnC của chúng. Danh sách danh từ, động từ và tính từ được đặt tên lần lượt là '*noun_dict*', '*verb_dict*' và '*adj_dict*', tương ứng với số lượng trong mỗi danh sách là 586, 1527 và 1123 từ được tổng hợp từ 4 họ word-based DGA botnet.

Một từ điển word-based DGA botnet có tên là '*dga_dict*' bao gồm 2,636 từ được xây dựng. Từ điển này bao gồm các từ có trong '*english_dict*' và thường được sử dụng để sinh các tên miền. Một từ điển word-based DGA botnet bổ sung được gọi là '*private_dict*' gồm 433 được xây dựng, từ điển bao gồm các từ được sử dụng để sinh tên miền được thống kê, nhưng không tồn tại trong '*english_dict*'.

### *2.3.5.2. Trích chọn các đặc trưng*




*Hình 2.19: So sánh độ dài của tên miền lành tính và DGA*

Hình 2.19 là biểu đồ so sánh độ dài của tên miền lành tính và tên miền DGA. Trên số lượng thống kê 100,000 tên miền của mỗi loại nhận thấy: trong vùng từ 3-14 ký tự, số lượng tên miền lành tính cao hơn đáng kể so với số lượng tên miền DGA. Tuy nhiên, trong vùng độ dài từ 15-17 ký tự, số lượng tên miền DGA là 59,878, cao hơn rất nhiều so với số lượng tên miền lành tính. Trong vùng độ dài từ 19-24 ký tự, số lượng 2 loại tên miền là tương đương. Mặc dù vậy, trong vùng 25 ký tự trở lên, số lượng tên miền DGA là 3,126 cao hơn đáng kể so với số lượng tên miền lành tính chỉ là 232. Từ đó có thể nhận thấy, đặc trưng độ dài tên miền có thể sử dụng trong tập đặc trưng phân loại tên miền.

- *f1*: độ dài của tên miền *d* tính bằng ký tự, biểu diễn là *len(d)* [96];

Trong bất kỳ một ngôn ngữ nào, việc lựa chọn tên miền đều hướng tới mục đích để người sử dụng dễ nhớ và các tên miền này gần với ngôn ngữ tự nhiên và nói lên

lĩnh vực hoạt động của tên miền đó. Do đó, tần suất các ký tự thường gặp sẽ nhiều hơn so với các ký tự khác. Mặt khác, các tên miền word-based DGA thường tạo sử dụng các từ dùng chung trong một từ điển có sẵn. Từ nhận định trên, đặc trưng tổng giá trị tính theo giá trị của từng ký tự (*mã ASCII*) được sử dụng trong tập đặc trưng phân loại tên miền.

- *f2*: tổng giá trị của mã ASCII của tất cả các ký tự trong tên miền d, được tính theo công thức sau:

$$ascii\_value(d) = \sum_{i=1}^{len(d)} ord(d[i]) \quad (2.15)$$

Các đặc trưng f3-f6 kế thừa từ mô hình CDM được trình bày tại mục 2.2.4.

- *f3*: số nguyên âm của tên miền d, được ký hiệu là *countnv(d)* [24];

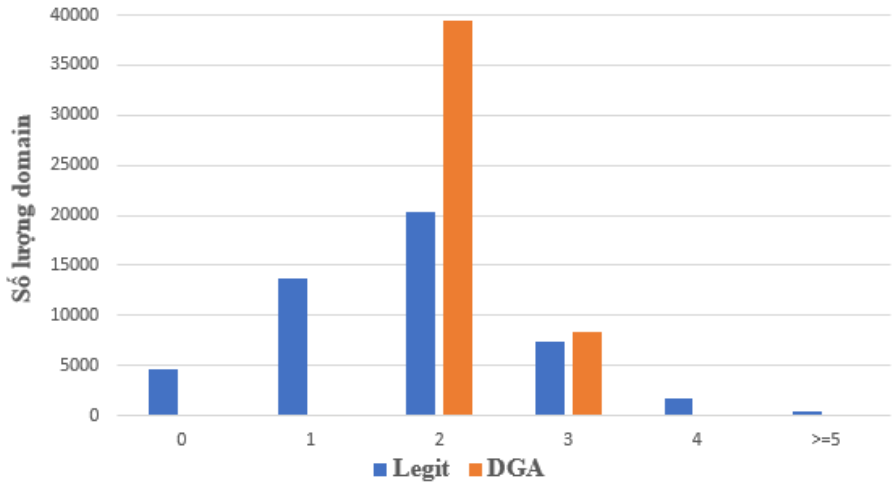
- *f4*: sự phân bố nguyên âm tên miền *d*, được tính theo công thức sau [24]:

$$tanv(d) = \frac{countnv(d)}{len(d)} \quad (2.16)$$

- f5: số ký tự số và ký tự '-' của tên miền *d*, ký hiệu là *countdi (d)*;

- *f6*: phân phối chữ số và ký tự '-' tên miền *d*, được tính theo công thức sau [90]:

$$tandi(d) = \frac{countdi(d)}{len(d)} \quad (2.17)$$



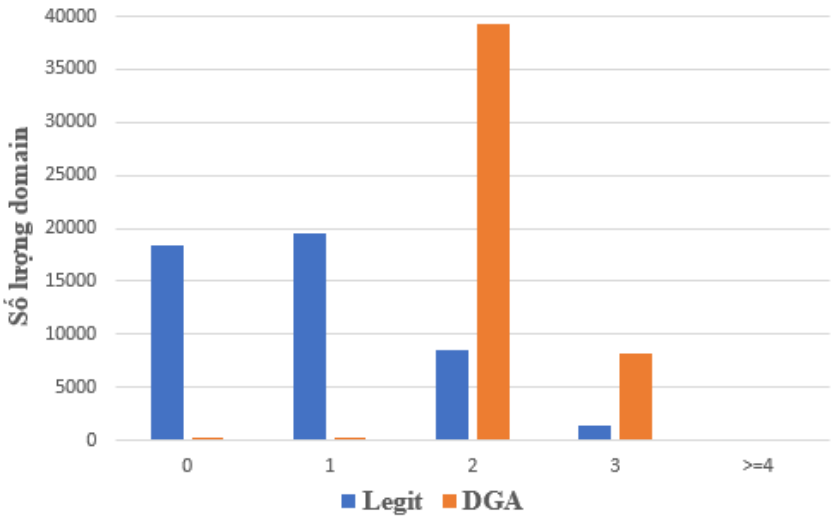

*Hình 2.20: Thống kê số lượng từ trong tên miền*

Hình 2.20 thống kê số lượng từ được trích xuất từ 48,000 tên miền lành tính và 48,000 của 4 họ tên miền word-based DGA botnet. Đối với các tên miền word-based

DGA botnet, số từ được sử dụng là 2 từ chiếm tỷ lệ 82.20% , 3 từ chiếm tỷ lệ 17.46%. Đối với tên miền lành tính thì số lượng tên miền không chứa từ hoặc có chứa 1 từ chiếm tỷ lệ chiếm 37.91%. Ở đây, sự phân bổ số từ trên mỗi tên miền giữa tên miền lành tính và tên miền word-based DGA botnet có sự chênh lệch khá rõ ràng.

- f7: số từ được trích xuất từ tên miền *d* và tồn tại trong từ điển '*english_dict*'. Đặc trưng này được ký hiệu là *word_norm(d)*;

Bốn họ tên miền word-based DGA botnet được liệt kê tại Bảng 2.17 sử dụng 2636 từ các loại. Để phát hiện các tên miền có sử dụng những từ này, xây dựng một từ điển tổng hợp các từ được lấy ngẫu nhiên làm đặc trưng so với các từ trong từ điển tiếng Anh thông dụng. Hình 2.21 thống kê số lượng tên miền có sử dụng từ trong từ điển DGA. Đối với tên miền word-based DGA botnet, số từ trong từ điển DGA được sử dụng trong mỗi tên miền là từ 2 đến 3 từ chiếm tỷ lệ 98.90%. Trong khi đó, số từ trong trong từ điển DGA không tồn tại ở tên miền lành tính hoặc chỉ tồn tại 1 từ chiếm tỷ lệ tương ứng là 38.39% và 40.65%. Sự chênh lệch này là lý do đề xuất đặc trưng f8 trong mô hình WDM.




*Hình 2.21: Thống kê số từ trong từ điển DGA của tên miền*

- *f8*: số lượng từ được trích xuất từ tên miền *d* và tồn tại trong từ điển '*dga_dict*'. Đặc trưng này được ký hiệu là *word_dga(d)*;

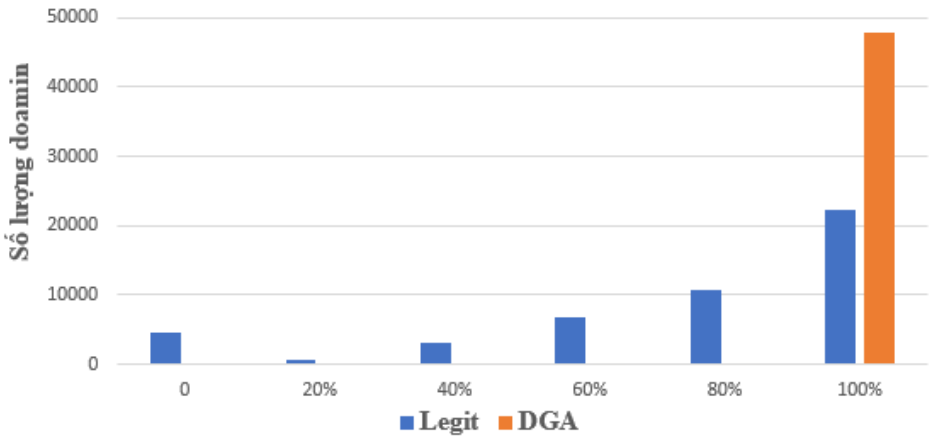
Hai họ tên miền ‘bigviktor’ và ‘mastnu’ sử dụng các từ điển định sẵn phân chia thành các từ điển danh từ, động từ và tính từ. Đối với các tên miền lành tính, việc sử dụng kết hợp này thường ít xảy ra. Do đó, luận án xây dựng các bộ từ điển danh từ, động từ, tính từ để trích chọn các đặc trưng f9-f11 trong tập đặc trưng phân loại tên miền.

- *f9*: số từ được trích ra từ tên miền *d* và tồn tại trong từ điển ‘*noun_dict*’. Đặc trưng này được ký hiệu là *noun_count(d)*;
- *f10*: số lượng từ được trích xuất từ tên miền *d* và tồn tại trong từ điển ‘*verb_dict*’. Đặc trưng này được ký hiệu là *verb_count(d)*;
- *f11*: số từ được trích xuất từ tên miền *d* và tồn tại trong từ điển ‘*adj_dict*’. Đặc trưng này được ký hiệu là *adj_count(d)*;
- *f12*: số từ được trích xuất từ tên miền *d* và tồn tại trong từ điển ‘*private_dict*’. Đặc trưng này được ký hiệu là *private_count(d)*;
- *f13*: tỷ lệ giữa *word_dga(d)* và *word_norm(d)*, và được tính theo công thức sau:

$$ratio\_dga(d) = \frac{word\_dga(d)}{word\_norm(d)} \qquad (2.18)$$

Nguyên tắc chung khi lựa chọn tên miền để dễ nhớ và dễ sử dụng, do đó khi lựa chọn các từ để đưa vào tên miền thường sử dụng các từ có độ dài ngắn hoặc viết tắt các ký tự đầu của từ. Đối với các tên miền word-based DGA thì việc lựa chọn này được lấy ngẫu nhiên từ những bộ từ điển định sẵn. Do đó, các đặc trưng f14-f16 trong tập đặc trưng phân loại tên miền.

- *f14*: độ dài của từ dài nhất của tên miền *d*. Đặc trưng này được ký hiệu là *max_len_word*(d);
- *f15*: độ dài của từ ngắn nhất của tên miền d. Đặc trưng này được ký hiệu là *min_len_word*(d);




*Hình 2.22: Tỷ lệ ký tự sử dụng trong từ của mỗi tên miền*

Hình 2.22 thể hiện tỷ lệ số ký tự được sử dụng trong các từ so với độ dài của mỗi tên miền. Đối với các tên miền word-based DGA botnet, tỷ lệ này thường đạt 100% với 47,772 tên miền, tức là trong tên miền là sự kết hợp của các từ. Đối với tên miền lành tính có 25,838 tên miền tỷ lệ từ 0-80% tức là trong tên miền bao gồm các từ và các ký tự khác. Đây cũng là điểm khác biệt giữa tên miền lành tính và tên miền word-based DGA botnet.

- *f16*: tỷ lệ giữa số ký tự của các từ của tên miền *d* và độ dài của tên miền *d*, được tính theo công thức sau:

$$ratio\_char(d) = \frac{len(words(d))}{len(d)} \quad (2.19)$$

### 2.3.6. Thử nghiệm và kết quả

#### *2.3.6.1. Kịch bản thử nghiệm*

Các thử nghiệm được thực hiện theo các kịch bản sau:

Kịch bản 1: Huấn luyện và kiểm tra chéo mô hình phát hiện sử dụng "Tập huấn luyện" thuộc tập DATASET-01. Các thuật toán học máy có giám sát, bao gồm Naïve Bayes (*NB*), cây quyết định, rừng ngẫu nhiên, hồi quy Logistic và SVM được sử dụng theo trình tự để xây dựng các mô hình phát hiện, trong đó 80% dữ liệu được sử dụng để huấn luyện mô hình và 20% dữ liệu được sử dụng để kiểm tra chéo. Thuật toán rừng ngẫu nhiên sử dụng với 35 cây (*RF-35*).

Kịch bản 2: Kiểm thử mô hình phát hiện được xây dựng trong kịch bản 1 bằng cách sử dụng "Tập kiểm thử" của tập DATASET-01. Mục đích của kịch bản này là tìm tỷ lệ phát hiện (*DR*) của mô hình trên một số word-based DGAbotnet.

Kịch bản 3: Huấn luyện và kiểm tra chéo mô hình phát hiện bằng cách sử dụng 'Tập huấn luyện' của DATASET-02. Các thuật toán NB, J48 tree, RF-35, hồi quy logistic và SVM được sử dụng theo trình tự để xây dựng các mô hình phát hiện, trong đó 80% dữ liệu được sử dụng để huấn luyện xây dựng mô hình và 20% dữ liệu được sử dụng để kiểm tra chéo.

Kịch bản 4: Kiểm thử các mô hình phát hiện được xây dựng trong kịch bản 3 bằng cách sử dụng 'Tập thử nghiệm' của DATASET-02. Mục đích của kịch bản này là để tìm tỷ lệ phát hiện (*DR*) của mô hình trên các DGA botnet điển hình, bao gồm cả word-based DGA botnet và character-based DGA botnet.

#### *2.3.6.2. Kết quả*

*Bảng 2.18 trình bày hiệu suất phát hiện của mô hình đề xuất dựa trên 5 thuật toán học máy sử dụng "Tập huấn luyện" của DATASET-01. Các độ đo hiệu suất trên bảng này xác nhận rằng mô hình đề xuất hoạt động rất tốt trên DATASET-01 với tất cả 5 thuật toán học máy. Mô hình được xây dựng từ 'Tập huấn luyện' của DATADET-01 cũng cho tỷ lệ phát hiện cao đối với tất cả 4 word-based DGAbotnet, như được hiển thị trong*

Bảng 2.19.

*Bảng 2.18: Hiệu suất phát hiện của mô hình sử dụng DATASET-01 (%)*

| **Thuật toán** | **PPV** | **TPR** | **FPR** | **FNR** | **ACC** | **F1** |
|---|---|---|---|---|---|---|
| NB | 98.47 | 91.16 | 1.64 | 8.84 | 94.48 | 94.67 |
| *J48* | *98.25* | *95.81* | *1.78* | *4.19* | *96.99* | *97.01* |
| RF-35 | 97.27 | 95.95 | 2.74 | 4.05 | 96.60 | 96.61 |
| Logistic | 98.63 | 92.97 | 1.45 | 7.03 | 95.60 | 95.71 |
| SVM | 98.70 | 93.73 | 1.36 | 6.27 | 96.07 | 96.15 |

*Bảng 2.19: Tỷ lệ phát hiện (DR) của mô hình sử dụng DATASET-01 (%)*

| Thuật toán<br>Bonet | NB | J48 | RF-35 | Logistic | SVM |
|---|---|---|---|---|---|
| Bigviktor | 96.35 | 96.78 | 95.28 | 96.88 | 97.08 |
| Matsnu | 99.13 | 97.78 | 97.55 | 99.10 | 99.03 |
| Pizd | 98.98 | 98.63 | 97.50 | 98.98 | 98.98 |
| Suppobox | 99.48 | 99.30 | 96.93 | 99.48 | 99.48 |
| **Trung bình** | **98.51** | **98.19** | **96.81** | **98.63** | **98.66** |

Bảng 2.20 thể hiện hiệu suất phát hiện của mô hình đề xuất dựa trên 5 thuật toán học máy sử dụng 'Tập huấn luyện' của DATASET-02. Các độ đo hiệu suất trên bảng này cũng xác nhận rằng mô hình đề xuất hoạt động tương đối tốt trên DATASET-02 với tất cả 5 thuật toán học máy. Mô hình được xây dựng từ 'Tập huấn luyện' của DATASET-02 cũng tạo ra tỷ lệ phát hiện khá tốt đối với hầu hết các DGA botnet, như trình bày trong Bảng 2.21.

*Bảng 2.20: Hiệu suất phát hiện của mô hình sử dụng DATASET-02 (%)*

| Thuật toán | PPV | TPR | FPR | FNR | ACC | F1 |
|---|---|---|---|---|---|---|
| NB | 65.30 | 89.13 | 27.18 | 10.78 | 78.77 | 75.38 |
| J48 | 96.89 | 94.62 | 3.15 | 5.38 | 95.71 | 95.75 |
| RF-35 | 96.02 | 94.78 | 3.99 | 5.22 | 95.39 | 95.40 |
| Logistic | 88.34 | 90.47 | 11.29 | 9.53 | 89.57 | 89.40 |
| SVM | 88.79 | 90.15 | 10.94 | 9.85 | 89.59 | 89.47 |

*Bảng 2.21: Tỷ lệ phát hiện (DR) của mô hình (%) sử dụng DATASET-02*

| Thuật toán<br>Bonet | NB | J48 | RF-35 | Logistic | SVM |
|---|---|---|---|---|---|
| Bigviktor | 77.80 | 70.70 | 67.70 | 88.60 | 90.00 |
| Matsnu | 60.33 | 98.34 | 94.59 | 78.01 | 81.99 |
| Pizd | 8.40 | 97.90 | 99.40 | 73.60 | 75.70 |
| Suppobox | 7.30 | 99.10 | 97.70 | 94.10 | 97.30 |
| Flubot | 73.90 | 99.20 | 99.10 | 96.00 | 96.10 |
| Necurs | 53.40 | 91.70 | 90.20 | 83.10 | 83.10 |
| Ramnit | 51.30 | 92.10 | 91.20 | 84.50 | 84.50 |
| Ranbyus | 72.80 | 98.00 | 97.20 | 94.60 | 94.90 |

| | | | | | |
|---|---|---|---|---|---|
| Rovnix | 100.00 | 99.30 | 99.60 | 99.30 | 99.40 |
| Tinba | 27.40 | 98.90 | 97.60 | 61.50 | 91.40 |
| Cryptolocker | 48.50 | 96.70 | 95.80 | 91.80 | 92.20 |
| Dyre | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 |
| Emotet | 96.50 | 99.40 | 99.10 | 97.40 | 97.70 |
| Gameover | 100.00 | 99.80 | 99.80 | 99.90 | 99.90 |
| Murofet | 84.00 | 99.50 | 99.70 | 99.00 | 99.00 |
| Shiotob | 74.60 | 95.20 | 94.80 | 84.00 | 85.00 |
| **Trung bình** | **64.82** | **95.98** | **95.32** | **91.14** | **91.92** |

### 2.3.7. Đánh giá

*Từ kết quả thực nghiệm cho trong Bảng 2.18,*

*Bảng 2.19, Bảng 2.20, Bảng 2.21, có thể rút ra các nhận xét sau: Mô hình phát hiện WDM mang lại hiệu suất cao trên DATASET-01 với độ chính xác phát hiện tổng thể (ACC) và độ đo F1 trên 95% sử dụng 5 thuật toán học máy. Trong đó, thuật toán cây quyết định J48 hoạt động tốt nhất với tỷ lệ phát hiện cao nhất và tỷ lệ cảnh báo sai thấp nhất, như được hiển thị trong Bảng 2.18. Tỷ lệ phát hiện của 4 word-based DGA botnet điển hình biểu diễn trong*

Bảng 2.19 cũng xác nhận rằng mô hình WDM có khả năng phát hiện hiệu quả các word-based DGA botnet. Điều này có nghĩa là 16 đặc trưng từ sử dụng là phù hợp cho việc phân loại tên miền word-based DGA và tên miền lành tính.

Mô hình phát hiện WDM cũng tạo ra hiệu suất khá tốt trên DATASET-02 với độ chính xác phát hiện tổng thể (ACC) và độ đo F1 trên 95% sử dụng thuật toán cây quyết định và rừng ngẫu nhiên. Trong khi các mô hình dựa trên hồi quy logistic và SVM đạt được độ chính xác phát hiện tổng thể (ACC) và độ đo F1 trên 89%, thì mô hình dựa trên Naïve Bayes chỉ đạt độ đo F1 khoảng 75%, như trình bày trong Bảng 2.20. Tỷ lệ phát hiện 4 word-based DGA botnet và 12 character-based DGA botnet

được hiển thị trong Bảng 2.21 xác nhận rằng mô hình dựa trên cây quyết định J48 hoạt động tốt trên hầu hết các botnet thử nghiệm, ngoại trừ ‘Bigviktor’. Mặc dù mô hình dựa trên SVM có tỷ lệ phát hiện trên ‘Bigviktor’ cao hơn so với mô hình dựa trên J48, tuy nhiên mô hình dựa trên J48 có tỷ lệ phát hiện trên hầu hết các botnet tốt hơn đáng kể so với mô hình dựa trên SVM.

*Bảng 2.22: Hiệu suất phát hiện của WDM so với các đề xuất khác (%)*

| Mô hình phát hiện | PPV | TPR | FPR | FNR | ACC | F1 |
|---|---|---|---|---|---|---|
| Truong và cộng sự [96] | 94.70 | | 4.80 | | 92.30 | |
| Hoang và cộng sự [24] | 90.70 | 91.00 | 9.30 | | 90.90 | 90.90 |
| Qiao và cộng sự [70] | 95.05 | 95.14 | | | | 94.58 |
| Zhao và cộng sự [26] | | | 6.14 | 7.42 | 94.04 | |
| Mô hình CDM | 99.57 | 99.62 | 0.43 | 0.38 | 99.60 | 99.60 |
| Mô hình WDM (J48) DATASET-01 | 98.25 | 95.81 | 1.78 | 4.19 | 96.99 | 97.01 |
| Mô hình WDM (J48) DATASET-02 | 96.89 | 94.62 | 3.15 | 5.38 | 95.71 | 95.75 |

*Bảng 2.23: So sánh tỷ lệ phát hiện của 2 mô hình WDM và CDM*

| Bonet | WDM | CDM |
|---|---|---|
| Bigviktor | 70.70 | 3.00 |
| Matsnu | 98.34 | 1.14 |
| Pizd | 97.90 | |
| Suppobox | 99.10 | 0.95 |
| Flubot | 99.20 | |
| Necurs | 91.70 | 98.67 |
| Ramnit | 92.10 | 97.20 |
| Ranbyus | 98.00 | 99.82 |
| Rovnix | 99.30 | 100.00 |
| Tinba | 98.90 | 98.77 |
| Cryptolocker | 96.70 | 99.00 |
| Dyre | 100.00 | 98.00 |
| Emotet | 99.40 | 99.85 |
| Gameover | 99.80 | 100.00 |

| | | |
|---|---|---|
| Murofet | 99.50 | 99.85 |
| Shiotob | 95.20 | 99.55 |

Bảng 2.22 hiển thị so sánh hiệu suất phát hiện giữa mô hình WDM và các đề xuất phát hiện DGA botnet khác và Bảng 2.23 so sánh tỷ lệ phát hiện của 16 word-based DGA botnet và character-based DGA botnet giữa mô hình WDM dựa trên cây quyết định J48 và mô hình phát hiện CDM. Từ kết quả trình bày trong hai bảng này, có thể rút ra các nhận xét sau: (i) Mô hình WDM hoạt động tốt hơn nhiều so với các đề xuất phát hiện DGA botnet khác, bao gồm Truong và cộng sự [96], Hoang và cộng sự [24], Qiao và cộng sự [70], Zhao và cộng sự [26]; (ii) Mô hình WDM có khả năng phát hiện hiệu quả các character-based DGA botnet, mặc dù có tỷ lệ phát hiện thấp hơn so với mô hình CDM; (iii) Mặc dù mô hình CDM đạt hiệu suất tốt hơn so với mô hình WDM trên các character-based DGA botnet, nhưng mô hình CDM lại hầu như không thể phát hiện các word-based DGA botnet. Trong khi đó, mô hình WDM có khả năng phát hiện hiệu quả các word-based DGA botnet, gồm *matsnu*, *pizd* và *suppobox*.

## 2.4. KẾT LUẬN CHƯƠNG

Chương này giới thiệu chi tiết về DGA botnet, các loại DGA botnet và cơ chế các DGA botnet khai thác hệ thống DNS để duy trì hoạt động. Nhờ khả năng sinh và gán tên miền, địa chỉ IP tự động cho máy chủ CnC, đồng thời với khả năng tự động sinh và truy vấn các tên miền bởi các bot, các DGA botnet có khả năng lẩn tránh rà quét và kéo dài thời gian tồn tại của mình. Mặc dù vậy, do các bot trong botnet thường xuyên tương tác với hệ thống DNS, việc giám sát và phân tích các truy vấn DNS có thể giúp phát hiện sự tồn tại của các bot và botnet.

Phần tiếp theo của chương 2 khảo sát các phương pháp phát hiện DGA botnet theo 3 nhóm: phát hiện dựa trên phân tích truy vấn DNS, phát hiện dựa trên thống kê và phát hiện dựa trên học máy. Mặc dù có nhiều ưu điểm, nhưng hạn chế lớn nhất của các phương pháp đã có là tập đặc trưng phân loại tên miền được lựa chọn chưa thực sự phù hợp dẫn đến tỷ lệ phát hiện sai còn tương đối cao (đến 10%). Để khắc phục vấn đề này, chương 2 phát triển mô hình CDM cho phát hiện DGA botnet với

mục tiêu là cải thiện tỷ lệ phát hiện đúng và giảm cảnh báo sai. Mô hình CDM đề xuất sử dụng 24 đặc trưng ký tự cho phân loại tên miền DGA với tên miền lành tính, trong đó kế thừa 18 đặc trưng từ [24] và bổ sung 6 đặc trưng mới. Thuật toán học máy Rừng ngẫu nhiên được lựa chọn để xây dựng mô hình phát hiện. Các thử nghiệm trên tập dữ liệu gồm 100.000 tên miền lành tính và 153.000 tên miền DGA cho thấy, mô hình CDM đạt các độ đo đánh giá vượt trội so với các mô hình đã có. Cụ thể, mô hình CDM đạt độ chính xác chung trên 99.60% và tỷ lệ cảnh báo sai khoảng 0.4%. Mô hình phát hiện CDM và các kết quả thử nghiệm đã được công bố trong bài báo "An Improved Model for Detecting DGA botnets Using Random Forest Machine Learning Algorithm", đăng trên tạp chí Information Security Journal, 2021, ESCI Scopus Q2 [CT1].

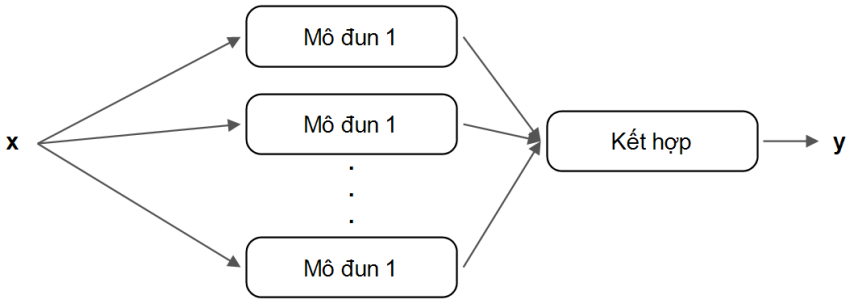
Phần cuối của chương đề xuất mô hình WDM cho phép phát hiện hiệu quả các word-based DGA botnet. Mô hình WDM cải thiện hiệu suất phát hiện các word-based DGA botnet bằng cách sử dụng một bộ 16 đặc trưng mới dựa trên từ để phân biệt các tên miền word-based DGA và tên miền lành tính. Kết quả thử nghiệm xác nhận rằng mô hình WDM đạt độ đo F1 là 97,01% đối với bộ dữ liệu word-based DGA botnet (*DATASET-01*). Hơn nữa, mô hình dựa trên cây quyết định J48 cũng hoạt động tương đối tốt trên tập dữ liệu kết hợp (*DATASET-02*) gồm các tên miền word-based DGA botnet và character-based DGA botnet với độ đo F1 là 95,75%. Mô hình WDM và các kết quả thử nghiệm đã được công bố trong bài báo "An Novel Machine Learning-based Approach for Detecting Word-based Botnets", đăng trên tạp chí Journal of Theoretical and Applied Information Technology, 2021, Scopus Q4 [CT2].

# CHƯƠNG 3: PHÁT HIỆN DGA BOTNET DỰA TRÊN HỌC KẾT HỢP

## 3.1. KHÁI QUÁT VỀ HỌC KẾT HỢP

### 3.1.1. Giới thiệu

Phương pháp học kết hợp (*ensemble learrning*) thực hiện huấn luyện đồng thời nhiều mô hình hoặc mô đun thành phần để giải quyết một vấn đề. Khác với các phương pháp học thông thường cố gắng xây dựng một mô đun từ dữ liệu huấn luyện, các phương pháp học kết hợp cố gắng xây dựng một tập hợp các mô đun và kết hợp chúng. Học kết hợp còn được gọi là học dựa trên hội đồng (*committee-based*), hoặc học nhiều hệ thống phân loại [106].




*Hình 3.1: Kiến trúc tổng thể chung của học kết hợp*

Hình 3.1 cho thấy một mô hình kiến trúc tổng thể của học kết hợp. Một nhóm chứa một số mô đun được gọi là mô đun cơ sở. Mô đun cơ sở thường được tạo ra từ dữ liệu huấn luyện bởi một thuật toán học cơ sở có thể là cây quyết định, mạng nơ-

ron hoặc các loại thuật toán học máy khác. Hầu hết các phương pháp học kết hợp sử dụng một thuật toán học cơ sở duy nhất để tạo ra những mô đun cơ sở đồng nhất, tức là những mô đun cùng loại, dẫn đến những tập thể đồng nhất. Tuy vậy, cũng có một số phương pháp sử dụng nhiều thuật toán học cơ sở để tạo ra những mô đun không đồng nhất, tức là những mô đun thuộc nhiều loại khác nhau, dẫn đến các quần thể không đồng nhất. Trong trường hợp này, không có thuật toán học cơ sở duy nhất và do đó, một số người gọi là mô đun cá nhân hoặc mô đun thành phần thay cho mô đun cơ sở.

Khả năng khái quát hóa của một nhóm thường mạnh hơn nhiều so với khả năng của những mô đun cơ sở. Trên thực tế, các phương pháp học kết hợp hấp dẫn chủ yếu vì chúng có thể thúc đẩy những mô đun yếu thậm chí chỉ tốt hơn một chút so với phỏng đoán ngẫu nhiên cho những mô đun mạnh có thể đưa ra dự đoán rất chính xác. Vì vậy, mô đun cơ sở còn được gọi là mô đun yếu.

Có ba chủ đề của những đóng góp ban đầu dẫn đến các phương pháp học kết hợp hiện tại; đó là, (i) kết hợp các bộ phân loại, (ii) tập hợp những mô đun yếu và (iii) hỗn hợp chuyên gia. *Kết hợp các bộ phân loại* chủ yếu được nghiên cứu trong cộng đồng nhận dạng mẫu. Trong chủ đề này, các nhà nghiên cứu thường làm việc trên các bộ phân loại mạnh và cố gắng thiết kế các quy tắc kết hợp mạnh mẽ để có được các bộ phân loại kết hợp mạnh mẽ hơn. Do đó, chuỗi công việc này đã tích lũy được sự hiểu biết sâu sắc về thiết kế và sử dụng các quy tắc kết hợp khác nhau. *Tập hợp mô đun yếu* hầu hết được nghiên cứu trong cộng đồng học máy. Trong chủ đề này, các nhà nghiên cứu thường làm việc trên những mô đun yếu và cố gắng thiết kế các thuật toán mạnh mẽ để tăng hiệu suất từ yếu đến mạnh. Chuỗi công việc này đã dẫn đến sự ra đời của các phương pháp học kết hợp nổi tiếng như AdaBoost, Bagging, v..v. và sự hiểu biết lý thuyết về lý do tại sao và làm thế nào những mô đun yếu có thể được tăng cường thành những mô đun mạnh. Trong chủ đề này, các nhà nghiên cứu thường xem xét chiến lược chia để trị, cố gắng tìm hiểu hỗn hợp các mô hình tham số cùng nhau và sử dụng các quy tắc kết hợp để có được giải pháp tổng thể.

Phần tiếp theo của mục này giới thiệu về một số kỹ thuật học kết hợp đơn giản, bao gồm max voting, averaging và weighted averaging, và một số kỹ thuật học kết hợp nâng cao, bao gồm Bagging, Stacking và Boosting.

### 3.1.2. Kỹ thuật học kết hợp đơn giản

Phần này xem xét một số kỹ thuật học kết hợp đơn giản nhưng mạnh mẽ, bao gồm: số phiếu tối đa (*max voting*), tính trung bình (*averaging*), tính trung bình có trọng số (*weighted averaging*).

Max Voting là phương pháp biểu quyết tối đa thường được sử dụng cho các vấn đề phân loại. Trong kỹ thuật này, nhiều mô hình được sử dụng để đưa ra dự đoán cho mỗi điểm dữ liệu. Các dự đoán của mỗi mô hình được coi như một "phiếu bầu". Các dự đoán nhận được từ phần lớn các mô hình được sử dụng làm dự đoán cuối cùng.

Averaging tương tự như kỹ thuật bỏ phiếu tối đa, trong đó nhiều dự đoán được thực hiện cho mỗi điểm dữ liệu trong tính trung bình. Trong phương pháp này, lấy trung bình các dự đoán từ tất cả các mô hình và sử dụng nó để đưa ra dự đoán cuối cùng. Tính trung bình có thể được sử dụng để đưa ra dự đoán trong các bài toán hồi quy hoặc trong khi tính toán xác suất cho các bài toán phân loại.

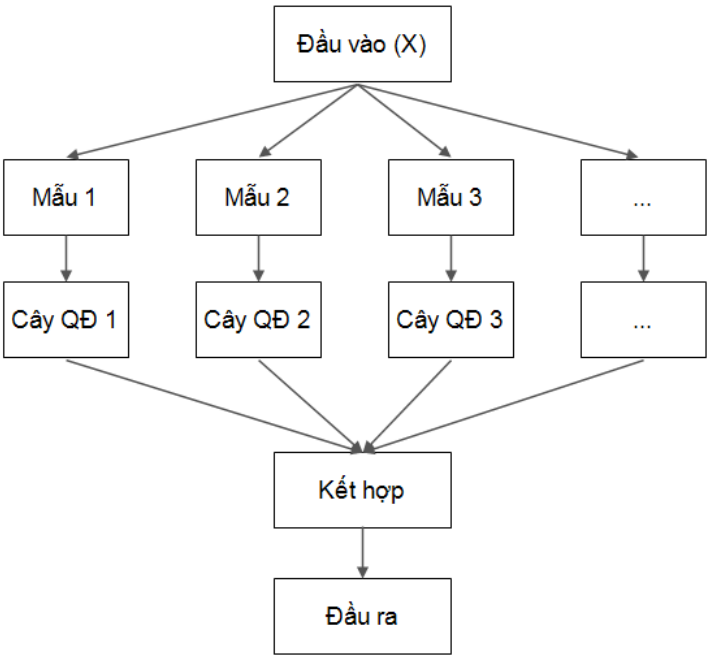
Weighted Averaging là một phần mở rộng của phương pháp tính trung bình. Tất cả các mô hình được ấn định các trọng số khác nhau xác định tầm quan trọng của từng mô hình để dự đoán. Ví dụ, nếu có hai mô hình có tầm ảnh hưởng lớn hơn, trong khi những mô hình khác không có nhiều ảnh hưởng trong dự đoán, thì câu trả lời của hai mô hình có tầm quan trọng cao hơn so với những mô hình còn lại.

### 3.1.3. Kỹ thuật học kết hợp nâng cao

Mục này đề cập 3 phương pháp học kết hợp nâng cao, bao gồm bagging, stacking và boosting.

Bagging: Tập hợp bootstrap, hay gọi tắt là đóng gói, là một phương pháp học kết hợp nhằm tìm kiếm một nhóm đa dạng các thành viên trong nhóm bằng cách thay đổi dữ liệu huấn luyện. Điều này thường liên quan đến việc sử dụng một thuật toán

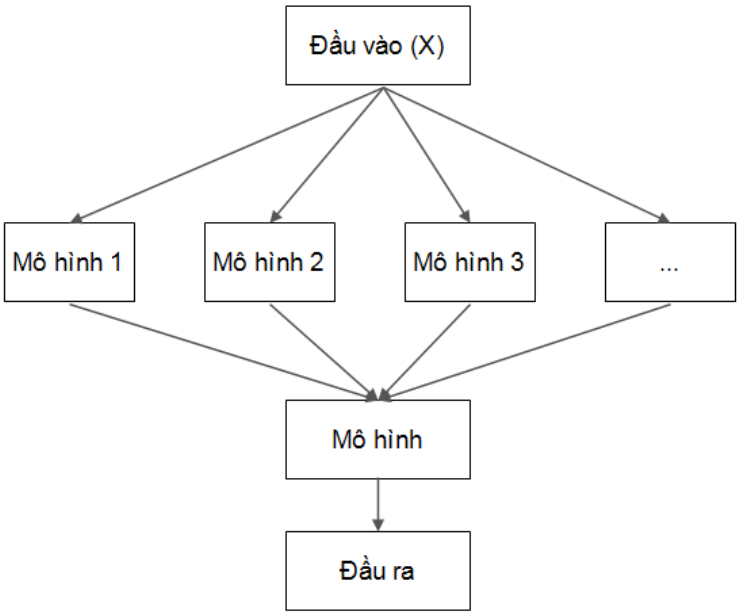
học máy duy nhất, hầu như luôn luôn là một cây quyết định chưa được điều chỉnh và huấn luyện mỗi mô hình trên một mẫu khác nhau của cùng một tập dữ liệu huấn luyện. Các dự đoán do các thành viên trong nhóm đưa ra sau đó được kết hợp bằng cách sử dụng các thống kê đơn giản, chẳng hạn như bỏ phiếu hoặc lấy trung bình. Đây là một kỹ thuật thường được sử dụng trong thống kê với các tập dữ liệu nhỏ để ước tính giá trị thống kê của một mẫu dữ liệu. Bằng cách chuẩn bị nhiều mẫu bootstrap khác nhau và ước tính đại lượng thống kê và tính giá trị trung bình của các ước tính, có thể đạt được ước tính tổng thể tốt hơn về đại lượng mong muốn hơn là chỉ ước tính trực tiếp từ tập dữ liệu. Theo cách tương tự, nhiều bộ dữ liệu huấn luyện khác nhau có thể được chuẩn bị, được sử dụng để ước tính một mô hình dự đoán và đưa ra dự đoán. Tính trung bình các dự đoán trên các mô hình thường dẫn đến dự đoán tốt hơn so với một mô hình duy nhất phù hợp trực tiếp trên tập dữ liệu huấn luyện. Có thể tóm tắt các yếu tố chính của đóng gói như sau: (i) các mẫu bootstrap của tập dữ liệu huấn luyện; (ii) các cây quyết định chưa được cắt ghép phù hợp với từng mẫu; (iii) bỏ phiếu đơn giản hoặc lấy trung bình các dự đoán. Tóm lại, đóng góp là ở sự khác nhau của dữ liệu huấn luyện được sử dụng để phù hợp với từng thành viên trong nhóm, do đó, dẫn đến các mô hình khéo léo nhưng khác nhau.




*Hình 3.2: Kỹ thuật học kết hợp Bagging [9]*

Stacking: Tổng quát hóa xếp chồng, hay viết tắt là xếp chồng, là một phương pháp tổng hợp nhằm tìm kiếm một nhóm thành viên đa dạng bằng cách thay đổi các loại mô hình phù hợp với dữ liệu huấn luyện và sử dụng một mô hình để kết hợp các dự đoán. Xếp chồng có danh pháp riêng trong đó các thành viên tập hợp được gọi là mô hình cấp 0 và mô hình được sử dụng để kết hợp các dự đoán được gọi là mô hình cấp 1. Hệ thống phân cấp hai cấp của mô hình là cách tiếp cận phổ biến nhất, mặc dù có thể sử dụng nhiều lớp mô hình hơn. Ví dụ: thay vì một mô hình cấp 1, có thể có 3 hoặc 5 mô hình cấp 1 và một mô hình cấp 2 kết hợp các dự đoán của các mô hình cấp 1 để đưa ra dự đoán. Bất kỳ mô hình học máy nào cũng có thể được sử dụng để tổng hợp các dự đoán, mặc dù thường sử dụng mô hình tuyến tính, chẳng hạn như hồi quy tuyến tính cho hồi quy và hồi quy logistic cho phân loại nhị phân. Điều này khuyến khích sự phức tạp của mô hình nằm ở các mô hình thành viên tổng hợp cấp thấp hơn và các mô hình đơn giản để học cách khai thác sự đa dạng của các dự đoán được đưa ra. Có thể tóm tắt các yếu tố chính của việc xếp chồng như sau: (i) Tập dữ liệu huấn luyện không thay đổi; (ii) Các thuật toán học máy khác nhau cho từng thành viên trong nhóm; (iii) Mô hình học máy để tìm hiểu cách kết hợp các dự đoán tốt nhất. Sự đa dạng đến từ các mô hình học máy khác nhau được sử dụng như các thành viên tập

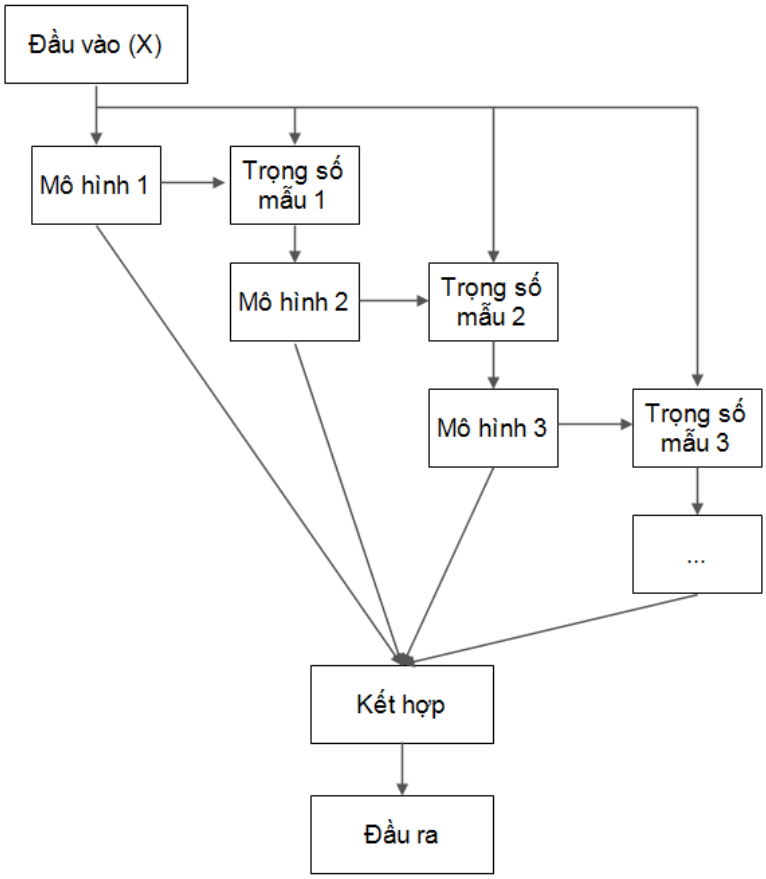
hợp. Do đó, nên sử dụng một bộ các mô hình đã học hoặc được xây dựng theo những cách rất khác nhau, đảm bảo rằng chúng đưa ra các giả định khác nhau và do đó, có ít sai số dự đoán tương quan hơn.




*Hình 3.3: Kỹ thuật học kết hợp Stacking [9]*

Boosting: là một phương pháp tổng hợp tìm cách thay đổi dữ liệu huấn luyện để tập trung sự chú ý vào các ví dụ mà các mô hình phù hợp trước đó trên tập dữ liệu huấn luyện đã mắc lỗi. Đặc tính quan trọng của việc thúc đẩy các nhóm là ý tưởng sửa chữa các lỗi dự đoán. Các mô hình phù hợp và được thêm vào nhóm một cách tuần tự sao cho mô hình thứ hai cố gắng sửa chữa các dự đoán của mô hình đầu tiên, mô hình thứ ba sửa chữa mô hình thứ hai, v..v. Điều này thường liên quan đến việc sử dụng các cây quyết định rất đơn giản chỉ đưa ra một hoặc một vài quyết định, được gọi là tăng cường là những "weak-leaners". Dự đoán của những "weak-leaners" được kết hợp bằng cách sử dụng biểu quyết đơn giản hoặc tính trung bình, mặc dù những đóng góp được cân nhắc tỷ lệ thuận với thành tích hoặc năng lực của chúng. Mục tiêu là phát triển cái gọi là "strong-leaners" từ nhiều "weak-leaners" được xây dựng có mục đích. Thông thường, tập dữ liệu huấn luyện được giữ nguyên và thay vào đó, thuật toán học tập được sửa đổi để chú ý ít nhiều đến các ví dụ cụ thể (*các hàng dữ liệu*) dựa trên việc chúng đã được dự đoán đúng hay sai bởi các thành viên nhóm đã

thêm trước đó. Ví dụ: các hàng dữ liệu có thể được cân nhắc để chỉ ra trọng số mà một thuật toán học phải đưa ra trong khi học mô hình.




*Hình 3.4: Kỹ thuật học kết hợp Boosting [9]*

Có thể tóm tắt các bước như sau: (i) Dữ liệu huấn luyện thiên về những ví dụ khó dự đoán; (ii) Thêm lặp đi lặp lại các thành viên tổng hợp để sửa các dự đoán của các mô hình trước đó; (iii) Kết hợp các dự đoán bằng cách sử dụng trung bình có trọng số của các mô hình. Ý tưởng kết hợp nhiều người học yếu thành người học mạnh lần đầu tiên được đề xuất về mặt lý thuyết và nhiều thuật toán đã được đề xuất với ít thành công. Mãi cho đến khi thuật toán thúc đẩy thích ứng (*AdaBoost*) được phát triển, thì việc thúc đẩy được chứng minh là một phương pháp tổng hợp hiệu quả.

## 3.2. PHƯƠNG PHÁP PHÁT HIỆN BOTNET DỰA TRÊN HỌC KẾT HỢP

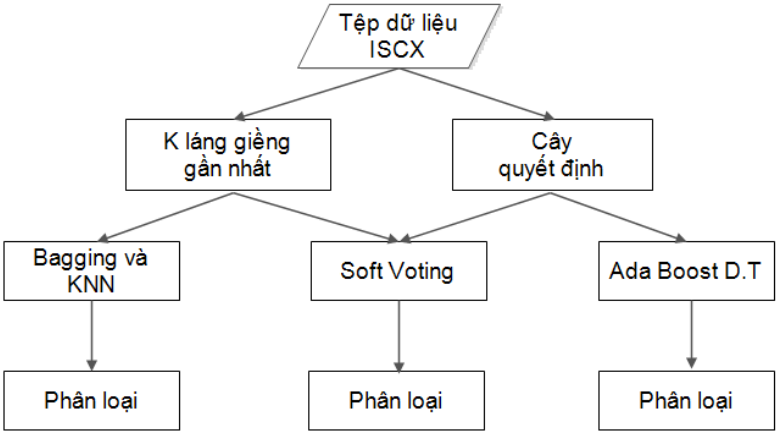
Mặc dù học kết hợp đã được sử dụng trong nhiều lĩnh vực, nhưng trong lĩnh vực phát hiện botnet nói chung và phát hiện DGA botnet nói riêng, số lượng các các đề

xuất phát hiện botnet sử dụng học kết hợp còn chưa nhiều. Mục này thực hiện khảo sát một số đề xuất tiêu biểu, gồm Bijalwan và cộng sự [1], Zahraa và cộng sự [104], Charan và cộng sự [10], Rezaei [76] và Liu và cộng sự [49].

### 3.2.1. Các phương pháp phát hiện DGA botnet dựa trên học kết hợp

Bijalwan và cộng sự [1] đề xuất phương pháp phân tích lưu lượng botnet dựa trên thuật toán phân loại kết hợp để tìm ra các bằng chứng về hoạt động của các bot sử dụng tập dữ liệu ISCX [4] được mô tả tại Hình 3.5. Sau khi trích xuất các đặc trưng của tập dữ liệu này, thực hiện chia các đặc trưng này thành hai lớp, gồm lớp lưu lượng truy cập thông thường và lớp lưu lượng truy cập botnet. Sau đó, tập dữ liệu được đưa vào huấn luyện sử dụng thuật toán phân loại tổng hợp.

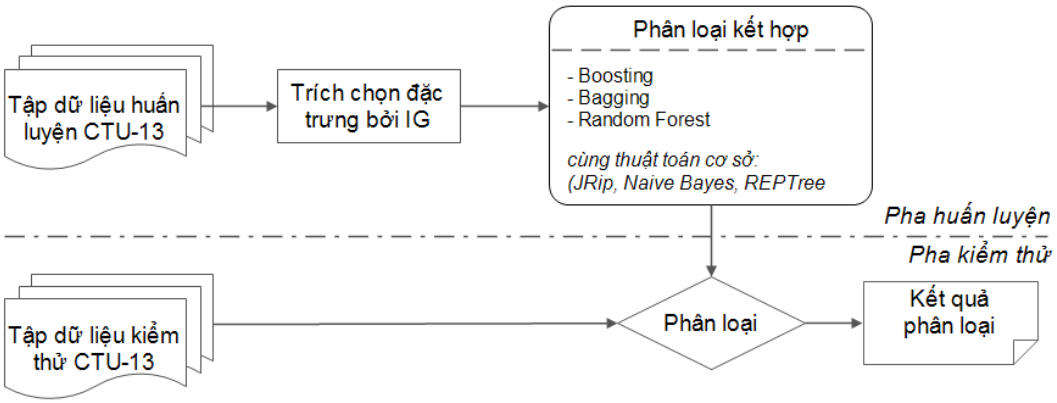
Kết quả thử nghiệm cho thấy rằng phương pháp sử dụng kết hợp các bộ phân loại tốt hơn so với từng bộ phân loại đơn lẻ. Bộ phân loại kết hợp hoạt động tốt hơn bộ phân loại đơn lẻ bằng cách kết hợp sức mạnh của nhiều thuật toán hoặc giới thiệu đặc trưng phân loại cho cùng một bộ phân loại bằng cách thay đổi đầu vào trong phân tích bot.




*Hình 3.5: Mô hình phân loại dựa trên kết hợp*

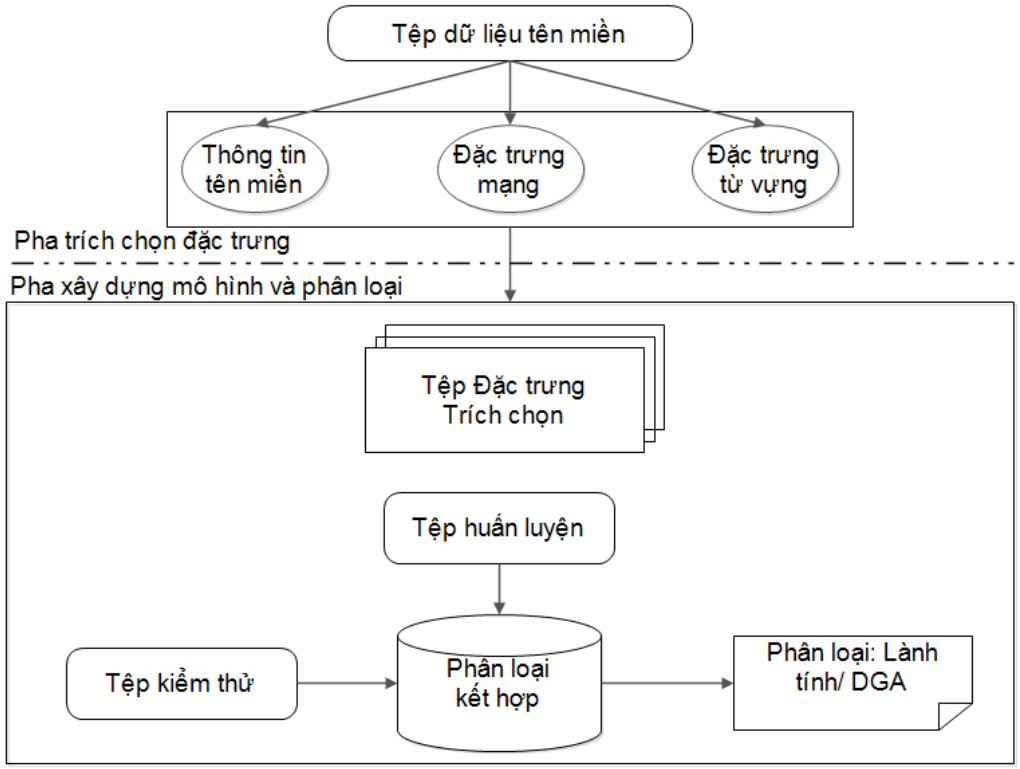
Zahraa và cộng sự [104] đề xuất mô hình dựa trên các phương pháp phân loại tổng hợp cho hoạt động tốt hơn thông qua việc kết hợp nhiều thuật toán trong quá trình phân tích mạng botnet. Ngoài ra, thông qua quá trình lựa chọn đặc trưng, các

đặc trưng quan trọng nhất đã được trích xuất cho quá trình phân tích để tăng độ chính xác và giảm thời gian cũng như nguồn lực. Phương pháp đề xuất minh họa tại Hình 3.6 đã thực hiện đánh giá thử nghiệm trên tập dữ liệu mạng NetFlow CTU botnet [81] và hiệu suất của mô hình đề xuất đã được đánh giá sử dụng phương pháp kiểm tra chéo 10 lần. Các tập dữ liệu NetFlow được sử dụng bao gồm các thuộc tính sau: thời gian bắt đầu, thời lượng, địa chỉ IP nguồn, cổng nguồn, hướng, địa chỉ IP đích, cổng đích, trạng thái giao thức (*ví dụ: UTP, TCP*), SToS (*Loại dịch vụ*), tổng gói (*đã trao đổi giữa nguồn và đích*), tổng số byte và nhãn (*ví dụ: nền, bình thường và mạng botnet*). Kết quả cho thấy mô hình đề xuất có nhiều triển vọng.




*Hình 3.6: Mô hình phát hiện botnet dựa trên học kết hợp của Zahraa*

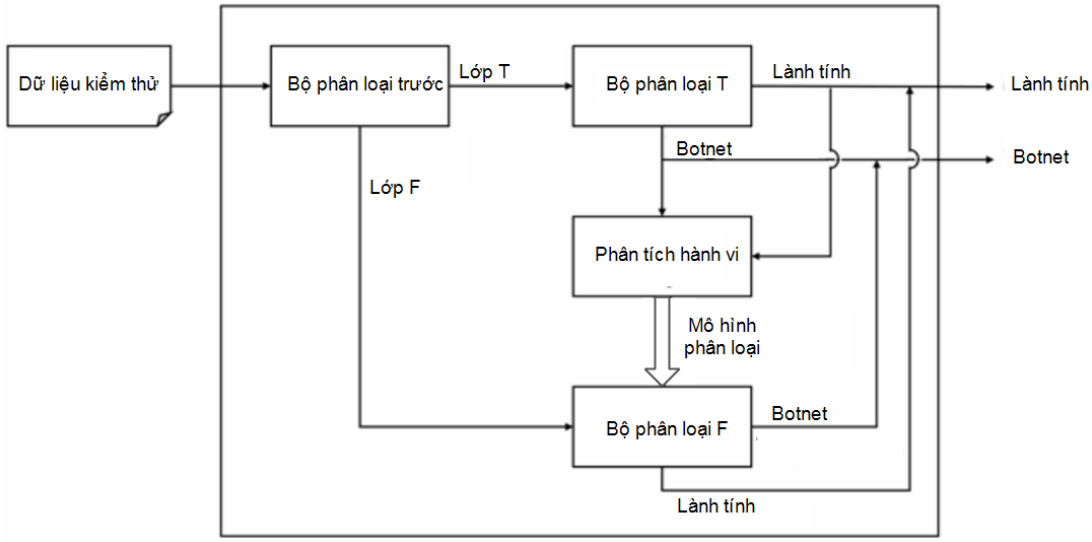
Charan và cộng sự [10] đề xuất một phương pháp mới cho phát hiện tên miền word-based DGA sử dụng các phương pháp tiếp cận tổng hợp với 15 đặc trưng. Nghiên cứu này đã giải quyết vấn đề phát hiện tên miền word-based DGA (gồm *Matsnu, Gozi và Suppobox*) sử dụng các mô hình tổng hợp với khả năng phát hiện gần thời gian thực, trong đó có xem xét cả các đặc trưng từ vựng và cấp độ mạng. Nghiên cứu cũng áp dụng các kỹ thuật giảm kích thước tuyến tính và phi tuyến tính khác nhau để hiểu cấu trúc cơ bản của dữ liệu. Ngoài ra, nghiên cứu cũng sử dụng CTGAN để tạo dữ liệu thử nghiệm tổng hợp nhằm đo lường mức độ mạnh mẽ của mô hình. Kết quả thử nghiệm cho thấy, thuật toán cây quyết định C5.0 cho kết quả tốt nhất với độ chính xác dự đoán là 95.03%.




*Hình 3.7: Mô hình phát hiện kết hợp của Charan*

Rezaei [76] đề xuất phương pháp nhằm mục đích tạo ra một mô hình học kết hợp bằng cách sử dụng các phương pháp học máy tốt nhất giữa học có giám sát, học không giám sát và học hồi quy để tối ưu hóa độ chính xác của việc phát hiện botnet trên IoT và giảm thiểu số lượng tính năng được yêu cầu. Trong nghiên cứu này, một thuật toán học máy mới (*học kết hợp*) cho phát hiện botnet và bot trong mạng IoT đã được đề xuất bằng cách kết hợp hai thuật toán tốt nhất được lựa chọn từ một số phương pháp học có giám sát, học không giám sát và phương pháp học hồi quy được chọn, đó là: (i) ANN và (ii) DT. Thông qua thuật toán học kết hợp, độ chính xác đã đạt được hơn 99.00% chỉ trong thời gian 11.36 giây, chỉ yêu cầu sử dụng 20 đặc trưng để phát hiện botnet và bot trong mạng IoT. Nghiên cứu này cũng so sánh kết quả với các nghiên cứu trước đây đã được thực hiện trong lĩnh vực này, cho thấy nghiên cứu này đạt được độ chính xác cao nhất so với các nghiên cứu trước đó. Có một vài điểm trở ngại của nghiên cứu này. Thuật toán học máy không thể được kiểm tra trong bất kỳ mạng thực nào và chỉ có thể được kiểm tra trong môi trường phòng thí nghiệm. Điều này gây ra một số hạn chế về tập dữ liệu. Một trong những hạn chế là có quyền

truy cập hạn chế vào các loại thiết bị IoT khác nhau, do đó, việc kiểm tra chúng trong lớp vật lý trở nên khó khăn.




*Hình 3.8: Khung phân loại*

Liu và cộng sự [49] đề xuất khung phân loại được được thể hiện trong Hình 3.8 Đây là một bộ phân loại tổng hợp chứa ba bộ phân loại: bộ phân loại trước (*Pre-classifier*), bộ phân loại T (*T-classifier*) và bộ phân loại F (*F-classifier*). Bộ phân loại trước là bộ phân loại đầu tiên của khung phân loại. Trong mô hình này kết quả phân loại của bộ phân loại trước chỉ là một dự đoán dữ liệu thử nghiệm. Lớp T là một tập hợp dữ liệu thử nghiệm sẽ được phân loại chính xác bởi bộ phân loại T; Lớp F là một tập hợp dữ liệu thử nghiệm sẽ được phân loại sai bởi bộ phân loại T. Bộ phân loại T chia dữ liệu thử nghiệm trong lớp T thành 2 tập con lành tính và botnet.

Bộ phân loại F chia dữ liệu thử nghiệm trong lớp F thành 2 tập con lành tính và botnet. Vì dữ liệu thử nghiệm trong lớp F sẽ bị phân loại sai bởi bộ phân loại T, dữ liệu thử nghiệm trong lớp F là nhiễu. Trong bộ phân loại F, không thể sử dụng các tính năng hoặc thuật toán phân loại được sử dụng trong bộ phân loại T. Do đó, sẽ tìm các tính năng và thuật toán phân loại khác để huấn luyện mô hình phân loại của bộ phân loại F. Dữ liệu thử nghiệm trong lớp F sẽ được phân loại theo mô hình phân loại được huấn luyện bằng cách sử dụng kết quả phân loại của dữ liệu thử nghiệm trong lớp T. Trong đề xuất này, tác giả sử dụng các tệp dữ liệu CTU(42), CTU(43),

CTU(46), CTU(50) và CTU(54) với các thuật toán học máy SVM, Naïve Beyes và cây quyết định cho hiệu suất chung đạt trên 99.00%.

### 3.2.2. Ưu, nhược điểm của các đề xuất phát hiện botnet dựa trên học kết hợp

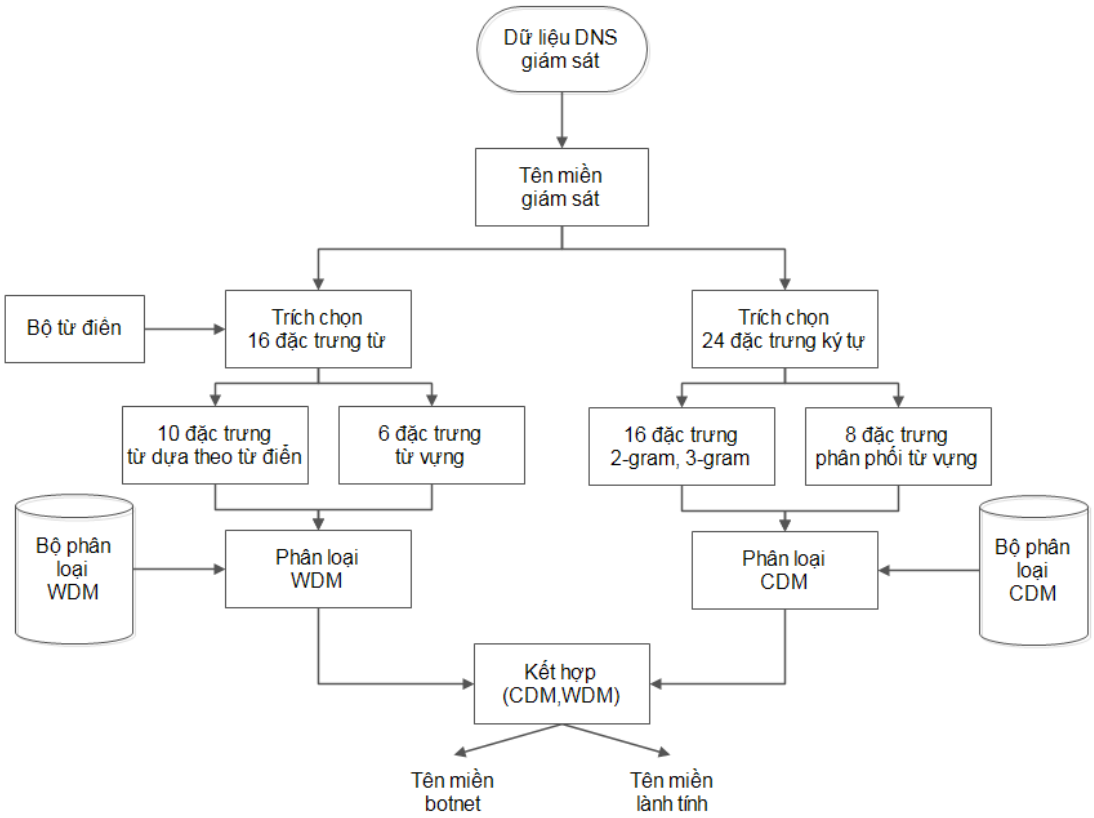
Qua các khảo sát ở trên cho thấy, các phương pháp đã sử dụng các thuật toán học kết hợp có thể kể đến như Adaboost, Stacking, Bagging và các phương pháp học kết hợp khác làm tăng hiệu quả phát hiện đáng kể. Chẳng hạn, như đề xuất của Bijanwan và cộng sự [1] kết quả cho thấy rằng bộ phân loại kết hợp sử dụng phương pháp voting cho độ chính xác tăng từ 93.37% lên 96.41%. Các đề xuất của Rezaei [76], Liu và cộng sự [49] đều cho hiệu suất chung đạt tỷ lệ trên 99.00%.

Trong các phương pháp phát hiện botnet dựa trên kết hợp nêu trên đều có sử dụng đến các đặc trưng mạng. Khi sử dụng các đặc trưng mạng đòi hỏi nhiều các chi phí liên quan đến lưu trữ, xử lý tài nguyên mạng lớn. Cũng với phương pháp học kết hợp được nêu ở trên, phần còn lại của chương này đề xuất mô hình phát hiện DGA botnet dựa trên học kết hợp giữa 2 mô hình đã đề xuất ở chương 2 đó là: mô hình CDM và mô hình WDM.

## 3.3. MÔ HÌNH PHÁT HIỆN DGA BOTNET DỰA TRÊN HỌC KẾT HỢP

### 3.3.1. Giới thiệu mô hình

Từ kết quả 2 mô hình phát hiện character-based DGA botnet (CDM) và word-based DGA botnet (WDM) đã được trình bày tại mục 2.2 và mục 2.3, có thể thấy mỗi mô hình đều có điểm mạnh và điểm yếu riêng. Mô hình CDM có khả năng phát hiện hiệu quả các character-based DGA botnet, nhưng hầu như không thể phát hiện các word-based DGA botnet. Ngược lại, mô hình WDM có khả năng phát hiện hiệu quả các word-based DGA botnet, nhưng tỷ lệ phát hiện các character-based DGA botnet của WDM nói chung thấp hơn CDM. Do vậy, phần này đề xuất mô hình dựa trên học kết hợp để hợp nhất 2 mô hình CDM và WDM nhằm phát huy ưu điểm của cả hai mô hình trong một mô hình phát hiện DGA botnet thống nhất.




*Hình 3.9: Giai đoạn phát hiện của mô hình học kết hợp đề xuất*

Mô hình kết hợp có giai đoạn huấn luyện được kế thừa từ hai mô hình thành phần CDM và WDM đã được mô tả ở chương 2. Theo đó, các mô hình CDM và WDM được huấn luyện riêng và kết quả là hai bộ phân loại CDM và WDM. Trong giai đoạn phát hiện của mô hình kết hợp như biểu diễn trên Hình 3.9, 2 mô hình CDM và WDM được sử dụng để xử lý tên miền song song và kết quả từ 2 mô hình sẽ được kết hợp để tăng hiệu suất phát hiện chung. *Kết hợp (CDM, WDM)* là phép hợp các phát hiện chính xác từ 2 mô hình. Việc kết hợp này xuất phát từ ý tưởng lấy kết quả phát hiện chính xác của từng mô hình, xếp chồng cho kết quả là số lượng tối đa các DGA bonet được phát hiện. Giả sử rằng CDM phát hiện ra x mẫu, WDM phát hiện ra y mẫu, trong đó: $x \in X$ là số mẫu CDM phát hiện được nhưng WDM thì không, $y \in Y$ là số mẫu CDM phát hiện được và ngược lại. Khi kết hợp sẽ được $X \cup Y$. Vì sử dụng chung dữ liệu kiểm thử đối với cả 2 mô hình nên trọng số khi kết hợp được tính theo tỷ lệ 1:1.

Mô hình được đề xuất là một kiểu kết hợp "muộn" trong học kết hợp, ở mô hình này CDM và WDM tạo ra kết quả của riêng chúng và kết quả cuối cùng là sự kết hợp của cả hai mô hình. Để tìm ra phương pháp kết hợp hiệu quả nhất, thực hiện kết hợp "sớm" giữa CDM và WDM, sau đó so sánh với kết quả của kết hợp "muộn" của mô hình được đề xuất. Trong phương pháp kết hợp "sớm", việc kết hợp mô hình được thực hiện ở bước trích xuất đặc trưng. Theo cách tiếp cận này, mô hình "tập hợp ban đầu" được xây dựng bằng cách sử dụng bộ 38 đặc trưng kết hợp gồm 24 đặc trưng ký tự của CDM và các 16 đặc trưng từ của WDM trong giai đoạn huấn luyện. Trong giai đoạn phát hiện, mỗi tên miền được giám sát sẽ được xử lý và chuyển đổi thành một vector gồm 38 đặc trưng để phân loại nhằm xác định xem đó là tên miền thông thường hay DGA.

### 3.3.2. Tập dữ liệu huấn luyện và kiểm thử

**Dữ liệu huấn luyện:** Đối với mô hình học kết hợp đề xuất, sử dụng bộ phân loại đã được huấn luyện từ hai mô hình CDM và WDM đề xuất tại chương 2. Để tăng hiệu quả của phương pháp kết hợp khi số mẫu CDM phán đoán sai sẽ được WDM phán đoán, do đó các tính năng và thuật toán trong CDM và WDM sẽ khác nhau.

**Dữ liệu kiểm thử:** như đã đề cập ở trên, để so sánh hiệu suất của các mô hình, tập dữ liệu kiểm thử sẽ được lấy từ tệp dữ liệu kiểm thử trong mô hình phát hiện character-based DGA botnet được trình bày tại mục 2.2.3. Dữ liệu kiểm thử gồm các tên miền của 39 họ DGA botnet với 71,393 mẫu tên miền DGA botnet. Ngoài ra, sử dụng 31,000 tên miền DGA botnet với 7 họ trên tập dữ liệu UMUDGA để đánh giá khả năng phát hiện các tên miền DGA botnet mới, không tồn tại trong tập dữ liệu huấn luyện lấy từ Netlab360.

### 3.3.3. Tiền xử lý, huấn luyện và phát hiện

Trong giai đoạn huấn luyện xây dựng mô hình, khâu tiền xử lý dữ liệu được thực hiện riêng với từng mô hình phát hiện thành phần: 24 đặc trưng từ được trích xuất với mô hình CDM và 16 đặc trưng từ được trích xuất với mô hình WDM. Việc huấn luyện được thực hiện riêng với mỗi mô hình CDM và WDM. Trong quá trình

phát hiện, khâu tiền xử lý dữ liệu được thực hiện riêng theo thủ tục tương tự giai đoạn huấn luyện cho mỗi mô hình thành phần. Mỗi mô hình thành phần CDM và WDM đầu tiên phân loại vector đặc trưng của tên miền riêng, sau đó các kết quả thành phần được kết hợp lại để đưa ra kết quả tổng hợp.

### 3.3.4. Các kết quả

Bảng 3.1 biểu diễn tỷ lệ phát hiện (DR) của các mô hình CDM, WDM và mô hình kết hợp với các DGA botnet có $DR_{\text{kết hợp}} > 90\%$. Bảng 3.10 biểu diễn tỷ lệ phát hiện (DR) của các mô hình CDM, WDM và mô hình kết hợp với các DGA botnet có $DR_{\text{kết hợp}} < 90\%$. Từ các kết quả trên có thể thấy, mô hình kết hợp đã kết hợp được các ưu điểm của các mô hình CDM và WDM: phát hiện hiệu quả cả các character-based và word-based DGA botnet.

*Bảng 3.1: Các DGA botnet có tỷ lệ DR lớn hơn 90% với mô hình đề xuất*

| STT | Họ botnet | Số lượng | CDM % | WDM % | Kết hợp % |
|---|---|---|---|---|---|
| 1 | Rovnix | 4000 | 100.00 | 99.50 | 100.00 |
| 2 | Dyre | 1000 | 100.00 | 99.90 | 100.00 |
| 3 | Chinad | 1000 | 100.00 | 97.90 | 100.00 |
| 4 | Fobber_v1 | 298 | 100.00 | 100.00 | 100.00 |
| 5 | Tinynuke | 32 | 100.00 | 100.00 | 100.00 |
| 6 | Gameover | 4000 | 100.00 | 99.98 | 100.00 |
| 7 | Murofet | 4000 | 99.80 | 99.78 | 100.00 |
| 8 | Cryptolocker | 1000 | 99.70 | 96.20 | 100.00 |
| 9 | Padcrypt | 168 | 98.21 | 98.21 | 100.00 |
| 10 | Dircrypt | 762 | 99.34 | 93.31 | 100.00 |
| 11 | Fobber_v2 | 299 | 100.00 | 89.30 | 100.00 |
| 12 | Vidro | 100 | 100.00 | 49.00 | 100.00 |
| 13 | Emotet | 4000 | 99.68 | 99.55 | 99.98 |
| 14 | Tinba | 4000 | 99.98 | 99.08 | 99.98 |
| 15 | Ranbyus | 4000 | 99.58 | 99.30 | 99.93 |
| 16 | Shiotob | 4000 | 99.68 | 95.95 | 99.88 |
| 17 | Pykspa_v1 | 4000 | 99.70 | 58.90 | 99.83 |
| 18 | Necurs | 4000 | 99.35 | 87.75 | 99.78 |
| 19 | Ramnit | 4000 | 99.55 | 91.45 | 99.75 |
| 20 | Virut | 4000 | 99.75 | 0.00 | 99.75 |
| 21 | Qadars | 2000 | 99.05 | 95.40 | 99.65 |
| 22 | Simda | 4000 | 99.65 | 0.00 | 99.65 |

| | | | | | |
|---|---|---|---|---|---|
| 23 | Pykspa_v2_fake | 799 | 98.87 | 61.08 | 99.25 |
| 24 | Locky | 1158 | 99.05 | 83.16 | 99.14 |
| 25 | Pykspa_v2_real | 199 | 98.99 | 63.32 | 98.99 |
| 26 | Shifu | 2546 | 98.59 | 34.92 | 98.82 |
| 27 | Matsnu | 881 | 12.15 | 98.41 | 98.64 |
| 28 | Proslikefan | 100 | 98.00 | 50.00 | 98.00 |
| 29 | Tempedreve | 195 | 97.44 | 64.62 | 97.44 |
| 30 | Vawtrak | 827 | 96.61 | 61.67 | 97.10 |
| 31 | Symmi | 1200 | 96.58 | 31.67 | 96.83 |
| 32 | Suppobox | 2205 | 19.27 | 92.83 | 96.05 |
| 33 | Nymaim | 480 | 94.79 | 61.25 | 95.21 |
| 34 | Mydoom | 50 | 88.00 | 74.00 | 94.00 |
| | **Tổng cộng** | **65299** | **95.59** | **77.18** | **99.53** |

*Bảng 3.2: Các DGA botnet có tỷ lệ DR nhỏ hơn 90% với mô hình đề xuất*

| STT | Họ botnet | Số lượng | CDM % | WDM % | Kết hợp % |
|---|---|---|---|---|---|
| 1 | Conficker | 495 | 89.29 | 52.93 | 89.49 |
| 2 | Bigviktor | 999 | 11.11 | 70.97 | 76.18 |
| 3 | Gspy | 100 | 76.00 | 8.00 | 76.00 |
| 4 | Enviserv | 500 | 50.40 | 19.40 | 52.00 |
| 5 | Banjori | 4000 | 0.00 | 0.00 | 0.00 |
| | **Tổng cộng** | **6094** | **14.46** | **17.66** | **25.24** |

*Bảng 3.3: Tỷ lệ phát hiện đối với tập dữ liệu UMUDGA*

| STT | Họ botnet | Số lượng | CDM % | WDM % | Kết hợp % |
|---|---|---|---|---|---|
| 1 | Alureon | 5,000 | 98.22 | 85.32 | 98.94 |
| 2 | Bedep | 5,000 | 99.82 | 97.80 | 99.92 |
| 3 | Corebot | 5,000 | 99.76 | 98.24 | 99.94 |
| 4 | Kraken | 2,000 | 98.40 | 69.50 | 99.10 |
| 5 | Pushdo | 5,000 | 94.36 | 35.90 | 95.08 |
| 6 | Zeus | 5,000 | 100.00 | 99.96 | 100.00 |
| | | **27,000** | **98.43** | **82.41** | **98.80** |
| 7 | Pizd | 4,000 | 16.05 | 97.93 | 98.05 |
| | **Tổng cộng** | **31,000** | **87.80** | **84.41** | **98.70** |

*Bảng 3.4: Tỷ lệ phát hiện giữa CDM, WDM, mô hình kết hợp đề xuất và kết hợp "sớm"*

| STT | Họ botnet | Kiểu DGA | Tỷ lệ phát hiện (DR%) | | | |
|---|---|---|---|---|---|---|
| | | | CDM | WDM | Kết hợp đề xuất | Kết hợp "sớm" |

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | Rovnix | char-based | 100.00 | 99.50 | 100.00 | 100.00 |
| 2 | Dyre | char-based | 100.00 | 99.90 | 100.00 | 100.00 |
| 3 | Chinad | char-based | 100.00 | 97.90 | 100.00 | 99.80 |
| 4 | Fobber_v1 | char-based | 100.00 | 100.00 | 100.00 | 100.00 |
| 5 | Tinynuke | char-based | 100.00 | 100.00 | 100.00 | 100.00 |
| 6 | Gameover | char-based | 100.00 | 99.98 | 100.00 | 100.00 |
| 7 | Murofet | char-based | 99.80 | 99.78 | 100.00 | 99.83 |
| 8 | Cryptolocker | char-based | 99.70 | 96.20 | 100.00 | 98.80 |
| 9 | Padcrypt | char-based | 98.21 | 98.21 | 100.00 | 98.81 |
| 10 | Dircrypt | char-based | 99.34 | 93.31 | 100.00 | 96.06 |
| 11 | Fobber_v2 | char-based | 100.00 | 89.30 | 100.00 | 94.98 |
| 12 | Vidro | char-based | 100.00 | 49.00 | 100.00 | 64.00 |
| 13 | Emotet | char-based | 99.68 | 99.55 | 99.98 | 99.85 |
| 14 | Tinba | char-based | 99.98 | 99.08 | 99.98 | 99.98 |
| 15 | Ranbyus | char-based | 99.58 | 99.30 | 99.93 | 99.75 |
| 16 | Shiotob | char-based | 99.68 | 95.95 | 99.88 | 99.30 |
| 17 | Pykspa_v1 | char-based | 99.70 | 58.90 | 99.83 | 71.53 |
| 18 | Necurs | char-based | 99.35 | 87.75 | 99.78 | 93.50 |
| 19 | Ramnit | char-based | 99.55 | 91.45 | 99.75 | 95.33 |
| 20 | Virut | char-based | 99.75 | 0.00 | 99.75 | 0.00 |
| 21 | Qadars | char-based | 99.05 | 95.40 | 99.65 | 98.65 |
| 22 | Simda | char-based | 99.65 | 0.00 | 99.65 | 7.83 |
| 23 | Suppobox | word-based | 19.27 | 99.30 | 99.30 | 72.22 |
| 24 | Pykspa_v2_fake | char-based | 98.87 | 61.08 | 99.25 | 89.72 |
| 25 | Locky | char-based | 99.05 | 83.16 | 99.14 | 72.36 |
| 26 | Pykspa_v2_real | char-based | 98.99 | 63.32 | 98.99 | 59.58 |
| 27 | Shifu | char-based | 98.59 | 34.92 | 98.82 | 98.30 |
| 28 | Matsnu | word-based | 12.15 | 98.41 | 98.64 | 66.00 |
| 29 | Proslikefan | char-based | 98.00 | 50.00 | 98.00 | 77.44 |
| 30 | Tempedreve | char-based | 97.44 | 64.62 | 97.44 | 71.10 |
| 31 | Vawtrak | char-based | 96.61 | 61.67 | 97.10 | 46.75 |
| 32 | Symmi | char-based | 96.58 | 31.67 | 96.83 | 95.96 |
| 33 | Bigviktor | word-based | 11.11 | 96.78 | 96.78 | 70.00 |
| 34 | Nymaim | char-based | 94.79 | 61.25 | 95.21 | 86.00 |
| 35 | Mydoom | char-based | 88.00 | 74.00 | 94.00 | 61.21 |
| 36 | Gspy | char-based | 91.00 | 8.00 | 91.00 | 94.39 |
| 37 | Conficker | char-based | 89.29 | 52.93 | 89.49 | 2.00 |
| 38 | Enviserv | char-based | 76.00 | 19.40 | 76.00 | 13.60 |
| 39 | Banjori | mix-based | 0.00 | 0.00 | 0.00 | 0.00 |

### 3.3.5. Đánh giá

Bảng 3.1 thống kê các botnet có tỷ lệ DR lớn hơn 90% khi sử dụng mô hình kết hợp đề xuất, trong tổng số 39 họ DGA botnet thì có 34 họ cho DR lớn hơn 94.00%. 12 họ DGA botnet có DR đạt 100%, gồm Rovnix, Dyre, Chinad, Fobber_v1, Tinynuke, Gameover, Murofet, Cryptolocker, Padcrypt, Dircrypt, Fobber_v2 và Vidro. Tỷ lệ phát hiện trung bình của nhóm này đạt tới 99.53%.

Bảng 3.2 liệt kê 5 họ DGA botnet có DR không cao khi sử dụng mô hình kết hợp. Trong đó, Conficker có DR đạt 89.49%, Bigviktor và Gspy có DR đạt khoảng 76%, Enviserv có DR đạt 52%. Đặc biệt, mô hình kết hợp không thể phát hiện Banjori botnet do các mô hình thành phần cũng không thể phát hiện botnet này.

Bảng 3.3 thể hiện tỷ lệ phát hiện của các mô hình CDM, WDM và kết hợp đối với tệp dữ liệu kiểm thử lấy từ tập UMUDGA. Kết quả cho thấy, đối với các character-based DGA botnet (6 họ botnet đầu danh sách), mô hình CDM cho tỷ lệ phát hiện đạt 98.43%. Đối với ‘Pizd’ là word-based DGA botnet, mô hình kết hợp cho tỷ lệ phát hiện đạt 98.05%. Tỷ lệ phát hiện tổng thể của mô hình kết hợp đạt 98.70% đối với 31,000 tên miền DGA botnet, bao gồm cả character-based và word-based DGA botnet.

Mô hình phát hiện botnet DGA được đề xuất dựa trên học kết hợp “muộn” có thể phát hiện hiệu quả hầu hết các botnet DGA, bao gồm character-based và word-based DGA botnet vì nó có thể tận dụng lợi thế của cả mô hình CDM và WDM thành phần. Kết quả thực nghiệm đưa ra trong Bảng 3.4 cho thấy mô hình kết hợp được đề xuất có khả năng phát hiện hiệu quả 37 trong số 39 họ botnet DGA có DR> 89%, trong đó 12 họ botnet DGA có DR = 100% và 31 botnet có DR> 97%. Bảng 3.4 cũng cho thấy mô hình kết hợp “muộn” được đề xuất hoạt động tốt hơn nhiều so với mô hình kết hợp “sớm”. Tỷ lệ phát hiện (DR) của mô hình kết hợp “muộn” được đề xuất cao hơn đáng kể so với mô hình kết hợp “sớm” cho tất cả các họ botnet, ngoại trừ

*gspy*. Mô hình kết hợp "sớm" thậm chí không phát hiện được một số botnet DGA, chẳng hạn như *virus*, *simda*, *conficker* và *enviserv*.

Tóm lại, mô hình phát hiện kết hợp đề xuất đã khai thác được điểm mạnh của cả 2 mô hình thành phần là CDM và WDM: mô hình phát hiện kết hợp có khả năng phát hiện hiệu quả hầu hết các character-based DGA botnet và word-based DGA botnet. Theo đó, mô hình phát hiện kết hợp có DR cao hơn CDM với character-based DGA botnet và mô hình phát hiện kết hợp có DR cao hơn WDM với word-based DGA botnet.

Hạn chế của mô hình kết hợp là thời gian huấn luyện và phát hiện dài hơn các mô hình thành phần, nhưng với hiệu quả phát hiện vượt trội, mô hình kết hợp cho hiệu quả tổng hợp tốt hơn. Một hạn chế khác của mô hình kết hợp là không thể phát hiện một số DGA botnet có phương pháp tạo tên miền đặc biệt như *banjori*.

## 3.4. KẾT LUẬN CHƯƠNG

Chương 3 giới thiệu khái quát về học kết hợp, các phương pháp học kết hợp, bao gồm các phương pháp học kết hợp đơn giản và học kết hợp nâng cao. Các phương pháp học kết hợp đơn giản bao gồm max voting, averaging và weighted averaging, và các phương pháp học kết hợp nâng cao bao gồm bagging, stacking và boosting. Phần tiếp theo của chương này khảo sát một số nghiên cứu phát hiện botnet dựa trên học kết hợp. Nhìn chung, các đề xuất phát hiện botnet dựa trên học kết hợp có số lượng khá ít và hiệu quả của học kết hợp chưa thực sự rõ ràng.

Phần cuối của chương tập trung giải quyết vấn đề phát hiện cả character-based và word-based DGA botnet trong một mô hình thống nhất bằng cách đề xuất mô hình phát hiện DGA botnet dựa trên học kết hợp. Mô hình phát hiện kết hợp sử dụng hai mô hình thành phần là CDM và WDM đã được thử nghiệm và đánh giá ở chương 2. Mô hình kết hợp đã khai thác được điểm mạnh của cả 2 mô hình thành phần là CDM và WDM: mô hình kết hợp có khả năng phát hiện hiệu quả hầu hết các DGA botnet, bao gồm cả character-based DGA botnet và word-based DGA botnet. Kết quả thử nghiệm cho thấy, trong số 39 họ DGA botnet thử nghiệm, mô hình kết hợp có tỷ lệ

phát hiện DR từ 94% trở lên với 34 họ DGA botnet, trong đó 12 họ botnet có DR đạt 100%. Mô hình kết hợp chỉ không thể phát hiện 1 họ botnet là Banjori do các mô hình thành phần cũng không thể phát hiện botnet này. Mô hình kết hợp đề xuất và kết quả thử nghiệm, đánh giá được đăng trên bài báo "Một mô hình phát hiện DGA botnet dựa trên học kết hợp", tạp chí Khoa học Công nghệ Thông tin và Truyền trông, ISSN: 2525-2224, Vol. 1, No. 1, 2022 [CT3].

# KẾT LUẬN

Botnet đã và đang trở thành một trong các nguy cơ gây mất an toàn thông tin hàng đầu do chúng không ngừng phát triển về cả quy mô và mức độ tinh vi trong các kỹ thuật chỉ huy và kiểm soát. Nhiều dạng botnet sử dụng kỹ thuật DGA để sinh và đăng ký nhiều tên miền ngẫu nhiên khác nhau cho máy chủ CnC của chúng nhằm chống lại việc bị kiểm soát và vô hiệu hóa. Các DGA botnet thường khai thác hệ thống DNS để duy trì hoạt động, do vậy việc phân tích phát hiện các tên miền truy vấn hệ thống DNS có thể giúp phát hiện các hoạt động của botnet. Luận án này tập trung giải quyết hai vấn đề: (1) nghiên cứu, đề xuất tập đặc trưng phân loại tên miền mới phù hợp hơn cho xây dựng các mô hình phát hiện DGA botnet, nhằm tăng tỷ lệ

phát hiện đúng và giảm tỷ lệ cảnh báo sai và (2) nghiên cứu, lựa chọn sử dụng phương pháp học máy phù hợp cho xây dựng các mô hình phát hiện DGA botnet, nhằm xây dựng một mô hình phát hiện thống nhất cho phép phát hiện hiệu quả nhiều dạng DGA botnet.

*Với vấn đề (1) nghiên cứu, đề xuất tập đặc trưng phân loại tên miền mới phù hợp hơn cho xây dựng các mô hình phát hiện DGA botnet, nhằm tăng tỷ lệ phát hiện đúng và giảm tỷ lệ cảnh báo sai,* luận án đề xuất mô hình CDM cho phát hiện character-based DGA botnet và mô hình WDM cho phát hiện word-based DGA botnet. Mô hình phát hiện CDM đề xuất sử dụng 24 đặc trưng ký tự để phân loại tên miền lành tính với tên miền sinh bởi các DGA botnet, gồm 16 đặc trưng thống kê n-gram, 6 đặc trưng phân bố nguyên âm, ký tự, chữ số, và 2 đặc trưng entropy theo ký tự và giá trị kỳ vọng của tên miền. Các thử nghiệm trên tập dữ liệu gồm 100,000 tên miền lành tính và 153,000 tên miền DGA cho thấy, mô hình CDM đề đạt các độ đo đánh giá vượt trội so với các mô hình đã có. Cụ thể, mô hình CDM đạt độ chính xác chung trên 99.60% và tỷ lệ cảnh báo sai rất thấp, chỉ khoảng 0.4%. Như vậy có thể khẳng định tập 24 đặc trưng ký tự sử dụng trong mô hình CMD là phù hợp cho phát hiện các họ character-based DGA botnet.

Mặc dù mô hình CDM đạt hiệu xuất phát hiện tốt cho hầu hết các character-based DGA botnet, CDM không có khả năng phát hiện hiệu quả các họ word-based DGA botnet, như ‘banjori’, ‘matsnu’, ‘bigviktor’ và ‘suppobox’. Điều này là do các word-based DGA botnet có khả năng sinh các tên miền rất giống các tên miền lành tính sử dụng tổ hợp các từ tiếng Anh lấy từ các danh sách dựng sẵn. Để giải quyết vấn đề này, luận án đề xuất mô hình WDM cho phép phát hiện hiệu quả các họ word-based DGA botnet. Mô hình WDM đề xuất sử dụng 16 đặc trưng từ cho phân loại tên miền word-based DGA botnet với các tên miền lành tính, bao gồm 10 đặc trưng từ dựa trên từ điển và 6 đặc trưng từ vựng. Luận án sử dụng 5 thuật toán học máy có giám sát, bao gồm Naïve Bayes, cây quyết định, rừng ngẫu nhiên, hồi quy logistic và SVM để xây dựng và kiểm thử mô hình phát hiện. Các kết quả thử nghiệm trên các tập dữ liệu DATASET-01 và DATASET-02 với 4 kịch bản cho thấy mô hình WDM

có khả năng phát hiện hiệu quả các word-based DGA botnet, cũng như có khả năng phát hiện tốt nhiều character-based DGA botnet với độ đo F1 đạt trên 95%. Trong các thuật toán học máy sử dụng, thuật toán học máy cây quyết định J48 cho hiệu xuất phát hiện tổng thể tốt nhất trong các thuật toán thử nghiệm. Như vậy có thể khẳng định tập 16 đặc trưng từ sử dụng trong mô hình WDM là phù hợp cho phát hiện các họ word-based DGA botnet. Tuy nhiên, hạn chế của mô hình đề xuất này là giới hạn phạm vi các word-based DGA botnet dựa trên từ điển tiếng Anh, chưa sử dụng các bộ từ điển khác ở dạng chữ Latin hoặc tiếng Việt không dấu. Đây cũng chính là hướng mở cho những nghiên cứu tiếp theo.

Với vấn đề (2) *nghiên cứu, lựa chọn sử dụng phương pháp học máy phù hợp cho xây dựng các mô hình phát hiện DGA botnet, nhằm xây dựng một mô hình phát hiện thống nhất cho phép phát hiện hiệu quả nhiều dạng DGA botnet*, luận án đề xuất mô hình phát hiện DGA botnet dựa trên học kết hợp. Mô hình phát hiện kết hợp đề xuất nhằm khai thác điểm mạnh của 2 mô hình thành phần là CDM và WDM: mô hình phát hiện kết hợp có khả năng phát hiện hiệu quả hầu hết các DGA botnet, bao gồm cả character-based DGA botnet và word-based DGA botnet. Các kết quả thử nghiệm cho thấy, mô hình phát hiện dựa trên học kết hợp đạt tỷ lệ phát hiện trung bình là 99.53% trên 39 họ DGA botnet thử nghiệm. Cụ thể, mô hình kết hợp có tỷ lệ phát hiện đạt từ 94% trở lên với 34 họ DGA botnet, trong đó 12 họ botnet có tỷ lệ phát hiện đạt 100%. Trong số 39 họ DGA botnet, chỉ có 5 họ DGA botnet có tỷ lệ phát hiện dưới 90%. Ngoài ra, mô hình kết hợp cũng có khả năng phát hiện hiệu quả các character-based và word-based DGA botnet mới trong tập dữ liệu UMUDGA với tỷ lệ phát hiện trung bình đạt 98,70%.

Các đề xuất phát hiện DGA botnet dựa trên tên miền thực thi hiệu quả hơn so với các phương pháp dựa trên lưu lượng mạng bởi giảm thiểu các đặc trưng, xử lý dữ liệu luồng và gói tin, do đó sẽ nhanh hơn, chi phí đỡ tốn kém hơn. Các mô hình khi đưa vào ứng sẽ được cài đặt tại DNS server nhằm ngăn chặn các bot có thể liên lạc được với CnC server hoặc trước firewall trong các hệ thống đơn lẻ nhằm phát hiện máy tính nào là bot.

Các hạn chế của mô hình kết hợp bao gồm: (1) thời gian huấn luyện và phát hiện dài hơn so với mô hình thành phần và (2) mô hình kết hợp không có khả năng phát hiện một số DGA botnet thuộc họ mixed DGA, như Banjori. Đây cũng là các vấn đề cần giải quyết cho hướng phát triển trong tương lai của luận án. Ngoài ra, việc phát triển hệ thống phát hiện DGA botnet dựa trên các mô hình phát hiện đề xuất cũng là một hướng mở của đề tài luận án.

# DANH MỤC CÁC CÔNG TRÌNH CÔNG BỐ

## TẠP CHÍ KHOA HỌC

[CT1] Xuan Dau Hoang, **Xuan Hanh Vu**, 2021: An improved model for detecting DGA botnets using random forest algorithm, Information Security Journal: A Global Perspective, DOI: 10.1080/19393555.2021.1934198. ESCI Scopus Q2.

[CT2] **Xuan Hanh Vu**, Xuan Dau Hoang, 2021: An Novel Machine Learning-based Approach for Detecting Word-based Botnets, Journal of Theoretical and Applied Information Technology, Vol 99 – 24. Scopus Q4.

[CT3] **Vũ Xuân Hạnh**, Hoàng Xuân Dậu, Đinh Trường Duy, 2022, "Một mô hình phát hiện DGA botnet dựa trên học kết hợp", tạp chí Khoa học Công nghệ Thông tin và Truyền trông, ISSN: 2525-2224, Vol. 1, No. 1, 2022.

## HỘI THẢO KHOA HỌC

[CT4] Hoang X.D., **Vu X.H**, 2021. An Enhanced Model for DGA Botnet Detection Using Supervised Machine Learning. Intelligent Systems and Networks, ICISN 2021. Lecture Notes in Networks and Systems, vol 243. Springer, Singapore. DOI: 10.1007/978-981-16-2094-2_6. Scopus Q4.

[CT5] **Vũ Xuân Hạnh**, Hoàng Xuân Dậu, 2019. Phát hiện DGA Botnet sử dụng kết hợp nhiều nhóm đặc trưng phân loại tên miền. Hội nghị KHCN Quốc gia lần thứ XII (FAIR); Huế, ngày 07-08/6/2019. DOI: 10.15625/vap.2019.00047.

[CT6] Nguyễn Trọng Hưng, Hoàng Xuân Dậu, **Vũ Xuân Hạnh**, 2018 "Phát hiện botnet dựa trên phân loại tên miền sử dụng kỹ thuật học máy", Hội thảo lần III: Một số vấn đề lựa chọn về an toàn an ninh thông tin, Tạp chí Thông tin và truyền thông 12/2018, ISSN: 1859 – 3550.

# TÀI LIỆU THAM KHẢO


1. Anchit B., Nanak C., Emmanuel P., and Rama C. *Botnet Analysis Using Ensemble Classifier.* Perspectives in Science, 2016. **Volume 8**: p. 502-504.
2. Bader J. *Collection of Domain Generation Algorithms*. 2018; Available from: https://zenodo.org/record/1209901.
3. Barsamian A.V. *Network Characterization for Botnet Detection Using Statistical-Behavioral Methods*. 2009, Dartmouth College.
4. Beigi E.B., Jazi H.H., Stakhanova N., and Ghorbani A.A. *Towards effective feature selection in machine learning-based botnet detection approaches*. in *Communications and Network Security (CNS)*. 2014. IEEE.
5. Belcic I. *Botnet definition: What is a botnet?* 2021 [cited 2021; Available from: https://www.avast.com/c-botnet.
6. Beneš M. *Botnet detection based on network traffic classification*. 2015, Masaryk university.
7. Bin Y., Daniel G., Jie P., Martine C., and Anderson N. *Inline DGA Detection with Deep Networks*. 2017. 683-692.
8. Bin Y., Jie P., Jiaming H., Anderson N., and Martine D.C. *Character Level based Detection of DGA Domain Names*. 2018. 1-8.
9. Brownlee J. *A Gentle Introduction to Ensemble Learning Algorithms*. 2021 [cited 2021; Available from: https://machinelearningmastery.com/tour-of-ensemble-learning-algorithms/.
10. Charan P.V.S. and Anand S.K.S.P.M. *Detecting Word Based DGA Domains Using Ensemble Models*. 2020. Cham: Springer International Publishing.
11. Copyright@2019 Qihoo 360 Technology Co. L. *DGA Families*. 2020 12/26/2020]; Available from: https://data.netlab.360.com/dga/.
12. Cranor C.D., Gansner E., Krishnamurthy B., and Spatscheck O. *Characterizing large DNS traces using graphs*, in *Proceedings of the 1st ACM SIGCOMM Workshop on Internet measurement*. 2001, Association for Computing Machinery: San Francisco, California, USA. p. 55–67.
13. Daniel G., Carles M., and Jordi P. *The rise of machine learning for detection and classification of malware: Research developments, trends and challenges.* Journal of Network and Computer Applications, 2020. **153**: p. 102526.
14. David D., Guofei G., and P L.C. *A Taxonomy of Botnet Structures*, in *Botnet Detection: Countering the Largest Security Threat*, W. Lee, C. Wang, and D. Dagon, Editors. 2008, Springer US: Boston, MA. p. 143-164.
15. Durmaz E. *DGA classification and detection for automated malware analysis*. 2017 [cited 2019; Available from: https://cyber.wtf/2017/08/30/dga-classification-and-detection-for-automated-malware-analysis/.
16. Ebastian Garcia M.G., Jan Stiborek and Alejandro Zunino. *An empirical comparison of botnet detection methods.* Computers and Security Journal, Elsevier, 2014. **45**: p. 100-123.
17. Ghodke S. *Top 1M Alexa*. 2018; Available from: https://www.kaggle.com/datasets/cheedcheed/top1m.

18. Graham M. *BotProbe - botnet traffic capture using IPFIX*, in *BSides*. 2018: London.
19. Gu G., Zhang J., and Lee W. *BotSniffer: Detecting Botnet Command and Control Channels in Network Traffic*. 2008.
20. Guofei G., Phillip P., Vinod Y., Martin F., and Wenke L. *BotHunter: Detecting Malware Infection Through IDSDriven Dialog Correlation.* 2007. **7**: p. 12.
21. Guofei G., Roberto P., Junjie Z., and Wenke L. *BotMiner: Clustering Analysis of Network Traffic for Protocol- and Structure-Independent Botnet Detection*. 2008. 139-154.
22. Hachem N., MustaphaYosra Y.B., Gonzalez B.M., and Debar H. *Botnets: Lifecycle and Taxonomy.* 2011.
23. Hao M. *Botnet Trend Report*. 2020; Available from: https://nsfocusglobal.com/botnet-trend-report-2019/.
24. Hoang D. and Nguyen C. *Botnet Detection Based On Machine Learning Techniques Using DNS Query Data.* Future Internet, 2018. **10**.
25. Holz T., Steiner M., Dahl F., Biersack E., and Freiling F. *Measurements and mitigation of peer-to-peer-based botnets: a case study on storm worm*, in *Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*. 2008, USENIX Association: San Francisco, California. p. Article 9.
26. Hong Z., Zhaobin C., Guangbin B., and Xiangyan Z. *Malicious Domain Names Detection Algorithm Based on n-Gram.* Journal of Computer Networks and Communications, 2019. **2019**: p. 4612474.
27. Hossein Z., Mohammad S., Payam V.A., Safari M., and Zamani M. *A taxonomy of Botnet detection techniques*. Vol. 2. 2010. 158-162.
28. Hu X. and Knysz M. *RB-Seeker: Auto-detection of Redirection Botnets*. Vol. 0. 2009.
29. Huang S.-Y., Mao C.-H., and Lee H.-M. *Fast-flux service network detection based on spatial snapshot mechanism for delay-free detection*. 2010. 101-111.
30. Hyunsang C., Heejo L., and Hyogon K. *BotGAD: Detecting botnets by capturing group activities in network traffic*. 2009. 2.
31. Jaiswal S. *Machine Leaning*. 2019 [cited 2019; Available from: https://www.javatpoint.com/machine-learning.
32. Jérôme F., Shaonan W., Radu S., and Thomas E. *BotTrack: Tracking Botnets Using NetFlow and PageRank*. in *NETWORKING 2011*. 2011. Berlin, Heidelberg: Springer Berlin Heidelberg.
33. Jiang N., Cao J., Jin Y., Li L.E., and Zhang Z.-L. *Identifying suspicious activities through DNS failure graph analysis*, in *Proceedings of the The 18th IEEE International Conference on Network Protocols*. 2010, IEEE Computer Society. p. 144–153.
34. Johannes Bader B.Y. *DGA algorithms*. 2018 [cited 2021; Available from: https://github.com/baderj/domain_generation_algorithms.
35. Jonathan W., H. A., Anjum A., and Daniel G. *Predicting Domain Generation Algorithms with Long Short-Term Memory Networks.* ArXiv, 2016. **abs/1611.00791**.
36. Kamal Alieyan A.A., Ahmad Manasrah & Mohammed M. Kadhum *A survey of botnet detection based on DNS.* Neural Computing and Applications, 2017. **28**.

37. Karim A., Salleh R.B., Shiraz M., Shah S.A.A., Awan I., and Anuar N.B. *Botnet detection techniques: review, future trends, and issues.* Journal of Zhejiang University SCIENCE C, 2014. **15**(11): p. 943-983.
38. Kaspersky. *Bots and botnets in 2018*. [cited 2019 13/11]; Available from: https://securelist.com/bots-and-botnets-in-2018/90091/.
39. Kate H., Domenic P., Song L., and R. J.N. *Real-Time Detection of Dictionary DGA Network Traffic Using Deep Learning.* SN Computer Science, 2021. **2**(2): p. 110.
40. Kheir N., Tran F., Caron P., and Deschamps N. *Mentor: Positive DNS Reputation to Skim-Off Benign Domains in Botnet C&C Blacklists*. in *SEC*. 2014.
41. Koh J. and Rhodes B. *Inline Detection of Domain Generation Algorithms with Context-Sensitive Word Embeddings*. 2018. 2966-2971.
42. Kuochen W., Chun-Ying H., Shang-Jyh L., and R. L.Y. *A fuzzy pattern-based filtering algorithm for botnet detection.* Computer Networks, 2011. **55**: p. 3275-3286.
43. Labs S.M. *Spamhaus Botnet Threat Report 2019*. 2020; Available from: https://www.spamhaus.org/news/article/793/spamhaus-botnet-threatreport-2019.
44. Leyla B., Engin K., Christopher K., and Marco B. *EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis*. 2011.
45. Li X. W.J., and Zhang X. *Botnet Detection Technology Based on DNS.* Journal of Future Internet 2017. **9**.
46. Li Z., Goyal A., Chen Y., and Paxson V. *Automating analysis of large-scale botnet probing events*. 2009. 11-22.
47. Liu J., Xiao Y., Ghaboosi K., Deng H., and Zhang J. *Botnet: Classification, Attacks, Detection, Tracing, and Preventive Measures.* EURASIP J. Wireless Comm. and Networking, 2009. **2009**.
48. Liu L., Chen S., Yan G., and Zhang Z. *BotTracer: Execution-Based Bot-Like Malware Detection*. Vol. 5222. 2008. 97-113.
49. Liu T.-J. and Chen T.-S.L.C.-W. *An Ensemble Machine Learning Botnet Detection Framework Based on Noise Filtering.* Journal of Internet Technology 2021. **22**.
50. Luhui Y., Jiangtao Z., Weiwei L., Xiaopeng J., Huiwen B., Guangjie L., and Yuewei D. *Detecting Word-Based Algorithmically Generated Domains Using Semantic Analysis.* Symmetry, 2019. **11**(2).
51. Luz P.M.d. *Botnet Detection Using Passive DNS,*. 2014. p. 7-8.
52. Ma J., Saul L., Savage S., and Voelker G. *Beyond blacklists: learning to detect malicious Web sites from suspicious URLs*. 2009. 1245-1254.
53. Ma X., Zhang J., Li Z., Li J., Tao J., Guan X., Lui J.C.s., and Towsley D. *Accurate DNS query characteristics estimation via active probing.* Journal of Network and Computer Applications, 2015. **47**.
54. Mac H., Tran D., Tong V., Nguyen G., and Tran H.-A. *DGA Botnet Detection Using Supervised Learning Methods*. 2017. 211-218.
55. Manos Antonakakis R.P., David Dagon, Wenke Lee, Nick Feamster. *Building a dynamic reputation system for dns*. in *USENIX security symposium*. 2011.
56. Marko P. and Vilhan P. *Efficient detection of malicious nodes based on DNS and statistical methods*. 2012. 227-230.
57. Martin Ester H.-P.K., Jiirg Sander, Xiaowei Xu. *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise.* 1996: p. 226-231.

58. Marupally P.R. and Paruchuri V. *Comparative Analysis and Evaluation of Botnet Command and Control Models*. in *2010 24th IEEE International Conference on Advanced Information Networking and Applications*. 2010.
59. Maryam F., Alireza S., and Sureswaran R. *A Survey of Botnet and Botnet Detection*. in *Third International Conference on Emerging Security Information, Systems and Technologies*. 2009. IEEE.
60. Mattia Zago, Manuel Gil Pérez, and Perez G.M. *UMUDGA - University of Murcia Domain Generation Algorithm Dataset*. 2020.
61. Michael B., Evan C., Farnam J., Yunjing X., and Manish K. *A Survey of Botnet Technology and Defenses.* Conference For Homeland Security, Cybersecurity Applications & Technology, 2009. **0**: p. 299-304.
62. Micro T. *Taxonomy of Botnet threats.* TREND MICRO, 2006.
63. Mitchell T.M. *Machine Learning*. 1997: McGraw-Hill Science.
64. Mohssen Mohammed M.B.K., Eihab Bashier Mohammed Bashier. *Machine Learning - Algorithms and Applications*. 2017: Taylor & Francis.
65. Nilaykumar Kiran Sangani H.Z. *Machine Learning in Application Security*, in *Advances in Security in Computing and Communications*. 2017.
66. Paxton N., Ahn G., and Chu B. *Towards Practical Framework for Collecting and Analyzing Network-Centric Attacks*. in *2007 IEEE International Conference on Information Reuse and Integration*. 2007.
67. PentaSecurity. *Top 5 Botnets of 2017*. 2018 [cited 2019 1/9]; Available from: https://www.pentasecurity.com/blog/top-5-botnets-2017/.
68. Perdisci R., Corona I., Dagon D., and Lee W. *Detecting Malicious Flux Service Networks through Passive Analysis of Recursive DNS Traces*. 2009. 311-320.
69. Pereira M., Coleman S., Yu B., DeCock M., and Nascimento A. *Dictionary Extraction and Detection of Algorithmically Generated Domain Names in Passive DNS Traffic: 21st International Symposium, RAID 2018, Heraklion, Crete, Greece, September 10-12, 2018, Proceedings*. 2018. p. 295-314.
70. Qiao Y., Zhang B., Zhang W., Sangaiah A.K., and Wu H. *DGA Domain Name Classification Method Based on Long Short-Term Memory with Attention Mechanism.* Applied Sciences, 2019. **9**(20): p. 4205.
71. R. Z.H. and A. M.A. *Botnet Command and Control Mechanisms*. in *Second International Conference on Computer and Electrical Engineering*. 2009. IEEE.
72. Raghava N.S., Sahgal D., and Chandna S. *Classification of Botnet Detection Based on Botnet Architechture*. in *2012 International Conference on Communication Systems and Network Technologies*. 2012.
73. Rahim A. and bin Muhaya F.T. *Discovering the Botnet Detection Techniques*. 2010. Berlin, Heidelberg: Springer Berlin Heidelberg.
74. Rajab M.A., Zarfoss J., Monrose F., and Terzis A. *A multifaceted approach to understanding the botnet phenomenon*, in *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*. 2006, Association for Computing Machinery: Rio de Janeriro, Brazil. p. 41–52.
75. Ramachandran A., Feamster N., and Dagon D. *Revealing botnet membership using DNSBL counter-intelligence.* Proceedings of the 2nd Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI), 2006. **2**: p. 8-8.
76. Rezaei A. *Using Ensemble Learning Technique for Detecting Botnet on IoT.* SN Computer Science, 2021.

77. Sanchez F., Duan Z., and Dong Y. *Blocking spam by separating end-user machines from legitimate mail server machines*. Vol. 9. 2011. 116-124.
78. Sangani N.K., Zarger, H. *Machine Learning in Application Security*, in *Advances in Security in Computing and Communications*. 2017, IntechOpen.
79. Satoh A., Fukuda Y., Kitagata G., and Nakamura Y. *A Word-Level Analytical Approach for Identifying Malicious Domain Names Caused by Dictionary-Based DGA Malware.* Electronics, 2021. **10**(9): p. 1039.
80. Saxe J. and Berlin K. *eXpose: A Character-Level Convolutional Neural Network with Embeddings For Detecting Malicious URLs, File Paths and Registry Keys.* 2017.
81. Sebastian Garcia M.G., Jan Stiborek and Alejandro Zunino. *An empirical comparison of botnet detection methods.* Computers and Security Journal, Elsevier, 2014. **45**.
82. Sergio S., Rodrigo S., Raquel P., and Ronaldo S. *Botnets: A survey.* Computer Networks, 2013. **57**: p. 378–403.
83. Seungwon S., Zhaoyan X., and Guofei G. *EFFORT: Efficient and effective bot malware detection*. in *2012 Proceedings IEEE INFOCOM*. 2012.
84. Shaofang Z., Lanfen L., Junkun Y., Feng W., Zhaoting L., and Jia C. *CNN-based DGA Detection with High Coverage*. in *International Conference on Intelligence and Security Informatics (ISI)*. 2019.
85. Shin S., Zhaoyan X., and Guofei G. *EFFORT: A new host-network cooperated framework for efficient and effective bot malware detection.* Computer Networks: The International Journal of Computer and Telecommunications Networking, 2013. **57**: p. 2628-2642.
86. Smith D. *More Destructive Botnets and Attack Vectors Are on Their Way*. Radware Blog 2019; Available from: https://blog.radware.com/security/botnets/2019/10/scan-exploit-control/.
87. Stalmans E. *A framework for DNS based detection and mitigation of malware infections on a network.* 2011 Information Security for South Africa, 2011: p. 1-8.
88. Stevanovic M. and Pedersen J.M. *Machine learning for identifying botnet network traffic*. 2013. IEEE.
89. Stinson E. and Mitchell J.C. *Characterizing Bots' Remote Control Behavior*. 2007. Berlin, Heidelberg: Springer Berlin Heidelberg.
90. Symantic. *Botnets now produce 95% of spam*. 2010; Available from: https://www.bizjournals.com/sanjose/stories/2010/08/23/daily29.html.
91. TalkEnglish. *Top 1500 English Nouns.* [cited 2021; Available from: https://www.talkenglish.com/vocabulary/top-1500-nouns.aspx.
92. Tegeler F., Fu X., Vigna G., and Krügel C. *BotFinder: finding bots in network traffic without deep packet inspection*. in *CoNEXT '12*. 2012.
93. Tiep V.H. *Machine Learning*. 2016-2020.
94. Tran D., Mac H., Tong V., Tran H.-A., and Nguyen G. *A LSTM based Framework for Handling Multiclass Imbalance in DGA Botnet Detection.* Neurocomputing, 2017. **275**.
95. Tronk M. *English dictionary - 58 000 English words.* [cited 2020; Available from: http://www.mieliestronk.com/wordlist.html

96. Truong D.T. and Cheng G. *Detecting domain-flux botnet based on DNS traffic features in managed network.* Security and Communication Networks, 2016. **9**(14): p. 2338-2347.
97. Umbrella C. *Umbrella Popularity List*. 2016; Available from: http://s3-us-west-1.amazonaws.com/umbrella-static/index.html.
98. Villamarin R. and Brustoloni J. *Identifying Botnets Using Anomaly Detection Techniques Applied to DNS Traffic*. 2008. 476-481.
99. Wang B., Li Z., Li D., Liu F., and Chen H. *Modeling Connections Behavior for Web-Based Bots Detection*. in *2010 2nd International Conference on E-business and Information System Security*. 2010.
100. Wielogorska M.a.O.B., Darragh. *DNS Traffic analysis for botnet detection*. in *25th Irish Conference on Artificial Intelligence and Cognitive Science*. 2017. CEUR-WS.
101. Xiang Z., Junbo Z., and Yann L. *Character-level Convolutional Networks for Text Classification*. in *the 28th International Conference on Neural Information Processing Systems*. 2015. MIT Press.
102. Yadav S., Reddy A., Reddy A., and Ranjan S. *Detecting Algorithmically Generated Malicious Domain Names*. 2010. 48-61.
103. Yong-lin Zhou Q.-s.L., Qidi Miao and Kangbin Yim. *DGA-Based Botnet Detection Using DNS Traffic.* Journal of Internet Services and Information Security (JISIS), 2013. **3**: p. 116-123.
104. Zahraa A., Eman A., Dalia A.-W., and Radhwan H.A.A.-S. *Botnet detection using ensemble classifiers of network flow.* International Journal of Electrical and Computer Engineering, 2020. **Volume 10**: p. 2543-2550.
105. Zhaosheng Z., Guohan L., Yan C., Zhi F., Phil R., and Keesook H. *Botnet Research Survey*. 2008. 967-972.
106. Zhou Z.-H. *Ensemble Methods*. 2012: CRC Press, Taylor & Francis Group, LLC.