# CẢI TIẾN PHÁT HIỆN TẤN CÔNG SỬ DỤNG VĂN PHẠM NỐI CÂY TRONG LẬP TRÌNH GEN

**Vũ Văn Cảnh[a,b*], Hoàng Tuấn Hảo[a], Nguyễn Văn Hoàn[b]**

*[a]Khoa Công nghệ Thông tin, Trường Đại học Kỹ thuật Lê Quý Đôn, Hà Nội, Việt Nam*
*[b]Khoa Công nghệ Thông tin, Trường Đại học Thông tin Liên Lạc, Khánh Hòa, Việt Nam*

**Lịch sử bài báo**
Nhận ngày 07 tháng 01 năm 2017 | Chỉnh sửa ngày 13 tháng 07 năm 2017
Chấp nhận đăng ngày 20 tháng 07 năm 2017

**Tóm tắt**

*Những năm gần đây vấn đề an ninh mạng đã trở nên cấp thiết và tác động lớn tới hiệu quả hoạt động của các mạng máy tính hiện đại. Phát hiện và ngăn chặn tấn công mạng máy tính đã và đang là chủ điểm nghiên cứu của nhiều nhà nghiên cứu trên thế giới. Một trong những biện pháp bảo đảm an toàn cho các hệ thống mạng là Hệ thống phát hiện xâm nhập trái phép. Tuy nhiên, các biện pháp này tỏ ra không hiệu quả và khá tốn kém, độ tin cậy không cao và không có khả năng phát hiện các tấn công, xâm nhập mới, chưa biết trước dấu hiệu. Kỹ thuật học máy được sử dụng trong việc phát hiện các tấn công, xâm nhập đã khắc phục được các hạn chế trên và ngày càng thể hiện tính ưu việt hơn các phương pháp trước. Trong bài báo này, chúng tôi sử dụng kỹ thuật lập trình Gen (Genetic Programming - GP) để cải thiện chất lượng phát hiện tấn công mạng. Trong thí nghiệm, chúng tôi sử dụng GP chuẩn và kỹ thuật văn phạm nối cây (TAG3P), tiến hành trên bộ dữ liệu nhân tạo do nhóm tác giả Pham, Nguyen, và Nguyen (2014) đề xuất. Trên cơ sở các kết quả thí nghiệm và so sánh với một số kỹ thuật đã được đề xuất trước, chúng tôi nhận thấy ứng dụng GP và TAG3P trong phát hiện tấn công đạt hiệu quả tốt hơn các phương pháp trước đó.*

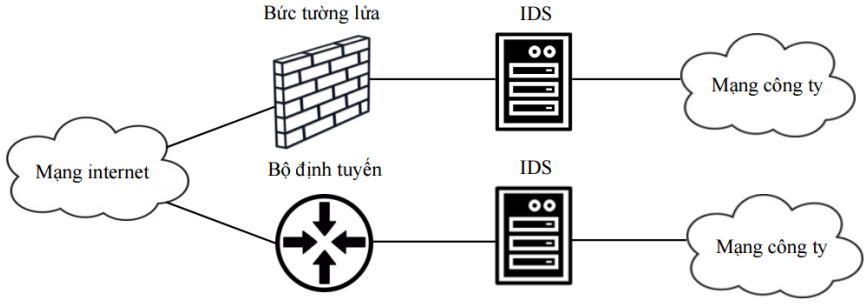
**Từ khóa**: Lập trình Gen; Phát hiện xâm nhập; Phân loại tấn công; Văn phạm nối cây.

## 1. GIỚI THIỆU CHUNG

Ngày nay mạng máy tính đã trở thành một phần của cuộc sống hiện đại và ngày càng đóng vai trò quan trọng trong hầu hết các lĩnh vực của cuộc sống từ kinh tế, chính trị, quân sự, các lĩnh vực giải trí đến giáo dục và đào tạo… Cùng với sự phát triển của mạng máy tính, nguy cơ mất an toàn, an ninh đối với các thông tin ngày càng cao. Ngày càng có nhiều tấn công vào không gian mạng để truy cập trái phép vào thông tin và hệ thống, hoặc lạm dụng các tài nguyên mạng. Việc lạm dụng có thể dẫn tới hậu quả khiến cho tài nguyên mạng trở lên không đáng tin cậy hoặc không sử dụng được. Một số cuộc

* Tác giả liên hệ: Email: canhvuvan@yahoo.com

tấn công có thể dẫn đến phá hủy hệ thống, hoặc đánh cắp thông tin, hay làm ngừng hoạt động của hệ thống. Nhìn chung các tấn công thường gây nên tổn thương đến các thuộc tính bảo mật thông tin và hệ thống. Vì vậy, vấn đề đảm bảo an ninh, an toàn thông tin khi sử dụng môi trường mạng cần phải được đặc biệt quan tâm. Phát hiện tấn công, xâm nhập mạng là một vấn đề lớn đã và đang được nhiều nhà nghiên cứu quan tâm. Trong thực tế, có khá nhiều nguy cơ xuất phát từ các cuộc tấn công mạng. Vì vậy, các hệ thống khác nhau đã được thiết kế và xây dựng để ngăn cản các cuộc tấn công này, đặc biệt là các hệ thống phát hiện xâm nhập (*Intrusion Detection System - IDS*) giúp các mạng chống lại các cuộc tấn công từ bên ngoài. Mục tiêu của IDS là cung cấp một bức tường bảo vệ, giúp các hệ thống mạng có khả năng chống lại các cuộc tấn công từ bên ngoài. Các IDS có thể được sử dụng để phát hiện việc sử dụng các loại truyền thông mạng và hệ thống máy tính độc hại, nhiệm vụ mà các bức tường lửa quy ước không thể thực hiện được. Devarakonda và Pamidi (2012) đã đề xuất việc phát hiện tấn công dựa trên giả thiết là hành vi của kẻ tấn công khác với người sử dụng hợp lệ. Phát hiện xâm nhập được triển khai bởi một hệ thống phát hiện xâm nhập và ngày nay đã có nhiều hệ thống phát hiện xâm nhập thương mại hiệu quả. Hình 1 mô tả các vị trí điển hình của IDS trong một hệ thống mạng.




**Hình 1. Vị trí của các IDS trong giám sát mạng**

Hệ thống phát hiện tấn công là một công cụ giám sát các sự kiện diễn ra trong hệ thống mạng máy tính và phân tích chúng thành các dấu hiệu của các mối đe dọa an ninh. Một tấn công có thể gây ra từ bên trong hoặc bên ngoài của tổ chức. Tấn công từ bên trong là tấn công được khởi tạo bởi một thực thể bên trong vành đai an ninh (tay trong), nghĩa là thực thể được phép truy cập vào tài nguyên hệ thống nhưng sử dụng theo cách không được chấp nhận bởi người cấp quyền. Tấn công từ bên ngoài được khởi tạo từ bên ngoài vành đai an ninh bởi người dùng trái phép và không hợp pháp của hệ thống. Trên

mạng Internet luôn tiềm tàng những kẻ tấn công từ bên ngoài với phạm vi từ những kẻ tấn công nghiệp dư đến những tổ chức tội phạm, khủng bố quốc tế, và chính phủ thù địch.

Có hai nhóm hệ thống phát hiện tấn công là phát hiện lạm dụng và phát hiện bất thường. Hệ phát hiện lạm dụng thực hiện dò tìm tấn công qua việc so khớp với mẫu đã biết, và hệ thống phát hiện bất thường nhận dạng bất thường từ hành vi mạng bình thường. Hệ thống phát hiện lai là tổ hợp cả hệ thống phát hiện lạm dụng và bất thường.

Hệ thống phát hiện tấn công dựa trên sự bất thường cố gắng xác định độ lệch so với các mẫu sử dụng thông thường đã được thiết lập trước để đánh dấu các tấn công. Vì vậy, các hệ thống dựa trên sự bất thường cần được huấn luyện dựa trên các hành vi thông thường. Các kỹ thuật học máy khác nhau đã được sử dụng rộng rãi để phục vụ cho mục đích này. Khi đó, với mỗi gói tin bắt được, sau khi qua các công đoạn tiền xử lý và chọn lựa thuộc tính sẽ được phân lớp bởi các bộ phân lớp (*classifier*) đã được huấn luyện. Việc huấn luyện các bộ phân lớp được thực hiện qua pha huấn luyện và kiểm tra với tập dữ liệu huấn luyện đã lưu trữ.

Đã có nhiều kỹ thuật phát hiện tấn công đã được các học giả đề xuất như các phương pháp học máy, mạng nơ-ron… Trong bài viết này, chúng tôi trình bày các nghiên cứu về kỹ thuật lập trình Gen và phân tích các thuộc tính của các kiểu tấn công mạng để từ đó đề xuất ứng dụng lập trình Gen nhằm nâng cao khả năng phát hiện tấn công mạng. Bố cục bài báo được trình bày như sau: Sau Mục 1 giới thiệu, Mục 2 kiến thức nền tảng sẽ giới thiệu các công trình nghiên cứu trước đây, bộ dữ liệu huấn luyện KDD’99, tổng quan về lập trình Gen; Mục 3 giới thiệu mô hình đề xuất phát hiện tấn công dựa trên GP/TAG3P, cài đặt thử nghiệm và phân tích đánh giá các kết quả đạt được.

## 2. KIẾN THỨC NỀN TẢNG

### 2.1. Một số nghiên cứu trước đây

Hiện nay đã có nhiều nhà nghiên cứu đề xuất các giải pháp áp dụng kỹ thuật tính toán thông minh trong phát hiện tấn công mạng. Một số nghiên cứu sử dụng giải thuật di truyền (GA) và lập trình Gen (GP) để dò tìm các loại tấn công tấn công trong các kịch

bản khác nhau. Botha và Solms (2004); Leung, So và Yam (1992); Li (2004); và Peng, Leckie, và Kotagiri (2007) sử dụng GA và GP để tìm ra các quy tắc phân loại. Bridges và Vaughn (2000); Gomez và Dasgupta (2002); và Lu và Traore (2004) sử dụng GA và GP được sử dụng để chọn các đặc trưng yêu cầu và xác định các tham số tối ưu và tối thiểu của một số chức năng lõi trong những phương pháp tính toán thông minh khác để có thể tiếp nhận các quy tắc dò tìm tấn công.

Crosbie và Spafford (1995) đã đề xuất giải pháp sử dụng GA để phát hiện xâm nhập, áp dụng công nghệ đa tác nhân và sử dụng GP để phát hiện mạng bất thường thông qua việc giám sát một số tham số của dữ liệu dấu vết mạng. Các phương pháp đề xuất có lợi thế khi sử dụng nhiều tác nhân tự trị nhỏ nhưng khó khăn khi giao tiếp giữa các tác nhân và nếu khởi tạo không đúng tiến trình huấn luyện có thể ảnh hưởng lớn đến thời gian thực hiện.

Li (2004) đã đề xuất phương pháp sử dụng GA để phát hiện xâm nhập mạng dị thường, phương pháp này được sử dụng để định lượng và phân loại các đặc trưng của dữ liệu mạng nhằm mục tiêu tìm ra các quy tắc phân loại. Tuy nhiên, định lượng đặc trưng có thể làm tăng tốc độ tìm kiếm nhưng kết quả thí nghiệm không hiệu quả. Abdullah, Abd-alghafar, Gouda, và Abd-Alhafez (2009); và Anup và Chetan (2008) đề xuất thuật toán dựa trên GA để phân loại tất cả các loại tấn công Smurf sử dụng bộ dữ liệu huấn luyện với tỷ lệ phát hiện sai rất thấp (khoảng 0.2%) và tỷ lệ phát hiện hầu hết là 100%.

Lu và Traore (2004) sử dụng GP để phân loại tập dữ liệu lịch sử mạng, họ sử dụng nền tảng hỗ trợ tin cậy như hàm mục tiêu và phân loại chính xác một vài loại xâm nhập mạng. Tuy nhiên việc sử dụng GP của họ để tạo ra các thủ tục thực thi rất khó và thủ tục huấn luyện trên tập dữ liệu yêu cầu thời gian nhiều hơn. Wong, Leung, và Cheng (2000) đã sử dụng GA để phát hiện các hành vi mạng bất thường trên các thông tin lịch sử mạng. Một số đặc trưng mạng có thể được định nghĩa với các loại tấn công mạng dựa trên các thông tin tương hỗ giữa đặc trưng mạng và dạng tấn công, sau đó sử dụng những đặc trưng này để tạo ra các cấu trúc quy tắc tuyến tính cho GA. Phương pháp này sử dụng thông tin tương hỗ và kết quả quy tắc tuyến tính có hiệu quả trong nâng cao tỷ lệ phát hiện và giảm thời gian thực hiện, tuy nhiên họ chỉ coi các đặc trưng là rời rạc.

Gong, Zulkernine, và Abolmaesumi (2005) đề xuất sử dụng GA để thực hiện phát hiện tấn công mạng và đã đưa ra phần mềm thực thi với phương pháp tìm một tập quy tắc phân loại và sử dụng một nền tảng hỗ trợ tin cậy để xem xét hàm mục tiêu. Abdullah và ctg. (2009) đã sử dụng thuật toán đánh giá hiệu suất dựa trên GA để phát hiện xâm nhập mạng, phương pháp này sử dụng lý thuyết thông tin để lọc lưu lượng mạng. Faraoun, Boukelif, và Algeria (2006) đề xuất phương pháp phân loại tấn công sử dụng GP, kỹ thuật đề xuất bao gồm kết hợp tiến hóa của quần thể với sự chuyển đổi tuyến tính trên tập dữ liệu đầu vào được phân loại, sau đó ánh xạ chúng tới không gian mới với số chiều giảm để đạt được sự khác biệt tối đa giữa các lớp.

Ahmad, Hussain, Alghamdi, và Alelaiwi (2013) sử dụng kỹ thuật VSM để cải thiện hiệu suất của các kỹ thuật phát hiện tấn công bằng cách lựa chọn các đặc trưng với trị số đặc trưng cao như PCA (*Principal Component Analysis*), nghiên cứu này áp dụng GA để tìm kiếm các thành phần di truyền ban đầu mà có thể tạo ra một tập con đặc trưng với độ nhạy tối ưu và sự phân biệt cao nhất.

## 2.2. Bộ dữ liệu KDDCup 99

Năm 1999, Stolfo đề xuất bộ dữ liệu KDD'99 (UCI KDD Archive, 1999) dựa trên các dữ liệu bắt được bởi chương trình đánh giá hệ thống phát hiện xâm nhập DARPA'98. Bộ dữ liệu này gồm gần 5 triệu bản ghi, mỗi bản ghi có 41 thuộc tính và được gán nhãn là bình thường hay các dạng tấn công đặc trưng. KDD'99 đã được sử dụng rộng rãi để đánh giá các kỹ thuật phát hiện bất thường. Các dạng tấn công được phân thành các nhóm như sau:

- *Tấn công từ chối dịch vụ (DoS):* Là thủ đoạn nhằm ngăn cản người dùng hợp pháp truy cập và sử dụng vào một dịch vụ nào đó, DoS có thể làm ngưng hoạt động của hệ thống mạng, máy tính. Về bản chất nhằm chiếm dụng một lượng lớn tài nguyên mạng như băng thông, bộ nhớ... và làm mất khả năng xử lý các yêu cầu dịch vụ từ các khách hàng.

- *User to Root Attack (U2R):* Kẻ tấn công với quyền của một người dùng bình

thường cố gắng để đạt được quyền truy nhập cao nhất vào hệ thống một cách bất hợp pháp. Một cách phổ biến của lớp tấn công này là thực hiện bằng phương pháp gây tràn bộ đệm.

- *Remote to Local Attack (R2L):* Kẻ tấn công cố gắng đạt được quyền truy cập vào hệ thống máy tính bằng việc gửi các gói tin tới hệ thống thông qua mạng. Một vài cách phổ biến mà loại này thực hiện là đoán mật khẩu thông qua phương pháp từ điển Brute-force, FTP Write…

- *Probing Attack:* Kẻ tấn công thực hiện quét mạng hoặc máy tính để tìm ra điểm yếu dễ tấn công mà thông qua đó tin tặc có thể khai thác hệ thống. Một cách phổ biến của loại tấn công này là thực hiện thông qua việc quét các cổng của hệ thống máy tính.

Một số chuyên gia cho rằng hầu hết các tấn công mới đều là biến thể của các tấn công đã biết và các dấu hiệu của các tấn công đã biết có thể đủ để nhận dạng các biến thể mới. Bộ dữ liệu huấn luyện KDD'99 bao gồm 24 loại tấn công khác nhau (Bảng 1) và có thêm 14 loại tấn công mới được thêm vào trong bộ dữ liệu kiểm tra. Dựa vào các đặc trưng tấn công có thể phân loại KDD'99 thành các nhóm chính như sau:

- *Đặc trưng cơ bản:* Gồm tất cả các thuộc tính có thể có từ các kết nối TCP/IP.

- *Đặc trưng lưu lượng:* Gồm các đặc trưng được tính toán với mối liên hệ với khoảng thời gian.

- *Đặc trưng same host:* Chỉ kiểm tra các kết nối trong khoảng thời gian dưới 2 giây có cùng host đích như kết nối hiện hành và thống kê liên quan đến các hành vi giao thức, dịch vụ, …

- *Đặc trưng same service:* Chỉ kiểm tra những kết nối trong khoảng thời gian dưới 2 giây có cùng dịch vụ như kết nối hiện hành.

- *Đặc trưng nội dung:* Khác với hầu hết tấn công DoS, Probing, R2L và U2R không có bất cứ một mẫu tấn công nào. Bởi vì DoS và Probing liên quan đến

nhiều kết nối với một số host trong một khoảng thời gian rất ngắn, tuy nhiên tấn công R2L và U2R được nhúng trong đoạn gói dữ liệu và thường xuyên chỉ bao gồm một kết nối. Để phát hiện những loại tấn công này, cần một số đặc trưng để có thể tìm kiếm những hành vi nghi ngờ trong phần dữ liệu, chẳng hạn số lần cố gắng đăng nhập thất bại. Đây được gọi là đặc trưng nội dung.

Hai loại kể trên của đặc trưng lưu lượng được gọi dựa trên thời gian. Tuy nhiên, có một số tấn công thăm dò quét host (cổng) sử dụng khoảng thời gian lớn hơn 2 giây, có thể trong 1 phút. Kết quả là tấn công này không tạo ra các mẫu tấn công trong khoảng thời gian 2 giây.

**Bảng 1. Phân loại 24 loại tấn công trong KDDCup 99**

| Loại | Các tấn công trong bộ dữ liệu KDDCup 99 |
|---|---|
| Probe | Ipsweep, Nmap, Portsweep, Satan |
| DoS | Back, Land, Neptune, Pod, Smurf, Teardrop |
| U2R | Buffer_overflow, Loadmodule, Perl, Rootkit |
| R2L | Ftp_write, Guess_passwd, Imap, Multihop, Phf, Spy, Warezclient, Warezmaster |

## 2.3. Lập trình Gen

### *2.3.1. Thuật toán lập trình Gen*

Lập trình Gen (GP) là sự mở rộng của thuật toán di truyền (GA), đây là một phương pháp tìm kiếm tổng quát sử dụng phép loại suy từ chọn lọc tự nhiên và tiến hóa. Sự khác biệt chính giữa GP và GA là phương pháp mã hóa các giải pháp tìm kiếm, GA mã hóa các giải pháp tiềm năng cho vấn đề như một quần thể của các chuỗi nhị phân có chiều dài cố định gọi là nhiễm sắc thể, sau đó áp dụng các thao tác di truyền lên các nhiễm sắc thể này để tạo ra các nhiễm sắc thể mới. Ngược lại với GA, GP mã hóa các giải pháp đa tiềm năng cho các vấn đề cụ thể như là một quần thể của các chương trình hoặc các hàm, các chương trình có thể được biểu diễn dưới dạng cây phân tích cú pháp. Thông thường, cây phân tích cú pháp bao gồm các nút nội bộ và các nút lá. Các nút nội bộ được gọi là các nguyên hàm (*function*), và các nút lá được gọi là các ký hiệu kết thúc (*terminal*). Các *terminal* có thể được xem như là đầu vào cho các vấn đề cụ thể (các biến độc lập và

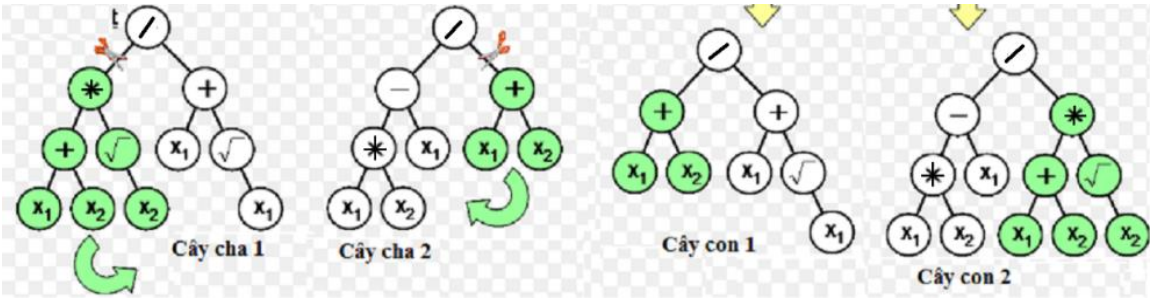
tập các hằng số). Các *function* có thể là các hàm toán học, các toán tử… Ví dụ, GP có thể được sử dụng để tiến hóa các quy tắc mới từ một quy tắc tổng quát, các quy tắc được biểu diễn dạng như *if* $condition_1$ *and* $condition_2$ *and... and* $condition_N$ *then attack*. Trong trường hợp này, các *function* tương ứng với toán tử *and* và các *terminal* là các *condition* (như: $condition_1$, $condition_2$… $condition_N$).

GP tạo ngẫu nhiên một quần thể của các giải pháp ban đầu, sau đó áp dụng các toán tử di truyền trên quần thể này để tạo ra quần thể mới. Các toán tử di truyền bao gồm tái sinh (*Reproduction*), lai ghép (*Crossover*), đột biến (*Mutation*), loại bỏ theo điều kiện (*Dropping condition*) … Quá trình tiến hóa từ quần thể này sang quần thể tiếp theo được gọi là thế hệ. Giải thuật GP có thể được mô tả tổng quát như sau:

- *Bước 1.* Tạo ngẫu nhiên một quần thể các chương trình, các quy tắc, sử dụng biểu thức hồi quy để cung cấp như khởi tạo quần thể ban đầu;
- *Bước 2.* Đánh giá độ thích nghi của mỗi chương trình, quy tắc bởi hàm thích nghi được định nghĩa để đo khả năng của quy tắc hoặc chương trình để giải quyết vấn đề;
- *Bước 3.* Sử dụng các toán tử tái sinh để chép chương trình hiện tại vào thế hệ mới;
- *Bước 4.* Tạo ra quần thể mới với các toán tử lai ghép, đột biến hoặc các toán tử khác từ một tập lựa chọn ngẫu nhiên của các cá thể cha mẹ;
- *Bước 5.* Lặp lại từ Bước 2 trở đi đối với quần thể mới cho đến khi thỏa mãn một tiêu chuẩn dừng đã được định nghĩa trước hoặc một số cố định các thế hệ đã được hoàn thành;
- *Bước 6.* Giải pháp cho vấn đề là chương trình di truyền với giá trị thích nghi cho tất cả các thế hệ.

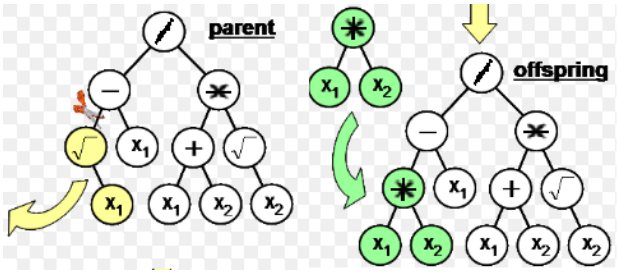
*2.3.2. Các toán tử di truyền*

Trong GP, để thực hiện toán tử lai ghép trước hết sao chép ngẫu nhiên hai cây cha mẹ từ quần thể ban đầu, sau đó hai điểm lai ghép sẽ được chọn ngẫu nhiên trên hai cây cha mẹ. Thực hiện hoán đổi hai nhánh con của hai cây cha mẹ tại các điểm đã được lựa chọn để tạo ra hai cây con. Cây con đạt được thường khác với cha mẹ chúng về kích thước và hình dáng. Hình 2 mô tả toán tử lai ghép giữa đa thức $\frac{(x_1+x_2)*\sqrt{x_2}}{x_1+\sqrt{x_1}}$ và $\frac{x_1*x_2-x_1}{x_1+x_2}$, kết quả thu được hai đa thức con mới là $\frac{(x_1+x_2)}{x_1+\sqrt{x_1}}$ và $\frac{x_1*x_2-x_1}{x_1+x_2*\sqrt{x_2}}$.




**Hình 2. Sử dụng toán tử lai ghép trong GP**

Trong toán tử đột biến, một cây cha/mẹ sẽ được sao chép từ quần thể ban đầu, sau đó chọn ngẫu nhiên một điểm đột biến (nút lá hoặc cây con). Sau đó, nút lá hoặc cây con được thay thế bởi một nút lá mới hoặc cây con được tạo ngẫu nhiên. Hình 3 mô tả một thao tác đột biến trên đa thức $\frac{\sqrt{x_1}-x_1}{(x_1+x_2)*\sqrt{x_2}}$ kết quả sau khi đột biến là $\frac{x_1*x_2-x_1}{(x_1+x_2)*\sqrt{x_2}}$.




**Hình 3. Sử dụng toán tử đột biến trong GP**

Toán tử "*dropping condition*" được đề xuất để tiến hóa quy tắc mới, toán tử này sẽ được lựa chọn ngẫu nhiên điều kiện trong quy tắc và sau đó thay đổi thành bất kỳ, như

vậy điều kiện này sẽ không cần thiết phải xem xét lại trong quy tắc đã chọn nữa. Ví dụ, quy tắc: *if* $condition_1$ *and* $condition_2$ *and* $condition_3$ *then attack* có thể đổi thành: *if* $condition_1$ *and* $condition_2$ *and any then attack.*

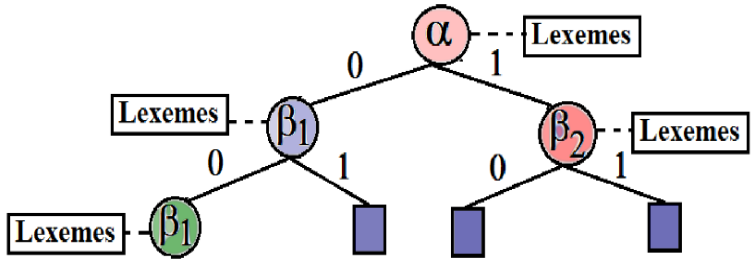
*2.3.3. Hàm thích nghi*

Để lựa chọn các cá thể cho thao tác lai ghép, tái tạo và đột biến, cũng như đánh giá độ thích nghi của từng cá thể trong việc giải quyết bài toán, hàm tính giá trị thích nghi là phương pháp để đánh giá độ thích nghi của từng cá thể trong quần thể. Hàm thích nghi nhằm đảm bảo cho sự tiến hóa hướng tới tối ưu bằng cách tính toán giá trị thích nghi cho mỗi cá thể trong quần thể. Giá trị thích nghi đánh giá hiệu suất của mỗi cá thể trong quần thể tại mỗi thế hệ. Độ thích nghi này được xác định trên cơ sở đánh giá chương trình so với kết quả tập dữ liệu được huấn luyện. Độ tốt của mỗi cá thể thường được chuẩn hóa trước khi được lựa chọn cho các phép toán di truyền (Koza, 1992).

## 2.4. Lập trình Gen định hướng bởi văn phạm nối cây

Hệ lập trình Gen định hướng bởi văn phạm nối cây (TAG3P) sử dụng văn phạm nối cây cùng với văn phạm phi ngữ cảnh để tạo ra những ràng buộc về cú pháp cũng như độ sai lệch khi tìm kiếm của chương trình tiến hóa. TAG3P bao gồm tất cả các thuộc tính của GP chuẩn dựa trên các biểu diễn dạng hình cây khác.

Trong TAG3P, cấu trúc văn phạm được xác định bằng tập hợp các cây $\alpha$ và cây $\beta$ cấu trúc quần thể là các cây dẫn xuất từ văn phạm này. Việc lượng giá độ tốt của mỗi cá thể được thực hiện bằng cách tạo ra các cây dẫn xuất được tương ứng từ cây dẫn xuất TAG, sau đó đánh giá biểu thức trên cây dẫn xuất được. Không gian tìm kiếm do đó được xác định bằng văn phạm, tập hợp tất cả các cây biểu thức GP đều do văn phạm cho trước tạo ra với giới hạn về độ phức tạp của cây này. Tuy nhiên, đặc tính thứ nguyên không xác định của cây giúp kiểm soát một cách dễ dàng theo kích thước của cây, do đó, kích thước của cây được sử dụng để kiểm soát độ phức tạp của cây trong TAG3P thay vì theo chiều cao của cây như trong các hệ GP khác. Hình 4 mô tả một ví dụ về cây dẫn xuất. Tương tự như GP chuẩn, TAG3P cũng gồm có 5 thành phần: Biểu diễn chương trình; Khởi tạo quần thể; Hàm thích nghi; Toán tử di truyền; và Các tham số.



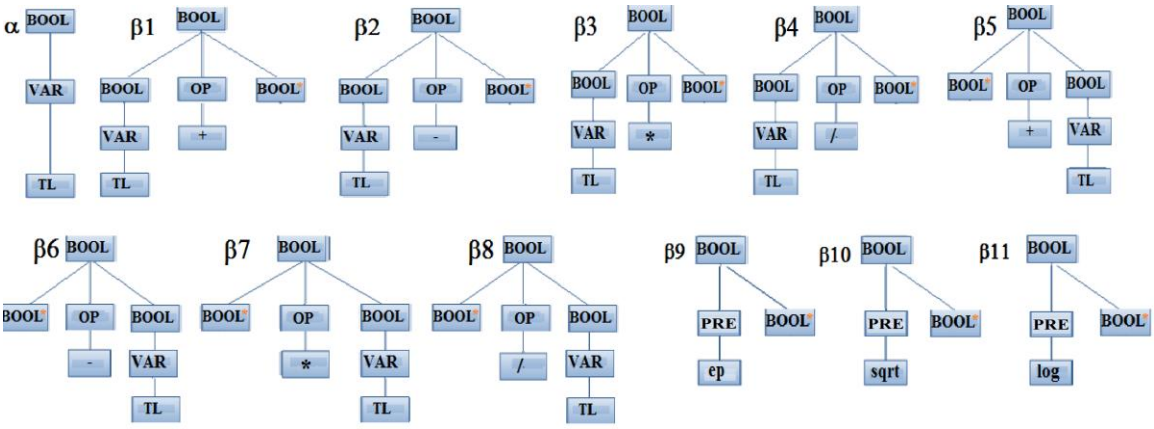

**Hình 4. Ví dụ về cây dẫn xuất**

*2.4.1. Biểu diễn chương trình*

TAG3P sử dụng sự chuyển đổi giữa kiểu Gen và kiểu hình, TAG3P có thể giải quyết bài toán với những ràng buộc cú pháp cảm ngữ cảnh, cú pháp phi ngữ cảnh hoặc không có ràng buộc về cú pháp. Do đó, kiểu hình có thể là một trong các trường hợp sau:

- Văn phạm LTAG $G_{lex}$ được sử dụng như là ngôn ngữ hình thức cho việc định nghĩa độ lệch, trong trường hợp này, kiểu hình là cây dẫn xuất của $G_{lex}$;
- Văn phạm phi ngữ cảnh (CFG) được sử dụng để tạo ra LTAG $G_{lex}$, khi đó, cây dẫn xuất của $G_{lex}$ sẽ được sử dụng là kiểu Gen, còn kiểu hình sẽ là cây dẫn xuất của G (cây dẫn xuất của văn phạm $G_{lex}$ - Xem Hình 5);
- Tập các hàm GP và ký hiệu kết được sử dụng để tạo ra văn phạm phi ngữ cảnh G = (N, T, P, {Bool}). Trong đó:
  - N={Bool, PRE, OP, VAR}: Tập các ký hiệu không kết thúc
  - T= {X, sin, cos, log, ep, +,-,*,/, (, )}: Tập các ký hiệu kết thúc
  - P = {Bool → Bool OP Bool, Bool → PRE (Bool), Bool → VAR, OP → +, OP → -, OP → *, OP → /, PRE → sin, PRE → cos, PRE → log, PRE → ep, VAR → TL}: Tập các luật dẫn xuất.

Từ đó ta có LTAG $G_{lex}$ được biểu diễn như sau: $G_{lex}$=[N={Bool, PRE, OP, VAR}, T={TL, sqrt, ep, log, +, -, *, /}, I, A] trong đó I∪A.



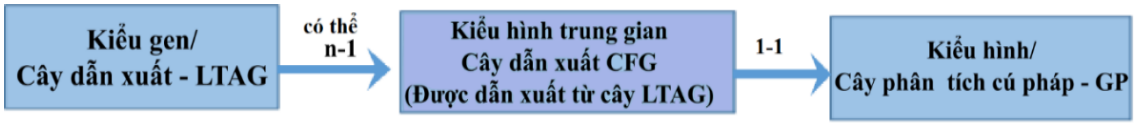

**Hình 5. Ví dụ các cây cơ sở của một văn phạm LTAG**

*2.4.2. Khởi tạo quần thể*

Chọn một số ngẫu nhiên trong khoảng cho trước, sau đó lấy ngẫu nhiên cây $\alpha$ từ tập cây cơ sở trong $G_{lex}$ để tạo cây dẫn xuất cho $G_{lex}$. Cây dẫn xuất này sẽ được mở rộng bằng phép nối với một cây $\beta$ được chọn ngẫu nhiên từ tập cây cơ sở. Quá trình này kết thúc khi kích thước quần thể đạt tới giá trị được chọn.

*2.4.3. Hàm thích nghi*

Để đánh giá sự thích nghi của các cá thể, trước hết chuyển các cá thể thành các cây dẫn xuất. Sau đó tính toán sự thích nghi của cá thể được thực hiện trên cây dẫn xuất (Hình 6).




**Hình 6. Quy trình chuyển đổi các cá thể thành cây dẫn xuất**

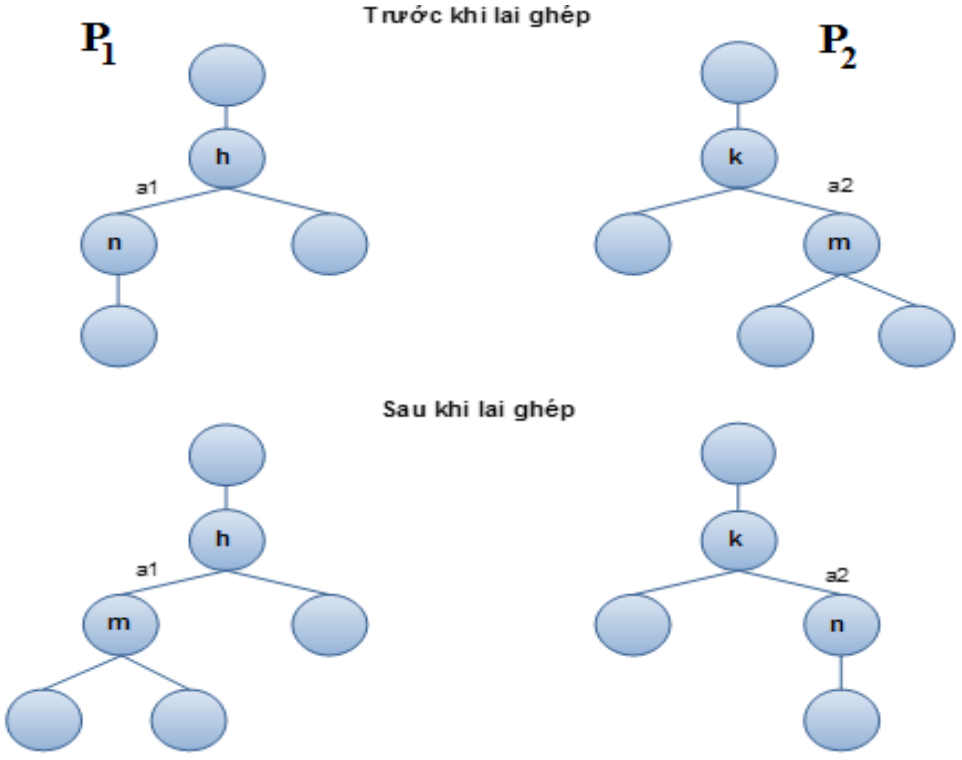
*2.4.4. Toán tử đi truyền*

TAG3P cũng có các toán tử di truyền chính như GP chuẩn đó là lựa chọn, tái tạo, lai ghép và đột biến.

- *Lựa chọn:* Trong TAG3P, các cơ chế lựa chọn đều có thể được sử dụng. Đặc biệt, cơ chế dựa trên độ thích nghi và lựa chọn cạnh tranh thường hay được sử dụng.

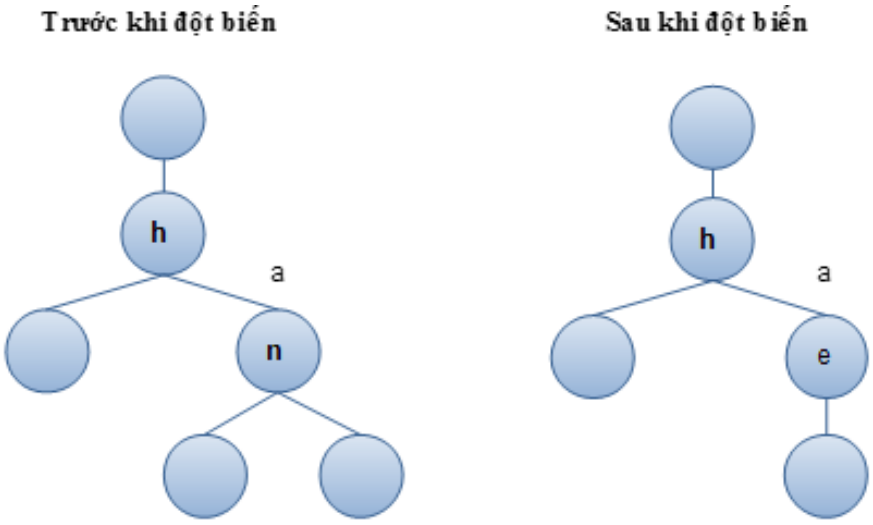
- *Tái tạo:* Một phần của quần thể được chọn dựa trên độ thích nghi và sao chép chúng vào trong thế hệ mới.

- *Lai ghép:* Tạo ra hai cá thể mới từ hai cá thể cha mẹ được lựa chọn từ quần thể dựa vào giá trị thích nghi. Đầu tiên, hai cây cha mẹ $P_1$ và $P_2$ được chọn thông qua cơ chế lựa chọn. Quá trình được thực hiện bằng cách chọn ngẫu nhiên hai nút tương thích từ hai cây cha mẹ, sau đó hoán đổi hai cây con của hai cây cha mẹ với nhau và thu được hai cây mới (Hình 7).




**Hình 7. Thao tác lai ghép trong TAG3P**

- *Đột biến:* Trong thao tác đột biến, một cây con được chọn ngẫu nhiên. Sau đó, cây con được loại bỏ và thay thế bởi cây con khác có cùng kích thước như trong Hình 8.

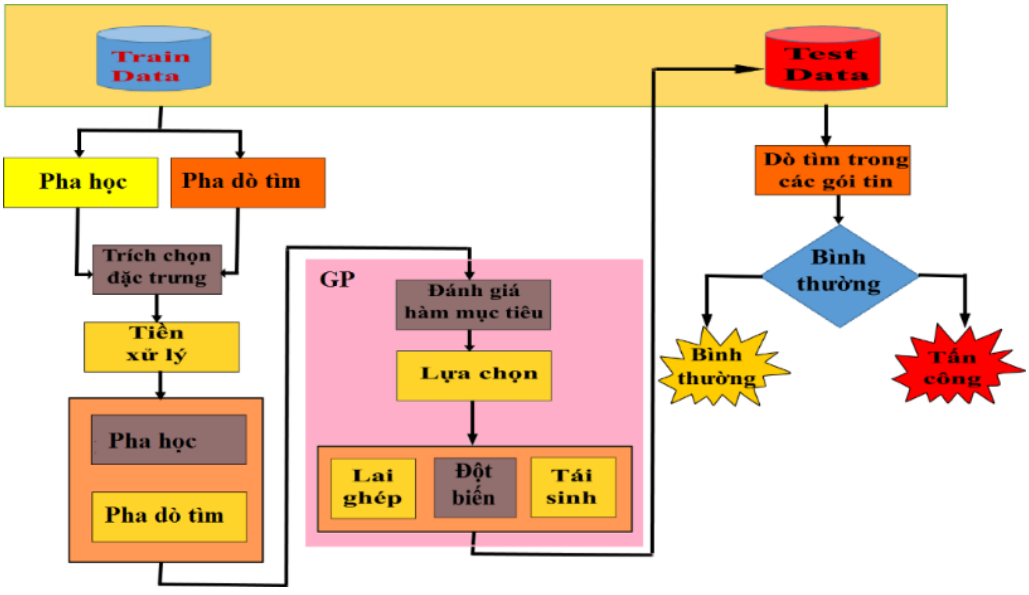



**Hình 8. Thao tác đột biến trong TAG3P**

## 3. HỆ THỐNG PHÁT HIỆN XÂM NHẬP DỰA TRÊN TAG3P

### 3.1. Mô hình phát hiện tấn công dựa trên lập trình Gen

Mô hình đề xuất bao gồm hai giai đoạn như được mô tả trong Hình 9. Trong giai đoạn huấn luyện, sử dụng dữ liệu dấu vết mạng để tạo ra tập các quy tắc phát hiện tấn công mạng. Giai đoạn 2 giá trị thích nghi cao và tập quy tắc tốt nhất được sử dụng để phát hiện các tấn công trên mạng.




**Hình 9. Mô tả thiết kế cho mô hình đề xuất phát hiện tấn công dựa trên GP**

Trong hình trên, GP thực hiện trên một cá thể và tiến hóa nhóm cá thể thành một

quần thể. Mỗi cá thể biểu diễn một kỹ thuật để giải quyết vấn đề. Một hàm thích nghi sẽ đánh giá cho mỗi quy tắc mà nó được thi hành. Sự tiến hóa quần thể bắt đầu từ quần thể được khởi tạo ban đầu bằng cách lựa chọn một cá thể mà cải thiện dần giá trị thích nghi của nó. Các toán tử di truyền như lựa chọn, lai ghép, đột biến và tái sinh được áp dụng cho mỗi cá thể suốt quá trình tạo ra thế hệ tiếp theo. Đầu tiên một số cá thể sẽ được lựa chọn dựa vào một chiến lược lựa chọn phù hợp, sau đó các cá thể sẽ được áp dụng các toán tử lại ghép, đột biến và tái sinh theo một tỷ lệ nhất định (tùy thuộc vào thí nghiệm). Cuối cùng các cá thể tốt nhất sẽ được lựa chọn để đưa vào thế hệ kế tiếp sao cho các cá thể này đảm bảo khả năng phát hiện tấn công từ quần thể đã được tạo ra trong thế hệ đó. Trong nghiên cứu này, chúng tôi sử dụng TAG3P để thực hiện phương pháp lựa chọn các toán tử di truyền thích nghi áp dụng cho hai toán tử di truyền là lai ghép và đột biến.

- *Lai ghép:* Hai cá thể được lựa chọn dựa trên giá trị thích nghi của chúng. Chọn ngẫu nhiên một điểm trên mỗi cây được chọn, tùy theo sự ràng buộc mỗi cây con có thể được nối với cây cha mẹ khác. Nếu điểm nối được tìm thấy, thì cây con này sẽ được nối với cha mẹ kia và ngược lại tại điểm kết nối, ngược lại hai cá thể này sẽ bị loại bỏ. Quá trình này sẽ được lặp lại cho đến khi một điểm lai ghép hợp lệ được tìm thấy hoặc vượt quá giới hạn;

- *Đột biến:* Chọn ngẫu nhiên một điểm trên cây đã chọn, sau đó tạo ngẫu nhiên một cây con mới để thay thế cây con tại điểm đã chọn trên cây cha/mẹ.

## 3.2. Cài đặt thử nghiệm

Nhóm tác giả đã tiến hành thử nghiệm phát hiện tấn công đối với mô hình đề xuất trên bộ dữ liệu nhân tạo do nhóm tác giả Pham, Nguyen, và Nguyen (2014) đề xuất với 10 thuộc tính cho mỗi loại tấn công. Thử nghiệm của chúng tôi đươc tiến hành tại Phòng Thí nghiệm An ninh mạng, Bộ môn An toàn Thông tin, Học viện Kỹ thuật Quân sự với các tham số di truyền được xác định như được trình bày trong Tiểu mục 3.2.1.

### *3.2.1. Các tham số và hàm mục tiêu*

- *Tham số:* Các tham số sử dụng trong quá trình tiến hóa để huấn luyện cho phát

hiện tấn công, xâm nhập được lựa chọn như trong Bảng 2.

**Bảng 2. Tập các tham số được lựa chọn trong quá trình tiến hóa**

| Tham số | Giá trị |
|---|---|
| Tỷ lệ lai ghép | 0.9 |
| Tỷ lệ đột biến | 0.1 |
| Kích thước quần thể | 200 |
| Số thể hệ thực hiện di truyền | 51 |
| Số mẫu dữ liệu huấn luyện | Phụ thuộc kịch bản |
| Số mẫu dữ liệu kiểm tra | Phụ thuộc kịch bản |
| Phương pháp lựa chọn | Lựa chọn cạnh tranh, size=10 |
| Tập Function | {add, sub, div, mul, sin, cos, log, ep} |
| Tập Terminal | x1, x2, …x10: 10 thuộc tính cho mỗi loại tấn công |
| Kích thước cá thể | MIN_SIZE=2, MAX_SIZE=40 |

- *Hàm thích nghi (fitness):* Giá trị thích nghi của mỗi cá thể sẽ được tính toán theo các bước như sau:

Tính thô (*rawfitness*):

$$\text{rawfitness(i)} = \sum_{i=1}^{\text{NumFitcase}} \frac{\left|f_i(x_1, x_2, ..., x_{10}) - y_i\right|}{\text{NumFitcase}} \quad (1)$$

Trong đó: *NumFitcase* là số mẫu trong bộ dữ liệu huấn luyện; $x_1$, $x_2$,. .., $x_{10}$ là thuộc tính lựa chọn cho kiểu tấn công; $f_i$ là hàm được xây dựng trong quá trình tiến hóa; và $y_i$ là giá trị phân loại mẫu dữ liệu là tấn công hay không tấn công.

Chuẩn hóa fitness tuần tự như sau:

$$\text{adjustfitness(i)} = \sum_{i=1}^{\text{poplen}} \frac{1}{1 + \text{rawfitness(i)}} \quad (2)$$

$$\text{nomalfitness(i)} = \frac{\text{adjustfitness(i)}}{\sum_{i=1}^{\text{polen}} \text{adjustfitness(i)}} \quad (3)$$

*3.2.2. Kịch bản thử nghiệm*

Chúng tôi đã tiến hành thử nghiệm trên ba kịch bản với các mẫu dữ liệu huấn luyện và kiểm tra cụ thể như sau.

***Kịch bản 1**:* Trong giai đoạn huấn luyện chỉ huấn luyện trên bộ dữ liệu không có mẫu dữ liệu tấn công. Trong giai đoạn kiểm tra, kiểm tra trên bộ dữ liệu có cả các mẫu dữ liệu bình thường và dữ liệu tấn công nhằm đánh giá khả năng phát hiện tấn công của phương pháp đề xuất. Thử nghiệm được tiến hành trên ba thí nghiệm sau:

- *Thí nghiệm cho kiểu tấn công DDoS*: Không có mẫu dữ liệu huấn luyện và 500 mẫu dữ liệu bình thường; Dữ liệu kiểm tra có 500 mẫu dữ liệu tấn công và 1000 mẫu dữ liệu bình thường;

- *Thí nghiệm cho kiểu tấn công PROBE*: Không có dữ liệu huấn luyện và 190 mẫu dữ liệu bình thường; Dữ liệu kiểm tra có 180 mẫu dữ liệu tấn công và 380 mẫu dữ liệu bình thường.

- *Thí nghiệm cho kiểu tấn công DDOS và PROBE*: Không có dữ liệu huấn luyện và 360 mẫu dữ liệu bình thường; Dữ liệu kiểm tra là 180 mẫu dữ liệu tấn công PROBE với 180 mẫu dữ liệu tấn công DDOS với 320 mẫu dữ liệu bình thường

***Kịch bản 2***: Trong giai đoạn huấn luyện, huấn luyện trên bộ dữ liệu có cả các mẫu dữ liệu tấn công và bình thường. Trong giai đoạn kiểm tra, kiểm tra trên bộ dữ liệu có cả các mẫu dữ liệu tấn công và mẫu không tấn công nhằm đánh giá khả năng phát hiện tấn công của phương pháp đề xuất. Kịch bản thử nghiệm trên ba thí nghiệm với các kiểu tấn công: DDOS, PROBE và hỗn hợp DDOS-PROBE.

- *Thí nghiệm cho kiểu tấn công DDOS*: Dữ liệu đầu vào bao gồm dữ liệu huấn luyện là 50 mẫu dữ liệu tấn công và 150 mẫu dữ liệu bình thường; Dữ liệu kiểm tra là 300 mẫu dữ liệu tấn công và 600 mẫu dữ liệu bình thường;

- *Thí nghiệm cho kiểu tấn công PROBE*: Dữ liệu đầu vào bao gồm dữ liệu huấn

luyện là 40 mẫu dữ liệu tấn công và 80 mẫu dữ liệu bình thường; Dữ liệu kiểm tra là 140 mẫu dữ liệu tấn công và 300 mẫu dữ liệu bình thường;

- *Thí nghiệm cho kiểu tấn công PROBE và DDOS*: Dữ liệu huấn luyện gồm 30 mẫu dữ liệu tấn công PROBE và 30 mẫu dữ liệu tấn công DDOS và 120 mẫu dữ liệu bình thường; Dữ liệu kiểm tra gồm 150 mẫu dữ liệu tấn công PROBE với 150 mẫu dữ liệu tấn công DDOS và 320 mẫu dữ liệu bình thường;

***Kịch bản 3***: Trong giai đoạn huấn luyện trên bộ dữ liệu có chứa các mẫu tấn công smurf và bình thường. Trong giai đoạn kiểm tra, kiểm tra trên bộ dữ liệu có cả các mẫu dữ liệu bình thường và các mẫu tấn công mới nhằm đánh giá khả năng phát hiện các mẫu tấn công mới, chưa biết của phương pháp đề xuất. Dữ liệu huấn luyện bao gồm 87 mẫu dữ liệu tấn công smurf và 400 mẫu dữ liệu bình thường; Dữ liệu kiểm tra gồm 400 mẫu dữ liệu tấn công các kiểu DDOS (land, back, neptune, pop, teardrop) và 800 mẫu dữ liệu bình thường

### 3.3. Kết quả và phân tích

Kết quả thử nghiệm phương pháp đề xuất với các tham số của thuật toán được đề cập đến trong Bảng 2, nhóm tác giả đã thực hiện với 30 lần chạy và lấy kết quả phân loại tấn công của tất cả các lần thực hiện để làm giá trị thống kê và so sánh với các phương pháp khác. Hiệu suất của phương pháp áp dụng cho mỗi tập dữ liệu thử nghiệm sẽ được tính theo tỷ lệ % của các phân loại chính xác trên tập dữ liệu kiểm tra và kết quả thử nghiệm được thống kê trên các bảng.

Các kết quả thống kê khi áp dụng phương pháp được đề xuất với GP chuẩn và TAG3P cho vấn đề phát hiện tấn công được so sánh với các phương pháp học máy khác nhau (cây quyết định (J48), SVM, hai kỹ thuật mạng thần kinh nhân tạo (Multilayer Perceptron: Perc và Resting Bitch Face: RBF), và mạng Bayes (mạng Bayes: Bayes và NaiveBayes: Naïve)).

***Kịch bản 1***: Các kết quả trong Bảng 3 cho thấy các phương pháp đề xuất trong các thí nghiệm của Kịch bản 1 cho kết quả phân loại tấn công cao hơn một số phương

pháp học máy khác. Điều này cho thấy GP và TAG3P đã cải thiện đáng kể tỷ lệ phát hiện tấn công, xâm nhập.

**Bảng 3. Kết quả thí nghiệm Kịch bản 1 (%)**

| Phương pháp | J48 | SVM | Perc | Bayes | Naïve | RBF | StandGP | TAG3P |
|---|---|---|---|---|---|---|---|---|
| Thí nghiệm 1 | 66.67 | 66.67 | 66.67 | 66.67 | 66.67 | 66.67 | 70.00 | 97.06 |
| Thí nghiệm 2 | 67.86 | 67.86 | 67.86 | 67.86 | 67.86 | 67.86 | 65.00 | 99.29 |
| Thí nghiệm 3 | 47.06 | 47.06 | 47.06 | 47.06 | 47.06 | 47.06 | 95.00 | 98.72 |

***Kịch bản 2:*** Kết quả Kịch bản 2 cho thấy các phương pháp đề xuất của GP chuẩn đã cải thiện tỷ lệ phát hiện tấn công trên thí nghiệm 2 cho các mẫu tấn công thăm dò và TAG3P đã cải thiện tỷ lệ phát hiện tấn công, xâm nhập trong thí nghiệm 3 bao gồm cả các mẫu tấn công từ chối dịch vụ mà tấn công thăm dò. Tuy nhiên đối với một số mẫu khác thì tỷ lệ phát hiện lại chưa cao (Bảng 4).

**Bảng 4. Kết quả thí nghiệm Kịch bản 2 (%)**

| Phương pháp | J48 | SVM | Perc | Bayes | Naïve | RBF | StandGP | TAG3P |
|---|---|---|---|---|---|---|---|---|
| Thí nghiệm 1 | 90.36 | 98.25 | 98.62 | 93.61 | 96.62 | 98.50 | 75.00 | 93.74 |
| Thí nghiệm 2 | 96.59 | 93.41 | 95.00 | 97.50 | 92.95 | 92.95 | 100.0 | 94.76 |
| Thí nghiệm 3 | 96.58 | 94.47 | 97.11 | 98.42 | 93.95 | 93.95 | 95.00 | 99.08 |

***Kịch bản 3:*** Kết quả Kịch bản 3 cho thấy TAG3P thực sự hiệu quả trong phát hiện các mẫu tấn công mới chưa biết trước dấu hiệu tấn công. Có thể nhận thấy rằng TAG3P thực sự hiệu quả trong khả năng học và đưa ra các dự đoán đối với các trường hợp chưa biết trước các dạng tấn công và các dạng tấn công mới (Bảng 5).

**Bảng 5. Kết quả thí nghiệm Kịch bản 3 (%)**

| Phương pháp | J48 | SVM | Perc | Bayes | Naïve | RBF | StandGP | TAG3P |
|---|---|---|---|---|---|---|---|---|
| Thí nghiệm | 67.17 | 67.17 | 69.33 | 67.58 | 89.42 | 65.92 | 67.17 | 93.09 |

## 4. KẾT LUẬN VÀ HƯỚNG NGHIÊN CỨU

Bài báo trình bày nghiên cứu về vấn đề cải thiện phát hiện tấn công mạng sử dụng lập trình gen dựa trên kỹ thuật văn phạm nối cây (TAG3P) và GP chuẩn. Các thực nghiệm cho thấy việc phân loại tấn công đã cải thiện đáng kể tỷ lệ phát hiện tấn công mạng. Qua

thí nghiệm cho thấy kết quả phát hiện tấn công đối với các mẫu tấn công mới đạt được hiệu quả hơn so với một số phương pháp học máy khác. Bên cạnh đó, TAG3P cũng đã cải thiện đáng kể tốc độ huấn luyện so với GP chuẩn.

Trong thời gian tới, nhóm nghiên cứu sẽ tiếp tục cải tiến các phương pháp phát hiện tấn công dựa trên hệ lập trình gen với định hướng nâng cao tốc độ huấn luyện bởi một số kỹ thuật như tự động đáp ứng các tham số đầu vào hay giữ lại các cá thể được cho là tốt nhất ở mỗi thế hệ, sau đó sao chép trực tiếp vào thế hệ tiếp theo mà không cần áp dụng bất cứ toán tử di truyền nào trên đó.

## TÀI LIỆU THAM KHẢO

Abadeh, M. S., Habibi, J., & Lucas, C. (2007). Intrusion detection using a fuzzy genetics-based learning algorithm. *Journal of Network and Computer Applications, 30*(1), 414-428.

Abdullah, B., Abd-Alghafar, I., Gouda, I., & Salama, A. A. (2009). *Performance avaluation of a genetic algorithm based approach to network intrusion detection system*. Paper presented at The 13th International Conference on Aerospace Sciences and Aviation Technology, USA.

Ahmad, I., Hussain, M., Alghamdi, A., & Alelaiwi, A. (2013). Enhancing SVM performance in intrusion detection using optimal feature subset selection based on genetic principal components. *Springer Open, 24*(7-8), 1671-1682.

Al-Jarrah, O. Y., Siddiqui, A., Elsalamouny, M., Yoo, P. D., Muhaidat, S., & Kim, K. (2014). *Machine learning based feature selection techniques for large-scale network intrusion detection*. Paper presented at The IEEE 34th International Conference on Distributed Computing Systems Workshops, USA.

Anup, G., & Chetan, K. (2008). *GA-NIDS: A Genetic algorithm based network intrusion detection system*. Retrieved from https://www.researchgate.net/publication/228791237_GA-NIDS_A_Genetic_Algorithm_based_Network_Intrusion_Detection_System

Botha, M., & Solms, R. (2004). *Utilizing neural networks for effective intrusion detection*. Retrieved from http://icsa.cs.up.ac.za/issa/2004/Proceedings/Full/040.pdf

Bridges, S. M., & Vaughn, R. B. (2000). *Fuzzy data mining and genetic algorithms applied to intrusion detection*. Paper presented at The Twenty-third National Information Systems Security Conference, *USA.*

Crosbie, M., & Spafford, E. (1995). Applying genetic programming to intrusion detection. *International Journal of Science and Research*, *2*(6), 480-483.

Devarakonda, N., & Pamidi, S. (2012). Intrusion detection system using Bayesian network and Hidden Markov model. *Procedia Technology, 4*(1), 506-514.

Faraoun, K. M., Boukelif, A., & Algeria, S. B. A. (2006). Genetic programming approach for multi-category pattern classification applied to network intrusions detection. *International Journal of Computational Intelligence and Applications, 6*(1), 3098-3109.

Gomez, J., & Dasgupta, D. (2002). *Evolving fuzzy rules for intrusion detection*. Paper presented at The Third Annual IEEE Information Assurance Workshop 2002 Conference, USA.

Gong, R. H., Zulkernine, M., & Abolmaesumi, P. (2005). *A software implementation of a genetic algorithm based approach to network intrusion detection*. Paper presented at The Sixth International Conference on Software Engineering, USA.

Koza, J. R. (1992). *Genetic programming: On the programming of computers by means of natural selection*. Massachusetts, USA: MIT Press.

Le, H. N., Hoang, T. H., & Vu, V. C. (2015). Self-adaptive srossover and mutation parameters in tree adjoining grammar guided genetic programming. *Tạp chí Khoa học và Kỹ thuật Học viện Kỹ thuật Quân sự, 15*(6), 5-15.

Leung, Y., So, L., & Yam, K. F. (1992). *Rule learning in expert systems using genetic algorithm*. Paper presented at The International Conference on Fuzzy Logic & Neural Networksm, Japan.

Li, W. (2004). *Using genetic algorithm for network intrusion detection*. Retrieved from https://pdfs.semanticscholar.org/9175/54c7cce69e6ee9708020863f2bd27fa986a6.pdf.

Lu, W., & Traore, I. (2004). Detecting new forms of network intrusion using genetic programming. *Computational Intelligence, 20*(3), 475-494.

Middlemiss, M., & Dick, G. (2003). *Feature selection of intrusion detection data using a hybrid genetic algorithm/KNN approach.* Amsterdam, Netherlands: IOS Press.

Mukkamala, S., Andrew, H. S., & Ajith, A. (2005). Intrusion detection using an ensemble of intelligent paradigms. *Journal of Network and Computer Applications, 28*(2), 167-182.

Nguyen, X. H., McKay, R. I., & Abbass, H. A. (2003). *Tree adjoining grammars, language bias, and genetic programming.* Paper presented at The EuroGP2003, Netherlands.

Peddabachigari, S., Ajith, A. G., & Thomas, J. (2007). Modeling intrusion detection system using hybrid intelligent systems. *Journal of Network and Computer Applications, 30*(1), 114-132.

Peng, T., Leckie, C., & Kotagiri, R. (2007). Information sharing for distributed intrusion detection systems. *Journal of Network and Computer Applications, 30*(3), 877-899.

Pham, T. S., Nguyen, Q. U., & Nguyen, X. H. (2014). *Generating artificial attack data for intrusion detection using machine learning*. Paper presented at The Fifth

Symposium on Information and Communication Technology Conference, Vietnam.

Pillai, M. M., Eloff, J. H. P., & Venter, H. S. (2004). *An approach to implement a network intrusion detection system using genetic algorithms.* Paper presented at The SAICSIT, South Africa.

UCI KDD Archive. (1999). *KDD cup 1999 data*. Retrieved from http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

Wong, M. L., Leung, K. S., & Cheng, J. C. Y. (2000). Discovering knowledge from noisy databases using genetic programming. *Journal of the American Society for Information Science and Technology, 51*(9), 870-881.

# IMPROVING INTRUSION DETECTION USING TREE ADJOINING GRAMMAR GUIDED GENETIC PROGRAMMING

**Vu Van Canh[a,b*], Hoang Tuan Hao[a], Nguyen Van Hoan[b]**

*[a]The Faculty of Information Technology, Lequydon Technical University, Hanoi, Vietnam*
*[b]The Faculty of Information Technology, Telecommunication University, Khanhhoa, Vietnam*
*[*]Corresponding author: Email: canhvuvan@yahoo.com*

**Article history**
Received: January 07th, 2017 | Received in revised form: July 13th, 2017
Accepted: July 20th, 2017

**Abstract**

*Nowadays, the problem of network security has become urgent and affect the performance of modern computer networks greatly. Detection and prevention of network attacks have been the main topic of many researchers in the World. One of the safety measures for networks is using the intrusion detection systems. However, these measures are costly, ineffective, unreliable and can-not detect new or unknown attacks. Some studies using machine learning technology have been applied in intrusion detection. In our work, we proposed using Genetic Programming (GP) to improve intrusion detection. In the experiments, we used GP and Tree Adjoining Grammar Guided Genetic Programming (TAG3P) on artifical datasets suggested by Pham, Nguyen, and Nguyen (2014). Compared with previous results, we found that GP and TAG3P are more effective in detecting attacks than previous measures.*

**Keywords**: Attack detection; Classification; Genetic Programming (GP); IDS; TAG3P.