NGHIÊN CỨU CÁC MÔ HÌNH MẠNG NƠ-RON ĐỒ THỊ CHO PHÁT HIỆN BOTNET

Nguyễn Ngọc Điệp, Nguyễn Thị Thanh Thủy
Học viện Công nghệ Bưu chính Viễn thông

Tóm tắt: Các mô hình mạng nơ-ron đồ thị (GNN) đã cho thấy hiệu quả vượt trội trong việc phát hiện botnet so với các phương pháp học máy và học sâu truyền thống nhờ khả năng học và khai thác các mối quan hệ phức tạp trong dữ liệu mạng bằng cách biểu diễn chúng dưới dạng đồ thị. Bài báo này tập trung nghiên cứu và đánh giá hiệu năng của các mô hình GNN tiên tiến như GCN, GAT, UniMP, và GIN trong việc phát hiện botnet trên cả dữ liệu mô phỏng và thực tế. Kết quả cho thấy, các mô hình GNN đều đạt hiệu quả cao với giá trị F1 trên 98,6% trên tất cả các bộ dữ liệu, khẳng định tiềm năng của GNN trong ứng dụng thực tiễn. Các mô hình tích hợp cơ chế tập trung như GATv2 và UniMP cũng cho kết quả khả quan chỉ với một lượng không nhiều các thông tin mạng. Đặc biệt, mô hình GIN cho thấy khả năng học cấu trúc hình học đồ thị tốt hơn, đạt hiệu năng cao nhất trên tất cả các bộ dữ liệu mô phỏng và thực tế. Ngoài ra, mô hình RevGIN, kết hợp GIN với kiến trúc kết nối dư nghịch đảo, cho phép xây dựng mô hình sâu hơn và tiết kiệm bộ nhớ đáng kể, mở ra hướng nghiên cứu mới cho việc phát hiện botnet trên các mạng quy mô lớn.

Từ khóa: tấn công mạng, botnet, học sâu, mạng nơ-ron đồ thị.

I. GIỚI THIỆU

Botnet là một mạng lưới các thiết bị bị xâm nhập và kiểm soát bởi kẻ tấn công thông qua các phần mềm độc hại. Số lượng các thiết bị trong mạng botnet càng nhiều thì mức độ nguy hiểm của các cuộc tấn công càng cao và đặc biệt là khó triệt phá hoàn toàn. Trên thực tế, một mạng botnet có thể gồm hàng ngàn, hàng trăm ngàn hoặc có thể lên đến hàng triệu thiết bị. Botnet thường được sử dụng trong các cuộc tấn công từ chối dịch vụ phân tán (DDoS) hoặc mã độc tống tiền, nhắm vào các ngành như tài chính, y tế và chính phủ [1]. Ví dụ, Akamai đã ghi nhận các cuộc tấn công DDoS kỷ lục, một trong số đó đạt đỉnh 900,1 Gbps, nhằm làm tê liệt các tổ chức tài chính và đánh cắp dữ liệu nhạy cảm [2]. Báo cáo an ninh mạng năm 2023 của Nokia cho thấy sự gia tăng mạnh mẽ các cuộc tấn công do botnet IoT điều khiển. Số lượng thiết bị IoT bị sử dụng làm bot đã tăng từ 200.000 lên khoảng 1 triệu, làm gia tăng đáng kể khả năng thực hiện các cuộc tấn công DDoS quy mô lớn và xâm nhập dữ liệu [3].

Tác giả liên hệ: Nguyễn Ngọc Điệp,
Email: diepnguyenngoc@ptit.edu.vn
Đến tòa soạn: 11/2024, chỉnh sửa: 12/2024, chấp nhận đăng: 01/2025.

Với mức độ vô cùng nghiêm trọng của các cuộc tấn công dựa trên botnet, một phương pháp hiệu quả được sử dụng là ngăn chặn sớm các cuộc tấn công này, bằng cách truy tìm và triệt phá các botnet đang tồn tại. Việc này sẽ giúp loại bỏ và giảm tối thiểu mức độ thiệt hại mà các cuộc tấn công dựa trên botnet có thể gây ra.

Các phương pháp truyền thống như Honeynet [4], phát hiện dựa trên chữ ký [5], và học máy truyền thống [5] đang dần trở nên kém hiệu quả trước các botnet mới và biến thể của chúng. Honeynet bị hạn chế về khả năng truy vết, do không cung cấp được thông tin về kích thước, phạm vi lây lan của botnet, điều này khiến cho việc triệt phá botnet gặp nhiều khó khăn. Các phương pháp dựa trên chữ ký có ưu điểm với tốc độ phát hiện nhanh và chính xác, tuy nhiên lại không hiệu quả trong việc phát hiện các botnet mới hoặc các biến thể do kẻ tấn công có thể dễ dàng thay đổi một số đặc điểm của botnet để tạo ra các biến thể mới, không có trong cơ sở dữ liệu chữ ký. Các tiếp cận dựa trên học máy truyền thống [5] có tốc độ xử lý nhanh và có độ chính xác cao, nhưng yêu cầu kiến thức chuyên gia để tạo ra các đặc trưng thủ công hiệu quả, và có thể dẫn đến việc bỏ sót các đặc trưng quan trọng. Ngoài ra, việc lựa chọn đặc trưng thủ công cũng có thể tạo cơ hội cho kẻ tấn công sử dụng các kỹ thuật chống học máy để tránh bị phát hiện.

Để vượt qua được những thách thức này, các nhà nghiên cứu gần đây sử dụng một hướng tiếp cận khác, đó là các mô hình học sâu với độ chính xác vượt trội [6], [7]. Học sâu có thể tự động trích xuất và học các đặc trưng phức tạp từ dữ liệu mạng lớn mà không cần sự can thiệp thủ công. Điều này giúp mô hình nhận diện tốt các mẫu bất thường trong lưu lượng mạng, ngay cả khi dữ liệu bị nhiễu hoặc có nhiều loại tấn công khác nhau.

Mặc dù có hiệu năng cao trong nhiều tác vụ khác nhau nhưng các mô hình học sâu truyền thống [8], ví dụ như mạng nơ-ron tích chập (CNN) hoặc mạng nơ-ron hồi tiếp (RNN) thường tập trung vào việc phát hiện các mẫu trong dữ liệu tuần tự hoặc hình ảnh, mà không trực tiếp khai thác mối quan hệ giữa các nút mạng, do đó không hiệu quả trong việc phát hiện các hành vi phối hợp theo nhóm. Các mô hình dựa trên mạng nơ-ron đồ thị được đề xuất gần đây [9], rất hiệu quả trong việc nhận diện các hành vi này. Tuy nhiên, các nghiên cứu này sử dụng GNN để phân tích các hành vi của các thực thể mạng và đang phụ thuộc vào sự hiểu biết của các nhà nghiên cứu về hành vi của botnet cũng như các đặc trưng thủ công cụ thể trong dữ liệu lưu lượng mạng, ví dụ như kích thước gói tin, số cổng để phân biệt botnet với lưu lượng nền. Các mẫu lưu lượng chi tiết

này có thể bị cố ý điều chỉnh một cách tinh vi để né tránh sự giám sát. Ngoài ra, quy mô khổng lồ của các giao tiếp trong mạng botnet hiện nay cũng gây ra khó khăn rất lớn cho việc xử lý dữ liệu. Để khắc phục khó khăn này, một số nghiên cứu gần đây đã đề xuất sử dụng mạng nơ ron đồ thị để nắm bắt hiệu quả cấu trúc liên kết của các đồ thị biểu diễn mạng botnet thay vì dựa vào các đặc trưng thủ công, và đã đạt được kết quả khả quan [12].

Nhằm khai thác sâu hơn khả năng ứng dụng GNN để giải quyết bài toán phát hiện botnet dựa trên cấu trúc liên kết của đồ thị biểu diễn cho mạng botnet, nghiên cứu này đề xuất sử dụng các kiến trúc GNN tiên tiến, bao gồm GCN (Graph Convolutional Networks) [10], GAT (Graph Attention Networks) [10], UniMP (Unified Message Passing Model) [11] và GIN (Graph Isomorphism Network) [10], và đánh giá hiệu năng của các mô hình này trên tập dữ liệu botnet tiêu chuẩn đề xuất trong [12].

Các đóng góp chính của bài báo như sau:

1) Đề xuất và thử nghiệm phương pháp phát hiện botnet dựa trên các mô hình mạng nơ-ron đồ thị khác nhau, bao gồm GCN, GAT, UniMP và GIN, để tìm ra mô hình hiệu quả nhất trong khả năng khai thác thông tin về cấu trúc liên kết của mạng botnet.
2) Đánh giá hiệu năng các mô hình đề xuất trên các bộ dữ liệu tiêu chuẩn về cấu trúc liên kết (hình học) của các loại mạng botnet khác nhau [12], đối với tiêu chí về độ đo F_1 và tiêu chí về khả năng tiết kiệm bộ nhớ.

Phần còn lại của bài báo được tổ chức như sau. Phần II mô tả các nghiên cứu liên quan. Phần III trình bày lý thuyết cơ sở về các mô hình GNN tiên tiến sẽ sử dụng và đề xuất phương pháp phát hiện botnet dựa trên GNN. Kết quả và những phân tích thực nghiệm được trình bày trong Phần IV. Cuối cùng, Phần V là kết luận bài báo và định hướng nghiên cứu.

II. CÁC NGHIÊN CỨU LIÊN QUAN

Có thể phân chia các nghiên cứu liên quan trước đây về các phương pháp phát hiện botnet thành ba nhóm chính: thứ nhất là các phương pháp phát hiện dựa trên luật và chữ ký, thứ hai là các phương pháp phát hiện dựa trên học máy truyền thống, và thứ ba là các phương pháp phát hiện dựa trên các kỹ thuật học sâu tiên tiến.

Các phương pháp phát hiện botnet dựa trên luật thường được thực hiện bằng cách thu thập các thông tin về botnet và mẫu tấn công, sau đó phân tích, so sánh để có thể xác định và loại bỏ chúng. Các phương pháp phổ biến có thể kể đến là phương pháp phát hiện dựa trên Honeynet và phương pháp phát hiện dựa trên chữ ký. Phát hiện dựa trên Honeynet [4] là phương pháp xây dựng một hệ thống tài nguyên nhằm đánh lừa những kẻ xâm nhập bất hợp pháp, thu hút chúng tiếp xúc với hệ thống này thay vì tiếp xúc với hệ thống thật. Từ đó, có thể thu thập các thông tin liên quan đến kẻ xâm nhập, phân tích và tìm ra các điểm yếu của chúng để có thể xác định và loại bỏ chúng. Một số thông tin thu thập bao gồm: cơ chế hoạt động của nút, chữ ký, kỹ thuật chúng sử dụng, lỗ hổng của hệ thống mà chúng sử dụng để khai thác (nhằm gây lây lan mã độc và mở rộng botnet). Hạn chế của phương pháp này là khó truy vết, không đưa ra được thông tin về độ lớn và phạm vi lây lan của botnet, do vậy, chỉ được sử dụng như bước đầu tiên để thăm dò. Phương pháp phát hiện dựa trên chữ ký [4] thực hiện so sánh các dấu hiệu thu thập được với chữ ký đã được xác định trước của các botnet, từ đó có thể phát hiện ra các loại mã độc hoặc các hành vi bất thường mà mã độc gây ra. Ưu điểm của phương pháp này là nhanh và có độ chính xác cao. Tuy nhiên, nhược điểm là khó phát hiện các loại bot mới, đồng thời cần liên tục cần cập nhật cơ sở dữ liệu chữ ký và các luật mới.

Các phương pháp phát hiện botnet dựa trên học máy truyền thống cũng đã được sử dụng nhiều trong những năm gần đây. Các phương pháp này sử dụng các tập dữ liệu botnet đã được công bố để phục vụ cho việc xây dựng các mô hình phân loại, nhằm xác định xem một nút (đại diện cho thiết bị/thực thể mạng) có thuộc botnet hay không. Một số các tập dữ liệu phổ biến đã được sử dụng trong các nghiên cứu gồm Bot-IoT [13], UNSW-NB15 [14], V-Sandbox [15]. Nhiều thuật toán học máy truyền thống được sử dụng trong các nghiên cứu phát hiện botnet [5] như Naïve Bayes, Decision Tree, KNN, Support Vector Machine (SVM), Random Forest. Các nghiên cứu thử nghiệm phát hiện botnet sử dụng học máy truyền thống cho kết quả tương đối khả quan. Tuy nhiên, nhược điểm của các phương pháp này cũng khá nhiều, đó là (1) cần phải có được tập dữ liệu thực (hoặc dữ liệu mô phỏng) đủ tốt, (2) cần có tri thức chuyên gia cho việc thiết kế tập đặc trưng thủ công hiệu quả, và (3) lựa chọn thuật toán học máy phù hợp và hiệu quả cho việc huấn luyện mô hình học. Thực tế, những công việc này rất tốn kém thời gian và chi phí.

Các phương pháp phát hiện botnet dựa trên học sâu đã chứng minh hiệu quả vượt trội so với mô hình học máy trước đó trong việc xử lý dữ liệu phức tạp và phát hiện các biến thể tấn công mới [5]. Các mô hình học sâu này có khả năng nhận diện các mối đe dọa trong dữ liệu mạng phức tạp với độ chính xác cao hơn. Nghiên cứu [16] đề xuất một hệ thống phát hiện dựa trên học sâu sử dụng dữ liệu luồng mạng để xác định hoạt động botnet, thông qua việc chuyển đổi dữ liệu mạng thành các bản ghi kết nối, giúp mô hình học sâu nhận diện các mối đe dọa với độ chính xác cao. Nghiên cứu [17] gần đây kết hợp mạng nơ-ron tích chập và GRU để phát hiện tấn công botnet trên mạng IoT. Kết quả cho thấy mô hình này đạt hiệu suất cao nhờ khả năng học sâu từ dữ liệu lưu lượng mạng và phát hiện mẫu bất thường theo thời gian.

Để đối phó với các cuộc tấn công botnet với cấu trúc mạng phức tạp và hành vi thay đổi tinh vi gây ra khó khăn cho các mô hình học sâu, một số nhà nghiên cứu đã cố gắng tận dụng biểu diễn đồ thị của các luồng mạng nhằm phát hiện các mẫu giao tiếp phức tạp giữa các máy chủ. Zhou và cộng sự [12] đã đề xuất một phương pháp giám sát dựa trên mạng GCN và các dấu vết mạng. Trong nghiên cứu này, chỉ có cấu trúc hình học của đồ thị được xem xét. Các đồ thị được sử dụng để huấn luyện có kích thước lớn, trung bình chứa hơn 140.000 nút và 700.000 cạnh. Mô hình GCN tính toán nhúng của các nút trên toàn bộ đồ thị

và các nút botnet độc hại sau đó được phân loại bằng một mạng nơ-ron ở cấp độ nút. Hiệu suất của mô hình đã được đánh giá trên một tập dữ liệu tổng hợp. Tương tự như vậy, Zhang và đồng sự [18] sử dụng GCN với 12 lớp để phát hiện botnet, nhằm nắm bắt mối quan hệ dài hạn trong kiến trúc botnet lớn. Tuy nhiên, mô hình học sâu như vậy dễ gặp vấn đề làm mịn quá mức (over-smoothing), nghĩa là biểu diễn của các nút bị mất sau nhiều lần lặp lại khi truyền thông điệp trên đồ thị.

Trong nghiên cứu này, chúng tôi đưa ra thử nghiệm và đánh giá sự hiệu quả của các mô hình mạng nơ-ron đồ thị khác nhau gồm GCN, GAT, UniMP và GIN trên cùng tập dữ liệu của Zhou và cộng sự [12]. Hiệu năng của các mô hình được so sánh và đánh giá để từ đó tìm ra được mô hình hiệu quả trong việc nắm bắt các đặc trưng của mạng, đặc biệt là cấu trúc liên kết mạng.

III. PHƯƠNG PHÁP ĐỀ XUẤT

Để hiểu chi tiết hơn về kiến trúc của các mô hình phát hiện botnet đề xuất, trước hết chúng tôi giới thiệu sơ bộ về cơ chế hoạt động của mạng nơ-ron đồ thị và các mô hình GNN tiên tiến được sử dụng như trong phần A dưới đây, sau đó mô tả về bài toán và mô hình đề xuất.

A. Một số mô hình GNN tiên tiến

1) Mạng nơ-ron đồ thị

Trong những năm gần đây, mạng nơ-ron đồ thị (GNN) [19], [20] đã thu hút sự chú ý lớn nhờ khả năng xử lý dữ liệu có cấu trúc đồ thị, cho phép học các đặc trưng của mạng và các mối quan hệ giữa các nút. GNN có thể được áp dụng cho các tác vụ ở các cấp độ khác nhau, bao gồm dự đoán cấp độ nút, cấp độ cạnh và cấp độ đồ thị.

Đối với một nút v trong đồ thị G, GNN học một biểu diễn ẩn, h_v , bằng cách sử dụng cả đặc trưng của nút x_v và cấu trúc đồ thị (các cạnh) E làm đầu vào. Đa số các mô hình GNN tiên tiến theo chiến lược truyền thông điệp và tổng hợp, gọi là kiến trúc MPNN (Message-Passing Neural Network) [10], trong đó thông tin của một nút được truyền lần lượt đến các nút khác dọc theo các cạnh, và sau đó, biểu diễn của nút được cập nhật bằng cách tổng hợp và học thông tin của chính nó cùng với các nút láng giềng. Khi thực hiện truyền thông điệp với GNN có l lớp, mỗi nút có thể thu nhận thông tin của các nút láng giềng trong phạm vi l bước nhảy. Không làm mất tính tổng quát, biểu diễn ẩn ở lớp thứ l của GNN có thể được mô tả như sau:

$$h_v^l = f^l(h_v^{l-1}, \mathrm{AGG}^l(\{h_u^{l-1}, \forall u \in N_v\})) \quad (1)$$

Trong đó, N_v đại diện cho các nút láng giềng của nút v. AGG^l đại diện cho các phép tổng hợp. f^l là một hàm với các tham số có thể học. Sự kết hợp giữa AGG^l và f^l xác định loại toán tử đồ thị, ví dụ như toán tử tích chập đồ thị và toán tử đồ thị tập trung.

Đối với dự đoán cấp độ nút, đầu ra của lớp ẩn cuối cùng có thể được truyền trực tiếp vào lớp đầu ra Φ, ví dụ như lớp kết nối đầy đủ. Đối với dự đoán cấp độ đồ thị, đầu ra sẽ được truyền vào một hàm đọc R (để kết hợp thông tin từ các lớp ẩn của mô hình thành một đầu ra có ý nghĩa) với các tham số có thể học. Một cách chính thức, biểu diễn của lớp đầu ra có thể được biểu diễn như sau:

$$h_v = \Phi(h_v^l) \quad (2)$$

$$h_G = R(h_v^l \mid v \in V) \quad (3)$$

2) Mô hình GCN

Mạng GCN (Graph Convolutional Network) [10] được xây dựng dựa trên các lớp GCNConv. Nguyên lý hoạt động của mỗi lớp này được mô tả bởi công thức:

$$H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}) \quad (4)$$

Trong đó: A là ma trận kề của đồ thị. $\tilde{A} = A + I$. D là ma trận bậc của đồ thị. $\tilde{D} = D + I$. $\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$ là phương pháp chuẩn hóa đối xứng cho ma trận đặc trưng $H^{(l)}$. $H^{(l)}$ là ma trận đặc trưng của các nút sau khi đi qua lớp l. $W^{(l)}$ là ma trận trọng số ứng với lớp l và có thể huấn luyện được. σ là hàm kích hoạt phi tuyến.

3) Mô hình GIN (Graph Isomorphism Network) và RevGIN

Phép thử Weisfeiler-Lehman trong lý thuyết đồ thị rất giống với cách mà các mạng nơ-ron đồ thị hoạt động [21]. Do đó, các nhà nghiên cứu kết hợp GNN với phép thử Weisfeiler-Lehman để tạo ra mạng GIN với khả năng phân loại tốt hơn [10]. Công thức biểu diễn nguyên lý hoạt động của mỗi lớp GINConv là:

$$h_v^{(k)} = \mathrm{MLP}^{(k)}((1+\epsilon^{(k)}) \cdot h_v^{(k-1)} + \sum_{u \in \mathcal{N}(v)} h_u^{(k-1)}) \quad (5)$$

Trong đó, $h_v^{(k)}$là vector biểu diễn đặc trưng của nút v sau khi đi qua lớp k. $\mathcal{N}(v)$ là tập các nút kề với nút v. $\sum_{u \in \mathcal{N}(v)} h_u^{(k-1)}$ chính là tổng vector đặc trưng của các nút kề với nút v. ϵ xác định tầm quan trọng của nút v so với các nút u kề nó. MLP được sử dụng để nhấn mạnh mạng nơ-ron được sử dụng để tổng hợp nhiều hơn 1 lớp.

Mô hình RevGIN (Reversible GIN): Mô hình GIN kết hợp với kiến trúc kết nối dư nghịch đảo sẽ trở thành mô hình RevGIN. Với kiến trúc kết nối dư nghịch đảo, thay vì lưu tất cả các trạng thái trung gian cho quá trình lan truyền ngược, RevGIN chỉ cần lưu trạng thái hiện tại và tính toán lại các trạng thái trước đó khi cần. Kiến trúc này giúp giảm nhu cầu lưu trữ toàn bộ các trạng thái trung gian cho từng lớp trong GNN, từ đó giúp tối ưu hóa bộ nhớ khi huấn luyện các mạng nơ-ron đồ thị lớn.

4) Mô hình GATv2 (Graph Attention Network v2)

GATv2 là một biến thể của GAT, được thiết kế để cải thiện khả năng học các mối quan hệ giữa các nút trong đồ thị. Cơ chế tập trung được giới thiệu trong [22] để học trọng số cạnh giữa hai nút kết nối, trong đó cường độ kết nối α_{ij} được tính toán theo cơ chế tự tập trung cho các nút láng giềng, được mô tả theo công thức:

$$\alpha_{ij} = \frac{exp(a^{\top} LeakyReLU(W \cdot [h_i \| h_j]))}{\sum_{j' \in N_i} exp\,(a^{\top} LeakyReLU(W \cdot [h_i \| h_{j'}]))} \quad (6)$$

Trong đó, a và W là các tham số có thể học được, và ký hiệu ‖ biểu thị phép nối. Tiếp đó, biểu diễn ẩn cho nút i có thể được tính bằng cách học các biểu diễn ẩn có trọng số của các nút láng giềng:

$$h_i' = \sigma(\sum_{j \in N_i} \alpha_{ij} \cdot W h_j) \quad (7)$$

Cơ chế tập trung có thể được mở rộng với cơ chế tập trung nhiều đầu:

$$h_i' = \|_{k=1}^{K} \sigma(\sum_{j \in N_i} \alpha_{ij}^{k} \cdot W^{k} h_j) \quad (8)$$

5) Mô hình UniMP

UniMP [11] là một biến thể của các mô hình GNN, kết hợp với cơ chế tập trung để cải thiện khả năng học biểu diễn trên đồ thị. Mỗi lớp UniMP thực hiện phép tích chập dựa trên công thức sau:

$$x_i' = W_1 x_i + \sum_{j \in N(i)} \alpha_{ij} W_2 x_j \quad (9)$$

Trong đó, α_{ij} trọng số chú ý giữa nút i và nút j, được tính theo công thức khác so với GATv2 như sau:

$$\alpha_{ij} = \text{softmax}(\frac{(W_3 x_i)^{\top} (W_4 x_j)}{\sqrt{d}}) \quad (10)$$

x_i là vector đặc trưng ban đầu của nút i; W_1, W_2, W_3, W_4 là các ma trận trọng số áp dụng cho x_i và các nút lân cận x_j, được học trong quá trình huấn luyện; $N(i)$ là tập các nút kề với nút i; và d là bậc của nút i.

So với GATv2, UniMP sử dụng tích vô hướng giữa hai vector trọng số đã qua ma trận W_3 và W_4. Kết quả được chuẩn hóa bởi softmax và chia cho d để ổn định giá trị khi kích thước vector đặc trưng lớn.

C. Mô hình đề xuất

Phần dưới đây sẽ mô tả bài toán và đề xuất mô hình phát hiện botnet trong mạng dựa trên mạng nơ ron đồ thị. Mô hình đề xuất gồm 2 phần chính: (1) xây dựng đồ thị để biểu diễn luồng dữ liệu mạng và (2) kiến trúc mạng nơ-ron đồ thị để đưa ra dự đoán về các nút trong đồ thị đầu vào có phải là bot hay không.

1) Biểu diễn dữ liệu mạng dưới dạng đồ thị

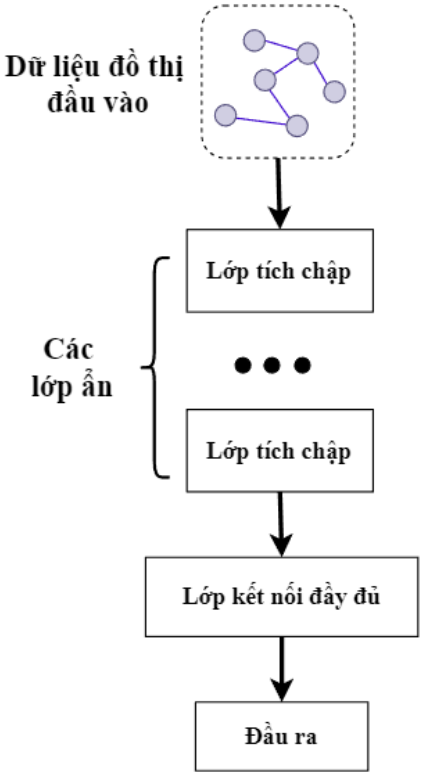
Để biểu diễn dữ liệu hoạt động mạng dưới dạng đồ thị, ta cần xác định rõ ràng cách định nghĩa nút và cạnh của đồ thị.

Nút: Mỗi nút trong đồ thị đại diện cho một thiết bị hoặc một thực thể trong mạng (ví dụ: máy chủ, máy trạm, router). Các thông tin đặc trưng của nút có thể gồm địa chỉ IP, cổng, giao thức được sử dụng, hệ điều hành, dịch vụ đang chạy,...

Do mô hình đề xuất chỉ học dựa trên cấu trúc liên kết mạng thuần túy, nên thuộc tính mà ta cần để xây dựng đồ thị lưu lượng mạng nền chỉ gồm: Địa chỉ IP nguồn và địa chỉ IP đích.

Cạnh: Mỗi cạnh trong đồ thị đại diện cho mối quan hệ giữa hai nút. Trong đồ thị biểu diễn mạng botnet, cạnh đại diện cho luồng giao tiếp giữa hai nút. Trong bài báo này, cạnh được thiết lập giữa các nút đại diện cho địa chỉ IP.

2) Kiến trúc mạng nơ-ron đồ thị đề xuất



Hình 1. Kiến trúc mô hình mạng nơ-ron đồ thị để phân loại các nút trong đồ thị đầu vào

Xét một đồ thị $G = \{V, E\}$ với V là tập n nút $\{v_1, \ldots, v_n\}$, đại diện cho các thiết bị/thực thể trong mạng. E là tập các cạnh $\{e_1, \ldots, e_m\}$. Xét cạnh e_k là một cạnh trong tập E đại diện kết nối giữa hai nút (v_i, v_j) trong mạng. $e_k = 1$ khi v_i và v_j có kết nối với nhau, và ngược lại $e_k=0$ khi không có kết nối giữa 2 nút này. Đồ thị G là dữ liệu đầu vào cho GNN.

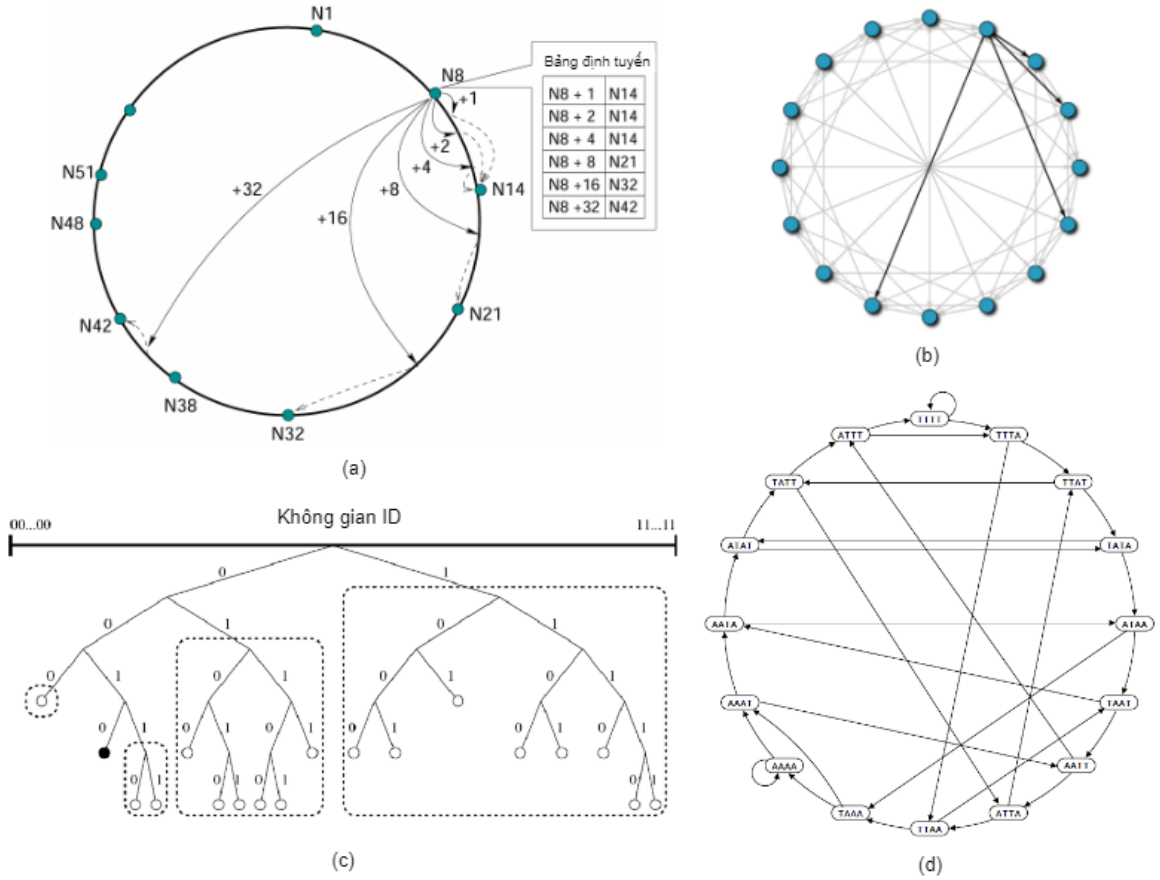
Mô hình GNN đề xuất có kiến trúc chung gồm nhiều lớp tích chập kết nối với nhau, dùng để học các vector biểu diễn cho các nút trong đồ thị (Hình 1). Tùy thuộc vào mô hình GNN cụ thể mà khả năng học của đồ thị khác nhau, dẫn tới số lượng các lớp trong GNN cũng như các chi tiết khác nhau. Trong mô hình mạng GNN, hàm kích hoạt được sử dụng là ReLU. Tương tự, ta có các kiến trúc mạng GNN dựa trên các mô hình khác như GATv2, GCN, UniMP, GIN.

Lớp cuối cùng là một lớp kết nối đầy đủ dùng để dự đoán liệu một nút trong đồ thị là thuộc botnet hay không, theo sau là hàm softmax để phân loại. Riêng mô hình GIN sử dụng một số lớp MLP.

IV. THỰC NGHIỆM VÀ KẾT QUẢ

A. Bộ dữ liệu

Nghiên cứu này sử dụng bộ dữ liệu tiêu chuẩn đề xuất trong [23] với hai loại dữ liệu: (1) lưu lượng botnet được tạo ra bằng phương pháp mô phỏng, gồm các 4 bộ *Chord*, *De Bruijn*, *Kademlia*, *Leet-Chord* và (2) lưu lượng botnet từ môi trường thực, với 2 bộ *C2* và *P2P*. Mỗi bộ dữ liệu bao gồm 960 đồ thị, mỗi đồ thị có hơn 140.000 nút và khoảng 830.000 cạnh. Đối với bộ dữ liệu mô phỏng, số lượng nút thuộc loại Bot trong mỗi đồ thị khoảng 10.000 nút. Đối với bộ dữ liệu thực, số lượng Bot trong mỗi đồ thị là khoảng 3.000 nút. Tất cả các bộ dữ liệu này đều đã được tiền xử lý.



Hình 2. Minh họa các kiến trúc mạng botnet tổng hợp: (a) Kiến trúc Chord với bảng định tuyến của nút 8 [24]; (b) Kiến trúc LEET-Chord với 16 nút; (c) Không gian ID (Identifier space) của kiến trúc Kademlia [30]; (d) Đồ thị 4 chiều theo kiến trúc De Brujin [31].

Bốn bộ dữ liệu *Chord*, *Debru*, *Kedem*, *Leet* đại diện cho 4 loại kiến trúc botnet được tổng hợp và minh họa như trong Hình 2.

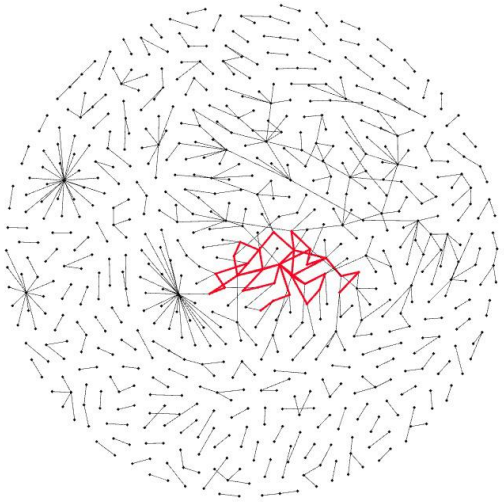
Chord [24]: đồ thị mạng tuân theo kiến trúc này được biểu diễn dưới dạng vòng tròn gồm N nút. Mỗi nút duy trì một bảng định tuyến (finger table) gồm m bản ghi. Bản ghi thứ i trong bảng định tuyến của nút n chứa ID của nút kế tiếp gần nhất s. s được tính bằng công thức: successor$((n + 2^{k-1}) \bmod 2^m)$, với $1 \le k \le m$.

LEET-Chord [25]: Kiến trúc LEET-Chord là một phiên bản giản lược của kiến trúc Chord. Đồ thị mạng tuân theo kiến trúc LEET-Chord được biểu diễn dưới dạng vòng tròn gồm N nút. Mạng LEET-Chord được xây dựng theo quy tắc sau: Mỗi nút thứ i sẽ kết nối đến các nút có ID xác định bởi công thức: $(i + 2^k) \bmod N$, với $k = 1,2,\dots,\log_2(N) - 1$.

De Bruijn [26]: Kiến trúc De Bruijn có thể được biểu diễn dưới dạng một đồ thị vòng tròn, trong đó các nút được đại diện bởi các chuỗi ký tự có độ dài d (thường gọi là đồ thị d-chiều). Các chuỗi này được sinh ra từ tập ký tự $\Sigma = \{A, T\}$. Hai nút u và v sẽ được liên kết với nhau nếu: khi loại bỏ ký tự đầu tiên của chuỗi u và ký tự cuối cùng của chuỗi v, phần còn lại của hai chuỗi là giống nhau. Số lượng nút trong đồ thị De Bruijn là 2^d nút, với d là số chiều của đồ thị.

Kademlia [27]: Đồ thị mạng tuân theo kiến trúc Kademlia được biểu diễn bằng đồ thị $G = (V, E)$, trong đó V là tập đỉnh và E là tập cạnh. Đồ thị này được xây dựng dựa trên bảng định tuyến của các nút trong mạng. Cụ thể, với mỗi cặp nút i và j được biểu diễn bởi các đỉnh v và w tương ứng, một cạnh (v, w) sẽ được thêm vào tập cạnh E nếu nút j nằm trong bảng định tuyến của nút i. Kademlia sử dụng một không gian khóa băm lớn để phân bố các nút trong mạng. Mỗi nút sẽ có một khóa băm duy nhất xác định vị trí của nó trong không gian này. Mỗi nút duy trì một bảng định tuyến chứa thông tin về các nút khác có khóa băm gần với khóa băm của nó nhất. Bảng định tuyến này được tổ chức theo một cấu trúc cây nhị phân, giúp tìm kiếm các nút một cách hiệu quả.

Hai mạng botnet thực tế P2P và C2 tới từ [28], trong đó P2P là một botnet phi tập trung và C2 là một botnet tập trung. Botnet tập trung có cấu trúc phân cấp mạnh với dạng hình sao. Trong khi đó, cấu trúc của botnet phi tập trung P2P được thiết kế để truyền thông tin nhanh chóng trong nội bộ botnet, giúp chúng nhận lệnh để khởi động một cuộc tấn công một cách hiệu quả. Các botnet phi tập trung sẽ khó nhận diện hơn so với botnet tập trung với cấu trúc phân cấp rõ ràng. Đặc trưng quan trọng nhất là tốc độ trộn cao trong botnet, giúp thông tin lan truyền rất nhanh giữa các nút. Hình 3 mô tả một ví dụ về cấu trúc mạng botnet P2P trong đồ thị mạng dữ liệu nền.



Hình 3. Ví dụ về botnet phi tập trung P2P [28], với các nút màu đỏ là thuộc botnet nằm trong dữ liệu nền

B. Thiết lập thực nghiệm và cấu hình

Hiệu năng của mô hình trích xuất được đo bằng độ đo F_1, được tính từ độ chính xác (precision), độ bao phủ (recall) theo các công thức như sau:

$$Precision = \frac{|A \cap B|}{|A|} \quad (11)$$

$$Recall = \frac{|A \cap B|}{|B|} \quad (12)$$

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (13)$$

Các tham số A và B ở công thức trên tương ứng là tập các nhãn được phát hiện và tập các nhãn đúng (được gán nhãn bởi người gán nhãn). Thử nghiệm được thực hiện với việc huấn luyện mô hình dựa trên tập dữ liệu huấn luyện (train), tối ưu mô hình dựa trên tập dữ liệu đánh giá (valid) và đánh giá mô hình dựa trên tập dữ liệu kiểm tra (test).

Trong các mô hình GNN thử nghiệm, chúng tôi thử nghiệm với các số lượng lớp tích chập khác nhau trong khoảng giá trị [2,16] và thấy rằng khi tăng số lớp tích chập thì hiệu năng cũng tăng lên. Tuy nhiên hầu hết các mô hình khi tăng lên 12 lớp trở lên thì hiệu suất hầu như tăng không đáng kể, dù chi phí tính toán rất lớn. Vấn đề này cũng xảy ra tương tự như với các thử nghiệm trong các nghiên cứu khác [12]. Do đó tham số về số lớp tích chập được đặt là 12 cho tất cả các mô hình GNN.

Các thực nghiệm đều sử dụng bộ tối ưu Adam optimizer với tham số *weight decay* là $1e^{-5}$. Hàm mất mát được sử dụng là CrossEntropyLoss để phân loại các nút. Để chọn các giá trị phù hợp cho kích thước vector đặc trưng của nút, tốc độ học (learning rate), chúng tôi đã thực hiện nhiều thử nghiệm trong khi giữ các tham số khác không đổi và điều chỉnh một tham số cụ thể. Kết quả cho thấy mô hình đạt hiệu suất tối ưu khi kích thước vector đặc trưng của nút 32, tốc độ học là 0,001. Chúng tôi cũng áp dụng cơ chế dừng sớm, quá trình huấn luyện sẽ dừng khi hiệu suất trên tập dữ liệu kiểm thử không được cải thiện nào trong ít nhất 5 epoch liên tiếp.

Đối với mô hình GCN, các lớp tích chập là lớp GCNConv. Bài báo sử dụng GCN với chuẩn hóa bước ngẫu nhiên (random walk), là cách chuẩn hóa tập trung vào việc tính xác suất chuyển tiếp từ một nút sang nút khác trong đồ thị, tương tự như bước ngẫu nhiên, với công thức là $\tilde{D}^{-1}\tilde{A}$. Đối với mô hình GIN có các lớp tích chập là GINConv, chúng tôi sử dụng số lớp MLP là 4 sau các thử nghiệm tối ưu. Mô hình GATv2 (lớp tích chập sử dụng là GATv2ConvLayer) và mô hình UniMP (lớp tích chập sử dụng là UniMPConvLayer) đều thuộc loại mô hình GCN kết hợp cơ chế tập trung, do đó các thiết lập chủ yếu liên quan đến số đầu tập trung (head attention) sử dụng và phương pháp tổng hợp (Agregation). Phương pháp tổng hợp được lựa chọn là lấy trung bình giá trị. Số đầu tập trung được sử dụng đối với cả hai mô hình này đều là 8 sau các thử nghiệm tối ưu, tương tự như trong nghiên cứu [12].

Thư viện được sử dụng để tạo các lớp tích chập cho GNN là PyG [29]. Thư viện PyG cung cấp lớp MessagePassing cơ sở, giúp tạo ra các mạng đồ thị nơ-ron truyền thông điệp như vậy bằng cách tự động xử lý quá trình truyền thông điệp.

C. Kết quả thực nghiệm

Phần dưới đây sẽ mô tả các thực nghiệm để đánh giá hiệu năng của các mô hình phát hiện botnet đã đề xuất thử nghiệm trên các tập dữ liệu tiêu chuẩn, khi so sánh với các mô hình cơ sở khác.

1) Đánh giá hiệu năng của các mô hình trên các bộ dữ liệu mô phỏng

Thử nghiệm đầu tiên được thực hiện trên các bộ dữ liệu mô phỏng. Trong thử nghiệm này, nghiên cứu sử dụng 2 mô hình GNN điển hình là GCN và GIN để thực hiện các tác vụ phát hiện bot. Các bộ dữ liệu được sử dụng trong phần này chủ yếu là các bộ dữ liệu mô phỏng Chord, De Bruijn, Kademia, LEET-Chord [12].

Kết quả thử nghiệm trong Bảng I cho thấy, cả 2 mô hình GNN đều có khả năng phát hiện botnet rất hiệu quả, với giá trị F_1 đều cho kết quả lớn hơn 98,6% trên các bộ dữ liệu. Đồng thời, thử nghiệm cũng cho thấy mô hình đề xuất sử dụng GIN có hiệu năng tốt hơn hẳn mô hình sử dụng GCN đề xuất trong [12] trên các tập dữ liệu mô phỏng. Điều này cũng chứng tỏ mô hình GIN có khả năng học các cấu trúc hình học của đồ thị tốt hơn, trong trường hợp các cấu trúc mạng rõ ràng như dữ liệu mô phỏng.

Bảng I. So sánh hiệu năng của mô hình GIN và GCN trên các bộ dữ liệu mô phỏng

Mô hình / Bộ dữ liệu	***F*$_1$ với mô hình GCN** [12] **(%)**	***F*$_1$ với mô hình GIN**
Chord	98,66	**99,45**
De Brujin	99,93	**99,95**
Kademlia	99,01	**99,48**
LEET-Chord	99,27	**99,58**

2) Đánh giá hiệu năng của các mô hình trên các bộ dữ liệu thực

Trong thử nghiệm thứ 2, nghiên cứu trình bày kết quả đạt được khi thử nghiệm với các mô hình GNN tiên tiến như GIN, GATv2, và UniMP và so sánh với mô hình GCN trong nghiên cứu [12]. Bộ dữ liệu sử dụng trong phần này là 2 bộ dữ liệu thực gồm C2 và P2P.

Để đánh giá được các mô hình tích hợp cơ chế tập trung như GATv2 và UniMP, chúng tôi bổ sung thêm đặc trưng cho các nút dựa trên giá trị mức của mỗi nút.

Kết quả thử nghiệm trong Bảng II cho thấy, kể cả đối với các bộ dữ liệu botnet thực tế như C2 (tập trung) và P2P (không tập trung, phổ biến trong thực tế) thì các mô hình GNN đều có khả năng phát hiện botnet rất hiệu quả và có tiềm năng rất lớn trong ứng dụng thực tiễn. Giá trị F_1 của tất cả các mô hình trên cả hai bộ dữ liệu thử nghiệm đều cho kết quả trên 98,9% trở lên. Mô hình GIN vẫn là mô hình cho hiệu năng cao nhất, với giá trị F_1 trên 99,5%. Tiếp sau đó là các mô hình sử dụng cơ chế tập trung là GATv2 và UniMP cũng cho kết quả khá tốt, với độ tăng khoảng 0,5% so với kết quả của mô hình GCN. Kết quả này là chấp nhận được do các bộ dữ liệu không có nhiều đặc trưng trên các nút, cạnh.

Bảng II. So sánh hiệu năng của các mô hình GNN trên hai bộ dữ liệu thực tế

Mô hình	***F*$_1$ *trên bộ dữ liệu C2 (%)***	***F*$_1$ *trên bộ dữ liệu P2P (%)***
GCN [57]	98,93	99,14
GIN	**99,54**	**99,57**
GATv2	99,48	99,55
UniMP	99,31	99,44

3) Đánh giá khả năng tiết kiệm bộ nhớ của các mô hình GNN

Các mô hình như GIN và nhất là các mô hình dựa trên cơ chế tập trung như GATv2 và UniMP thường cần nhiều tài nguyên và không phù hợp với các đồ thị lớn. Do đó, các mô hình này thường được kết hợp với kiến trúc kết nối dư nghịch đảo để tiết kiệm tài nguyên và đảm bảo hiệu năng phát hiện botnet. Trong thử nghiệm này, chúng tôi thử nghiệm mô hình RevGIN (GIN với kết nối dư nghịch đảo) và so sánh hiệu năng và tài nguyên sử dụng với mô hình GIN trên bộ dữ liệu P2P.

Bảng III. Hiệu năng của các kết hợp đặc trưng khác nhau trong mô hình đề xuất

Mô hình	**Số lớp**	***F*$_1$ (%)**	**Bộ nhớ GPU (GB)**	**Số tham số (nghìn)**
GIN	6	99,38	3,9	18
RevGIN	6	99,46	1,9	19
GIN	12	99,52	5,9	37
RevGIN	12	99,43	2,2	38
GIN	24	99,54	9,91	74
RevGIN	24	99,44	2,28	76

Kết quả thử nghiệm trong Bảng III cho thấy, khi tăng số lượng lớp trong mô hình GIN, nhu cầu về dung lượng RAM của GPU tăng lên nhanh chóng do số tham số rất lớn. Ngược lại, với RevGIN, dù tăng số lượng lớp, dung lượng RAM của GPU cần sử dụng vẫn gần như không đổi. Nhờ khả năng tiết kiệm bộ nhớ, RevGIN có thể xây dựng các mô hình có độ sâu lớn hơn nhiều so với GNN truyền thống, giúp mô hình học được các đặc trưng phức tạp hơn của dữ liệu. Yếu điểm duy nhất là khi thêm các kết nối dư nghịch đảo, mô hình RevGIN tốn nhiều thời gian huấn luyện hơn so với GIN. Do đó, tùy thuộc vào điều kiện thực tế, nếu ưu tiên độ sâu của mô hình và khả năng học các đặc trưng phức tạp, RevGIN là một lựa chọn tốt. Ngược lại, nếu ưu tiên tốc độ và tiết kiệm tài nguyên, GIN sẽ phù hợp hơn.

V. KẾT LUẬN

Nghiên cứu này đã chứng minh được tiềm năng to lớn của mạng nơ-ron đồ thị (GNN) trong lĩnh vực phát hiện botnet, cụ thể là về khả năng học hiệu quả các đặc trưng cấu trúc liên kết đồ thị của mạng botnet. Các mô hình GNN, bao gồm GCN, GAT, UniMP và GIN, đã được đánh giá trên các tập dữ liệu botnet mô phỏng (Chord, De Bruijn, Kademlia, Leet-Chord) và thực tế (C2, P2P). Kết quả cho thấy tất cả các mô hình đều đạt hiệu suất cao, với giá trị F_1 trên 98,6% trên tất cả các bộ dữ liệu. Đặc biệt, GIN thể hiện khả năng học cấu trúc liên kết đồ thị tốt hơn, đạt hiệu năng cao nhất trên các bộ dữ liệu. Các mô hình GATv2 và UniMP, tích hợp cơ chế tập trung, cũng cho kết quả khả quan, khẳng định tiềm năng ứng dụng thực tiễn của chúng. Bên cạnh đó, RevGIN, kết hợp GIN với kiến trúc kết nối dư nghịch đảo, cho phép xây dựng mô hình học sâu hơn và tiết kiệm bộ nhớ đáng kể, mở ra hướng nghiên cứu mới cho việc phát hiện botnet trên các mạng quy mô lớn. Nghiên cứu này không chỉ khẳng định hiệu quả của GNN trong việc phát hiện botnet mà còn cung cấp kết quả so sánh hiệu năng giữa các mô hình GNN khác nhau, góp phần định hướng cho các nghiên cứu tiếp theo trong lĩnh vực an ninh mạng.

Trong nghiên cứu tới, chúng tôi sẽ phát triển các mô hình mạng nơ-ron đồ thị có khả năng học kết hợp đặc trưng cấu trúc liên kết của đồ thị và các dữ liệu khác có trong luồng mạng, từ đó nâng cao khả năng phát hiện botnet trong mạng.

TÀI LIỆU THAM KHẢO

[1] Nuspire, "Q3 2023 Cyber Threat Report." Accessed: Nov. 24, 2024. [Online]. Available: https://www.nuspire.com/resources/q3-2023-cyber-threat-report/

[2] Sven Dummer and Sandeep Rath, "A Retrospective on DDoS Trends in 2023 and Actionable Strategies for 2024." Accessed: Nov. 24, 2024. [Online]. Available: https://www.akamai.com/blog/security/a-retrospective-on-ddos-trends-in-2023

[3] Nokia, "Nokia Threat Intelligence Report finds malicious IoT botnet activity has sharply increased." Accessed: Nov. 24, 2024. [Online]. Available: https://www.nokia.com/about-us/news/releases/2023/06/07/nokia-threat-intelligence-report-finds-malicious-iot-botnet-activity-has-sharply-increased/

[4] W. Zhang, B. Zhang, Y. Zhou, H. He, and Z. Ding, "An IoT Honeynet Based on Multiport Honeypots for Capturing IoT Attacks," IEEE Internet Things J, vol. 7, no. 5, 2020, doi: 10.1109/JIOT.2019.2956173.

[5] R. Gandhi and Y. Li, "Comparing Machine Learning and Deep Learning for IoT Botnet Detection," in Proceedings - 2021 IEEE International Conference on Smart Computing, SMARTCOMP 2021, 2021. doi: 10.1109/SMARTCOMP52413.2021.00053.

[6] A. Gangone, B. Bala, S. Gangone, and G. J. Bharat Kumar, "The Deep Learning and Machine Learning Methods for Botnet Identification in the Internet of Things," in Proceedings of International Conference on Contemporary Computing and Informatics, IC3I 2023, 2023. doi: 10.1109/IC3I59117.2023.10397881.

[7] A. K. Karthick Kumar, K. Vadivukkarasi, and R. Dayana, "A Novel Hybrid Deep Learning Model for Botnet Attacks Detection in a Secure IoMT Environment∗," in Proceedings of the 2023 International Conference on Intelligent Systems for Communication, IoT and Security, ICISCoIS 2023, 2023. doi: 10.1109/ICISCoIS56541.2023.10100396.

[8] S. Pouyanfar et al., "A survey on deep learning: Algorithms, techniques, and applications," 2018. doi: 10.1145/3234150.

[9] T. Bilot, N. El Madhoun, K. Al Agha, and A. Zouaoui, "Graph Neural Networks for Intrusion Detection: A Survey," IEEE Access, vol. 11, 2023, doi: 10.1109/ACCESS.2023.3275789.

[10] S. Khoshraftar and A. N. Aijun, "A Survey on Graph Representation Learning Methods," ACM Trans Intell Syst Technol, vol. 15, no. 1, Jan. 2024, doi: 10.1145/3633518.

[11] Y. Shi, Z. Huang, S. Feng, H. Zhong, W. Wang, and Y. Sun, "Masked Label Prediction: Unified Message Passing Model for Semi-Supervised Classification," in IJCAI International Joint Conference on Artificial Intelligence, 2021. doi: 10.24963/ijcai.2021/214.

[12] J. Zhou, Z. Xu, A. M. Rush, and M. Yu, "Automating botnet detection with graph neural networks," arXiv preprint arXiv:2003.06344, 2020.

[13] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," Future Generation Computer Systems, vol. 100, 2019.

[14] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in 2015 Military Communications and Information Systems Conference, MilCIS 2015 - Proceedings, 2015.

[15] H. V. Le and Q. D. Ngo, "V-Sandbox for Dynamic Analysis IoT Botnet," IEEE Access, vol. 8, 2020, doi: 10.1109/ACCESS.2020.3014891.

[16] S. Sriram, R. Vinayakumar, M. Alazab, and K. P. Soman, "Network flow based IoT botnet attack detection using deep learning," in IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS 2020, 2020. doi: 10.1109/INFOCOMWKSHPS50562.2020.9162668.

[17] E. B. Hadipuspito and V. Suryani, "Hybrid Deep Learning for Botnet Attack Detection on IoT Networks using CNN-GRU," in 2024 International Conference on Data Science and Its Applications (ICoDSA), 2024, pp. 133–139.

[18] B. Zhang, J. Li, C. Chen, K. Lee, and I. Lee, "A Practical Botnet Traffic Detection System Using GNN," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2022. doi: 10.1007/978-3-030-94029-4_5.

[19] K. Xu, S. Jegelka, W. Hu, and J. Leskovec, "How powerful are graph neural networks?," in 7th International Conference on Learning Representations, ICLR 2019, 2019.

[20] J. Zhou et al., "Graph neural networks: A review of methods and applications," 2020. doi: 10.1016/j.aiopen.2021.01.001.

[21] N. Shervashidze, P. Schweitzer, E. J. Van Leeuwen, K. Mehlhorn, and K. M. Borgwardt, "Weisfeiler-Lehman graph kernels," Journal of Machine Learning Research, vol. 12, 2011.

[22] P. Veličković, A. Casanova, P. Liò, G. Cucurull, A. Romero, and Y. Bengio, "Graph attention networks," in 6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings, 2018. doi: 10.1007/978-3-031-01587-8_7.

[23] S. T. Luu, K. Van Nguyen, and N. L.-T. Nguyen, "A large-scale dataset for hate speech detection on vietnamese social media texts," in Advances and Trends in Artificial Intelligence. Artificial Intelligence Practices: 34th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2021, Kuala Lumpur, Malaysia, July 26–29, 2021, Pro, 2021, pp. 415–426.

[24] I. Stoica et al., "Chord: A scalable peer-to-peer lookup protocol for Internet applications," IEEE/ACM Transactions on Networking, vol. 11, no. 1, 2003, doi: 10.1109/TNET.2002.808407.

[25] M. Jelasity and V. Bilicki, "Towards automated detection of peer-to-peer botnets: On the limits of local approaches," in LEET 2009 - 2nd USENIX Workshop on Large-Scale Exploits and Emergent Threats: Botnets, Spyware, Worms, and More, 2009.

[26] M. F. Kaashoek and D. R. Karger, "Koorde: A simple degree-optimal distributed hash table," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 2735, 2003, doi: 10.1007/978-3-540-45172-3_9.

[27] P. Maymounkov and D. Mazières, "Kademlia: A peer-to-peer information system based on the XOR metric," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 2429, 2002, doi: 10.1007/3-540-45748-8_5.

[28] S. García, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," Comput Secur, vol. 45, 2014, doi: 10.1016/j.cose.2014.05.011.

[29] M. Fey and J. E. Lenssen, "Fast Graph Representation Learning with PyTorch Geometric," in ICLR Workshop on Representation Learning on Graphs and Manifolds, 2019.

[30] F. Falchi, C. Gennaro, and P. Zezula, "A content-addressable network for similarity search in metric spaces," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2007. doi: 10.1007/978-3-540-71661-7_9.

[31] J. Quitzau and J. Stoye, "A space efficient representation for sparse de Bruijn subgraphs," bieson.ub.uni-bielefeld.de, 2008.

AN EMPIRICAL STUDY OF GNN MODELS FOR DETECTING BOTNET

Abstract: Graph Neural Networks (GNNs) have emerged as a powerful approach for botnet detection, exhibiting superior performance compared to traditional

machine learning and deep learning methods. This stems from their inherent ability to effectively learn and exploit the intricate relationships present within network data by representing it in the form of graphs. This paper specifically focuses on the investigation and evaluation of several state-of-the-art GNN models, including GCN, GAT, UniMP, and GIN, for the purpose of botnet detection, utilizing both simulated and real-world datasets. The experimental results demonstrate that all GNN models achieve high levels of effectiveness, consistently achieving an F1 score above 98.6% across all datasets. This strongly suggests the significant potential of GNNs for practical applications in this domain. Notably, the GIN model demonstrates a superior ability to learn graph structures, consistently attaining the highest performance on all simulated and real-world datasets. Models that incorporate attention mechanisms, such as GATv2 and UniMP, also yield promising results, further highlighting the benefits of such approaches. Additionally, the RevGIN model, which combines the GIN architecture with invertible residual connections, enables the development of deeper models while simultaneously achieving significant reductions in memory consumption. This innovative approach presents a new research direction for botnet detection in large-scale networks, paving the way for more efficient and scalable solutions.

Keywords: Cyberattacks, botnet, deep learning, graph neural networks.

Nguyễn Ngọc Điệp. Nhận học vị Tiến sĩ năm 2017. Hiện đang công tác tại Khoa An toàn thông tin, Lab Học máy và ứng dụng, Học viện Công nghệ Bưu chính Viễn thông. Lĩnh vực nghiên cứu: học máy, an toàn thông tin, xử lý ngôn ngữ tự nhiên.

Nguyễn Thị Thanh Thủy. Nhận học vị Tiến sĩ năm 2023. Hiện đang công tác tại Khoa Công nghệ Thông tin 1 và Lab Học máy và ứng dụng, Học viện Công nghệ Bưu chính Viễn thông. Lĩnh vực nghiên cứu: Trí tuệ nhân tạo, học máy, xử lý ngôn ngữ tự nhiên.