# Phương pháp cải tiến LSTM dựa trên đặc trưng thống kê trong phát hiện DGA botnet

Mạc Đình Hiếu, Tống Văn Vạn, Bùi Trọng Tùng, Trần Quang Đức, Nguyễn Linh Giang
Trường Đại học Bách khoa Hà Nội

Tác giả liên hệ: Nguyễn Linh Giang, giangnl@soict.hust.edu.vn
Ngày nhận bài: 15/07/2017, ngày sửa chữa: 15/12/2017, ngày duyệt đăng: 25/12/2018
Xem sớm trực tuyến: 28/12/2018, định danh DOI: 10.32913/rd-ict.vol3.no40.528
Biên tập lĩnh vực điều phối phản biện và quyết định nhận đăng: PGS. TS. Nguyễn Nam Hoàng

**Tóm tắt:** Phần lớn botnet sử dụng cơ chế sinh tên miền tự động (DGA: Domain Generation Algorithms) để kết nối và nhận lệnh từ máy chủ điều khiển. Việc tìm ra dạng DGA botnet thực hiện qua xác định cách thức tạo sinh tên miền đặc trưng cho loại botnet đó dựa trên những phân tích đặc trưng tên miền thu thập từ các truy vấn DNS. Trong bài báo này chúng tôi đề xuất phương pháp phân tích tên miền và phát hiện DGA botnet dựa trên sự kết hợp mạng LSTM (Long Short-Term Memory) với các đặc trưng thống kê như độ dài, entropy, mức độ ý nghĩa của tên miền nhằm tăng khả năng khái quát hóa cho mạng LSTM. Phương pháp đề xuất được thử nghiệm và đánh giá trên bộ dữ liệu tên miền thu thập trong thực tế bao gồm một triệu tên miền Alexa và hơn 750 nghìn tên miền được sinh bởi 37 loại DGA botnet. Kết quả thử nghiệm đã chứng minh tính hiệu quả của phương pháp đề xuất trong cả hai trường hợp phân loại hai lớp và phân loại đa lớp, với giá trị macro-averaging F1-score cao hơn 5% và nhận biết thêm được 3 loại DGA so với phương pháp phát hiện DGA botnet dựa trên mạng LSTM truyền thống.

**Từ khóa:** *Phát hiện DGA botnet, LSTM, phát hiện tấn công mạng, an ninh mạng.*

**Title:** **A Method to Improve LSTM using Statistical Features for DGA Botnet Detection**

**Abstract:** Recently, botnets have been the main mean for phishing, spamming, and launching Distributed Denial of Service attacks. Most bots today use Domain Generation Algorithms (DGA) (also known as domain fluxing) to construct a resilient Command and Control (C&C) infrastructure. Reverse Engineering has become the prominent approach to combat botnets. It however needs a malware sample that is not always possible in practice. This paper presents an extended version of the Long Short-Term Memory (LSTM) network, where the original algorithm is coupled with other statistical features, namely meaningful character ratio, entropy, and length of the domain names to further improve its generalization capability. Experiments are carried out on a real-world collected dataset that contains one non-DGA and 37 DGA malware families. They demonstrated that the new method is able to work on both binary and multi-class tasks. It also produces at least 5% macro-averaging F1-score improvement as compared to other state-of-the-art detection techniques while helping to recognize 3 additional DGA families.

**Keywords:** *DGA Botnet, NXDomain, Recurrent Neural Network, Long Short-Term Memory Network.*

## I. GIỚI THIỆU

Botnet là một mạng máy tính trong đó mỗi máy tính trong mạng bị lây nhiễm mã độc và được coi là một bot [1]. Phần lớn botnet ngày nay đều được xây dựng trên cơ sở cơ chế sinh tên miền tự động (DGA: Domain Generation Algorithms), trong đó bot tự động sinh ra một số lượng lớn tên miền và sử dụng một tập con để kết nối với máy chủ điều khiển (C&C: Command and Control). Điểm mạnh của DGA là nếu địa chỉ của C&C bị phát hiện và chặn tất cả kết nối đến địa chỉ này, mạng botnet không hoàn toàn bị loại bỏ [1–3]. Khi đó, bot vẫn có thể nhận lệnh điều khiển thông qua việc ánh xạ địa chỉ IP với một tập tên miền mới được sinh ra. Cách phát hiện botnet truyền thống là sử dụng kỹ thuật dịch ngược mã nguồn. Tuy nhiên quá trình dịch ngược đòi hỏi nhiều thời gian, công sức, trong khi danh sách các địa chỉ phải được cập nhật một cách thường xuyên.

Davuth và Kim trong công trình [2] đã đề xuất cơ chế phân loại tên miền sử dụng đặc trưng bi-gram và các thuật toán học máy vector hỗ trợ (SVM: Support Vector Machines). Kwon và cộng sự trong công trình [3] đã đề xuất PsyBoG, một cơ chế phát hiện DGA botnet dựa vào biểu hiện, các đặc trưng thu được từ người dùng từ lưu lượng DNS và cho phép triển khai trong môi trường dữ

liệu lớn. Grill và các cộng sự trong công trình [4] đã đề xuất một phương pháp phát hiện DGA botnet dựa vào số lượng truy vấn DNS, địa chỉ IP và khoảng thời gian truy vấn. Mowbray và Hagen trong công trình [5] đã đề xuất cơ chế phát hiện DGA botnet dựa vào phân phối về độ dài của tên miền. Schiavoni và các cộng sự trong công trình [6] đã sử dụng khoảng cách Mahalanobis. Tuy nhiên, các phương pháp trên lại khó triển khai thời gian thực và khó có thể tích hợp vào các hệ thống phát hiện DGA botnet thực tế.

Antonakakis và các cộng sự trong công trình [7] đã trích rút ra các đặc trưng của tên miền, sau đó sử dụng mô hình Markov ẩn (HMM: Hidden Markov Model) để phân loại thành tên miền DGA và non-DGA. Perdisci và các cộng sự trong công trình [8] đã sử dụng các đặc trưng trích rút từ lưu lượng DNS và áp dụng thuật toán học máy C4.5 để phân loại. Woodbridge và các cộng sự trong công trình [9] đã đề xuất cơ chế phát hiện DGA botnet sử dụng mạng LSTM (Long Short-Term Memory network), tuy nhiên do chỉ sử dụng tên miền nên hiệu quả phát hiện không cao.

Trong các công trình [1, 10, 11], nhóm tác giả cũng đề xuất phương pháp phát hiện DGA botnet sử dụng đặc trưng ngữ nghĩa, đặc trưng thống kê, áp dụng các phương pháp phân cụm dựa trên mật độ DBSCAN, lọc cộng tác (collaborative filtering) và map-reduce để tính độ tương hợp giữa các hành vi của các máy trạm, K-means và biến thể của khoảng cách Mahalanobis. Tuy nhiên những cách tiếp cận này này thường chỉ hiệu quả với một hoặc một số kiểu DGA botnet nhất định.

Những công trình nghiên cứu về DGA nói trên phần lớn đều tập trung vào một hoặc một vài đặc trưng trích rút ra từ tên miền. Bên cạnh đó, việc thử nghiệm cũng chỉ được tiến hành trên tập dữ liệu nhỏ nên rất khó đánh giá chính xác khả năng phát hiện botnet của hệ thống. Từ những nhận xét trên, chúng tôi đề xuất phương pháp mới sử dụng kết hợp mạng LSTM với các đặc trưng thống kê. Phương pháp này về cơ bản đã được thay đổi và cải tiến phương pháp do Woodbridge đề xuất trong [9]. Phương pháp đề xuất sử dụng mạng LSTM nhằm trích rút ra các đặc trưng nội hàm của tên miền và các đặc trưng này sẽ được sử dụng để tạo ra vec-tơ đặc trưng đại diện cho tên miền. Các đặc trưng nội hàm được trích rút sử dụng mạng LSTM nên sẽ giúp cho quá trình phát hiện DGA botnet hiệu quả hơn, điều này đã được chứng minh dựa vào kết quả thực nghiệm trong mục IV của bài báo này.

Những đóng góp trong bài báo được được trình bày cụ thể như sau. Thứ nhất là chúng tôi sử dụng thêm các đặc trưng thống kê từ tên miền đầu vào. Các công trình nghiên cứu [6, 9–12] đều đề cập các đặc trưng này và đã chứng minh tính hiệu quả của chúng trong phát hiện một số dạng DGA botnet nhất định. Vì vậy, đặc trưng thống kê có thể được sử dụng kết hợp nhằm nâng cao tỷ lệ phát hiện đúng của mạng LSTM truyền thống. Thứ hai là chúng tôi sử dụng bộ dữ liệu gồm 37 mẫu DGA được thu thập từ một tổ chức an ninh mạng uy tín [13]. Số lượng tên miền của mỗi dạng mã độc khác nhau phản ánh đúng mức độ xuất hiện của chúng trong thực tế. Qua thực nghiệm, chúng tôi thấy rằng phương pháp đề xuất cho kết quả tốt hơn so với phương pháp gốc của Woodbridge với mức tăng macro-averaging F1-score khoảng 5%. Nó cũng cho phép phát hiện thêm 3 mẫu mã độc mới mà mạng LSTM bình thường không tìm ra. Thông thường hệ thống phát hiện DGA botnet dựa trên tên miền gồm ba pha chính: pha tiền xử lí để trích rút ra tên miền và các đặc trưng, pha thuật toán phát hiện và pha cảnh báo. Phương pháp của chúng tôi có thể được triển khai tại pha thứ hai. Với thời gian trích rút các đặc trưng thống kê tương đối nhỏ, nó hoàn toàn phù hợp để xây dựng hệ thống IDPS (Intrusion Detection and Protection System) với độ chính xác cao và hỗ trợ phát hiện DGA thời gian thực.

Bài báo gồm ba mục chính sau. Mục III trình bày về các kiến thức cơ sở mạng LSTM và một số đặc trưng thống kê của tên miền sử dụng trong bài báo. Trong mục III, chúng tôi đề xuất phương pháp kết hợp LSTM truyền thống với các đặc trưng như độ dài, entropy và mức độ ý nghĩa của tên miền trong phát hiện DGA botnet. Mục IV trình bày về các kết quả thử nghiệm, nhận xét và đánh giá.

## II. KIẾN THỨC CƠ SỞ

### 1. Mạng LSTM

Mạng LSTM [14–17] là một dạng mạng nơ-ron hồi quy (RNN: Recurrent Neural Network), thường được sử dụng trong các bài toán xác định quan hệ giữa các thành phần của một chuỗi thời gian. Đối với RNN, đầu ra tại một lớp không chỉ dựa vào đầu vào ở thời điểm hiện tại mà còn phụ thuộc vào đầu ra ở thời điểm trong quá khứ. RNN cho phép lưu trữ trạng thái của các nút mạng, do đó chuỗi thao tác của nó có thể khá lớn, dẫn đến kết quả đầu ra có thể bị suy giảm theo hàm mũ. Mạng LSTM được đề xuất và đưa ra nhằm giải quyết vấn đề này của RNN.

Mạng LSTM thường có ba dạng là mạng LSTM truyền thống (Traditional LSTM Network), mạng LSTM khe hẹp (Peephole LSTM Network) và mạng LSTM tích chập (Convolutional LSTM Network), tuy nhiên trong phạm vi của bài báo này, chúng tôi chỉ sử dụng và trình bày về cơ chế của mạng LSTM truyền thống. Cấu trúc LSTM được mô tả trong hình 1 với các tham số:

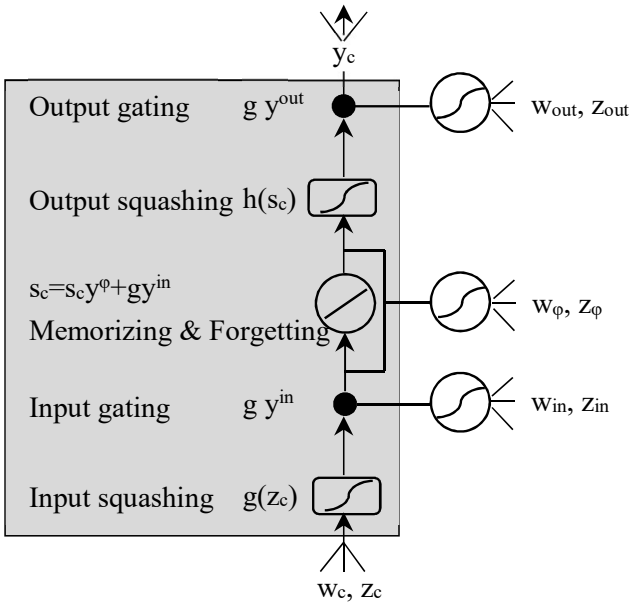
$$y^{\varphi} = \sigma_g(W_{\varphi} z_{\varphi} + U_{\varphi} y^{c} + b_{\varphi}), \quad (1)$$

$$y^{\text{in}} = \sigma_g(W_{\text{in}} z_{\text{in}} + U_{\text{in}} y^{c} + b_{\text{in}}), \quad (2)$$

$$y^{\text{out}} = \sigma_g(W_{\text{out}} z_{\text{out}} + U_{\text{out}} y^{c} + b_{\text{out}}), \quad (3)$$

$$s_c = s_c y^{q} + y^{\text{in}} \sigma_c(W_s z_c + U_s y^{c} + b_s), \quad (4)$$

$$y^{c} = y^{\text{out}} \sigma_c(s_c). \quad (5)$$




Hình 1. Cấu trúc một ô nhớ của mạng LSTM truyền thống.

Tại một thời điểm, với vector đầu vào $z_{\text{in}}$, $z_{\text{out}}$ và $z_\varphi$ sau khi qua Input Gate, Output Gate và Forget Gate sẽ thu được đầu ra là các vector $y^{\text{in}}$, $y^{\text{out}}$ và $y^\varphi$. Các công thức (1), (2) và (3) thể hiện quá trình biến đổi từ đầu vào thành đầu ra ở các cổng, trong đó $W$, $U$ và $b$ là các ma trận và vector tham số. Vector trạng thái $s_c$ được tính theo công thức (4), sau đó vector đầu ra $y^c$ được tính theo công thức (5). Các hàm $\sigma_g$ và $\sigma_c$ trong các biểu thức trên lần lượt là các hàm sigmoid và hàm hyperbolic tanh [18, 19].

## 2. Các đặc trưng thống kê

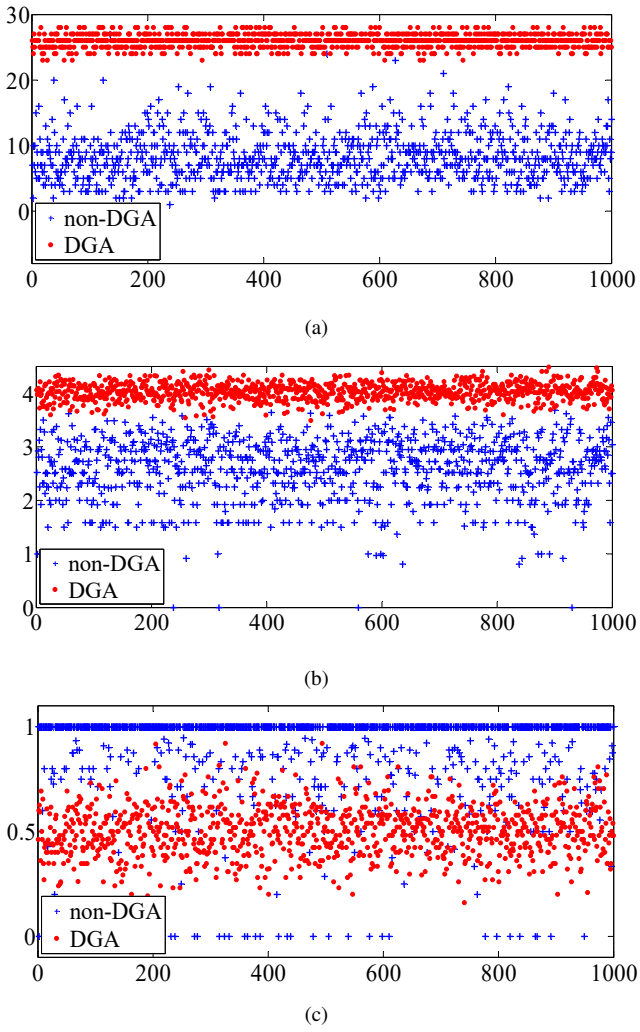
Trong bài báo, chúng tôi sử dụng thêm với mạng LSTM ba đặc trưng gồm độ dài, entropy và mức độ ý nghĩa của tên miền. Hình 2 thể hiện giá trị của các đặc trưng được tính toán từ 1.000 mẫu thuộc hai lớp *Alexa* (non-DGA) và *PT Goz* (DGA).

Độ dài là số kí tự trong tên miền đó. Tên miền do DGA botnet sinh ra thường có độ dài lớn hơn so với tên miền bình thường. Từ hình 2(a), ta thấy độ dài của tên miền bình thường nằm trong khoảng từ 5 đến 15 ký tự và thường khác biệt so với tên miền DGA (lớn hơn hoặc bằng 20 ký tự).

Entropy xác định độ bất định của một tên miền. Với tên miền $d$, entropy $E(d)$ được cho bởi

$$E(d) = -\sum_{i=1}^{|p|} \frac{\text{index}(t)}{N} \log\left(\frac{\text{index}(t)}{N}\right), \qquad (6)$$

với index($t$) là số lượng của kí tự $t$ trong tập tên miền, $|p|$ là số lượng ký tự phân biệt trong tên miền và $N$ là số ký tự của tập tên miền. Hình 2(b) cho thấy sự khác nhau giữa




Hình 2. Khả năng phân biệt tên miền bình thường và tên miền DGA của các đặc trưng (a) độ dài, (b) entropy và (c) mức độ ý nghĩa.

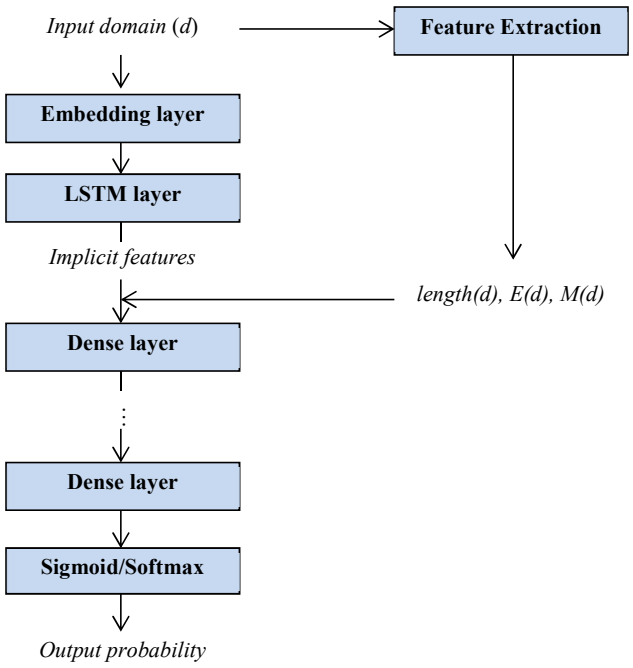
entropy của tên miền bình thường và tên miền do DGA sinh ra. Tên miền bình thường có dải entropy khá rộng từ 1,5 đến 3,4, còn đối với một mẫu DGA, entropy thường lớn hơn và có dải phân bố hẹp từ 3,7 đến 4,3.

Mức độ ý nghĩa của tên miền đặc trưng cho mức độ có ý nghĩa của các cụm $n$-gram [20] có trong tên miền. Tên miền được chia thành các cụm $w(i)$ có độ dài lớn hơn hoặc bằng 3. Với tên miền $d$, mức độ ý nghĩa $R$ được cho bởi

$$M(d) = \frac{\sum_{i=1}^{n} \text{len}(w(i))}{p}, \qquad (7)$$

trong đó $p$ là độ dài của tên miền $d$, $n$ là số từ có ý nghĩa trong tên miền. Ví dụ, đối với chuỗi kí tự *"stackoverflow"*, mức độ ý nghĩa $R$ được tính là

$$M(d) = \frac{\text{len}(|stack|) + \text{len}(|over|) + \text{len}(|flow|)}{13} = 1.$$




Hình 3. Sơ đồ phương pháp phát hiện DGA botnet sử dụng mạng LSTM truyền thống kết hợp với các đặc trưng thống kê.

Hình 2(c) minh họa sự khác nhau giữa mức độ ý nghĩa của tên miền bình thường và tên miền do DGA sinh ra. Đối với tên miền bình thường $M$ thường nằm khoảng 0,8 đến 1. Tên miền DGA thường có mức độ ý nghĩa nhỏ hơn do các ký tự được ghép một cách ngẫu nhiên theo hàm mật độ phân bố đều.

Qua thực nghiệm chúng tôi nhận thấy các đặc trưng thống kê hỗ trợ khá tốt cho quá trình phân loại tên miền. Đây là những đặc trưng độc lập với những đặc trưng nội hàm được trích chọn trong quá trình huấn luyện mạng LSTM.

## III. PHƯƠNG PHÁP PHÁT HIỆN DGA BOTNET ĐỀ XUẤT

Phần này đề xuất phương pháp phát hiện DGA botnet bằng mạng LSTM kết hợp với các đặc trưng thống kê của tên miền. Sơ đồ chung của phương pháp được mô tả trong hình 3. Những đặc trưng thống kê của tên miền được xác định trong mô-đun *Feature Extraction*.

Đối với LSTM, chuỗi tên miền đầu vào trước hết được chuẩn hóa về dạng số với giá trị 0 được bổ sung để đảm bảo chúng có cùng độ dài $l$. Tại tầng Embedding, tên miền sẽ được biến đổi thành tập vector $V^{d\times l}$ với $d = 128$ là tham số đại diện cho mạng LSTM. Giá trị của tham số $d$ được xác định dựa trên thực nghiệm và chúng tôi nhận thấy rằng việc tăng giá trị của $d$ không làm ảnh hưởng quá nhiều đến kết quả đầu ra, nhưng lại làm tăng khối lượng tính toán. Trong bài báo này, chúng tôi sử dụng mạng LSTM với 128 ô nhớ. Cấu trúc mỗi ô nhớ được biểu diễn như hình 1. LSTM đóng vai trò quan trọng để trích chọn ra các đặc trưng nội hàm được biểu diễn dưới dạng vector đặc tả mối liên hệ giữa các ký tự trong một tên miền. Đặc trưng nội hàm tương tự như $n$-gram được đề cập tại rất nhiều công trình, chẳng hạn [20], và cho kết quả phân loại tốt hơn mà không yêu cầu nhiều thời gian tính toán, trích chọn và xử lý [9].

Trong mô hình đề xuất, đặc trưng nội hàm kết hợp với đặc trưng thống kê được đưa qua tầng nén (dense layer) để làm mượt và tăng độ chính xác. Tầng nén là tầng kết nối đầy đủ (fully connected layer), trong đó mỗi nơ-ron kết nối đến từng nơ-ron trong tầng trước đó với trọng số xác định. Đầu ra của tầng này đi qua hàm kích hoạt (activation function) để chuẩn hóa về đoạn [0, 1]. Số lượng giá trị xác suất ở đầu ra phụ thuộc vào từng kiểu phân loại.

Khi phân loại hai phân lớp, đầu ra là xác suất đánh giá khả năng xuất hiện của tên miền. Tên miền được phân loại là non-DGA nếu xác suất lớn hơn 0,5, ngược lại sẽ là DGA. Hàm kích hoạt trong trường hợp này là hàm *Sigmoid* [18], được cho bởi

$$\sigma(z_{\text{out}}) = \frac{1}{1 + e^{-z_{\text{out}}}}. \tag{8}$$

Khi phân loại đa phân lớp, số lượng giá trị xác suất đầu ra từ tầng nén ứng với mỗi tên miền bằng với số phân lớp trong tập dữ liệu huấn luyện. Hàm kích hoạt trong trường hợp này là hàm *softmax* (hàm mũ chuẩn hóa) [19], biến đổi vector $K$ phần tử $z_{\text{out}}$ thành một vector $K$ phần tử $\sigma(z_{\text{out}_i})$ trong khoảng [0, 1], được cho bởi

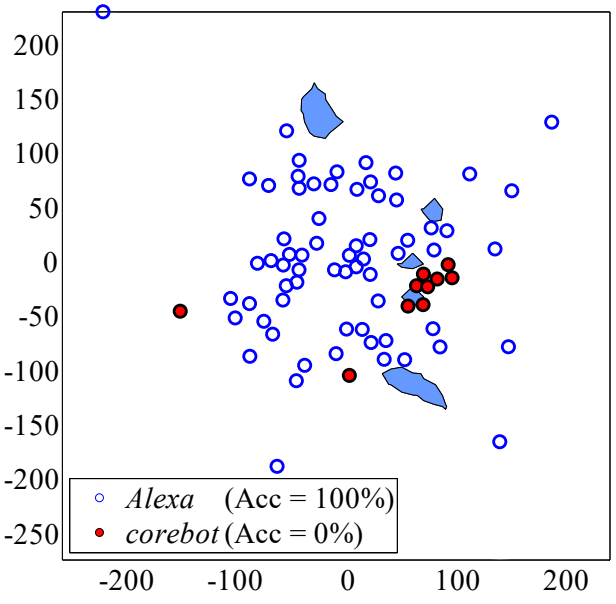
$$\sigma(z_{\text{out}_i}) = \frac{e^{z_{\text{out}_i}}}{\sum_{j=1}^{K} e^{z_{\text{out}_j}}}. \tag{9}$$

Tên miền sẽ được phân vào lớp ứng với giá trị xác suất (giá trị hàm *softmax* tương ứng) cao nhất. Trong đề xuất này, số lượng tầng nén được lựa chọn bằng 3 dựa vào các kết quả thực nghiệm. Khả năng phát hiện của phương pháp sử dụng mạng LSTM truyền thống và phương pháp đề xuất thể hiện trong hình 4. Hình tròn xanh đại diện cho các tên miền bình thường, trong khi hình tròn đỏ là các tên miền do DGA (*corebot*) sinh ra. Việc biểu diễn được thực hiện thông qua công cụ t-SNE [21]. Ta thấy rằng tỉ lệ phát hiện tên miền *Alexa* (non-DGA) của cả hai phương pháp đều là 100% ứng với những dữ liệu thử nghiệm. Tuy nhiên, đối với tên miền DGA, phương pháp sử dụng mạng LSTM truyền thống không phát hiện được do những điểm màu đỏ nằm ngoài vùng xanh, còn phương pháp do chúng tôi đề xuất có thể tỷ lệ phát hiện khoảng 66,7%.

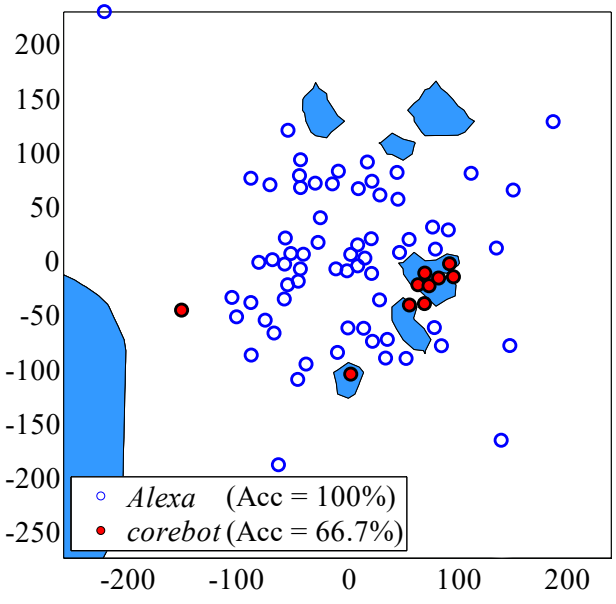
## IV. THỬ NGHIỆM VÀ ĐÁNH GIÁ

### 1. Môi trường và dữ liệu thử nghiệm

Các phương pháp phát hiện DGA botnet được chúng tôi thử nghiệm trên máy tính cài hệ điều hành Ubuntu 16.04,




(a)




(b)

Hình 4. Tỷ lệ phát hiện của hai phương pháp (a) LSTM truyền thống và (b) LSTM kết hợp với đặc trưng thống kê. Vùng xanh thể hiện vùng thuộc tên miền DGA.

Core i5 4235, 8 GB RAM. Tập dữ liệu được tổng hợp từ hai nguồn: bộ dữ liệu gồm một triệu tên miền có thứ hạng cao của Alexa [22] và bộ dữ liệu OSINT DGA feed của Bambenek Consulting, một tổ chức chuyên điều tra về an ninh mạng và tội phạm mạng [13].

OSINT DGA feed gồm 37 loại DGA, với hơn 750.000 tên miền tất cả. Tuy nhiên do khả năng hạn chế về cấu hình máy tính, chúng tôi lựa chọn ngẫu nhiên 88.357 tên miền

Bảng I
SỐ LƯỢNG TÊN MIỀN TRONG TẬP DỮ LIỆU 37 LOẠI DGA BOTNET VÀ TOP MỘT TRIỆU TÊN MIỀN ALEXA

| DGA | Số lượng | Ý nghĩa | DGA | Số lượng | Ý nghĩa |
|---|---|---|---|---|---|
| Gedo | 58 | ✗ | Fobber | 60 | ✗ |
| Beebone | 42 | ✓ | Alexa | 88347 | ✓ |
| Murofet | 816 | ✗ | Dyre | 800 | ✗ |
| Pykspa | 1422 | ✗ | Cryptowall | 94 | ✓ |
| Padcrypt | 58 | ✗ | Corebot | 28 | ✗ |
| Ramnit | 9158 | ✗ | P | 200 | ✗ |
| Volatile | 50 | ✓ | Bedep | 172 | ✗ |
| Ranbyus | 1232 | ✗ | Matsnu | 48 | ✓ |
| Qakbot | 4000 | ✗ | PT Goz | 6600 | ✗ |
| Simda | 1365 | ✗ | Necurs | 2398 | ✗ |
| Ramdo | 200 | ✗ | Pushdo | 168 | ✗ |
| Suppobox | 101 | ✓ | Cryptolocker | 600 | ✗ |
| Locky | 186 | ✗ | Dircypt | 57 | ✗ |
| Tempedreve | 25 | ✗ | Shifu | 234 | ✗ |
| Qadars | 40 | ✗ | Bamital | 60 | ✗ |
| Symmi | 64 | ✗ | Kraken | 508 | ✗ |
| Banjori | 42166 | ✗ | Nymaim | 600 | ✗ |
| Tinba | 6385 | ✗ | Shiotob | 1253 | ✗ |
| Hesperbot | 192 | ✗ | W32.Virut | 60 | ✗ |

Bảng II
CÁCH XÁC ĐỊNH CÁC THAM SỐ TP, FP, TN, FN

| | | Predicted condition | |
|---|---|---|---|
| | | Prediction positive | Prediction negative |
| True Condition | Condition positive | True Positive (TP) | False Negative (FN) |
| | Condition negative | False Positive (TP) | True Negative (TN) |

bình thường và 81.490 tên miền DGA. Trong bảng I là số lượng tên miền của các mẫu DGA botnet trong tập dữ liệu thử nghiệm cùng với thuộc tính về ý nghĩa.

## 2. Các tham số đánh giá

Trong bài báo, chúng tôi sử dụng các độ đo Precision, Recall, F1-score để đánh giá hiệu năng của các phương pháp. Các độ đo này được xác định qua True Positive (TP), False Positive (FP), False Negative (FN) và True Negative (TN) như trong bảng II. Precision (P) là tỉ lệ giữa số tên miền phân loại chính xác trên tổng số tên miền được dự đoán của mỗi lớp, được cho bởi

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}. \tag{10}$$

Bảng III
KẾT QUẢ PRECISION, RECALL VÀ F1-SCORE CHO TRƯỜNG HỢP HAI LỚP

| Mẫu DGA botnet | Precision | | | | Recall | | | | F1-score | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | HMM [23] | Features-C5.0 [20] | LSTM [9] | Proposed method | HMM [23] | Features-C5.0 [20] | LSTM [9] | Proposed method | HMM [23] | Features-C5.0 [20] | LSTM [9] | Proposed method |
| DGA | 0,846 | 0,965 | 0,978 | 0,979 | 0,788 | 0,961 | 0,987 | 0,987 | 0,816 | 0,963 | 0,983 | 0,983 |
| Non-DGA | 0,786 | 0,964 | 0,985 | 0,986 | 0,844 | 0,968 | 0,976 | 0,977 | 0,814 | 0,966 | 0,981 | 0,982 |
| **Micro-averaging** | 0,817 | 0,965 | **0,982** | **0,982** | 0,815 | 0,965 | **0,982** | **0,982** | 0,815 | 0,96 | 0,982 | **0,982** |
| **Macro-averaging** | 0,815 | 0,965 | 0,982 | **0,9825** | 0,816 | 0,964 | 0,9815 | **0,982** | 0,816 | 0,964 | 0,982 | **0,9825** |

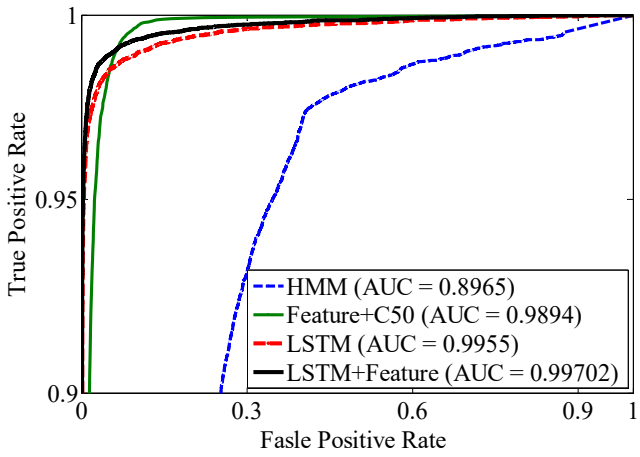
Recall (R) là tỉ lệ giữa số tên miền được phân loại chính xác theo một nhãn nhất định trên tổng số tên miền được gán theo nhãn đó, được cho bởi

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}. \tag{11}$$

Độ đo F1 là trung bình điều hòa [24] giữa hai giá trị Precision và Recall, được cho bởi

$$\mathrm{F1} = \frac{2\mathrm{PR}}{\mathrm{P} + \mathrm{R}}. \tag{12}$$

Thông thường, ta có thể tính Precision, Recall và F1-score trong hai trường hợp: Micro-averaging và Macro-averaging. Micro-averaging ước lượng các thang đo dựa trên tỷ lệ TP, FP và FN tích lũy, trong khi macro-averaging giả định các lớp dữ liệu có cùng mức độ quan trọng nên Precision, Recall và F1-score được tính bằng trung bình của các thang đo tại từng lớp.




Hình 5. Đường cong ROC trong trường hợp hai lớp ứng với bốn phương pháp.

## 3. Các phương pháp đánh giá

Trong thực nghiệm, chúng tôi tiến hành so sánh phương pháp đề xuất với phương pháp LSTM truyền thống [9], phương pháp sử dụng HMM [7] và C5.0 (phương pháp cải tiến của C4.5). Tương tự như LSTM, HMM có thể trích rút đặc trưng nội hàm trực tiếp từ một tên miền. Mỗi một mô hình sẽ tương ứng với một lớp dữ liệu. Trong [7], HMM đã được tác giả áp dụng với 4 loại DGA bao gồm *conficker*, *murofet*, *bobax* và *sinowal*. Trong bài báo này, chúng tôi thử nghiệm HMM trên tập dữ liệu lớn hơn gồm 37 loại DGA. Lưu ý rằng, *conficker*, *murofet*, *bobax* và *sinowal* sử dụng hàm mật độ phân bố đều của các ký tự để tạo ra tên miền, tương tự như *ramnit*. C5.0 được xây dựng dựa trên 8 đặc trưng ngữ nghĩa và thống kê, trong đó các đặc trưng ngữ nghĩa chúng tôi sử dụng là *n*-gram với. Tất cả các phương pháp được thử nghiệm trên tập dữ liệu trình bày ở trên bằng kiểm nghiệm chéo 5 lần.

## 4. Phân loại hai lớp

Phân loại hai lớp phân biệt tên miền DGA và tên miền non-DGA. Kết quả thử nghiệm của các phương pháp trên được thể hiện trong bảng IV với các tham số Precision, Recall và F1-score. Từ bảng III, phương pháp sử dụng HMM kết quả thu được khá khiêm tốn, giá trị của ba tham số chỉ trên 80%, phương pháp sử dụng đặc trưng ngữ nghĩa kết hợp với thuật toán cây quyết định C5.0 thu được kết quả khá tốt trên 96%, trong khi hai phương pháp sử dụng LSTM đều cho kết quả rất tốt trên 98%.

Để có thể thấy rõ hơn về hiệu quả phát hiện bốn phương pháp, chúng tôi đã vẽ đường cong đặc trưng hoạt động của bộ thu (ROC: Receiver Operating Characteristic curve) của các phương pháp, như trên hình 5. Đường cong ROC thể hiện mối tương quan giữa hai tỷ lệ TPR (True Positive Rate) và FPR (False Positive Rate). Đường gạch màu xanh nước biển ứng với phương pháp sử dụng HMM, giá trị diện tích dưới đường cong (AUC: Area Under the Curve) trung bình thu được là 0,8965. Đường màu xanh lá cây minh họa đường cong ROC của thuật toán C5.0 sử dụng kết hợp đặc trưng ngữ nghĩa và đặc trưng thống kê, giá trị AUC là 0,9894. Hai phương pháp sử dụng mạng LSTM có đường cong ROC đều chứng minh được tính hiệu quả trong trường hợp phân loại hai lớp. Cả hai cho tỷ lệ phát hiện DGA trên 94%, trong khi tỷ lệ phát hiện sai tên miền non-DGA là

Bảng IV
KẾT QUẢ PRECISION, RECALL VÀ F1-SCORE CHO TRƯỜNG HỢP ĐA LỚP

| Mẫu | Precision | | | | Recall | | | | F1-score | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DGA botnet | HMM [23] | Features-C5.0 [20] | LSTM [9] | Proposed method | HMM [23] | Features-C5.0 [20] | LSTM [9] | Proposed method | HMM [23] | Features-C5.0 [20] | LSTM [9] | Proposed method |
| geodo | **0,0127** | 0 | 0 | 0 | **0,4167** | 0 | 0 | 0 | **0,0246** | 0 | 0 | 0 |
| beebone | 0,0308 | 0,625 | 0,4 | **1** | 0,75 | **1** | 0,225 | 0,85 | 0,0591 | 0,7692 | 0,2872 | **0,9119** |
| murofet | **0,8235** | 0,381 | 0,7197 | 0,6996 | 0,2577 | 0,4706 | 0,5509 | **0,6061** | 0,3925 | 0,4211 | 0,6185 | **0,6523** |
| pykspa | 0,309 | 0 | **0,8294** | 0,7966 | 0,1937 | 0 | 0,6782 | **0,6964** | 0,2381 | 0 | 0,7457 | **0,7564** |
| padcrypt | 0,2069 | 0 | 0,9242 | **1** | **1** | 0 | 0,5833 | 0,75 | 0,3429 | 0 | 0,7077 | **0,8732** |
| ramnit | 0,1081 | 0 | 0,5786 | **0,5947** | 0,0551 | 0 | **0,8226** | 0,8011 | 0,073 | 0 | **0,6793** | 0,6634 |
| volatile | 0,0136 | 0 | **0,96** | 0,7083 | **0,6** | 0 | 0,4 | 0,42 | 0,0267 | 0 | 0,5543 | **0,6055** |
| ranbyus | 0,0424 | 0 | **0,4239** | 0,413 | 0,2236 | 0 | 0,504 | **0,6081** | 0,0713 | 0 | 0,4593 | **0,4822** |
| qakbot | 0,124 | **0,9773** | 0,7005 | 0,7116 | 0,0587 | **0,9835** | 0,5565 | 0,5407 | 0,0797 | **0,9804** | 0,6196 | 0,6383 |
| simda | 0,0137 | 0,7685 | **0,9067** | 0,8884 | 0,1465 | **0,964** | 0,8125 | 0,8425 | 0,025 | 0,8552 | 0,8525 | **0,8616** |
| ramdo | 0,0388 | 0 | 0,9658 | **0,9798** | 0,725 | 0 | **0,975** | 0,955 | 0,0737 | 0 | **0,9702** | 0,9643 |
| suppobox | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| locky | 0 | **0,3492** | 0 | 0 | 0 | **0,2767** | 0 | 0 | 0 | **0,3088** | 0 | 0 |
| tempedreve | 0,0015 | **0,9507** | 0 | 0 | 0,8 | **0,9766** | 0 | 0 | 0,0031 | **0,9635** | 0 | 0 |
| qadars | 0,0309 | **1** | 0 | 0,2 | 0,75 | **1** | 0 | 0,05 | 0,0594 | **1** | 0 | 0,08 |
| symmi | **0,0065** | 0 | 0 | 0 | **0,1538** | 0 | 0 | 0 | **0,0125** | 0 | 0 | 0 |
| banjori | 0,9143 | 0,6667 | 0,9992 | **0,9993** | 0,1051 | 0,2857 | **1** | 0,9998 | 0,1885 | 0,4 | **0,9996** | 0,9995 |
| tinba | 0 | 0,6 | 0,8884 | **0,8908** | 0 | 0,4167 | 0,9815 | **0,9855** | 0 | 0,4918 | 0,9277 | **0,9327** |
| hesperbot | **0,0037** | 0 | 0 | 0 | **0,0526** | 0 | 0 | 0 | **0.0069** | 0 | 0 | 0 |
| fobber | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Alexa | **1** | 0,9899 | 0,9727 | 0,9747 | 0,0002 | 0,9868 | **0,9929** | 0,9924 | 0,0003 | 0,9883 | **0,9827** | 0,9816 |
| dyre | 0,9697 | 0,1646 | 0,9755 | **0,9757** | 1 | 0,0567 | 0,9925 | **1** | 0,9846 | 0,0844 | 0,9839 | **0,9853** |
| cryptowall | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| corebot | 0,0017 | 0,3116 | 0 | **0,6** | **0,4** | 0,2191 | 0 | 0,24 | 0,0035 | 0,2573 | 0 | **0,4166** |
| P | 0,2727 | 0,0645 | **0,7521** | 0,7326 | 0,225 | 0,014 | 0,305 | **0,335** | 0,2466 | 0,023 | 0,3858 | **0,4278** |
| bedep | 0,006 | 0 | 0,8608 | **0,788** | 0,1471 | 0 | 0,2588 | **0,347** | 0,0115 | 0 | 0,3965 | **0,5621** |
| matsnu | 0 | 0,08 | 0 | 0 | 0 | 0,0435 | 0 | 0 | 0 | 0,0563 | 0 | 0 |
| PT Goz | 0,9811 | 0,9091 | 0,9958 | **0,9983** | 0,6682 | **1** | 0,9994 | 0,9992 | 0,795 | 0,9524 | 0,9976 | **0,9986** |
| necurs | 0,0244 | 0 | **0,4673** | 0,4535 | 0,0729 | 0 | 0,0583 | **0,0921** | 0,0366 | 0 | 0,1036 | **0,2157** |
| pushdo | 0,0036 | 0,1071 | **0,8806** | 0,7209 | 0,2353 | 0,0268 | 0,1706 | **0,2941** | 0,0071 | 0,0429 | 0,2744 | **0,4921** |
| cryptolocker | 0,0163 | **0,6406** | 0 | 0,0643 | **0,6917** | 0,5538 | 0 | 0,05 | 0,0318 | **0,594** | 0 | 0,0086 |
| dircrypt | **0,0017** | 0 | 0 | 0 | **0,0909** | 0 | 0 | 0 | **0,0034** | 0 | 0 | 0 |
| shifu | 0,025 | 0,2222 | **0,4064** | 0,3115 | **1** | 0,2 | 0,3064 | 0,2894 | 0,0489 | 0,2105 | **0,3416** | 0,2727 |
| bamital | 0,6316 | 0,4839 | 0,7833 | **1** | **1** | 0,5797 | 0,55 | 0,7333 | 0,7742 | 0,5275 | 0,6366 | **0,8602** |
| kraken | 0,0041 | **0,4545** | 0,1666 | 0,1765 | 0,0196 | **0,4545** | 0,0039 | 0,0235 | 0,0068 | **0,4545** | 0,0076 | 0,0519 |
| nymaim | 0,0085 | **0,3062** | 0,2875 | 0,2862 | 0,225 | **0,39** | 0,004 | 0,0566 | 0,0165 | **0,3431** | 0,0692 | 0,184 |
| shiotob | 0,2404 | 0,4767 | 0,9114 | **0,9239** | 0,2749 | 0,3761 | 0,8845 | **0,8908** | 0,2565 | 0,4205 | 0,8976 | **0,9119** |
| W32.Virut | 0,0035 | **0,4403** | 0 | 0 | **1** | 0.2439 | 0 | 0 | 0,007 | **0,3139** | 0 | 0 |
| **Micro-averaging** | 0,8085 | 0,8652 | 0,9193 | **0,9208** | 0,0782 | 0,8854 | 0,9315 | **0,9325** | 0,0964 | 0,8735 | 0,9201 | **0,9224** |
| **Macro-averaging** | 0,1808 | 0,315 | 0,4672 | **0,497** | 0,351 | 0,3031 | 0,3583 | **0,4065** | 0,1291 | 0,3015 | 0,3816 | **0,4417** |

0,1%. Đặc biệt, LSTM kết hợp với đặc trưng thống kế cho kết quả tốt nhất với AUC = 0,99702.

## 5. Phân loại đa lớp

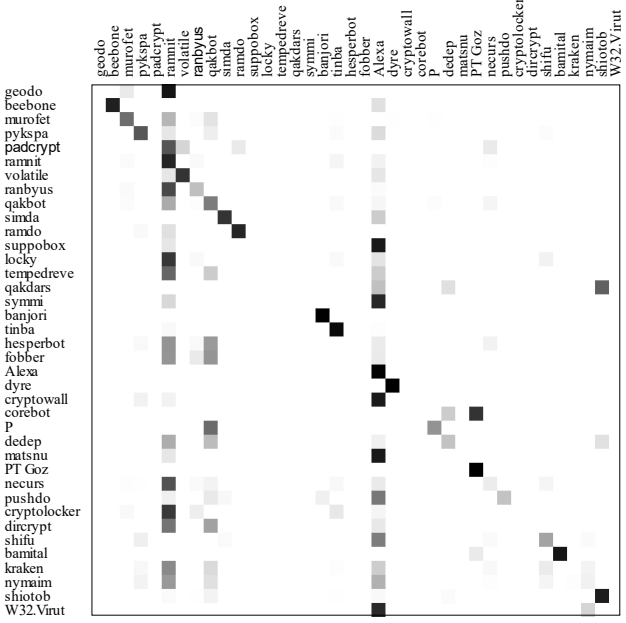
Quá trình này nhằm phân loại và phát hiện kiểu mã độc sinh ra tên miền được xác định là DGA. Bảng III cho chúng ta thấy rõ hơn các giá trị Precision, Recall và F1-score ứng với từng DGA botnet với hai phương diện micro và macro. Phương pháp sử dụng HMM có tỉ lệ phát hiện thấp hơn rất nhiều so với kỳ vọng. Điều này chứng minh HMM không hiệu quả khi áp dụng trên tập dữ liệu lớn với nhiều mẫu DGA khác nhau. Độ chính xác của C5.0 kém hơn so với hai phương pháp sử dụng mạng LSTM. Phương pháp đề xuất có tỷ lệ phát hiện cao nhất, điều này có thể dễ dàng nhìn thấy dựa vào giá trị trung bình của F1-score, Precision và Recall. Bên cạnh đó, phương pháp đề xuất còn phát hiện thêm 3 mẫu DGA botnet: *qadars*, *corebot* và *cryptolocker*.

Trong số 27 mẫu DGA botnet mà phương pháp đề xuất có thể phát hiện được, có nhiều mẫu có số lượng phần tử khá ít (dưới 50 tên miền) như *beebone*, *padcrypt*, *volatile*, *qadars*, *corebot* và *bamital*. Trong đó, một số mẫu có tỷ lệ Recall trên 99%. Tương tự như HMM, C5.0 và LSTM, phương pháp đề xuất có tỷ lệ phát hiện rất thấp hoặc không phát hiện được mẫu DGA (*suppobox*, *matsnu*, *cryptowall*) có cách đặt tên giống bình thường với nhiều cụm từ có ý nghĩa được trích rút trực tiếp từ từ điển (thường là tiếng Anh). Tuy vậy, phương pháp của chúng tôi lại có thể phát hiện những tên miền của beebone (F1-score = 0,9119).

Phần lớn những mẫu DGA botnet không phát hiện được là những mẫu DGA có số lượng phần tử khá ít chỉ vài chục tên miền. Lượng dữ liệu này không đủ đáp ứng, ảnh hưởng đến quá trình huấn luyện để trích rút ra vector đặc trưng cho tên miền. Những mẫu này thường bị phát hiện sai thành dạng DGA khác khiến tỷ lệ Recall trong kịch bản đa lớp kém hơn so với kịch bản hai lớp.

Hình 6 sẽ cho thấy rõ hơn về ma trận nhầm lẫn (confusion matrix) của phương pháp do chúng tôi đề xuất. Trục hoành ứng với giá trị thực tế của các lớp tên miền, trục tung ứng với giá trị được dự đoán của các lớp tên miền. Dải màu được sử dụng là dải màu đen trắng, màu càng nhạt ứng với số lượng càng ít và màu càng đậm ứng với số lượng càng nhiều tên miền. Số lượng tên miền đã được chuẩn hóa về dải có phạm vi từ 0 đến 1. Các mẫu DGA botnet chủ yếu bị phân loại sai thành *ramnit* và *alexa*. Trong tập dữ liệu thử nghiệm, có 22 mẫu DGA botnet bị nhận thành ramnit. Nhiều mẫu DGA botnet có tỉ lệ tên miền bị nhận sai khá lớn như *geodo*, *ranbyus*, *locky*, *tempedreve*, *necurs* và *cryptolocker*.

Số lượng tên miền bị phân loại sai thành *alexa* là lớn nhất, gồm 26 mẫu trong đó có nhiều mẫu DGA botnet có tỉ lệ bị phân loại nhầm khá lớn như *suppobox*, *symmi*, *cryptowall*, *matsnu* và *W32.Virut*.



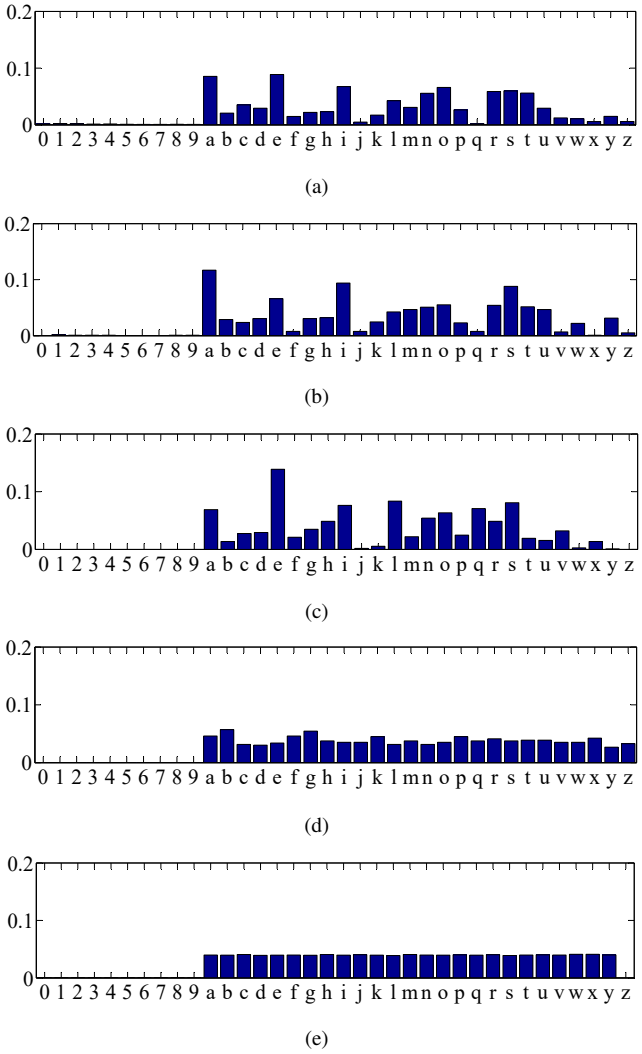

Hình 6. Ma trận nhầm lẫn của phương pháp sử dụng mạng LSTM kết hợp với đặc trưng thống kê.

Một điểm chú ý nữa đó là có 3 mẫu DGA botnet mà cả bốn phương pháp đều không thể phát hiện được là *suppobox*, *fobber* và *cryptowall*. Trong số đó, *suppobox* và *cryptowall* bị phân loại sai thành *alexa*, trong khi *fobber* bị phân loại sai thành *ramnit*. Từ các hình 7 (a)-(c) chúng ta có thể thấy rằng phân phối mật độ các ký tự [20] của *suppobox* và *cryptowall* khá giống so với phân phối của *alexa*. Trong thực tế, tên miền do *suppobox* và *cryptowall* sinh ra gồm cụm từ có ý nghĩa giống nên dễ bị phân loại sai thành *alexa*. Từ các hình 7 (d)-(e) chúng ra còn có thể thấy phân phối mật độ các ký tự của *fobber* và *ramnit* là như nhau, nhưng do số lượng tên miền của *ramnit* trong tập dữ liệu nhiều hơn nên những tên miền của *fobber* rất dễ bị phân loại sai thành tên miền của *ramnit*.

## V. KẾT LUẬN

Bài báo đề xuất phương pháp phát hiện DGA botnet sử dụng mạng LSTM kết hợp với đặc trưng thống kê. Bằng thực nghiệm, chúng tôi đã chứng minh tính hiệu quả của phương pháp đề xuất so với một số thuật toán phổ biến như HMM, C5.0 và mạng LSTM truyền thống.

Từ bảng IV, chúng tôi nhận thấy một số DGA rất khó phát hiện. Nguyên nhân là do số lượng mẫu của DGA đó quá ít (dưới 100) so với tên miền bình thường (non-DGA) hoặc DGA khác. Đây là vấn đề dữ liệu không đồng đều và hay gặp trong các bài toán xử lý dữ liệu lớn. Trong tương




Hình 7. Phân phối unigram của (a) alexa, (b) cryptowall, (c) suppobox, (d) fobber và (e) ramnit.

lai, chúng tôi sẽ triển khai các giải pháp để giải quyết vấn đề trên. Mục tiêu là nâng cao tỷ lệ phát hiện đúng DGA và giảm tỷ lệ cảnh báo sai khi hệ thống được triển khai trong thực tế.

## LỜI CẢM ƠN

Các nghiên cứu trong bài báo này được tài trợ từ Chương trình KH&CN trọng điểm cấp quốc gia KC.01/16-20 với đề tài "Nghiên cứu, phát triển tích hợp hệ thống hỗ trợ giám sát, quản lý, vận hành an toàn cho hệ thống mạng và hạ tầng cung cấp dịch vụ công trực tuyến", mã số KC.01.01/16-20.

## TÀI LIỆU THAM KHẢO

[1] Tống Văn Vạn, Nguyễn Linh Giang, and Trần Quang Đức, "Phân loại tên miền sử dụng các đặc trưng ngữ nghĩa trong phát hiện DGA Botnet," *Research and Development on Information and Communication Technology*, vol. 11, pp. 57–62, 2016.

[2] N. Davuth and S.-R. Kim, "Classification of malicious domain names using support vector machine and bi-gram method," *International Journal of Security and Its Applications*, vol. 7, no. 1, pp. 51–58, 2013.

[3] J. Kwon, J. Lee, H. Lee, and A. Perrig, "PsyBoG: A scalable botnet detection method for large-scale DNS traffic," *Computer Networks*, vol. 97, pp. 48–73, 2016.

[4] M. Grill, I. Nikolaev, V. Valeros, and M. Rehak, "Detecting DGA malware using NetFlow," in *Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2015, pp. 1304–1309.

[5] M. Mowbray and J. Hagen, "Finding domain-generation algorithms by looking at length distribution," in *Proceedings of the IEEE International Symposium on Software Reliability Engineering Workshops*, 2014, pp. 395–400.

[6] S. Schiavoni, F. Maggi, L. Cavallaro, and S. Zanero, "Phoenix: Dga-based botnet tracking and intelligence," in *Proceedings of the International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer, 2014, pp. 192–211.

[7] M. Antonakakis, R. Perdisci, Y. Nadji, N. Vasiloglou, S. Abu-Nimeh, W. Lee, and D. Dagon, "From throw-away traffic to bots: detecting the rise of dga-based malware," in *Proceedings of the 21st {USENIX} Security Symposium ({USENIX} Security 12)*, 2012, pp. 491–506.

[8] R. Perdisci, I. Corona, and G. Giacinto, "Early detection of malicious flux networks via large-scale passive DNS traffic analysis," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 5, pp. 714–726, 2012.

[9] J. Woodbridge, H. S. Anderson, A. Ahuja, and D. Grant, "Predicting domain generation algorithms with long short-term memory networks," *arXiv preprint arXiv:1611.00791*, 2016.

[10] V. Tong and G. Nguyen, "A method for detecting DGA botnet based on semantic and cluster analysis," in *Proceedings of the Seventh Symposium on Information and Communication Technology*. ACM, 2016, pp. 272–277.

[11] T.-D. Nguyen, T.-D. Cao, and L.-G. Nguyen, "DGA botnet detection using collaborative filtering and density-based clustering," in *Proceedings of the Sixth International Symposium on Information and Communication Technology*. ACM, 2015, pp. 203–209.

[12] H. Zhang, M. Gharaibeh, S. Thanasoulas, and C. Papadopoulos, "BotDigger: Detecting DGA Bots in a Single Network," Computer Science Technical Report, Tech. Rep., 2016.

[13] Osint DGA Feed. [Online]. Available: https://osint.bambenekconsulting.com/feeds/

[14] T. Robinson, "An application of recurrent nets to phone probability estimation," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, 1994.

[15] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *Proceedings of the Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] F. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: continual prediction with LSTM," *Neural computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[18] J. Han and C. Moraga, "The influence of the sigmoid function parameters on the speed of backpropagation learning," in *International Workshop on Artificial Neural Networks*. Springer, 1995, pp. 195–201.

[19] G. F. Becker, *Hyperbolic functions*, 1931.

[20] P. F. Brown, P. V. Desouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai, "Class-based n-gram models of natural language," *Computational Linguistics*, vol. 18, no. 4, pp. 467–479, 1992.
[21] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.
[22] Alexa. [Online]. Available: http://www.alexa.com
[23] T.-S. Wang, C.-S. Lin, and H.-T. Lin, "DGA botnet detection utilizing social network analysis," in *Proceedings of the International Symposium on Computer, Consumer and Control (IS3C)*. IEEE, 2016, pp. 333–336.
[24] D.-F. Xia, S.-L. Xu, and F. Qi, "A proof of the arithmetic mean-geometric mean-harmonic mean inequalities," *RGMIA Research Report Collection*, vol. 2, no. 1, 1999.

**Mạc Đình Hiếu** nhận bằng kỹ sư và thạc sĩ tại Trường Đại học Bách khoa Hà Nội vào các năm 2014 và 2016. Hiện nay, tác giả đang là nghiên cứu sinh, chuyên ngành Mạng máy tính và Truyền thông dữ liệu tại Trường Đại học Bách khoa Hà Nội. Lĩnh vực quan tâm nghiên cứu của tác giả là an toàn bảo mật thông tin và IoT.

**Tống Văn Vạn** nhận bằng kỹ sư tại Trường Đại học Bách khoa Hà Nội vào năm 2017. Lĩnh vực quan tâm nghiên cứu của tác giả là an ninh mạng, an toàn bảo mật thông tin và IoT.

**Bùi Trọng Tùng** nhận bằng kỹ sư và thạc sĩ tại Trường Đại học Bách khoa Hà Nội vào các năm 2008 và 2010. Hiện nay, tác giả là giảng viên Viện Công nghệ Thông tin và Truyền thông và là cán bộ kiêm nhiệm tại Trung tâm An toàn an ninh thông tin, Trường Đại học Bách khoa Hà Nội. Lĩnh vực quan tâm nghiên cứu của tác giả là an ninh mạng, an toàn và bảo mật thông tin.

**Trần Quang Đức** nhận bằng thạc sĩ tại Trường Đại học Bách khoa Budapest, năm 2008 và bằng tiến sĩ tại Trường Đại học City University London, Vương Quốc Anh, năm 2014. Hiện nay, ông là Giám đốc Trung tâm An toàn an ninh thông tin, Trường Đại học Bách khoa Hà Nội. Lĩnh vực quan tâm nghiên cứu của ông là học máy, nhận dạng, sinh trắc học, an toàn và bảo mật thông tin.

**Nguyễn Linh Giang** nhận học vị tiến sĩ chuyên ngành đảm bảo toán học cho máy tính, năm 1995, tại Cộng hòa Gruzia (Liên xô cũ). Hiện nay, ông đang công tác tại Bộ môn Truyền thông và mạng máy tính, Viện Công nghệ Thông tin và Truyền thông, Trường Đại học Bách khoa Hà Nội. Lĩnh vực quan tâm nghiên cứu của ông là các phương pháp học máy, an ninh mạng và phát hiện tấn công mạng.