ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG



Báo cáo cuối kì:

Hệ thống tìm kiếm, phát hiện và ngăn ngừa xâm nhập

Đề tài: Phương pháp lựa chọn đặc trưng dựa trên Thuật toán di truyền

GVHD: Đỗ Hoàng Hiển

**Nhóm 3:**

**Huỳnh Minh Tân Tiến 21521520**

**Nguyễn Ngọc Trà My 21520353**

**Lê Hoàng Oanh 21521253**

**Bùi Hoàng Trúc Anh 21521817**

mục lục

[Chương 1: Giới thiệu 4](#_Toc170416634)

[I. Giới thiệu vấn đề 4](#_Toc170416635)

[1. Hệ thống phát hiện xâm nhập 4](#_Toc170416636)

[2. Hệ thống phát hiện xâm nhập dựa trên học máy 4](#_Toc170416637)

[3. Feature Selection 4](#_Toc170416638)

[4. Vấn đề bài báo 5](#_Toc170416639)

[II. Mục tiêu của đồ án 5](#_Toc170416640)

[III. Phương pháp thực hiện 5](#_Toc170416641)

[Chương 2: Cơ sở lý thuyết và các nghiên cứu liên quan 6](#_Toc170416642)

[I. Các phương pháp, công cụ liên quan 6](#_Toc170416643)

[1. Chuẩn hóa (normalization) và tỉ lệ hóa (scaling) 6](#_Toc170416644)

[2. Feature Selection 6](#_Toc170416645)

[3. Genetic Algorithm 7](#_Toc170416646)

[4. Công thức của hàm thích nghi (Fitness Function) được sử dụng 12](#_Toc170416647)

[5. Các bộ phân loại được sử dụng 12](#_Toc170416648)

[6. Các tập dataset được sử dụng 13](#_Toc170416649)

[II. So sánh với những công trình nghiên cứu khác 14](#_Toc170416650)

[1. Công trình của (Sindhu, 2012) 14](#_Toc170416651)

[2. Công trình của (Aslahi-Shahri, 2016) 15](#_Toc170416652)

[3. Công trình của (Gharaee, 2016) 15](#_Toc170416653)

[4. Bảng so sánh 15](#_Toc170416654)

[Chương 3: Phân tích thiết kế hệ thống 16](#_Toc170416655)

[I. Chi tiết phương pháp thực hiện 16](#_Toc170416656)

[1. Phương pháp thực hiện 17](#_Toc170416657)

[2. Kiến trúc tổng quan 17](#_Toc170416658)

[3. Chuẩn bị dữ liệu (Datasets Preparation) 17](#_Toc170416659)

[4. Mô-đun học máy dựa trên GA (GA-based Machine Learning Module) 18](#_Toc170416660)

[5. Mô-đun phân loại (Classification Module) 22](#_Toc170416661)

[Chương 4: Hiện thực hệ thống 22](#_Toc170416662)

[I. Các bước cài đặt cấu hình hệ thống 22](#_Toc170416663)

[II. Hiện thực hệ thống 22](#_Toc170416664)

[1. Chuẩn bị dữ liệu 22](#_Toc170416665)

[2. Xử lý dữ liệu 23](#_Toc170416666)

[3. Genetic Algorithm for Feature Selection (GAbFS) 23](#_Toc170416667)

[4. Phân loại sử dụng các bộ phân loại khác nhau 28](#_Toc170416668)

[Chương 5: Thực nghiệm và đánh giá 32](#_Toc170416669)

[I. Kịch bản thực nghiệm 32](#_Toc170416670)

[II. Đánh giá kết quả 32](#_Toc170416671)

[1. Tiêu chí đánh giá 32](#_Toc170416672)

[2. Kết quả 33](#_Toc170416673)

[Chương 6: Kết luận và hướng phát triển 34](#_Toc170416674)

[I. Kết luận 34](#_Toc170416675)

[1. Tóm tắt vấn đề 34](#_Toc170416676)

[2. Phương pháp 34](#_Toc170416677)

[3. Kết quả 34](#_Toc170416678)

[II. Hướng phát triển 34](#_Toc170416679)

[Q&A 35](#_Toc170416680)

[Reference 36](#_Toc170416681)

# **Chương 1: Giới thiệu**

1. **Giới thiệu vấn đề**

Trong thời đại số với nhiều thiết bị kết nối, việc đảm bảo an ninh và bảo mật dữ liệu là thách thức lớn. Các cuộc tấn công mạng ngày càng phức tạp và đa dạng, đòi hỏi sự bảo vệ vượt ra ngoài giới hạn của bảo mật thông tin truyền thống. Tình báo đe dọa mạng (CTI) giúp thu thập và phân tích dữ liệu để nhận dạng và đối phó với các mối đe dọa an ninh mạng. Thách thức lớn là cung cấp thông tin này để hệ thống có thể phân tích và phản ứng kịp thời. Cuộc tấn công mạng thường có vòng đời từ trinh sát nạn nhân đến khai thác lỗ hổng. Phương pháp phòng thủ an ninh mạng bao gồm ba khía cạnh chính: phòng ngừa, phát hiện, và phản ứng.

1. **Hệ thống phát hiện xâm nhập**

Hệ thống phát hiện xâm nhập (IDS) là thành phần chính của một hệ thống an ninh nhằm tăng cường an ninh hệ thống. Nhiệm vụ của nó tương tự như những gì phần mềm diệt virus hoặc tường lửa phải làm. Mục đích chính của IDS là xác định hoạt động độc hại xảy ra trong mạng máy tính và cung cấp cơ chế để đối phó với các cuộc tấn công mạng. Hiện tại, vấn đề chính với IDS là chúng tạo ra nhiều cảnh báo dương tính giả (FP) và âm tính giả (FN). Do đó, cần triển khai các kỹ thuật trên dữ liệu thu thập được để giúp IDS tăng độ chính xác.

1. **Hệ thống phát hiện xâm nhập dựa trên học máy**

Trong bối cảnh các mối đe dọa mạng ngày càng tiến xa và tinh vi, việc sử dụng các phương pháp học máy và khai thác dữ liệu đã trở thành một xu hướng quan trọng trong việc phát hiện tấn công mạng. Các phương pháp học máy có thể phân loại các mẫu dữ liệu mạng vào các loại tấn công khác nhau, từ việc sử dụng các tệp tài liệu đa dạng đến việc lan truyền các phần mềm độc hại thông qua các kỹ thuật tấn công sâu. Tuy nhiên, để triển khai và sử dụng hiệu quả các phương pháp học máy, cần có một lượng lớn dữ liệu chất lượng để huấn luyện mô hình và đảm bảo tính chính xác của quá trình phân loại.

1. **Feature Selection**

Trong quá trình xử lý dữ liệu lớn và áp dụng học máy, việc lựa chọn các đặc trưng quan trọng và phù hợp là một phần quan trọng để cải thiện hiệu suất của mô hình. Feature Selection giúp giảm dimensionality của dữ liệu, loại bỏ nhiễu và các đặc trưng không cần thiết, từ đó cải thiện khả năng dự đoán và giảm thiểu hiện tượng overfitting.

1. **Vấn đề bài báo**

Nghiên cứu này tập trung vào giải quyết vấn đề lựa chọn một tập hợp con của các đặc trưng gốc để cải thiện độ chính xác phát hiện của một hệ thống phát hiện xâm nhập (IDS). Khi có nhãn class, việc lựa chọn đặc trưng trở nên dễ dàng hơn bằng cách đánh giá tác động của từng đặc trưng đối với việc dự đoán nhãn class. Tuy nhiên, ngay cả khi không có nhãn class (supervised-learning), việc lựa chọn đặc trưng vẫn có thể được thực hiện thông qua unsupervised-learning. Phương pháp này đảm bảo rằng tập hợp con của các đặc trưng từ các thuộc tính ban đầu là tối ưu và có thể đạt được độ chính xác phát hiện cao hơn.

Việc lựa chọn các đặc trưng đại diện tối ưu bằng học máy là một bước quan trọng. Giảm dimensionality được thực hiện để giảm số lượng đặc trưng bằng cách loại bỏ các đặc trưng không liên quan và trùng lặp. Bài báo này sử dụng Thuật toán Di truyền (Genetic Algorithm) để tìm kiếm các đặc trưng tối ưu và điều chỉnh thông số để cải thiện hiệu suất phân loại. Một hàm mới được thiết lập để tối ưu hóa quá trình này. Dựa trên các thông số đã được điều chỉnh, thuật toán Di truyền hội tụ nhanh chóng và cung cấp các đặc trưng tối ưu có độ chính xác phát hiện cao hơn.

1. **Mục tiêu của đồ án**

Mục tiêu của đồ án này là xây dựng được giải pháp được để xuất trong bài báo dựa trên thuật toán di truyền để tìm ra tập hợp con của các đặc trưng tối ưu cho hệ thống phát hiện xâm nhập, từ đó cải thiện hiệu suất và độ chính xác của hệ thống. Thử nghiệm và áp dụng các tập dataset lên module giải pháp GAbFS.

1. **Phương pháp thực hiện**

* Xử lý dữ liệu của ba tập dataset: CIRA-CIC-DOHBrw-2020, UNSW-NB15 và Bot-IoT.
* Huấn luyện các bộ phân loại học máy bằng cách sử dụng các đặc trưng tối ưu được chọn thông qua mô-đun GA đã phát triển.
* Đánh giá hiệu suất trên các tập dữ liệu thử nghiệm chuẩn, cụ thể là CIRA-CIC-DOHBrw-2020, UNSW-NB15 và Bot-IoT
* Kiểm tra kỹ thuật đề xuất với ba bộ phân loại, cụ thể là k-Nearest Neighbor (k-NN), Support Vector Machine (SVM), và XgBoost.

# **Chương 2: Cơ sở lý thuyết và các nghiên cứu liên quan**

1. **Các phương pháp, công cụ liên quan**
2. **Chuẩn hóa (normalization) và tỉ lệ hóa (scaling)**

Trong thực tế, một số dữ liệu có thể không ở dạng phù hợp để mô-đun học tập hoạt động, vì giải pháp hiện tại chỉ chấp nhận dữ liệu ở dạng số và tạo ra đầu ra trong một phạm vi cụ thể tùy thuộc vào hàm thích hợp. Các giá trị đầu vào của các dữ liệu khác nhau có thể không cùng tỷ lệ. Để xử lý điều này, việc tỉ lệ hoá là cần thiết. Phương pháp này tỉ lệ hoá tất cả các thuộc tính bằng Công thức (1):

(1)

Trong đó, lần lượt là các giá trị ban đầu, giá trị đã tỉ lệ hoá, giá trị tối đa và giá trị tối thiểu. Các thuộc tính được tỉ lệ hoá đầu tiên bằng Công thức (1) và sau đó dữ liệu được chuẩn hóa. Việc chuẩn hóa được thực hiện để hạn chế tất cả các giá trị đầu vào trong một phạm vi mong muốn để xử lý bởi mô-đun học tập. Công thức được sử dụng cho việc chuẩn hóa được liệt kê trong Công thức (2):

(2)

Để chuyển đổi các giá trị chữ số và chữ cái thành các giá trị số, 1-of-n encoding scheme được sử dụng. Các đặc điểm phân loại trong các tập dữ liệu được mã hóa thành một định dạng có thể thực thi cho mục đích học máy bằng cách sử dụng label encoder.

1. **Feature Selection**
   1. **Định nghĩa**

Feature Selection là quá trình chọn ra một tập hợp con của các đặc trưng từ tập dữ liệu gốc để sử dụng cho việc huấn luyện mô hình máy học. Mục tiêu của Feature Selection là giảm số lượng các đặc trưng để cải thiện hiệu suất của mô hình và giảm thiểu thời gian và chi phí tính toán.

Feature Selection và Dimensionality Reduction thường được nhắc đến cùng nhau, nhưng chúng có điểm khác biệt như sau:

* Feature Selection: Giảm số lượng đặc trưng bằng cách loại bỏ các đặc trưng không quan trọng, không ảnh hưởng đến mục tiêu dự đoán. Các đặc trưng còn lại vẫn là các đặc trưng ban đầu trong tập dữ liệu.
* Dimensionality Reduction: Giảm số lượng đặc trưng bằng cách ánh xạ dữ liệu từ không gian ban đầu sang không gian có số chiều nhỏ hơn. Các đặc trưng mới được tạo ra và không nhất thiết phải giữ nguyên các đặc trưng ban đầu.
  1. **Các phương pháp thực hiện Feature Selection**

Có hai phương pháp chính để thực hiện Feature Selection:

1. Supervised: Sử dụng nhãn (target variable) để chọn các đặc trưng quan trọng.

* Intrinsic: Thực hiện Feature Selection trong quá trình huấn luyện mô hình.
* Filter: Chọn ra các đặc trưng dựa trên mối quan hệ với nhãn, thường sử dụng các phương pháp thống kê.
* Wrapper: Tạo ra các tập con của các đặc trưng và đánh giá chúng bằng cách huấn luyện mô hình.

1. Unsupervised: Không sử dụng nhãn trong quá trình chọn đặc trưng.
   1. **Các phương pháp thống kê sử dụng cho Feature Selection**

Các phương pháp thống kê thường được sử dụng trong Feature Selection bao gồm các phương pháp đo tương quan giữa các biến đầu vào và biến đầu ra, dựa vào loại dữ liệu:

* Đối với dữ liệu số (numeric): Sử dụng các phép đo tương quan như hệ số tương quan Pearson, hệ số tương quan Spearman.
* Đối với dữ liệu phân loại (categorical): Sử dụng các phép đo như chi-squared test, mutual information.

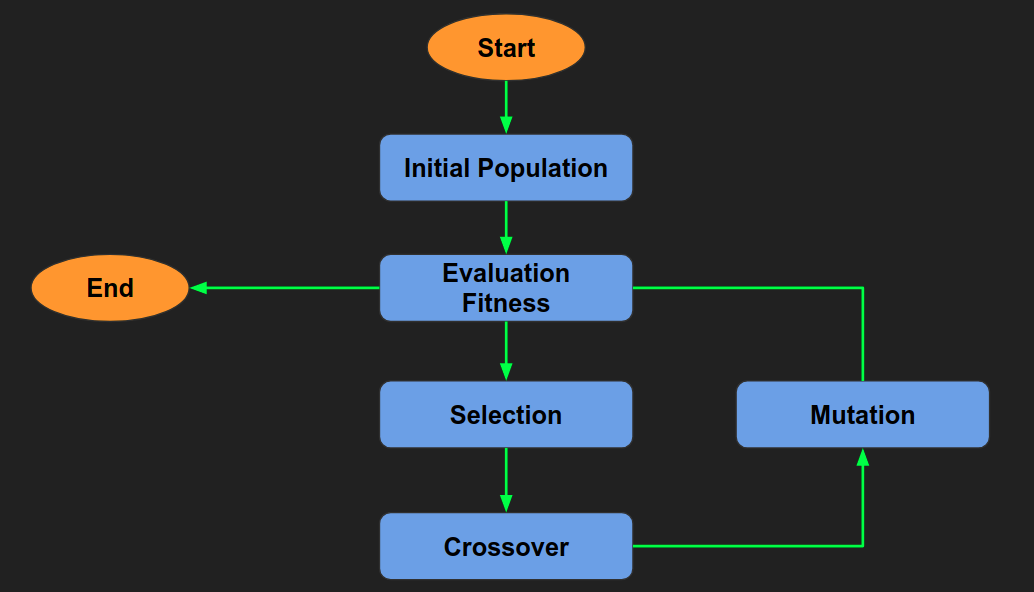
Việc hiểu và áp dụng các phương pháp Feature Selection là quan trọng để tối ưu hóa hiệu suất của mô hình máy học và giảm thiểu thời gian và chi phí tính toán.

1. **Genetic Algorithm**
   1. **Giới thiệu về Genetic Algorithm (GA)**

Genetic Algorithm (GA) là một phương pháp tối ưu hóa dựa trên cơ chế chọn lọc tự nhiên của Darwin. Nó được sử dụng rộng rãi trong các lĩnh vực như trí tuệ nhân tạo, học máy, và các bài toán tối ưu hóa phức tạp. GA bắt chước quá trình tiến hóa tự nhiên thông qua các nguyên tắc di truyền học như chọn lọc tự nhiên, lai ghép, và đột biến.

* 1. **Nguyên lý hoạt động của Genetic Algorithm**

GA bắt đầu với một tập hợp các giải pháp khả thi gọi là quần thể (population). Mỗi giải pháp trong quần thể được gọi là một cá thể (individual) hoặc nhiễm sắc thể (chromosome). Các bước chính của GA bao gồm:



*Hình 1. Nguyên lý hoạt động của GA*

* Khởi tạo quần thể (Initial population): Một quần thể ban đầu được tạo ra một cách ngẫu nhiên hoặc bằng các phương pháp heuristics.
* Đánh giá năng lực (Evaluation Fitness): Mỗi cá thể trong quần thể được đánh giá bằng một hàm thích nghi (fitness function) để đo lường mức độ tối ưu của nó.
* Chọn lọc (Selection): Các cá thể tốt hơn (có giá trị fitness cao hơn) được chọn lọc để sinh sản. Các phương pháp chọn lọc phổ biến bao gồm: roulette wheel selection, tournament selection, và rank selection.
* Lai ghép (Crossover): Các cặp cá thể được chọn để lai ghép, tạo ra con cái mới. Quá trình này mô phỏng sự lai ghép di truyền trong tự nhiên. Một số phương pháp lai ghép thông dụng là: single-point crossover, two-point crossover, và uniform crossover.
* Đột biến (Mutation): Một số cá thể trong quần thể bị thay đổi ngẫu nhiên để duy trì sự đa dạng di truyền và tránh mắc kẹt trong cực trị cục bộ. Các phương pháp đột biến phổ biến bao gồm: bit-flip mutation và swap mutation.

Các bước trên được lặp lại cho đến khi đạt được một điều kiện dừng nhất định, chẳng hạn như đạt đến số thế hệ tối đa hoặc giá trị fitness mong muốn.

* 1. **Ví dụ về Genetic Algorithm**

Xét bài toán Tìm mật khẩu, yêu cầu của bài toán như sau:

* Mật khẩu gồm 8 kí tự (bao gồm chữ cái, chữ số và khoảng trắng) - Ví dụ: hoilamgi.
* Mỗi lần thử, hệ thống sẽ báo về số lượng kí tự đúng với mật khẩu.
* Yêu cầu tìm ra chuỗi mật khẩu cho trước.

Thuật toán sẽ dừng lại khi tìm được cá thể đáp ứng được nhu cầu đề ra sau mỗi thế hệ mới. Quá trình sản sinh thế hệ tiếp theo sẽ là một vòng lặp (Evaluation Fitness -> Selection -> Crossover -> Mutation). Chúng ta sẽ cùng xây dựng những thành phần chính trong thuật toán để giải quyết bài toán này.

* Initial Population- Khởi tạo quần thể

Nhìn chung thuật toán sẽ mô phỏng lại hầu hết những hiện tượng xảy ra trong quá trình tiến hóa của động vật. Vì vậy để thuật toán có thể vận hành được, thì điều đầu tiên cần có chính là quần thể. Xét bài toán tìm mật khẩu, quần thể sẽ bao gồm những chuỗi 8 kí tự, được sinh ra ngẫu nhiên.

A screenshot of a computer screen

Description automatically generated

* Evaluation Fitness - Đánh giá năng lực

Tiếp theo, mỗi chuỗi mật khẩu sẽ được đánh giá sự chính xác so với mật khẩu cho trước, với mỗi kí tự giống với mật khẩu cho trước tại đúng vị trí sẽ được 1 point.

A screenshot of a computer

Description automatically generated

Thành phần Point ở đây sẽ đại diện cho khả năng sinh tồn của cá thể trong quần thể, càng lớn tức cá thể đó càng thích nghi với môi trường tốt.

* Selection - Chọn lọc

Sau khi đã đánh giá được quần thể, các cá thể có khả năng sinh tồn tốt hơn sẽ có cơ hội được sinh sản nhiều hơn các cá thể còn lại. Các chuỗi kí tự mật khẩu sẽ được lựa chọn theo số Point đang có.

A screenshot of a game

Description automatically generated

* Crossover - Sinh sản

Vào giai đoạn sinh sản, các cá thể con sẽ được kế thừa các đặc tính từ cả bố và mẹ. Thông thường, cá thể con sẽ nhận một nửa gen từ mỗi bố, mẹ.

A screenshot of a computer game

Description automatically generated

Cá thể con có thể sẽ thích nghi tốt hơn, hoặc kém hơn. Ngoài ra, có những kiểu lai tạo khác nhau như 2 Point, Uniform Selection.

A screenshot of a computer game

Description automatically generated

A diagram of a number

Description automatically generated with medium confidence

* Mutation - Đột biến

Có thể thấy rằng, nếu chỉ bằng việc sinh ngẫu nhiên và lai tạo, sẽ rất khó để tìm được nghiệm. Như đã nói từ đầu, đột biến chính là nguyên liệu của chọn lọc tự nhiên, bằng việc lựa chọn ngẫu nhiên các vị trí và thay thế bằng một kí tự ngẫu nhiên nào đó, chúng ta có thể mô phỏng lại hiện tượng đột biến - Đột biến điểm

A screenshot of a cell phone

Description automatically generated

Các cá thể đột biến có thể sẽ có khả năng thích nghi tốt hơn (1 -> 2), hoặc cũng có thể ngược lại (4 -> 3). Quá trình này sẽ lặp lại cho đến khi tìm được đáp án phù hợp.

1. **Công thức của hàm thích nghi (Fitness Function) được sử dụng**

* Mục tiêu chính của kỹ thuật được đề xuất là tối đa hóa cả giá trị trung bình của độ tương quan và độ chính xác cùng một lúc. Công thức (5) dưới đây cho thấy hàm fitness được đề xuất.

(3)

Giá trị Correlation Trung Bình (): Đây là giá trị trung bình của các hệ số tương quan được tính từ các giá trị nằm trên đường chéo của ma trận tương quan. Nó đo lường mức độ tương quan giữa các đặc trưng được chọn.

(4)

Giá trị Un-correlation Trung Bình Đã Chuyển Đổi (): Đây là giá trị trung bình của các hệ số không tương quan, được chuyển đổi từ giá trị tương quan trung bình. Nó được tính bằng cách lấy 1 trừ đi giá trị tương quan trung bình , nhằm mục đích đảo ngược giá trị để phản ánh mức độ không tương quan giữa các đặc trưng.

Trong đó:

Fi: Giá trị fitness của chromosome thứ i.

A1: Độ chính xác đạt được từ chromosome thứ i.

Mi: Ma trận độ tương quan đã tính toán.

* Hàm fitness này sử dụng hai quy trình: hàm mục tiêu và hàm tỷ lệ. Độ chính xác đạt được là hàm mục tiêu mà công trình này tối ưu hóa. Điều này được thực hiện thông qua hàm tỷ lệ, đó là hàm không tương quan, nó tỷ lệ hiệu suất của độ chính xác.

1. **Các bộ phân loại được sử dụng**
   1. **Support Vector Machines (SVMs)**

Support Vector Machines (SVMs) là một mô hình học có giám sát thường được sử dụng cho phân loại. Dù được ưa chuộng trong việc phân chia dữ liệu tuyến tính, SVM cũng có khả năng xử lý phân loại phi tuyến trong không gian đặc trưng có số chiều cao. Mô hình này tạo ra các siêu mặt phẳng (hay mặt quyết định) trong không gian đó để phân loại dữ liệu. Một số ưu điểm của SVM bao gồm hiệu quả trong không gian đa chiều, linh hoạt với khả năng sử dụng nhiều loại hàm kernel khác nhau, và hiệu quả về bộ nhớ. Tuy nhiên, SVM có thể gặp vấn đề nếu số lượng đặc điểm lớn hơn so với số lượng mẫu, dẫn đến hiện tượng overfitting.

* 1. **K-Nearest Neighbor (k-NN)**

K-Nearest Neighbor (k-NN) là một phương pháp được sử dụng cho cả phân loại và hồi quy. Phương pháp này phân loại dữ liệu dựa trên phiếu bầu đa số của k-lân cận. Trong trường hợp đơn giản khi k = 1, chỉ có một lớp dữ liệu được xác định. Giá trị tối ưu của k có thể được xác định thông qua việc kiểm tra dữ liệu hoặc thực hiện các thí nghiệm. Một giá trị k lớn hơn có thể tốt hơn vì nó giảm nhiễu và sự bất thường. Phiếu bầu được quyết định dựa trên khoảng cách giữa hai điểm, có thể sử dụng các phương pháp đo khoảng cách như Euclidean, Manhattan, Minkowski, hoặc các hàm khoảng cách khác tùy thuộc vào loại dữ liệu được sử dụng.

* 1. **XgBoost (eXtreme Gradient Boosting)**

XGBoost, hay eXtreme Gradient Boosting, là một thuật toán tinh vi được thiết kế để xử lý các không đều hiện diện trong tập dữ liệu. Chức năng chính của XGBoost bao gồm cung cấp chức năng regularization, xử lý song song, tính linh hoạt cao trong việc đánh giá tiêu chuẩn theo ý của người dùng, hỗ trợ các thuật toán tối ưu hóa, và quản lý giá trị thiếu.Ưu điểm của XGBoost bao gồm việc sử dụng cơ sở hỗ trợ cây tăng cường song song và giải quyết nhanh chóng nhiều vấn đề học máy với độ chính xác cao hơn.

1. **Các tập dataset được sử dụng**
   1. **CIRA-CIC-DOHBrw-2020**

Bộ dữ liệu CIRA-CIC-DoHBrw-2020 ghi lại lưu lượng mạng có chứa giao thức DNS over HTTPS (DoH), bao gồm cả lưu lượng benign (không có hại) và malicious (có hại), cũng như lưu lượng không phải DoH. Dữ liệu được tạo ra bằng cách truy cập vào 10.000 trang web hàng đầu theo thứ hạng của Alexa và sử dụng trình duyệt web, cùng với việc sử dụng các ứng dụng đào đường DNS hỗ trợ giao thức DoH. Bộ dữ liệu bao gồm tổng cộng 34 đặc điểm và được phân thành bốn lớp. Số lượng mẫu trong bộ dữ liệu này là khoảng 1,4 triệu. Trong quá trình thu thập dữ liệu, các gói tin nhỏ không mang dữ liệu đã bị loại bỏ để giảm kích thước của dữ liệu.

* 1. **UNSW-NB15**

Bộ dữ liệu UNSW NB-15 chứa các gói tin mạng gốc được tạo ra thông qua công cụ IXIA Perfect-Storm tại Trung tâm An ninh Mạng của Úc (ACCS). Bộ dữ liệu dựa trên các hoạt động bình thường thực tế và các hành vi tấn công tổng hợp. Có chín gia đình tấn công được bao gồm trong bộ dữ liệu UNSW NB-15, bao gồm fuzzers, analysis, backdoors, DoS, exploits, generic, reconnaissance, shellcode và worms (Moustafa và Slay, 2015). Bộ dữ liệu bao gồm tổng cộng 49 đặc điểm đại diện cho dữ liệu của các lớp khác nhau.

* 1. **Bot-IoT**

Bộ dữ liệu Bot-IoT được tạo ra tại trung tâm Cyber UNSW Canberra, thông qua một môi trường mạng thực tế tích hợp cả lưu lượng bình thường cũng như lưu lượng botnet. Bộ dữ liệu có khoảng 72.000.000 bản ghi bao gồm các lớp tấn công DDoS, DoS, OS, và quét dịch vụ, keylogging, và lớp tấn công rò rỉ dữ liệu.

1. **So sánh với những công trình nghiên cứu khác**

Do sự gia tăng theo cấp số nhân về lưu lượng mạng, nhu cầu về một phương pháp lựa chọn đặc trưng hiệu quả đã tăng lên. Điều này có tác dụng phát triển một Hệ thống Phát hiện Xâm nhập (IDS) với mức độ chính xác phát hiện tối ưu và tỷ lệ cảnh báo sai thấp. Các thuật toán Evotionary Algorithm (EA) là các phương pháp lấy cảm hứng từ sinh học dựa trên nguyên tắc của Darwin (Wang, 2020) . Phần này trình bày về việc đặt vấn đề và tổng quan tài liệu về các phương pháp lựa chọn đặc trưng sử dụng EA đặc biệt cho IDS.

1. **Công trình của (Sindhu, 2012)**

Decision tree based light weight intrusion detection using a wrapper approach: Nghiên cứu này sử dụng GA làm kỹ thuật lựa chọn đặc trưng. Họ khởi tạo quá trình tối ưu hóa bằng cách tạo ra dân số ngẫu nhiên và thông qua tính toán độ thích nghi của nhiễm sắc thể, họ tạo ra dân số cho thế hệ tiếp theo. Fitness function xem xét số lượng đặc trưng, độ nhạy và độ đặc hiệu để đánh giá tập hợp đặc trưng. Họ thực hiện so sánh với các kỹ thuật lựa chọn đặc trưng khác và đạt được độ chính xác 98,38% với 16 đặc trưng được trích xuất từ 41 đặc trưng ban đầu của tập dữ liệu KDD.

1. **Công trình của (Aslahi-Shahri, 2016)**

A hybrid method consisting of GA and SVM for intrusion detection system: Nghiên cứu này đánh giá hiệu suất của bộ phân loại SVM trên tập dữ liệu phát hiện xâm nhập. Theo các tác giả, SVM không đạt được độ chính xác cao khi hoạt động riêng lẻ. Điều này là do SVM cần một tập dữ liệu có mẫu hợp lý và chọn các đặc trưng tối ưu với độ dư thừa tối thiểu. Vì vậy, các tác giả triển khai GA để tìm kiếm các đặc trưng tối ưu cho SVM và sau đó thực hiện phân loại trên tập dữ liệu phát hiện xâm nhập.

1. **Công trình của (Gharaee, 2016)**

A new feature selection IDS based on genetic algorithm and SVM: Nghiên cứu này đề xuất một hệ thống IDS sử dụng GA để lựa chọn đặc trưng với một hàm fitness function đổi mới. Họ đạt được độ chính xác dự đoán cao trong khi duy trì tỷ lệ báo động giả thấp. Đề xuất của họ được kiểm tra trên các tập dữ liệu KDD cup và UNSW-NB 15. Họ báo cáo độ chính xác cho từng lớp trong bài báo của mình. Họ cũng tạo ra một tập dữ liệu riêng biệt cho từng lớp và sau đó áp dụng kỹ thuật của mình.

1. **Bảng so sánh**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nghiên cứu | Phương pháp | Dataset | Số lượng feature | Classifier | Kết quả |
| Sindhu (2012) | Sử dụng GA với hàm fitness function đánh giá số lượng đặc trưng | KDD CUP’99 | 16 | Decision Tree | Đạt được độ chính xác 98,38% với 16 đặc trưng |
| Aslahi-Shahri (2016) | Kết hợp GA và SVM để tối ưu hóa đặc trưng cho SVM | KDD CUP’99 | 10 | SVM | Đánh giá hiệu suất của SVM và đạt được độ chính xác cao. True Positive rate:  0.97 |
| Gharaee (2016) | Sử dụng GA để lựa chọn đặc trưng với hàm fitness function mới | KDD CUP’99,  UNSW-NB 15 | - | SVM | Đạt được độ chính xác dự đoán cao và duy trì tỷ lệ báo động giả thấp |
| Phương pháp được đề xuất | Sử dụng GA để lựa chọn đặc trưng và điều chỉnh tham số | KDD Cup’99,  UNSW-NB15, and  Bot-IoT | 10, variable | SVM, k -NN, and  XgBoost | Tối ưu hóa cả lựa chọn đặc trưng và điều chỉnh tham số. Accuracy: 99.80% |

# **Chương 3: Phân tích thiết kế hệ thống**

A diagram of a network diagram

Description automatically generated

*Hình 2. Kiến trúc tổng quan*

1. **Chi tiết phương pháp thực hiện**
2. **Phương pháp thực hiện**

Trong lĩnh vực an ninh mạng và phát hiện xâm nhập, chúng ta đối mặt với vấn đề về chiều cao của dữ liệu (high dimensionality), gây ảnh hưởng tiêu cực đến hiệu quả của các thuật toán. Nhiệm vụ là tạo ra một chiến lược giữ lại thông tin đặc biệt nhất trong khi giảm số lượng đặc trưng xuống mức tối thiểu có thể thực hiện. Đối với vấn đề lựa chọn thuộc tính trong lĩnh vực bảo mật mạng và phát hiện xâm nhập, bài báo này đóng góp một phương pháp lựa chọn tính năng dựa trên Thuật toán di truyền (Genetic Algorithm - GA) nâng cao, được đặt tên là Lựa chọn tính năng dựa trên GA (GA-based Feature Selection - GbFS), để tăng độ chính xác của bộ phân loại.

1. **Kiến trúc tổng quan**

Kiến trúc của giải pháp đề xuất được thể hiện ở mức độ khái quát cao hơn trong Hình 2. Sơ đồ khối cho thấy đầu vào là dữ liệu lưu lượng mạng và đầu ra cuối cùng là tỷ lệ phát hiện được cải thiện của lưu lượng phát hiện tấn công và lưu lượng bình thường. Toàn bộ quá trình bao gồm ba giai đoạn chính:

* Chuẩn bị dữ liệu (Data Preparation): Bao gồm việc thu thập và tiền xử lý dữ liệu lưu lượng mạng.
* Mô-đun học máy dựa trên GA (GA-based Machine Learning Module): Thực hiện lựa chọn đặc trưng bằng cách sử dụng thuật toán di truyền.
* Mô-đun phân loại (Classification Module): Sử dụng các bộ phân loại để dự đoán mối đe dọa.

1. **Chuẩn bị dữ liệu (Datasets Preparation)**

* Chuẩn bị dữ liệu là bước đầu tiên của bất kỳ hệ thống học nào. Bước tiền xử lý dữ liệu quan trọng để làm sạch dữ liệu khỏi những giá trị bất thường trước khi nó được trình bày cho mô-đun học máy.
* Dữ liệu lưu lượng mạng được thu thập từ các thiết bị mạng khác nhau và được lưu trữ vào các cơ sở dữ liệu.
* Dữ liệu thô được xử lý qua các bước mã hóa đầu vào/đầu ra (input/output coding), chuẩn hóa (normalization) và chia tỷ lệ (scaling) để chuẩn bị cho các bước tiếp theo:
* Mã hóa đầu vào/đầu ra: Dữ liệu thô được chuyển đổi về dạng phù hợp cho việc xử lý.
* Chuẩn hóa: Dữ liệu được chuẩn hóa để giảm sự chênh lệch giữa các đặc trưng, giúp thuật toán học máy hoạt động hiệu quả hơn.
* Chia tỷ lệ: Dữ liệu được điều chỉnh về một phạm vi nhất định, giúp các mô hình học máy hoạt động ổn định.

1. **Mô-đun học máy dựa trên GA (GA-based Machine Learning Module)**

Như một giải pháp cho vấn đề hiện tại, công việc này sử dụng GA để lựa chọn đặc trưng tối ưu theo cách không giám sát. Ở đây, GA được kiểm tra với các tham số khác nhau được liệt kê trong cài đặt thực nghiệm. GA bao gồm các bước sau: (a) tạo ra dân số ban đầu, (b) tạo ra hàm thích nghi, (c) chiến lược lựa chọn cha mẹ để tạo ra con cháu cho thế hệ tiếp theo, (d) lai ghép, (e) đột biến, và (f) sản xuất thế hệ tiếp theo cuối cùng.

A diagram of a process

Description automatically generated

*Hình 3. Mô tả quy trình hoạt động của GA*

* 1. **Khởi tạo Quần thể (Chromosome structure và initial population)**

1. Mô tả: Tạo ra một quần thể ban đầu của các cá thể (chromosomes), mỗi cá thể là một tập hợp các đặc trưng được chọn ngẫu nhiên từ dữ liệu gốc.
2. Quy trình:

* Tạo dân số ban đầu: Việc tạo dân số ban đầu là bước đầu tiên của GA. Dân số ban đầu được tạo ngẫu nhiên.
* Tạo nhiễm sắc thể: Nhiễm sắc thể được tạo bằng cách kết hợp các gen duy nhất được chọn ngẫu nhiên. Số gen trong mỗi nhiễm sắc thể được chọn ngẫu nhiên từ tập hợp các đặc trưng của dữ liệu gốc mà không có sự trùng lặp.
* Mô tả cấu trúc gen (được biểu diễn trong hình 4):
* Mỗi nhiễm sắc thể là một mảng một chiều có N ô.
* Giá trị của N được thiết lập là 10.
* Mỗi ô của nhiễm sắc thể chứa một giá trị số V, trong đó độ dài tập hợp đặc trưng ≥ V ≥ 0.
* Mỗi giá trị số trong gen chỉ số của nhiễm sắc thể đại diện cho một đặc trưng từ dữ liệu gốc. Ví dụ, nếu gen tại chỉ số 0 có giá trị là 8, nó đại diện cho đặc trưng thứ 8 từ dữ liệu gốc.
* Mỗi nhiễm sắc thể chứa 10 đặc trưng từ tập hợp đặc trưng, tạo thành một tập con được tối ưu hóa sau này để trở thành giải pháp tối ưu.

A diagram of a structure

Description automatically generated

*Hình 4. Cấu trúc gen*

* 1. **Hàm thích nghi (Fitness Function)**

1. Mô tả:

* Hàm Fitness dựa trên tính toán mối tương quan giữa các đặc trưng đã chọn mà không cần nhãn lớp.
* Mục tiêu của hàm là tìm ra các đặc trưng có độ tương đồng thấp.

1. Quy trình:

* Tính toán sự tương quan giữa các đặc trưng trong mỗi cá thể.
* Hàm tính trung bình của mối tương quan sau khi tính toán mối tương quan giữa các đặc trưng.
* Sau khi có giá trị trung bình tương quan, tối ưu hóa được thực hiện để tăng cả giá trị độ chính xác và trung bình tương quan qua các thế hệ GA.
* Biến đổi giá trị trung bình tương quan thành giá trị trung bình không tương quan để đảm bảo tính đa dạng và mối tương quan thấp.
* Các giá trị fitness có thể được tính bằng cách sử dụng giá trị trung bình của mối tương quan đã được biến đổi và độ chính xác của chromosome cụ thể đó.
* Công thức của hàm fitness được nói đến chi tiết trong chương 2, mục I, mục 4.
  1. **Chiến lược Chọn lọc (Selection Strategy)**

1. Mô tả:

* Sử dụng phương pháp lựa chọn bánh xe roulette để chọn các cha mẹ tạo ra con cái cho thế hệ tiếp theo.
* Phương pháp này có ưu điểm là giảm thời gian thực thi khi được triển khai song song và không đòi hỏi việc tỷ lệ/chuẩn hóa như các phương pháp lựa chọn khác

1. Phương pháp lựa chọn bánh xe roulette: Tìm kiếm tuyến tính được sử dụng làm nguyên tắc trong lựa chọn bánh xe roulette, nơi các khe của bánh xe được đo lường bằng giá trị thích nghi của từng nhiễm sắc thể. Nhiễm sắc thể có giá trị thích nghi cao hơn sẽ chiếm nhiều diện tích hơn trên bánh xe roulette, do đó có khả năng cao được chọn.

A pie chart with different colored circles

Description automatically generated

*Hình 5. Phương pháp lựa chọn bánh xe roulette*

1. Quy trình:

* Đánh giá độ phù hợp của từng nhiễm sắc thể.
* Chọn cha mẹ dựa trên giá trị độ phù hợp bằng cách sử dụng bánh xe roulette.
  1. **Lai ghép và Đột biến (Crossover and Mutation)**

1. Mô tả: Tạo ra thế hệ mới thông qua các quá trình lai ghép và đột biến.
2. Quy trình:

* Lai ghép:
* Chọn hai cha mẹ bằng phương pháp bánh xe roulette.
* Chuyển nửa sau của cha mẹ đầu tiên sang cá thể con.
* Thêm các gene còn lại từ cha mẹ thứ hai vào cá thể con theo thứ tự tương ứng.
* Lặp lại quá trình trên cho đến khi số lượng cá nhân cần thiết trong quần thể được hoàn thành.
* Chỉ tạo ra một con bằng cặp bố mẹ được chọn.
* Đột biến:
* Không giống như đột biến truyền thống, đột biến không được áp dụng cho mỗi nhiễm sắc thể con mới được tạo ra.
* Chọn một cha mẹ bằng phương pháp bánh xe roulette.
* Các gene của cả bố mẹ được chuyển trực tiếp sang con.
* Để tăng sự đa dạng hóa, các gene còn lại được chọn ngẫu nhiên trong phạm vi [1- n ] bằng cách tránh sự trùng lặp trong các đặc điểm.

A diagram of a diagram

Description automatically generated

*Hình 6. Quy trình của lai ghép và đột biến*

* 1. **Tiêu chí Dừng (Stopping Criteria)**
* Trong giải pháp đề xuất, có một tiêu chí chấm dứt chung, khi đến điểm đó, quá trình lặp của GA sẽ kết thúc.
* Trong các vòng lặp của GA, các giá trị fitness tối đa của thế hệ được lưu trữ.
* Trong thế hệ đầu tiên, giá trị tối đa Vi được lưu là giá trị fitness tối đa Vf.
* Đối với thế hệ thứ hai, trong quá trình đánh giá nhiễm sắc thể, nếu có giá trị nào lớn hơn Vf, nó sẽ thay thế giá trị hiện tại và giá trị fitness mới được coi là Vf.
* Nếu trong bất kỳ n thế hệ nào, giá trị fitness tối đa của thế hệ đó nhỏ hơn Vf, tiêu chí dừng của GA được đáp ứng vì giá trị fitness không tăng thêm nữa.
* Tuy nhiên, trong quá trình mô phỏng, phương pháp được đề xuất cũng được đánh giá bằng cách thực hiện nó cho một số lần lặp cố định.
* Khi thực hiện GA kết thúc, nhiễm sắc thể tốt nhất đại diện cho bộ đặc trưng tối ưu dự kiến sẽ tạo ra kết quả tốt hơn khi sử dụng với một bộ phân loại chuẩn.

1. **Mô-đun phân loại (Classification Module)**
   1. **Trước khi phân loại:**

* Tập dữ liệu mới được tạo ra chỉ sử dụng những đặc trưng đã được lựa chọn.
* Khi GA hội tụ, chỉ có các đặc trưng được đại diện bởi nhiễm sắc thể tốt nhất được xem xét.
* Dữ liệu sau đó được chia thành các tập huấn luyện và tập kiểm tra bằng phương pháp k-folds.
  1. **Quá trình phân loại:**

Quá trình được thực hiện bằng cách sử dụng tập dữ liệu kiểm tra và huấn luyện cho mỗi nhiễm sắc thể. Công việc hiện tại sử dụng ba bộ phân loại: SVM, k-NN và XgBoost, được nói đến chi tiết trong chương 2, mục I, mục 5.

# **Chương 4: Hiện thực hệ thống**

1. **Các bước cài đặt cấu hình hệ thống**

* Môi trường phát triển: Google Colab
* Ngôn ngữ lập trình: Python 3
* Cài đặt các thư viện cần thiết

1. **Hiện thực hệ thống**
2. **Chuẩn bị dữ liệu**
   1. **CIRA-CIC-DOHBrw-2020:**

* Đọc dữ liệu từ các tệp CSV cho các tập doh, non\_doh, benign, và mal sử dụng pandas.
* Kết hợp các tập dữ liệu doh và non\_doh thành dataset\_l1.
* Kết hợp các tập dữ liệu benign và mal thành dataset\_l2.
  1. **UNSW-NB15:**

Đọc dữ liệu từ các tệp CSV training và testing có sẵn trong tập dataset và lưu vào đường dẫn tương ứng.

* 1. **Bot-Iot**
* Bốn tệp CSV được tải vào các DataFrame của pandas.
* Kiểm tra phân bố các nhãn trong từng tập dữ liệu.
* Lấy mẫu từ các tập dữ liệu và kết hợp để có đầy đủ các nhãn
* Cân bằng tập dữ liệu: tập dữ liệu được điều chỉnh để số lượng mỗi nhãn gần tương đương nhau

1. **Xử lý dữ liệu**

* Sử dụng LabelEncoder để mã hóa các cột dạng chuỗi.

A screen shot of a computer code

Description automatically generated

*Hình 7. Sử dụng LabelEncoder để chuẩn hóa*

* Tỉ lệ hóa dữ liệu sử dụng hàm Min-Max scaling được viết dựa trên công thức (1).

A close-up of a computer code

Description automatically generated

*Hình 8. Hàm min\_max\_scaling để tỉ lệ hóa*

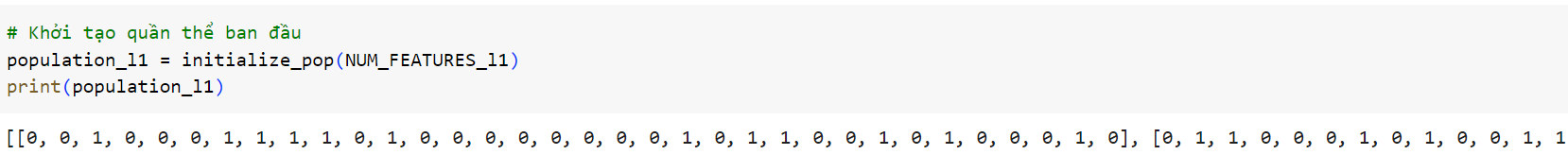
1. **Genetic Algorithm for Feature Selection (GAbFS)**
   1. **Khởi tạo quần thể:**

Hàm này khởi tạo một quần thể ban đầu gồm các cá thể (chromosomes), mỗi cá thể là một danh sách các giá trị 0 hoặc 1, biểu diễn cho việc có hoặc không có mặt của mỗi đặc trưng.

A close-up of a computer screen

Description automatically generated

*Hình 9. Hàm initialize\_pop để khởi tại quần thể*



*Hình 10. Khởi tạo quần thể ban đầu*

* 1. **Fitness function:**
* Hàm fitness\_function đánh giá fitness của mỗi cá thể trong quần thể. Xây dựng hàm tính toán giá trị fitness của mỗi chromosome dựa trên độ chính xác của mô hình Logistic Regression.

A screenshot of a computer code

Description automatically generated

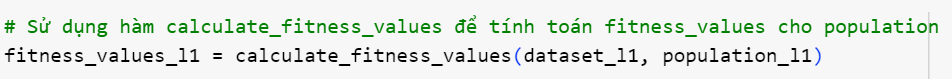
*Hình 11. Hàm fitness function*

* calculate\_fitness\_values: Hàm này tính toán giá trị Fitness cho toàn bộ quần thể của các cá thể. Đối với mỗi chromosome trong quần thể, hàm gọi fitness\_function để tính toán độ chính xác của mô hình sử dụng các đặc trưng từ chromosome đó.

A screenshot of a computer code

Description automatically generated

*Hình 12. Hàm calculate\_fitness\_values*



*Hình 13. Gọi hàm calculate\_fitness\_values*

* 1. **Lựa chọn các cá thể cha mẹ**
* Sử dụng phương pháp chọn lọc bánh xe roulette để chọn cha mẹ từ quần thể, trong đó xác suất chọn một cá thể là tỉ lệ thuận với fitness của nó.

A screenshot of a computer program

Description automatically generated

*Hình 14. Hàm phương pháp chọn lọc bánh xe roulette*

* 1. **Lai ghép (crossover) và đột biến (mutation):** Thực hiện toán tử lai ghép và đột biến để tạo ra thế hệ mới.
* Hàm crossover thực hiện toán tử lai ghép giữa hai cá thể cha mẹ để tạo ra một cá thể con mới.

A screenshot of a computer code

Description automatically generated

*Hình 15. Hàm crossover*

* Hàm mutation thực hiện toán tử đột biến trên một cá thể, tỷ lệ đột biến được xác định bởi mutation\_rate.

A screenshot of a computer code

Description automatically generated

*Hình 16. Hàm mutation*

A computer screen shot of a program code

Description automatically generated

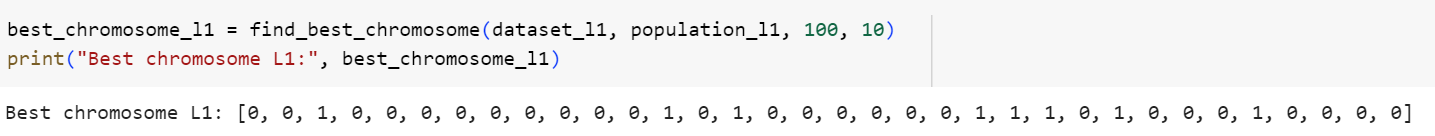
*Hình 17. Sử dụng hàm crossover và hàm mutation*

* 1. **Tìm kiếm chromosome tốt nhất:**
* Chạy vòng lặp qua nhiều thế hệ để tìm kiếm chromosome có giá trị fitness cao nhất.
* Theo dõi và dừng lại khi không có sự cải thiện trong một số thế hệ liên tiếp.

A screenshot of a computer program

Description automatically generated

*Hình 18. Hàm tìm kiếm chromosome tốt nhất*



*Hình 19. Gọi hàm tìm chromosome tốt nhất*

1. **Phân loại sử dụng các bộ phân loại khác nhau**

* Áp dụng các bộ phân loại SVM, KNN, và XGBoost lên tập dữ liệu với các đặc trưng được chọn bởi chromosome tốt nhất.
* Đánh giá độ chính xác của các mô hình sử dụng cross-validation.

A screenshot of a computer screen

Description automatically generated

*Hình 20.*

A screenshot of a computer code

Description automatically generated

*Hình 20 & 21. Hàm sử dụng các bộ phân loại và đánh giá kết quả*

* Kết quả accuracy và recall:
* CIRA-CIC-DOHBrw-2020

A screenshot of a computer program

Description automatically generated

*Hình 22. Kết quả của dataset* *CIRA-CIC-DOHBrw-2020 L1 trên ba bộ phân loại*

A screenshot of a computer program

Description automatically generated

*Hình 23. Kết quả của dataset* *CIRA-CIC-DOHBrw-2020 L2 trên ba bộ phân loại*

A number on a white background

Description automatically generated

*Hình 24. Kết quả tổng quát của dataset* *CIRA-CIC-DOHBrw-2020 trên ba bộ phân loại*

* UNSW-NB15

A screenshot of a computer code

Description automatically generated

*Hình 25. Kết quả của dataset* *UNSW-NB15 trên ba bộ phân loại*

* Bot-Iot

**A screenshot of a computer

Description automatically generated**

*Hình 26. Kết quả của dataset* *Bot-Iot trên ba bộ phân loại*

# **Chương 5: Thực nghiệm và đánh giá**

1. **Kịch bản thực nghiệm**

* Xử lý dữ liệu từ ba bộ dataset: CIRA-CIC-DOHBrw-2020, UNSW-NB15 và Bot-IoT.
* Áp dụng dữ liệu đã được xử lý vô GA-based learning mô đun.
* Tiến hành huấn luyện các mô hình phân loại máy học bằng cách sử dụng các đặc trưng tốt nhất được lựa chọn thông qua mô-đun GA đã được phát triển.
* Đánh giá hiệu suất của phương pháp đề xuất với ba mô hình phân loại: k-Nearest Neighbor (k-NN), Support Vector Machine (SVM), và XgBoost.

1. **Đánh giá kết quả**
2. **Tiêu chí đánh giá**

Kết quả được đánh giá bằng hai tiêu chí chuẩn: accuracy và recall/sensitivity.

* Accuracy: Đo lường hiệu suất của bộ phân loại, là tỷ lệ giữa số quan sát đúng và tổng số quan sát. Giá trị dao động từ 0 (tệ nhất) đến 1 (tốt nhất). Công thức tính:

Trong đó, TP là số lượng mẫu dương tính đúng, FP là số lượng mẫu âm tính bị dự đoán nhầm là dương tính, FN là số lượng mẫu dương tính bị dự đoán nhầm là âm tính, và TN là số lượng mẫu âm tính đúng.

* Recall/Sensitivity: Tỷ lệ giữa số dự đoán dương tính đúng và tổng số mẫu dương tính, dao động từ 0 (tệ nhất) đến 1 (tốt nhất). Còn được gọi là Tỷ lệ Dương tính Thực (TPR) hoặc sensitivity. Công thức tính:

1. **Kết quả** 
   1. **Tìm kiếm chromosome tốt nhất:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Classifier  Tiêu chí | k-NN | XgBoost | SVM |
| CIRA-CIC-DOHBrw-2020 | Accuracy | 0.8709 | 0.8638 | 0.8756 |
| Recall | 0.8781 | 0.8637 | 0.8736 |
| UNSW-NB15 | Accuracy | 0.7927 | 0.7804 | 0.7883 |
| Recall | 0.7927 | 0.7804 | 0. 7883 |
| Bot-IoT | Accuracy | 0.851 | 0.823 | 0.811 |
| Recall | 0.85 | 0.82 | 0.811 |

* 1. **Sau khi sử dụng GA Feature selection**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Classifier  Tiêu chí | k-NN | XgBoost | SVM |
| CIRA-CIC-DOHBrw-2020 | Accuracy | 0.9463 | 0.9533 | 0.9125 |
| Recall | 0.9462 | 0.9531 | 0.9123 |
| UNSW-NB15 | Accuracy | 0.9478 | 0.9656 | 0.9463 |
| Recall | 0.9163 | 0.9098 | 0.9346 |
| Bot-IoT | Accuracy | 0.9466 | 0.9975 | 0.9975 |
| Recall | 0.9466 | 0.9975 | 0.9975 |

# **Chương 6: Kết luận và hướng phát triển**

1. **Kết luận**
2. **Tóm tắt vấn đề**

Trong thế giới hiện đại, lưu lượng và các hacker đang tiến triển, tạo ra những thách thức mới cho các hệ thống phát hiện xâm nhập do sự hiện diện của dữ liệu lớn. Để đạt được hiệu suất tối đa từ một hệ thống phát hiện xâm nhập, vấn đề về số chiều trong các tập dữ liệu lớn cần phải được giải quyết.

1. **Phương pháp**

Phương pháp sử dụng Genetic Algorithm (GA) để lựa chọn đặc trưng, tăng cường độ chính xác của hệ thống IDS. Phương pháp này chọn ra các đặc trưng quan trọng nhất từ các tập dữ liệu IDS. Nghiên cứu này cũng trình bày một hàm mục tiêu mới cho GA mà gán các giá trị thích nghi cho các cá thể trong quần thể GA, cho phép lựa chọn các nhiễm sắc thể đại diện cho tập đặc trưng tối ưu.

1. **Kết quả**

* Huấn luyện các bộ phân loại học máy bằng cách sử dụng các đặc trưng tối ưu được chọn thông qua mô-đun GA đã phát triển.
* Kiểm tra kỹ thuật đề xuất với ba bộ phân loại, cụ thể là k-Nearest Neighbor (k-NN), Support Vector Machine (SVM), và XgBoost.

1. **Hướng phát triển**

* GA được sử dụng ở lõi của giải pháp hiện tại, mà về cơ bản là chậm do nhiều vòng lặp lặp lại và các hoạt động sinh sản. Để giải quyết vấn đề này, áp dụng (1 + 1)-Evolutionary Strategy (ES) để tăng tốc độ tính toán.
* Các thuật toán di truyền hỗn loạn nhị phân gần đây đã cho thấy kết quả hứa hẹn. Trong tương lai, các kỹ thuật tối ưu hóa như vậy có thể được sử dụng để cải thiện hiệu suất tối ưu hóa.
* Nghiên cứu về việc sử dụng phương pháp học không giám sát như phân cụm để tự động hóa việc phát hiện các loại tấn công mới.

# **Q&A**

**Câu hỏi:** Có thể điều chỉnh số lượng đặc trưng đầu ra của thuật toán GA thông qua việc cấu hình các tham số trong quá trình thực thi thuật toán, đặc biệt là ở bước tạo quần thể ban đầu và quá trình chọn lọc đặc trưng.

**Trả lời**

Trong tài liệu, quá trình của GA được mô tả rõ ràng bao gồm các bước: tạo quần thể ban đầu, hàm đánh giá độ thích nghi, chiến lược chọn lọc cha mẹ, lai ghép (crossover), đột biến (mutation), và tạo ra thế hệ tiếp theo​​.

1. Tạo quần thể ban đầu (Initial Population): Ban đầu, các nhiễm sắc thể (chromosome) được tạo ngẫu nhiên, mỗi nhiễm sắc thể đại diện cho một giải pháp cho vấn đề chọn lọc đặc trưng. Mỗi nhiễm sắc thể chứa các gene (đặc trưng) được chọn ngẫu nhiên từ tập dữ liệu gốc.

2. Hàm đánh giá độ thích nghi (Fitness Function): Hàm đánh giá độ thích nghi được thiết kế để tìm ra những đặc trưng có độ tương đồng thấp, tức là các đặc trưng có tính đa dạng cao và có thể đại diện cho phần lớn thông tin của tập dữ liệu​​.

**Câu hỏi:** Có thể điều chỉnh số lượng đặc trưng đầu ra không?

**Trả lời**

Để ép buộc số lượng đặc trưng đầu ra thành 20, ta có thể điều chỉnh các yếu tố sau:

* Cấu hình số lượng đặc trưng trong nhiễm sắc thể: Trong quá trình tạo quần thể ban đầu, ta có thể cấu hình số lượng đặc trưng mà mỗi nhiễm sắc thể chứa. Ví dụ, trong tài liệu, giá trị N (số lượng gene trong mỗi nhiễm sắc thể) được đặt là 10, nhưng có thể điều chỉnh tùy theo yêu cầu thí nghiệm​​.
* Chiến lược chọn lọc (Selection Strategy): Sử dụng chiến lược chọn lọc phù hợp để đảm bảo rằng các nhiễm sắc thể có số lượng đặc trưng mong muốn được chọn lọc và tạo ra thế hệ tiếp theo.
* Điều chỉnh hàm đánh giá độ thích nghi: Điều chỉnh hàm đánh giá độ thích nghi để ưu tiên chọn các nhiễm sắc thể có số lượng đặc trưng là 20. Điều này có thể thực hiện bằng cách thay đổi cách tính toán hoặc thang điểm của hàm đánh giá để khuyến khích việc chọn lọc đúng số lượng đặc trưng cần thiết​​.

Vì vậy: Thông qua việc điều chỉnh cấu hình của nhiễm sắc thể và hàm đánh giá độ thích nghi, ta có thể ép buộc số lượng đặc trưng đầu ra của thuật toán GA thành 20, đáp ứng yêu cầu cụ thể của bài toán.

Trích dẫn (Halim, 2021):

*"Chromosomes are created by joining the randomly selected unique genes... The value of N is set to 10 in this work. However, later a number of experiments have been performed by varying the value of N"​​.*

*"Fitness function provides an instrument to find out those features which have low similarity among them... After the correlation of the given features is calculated, the proposed fitness function computes the correlation average"​​.*

# **Reference**

(Wang, 2020). Wang, J., Liu, C., & Zhou, M. (2020). Improved bacterial foraging algorithm for cell formation and product scheduling considering learning and forgetting factors in cellular manufacturing systems. IEEE Systems Journal, 14(2), 3047-3056.

(Sindhu, 2012). Sindhu, S. S. S., Geetha, S., & Kannan, A. (2012). Decision tree based light weight intrusion detection using a wrapper approach. Expert Systems with applications, 39(1), 129-141.

(Gharaee, 2016). Gharaee, H., & Hosseinvand, H. (2016, September). A new feature selection IDS based on genetic algorithm and SVM. In 2016 8th International Symposium on Telecommunications (IST) (pp. 139-144). IEEE.

(Aslahi-Shahri, 2016). Aslahi-Shahri, B. M., Rahmani, R., Chizari, M., Maralani, A., Eslami, M., Golkar, M. J., & Ebrahimi, A. (2016). A hybrid method consisting of GA and SVM for intrusion detection system. Neural computing and applications, 27, 1669-1676.

(Halim, 2021). Halim, Z., Yousaf, M. N., Waqas, M., Sulaiman, M., Abbas, G., Hussain, M., ... & Hanif, M. (2021). An effective genetic algorithm-based feature selection method for intrusion detection systems. Computers & Security, 110, 102448.

(Yusof, 2022). Yusof, M. H. M., Almohammedi, A. A., Shepelev, V., & Ahmed, O. (2022). Visualizing realistic benchmarked ids dataset: Cira-cic-dohbrw-2020. IEEE Access, 10, 94624-94642.

(Peterson, 2021). Peterson, J. M., Leevy, J. L., & Khoshgoftaar, T. M. (2021, August). A review and analysis of the bot-iot dataset. In 2021 IEEE International Conference on Service-Oriented System Engineering (SOSE) (pp. 20-27). IEEE.

(Moustafa, 2015). Moustafa, N., & Slay, J. (2015, November). 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) (pp. 1-6). IEEE.