**ĐẠI HỌC QUỐC GIA TP.HCM**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KHOA HỌC MÁY TÍNH**

****

**KHÓA LUẬN TỐT NGHIỆP**

**NGHIÊN CỨU MỘT SỐ THUẬT TOÁN MÁY HỌC VÀ**

**HỆ MIỄN DỊCH NHÂN TẠO TRONG**

**PHÁT HIỆN VIRUS MÁY TÍNH**

Giảng viên hướng dẫn : **PGS.TS VŨ THANH NGUYÊN**

Sinh viên thực hiện :

**MAI TRỌNG KHANG 09520373**

**NGUYỄN HOÀNG NGÂN 09520575**

Lớp : **CNTN04**

Niên khóa : **2009 -2013**

***TP. Hồ Chí Minh, tháng 0*7 *năm* 2013**

**LỜI MỞ ĐẦU**

Trong sự phát triển của khoa học, nhiều lý thuyết mới đã được ra đời bằng cách quan sát các hoạt động trong thế giới tự nhiên, học thuyết tiến hóa của Đác-uyn là một ví dụ điển hình. Gần đây, máy tính được sử dụng như một công cụ để nghiên cứu các tiến trình sinh học nhằm đạt được hiểu biết tốt hơn về chúng, cũng như mong muốn áp dụng các nguyên tắc đã được hoàn thiện qua hàng triệu năm tiến hóa của tự nhiên vào giải quyết các bài toán của cuộc sống. Từ đây, nhiều giải thuật và mô hình mô phỏng sinh học được hình thành và phát triển, được ứng dụng rộng rãi và hứa hẹn mở ra các hướng tiếp cận mới cho những bài toán mà con người chưa tìm ra lời giải.

Luận văn tập trung nghiên cứu về hệ miễn dịch nhân tạo (Artificial Immune Systems- AIS), một hệ thống thông minh nhân tạo, giải quyết vấn đề dựa trên các nguyên lý, chức năng và mô hình hoạt động của hệ miễn dịch của con người. Mặc dù có cơ sở sinh học khá phức tạp nhưng các giải thuật chính lại được xây dựng dựa trên các ý tưởng khá đơn giản, mô phỏng các quá trình chính của chức năng miễn dịch trong cơ thể chúng ta.

Từ ý tưởng xem xét đồng nhất một máy tính như một cơ thể sinh vật, nhóm nghiên cứu nảy sinh ý tưởng sử dụng hệ miễn dịch nhân tạo để giải quyết bài toán phát hiện virus máy tính, tương tự như cách hệ miễn dịch sinh học bảo vệ chúng ta trước các tác nhân độc hại từ môi trường. Từ quan điểm này, một hướng tiếp cận mới được hình thành trên cơ sở sự kết hợp của AIS và một số thuật toán máy học khác như mạng nơ ron nhân tạo (Artificial Neural Networks- ANNs), giải thuật gom cụm Kmeans...Mặc dù chỉ mới là hướng tiếp cận bước đầu trong giải quyết bài toán phát hiện virus máy tính, nhưng đề tài mở ra những nghiên cứu xa hơn nhằm giải quyết bài toán bảo mật ở các mức độ lớn hơn.

Nhóm thực hiện hy vọng đề tài nghiên cứu của mình góp phần giải quyết vấn đề bảo vệ an toàn dữ liệu, nâng cao khả năng bảo mật cho các hệ thống Công nghệ thông tin.

**LỜI CẢM ƠN**

Đầu tiên, chúng em xin gửi lời cảm ơn chân thành và lòng biết ơn sâu sắc đến PGS. TS Vũ Thanh Nguyên. Xin cảm ơn thầy đã tận tình quan tâm, hướng dẫn, cung cấp kiến thức, truyền đạt những kinh nghiệm quý báu giúp chúng em hoàn thành tốt khoá luận này.

Chúng em xin cảm ơn anh Nguyễn Phương Anh – giảng viên Khoa Công nghệ phần mềm của trường Đại học Công nghệ thông tin đã sát cánh bên tụi em vượt qua những khó khăn cũng như định hướng, hỗ trợ chúng em trong quá trình thực hiện đề tài.

Chúng em xin gởi lời cảm ơn chân thành đến tập thể quý thầy cô Trường Đại học Công nghệ Thông tin – Đại học quốc gia TP.HCM và quý thầy cô khoa Khoa học máy tính. Trong suốt quá trình đào tạo, những kiến thức mà thầy cô truyền đạt đã làm nền tảng cơ bản để chúng em thực hiện đề tài này. Xin gửi lời cảm ơn đến các anh chị, cùng các bạn đã có những nhận xét, đóng góp ý kiến, động viên, quan tâm và giúp đỡ nhóm tác giả vượt qua giai đoạn khó khăn trong quá trình thực hiện đề tài này.

Cuối cùng, chúng em xin gửi lời cảm ơn gia đình đã tạo mọi điều kiện về vật chất, tinh thần, động viên, khích lệ và hỗ trợ chúng em. Con xin cảm ơn cha mẹ, những người sinh ra con, chăm lo cho con từng miếng ăn giấc ngủ, nuôi nấng, dạy bảo chúng con nên người.

|  |  |
| --- | --- |
|  | *TP. Hồ Chí Minh, ngày 15 tháng 07 năm 2013* |
|  | Nhóm thực hiện |
|  | Mai Trọng Khang – Nguyễn Hoàng Ngân |

**NHẬN XÉT**

**(Của giảng viên hướng dẫn)**

**NHẬN XÉT**

**(Của giảng viên phản biện)**

**MỤC LỤC**

[**Mở đầu**](#_Toc363405385) **trang**

[DANH MỤC CÁC BẢNG BIỂU ix](#_Toc363405386)

[DANH MỤC HÌNH ẢNH, SƠ ĐỒ x](#_Toc363405387)

[DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT xii](#_Toc363405388)

[Chương 1. MỞ ĐẦU 1](#_Toc363405389)

[1.1. Giới thiệu đề tài 1](#_Toc363405390)

[1.1.1. Lý do chọn đề tài 1](#_Toc363405391)

[1.1.2. Mục tiêu của đề tài và kết quả bước đầu thực hiện đề tài 2](#_Toc363405392)

[1.1.3. Các giai đoạn thực hiện đề tài 3](#_Toc363405393)

[1.2. Đối tượng, phạm vi nghiên cứu của đề tài 3](#_Toc363405394)

[1.2.1. Virus và hệ thống đích 3](#_Toc363405395)

[1.2.2. Hệ miễn dịch nhân tạo 4](#_Toc363405396)

[1.2.3. Mạng nơ-ron nhân tạo 5](#_Toc363405397)

[1.3. Các nghiên cứu liên quan 5](#_Toc363405398)

[1.4. Ý nghĩa khoa học và thực tiễn của đề tài 7](#_Toc363405399)

[1.5. Cấu trúc của luận văn 8](#_Toc363405400)

[Chương 2. VIRUS MÁY TÍNH VÀ MỘT SỐ VẤN ĐỀ LIÊN QUAN 9](#_Toc363405401)

[2.1. Virus máy tính 9](#_Toc363405402)

[2.1.1. Định nghĩa 9](#_Toc363405403)

[2.1.2. Hình thức lây nhiễm 9](#_Toc363405404)

[2.1.3. Cơ chế chẩn đoán 12](#_Toc363405405)

[2.2. Mạng Nơ-ron nhân tạo 13](#_Toc363405406)

[2.2.1. Mạng nơ-ron sinh học 13](#_Toc363405407)

[2.2.2. Khái niệm mạng nơ-ron nhân tạo 16](#_Toc363405408)

[2.2.3. Cấu trúc cơ bản của mạng nơ-ron nhân tạo 16](#_Toc363405409)

[2.2.4. Phân loại mạng nơ-ron nhân tạo 35](#_Toc363405410)

[2.2.5. Một số loại mạng nơ-ron nhân tạo 18](#_Toc363405411)

[2.2.6. Cơ chế hoạt động 23](#_Toc363405412)

[2.2.7. Ứng dụng 27](#_Toc363405413)

[2.3. Thuật toán phân cụm K – Means 32](#_Toc363405414)

[Chương 3. HỆ MIỄN DỊCH NHÂN TẠO 35](#_Toc363405415)

[3.1. Khái niệm 35](#_Toc363405416)

[3.2. Cấu trúc cơ bản 36](#_Toc363405417)

[3.3. Các mô hình trừu tượng của hệ miễn dịch và tương tác giữa chúng 38](#_Toc363405418)

[3.3.1. Tổng quát hoá các thành phần sinh học của hệ miễn dịch 38](#_Toc363405419)

[3.3.2. Không gian hình dạng (Shape - Space) 38](#_Toc363405420)

[3.3.3. Đánh giá tương tác giữa các phần tử 40](#_Toc363405421)

[3.4. Một số thuật toán trong hệ miễn dịch nhân tạo 41](#_Toc363405422)

[3.4.1. Thuật toán chọn lọc âm tính 41](#_Toc363405423)

[3.4.2. Thuật toán chọn lọc nhân bản 42](#_Toc363405424)

[3.5. Ứng dụng 47](#_Toc363405425)

[3.5.1. Điều khiển 48](#_Toc363405426)

[3.5.2. An ninh máy tính 48](#_Toc363405427)

[3.5.3. Phát hiện lỗi 48](#_Toc363405428)

[3.5.4. Phát hiện bất thường trong hệ thống 48](#_Toc363405429)

[3.5.5. Tối ưu hóa 49](#_Toc363405430)

[3.5.6. Khai phá dữ liệu 49](#_Toc363405431)

[Chương 4. TIẾP CẬN MÁY HỌC VÀ HỆ MIỄN DỊCH NHÂN TẠO TRONG PHÁT HIỆN VIRUS MÁY TÍNH 50](#_Toc363405432)

[4.1. Ý tưởng 50](#_Toc363405433)

[4.2. Bài toán Xây dựng dữ liệu huấn luyện 51](#_Toc363405434)

[4.2.1. Phát biểu bài toán 51](#_Toc363405435)

[4.2.2 Phân tích bài toán 52](#_Toc363405436)

[4.2.3. Hướng giải quyết 52](#_Toc363405437)

[4.3. Bài toán Xây dựng các bộ phân lớp 54](#_Toc363405438)

[4.3.1. Phát biểu bài toán 54](#_Toc363405439)

[4.3.2. Phân tích 54](#_Toc363405440)

[4.3.3. Hướng giải quyết 55](#_Toc363405441)

[4.4. Bài toán Xây dựng bộ phân lớp tập tin 64](#_Toc363405442)

[4.4.1. Phát biểu bài toán 64](#_Toc363405443)

[4.4.2. Cơ chế xác định mức độ nguy hiểm của một chuỗi nhị phân 65](#_Toc363405444)

[4.4.3. Cơ chế xác định mức độ nguy hiểm của một tập tin 66](#_Toc363405445)

[4.4.4. Xây dựng bộ phân lớp xác định nhãn của một tập tin 67](#_Toc363405446)

[Chương 5. CÀI ĐẶT VÀ THỬ NGHIỆM 69](#_Toc363405447)

[5.1. Cài đặt 69](#_Toc363405448)

[5.1.1. Hiện thực hóa hướng tiếp cận 69](#_Toc363405449)

[5.1.2. Mô-đun hóa các giai đoạn 74](#_Toc363405450)

[5.2. Kết quả thử nghiệm 83](#_Toc363405451)

[5.2.1. Dữ liệu đầu vào 83](#_Toc363405452)

[5.2.2. Xác định thông số 84](#_Toc363405453)

[5.2.3. Trình bày kết quả 87](#_Toc363405454)

[Chương 6. KẾT LUẬN 92](#_Toc363405455)

[6.1. Các kết quả đạt được 92](#_Toc363405456)

[6.2. Hạn chế 93](#_Toc363405457)

[6.3. Hướng phát triển 94](#_Toc363405458)

[PHỤ LỤC 1. CÁC LOẠI VIRUS MÁY TÍNH 96](#_Toc363405459)

[PHỤ LỤC 2. CÔNG TRÌNH ĐÃ CÔNG BỐ 104](#_Toc363405466)

[DANH MỤC TÀI LIỆU THAM KHẢO 107](#_Toc363405467)

[TÀI LIỆU TIẾNG VIỆT 107](#_Toc363405468)

[TÀI LIỆU TIẾNG ANH 107](#_Toc363405469)

# DANH MỤC CÁC BẢNG BIỂU

[Bảng 3.1. So sánh giải thuật tiến hoá cơ bản và giải thuật chọn lọc nhân bản 46](#_Toc363404362)

[Bảng 5.1. Các đối tượng chính của giai đoạn xây dựng dữ liệu huấn luyện 70](#_Toc363404363)

[Bảng 5.2 Các phương thức chính của giai đoạn xây dựng dữ liệu huấn luyện 70](#_Toc363404364)

[Bảng 5.3. Các đối tượng chính hiện thực hóa CLONALG 72](#_Toc363404365)

[Bảng 5.4. Các phương thức chính trong giai đoạn xây dựng các bộ phát hiện 72](#_Toc363404366)

[Bảng 5.5. Các đối tượng và phương thức chính trong giai đoạn xây dựng các bộ phân lớp 73](#_Toc363404367)

[Bảng 5.6. Danh sách thông số của mô-đun xây dựng dữ liệu huấn luyện 76](#_Toc363404368)

[Bảng 5.7. Danh sách các thông số của mô đun xây dựng các bộ phát hiện 79](#_Toc363404369)

[Bảng 5.8. Danh sách thông số của mô-đun xây dựng bộ phân lớp 81](#_Toc363404370)

[Bảng 5.9. Dữ liệu huấn luyện và kiểm tra 84](#_Toc363404371)

[Bảng 5.10. Thử nghiệm sự tương quan giữa giá trị r và quá trinh xây dựng dữ liệu huấn luyện trên Data 1, Data 2, Data 3. 85](#_Toc363404372)

[Bảng 5.11. Dữ liệu huấn luyện và kiểm tra trong thử nghiệm 1 89](#_Toc363404373)

[Bảng 5.12. Kết quả kiểm tra trên dữ liệu virus trong thử nghiệm 1 90](#_Toc363404374)

[Bảng 5.13. Kết quả kiểm tra trên dữ liệu sạch trong thử nghiệm 1 90](#_Toc363404375)

[Bảng 5.14. Dữ liệu huấn luyện và dữ liệu kiểm tra trong thử nghiệm 2 90](#_Toc363404376)

[Bảng 5.15. Kết quả kiểm tra trên dữ liệu virus trong thử nghiệm 2 90](#_Toc363404377)

[Bảng 5.16. Kết quả kiểm tra trên dữ liệu sạch trong thử nghiệm 2 91](#_Toc363404378)

# DANH MỤC HÌNH ẢNH, SƠ ĐỒ

[[Hình 2.1. Cấu tạo nơ-ron sinh học 14](#_Toc361341429)](#_Toc365193765)

[[Hình 2.2. Cấu tạo nơ-ron nhân tạo 16](#_Toc361341429)](#_Toc365193766)

[[Hình 2.3. Cấu trúc mạng nơ-ron đa lớp 18](#_Toc361341429)](#_Toc365193767)

[[Hình 2.4. Mạng dẫn tiến 19](#_Toc361341429)](#_Toc365193768)

[[Hình 2.5. Mạng hồi quy 20](#_Toc361341429)](#_Toc365193769)

[[Hình 2.6. Mạng nơ-ron hàm radial 20](#_Toc361341429)](#_Toc365193770)

[[Hình 2.7. Mạng nơ-ron mờ hồi quy 22](#_Toc361341429)](#_Toc365193771)

[[Hình 2.8. Mô hình Fuzzy ArtMap Neural Network 23](#_Toc361341429)](#_Toc365193772)

[[Hình 2.9. Cấu trúc mạng nơ-ron trong robot di chuyển 27](#_Toc361341429)](#_Toc365193773)

[[Hình 2.10. Mạng nơ-ron huấn luyện logic 28](#_Toc361341429)](#_Toc365193774)

[[Hình 2.11. Mạng nơ-ron nhận dạng mẫu theo thời gian 29](#_Toc361341429)](#_Toc365193775)

[[Hình 2.12. Phân loại kỹ thuật phân cụm 33](#_Toc361341429)](#_Toc365193776)

[[Hình 2.13. Sơ đồ thuật toán K-Means 34](#_Toc361341429)](#_Toc365193777)

[[Hình 3.1. Cấu trúc phân tầng của AIS 37](#_Toc361341429)](#_Toc365193778)

[[Hình 3.2. Kháng thể nhận diện kháng nguyên dựa vào phần bù 39](#_Toc361341429)](#_Toc365193779)

[[Hình 3.3. Mô hình thuật toán chọn lọc âm tính 42](#_Toc361341429)](#_Toc365193781)

[[Hình 3.4. Sơ đồ khối thuật toán chọn lọc nhân bản 44](#_Toc361341429)](#_Toc365193782)

[[Hình 4.1. Cơ chế rút trích chuỗi nhị phân 53](#_Toc361341429)](#_Toc365193783)

[[Hình 4.2. Sơ đồ quy trình xây dựng dữ liệu huấn luyện 54](#_Toc361341429)](#_Toc365193784)

[[Hình 4.3. Sơ đồ giải thuật chọn lọc nhân bản 62](#_Toc361341429)](#_Toc365193785)

[[Hình 4.4. Mạng nơ-ron nhân tạo 64](#_Toc361341429)](#_Toc365193786)

[[Hình 5.1. Mô hình tổng thể của hệ thống 74](#_Toc361341429)](#_Toc365193787)

[[Hình 5.2. Mô hình tổng quát của mô-đun xây dựng dữ liệu huấn luyện 75](#_Toc361341429)](#_Toc365193788)

[[Hình 5.3. Giao diện mô-đun xây dựng dữ liệu huấn luyện 77](#_Toc361341429)](#_Toc365193789)

[[Hình 5.4. Mô hình tổng quát của mô đun xây dựng các bộ phát hiện 78](#_Toc361341429)](#_Toc365193790)

[[Hình 5.5. Giao diện mô đun xây dựng các bộ phát hiện 79](#_Toc361341429)](#_Toc365193791)

[[Hình 5.6. Mô hình tổng quát của mô-đun xây dựng bộ phân lớp 80](#_Toc361341429)](#_Toc365193792)

[[Hình 5.7. Giao diện mô-đun xây dựng bộ phân lớp 81](#_Toc361341429)](#_Toc365193793)

[[Hình 5.8. Mô hình tổng quát mô-đun kiểm tra tập tin 82](#_Toc361341429)](#_Toc365193794)

[[Hình 5.9. Giao diện mô đun kiểm tra tập tin 83](#_Toc361341429)](#_Toc365193795)

[[Hình 5.10. Kết quả thử nghiệm giai đoạn xây dựng dữ liệu huấn luyện 87](#_Toc361341429)](#_Toc365193796)

[[Hình 5.11. Kết quả thử nghiệm giai đoạn xây dựng các bộ phát hiện 88](#_Toc361341429)](#_Toc365193797)

[[Hình 5.12. Kết quả thử nghiệm giai đoạn xây dựng bộ phân lớp 89](#_Toc361341429)](#_Toc365193798)

# DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **Viết tắt** | **Viết đầy đủ** | **Ý nghĩa** |
| CNTT | Công nghệ thông tin | Công nghệ thông tin |
| AV | Anti - Virus | Phần mềm chống virus |
| HĐH | Hệ điều hành | Hệ điều hành |
| ANN | Artificial Neural Network | Mạng nơ-ron nhân tạo |
| GA | Genetic Algorithm | Giải thuật di truyền |
| AIS | Artificial Immune System | Hệ miễn dịch nhân tạo |
| CLONALG | Clonal Selection Algorithm | Thuật toán chọn lọc nhân bản |
| RFNN | Radial Fuzzy Neural Network | Mạng nơ-ron mờ hồi quy |
| NSA | Negative Selection Algorithm | Thuật toán chọn lọc âm tính |
| TD | Training Data | Dữ liệu huấn luyện |
| FTDNN | Fast Time Delay Neural Network | Mạng nơ-ron phản hồi nhanh |
| MAV | Machine Learning Approach to Anti-Virus Expert System | Hệ phòng chống virus máy tính hướng tiếp cận Máy học và Hệ chuyên gia |
| VDCA | Virus Detection Clonal Algorithm | Thuật toán nhân bản phát hiện virus |

Chương 1 MỞ ĐẦU

## 1.1. Giới thiệu đề tài

### Lý do chọn đề tài

Với sự biến đổi mạnh mẽ bộ mặt kinh tế - xã hội loài người do những thành tựu khoa học công nghệ mang lại, nhân loại đã và đang bước vào kỷ nguyên mới - kỷ nguyên của nền kinh tế tri thức mà công nghệ thông tin trở thành một động lực chính, đóng vai trò quan trọng trong tất cả các lĩnh vực của nền kinh tế đó.

Theo đà phát triển của CNTT (Công nghệ thông tin), nhiều loại hình kinh tế mới xuất hiện đáp ứng nhu cầu phát triển của nền kinh tế dựa trên tri thức nhưng cũng đồng thời từ đây, nhiều loại tội phạm mới xuất hiện, khai thác lỗ hỏng bảo mật trong các hệ thống CNTT để tiến hành các hành vi phạm pháp trên quy mô lớn.

Kể từ năm 1981 khi virus đầu tiên xuất hiện trên hệ điều hành của máy tính Apple II, đến nay, vấn đề virus máy tính trở thành một trong 14 thách thức công nghệ lớn nhất của thế kỷ 21 [17].

Ngày nay, việc phòng chống virus không còn là giải pháp đã rồi mà trở thành một vấn đề cấp bách. Hầu hết các chương trình diệt virus (antivirus gọi tắt là AV) đang chiến đấu trong một trận đấu gay go mà cái kết thất bại đã quá rõ ràng. Các hướng tiếp cận hiện tại của các AV ngày càng mất đi tính hiệu quả của mình do sự phát triển, biến đổi không ngừng của các kĩ thuật lập trình virus. Không những vậy, phần lớn các AV chỉ phát huy tác dụng trên những mẫu virus xác định, khả năng dự đoán virus mới còn hạn chế nên thường bị động khi có dịch virus bùng phát.

Các hướng tiếp cận mới không ngừng được nghiên cứu và thử nghiệm nhằm tìm kiếm các giải pháp mới cho bài toán phòng chống virus máy tính nói chung và bài toán phát hiện virus máy tính nói riêng. Theo đà phát triển và xuất hiện của những thuật toán mô phỏng sinh học mới những năm gần đây, nhiều nghiên cứu được tiến hành với mục tiêu khai thác khả năng của các hệ thống sinh học trong giải quyết các vấn đề của con người mà đặc biệt là vấn đề virus máy tính.

Trong bối cảnh đó đề tài, “Nghiên cứu một số thuật toán máy học và hệ miễn dịch nhân tạo trong phát hiện virus máy tính” được tiến hành nhằm nghiên cứu các giải thuật mô phỏng Hệ miễn dịch sinh học và Hệ thần kinh của con người nhằm đưa ra hướng tiếp cận mới giải quyết bài toán phát hiện virus máy tính.

### Mục tiêu của đề tài và kết quả bước đầu thực hiện đề tài

Bài toán phòng chống virus máy tính nói chung và phát hiện virus máy tính nói riêng là một bài toán rất lớn. Do sự hạn hẹp về tri thức cũng như phương tiện, nhóm thực hiện chỉ tập trung giải quyết một phần nhỏ của bài toán này.

Trong phạm vi của đề tài này, nhóm nghiên cứu khả năng ứng dụng của hệ miễn dịch sinh học và mạng nơ ron nhân tạo cũng như một số thuật toán máy học khác trong phát hiện virus máy tính, từ đó:

* Đưa ra một hướng tiếp cận mới giải quyết bài toán, khắc phục nhược điểm của các hướng tiếp cận truyền thống và góp phần nâng cao hiệu quả bảo mật máy tính và các hệ thống CNTT.
* Đề tài không hướng đến việc nhận dạng virus máy tính tức là giải pháp đề ra không có khả năng xác định tên của virus, nhưng bù lại có khả năng phát hiện các virus mới chứa phần mã tương đồng ở mức độ nào đó với các virus trong dữ liệu huấn luyện.
* Xây dựng một chương trình AV nhỏ hiện thực hóa hướng tiếp cận đã đề ra, có khả năng xác định một tập tin nào đó là virus hay chương trình sạch.
* Đặt nền móng cho các nghiên cứu sau này, mở rộng hướng tiếp cận để có thể giải quyết các bài toán lớn hơn cũng như có khả năng ứng dụng trong thực tiễn và thương mại.

Sau khoảng hơn một năm nghiên cứu, thực hiện đề tài, nhóm đã bước đầu đạt được những kết quả khá tốt. Trải qua một quá trình mở rộng và thu gọn liên tục, nhóm đã xây dựng được các phiên bản khác nhau của hướng tiếp cận bài toán phát hiện virus máy tính, nắm được ưu, nhược điểm của các phiên bản này để hoàn thiện hơn hướng tiếp cận thiện tại. Bên cạnh đó, nhóm cũng đã cài đặt thành công hướng tiếp cận này dưới dạng một chương trình nhỏ, mô tả các giai đoạn và quá trình quan trọng. Nhìn chung, những mục tiêu được đặt ra khi thực hiện đề tài đã cơ bản hoàn thành, tuy vậy, vẫn còn nhiều thiếu sót do năng lực của người thực hiện chưa đủ. Nhóm sẽ tiếp tục nghiên cứu và hoàn thiện hướng tiếp cận của mình trong các công trình nghiên cứu xa hơn.

### Các giai đoạn thực hiện đề tài

Quá trình nghiên cứu đề tài được thực hiện qua giai đoạn sau:

* Giai đoạn 1: Khảo sát tình hình virus, hacker, xâm nhập mạng hiện nay;
* Giai đoạn 2: Nghiên cứu các tài liệu liên quan đến hệ miễn dịch nhân tạo;
* Giai đoạn 3: Nghiên cứu các tài liệu liên quan đến mạng nơ-ron nhân tạo;
* Giai đoạn 4: Nghiên cứu hướng tiếp cận virus, đưa ra mô hình kết hợp hệ miễn dịch nhân tạo và một số thuật toán máy học;
* Giai đoạn 5: Cài đặt chương trình thử nghiệm;
* Giai đoạn 6: Đánh giá hiệu năng và định hướng phát triển mô hình.

## Đối tượng, phạm vi nghiên cứu của đề tài

### Virus và hệ thống đích

Có ba vấn đề cân nhắc khi nghiên cứu virus máy tính là môi trường (hệ điều hành, kiến trúc máy), phương tiện (vật chứa tin, cơ chế lan tỏa) và cơ hội (cộng đồng sử dụng, tần suất kích hoạt, kỹ thuật lây lan…). Mặc dù có nhiều loại virus máy tính lây nhiễm trên nhiều hệ thống và môi trường khác nhau, nhưng do tính phổ biến của Windows nên virus máy tính trên hệ điều hành này cũng nhiều hơn.

Để đáp ứng nhu cầu thực tế bức thiết, đề tài tập trung nghiên cứu các loại virus máy tính hoạt động trên các hệ điều hành Windows dành cho máy tính IBM-PC (máy vi tính cá nhân để bàn hoặc xách tay, sử dụng kiến trúc vi xử lý x86/Pentium hoặc tương thích). Trong các loại virus hoạt động trên môi trường HĐH (Hệ điều hành) Windows, đề tài thực hiện thử nghiệm trên loại virus lây nhiễm trên tập tin thi hành (EXE, COM, BAT). Đây là loại virus nguy hiểm vào dạng bậc nhất đối với máy tính. Loại virus này có những tiến trình chạy ngầm phá hoại hệ thống, làm tốn hao bộ nhớ. Một số virus thuộc loại này còn có thể vô hiệu hoá các AV, mở backdoor cho các hacker, mức độ lây lan khó kiểm soát bởi nằm ở thư mục Temp của HĐH, người dùng thường không biết hoạt động của những tập tin này. Trong một số trường hợp, loại virus này giả dạng các biểu tượng phần mềm khiến người dùng tự kích hoạt chúng.

Mặc dù vậy, đề tài cũng được định hướng nghiên cứu để có thể mở rộng kết quả nghiên cứu cho các hệ anti-virus sử dụng các HĐH khác Windows và các loại virus khác.

### Hệ miễn dịch nhân tạo

Hệ miễn dịch nhân tạo AIS (Artificial Immune System)- một phương pháp cách tiếp cận của tin sinh học - là khái niệm chỉ các hệ thống thông minh nhân tạo, giải quyết vấn đề dựa trên các nguyên lý, chức năng và mô hình hoạt động của hệ miễn dịch con người.

Giống như miễn dịch sinh học, hệ miễn dịch nhân tạo có một số đặc trưng chính quan trọng như: chống chịu nhiễu, học tập, ghi nhớ, phân tán và tự tổ chức. Hệ miễn dịch nhân tạo được đánh giá như một phương pháp tính toán mềm mới có hiệu quả. Phạm vi ứng dụng của hệ miễn nhân tạo không chỉ đơn thuần giới hạn ở các bài toán nhận dạng mà nó thực sự thu hút được sự chú ý của các nhà nghiên cứu thông qua những ứng dụng trong các lĩnh vực như bảo mật và an toàn thông tin, học máy, robot học, điều khiển học, tối ưu hoá…

### Mạng nơ-ron nhân tạo

Không giống như hệ miễn dịch nhân tạo chỉ mới vừa xuất hiện gần đây, mạng nơ-ron nhân tạo đã có một quá trình hình thành và phát triển khá dài. Mô hình mạng nơ-ron nhân tạo đầu tiên được Warren McCulloch and Walter Pitts đề ra vào năm 1943 [20]. Cho đến nay, lý thuyết mạng nơ-ron nhân tạo đã được hoàn thiện và củng cố, được ứng dụng rộng rãi trong nhiều lĩnh vực. Từ việc nhìn nhận hiệu quả thực tế của mạng nơ-ron trong giải quyết nhiều bài toán hóc búa của con người, nhóm xác định việc nghiên cứu mạng nơ-ron nhân tạo là cần thiết và hướng tới vận dụng giải quyết bài toán phát hiện virus máy tính.

## Các nghiên cứu liên quan

Vào năm 1995, ANNs (Artificial Neural Networks) được Anastasia Doumas và các cộng sự sử dụng trong việc nhận dạng và phân lớp các virus máy tính. Ở đây, ANNs được cài đặt nhằm nhận dạng một tập các thuộc tính của hoạt động hệ thống khi có sự hiện diện của các virus đặt trưng. Các giải thuật như Back propagation và self-organizing map được chọn để huấn luyện mạng nơ-ron. Công trình nghiên cứu này đã chứng minh được mạng nơ-ron perceptron đa lớp với giải thuật lan truyền ngược là thích hợp giải quyết bài toán, nhóm nghiên cứu cũng chỉ ra một số nhược điểm như việc chọn lựa cấu trúc mạng thích hợp khi mở rộng [4].

Một năm sau đó, Gerald Tesauro và các cộng sự của ông phát triển mạng nơ-ron nhân tạo để phát hiện lớp các virus máy tính riêng biệt. Lớp virus này bao gồm các loại virus lây nhiễm lên BootSecter trên đĩa mềm hoặc Master Boot Record và Disk Bott Record trên đĩa cứng, bảng cấp phát file và thư mục (File Allocation Table-Fat), bảng đăng ký (Windows Registry) của hệ điều hành Windows... Kết quả nghiên cứu cho thấy mạng nơ-ron nhân tạo có thể giải quyết loại virus này với độ chính xác khá cao [9].

Một nghiên cứu khác được thực hiện vào năm 2000 và các tác giả của công trình nghiên cứu này đề ra một phương pháp phát sinh luật mới dựa trên mạng nơ-ron hiện diện. Mạng nơ-ron nhân tạo được tạo thành bằng giải thuật di truyền, sự lây nhiễm virus và sự đột biến nhất định được sử dụng để thể hiện các quy tắc trong dữ liệu huấn luyện. Phương pháp này xây dựng một cấu trúc GA (Genetic Algorithm) dưới dạng mô-đun, mỗi mô-đun học một cấu trúc mạng khác nhau và các mô-đun này giao tiếp thông tin thông qua quá trình lây nhiễm virus. Kết quả mô phỏng và cài đặt trên máy tính chỉ ra hướng tiếp cận này có thể phát sinh ra các cấu trúc mạng như là kết quả các luật đơn giản [14].

Bằng việc sử dụng high-speed time delay neural networks (FTDNNs – fast time delay neural network), Hazem M. El-Bakry và cộng sự đưa ra một hướng tiếp cận thông minh để phát hiện các mã độc chưa được biết đến [11]. Trong mô hình này, toàn bộ dữ liệu được thu thập trong một vecto dài và được kiểm tra như một mẫu đầu vào. FTDNNs sử dụng tương quan chéo trong miền tần số giữa dữ liệu kiểm tra và trọng số đầu vào của mạng nơ-ron nhân tạo.

Vào năm 2009, hai nhà nghiên cứu là Rui Chao và Ying Tan đề ra một hệ thống phát hiện virus dựa trên hệ miễn dịch nhân tạo [15]. Họ sử dụng giải thuật chọn lọc âm tính và chọn lọc nhân bản để phát sinh các bộ phát hiện, tiếp theo, các bộ phát hiện được sử dụng để tính toán mức độ nguy hiểm của một tập tin và xây dựng một bộ phân lớp có khả năng phân biệt các tập tin là virus hay sạch dựa trên mức độ nguy hiểm của nó.

Vào năm 2010, nhóm nghiên cứu của Vladimir Golovko đề ra mô hình kết hợp giữa ANN và AIS [18]. Trong mô hình này, họ mô tả các quy tắc chính để xây dựng hệ thống phát hiện tấn công dựa trên mạng nơ-ron miễn dịch nhân tạo.

Gần đây nhất là vào năm 2013, [Suha Afaneh](http://www.sciencedirect.com/science/article/pii/S1568494612003833)and [Raed Abu Zita](http://www.sciencedirect.com/science/article/pii/S1568494612003833) phát triển một giải thuật dựa trên quan điểm của hệ miễn dịch nhân tạo nhằm mục đích phát hiện virus máy tính, được gọi là giải thuật nhân bản phát hiện virus (Virus detection clonal algorithm - VDCA) [16]. Giải thuật di truyền được sử dụng để tối ưu hóa VDCA. Kết quả thực nghiêm của nghiên cứu cho thấy tỉ lệ phát hiện virus khá cao, 94.4% trong khi tỉ lệ cảnh báo nhầm là 0%.

Riêng ở Việt Nam, cũng đã có một số nghiên cứu được thực hiện nhằm xây dựng một hệ thống phát hiện và nhận dạng virus máy tính. Nổi bật nhất chính là luận án tiến sĩ của thầy Trương Minh Nhật Quang mang tên “Tiếp cận máy học và hệ chuyên gia để nhận dạng, phát hiện virus máy tính” [3]. Như tên gọi của mình, đề tài tập trung vào việc xây dựng một hướng tiếp cận dựa trên phương pháp máy học và hệ chuyên gia để giải quyết bài toán phát hiện, nhận dạng virus máy tính. Đề tài trình bày các cơ chế máy học chẩn đoán virus máy tính. Bằng chiến lược “chia để trị”, bài toán nhận dạng virus máy tính được phân hoạch thành năm bài toán con. Mỗi bài toán được xây dựng dựa vào các kỹ thuật học từ đơn giản đến phức tạp: học vẹt (lớp text virus), học tương tự (lớp macro virus), học chỉ dẫn (lớp boot virus), học tình huống (lớp file virus) và học quy nạp (lớp mã độc, sâu trình và Trojan horse). Trong khi đó, hệ chuyên gia được ứng dụng trong việc xây dựng một hệ cơ sở tri thức về virus máy tính nhằm tư vấn chữa bệnh cho máy dựa trên kinh nghiệm của chuyên gia anti-virus đã tích hợp trong hệ thống. Đề tài của thầy cũng đã xây dựng thành công Hệ phòng chống virus máy tính hướng tiếp cận Máy học và Hệ chuyên gia MAVES- Machine Learning Approach to Anti-Virus Expert System (gọi tắt là MAV).

## Ý nghĩa khoa học và thực tiễn của đề tài

Đề tài hướng đến việc tìm ra một hướng tiếp cận mới nhằm giải quyết bài toán phát hiện virus máy tính, từ đó góp phần đặt nền tảng cho nhiều hướng nghiên cứu xa hơn, nhìn nhận vấn đề từ nhiều hướng khác nhau.

Đề tài được thực hiện phản ánh xu hướng phát triển lĩnh vực CNTT, vận dụng các nghiên cứu xuất phát từ chính tự nhiên để giải quyết các vấn đề của con người.

Đề tài không mong muốn xây dựng một hướng tiếp cận có thể giải quyết tuyệt đối bài toán bảo mật máy tính nói chung cũng như phòng chống virus máy tính nói riêng, mà hướng đến xây dựng một hướng tiếp cận mới, cùng với các hướng tiếp cận sẵn có, góp phần nâng cao hiệu quả bảo mật của các hệ thống CNTT.

Bên cạnh việc giải quyết bài toán phát hiện virus máy tính, hướng tiếp cận trong đề tài hoàn toàn có thể được sử dụng giải quyết một số bài toán khác cũng như mở ra khả năng phát triển và giải quyết các bài toán bảo mật lớn hơn.

Ở nước ta mặc dù đã có nhiều đề tài nghiên cứu về virus máy tính, nhưng hầu hết vẫn giải quyết bài toán nhận dạng biến thể virus máy tính ở các hướng tiếp cận hiện tại, rất ít công trình đề cập tới việc phát triển một hướng tiếp cận mới như kết hợp hệ miễn dịch nhân tạo và thuật toán máy học.

## Cấu trúc của luận văn

Luận văn gồm 6 chương. Chương 1 dẫn nhập và giới thiệu chung về đề tài. Ở chương 2, các tri thức cơ bản về virus máy tính được trình bày, bên cạnh đó là tri thức về mạng nơ-ron nhân tạo và giải thuật phân cụm K-Means. Hệ miễn dịch nhân tạo sẽ được trình bày chi tiết ở chương 3.

Các chương 4 và chương 5 và 6 chứa nội dung chính của đề tài. Chương 4 trình bày hướng tiếp cận để giải quyết bài toán phát hiện virus máy tính dựa trên sự kết hợp của hệ miễn dịch nhân tạo và các thuật toán máy học như mạng nơ-ron nhân tạo và K-Means. Hướng tiếp cận này sẽ được trình bày thông qua việc giải quyết các bài toán con liên quan đến việc xây dựng dữ liệu huấn luyện, xây dựng các bộ phát hiện và xây dựng bộ phân lớp tập tin. Tiếp đó, ở chương 5, hướng tiếp cận được hiện thực hóa dưới dạng một chương trình với nhiều mô-đun riêng biệt, mỗi mô-đun chính là sự cài đặt của hướng giải quyết cho mỗi bài toán con tương ứng. Người dùng tương tác với hệ thống thông qua giao diện để lựa chọn thông số và kiểm soát quá trình thực thi chương trình. Quá trình và kết quả thực nghiệm sẽ được trình bày ở phần cuối chương 5. Chương 6 tổng kết các kết quả mà đề tài đã đạt được, bên cạnh đó là việc phân tích những hạn chế cũng như đề xuất những hướng phát triển tương lai của đề tài.

Phần phụ lục phần phụ lục n ằm làm rõ các vấn đề liên quan đến đề tài, trong đó phụ lục 1 sẽ trình bày về các loại virus máy tính, phụ lục 2 có nội dung liên quan đến một bài báo khoa học mà nhóm đã thực hiện liên quan đến đề tài và được nhận đăng trên “Tạp chí khoa học và công nghệ “.

Phần tài liệu tham khảo bao gồm các tài liệu tiếng Việt và tiếng Anh được sử dụng trong quá trình nghiên cứu đề tài.

Chương 2 VIRUS MÁY TÍNH VÀ MỘT SỐ VẤN ĐỀ LIÊN QUAN

## Virus máy tính

### Định nghĩa

Virus máy tính (computer virus) là loại chương trình máy được thiết kế để thực hiện các chỉ thị của nó sau chương trình khác. Bí mật sao chép bản thân nó vào các hệ thống máy tính, virus lây từ máy này sang máy khác, làm suy giảm năng lực hoạt động hệ thống và xâm phạm dữ liệu người dùng. Kể từ khi virus Brain xuất hiện đầu tiên (1986), ngày nay có đến hàng chục ngàn biến thể virus máy tính lây lan trên toàn thế giới. Biến thái qua nhiều thế hệ, virus máy tính ngày càng tinh vi, mức độ quấy rối phá hoại càng nguy hiểm và nghiêm trọng.

Có nhiều định nghĩa về virus máy tính. Theo Bordera, virus máy tính là:

“Bất cứ chỉ thị, thông tin, dữ liệu hoặc chương trình làm suy giảm tính hoàn thiện của tài nguyên máy tính, làm vô hiệu, gây nguy hiểm hoặc phá hủy, hoặc ghép bản thân nó vào tài nguyên của máy tính khác và thi hành khi chương trình máy tính thi hành”.

Ngoài những tính chất chung, mỗi loại virus có nguyên tắc lây nhiễm, đặc điểm dữ liệu khác nhau. Lúc đầu các chuyên gia phân loại virus theo hình thức lây nhiễm. Về sau virus máy tính phát triển đa dạng với nhiều xu hướng kết hợp, phân hóa, lai tạp, kế thừa… khiến việc phân loại gặp rất nhiều khó khăn.

### Hình thức lây nhiễm

#### 2.1.2.1. Lây nhiễm theo cách cổ điển

Cách cổ điển nhất của sự lây nhiễm, bành trướng của các loai virus máy tính là thông qua các thiết bị lưu trữ di động: Trước đây đĩa mềm và đĩa CD chứa chương trình thường là phương tiện bị lợi dụng nhiều nhất để phát tán. Ngày nay khi đĩa mềm rất ít được sử dụng thì phương thức lây nhiễm này chuyển qua các ổ USB, các đĩa cứng di động hoặc các thiết bị giải trí kỹ thuật số.

#### Lây nhiễm qua thư điện tử

Khi thư điện tử (e-mail) được sử dụng rộng rãi trên thế giới thì virus chuyển hướng sang lây nhiễm thông qua thư điện tử thay cho các cách lây nhiễm truyền thống.

Sau khi đã lây nhiễm vào máy nạn nhân, virus có thể tự tìm ra danh sách các địa chỉ thư điện tử sẵn có trong máy và nó tự động gửi đi hàng loạt (mass mail) cho những địa chỉ tìm thấy. Nếu các chủ nhân của các máy nhận được thư bị nhiễm virus mà không bị phát hiện, tiếp tục để lây nhiễm vào máy, virus lại tiếp tục tìm đến các địa chỉ và gửi tiếp theo. Chính vì vậy số lượng phát tán có thể tăng theo cấp số nhân khiến cho trong một thời gian ngắn hàng hàng triệu máy tính bị lây nhiễm, có thể làm tê liệt nhiều cơ quan trên toàn thế giới trong một thời gian rất ngắn.

Khi mà các phần mềm quản lý thư điện tử kết hợp với các phần mềm diệt virus có thể khắc phục hành động tự gửi nhân bản hàng loạt để phát tán đến các địa chỉ khác trong danh bạ của máy nạn nhân thì chủ nhân phát tán virus chuyển qua hình thức tự gửi thư phát tán virus bằng nguồn địa chỉ sưu tập được trước đó.

Phương thức lây nhiễm qua thư điển tử bao gồm:

* Lây nhiễm vào các file đính kèm theo thư điện tử (attached mail): Khi đó người dùng sẽ không bị nhiễm virus cho tới khi file đính kèm bị nhiễm virus được kích hoạt (do đặc diểm này các virus thường được "trá hình" bởi các tiêu đề hấp dẫn như sex, thể thao hay quảng cáo bán phần mềm với giá vô cùng rẻ).
* Lây nhiễm do mở một liên kết trong thư điện tử: Các liên kết trong thư điện tử có thể dẫn đến một trang web được cài sẵn virus, cách này thường khai thác các lỗ hổng của trình duyệt và hệ điều hành. Một cách khác, liên kết dẫn tới việc thực thi một đoạn mã, và máy tính bị có thể bị lây nhiễm virus.
* Lây nhiễm ngay khi mở để xem thư điện tử: Cách này vô cùng nguy hiểm bởi chưa cần kích hoạt các file hoặc mở các liên kết, máy tính đã có thể bị lây nhiễm virus. Cách này cũng thường khai thác các lỗi của hệ điều hành.

#### Lây nhiễm qua mạng Internet

Theo sự phát triển rộng rãi của Internet trên thế giới mà hiện nay các hình thức lây nhiễm virus qua Internet trở thành các phương thức chính của virus ngày nay. Có các hình thức lây nhiễm virus và phần mềm độc hại thông qua Internet như sau:

* Lây nhiễm thông qua các file tài liệu, phần mềm: Là cách lây nhiễm cổ điển, nhưng thay thế các hình thức truyền file theo cách cổ điển (đĩa mềm, đĩa USB...) bằng cách tải từ Internet, trao đổi, thông qua các phần mềm...
* Lây nhiễm khi đang truy cập các trang web được cài đặt virus (theo cách vô tình hoặc cố ý): Các trang web có thể có chứa các mã hiểm độc gây lây nhiễm virus và phần mềm độc hại vào máy tính của người sử dụng khi truy cập vào các trang web đó
* Lây nhiễm virus hoặc chiếm quyền điều khiển máy tính thông qua các lỗi bảo mật hệ điều hành, ứng dụng sẵn có trên hệ điều hành hoặc phần mềm của hãng thứ ba: Điều này có thể khó tin đối với một số người sử dụng, tuy nhiên tin tặc có thể lợi dụng các lỗi bảo mật của hệ điều hành, phần mềm sẵn có trên hệ điều hành (ví dụ Windows Media Player) hoặc lỗi bảo mật của các phần mềm của hãng thứ ba (ví dụ Acrobat Reader) để lây nhiễm virus hoặc chiếm quyền kiểm soát máy tính nạn nhân khi mở các file liên kết với các phần mềm này.

### Cơ chế chẩn đoán

Có ba kỹ thuật nhận dạng virus máy tính đã được áp dụng: dựa vào chuỗi nhận dạng virus (signature-based approach), dựa vào hành vi nghi ngờ virus (suspicious behavior-based approach) và dựa vào ý định virus (intention-based approach).

#### Phát hiện virus dựa vào chuỗi nhận dạng

Hoạt động theo nguyên lý nhận dạng mẫu, các AV sử dụng một cơ sở dữ liệu chứa mẫu virus (ID-virus library). Mỗi khi có virus mới, các chuyên gia anti-virus sẽ giải mã, trích chọn và cập nhật chuỗi nhận dạng virus vào thư viện. Thông tin về đối tượng chẩn đoán (ghi nhận từ hệ thống đích) cùng với thông tin của virus (trong thư viện mẫu) sẽ cho kết luận về tình trạng của đối tượng. Nhận dạng mẫu giúp AV phát hiện các virus đã biết trên tập dữ liệu chẩn đoán với độ chính xác cao. Tuy nhiên phương pháp này có khá nhiều nhược điểm:

* Cồng kềnh: Kích thước thư viện mẫu tỷ lệ thuận với số virus đã cập nhật và tỷ lệ nghịch với tốc độ tìm kiếm.
* Bị động: AV chỉ hiệu quả trên các mẫu virus đã cập nhật, không đáp ứng kịp thời dịch bệnh do tốn thời gian cho việc thu thập mẫu virus mới, giải mã, phân tích, lập thuật giải, cập nhật phiên bản mới, phát hành…
* Nhầm lẫn: Các hacker cố gắng tạo vỏ bọc an toàn cho virus. Khi AV so mẫu chẩn đoán giống với virus, dữ liệu sạch của hệ thống sẽ bị tẩy (clean) nhầm.

#### Phát hiện virus dựa vào hành vi

Tiếp cận này nghiên cứu virus máy tính dưới góc độ thi hành của tập mã lệnh. Cũng là chương trình máy tính, nhưng khác với các phần mềm hữu ích, virus chỉ chứa các lệnh nguy hiểm. Nghiên cứu trật tự, quy luật hình thành các lệnh máy của virus, tiếp cận này dựa vào khái niệm hành vi để xây dựng cơ chế nhận dạng thông qua tập các thủ tục/hành vi của chúng.

Sử dụng tri thức hành vi từ kinh nghiệm chuyên gia nên tiếp cận này còn gọi là phương pháp heuristic. Do các virus giống nhau thường có hành vi như nhau nên AV có thể nhận dạng các virus cùng họ. Tuy nhiên AV khó phân biệt được các hành vi giống nhau nhưng mục đích khác nhau (ví dụ các phần mềm thường tạo và xóa tập tin tạm, trong khi virus tạo bản sao chính nó và xóa dữ liệu người dùng…) nên tiếp cận này ít được sử dụng cho máy lẻ/trạm làm việc, vốn dành cho người dùng ít kinh nghiệm.

#### Phát hiện virus dựa vào ý định

Do hãng Sandrasoft (Ấn Độ) đề xướng từ năm 2005, tiếp cận intention-based (tên mã Rudra) lưu giữ hình ảnh chi tiết của máy tính trong tình trạng sạch, sau đó tiếp tục theo dõi trạng thái hệ thống. Những thay đổi quan trọng trong tập tin, cấu hình hệ thống hay HĐH đều được cảnh báo như một mối hiểm họa tiềm tàng. Khi những thay đổi này được đánh giá nguy hiểm, hệ sẽ khôi phục máy về tình trạng ban đầu. Mặc dù đơn giản nhưng tiếp cận này tỏ ra khá hiệu quả vì nó có thể bảo vệ máy tính khỏi các mối đe dọa chưa được biết đến, kể cả virus máy tính.

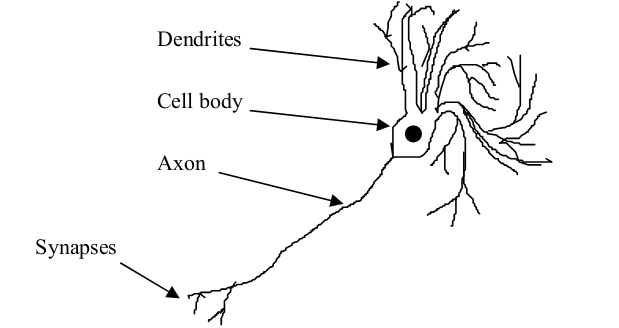
Trong thực tế, tiếp cận “quay về quá khứ” đã được nhiều hãng phần mềm hệ thống sử dụng: Symantec có Norton Ghost và Norton Goback; VMware có System Image Snapshot; Faronics có Deep Freeze… Bản thân Windows XP cũng có chức năng phục hồi hệ thống bằng System Restore. Tuy nhiên tiếp cận này kém hiệu quả khi các điểm trạng thái được ghi nhận lúc hệ thống bị nhiễm virus lạ. Mặt khác, hệ cũng cần bộ nhớ ngoài đủ lớn để lưu toàn bộ hình ảnh hệ thống qua từng thời điểm.

## Mạng Nơ-ron nhân tạo

### Mạng nơ-ron sinh học

#### Bộ não con người

Bộ não của chúng ta bao gồm khoảng 100 tỷ đơn vị xử lý liên kết nhau để tạo thành mạng lưới xử lý thông tin mà ta gọi là các Nơ-ron thần kinh. Mỗi nơ-ron hoạt động như một bộ xử lý đơn giản. Chính sự tương tác khổng lồ giữa tất cả các tế bào này cùng với quá trình xử lý song song của chúng tạo nên khả năng tuyệt vời của bộ não.



Hình 2.1. Cấu tạo nơ-ron sinh học

Các sợi nhánh (Dendrite) là các ống phân nhánh như cành cây từ thân nơ-ron, chúng tiếp nhận tín hiệu từ bên ngoài.

Thân nơ-ron (Cell body) chứa nhân và các cấu trúc khác, hỗ trợ quá trình xử lý hóa học và sản xuất ra chất dẫn truyền thần kinh.

Sợi trục (Axon) là một ống đặc biệt dẫn truyền tín hiệu đến các nơ-ron khác, cơ quan phản ứng (vận động) hoặc vùng đệm. Các sợi trục đi chung với nhau thành từng bó gọi là dây thần kinh.

Khu vực kết thúc của sợi trục tiếp xúc các sợi nhánh của nơ-ron khác được gọi là xi-nap (Synapse)

Tất cả các tác động vào cơ thể đều được xử lý bởi các nơ-ron. Các cảm quan trong mắt của ta (tế bào hình que và hình nón) chính là các nơ-ron mà sợi nhánh biến đổi trở nên nhạy sáng. Còn ở vùng da thì có các nơ-ron cảm giác để cảm nhận nhiệt độ, cơn đau và vô số các nơ-ron khác biến đổi để cảm nhận những thứ xung quanh chúng ta.

* Nơ-ron hướng tâm (nơ-ron cảm giác) có thân nằm ngoài trung ương thần kinh dẫn xung thần kinh về trung ương thần kinh.
* Nơ-ron trung gian (nơ-ron liên lạc) nằm trong trung ương thần kinh, gồm những sợi hướng tâm và li tâm, làm nhiệm vụ liên lạc.
* Nơ-ron li tâm (nơ-ron vận động) có thân nằm trong trung ương thần kinh (hoặc ở hạch thần kinh sinh dưỡng), dẫn các xung li tâm từ bộ não và tủy sống đến các cơ quan phản ứng để gây ra sự vận động hoặc bài tiết

Thông tin tiếp nhận chạy dọc sợi trục của nơ-ron cảm giác đi đến trung ương thần kinh và kết nối đến các nơ-ron khác (nội nơ-ron). Cuối cùng, tín hiệu kết quả quá trình xử lý truyền tới nơ-ron vận động kích thích các cơ quan phản ứng lại tác động từ bên ngoài.

#### Hoạt động của nơ-ron

Ở mức độ đơn giản nhất, khi bị kích thích từ các tác nhân (có thể là các nơ-ron khác hoặc ảnh hưởng bên ngoài) các nơ-ron sản sinh ra các xung thần kinh mà ta gọi là điện thế hoạt động.

Khi một nơ-ron ở trạng thái nghỉ ngơi, nó được phân cực. Mặc dù nó không nhận được bất kì tín hiệu điện nào từ các nơ-ron khác, nó vẫn được nạp và sẵn sàng phát ra các xung thần kinh. Mỗi nơ-ron có một ngưỡng kích thích riêng, nếu tín hiệu tiếp nhận vượt qua ngưỡng này sẽ khiến nơ-ron sản sinh ra các xung thần kinh. Do đó, chỉ khi một nơ-ron nhận đủ kích thích (từ một hay nhiều nguồn) nó mới tạo ra một xung.

Cơ chế truyền tải các xung dựa trên sự trao đổi các ion trong dung môi xung quanh tế bào. Các tín hiệu di chuyển rất chậm chạp, khoảng vài trăm mét mỗi giây.

Các xung thần kinh chỉ kéo dài khoảng vài mili giây. Nếu một nơ-ron nhận rất nhiều kích thích thì sẽ tự sản xuất ra một tín hiệu mạnh, gồm nhiều xung mỗi giây.

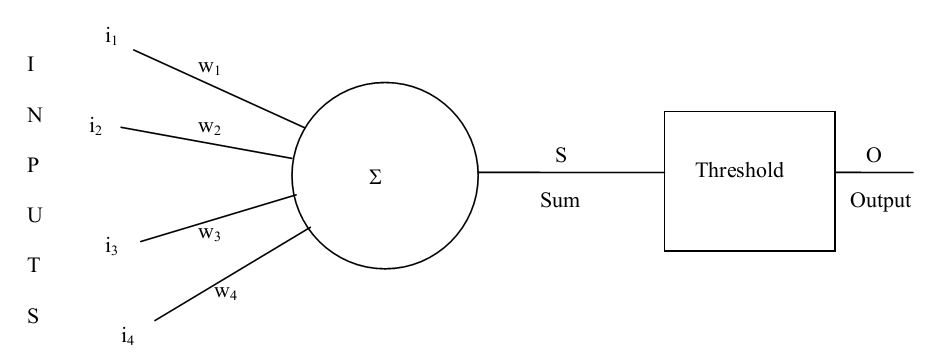
### Khái niệm mạng nơ-ron nhân tạo

Mạng Nơ-ron nhân tạo là mô hình xử lý thông tin được mô phỏng dựa trên hoạt động của hệ thống thần kinh của sinh vật, bao gồm số lượng lớn các Nơ-ron được gắn kết để xử lý thông tin. ANN (Artificial Neural Network – mạng nơ-ron nhân tạo) giống như bộ não con người, được học bởi kinh nghiệm (thông qua huấn luyện), có khả năng lưu giữ những kinh nghiệm hiểu biết (tri thức) và sử dụng những tri thức đó trong việc dự đoán các dữ liệu chưa biết (unseen data).

### Cấu trúc cơ bản của mạng nơ-ron nhân tạo

#### Cấu trúc của một nơ-ron nhân tạo

Mỗi Nơ-ron (nút) là một đơn vị xử lý thông tin của mạng nơ-ron, là yếu tố cơ bản để cấu tạo nên mạng nơ-ron.



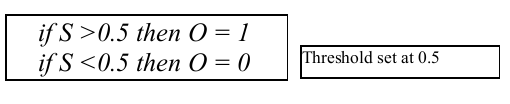
Hình 2.2. Cấu tạo nơ-ron nhân tạo

Đầu vào của một nơ-ron được biểu diễn bởi i (theo quan điểm sinh học, đây chính là các tác động từ các nơ-ron khác kết nối đến hoặc từ thế giới bên ngoài, truyền đến nơ-ron thông qua các sợi nhánh). Mỗi đầu vào tỉ trọng với một hệ số biểu diễn cường độ kết nối xi-nap của các sợi nhánh của nơ-ron, được kí hiệu là w. Tổng các tín hiệu vào và trọng số của nó được gọi là sự hoạt hóa nơ-ron, được kí hiệu bởi S.



Sau khi đã có S, ta sử dụng một ngưỡng để xác định đầu ra của nơ-ron (cấp độ nhị phân đơn giản và ngưỡng t =0.5).

Các nơ-ron tiếp nhận tín hiệu vào, tùy thuộc vào cường độ kết nối mà mỗi tín hiệu được tăng cường hay kìm hãm theo giá trị trọng số, rồi tổng hợp lại và kiểm tra với ngưỡng kích hoạt, nếu trên ngưỡng này, nơ-ron được hoạt hóa.



Ngoài hàm nhị phân để tính đầu ra cho nơ-ron như trên, có thể sử dụng một số hàm kích hoạt khác.

* + Hàm Threhold

f(u)=

* + Hàm piecewise – linear

f(u)=

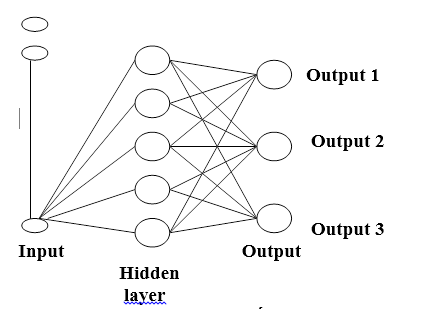
* + Hàm sigmoid (logistic)

f (u) =

* + Hàm tang- hyperbol

f (u) = tang (u) =

#### Cấu trúc của mạng nơ-ron



Hình 2.3. Cấu trúc mạng nơ-ron đa lớp

Cấu trúc chung của ANN gồm 3 thành phần gồm lớp input, lớp output và lớp ẩn, mỗi lớp chứa các nơ-ron.

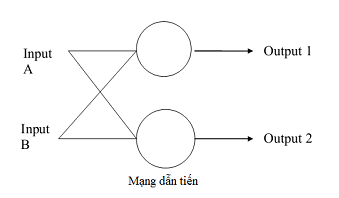
Mỗi Nơ-ron nhận các dữ liệu vào (Inputs) xử lý chúng và cho ra một kết quả (output) duy nhất. Kết quả xử lý của một nơ-ron có thể làm Input cho các nơ-ron khác. Lớp ẩn (Hidden Layer) gồm các nơ-ron, nhận dữ liệu input từ các nơ-ron ở lớp (Layer) trước đó và chuyển đổi các input này cho các lớp xử lý tiếp theo. Trong một ANN có thể có nhiều Hidden Layer.

### Một số loại mạng nơ-ron nhân tạo

#### Mạng dẫn tiến

Trong mạng dẫn tiến liên kết giữa các nơ-ron không tạo thành chu trình. Tín hiệu đi từ các nơ-ron lớp vào lần lượt qua các lớp ẩn và cuối cùng đi ra ở nơ-ron lớp ra. Kiến trúc này đáp ứng nhanh và ổn định đối với một tín hiệu đưa vào mạng. Liên kết giữa các lớp có thể là loại liên kết đầy đủ (fully connected) hoặc liên kết một phần (partly connected).

Giải thuật học của mạng dẫn tiến là giải thuật lan truyền ngược.



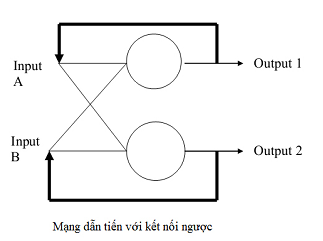
Hình 2.4. Mạng dẫn tiến

Trong các kiểu mạng đã tìm hiểu, kiểu mạng dẫn tiến rất phù hợp với mục tiêu nghiên cứu của nhóm tác giả. Đây là kiểu mạng được trình bày và sử dụng xuyên suốt trong luận văn.

#### Mạng hồi quy Hopfield

Mạng hồi quy của Hopfield cũng có cấu trúc giống như kiểu mạng dẫn tiến, nhưng mạng Hopfield có các kết nối ngược để dòng dữ liệu đầu ra trở lại đầu vào, do đó mạng nơ-ron có thể ghi nhớ.

Hoạt động của mạng hồi quy tương tự như mạng dẫn tiến, các dữ liệu được đưa vào mạng và tính toán giá trị đầu ra. Điểm khác biệt là sau khi giá trị ra được xác định, giá trị này được sử dụng lại để làm giá trị vào cho mạng và quá trình này được lặp lại cho đến khi giá trị đầu ra không thay đổi nữa, ở thời điểm này, mạng nơ-ron đạt đến trạng thái nghỉ.



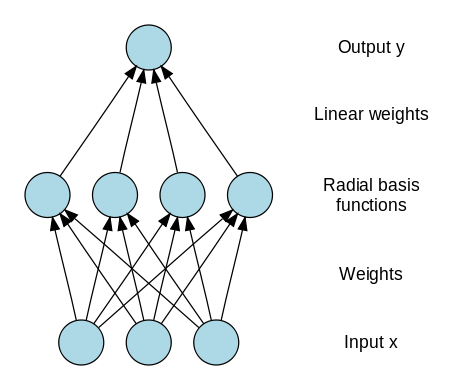
Hình 2.5. Mạng hồi quy

#### Mạng cạnh tranh

Là kiểu mạng học không giám sát, trong đó các nút mạng cạnh tranh để giành được quyền phản hồi lại với tập dữ liệu đầu vào. Kiểu mạng này có những thuộc tính hoàn toàn khác so với mạng lan truyền ngược. Người sử dụng không thực sự kiểm soát nơ-ron nào sẽ chiến thắng và mẫu nào sẽ được học. Ngược lại, mạng có thể tìm thấy những mẫu nào đó trong dữ liệu, ngay cả khi người sử dụng không hề biết liệu mẫu đó có tồn tại ở đó. Nhờ vậy, các mạng kiểu này được sử dụng để xác định những mẫu trong thị trường tiền tệ và giao dịch chứng khoán.

#### Mạng nơ-ron hàm radial

Là kiểu mạng sử dụng hàm radial làm hàm kích hoạt. Dữ liệu nhập qua các hàm radial, tính toán các thông số, các giá trị này được tổ hợp tuyến tính và trả về đầu ra của mạng.



Hình 2.6. Mạng nơ-ron hàm radial

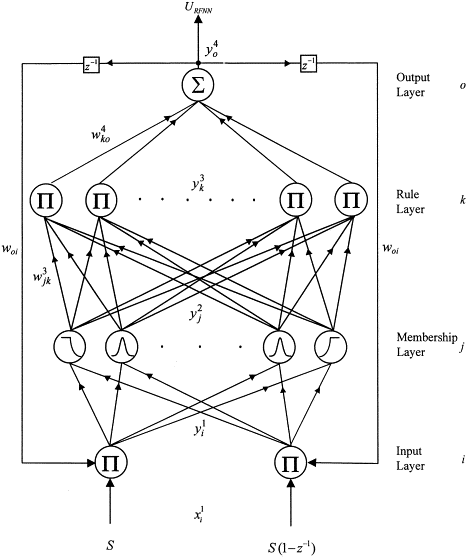
#### Mạng nơ-ron mờ hồi quy (RFNN)

Là sự kết hợp từ lý thuyết tập mờ và mô hình mạng nơ-ron, tận dụng những ưu điểm như khả năng xấp xỉ một hàm liên tục với độ chính xác cho trước, khả năng xử lý tri thức con người.

RFNN (Radial Fuzzy Neural Network) đạt hiệu quả cao trong dự đoán chuỗi thời gian, nhận dạng và điều khiển các hệ phi tuyến.

Cấu trúc của RFNN gồm bốn lớp:

* Lớp 1: Là lớp nhập gồm N dữ liệu nhập
* Lớp 2: Là lớp các hàm thành viên, các nút trong lớp này thực hiện việc mờ hóa
* Lớp 3: Lớp các luật mờ, các nút trong lớp này tạo thành cơ sở của luật mờ
* Lớp 4: Lớp xuất gồm P nút.

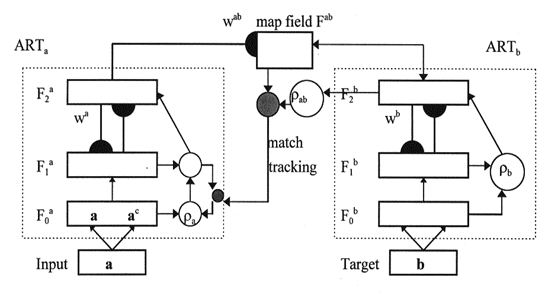


Hình 2.7. Mạng nơ-ron mờ hồi quy

#### Mô hình Fuzzy ArtMap Neural Network

Hệ thống Fuzzy ArtMap Neural Network là mạng nơ-ron học có giám sát, có khả năng nhận dạng nhanh và ổn định, có khả năng dự báo chính xác cho phân lớp.

Cấu trúc của mạng là cấu trúc tiến hóa và là sự kết hợp của hai mạng fuzzy ART (ARTa – ARTb). Hai mạng này liên kết với nhau bởi nhau mạng Neural MAP Field. Đây cũng là một kiểu mạng cạnh tranh.



Hình 2.8. Mô hình Fuzzy ArtMap Neural Network

### Cơ chế hoạt động

Tùy theo từng kiểu mạng mà giải thuật học có sự khác nhau. Các giải thuật học này đảm bảo các tham số tự do của mạng nơ-ron được điều chỉnh lại cho phù hợp thông qua một quá trình tương tác với môi trường. Ở đây, chúng ta tìm hiểu giải thuật học lan truyền ngược, giải thuật được sử dụng trong mạng dẫn tiến.

#### Cơ chế học có giám sát

Quá trình học giám sát được tiến hành trên một tập dữ liệu mẫu với giá trị được phân loại (gán nhãn) sẵn.

Tập dữ liệu luyện gồm:

S= {xi, cj| i = 1,…, M; j = 1,…, C}

Trong đó: xi là vectơ n chiều (gọi là đặc trưng của dữ liệu)

cj là sốlớp biết trước

Thuật toán học sẽ tìm kiếm trên không gian giả thuyết giải pháp tốt nhất cho ánh xạ f với c = f(x).

Mô hình học gồm có:

* Học vẹt: Hệ thống luôn được dạy những luật đúng. Nếu tạo ra một quyết định không đúng, hệ sẽ đưa ra các luật/quan hệ đúng đã sử dụng.
* Học bằng phép loại suy: Hệ thống được dạy phản hồi đúng cho một công việc tương tự nhưng không xác định. Vì thế hệ thống phải hiệu chỉnh phản hồi trước đó bằng cách tạo ra một luật mới có thể áp dụng cho trường hợp mới.
* Học dựa trên trường hợp: Hệ thống học lưu trữ tất cả các trường hợp cùng với kết quả đầu ra tạo ra các trường hợp này. Khi gặp một trường hợp mới, hệ thống sẽ cố gắng hiệu chỉnh đến trường hợp mới này về cách xử lý trước đó đã được lưu trữ.
* Học dựa trên sự giải thích: Hệ thống sẽ phân tích tập hợp những giải pháp nhằm chỉ ra tại sao mỗi phương pháp là thành công hay không thành công. Sau khi được tạo ra, những giải thích này được dùng để giải quyết những vấn đề mới.

#### Cơ chế học không giám sát

Hệ thống khai thác dữ liệu được ứng dụng với những đối tượng nhưng không có lớp được định nghĩa trước, do đó hệ thống tự quan sát các mẫu và nhận ra mẫu, dẫn đến một tập lớp, mỗi lớp có một tập mẫu được khám phá trong tập dữ liệu.

Trong trường hợp chỉ có ít hay gần như không có tri thức về dữ liệu đầu vào, hệ thống học không giám sát sẽ khám phá ra những phân lớp của dữ liệu bằng cách tìm ra những đặc trưng, thuộc tính chung của mẫu hình thành nên tập dữ liệu.

#### Giải thuật học lan truyền ngược

Thuật toán lan truyền ngược là phương pháp học có giám sát. Từ các dữ liệu mẫu đầu vào, thuật toán điều chỉnh dần các trọng số của mạng nơ-ron cho phù hợp, nghĩa là mạng sẽ học thông qua những sai sót của nó. Do đó, sau khi kết thúc quá trình huấn luyện, với một đầu vào cụ thể, mạng cho ra một đầu ra mong muốn.

Thuật toán này là dạng tổng quát của luật delta [6]. Thuật toán gồm 2 giai đoạn, đó là lan truyền và cập nhật trọng số.

* + - * Giai đoạn 1: Lan truyền

Với mỗi lan truyền thực hiện các bước sau:

Bước 1: Lan truyền về trước 1 mẫu dữ liệu huấn luyện đầu vào trong mạng để sinh ra việc lan truyền các kích hoạt đầu ra.

Bước 2: Lan truyền ngược các kích hoạt đầu ra trong mạng sử dụng điểm đích mẫu huấn luyện để sinh ra các delta của tất cả đầu ra và các nơ-ron ẩn.

* + - * Giai đoạn 2: Cập nhật trọng số

Với mỗi trọng số thực hiện các bước sau:

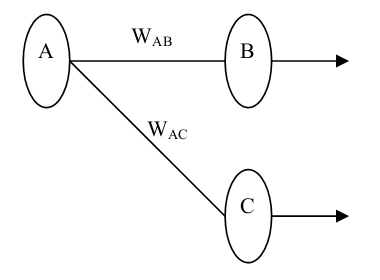
Bước 1: Lấy tích của delta đầu ra và kích hoạt đầu vào để lấy độ giảm của trọng số.

Bước 2: Đưa trọng số vào theo chiều ngược lại của độ giảm bằng cách trừ đi 1 tỉ lệ của nó từ trọng số.

Tỉ lệ này phụ thuộc vào tốc độ và chất lượng của việc học, gọi là tốc độ học. Dấu hiệu của độ giảm trọng số biểu thị qua việc tăng lỗi.

Lặp lại 2 giai đoạn cho tới khi mạng thỏa mãn.

Để hiểu rõ hơn cách mà thuật toán lan truyền ngược hoạt động, ta xét qua một ví dụ với một mạng nơ-ron đơn giản:



Chúng xem xét kết nối giữa nơ-ron A ( thuộc lớp ẩn) và nơ-ron B (nơ-ron đầu ra) có trọng số WAB. Giải thuật được tiến hành như sau :

1. Đầu tiên, ta áp dụng các đầu vào cho mạng và tính toán giá trị đầu ra ( có thể mang giá trị bất kì do các trọng số được tạo ngẫu nhiên ).
2. Kế đến, ta tính toán sai số cho nơ-ron B. Sai số được tính bằng (Giá trị mong muốn- giá trị thu được thực sự)

Cần chú ý rằng, OutputB(1- OutputB) là cần thiết do ta sử dụng hàm Xích- ma, có thể được bỏ qua khi ta dùng hàm ngưỡng.



1. Điều chỉnh trong số : Đặt W+AB là giá trị trọng số mới và WAB là giá trị trọng số ban đầu. Ta có :



Ta cần chú ý rằng, OutputA là đầu ra của nơ-ron A không phải B hay chính là đầu vào của kết nối. Đây là cách mà ta dùng để điều chỉnh các trọng số trong lớp xuất.

1. Ta tính tiếp sai số của các nơ-ron lớp ẩn, các sai số này không thể tính toán trực tiếp do không tồn tại giá trị đích cho chúng, ta tiến hành lan truyền ngược từ lớp xuất. Ta sử dụng các sai số đã tính trước đó từ lớp xuất đề tính các sai số của lớp ẩn. Xét ví dụ ở trên, Nơ-ron A kết nối đến nơ-ron B và C nên ta sử dụng các sai số của B và C để tính A như sau :

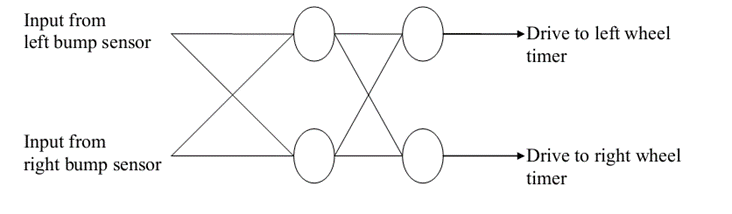


Sau khi đã tính được giá trị sai số cho nơ-ron lớp ẩn, ta lặp lại bước 3 để huấn luyện trọng số lớp ẩn này . Và bằng cách này, ta có thể đào tạo một mạng nơ-ron với số lớp bất kì.

### Ứng dụng

#### Ứng dụng trong khoa học người máy

Các ứng dụng phức tạp trong khoa học người máy của mạng nơ-ron bao gồm control of drive and manipulators, hệ thống quan sát và hệ thống cảm biến khác, bộ cung ứng năng lượng thông minh. Dưới đây là một mạng nơ-ron đơn giản minh họa một robot, hãy cùng quan sát một phương tiện di chuyển với hai cảm biến va chạm phía trước:

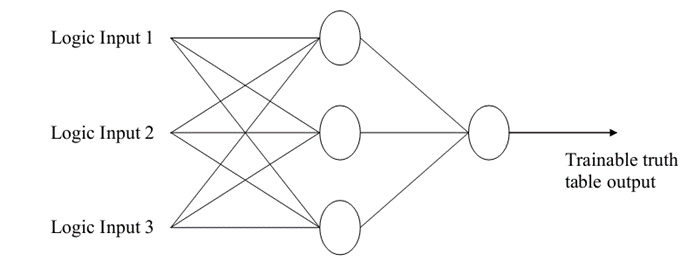


Hình 2.9. Cấu trúc mạng nơ-ron trong robot di chuyển

Mạng nơ-ron này có thể được huấn luyện bằng giải thuật lan truyền ngược để di chuyển qua vật cản trên đường đi của robot. Tập huấn luyện bao gồm những đầu vào tương ứng với các tính thế khác nhau mà robot có thể nhận ra được và các hành vi đích cho phép robot né được vật cản. Ta sẽ cần thêm hệ thống mạch điện nhằm định thời cho phép robot xử lý trong những tính huống phức tạp, ví dụ như lái ra khỏi góc đường (khi đó động cơ bên trái hoặc phải sẽ được tắt trong khoản thời gian ngắn rồi được bật lại).

#### Hệ logic huấn luyện phổ biến

Robot điều khiển như trên là một ví dụ của mạng nơ-ron được huấn luyện để tạo ra một bảng chân trị. Mạng nơ-ron có thể được xem như một hệ logic phổ biến, có khả năng học để tạo ra bất kì bảng chân trị nào.



Hình 2.10. Mạng nơ-ron huấn luyện logic

Nếu đầu ra của nơ-ron là hàm Xích-ma, thì các nơ-ron có thể xem như là loại logic mờ, và tạo ra các đầu ra tuần tự, hữu ích trong xử lý các vấn đề của thế giới thực. Hai ưu điểm khi sử dụng mạng nơ-ron như một hệ logic là khả năng kháng nhiễu và không cần quan tâm đến thiết kế chi tiết của nó.

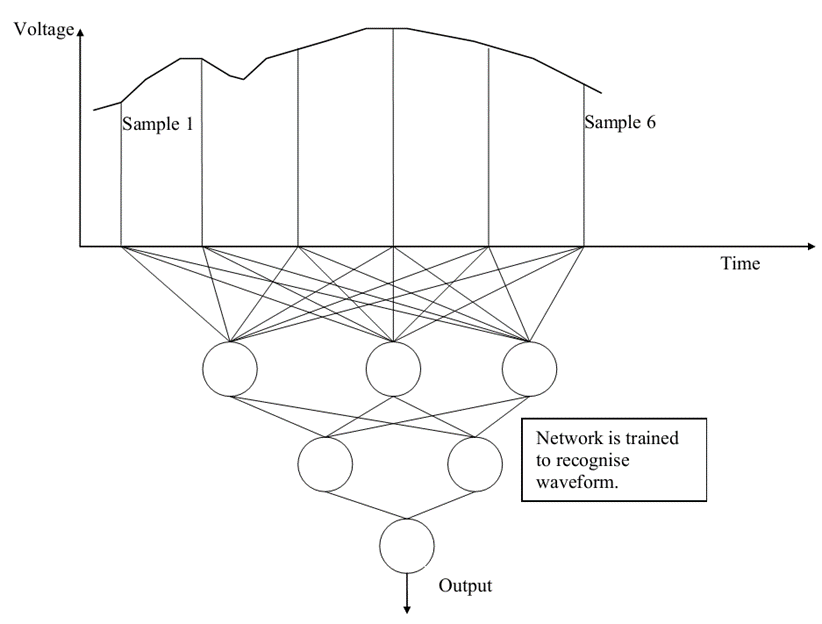
#### Ứng dụng trong thương mại điện tử

Trong lĩnh vực thương mại điện tử, mạng nơ-ron nhân tạo được ứng dụng rất phổ biến, mà đặt biệt là trong đồ gia dụng. Ví dụ như điều khiển tự động trong máy giặt. Dữ liệu đầu vào của mạng có thể là khối lượng quần áo, lượng bột giặt, độ cứng của nước và chế độ giặt do người dung chọn. Đầu ra có thể là nhiệt độ nước, kiểu chu kì giặt và độ dài mỗi công đoạn giặt. Ưu điểm của hệ thống là khả năng nội suy các phản ứng theo các loại tải trọng mà nó chưa từng được huấn luyện. Tương tự, ta có các ứng dụng của mạng nơ-ron trong máy hút bụi, máy rửa chén, lò nướng và các dụng cụ nấu nướng khác.

Các ứng dụng tương tự cũng có thể được tìm thấy bên ngoài căn nhà của ta, ví dụ như hệ thống quản lý máy tự động được điều khiển bởi mạng nơ-ron, với các đầu vào là tốc độ máy, momen xoắn, áp lực, nhiệt độ và đầu ra là thời gian và các dạng pha trộn cần để đạt hiệu quả tốt nhất.

#### Nhận dạng sóng và xử lý tín hiệu

Một mạng nơ-ron có thể nhận dạng được các mẫu theo thời gian, như một dạng sóng, bằng cách lấy mẫu chúng và chuyển thành đầu vào:



Hình 2.11. Mạng nơ-ron nhận dạng mẫu theo thời gian

Mạng nơ-ron này có thể được sử dụng trong các ứng dụng điều khiển, vì mạng thực hiện một hàm ánh xạ, nó có thể được huấn luyện để xử lý các dữ liệu dạng sóng và thực hiện các biến đổi toán học lên chúng để tạo thành các hàm xử lý tín hiệu số (DSP). Mạng nơ-ron chỉ có thể nhận ra dữ liệu dạng sóng nếu nó cùng kích thước với dữ liệu mà mạng được huấn luyên, nếu dạng sóng này nhanh hoặc chậm hơn sẽ khiến mạng không thể nhận ra đó là cùng một mẫu.

#### Ứng dụng y sinh học và cảm biến thông minh

Trong các hệ thống điều khiển được liệt kê trong các chương trước, mạng phản ứng lại các cảm biến trên mô-tô và tạo các đầu ra thích hợp. Một ưu điểm của mạng nơ-ron là khả năng xử lý các đầu vào tương quan phức tạp. Người sử dụng sẽ không cần phải hiểu chi tiết cách hệ thống làm việc mà chỉ cần đưa ra các ví dụ để huấn luyện mạng. Thông qua quá trình học, mạng nơ-ron sẽ tự nhận thức được quan hệ phức tạp giữa các tín hiệu vào và ra.

Các hệ thống loại này rất hiệu quá trong các máy điều khiển trong các điều kiện lỗi tiềm ẩn, theo đó, các lỗi này có thể được xác định trước khi gây ra các hỏng hóc đáng kể.

Ví dụ, dòng điện và điện áp của các động cơ mô tô điện và máy phát điện có thể được điều khiển và xử lý bằng một mạng nơ-ron được huấn luyện để tìm kiếm các dấu hiệu điện của một tình trạng lỗi. Các thông số vật lí tương tự như rung động, áp lực, nhiệt độ và sức căng có thể được sử dụng trong các máy cơ như động cơ chạy bằng xăng. Các tập mẫu huấn luyện có thể được tạo bằng cách gây ra các lỗi cụ thể trong môi trường thử nghiệm dưới các điều kiện được kiểm soát.

Một ứng dụng khác của hệ cảm biến thông minh được sử dụng trong lĩnh vực y sinh học. Nhiều cảm biến y học theo dõi nhịp tim, máu hoặc các thay đổi hóa chất và hoạt động thần kinh tạo các đầu ra phức tạp. Mạng nơ-ron cũng có thể được huấn luyện để thống nhất các đầu vào từ các lĩnh vực khác nhau như các thông tin lý, hóa.

Trong một ví dụ, khi ta muốn tạo ra các bộ phận (tay chân) giả và muốn hệ thần kinh nhận dạng được nó là một cộng việc vô cùng khó khăn. Vì ta sẽ phải xử lý các tín hiệu từ rất nhiều dây thần kinh, các tín hiệu này có thể độc lập nhau trong khi các tín hiệu khác liên quan chặt chẽ và phức tạp. Khả năng xử lý thông tin loại này của một hệ thống phi tuyến và phức tạp cũng là một điểm trội của mạng nơ-ron.

#### Khả năng dự đoán

Một lĩnh vực cuối đáng được đề cập trong các ứng dụng của mạng nơ-ron là khả năng dự đoán các hành vi tương lai của hệ thống bằng cách huấn luyện nó dựa trên các kết quả đã biết để dự đoán những gì sẽ xảy ra khi mạng phát hiện ra một tình huống tương tự. Ví dụ như khả năng dự đoán xu hướng trong thị trường tài chính và hành vi xã hội. Ở mức độ kĩ thuật, mạng có thể được dùng để xác định loại hệ thống và phản ứng lại trong cài đặt điều khiển.

#### Ứng dụng trong quản lý xây dựng

**Một số nghiên cứu ứng dụng ANN trong quản lý xây dựng**

**1. Florence Yean Yng (Singapore) và MinLiu (USA)**: Đã nghiên cứu “ứng dụng neural network để dự báo kế hoạch thực hiện xây dựng dự án ở Singapore”. Theo nghiên cứu này tác giả đã thực hiện trên 11 phép đo, 65 nhân tố tác động đến sự thành công của 33 dự án trước đây. Chỉ ra 6 phép đo có thể dự báo sự thành công của kế hoạch thực hiện dự án với mức độ chính xác hợp lý: tầm quan trọng dự án, tốc độ xây dựng, tốc độ giải quyết vấn đề khó khăn, sự luân phiên thay thế công nhân nghỉ việc, chất lượng của hệ thống và thiết bị.

**2. Về lĩnh vực dự toán chi phí cho một dự án**: Tarek Hegazy và Amr Ayed đã ứng dụng ANN để dự toán chi phí cho một dự án đường cao tốc.

**3. Hoijat Adeli và Mingyang Wu** dự toán giá bê tông cốt thép vỉa hè cho dự án xây dựng đường.

**4.** **Xishi Huang, Danny Ho, Jing Ren, Luiz F.Caprest** đã ứng dụng Neural Fuzzy phát triển mô hình COMOCO, trên cơ sở dữ liệu kinh nghiệm của các dự án trước đây, mô hình này cho phép ước lượng được chi phí của các dự án khác lớn hơn và được ứng dụng trong ước lượng chi phí của dự án công nghiệp.

**5. Jason Portas và Simaan AbouRizk** ứng dụng mô hình Neural network để đánh giá khả năng sản xuất xây dựng.

**6. Irem Dikmen và M. Talat Birgonul** ứng dụng ANN để đánh giá thị trường quốc tế trong quyết định thực hiện dự án, dựa trên các nhân tố của một dự án quốc tế gồm: nguồn tiền, khối thị trường, sự thành công của nền kinh tế, kiểu hợp đồng, hệ số rủi ro.

**7. Hashem Al- Tabtabai** ứng dụng ANN để xây dựng mô hình phân tích kinh nghiệm và hệ thống dự báo cho dự án xây dựng. Trên cơ sở dữ liệu các dự án trước đây về: tiến độ thực hiện của nhà thầu, dòng ngân lưu, vật liệu và thiết bị, thời tiết và môi trường, phần trăm công việc hoàn thành, chất lượng, giá cả vật tư, thuế, bảo hiểm...để dự báo phần trăm thay đổi của kế hoạch, phần trăm thay đổi của chất lượng, công nhân lao động sản xuất, thay đổi mức lương lao động, thay đổi giá vật tư, thay đổi chi phí đầu tư, thay đổi chi phí trong quá trình thực hiện dự án.

**8. VK Gupta, JG Chen, MB Murtaza** đã ứng dụng để phân loại các dự án xây dựng công nghiệp, theo các biến quan hệ: vị trí xây dựng, lao động, vấn đề về tổ chức, đặc điểm khu vực, rủi ro của dự án và môi trường.

## Thuật toán phân cụm K – Means

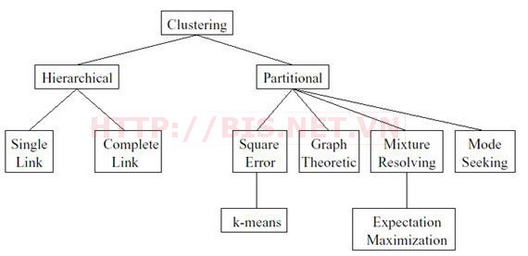
Phân cụm là kỹ thuật rất quan trọng trong khai phá dữ liệu, nó thuộc lớp các phương pháp học không giám sát trong Máy học. Có rất nhiều định nghĩa khác nhau về kỹ thuật này, nhưng về bản chất ta có thể hiểu phân cụm là các qui trình tìm cách nhóm các đối tượng đã cho vào các cụm (clusters), sao cho các đối tượng trong cùng 1 cụm tương tự (similar) nhau và các đối tượng khác cụm thì không tương tự (Dissimilar) nhau.

Mục đích của phân cụm là tìm ra bản chất bên trong các nhóm của dữ liệu. Các thuật toán phân cụm (Clustering Algorithms) đều sinh ra các cụm (clusters). Tuy nhiên, không có tiêu chí nào là được xem là tốt nhất để đánh hiệu quả của phân tích phân cụm, điều này phụ thuộc vào mục đích của phân cụm như: data reduction, “natural clusters”, “useful” clusters, outlier detection.

Kỹ thuật phân cụm có thể áp dụng trong rất nhiều lĩnh vực như:

* Marketing: Xác định các nhóm khách hàng (khách hàng tiềm năng, khách hàng giá trị, phân loại và dự đoán hành vi khách hàng,…) sử dụng sản phẩm hay dịch vụ của công ty để giúp công ty có chiến lược kinh doanh hiệu quả hơn;
* Biology: Phận nhóm động vật và thực vật dựa vào các thuộc tính của chúng;
* Libraries: Theo dõi độc giả, sách, dự đoán nhu cầu của độc giả…;
* Insurance, Finance: Phân nhóm các đối tượng sử dụng bảo hiểm và các dịch vụ tài chính, dự đoán xu hướng (trend) của khách hàng, phát hiện gian lận tài chính (identifying frauds);
* WWW: Phân loại tài liệu (document classification); phân loại người dùng web (clustering weblog); …

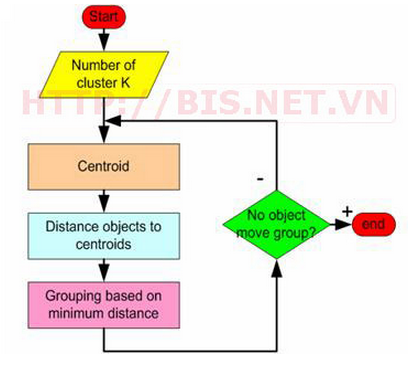
Các kỹ thuật phân cụm được phân loại như sau:



Hình 2.12. Phân loại kỹ thuật phân cụm

K-Means là thuật toán rất quan trọng và được sử dụng phổ biến trong kỹ thuật phân cụm. Tư tưởng chính của thuật toán K-Means là tìm cách phân nhóm các đối tượng (objects) đã cho vào K cụm (K là số các cụm được xác đinh trước, K nguyên dương) sao cho tổng bình phương khoảng cách giữa các đối tượng đến tâm nhóm (centroid) là nhỏ nhất.

Sơ đồ thuật toán:



Hình 2.13. Sơ đồ thuật toán K-Means

Thuật toán K-Means thực hiện qua các bước chính sau:

* Bước 1. Chọn ngẫu nhiên K tâm (centroid) cho K cụm (cluster). Mỗi cụm được đại diện bằng các tâm của cụm
* Bước 2. Tính khoảng cách giữa các đối tượng (objects) đến K tâm (thường dùng khoảng cách Euclidean)
* Bước 3. Nhóm các đối tượng vào nhóm gần nhất
* Bước 4. Xác định lại tâm mới cho các nhóm
* Bước 5. Thực hiện lại bước 2 cho đến khi không có sự thay đổi nhóm nào của các đối tượng.

Chương 3 HỆ MIỄN DỊCH NHÂN TẠO

### Phân loại mạng nơ-ron nhân tạo

Có nhiều cách phân loại mạng nơ-ron nhân tạo. Mỗi cách phân loại ứng với mỗi đặc trưng của mạng nơ-ron nhân tạo.

Theo cách sắp xếp nơ-ron:

* Mạng một lớp (single - layer) là mạng chỉ có một lớp ra.
* Mạng đa lớp (multiple – layer) là mạng có các lớp ẩn.

Theo cách liên hệ giữa các nơ-ron:

* + - Mạng truyền thẳng (feedforward networks)
    - Mạng hồi quy (recurrent networks)

Ngoài ra, theo sự phân bố các nơ-ron trong không gian hai chiều trong một lớp, gọi là liên kết bên (lateral connection). Từ liên kết này Kohonen đã tạo ra mạng nơ-ron tự tổ chức (self – organizing neural network).

Theo thuật toán học:

* Học có giám sát (supervised)
* Học không giám sát (unsupervised)

## Khái niệm

Trong sự phát triển của khoa học, nhiều lý thuyết mới đã được ra đời bằng cách quan sát các hoạt động trong thế giới tự nhiên, phát minh vật lý của Niutơn gắn với hiện tượng quả táo rơi là một ví dụ điển hình. Gần đây, máy tính được sử dụng như một công cụ để nghiên cứu các tiến trình sinh học nhằm đạt được hiểu biết tốt hơn về chúng, cũng như mong muốn áp dụng các nguyên tắc đã được hoàn thiện qua hàng triệu năm tiến hóa của tự nhiên vào giải quyết các bài toán của cuộc sống. Các ý tưởng xuất phát từ sinh học đã làm xuất hiện một số lĩnh vực nghiên cứu mới như: mạng nơron nhân tạo, mạng nơron tế bào, giải thuật di truyền, sự sống nhân tạo, otômat tế bào, tin sinh học… Hệ miễn dịch nhân tạo (Artificial Immune System - AIS) là một cách tiếp cận của tin sinh học, đó là khái niệm chỉ các hệ thống thông minh nhân tạo, giải quyết vấn đề dựa trên các nguyên lý, chức năng và mô hình hoạt động của hệ miễn dịch của con người.

Có rất nhiều định nghĩa về hệ miễn dịch nhân tạo nhưng có thể đưa ra bốn định nghĩa cơ bản sau:

* Định nghĩa 1: “Hệ miễn dịch nhân tạo là các phương pháp thao tác, phân lớp, biểu diễn và phân tích dữ liệu dựa trên sơ đồ sinh học đáng tin cậy là hệ miễn dịch của con người” (Starlab)
* Định nghĩa 2: “Hệ miễn dịch nhân tạo là một hệ thống tính toán bắt chước hệ miễn dịch tự nhiên” (Timmis, 2000)
* Định nghĩa 3: “Hệ miễn dịch nhân tạo là phương pháp tính toán thông minh lấy ý tưởng của hệ miễn dịch hướng vào việc giải quyết các bài toán trong thực tế” (Dasgupta - 1999)
* Định nghĩa 4: “Hệ miễn dịch nhân tạo là một hệ thống thích nghi lấy ý tưởng của miễn dịch học thuyết và những chức năng, nguyên tắc, mô hình miễn dịch quan sát được, áp dụng giải các bài toán thực tế” (Castro & Timmis -2002)

Trong các định nghĩa trên định nghĩa thứ 4 là rõ ràng và chặt chẽ hơn cả, định nghĩa này rất phù hợp với những nội dung mà luận văn sẽ trình bày.

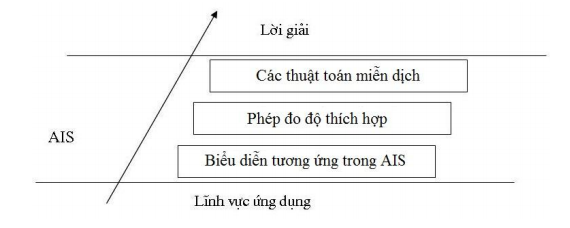
## Cấu trúc cơ bản

AIS có cấu trúc phân tầng, điều này đảm bảo các yếu tố cơ bản của một hệ thống phỏng sinh học:

- Biểu diễn các thành phần của hệ thống.

- Cơ chế đánh giá tương tác của các cá thể với môi trường và các cá thể với nhau. Môi trường thường mô phỏng bởi một tập kích thích vào, một hoặc nhiều hàm đo độ thích nghi của cá thể với môi trường.

- Các thủ tục thích nghi điều khiển tính động của hệ thống, tức là làm cho hoạt động của hệ thống thay đổi theo thời gian.



Hình 3.1. Cấu trúc phân tầng của AIS

Tầng đầu tiên là lĩnh vực ứng dụng. Tầng này có vai trò quyết định thành phần hệ thống, cách biểu diễn và thao tác trên các thành phần. Với lĩnh vực ứng dụng khác nhau thì các thành phần này cũng khác nhau.

Tầng thứ hai là biểu diễn tương ứng trong AIS. AIS cần phải biểu diễn được hai thành phần quan trọng: kháng thể và kháng nguyên. Bên cạnh đó là các thành phần khác nếu có yêu cầu từ các tầng khác trong hệ thống.

Tầng thứ ba là các phương pháp đánh giá độ thích hợp. Tùy thuộc vào bài toán, cách biểu diễn các thành phần có thể sử dụng nhiều phương pháp khác nhau như khoảng cách Hamming, khoảng cách Euclid, hoặc khoảng cách Mahattan.

Tầng thứ tư là các thuật toán miễn dịch. Có thể sử dụng một hoặc kết hợp nhiều thuật toán như chọn lọc tích cực, chọn lọc tiêu cực, chọn lọc Clonal, thuật toán aiNet, thuật toán RAIN để điều chỉnh tính động của hệ AIS.

Tầng thứ năm là lời giải cho bài toán. Lời giải của bài toán sẽ được cập nhật lại sau khi một quần thể mới được tạo ra và đưa ra kết quả cuối cùng khi đạt đến điều kiện kết thúc nào đó, chẳng hạn như sau một số lần lặp nhất định.

## Các mô hình trừu tượng của hệ miễn dịch và tương tác giữa chúng

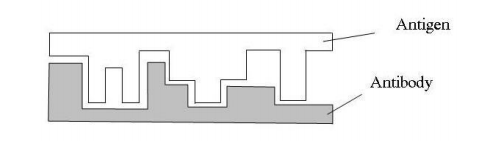
### Tổng quát hoá các thành phần sinh học của hệ miễn dịch

Trong hệ miễn dịch bạch cầu (leukocyte) là phần tử hoạt động chính của hệ miễn dịch; bao gồm nhiều loại khác nhau như bạch cầu hạt (granulocyte) là đại thực bào (macgophage) trong hệ miễn dịch bẩm sinh và lympho bào bao gồm B-cell và T-cell trong hệ miễn dịch thích nghi. Mỗi bạch cầu lại có những cơ quan thụ cảm riêng trên bề mặt (như B-cell Receptor - BCR, T-cell Receptor - TCR) có khả năng nhận diện và kết hợp với các loại phần tử khác nhau.

Để tạo một quy ước chặt chẽ và có tính bao hàm, ta giả thiết rằng các cơ quan thụ cảm trên bề mặt của mọi bạch cầu được gọi chung là kháng thể và phần tử mà các cơ quan thụ cảm này nhận diện được gọi chung là kháng nguyên. Khả năng nhận diện của các cơ quan thụ cảm với phần tử gây bệnh được gọi là độ thích hợp (Affinity hay Degree of match) của kháng thể với kháng nguyên.

### Không gian hình dạng (Shape - Space)

Kết hợp với kháng thể và kháng nguyên chính là sự kết hợp của khối lồi và lõm trên bề mặt 2 phần tử như hình 3.2. Do vậy để mô tả hình dạng tổng quát của kháng thể và kháng nguyên cũng như thể hiện sự kết hợp giữa chúng, ta có thể biểu diễn kháng thể và kháng nguyên là một tập các khối lồi, lõm với 3 tham số chiều cao, chiều rộng và chiều dài. Tổng quát hơn ta có thể biểu diễn kháng thể và kháng nguyên như một tập L tham số.



Hình 3.2. Kháng thể nhận diện kháng nguyên dựa vào phần bù

Như vậy mỗi kháng thể và kháng nguyên có thể coi như một chuỗi thuộc tính m=<m1, m2… mL> trong không gian hình dạng S, L chiều. Mỗi phần tử chính là một điểm trong không gian S.

Chuỗi thuộc tính có thể được tạo bởi bất kì kiểu thuộc tính nào như kiểu số thực, số nguyên, bit và ký hiệu. Kiểu thuộc tính của các phần phụ thuộc vào bài toán và là yếu tố quan trọng để từ đó xác định tương tác của phần tử. Có một số kiểu không gian hình dạng chủ yếu sau:

* Không gian thực (Real - shape - space): Chuỗi thuộc tính có các phần tử nhận giá trị thực.
* Không gian nguyên (Integer - shape - space): Chuỗi thuộc tính có các phần tử nhận giá trị nguyên.
* Không gian Hamming (Hamming - shape - space): Chuỗi thuộc tính có các phần tử nhận một giá trị nào đó trong bảng hữu hạn k mẫu tự.
* Không gian ký hiệu (Symbolic - shape - space): Thường tạo bởi các thuộc tính có kiểu khác nhau, trong đó có ít nhất một thuộc tính là kiểu ký hiệu. Ví dụ như không gian các chuyến bay có thể tạo từ các thuộc tính: mô tả, ngày, chuyến bay, nước xuất phát, nước kết thúc và giá. Mỗi thuộc tính này có kiểu dữ liệu khác nhau có giá trị của nó nằm trong một vùng nào đó như thuộc tính nước (kiểu ký hiệu) có thể chọn trong tập {Brazil, U.K., U.S.A}.

Không mất tính tổng quát ta có thể giả sử kháng thể và kháng nguyên có cùng độ dài là L. Tổng quát một kháng thể được biểu diễn bởi vector Ab=<Ab1,Ab2,…,AbL> và một kháng nguyên được biểu diễn bởi vector Ag=<Ag1,Ag2,…,AgL> để xác định tương tác của các kháng thể với nhau và kháng thể với kháng nguyên có thể dùng công thức tính khoảng cách để đo độ thích hợp của 2 phần tử.

### Đánh giá tương tác giữa các phần tử

#### Không gian Euclid và không gian Mahattan

Không gian toạ độ thực dùng công thức tính toán khoảng cách 3.1 được gọi là không gian Euclid, dùng công thức tính toán khoảng cách 3.2 được gọi là không gian Mahattan.

 (3.1)

 (3.2)

Để đánh giá độ thích hợp giữa các thành phần của AIS, trong không gian thực có thể dùng khoảng cách Euclid và khoảng cách Mahattan.

#### Không gian Hamming

Không gian hình dạng Hamming, các kháng nguyên và kháng thể được biểu diễn như các dãy ký hiệu trên một tập hữu hạn k mẫu tự. Công thức tính toán cách Hamming (3.3) dùng để đánh giá độ thích hợp giữa 2 chuỗi thuộc tính có độ dài L trong không gian Hamming. Với k=2 ta có không gian Hamming nhị phân, k=3 ta có không gian Hamming tam phân.

 ,



Tuỳ thuộc vào bài toán, một không gian nguyên cũng có thể được áp dụng. Trong không gian này các thuộc tính tương ứng với các biến có giá trị nguyên. Không gian nguyên có thể được xem như một trường hợp đặc biệt của không gian Hamming.

Các công thức trên không chỉ dùng để biểu diễn tương tác giữa các kháng thể với kháng nguyên, mà còn có thể dùng để biểu diễn tương tác của kháng thể với kháng nguyên. Ta có thể coi các công thức trên là những công thức chung để xác định tương tác giữa các phần tử trong hệ miễn dịch.

Ngoài ra để xác định tương tác giữa các phần tử, ta quy định một ngưỡng ε nào đó, tuỳ từng bài toán và từng trường hợp cụ thể nếu khoảng cách D giữa 2 phần tử nhỏ hơn hoặc lớn hơn ngưỡng ε ta có 2 phần tử tương tác với nhau, hay có thể nhận diện được nhau.

## Một số thuật toán trong hệ miễn dịch nhân tạo

AIS có nhiều thuật toán phỏng theo cơ chế của hệ miễn dịch sinh học bao gồm: Chọn lọc dương tính (Positive Selection), chọn lọc âm tính (Negative Selection), chọn lọc nhân bản (Clonal Selection) và mạng miễn dịch... Trong phạm vi của đề tài, nhóm thực hiện tập trung vào hai thuật toán được trình bày sau đây.

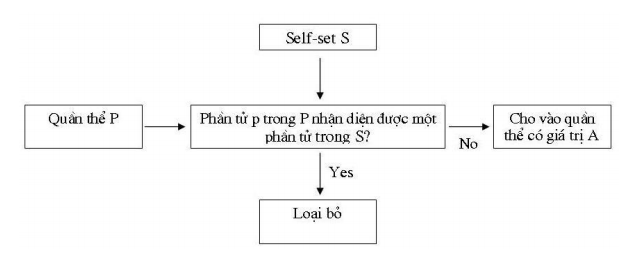
### Thuật toán chọn lọc âm tính

Trong hệ miễn dịch sinh học, tế bào lympho T trưởng thành trong tuyến ức, trải qua một quá trình chọn lọc gọi là chọn lọc tiêu cực. Trong phép chọn này, các tế bào lympho T mà nhận dạng các tế bào của cơ thể sẽ bị tiêu diệt [12] [21]. Sau khi chọn lọc, các lympho T không nhận dạng các tế bào của cơ thể được phóng thích và cung cấp khả năng bảo vệ cơ thể chống lại các kháng nguyên. Cơ chế này trong hệ miễn dịch rất mạnh vì sự phân loại tự nhiên và hiệu quả cao của nó.

Các bước chính của thuật toán chọn lọc âm tính NSA (Negative Selection Algorithm):

* Bước 1. Khởi tạo: Sản sinh một quần thể tiềm năng P những T-cell chưa trưởng thành. Giả thiết tất cả các phần tử (các cơ quan thụ cảm và các selfpeptide) được biểu diễn bằng một chuỗi nhị phân L bit.
* Bước 2. Đánh giá độ thích hợp: Xác định độ thích hợp của tất cả T-cell trong P với mọi phần tử của tập self-set S
* Bước 3. Tạo một quần thể có giá trị: Nếu độ thích hợp của một T-cell chưa trưởng thành với ít nhất một phần tử self-peptide lớn hơn hoặc bằng một ngưỡng tương tác chéo nào đó, thì T-cell nhận diện được self-peptide này và bị loại bỏ, trái lại T-cell được bổ sung vào quần thể có giá trị A.

Mô hình thuật toán:



Hình 3.3. Mô hình thuật toán chọn lọc âm tính

### Thuật toán chọn lọc nhân bản

Thuật toán chọn lọc nhân bản CLONALG (Clonal Selection Algorithm) được sử dụng trong AIS để xác định các đặc điểm cơ bản của một đáp ứng miễn dịch đối với một kháng nguyên. Trong thuật toán này, chỉ các kháng thể nhận diện được kháng nguyên mới được chọn để sinh trưởng. Các kháng thể được chọn trải qua một quá trình thành thục ái lực để củng cố ái lực của chúng đối với các kháng nguyên.

Giải thuật chọn lọc nhân bản được phát triên dựa trên lý thuyết chọn lọc nhân bản (Burnet, 1959) được đưa ra gần 50 năm trước [5]. Các nhân tố miễn dịch chính được sử dụng trong giải thuật này là:

* Sự duy trì tập nhớ cụ thể
* Sự chọn lọc và nhân bản phần lớn các kháng thể được kích thích.
* Sự đào thải các kháng thể mang tính kích thích yếu hoặc không được kích thích.
* Sự trưởng thành ái lực (siêu đột biến) của các tế bào miễn dịch đã được kích hoạt.
* Phát sinh và duy trì tập kháng thể đa dạng.

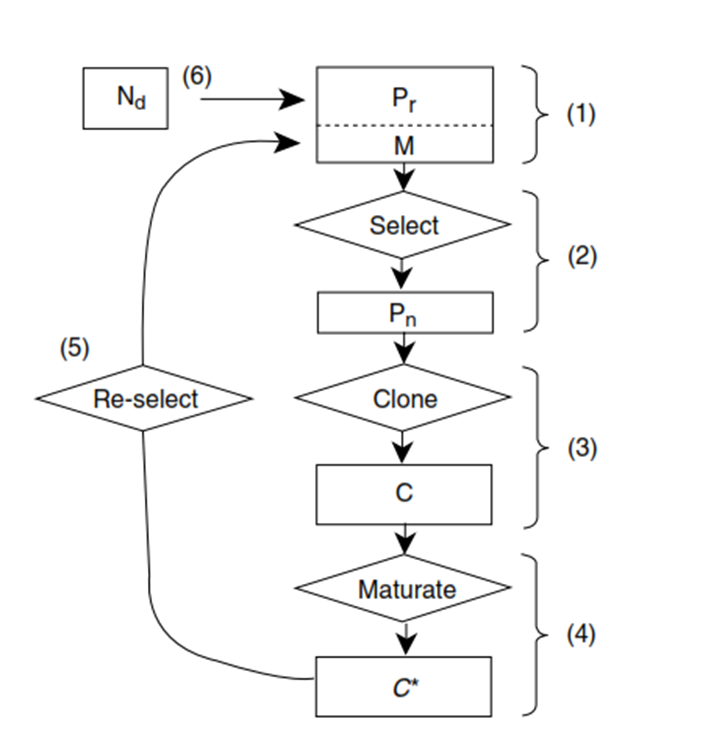
Giải thuật chọn lọc nhân bản (De Castro và Von ZUben, 2000) [13], rất giống với một loại giải thuật tiến hóa mang tên chiến lược tiến hóa (Beyer và Schwefel 2002) [10], mặc dù chúng bắt nguồn từ hai quá trình sinh học khác nhau. Các giải thuật chọn lọc nhân bản cũng là dạng tìm kiếm trên quần thể và các thuật toán tối ưu sản sinh ra một vùng nhớ các kháng thể phù hợp để giải quyết một vấn đề cụ thể nào đó.

Trong các giải thuật chọn lọc nhân bản, mỗi kháng thể và kháng nguyên được biểu diễn bởi 1 tập các thuộc tính {x1, x2 ... xn}. Theo đó, các kháng thể và kháng nguyên có thể được biểu diễn như các điểm n chiều trong không gian (không gian Euclidean chẳng hạn) hoặc sử dụng mã hóa nhị phân các thuộc tính; bên cạnh đó, các dạng biểu diễn khác cũng được sử dụng.

Ái lực kháng nguyên của mỗi kháng thể được đặc trưng bởi giá trị đo, thường là khoảng cách Euclidean. Một số toán tử được xác định để gây ra sự biến đổi gen trên kháng thể dựa trên ái lực kháng nguyên của chúng. Đầu tiên, một phép nhân bản được xác định để tạo ra một bản sao chính xác của những kháng thể có ái lực kháng nguyên cao; ái lực này càng cao, số lượng bản sao càng lớn. Khi đó, một vài biến đổi gen được áp dụng lên các kháng thể này (thông qua phép đột biến) để chúng có khả năng gắn khớp với kháng nguyên tốt hơn.

Mặc dù một vài biến thể của giải thuật chọn lọc nhân bản đã được giới thiệu, song phần lớn chúng đều có chung những đặc điểm như giải thuật gốc.

Trong suốt quá trình trưởng thành ái lực, khi những kháng thể đã biến đổi gen được thêm vào quần thể hiện tại để tái chọn lọc cá thể tốt nhất và giữ những cá thể tốt nhất này làm vùng nhớ của kháng nguyên thực tại, ta cần tính các ái lực giữa các kháng thể mới hướng đến kháng nguyên; toàn bộ tập kháng thể cần được xếp hạng, tiếp theo sau đó là một quá trình chọn lọc.



Hình 3.4. Sơ đồ khối thuật toán chọn lọc nhân bản

Thuật toán chọn lọc nhân bản:

1. Khởi tạo: Tạo một quần thể ngẫu nhiên các kháng thể P

2. Trình diện kháng nguyên

**While** các điều kiện dừng chưa thỏa **do**

**For each** kháng nguyên **do**

2.1 Ước lượng ái lực

Trình diện kháng nguyên hiện tại cho kháng thể hiện tại quần thể P

**For each** kháng thể **do**

Tính ái lực giữa kháng thể hướng đến kháng nguyên hiện tại

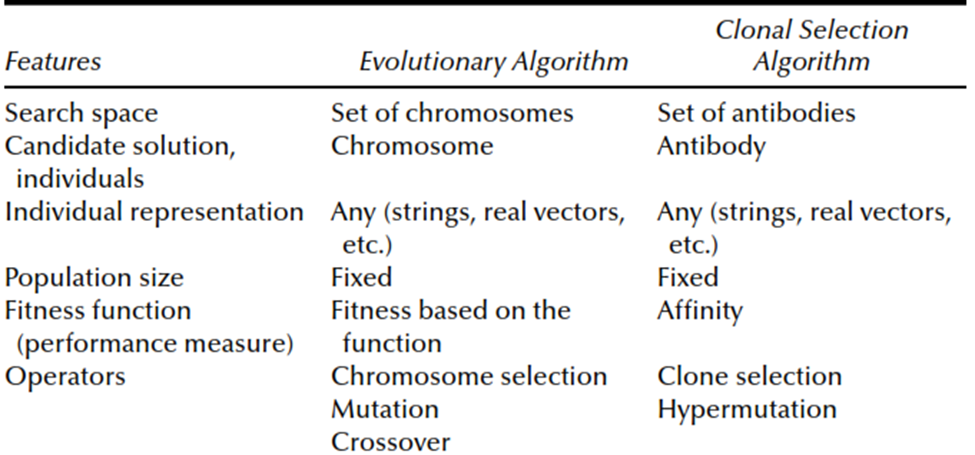
2.2 Chọn lọc và mở rộng nhân bản

Chọn ra 1 nhóm các kháng thể trong P có ái lực cao nhất.

Phát sinh các bản sao chính xác của các kháng thể này dựa theo ái lực của chúng, ái lực càng cao, càng nhiều bản sao

2.3 Trưởng thành ái lực: Biến đổi tất cả các bản sao với tỉ lệ nghịch với ái lực của chúng. Thêm các kháng thể biến đổi này vào P, tái chọn lọc cá thể tốt nhất và giữ chúng thành vùng nhớ của kháng nguyên này.

2.4 Siêu động lực: Thay kháng thể có ái lực nhỏ nhất bởi kháng thể được phát sinh ngẫu nhiên.



Bảng 3.1. So sánh giải thuật tiến hoá cơ bản và giải thuật chọn lọc nhân bản

Ở bảng trên, giải thuật chọn lọc nhân bản được so sánh với giải thuật tiến hóa cơ bản. Cần chú ý rằng, những điểm khác biệt nằm chủ yếu ở kĩ thuật được sử dụng. Trong các thuật toán chọn lọc nhân bản, cơ chế để chọn kháng thể phù hợp dựa trên ái lực với kháng nguyên. Theo đó, các cơ chế chọn lọc truyền thống được sử dụng trong tính toán tiến hóa, ví như chọn lọc cân bằng hoặc chọn lọc cạnh tranh sẽ dễ dàng áp dụng trong các giải thuật chọn lọc nhân bản. Trong thuật toán tiến hóa, xác suất để một cá thể được chọn được xác định dựa trên độ phù hợp của nó, và tương ứng với ái lực giữa kháng thể, kháng nguyên trong chọn lọc nhân bản.

Một vài phiên bản của các giải thuật chọn lọc nhân bản được đề ra bởi Castro và Von Zuben (2000) và de Castro (2003) và được sử dụng để nhận dạng mẫu và việc tối ưu hàm đa mô hình, được mô tả như sau:

**clonAlg():**

1. Chọn ngẫu nhiên một kháng nguyên Agj và trình diện nó với tất cả các kháng thể trong AB (repertoire)
2. Xác định vector fj chứa tất cả các ái lực giữa kháng nguyên Agj và toàn bộ kháng thể trong AB.
3. Chọn ra n kháng thể với ái lực cao nhất, tạo nên tập Abj{n} ái lực cao ứng với kháng nguyên Agj
4. Nhân bản n kháng thể được chọn một cách độc lập và cân xứng với ái lực kháng nguyên của chúng, tạo ra một tập các bản sao Cj các bản sao: ái lực càng cao, thì số bản sao được phát sinh của kháng thể đó càng nhiều.
5. Cj trải qua quá trình trưởng thành ái lực, biến đổi các kháng thể tỉ lệ nghịch với ái lực kháng nguyên của chúng để tạo ra quần thể các bản sao trưởng thành theo nguyên tắc ái lực càng cao, biến đổi càng ít
6. Xác định ái lực fj\* của các bản sao trưởng thành Cj\* ứng với kháng nguyên Agj
7. Tái chọn lọc cá thể có ái lực cao nhất (trong tập Cj\*) ứng với kháng nguyên Agj để trở thành phần tử của tập kháng thể ghi nhớ (Ab{m}). Nếu ái lực hướng kháng nguyên Agj của kháng thể này lớn hơn kháng thể ghi nhớ tương ứng Ab, Abj\* sẽ thay thế Ab.
8. Cuối cùng, thay những kháng nguyên có ái lực nhỏ nhất trong tập Abj{r} bởi những cá thể mới.

Có các phiên bản khác của giải thuật chọn lọc nhân bản, phiên bản áp dụng để nhận dạng mẫu đòi hỏi một tập các mẫu cần nhận dạng như đầu vào, trong khi các phiên bản cho việc tối ưu cần một hàm đối tượng để tối ưu.

## Ứng dụng

AIS có phạm vi ứng dụng rất rất rộng rãi trong nhiều lĩnh vực, đặc biệt là: nhận dạng mẫu, kiểm tra lỗi và những dấu hiệu bất thường của hệ thống, phân tích dữ liệu, lập lịch, học máy, điều khiển tự động, tìm kiếm và tối ưu, an toàn thông minh.

### Điều khiển

AIS có thể giải quyết vấn đề về tính không chắc chắn và các phép biến đổi. Do đó, AIS là một phương pháp mới để cải tiến chất lượng của các hệ thống điều khiển. Đặc biệt khi kết hợp với mạng nơron và giải thuật di truyền, có thể xây dựng hệ thống có khả năng lọc nhiễu, điều khiển các thiết bị rất phức tạp, ra quyết định trong điều khiển robot (quyết định của robot được coi như các kháng thể, còn tác động của môi trường là các kháng nguyên),…

### An ninh máy tính

Hai nội dung được quan tâm trong lĩnh vực này là chống các truy cập bất hợp pháp (ví dụ từ các hacker, người dùng không đủ thẩm quyền…) và các mã có hại (ví dụ như virus, worm, spyware…). Các nguyên tắc phát hiện và loại bỏ vi sinh vật lây nhiễm của hệ miễn dịch sinh học được áp dụng cho thiết kế hệ thống an ninh máy tính. Cách tiếp cận tự nhiên này cho phép xây dựng các hệ phát hiện và phòng chống đột nhập (Intrusion Detection System) có khả năng tự động phát hiện, cảnh báo sớm các truy cập bất hợp pháp. Phương pháp này đang được nghiên cứu và thực hành về an ninh mạng được xem là hướng phát triển có triển vọng nhất.

### Phát hiện lỗi

Lý thuyết mạng miễn dịch đã được sử dụng để xây dựng các cơ chế phát hiện lỗi. Một số ứng dụng chính bao gồm: Phát hiện và chẩn đoán lỗi của các bộ cảm biến, công cụ cắt gọt, máy công cụ, phần cứng, phần mềm…

### Phát hiện bất thường trong hệ thống

AIS có thể sử dụng để giải bài toán phát hiện bất thường trong các chuỗi thời gian, trong phân đoạn ảnh hay kiểm tra ảnh bằng kỹ thuật xây dựng tập bộ phát hiện (Detector set). Các mẫu chuẩn của hệ thống được coi là self, còn trong dữ liệu quan sát thì những thay đỗi bất thường vượt ngưỡng cho phép được coi là non-self.

### Tối ưu hóa

Các phương pháp của hệ miễn dịch được kết với giải thuật di truyền, mạng nơron để giải quyết nhiều bài toán tối ưu nhằm tăng cường khả năng hội tụ của thuật toán, duy trì tính đa dạng của quần thể, quản lý các ràng buộc.

### Khai phá dữ liệu

AIS là một phương pháp có hiệu quả có thể dùng cho nhận dạng dữ liệu. Một số áp dụng như: Phân lớp các DNA dùng trong sinh học, loại bỏ dữ liệu dư thừa, giám sát bệnh truyền nhiễm, phân tích dữ liệu y học…

Ngoài ra, AIS cũng được áp dụng trong một số lĩnh vực khác như: máy học, robot học, lập lịch, ghi nhớ kết hợp, sinh thái học, hệ sản xuất (production system), ngôi nhà thông minh, phỏng đoán cấu trúc protein, trung hoà nhiễu thích nghi (Adaptive noise neutralization).

Chương 4 TIẾP CẬN MÁY HỌC VÀ HỆ MIỄN DỊCH NHÂN TẠO TRONG PHÁT HIỆN VIRUS MÁY TÍNH

Trong chương này, giải pháp tiếp cận cho bài toán phát hiện virus máy tính được trình bày. Hướng tiếp cận được xây dựng dựa trên sự kết hợp hệ miễn dịch nhân tạo và mạng nơ-ron nhân tạo và được phân nhỏ thành các bài toán con, bao gồm các nội dung chính:

* Sử dụng giải thuật chọn lọc âm tính và thuật toán phân lớp K-Means để xây dựng dữ liệu huấn luyện.
* Từ dữ liệu huấn luyện này, giải thuật chọn lọc nhân bản được sử dụng để huấn luyện các mạng nơ-ron để xây dựng nên các bộ phát hiện.
* Sử dụng các bộ phát hiện để xác định mức độ nguy hiểm của một tập tin, từ đó xây dựng nên dữ liệu huấn luyện phân lớp tập tin.
* Bộ phân lớp được xây dựng dựa trên dữ liệu phân lớp tập tin, có khả năng xác định một tập tin nào đó là sạch hay virus.

## Ý tưởng

Bài toán phòng chống virus máy tính nói chung và bài toán phát hiện máy tính nói riêng là một bài toán rất khó, không một chương trình AV nào có thể bảo vệ máy tính một cách tuyệt đối cũng như không một cá nhân tổ chức nào có thể đảm bảo rằng mình sẽ không bị xâm phạm. Bởi vì, “không thuật toán nào có thể phát hiện tất cả các virus máy tính” [8] nên cũng sẽ không có một hướng tiếp cận nào hoàn hảo trong giải quyết bài toán. Các chương trình AV hiện nay đều sử dụng nhiều hướng tiếp cận khác nhau để bù trừ ưu nhược điểm giữa chúng.

Mặc dù các AV cũng như các hướng tiếp cận hiện tại đã hoàn thành rất tốt phần việc của mình nhưng trên thực tế, ta vẫn thấy những cuộc tấn công trên quy mô lớn gây thiệt hại hàng triệu thậm chí hàng tỉ đô-la. Không gì đảm bảo cho chúng ta, những người sử dụng máy tính, sẽ an toàn tuyệt đối dù máy tính của ta có trang bị chương trình AV hiện đại nhất.

Các hướng tiếp cận hiện tại ngày càng mất đi tính hiệu quả của mình do sự phát triển, biến đổi không ngừng của các kĩ thuật lập trình virus. Điều này đòi hỏi những hướng nghiên cứu, tiếp cận mới giải quyết bài toán phát hiện virus này phải được liên tục nghiên cứu và phát triển.

Qua quá trình tìm hiểu, nghiên cứu các hướng tiếp cận để giải quyết bài toán này, nhóm nghiên cứu chia bài toán phát hiện virus máy tính thành những bài toán nhỏ hơn và đưa ra các giải pháp để giải quyết từng bài toán nhỏ. Mỗi bài toán thể hiện một giai đoạn nghiên cứu hình thành nên giải pháp chung cho phát hiện virus máy tính, đồng thời mang tính nhất quán trong quá trình hình thành hướng tiếp cận từ việc xây dựng dữ liệu huấn luyện, học dữ liệu huấn luyện và phân lớp tập tin. Trong các bài toán này, bài toán học dữ liệu huấn luyện để xây dựng nên các bộ phát hiện (bài toán 2) là bài toán chính, được nhóm nghiên cứu đặt ra và giải quyết đầu tiên trong mô hình lý thuyết của hướng tiếp cận. Kế đó, khi bước vào quá trình thực nghiệm, các bài toán khác phát sinh và được giải quyết nhằm đảm bảo tính khả thi của hướng tiếp cận trong môi trường thực nghiệm. Ở đây, các bài toán cùng hướng giải quyết được trình bày theo sự lưu chuyển của dòng dữ liệu, từ việc xây dựng dữ liệu huấn luyện từ các tập tin, việc học các dữ liệu huấn luyện để tạo nên các bộ phát hiện đến việc sử dụng các bộ phát hiện này trong quá trình xây dựng nên bộ phân lớp tập tin.

## Bài toán Xây dựng dữ liệu huấn luyện

### Phát biểu bài toán

Cho một tập các tập tin thực thi được phân thành hai nhóm là virus và chương trình sạch. Từ tập hợp các chuỗi nhị phân trích xuất từ các tập tin này, ta tìm cách phân chia thành hai tập con: tập chứa các chuỗi nhị phân thường xuất hiện trong các tập tin virus (gọi tắt là tập chuỗi virus) và tập chứa các chuỗi nhị phân xuất hiện trong các chương trình sạch.

### Phân tích bài toán

Input:

* + Thư mục tập tin virus.
  + Thư mục tập tin thực thi sạch.

Output: Một cơ sở dữ liệu bao gồm các chuỗi nhị phân đã được gán nhãn là virus hoặc benign (training data TD)

Ta có thể thấy rằng, mục đích của bài toán 1 là xây dựng các bộ dữ liệu huấn luyện chuỗi nhị phân đã được phân vào hai nhóm.

### Hướng giải quyết

Ý tưởng giải quyết đơn thuần sẽ xây dựng mỗi tập con bao gồm các chuỗi nhị phân trích xuất từ nhóm tập tin tương ứng. Mặc dù nghe có vẻ rất hợp lý nhưng bộ dữ liệu huấn luyện sẽ chứa rất nhiều lỗi, do không có tồn tại sự phân biệt rạch ròi giữa hai tập con. Lý do đầu tiên là bởi vì bản chất của virus cũng chính là một chương trình, mặc khác, virus thường lậy nhiễm thông qua các vật chủ trung gian, nên tập tin virus cũng sẽ chứa phần lớn các chỉ thị máy tính như một chương trình bình thường.

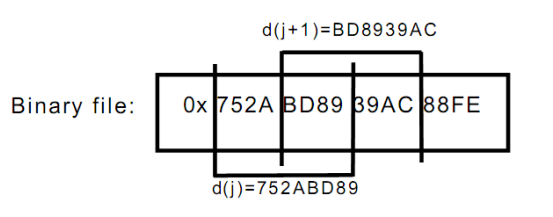
Luận điểm: Mặc dù bản chất virus cũng là một chương trình máy tính nhưng do kích thước tập tin virus thường phải rất nhỏ nên chúng thường chứa các chỉ dẫn thực thi ngắn mà ít xuất hiện trong các chương trình sạch. Mục tiêu của ta là hướng đến nhận dạng các đoạn mã này.

Giải pháp: Nhóm nghiên cứu khai thác ý tưởng được nêu ra ở trên kết hợp sử dụng một giải thuật nổi tiếng trong AIS là NSA để loại ra các chuỗi nhị phân thường gặp trong các tập tin sạch trong tập chuỗi virus.

**Giải thuật chọn lọc âm tính:**

Ở đây, NSA được sử dụng để loại trừ ra khỏi tập chuỗi virus các đoạn mã tồn tại trong cả các tập tin virus và tập tin chương trình sạch.

Đầu tiên, ta trích xuất các chuỗi nhị phân từ các tập tin virus và tập tin sạch tạo thành hai tập chuỗi tương ứng. Mỗi đoạn chuỗi nhị phân sẽ có chiều dài L và hai chuỗi liên tiếp nhau sẽ trùng lên nhau một đoạn L/2.



Hình 4.1. Cơ chế rút trích chuỗi nhị phân

Kế tiếp, ta loại ra những phần tử trong tập chuỗi virus mà đươc cho là khớp với bất kì phần tử nào trong tập chuỗi nhị phân sạch. Việc xác định trạng thái khớp hay không được dựa trên các luật so khớp chuỗi như Hamming [6], R-Contiguous [7]. Trong phạm vi đề tài, nhóm thực hiện tập trung chủ yếu vào luật so khớp R-contiguous.

Virus Files

Benign Files

Benign strings

Virus strings

Match

Discard

Output

Hình 4.2. Sơ đồ quy trình xây dựng dữ liệu huấn luyện

## Bài toán Xây dựng các bộ phân lớp

### Phát biểu bài toán

Cho một chuỗi nhị phân với chiều dài cố định L, tìm cách phân chuỗi này vào một trong hai tập con ở bài toán 1 (tập virus hoặc tập benign).

### Phân tích

Ban đầu, bài toán được đặt ra dưới dạng nhận dạng chuỗi. Giả sử các virus hay chương trình sạch đều biểu diễn dưới dạng các chuỗi nhị phân có kích thước bằng nhau, ta tìm cách phân loại một chuỗi nào đó là chuỗi virus hay chuỗi sạch.

Đây được xem là bài toán chính được đặt ra trong đề tài. Là vấn đề đầu tiên mà nhóm nghiên cứu hướng đến giải quyết rồi từ đó mở rộng sang hai bài toán còn lại.

Ở đây, ta dễ dàng nhận thấy rằng, bài toán 1 là quá trình xây dựng dữ liệu huấn luyện cho bài toán 2, trong khi bài toán 3 là giải pháp đề ra sử dụng kết quả của bài toán hai để ứng dụng vào môi trường thực tiễn.

Mục tiêu của bài toán 2 là xây dựng một bộ phân lớp có khả năng phân biệt giữa một chuỗi nhị phân virus và một chuỗi nhị phân sạch.

Giả sử sau khi giải quyết thành công bài toán 1, ta có được một cơ sở dữ liệu huấn luyện bao gồm các chuỗi nhị phân đã được gán nhãn là virus hoặc sạch (TD). Giải quyết bài toán 2 chính là ta xây dựng một hệ học giám sát, sử dụng TD (Dữ liệu huấn luyện) như đầu vào và đầu ra là một (hoặc nhiều) bộ phát hiện (detector).

* Input: Các bộ dữ liệu huấn luyện chuỗi nhị phân được gán nhãn.
* Output : Một (hoặc nhiều) detectors

### Hướng giải quyết

Trong quá trình nghiên cứu giải quyết bài toán này, nhóm thực hiện đã đưa ra nhiều ý tưởng giải quyết. Đặc điểm chung của các ý tưởng này đều sử dụng mạng nơ ron nhân tạo đóng vai trò là bộ phát hiện.

#### Ý tưởng sử dụng mạng nơ-ron với giải thuật học lan truyền ngược

Ý tưởng đầu tiên được đề ra chính là sử dụng mạng nơ-ron với giải thuật học lan truyền ngược. Bên cạnh những ưu điểm là một hướng tiếp cận đơn giản, ít thông số điều chỉnh… Ý tưởng này cũng vấp phải nhiều nhược điểm.

Nhược điểm:

* Kích thước mạng cố định, không thích hợp với các dữ liệu huấn luyện lớn.
* Quá trình huấn luyện rất lâu hoặc không thể hội tụ.
* Chỉ sử dụng 1 mạng sẽ gây ra khả năng nhận dạng lỗi.

#### Ý tưởng kết hợp mạng nơ-ron nhân tạo với thuật toán tiến hóa

Trong các ứng dụng sử dụng mạng nơ-ron, việc lựa chọn kích thước mạng rất quan trọng, ảnh hưởng trực tiếp đến khả năng học cũng như tốc độ học của nó. Nhưng việc cố định kích thước mạng trong quá trình huấn luyện đòi hỏi ta phải theo dõi quá trình huấn luyện, thực hiện lặp lại nhiều lần để thống kê và lựa chọn kích thước mạng phù hợp nhất. Mặc dù vậy, kích thước mạng có thể là phù hợp với dữ liệu huấn luyện này nhưng khi đổi dữ liệu huấn luyện khác, ta lại phải thực hiện toàn bộ lại các bước chọn kích thước mạng.

Ý tưởng đề ra là khai thác khả năng của thuật toán tiến hóa (mà cụ thể là giải thuật di truyền) để huấn luyện mạng. Thêm vào đó, cải tiến thuật toán tiến hóa nhằm xây dựng một hệ học có khả năng học cả trọng số và cấu trúc mạng, giảm mức độ tương tác với người dùng trong quá trình huấn luyện.

Ý tưởng này được đề ra trong [19].

Chúng ta cùng phân tích ưu nhược điểm của hướng tiếp cận này.

Ưu: Nhờ khả năng học cả trọng số và cấu trúc mạng, việc lựa chọn cấu trúc mạng do người dùng là không cần thiết. Khi dữ liệu huấn luyện thay đổi, hệ học tìm kiếm cấu trúc mạng mới để học tốt nhất.

Nhược: GA tìm kiếm 1 tối ưu toàn cục, tức là tìm kiếm một mạng nơ-ron tốt nhất trong toàn bộ quần thể, điều này dẫn đến một quá trịnh huấn luyện rất lâu.

Bên cạnh đó, chi phí để huấn luyện cấu trúc mạng cũng không hề nhỏ, độ phức tạp tỉ lệ thuận với số cấu trúc mạng được khởi tạo và tiến hóa.

Mặt khác, do đặc điểm của GA, kết thúc quá trình học là một mạng duy nhất mang tính thích nghi nhất trong toàn bộ quần thể. Sử dụng một mạng nơ-ron để thực hiện toàn bộ công việc phân biệt virus/ sạch dẫn đến tỉ lệ báo động giả đôi khi rất cao.

#### Ý tưởng kết hợp CLONALG và ANN.

Dựa trên ý tưởng về sự kết hợp giữa ANN và GA, nhóm thực hiện đề ra một ý tưởng mới khai thác ưu điểm của giải thuật chọn lọc nhân bản trong AIS và mạng nơ-ron nhân tạo. Bên cạnh đó, ý tưởng còn đưa vào một số cải tiến giải thuật CLONALG (Chọn lọc nhân bản) để phù hợp với cơ chế học giám sát, cũng như có khả năng tiến hóa cấu trúc mạng và trọng số mạng. Đây cũng là hướng tiếp cận mà nhóm thực hiện đề tài chọn như là một giải pháp cho bài toán 2.

##### Biểu diễn kháng thể / kháng nguyên

Trong phiên bản thông thường của CLONALG, các kháng thể và kháng nguyên đều được biểu diễn ở dạng các chuỗi. Tùy vào dạng ứng dụng mà các chuỗi này có thể là chuỗi chứa các giá trị nhị phân hoặc số thực. Trong hướng tiếp cận mà nhóm thực hiện trình bày trong luận văn, việc xây dựng cơ chế biểu diễn các chuỗi nhị phân virus và chuỗi nhị phân sạch thành các kháng thể, kháng nguyên trong AIS mang nhiều cải tiến cho phù hợp với ý định lựa chọn các mạng nơ ron đóng vai trò là các bộ phát hiện.

Ở đây, các chuỗi kháng nguyên được biểu diễn ở dạng các chuỗi nhị phân với chiều dài L. Đây chính là các chuỗi nhị phân ta có được sau khi giải quyết bài toán 1.

Một đặc điểm của giải thuật CLONALG là độ phức tạp của nó phụ thuộc rất nhiều và tỉ lệ thuận vào số lượng các kháng nguyên. Do vậy, để cải tiến tốc độ học của giải thuật CLONALG, thuật toán K-Means được áp dụng để gom cụm các chuỗi nhị phân trong tập chuỗi virus. Bằng cách này, các chuỗi nhị phân virus được xem là tương đồng nhau sẽ được gom vào cùng một nhóm. Mỗi nhóm đóng vai trò như một kháng nguyên.

Mỗi kháng thể là dạng mã hóa của một mạng nơ-ron nhân tạo, được đặc trưng bởi hai chuỗi giá trị. Một chuỗi chỉ ra cấu trúc của mạng mà kháng thể mã hóa, chuỗi còn lại là danh sách các trọng số của mạng nơ-ron tương ứng với cấu trúc của nó. Quá trình xây dựng quần thể các kháng thể nhớ chính là quá trình huấn luyện các mạng nơ-ron cả về cấu trúc và trọng số.

Như vậy, mỗi kháng thể không chỉ đơn thuần là các chuỗi nhị phân mà thông tin chứa trong các chuỗi này cho phép ta xây dựng nên một mạng nơ-ron với cấu trúc và trọng số tương ứng. Việc sử dụng mạng nơ-ron đóng vai trò như các bộ phát hiện cho phép ta sử dụng số lượng mạng ít hơn nhiều so với sử dụng các bộ phát hiện chuỗi nhị phân trong khi khả năng học không hề thua kém. Mặc khác, không gian bao quát của các bộ phát hiện mạng nơ-ron không mang hình thù cố định, nâng cao tính linh hoạt trong việc bao quát không gian dữ liệu. Số lượng bộ phát hiện tương ứng với số kháng thể trong quần thể nhớ, điều này sẽ dẫn đến việc sử dụng nhiều mạng nơ-ron làm các bộ phát hiện, khai thác khả năng hỗ trợ giữa các mạng với nhau.

##### Phép đo độ thích nghi

Sau khi đã biểu diễn được hai thành phần quan trọng trong AIS là kháng thể và kháng nguyên, bước tiếp theo ta sẽ phải xác định phép độ độ thích nghi giữa các kháng thể và kháng nguyên này.

Bởi vì mỗi kháng thể là dạng mã hóa của một mạng nơ-ron nên độ thích nghi được xác định dựa trên phép đo sai số đầu ra giữa mạng nơ ron và các giá trị nhãn của các chuỗi nhị phân trong kháng nguyên, ta gọi sai số này là error.

Đặt f là giá trị thích nghi của một kháng thể, ta có f = 1/error. Đôi khi f còn được gọi là ái lực kháng nguyên.

Trong việc phát hiện virus máy tính, bên cạnh việc xác định đúng một tập tin là virus hay không thì việc hạn chế báo động nhầm cũng cần được quan tâm. Bởi vì những hậu quả gây ra cho hệ thống do AV khằng định một tập tin hệ thống thành virus có thể còn lớn hơn nhiều so với việc bỏ sót 1 vài virus nào đó. Vì lý do này, nhóm nghiên cứu đưa thêm vào một thông số gọi là fpr (False Positive Rate - Tỉ lệ dương tính giả), thông số này chỉ ra tỉ lệ nhận dạng nhầm của một kháng thể được tính toán dựa trên tập chuỗi sạch trong TD.

Như vậy, giá trị f được tính bằng: f = (nghịch đảo trung bình cộng của error và fpr)

Cần lưu ý ở đây, error là giá trị lỗi trung bình trên toàn bộ chuỗi nhị phân của kháng nguyên đang được trình diện. Cả hai giá trị error và fpr đều được chuẩn hóa về (0, 1).

##### Giải thuật chọn lọc nhân bản CLONALG

Phần này trình bày chi tiết giải thuật CLONALG được sử dụng để huấn luyện các kháng thể trở thành kháng thể nhớ, từ đó xây dựng nên các bộ phát hiện có khả năng nhận biết các chuỗi virus cũng như bỏ qua các chuỗi sạch.

Các đối tượng chính: Bên cạnh các đối tượng thường thấy trong các phiên bản CLONALG như kháng nguyên, kháng thể…, nhóm nghiên cứu đưa thêm vào CLONALG các đối tượng mới nhằm xây dựng giải thuật phù hợp với định hướng nghiên cứu được đặt ra.

* Kháng thể: Mỗi kháng thể là một mạng nơ-ron được mã hóa ở dạng chuỗi cùng với một giá trị thích nghi f.
* Kháng nguyên: Mỗi kháng nguyên là một nhóm các chuỗi nhị phân virus tương đồng nhau.
* Quần thể các kháng thể nhớ memorySet: là tập hợp các kháng thể tốt nhất được lưu giữ và sẽ trở thành các bộ phát hiện sau khi kết thúc quá trình huấn luyện.
* Quần thể các kháng thể population: quần thể đầu tiên, nơi cung cấp các kháng thể cũng như các nguồn mới để tuyển chọn vào memorySet.
* Total Set: quần thể bao gồm tất cả các kháng thể từ quần thể memory Set và population.
* Quần thể các kháng thể bản sao clonalSet: Là tập hợp các kháng thể bản sao khi các kháng thể trong totalSet trải qua quá trình sinh trưởng proliferation.
* Quần thể các kháng thể trưởng thành maturedSet:Các kháng thể trong clonal Set sẽ trải qua bước trưởng thành ái lực, từ đó tạo nên quần thể matured Set
* Môi trường environment: Nơi diễn ra sự tương tác giữa các kháng thể với kháng nguyên và các tế bào cơ thể. Ở đây, môi trường bao gồm các nhóm dữ liệu huấn luyện đóng vai trò kháng nguyên và một tập các chuỗi nhị phân sạch đóng vai trò tế bào cơ thể.
* Kỷ nguyên epoch: Mỗi kỷ nguyên là một giai đoạn mà hệ thống chuyển từ trạng thái hiện tại sang trạng thái mới
* Hàm thích nghi fitnessFunction: Hàm thích nghi fitnessFunction cho phép ta tính toán giá trị f của kháng thể tương ứng với môi trường.

Sau khi đã định nghĩa các đối tượng trong giải thuật tương đối hoàn chỉnh, ta kế tiếp xây dựng các thao tác chính của thuật toán CLONALG, chúng bao gồm:

1) Tính ái lực kháng nguyên Antigenic Affinity Evaluation: Thao tác tính ái lực kháng nguyên của các kháng thể hay có thể nói ta tính giá trị thích nghi của kháng thể với môi trường. Việc tính toán giá trị f này sử dụng đối tượng fitnessFunction theo như quy tắc tính toán đã trình bày trong các phần trên.

2) Sự sinh trưởng Proliferation: chọn ra từ tập totalSet n kháng thể có giá trị thích nghi cao nhất, ta tiến hành nhân bản chúng một cách độc lập và tỉ lệ thuận với giá trị f. Giá trị này càng cao, số bản sao tương ứng với kháng thể được tạo ra càng lớn. Sau quá trình này ta có được tập bản sao clonalSet.

3) Sự trưởng thành ái lực Affinity maturation: Tất cả các kháng thể trong tập clonalSet đều sẽ phải trải qua quá trình mà ta gọi là trưởng thành ái lực. Thực chất của quá trình này là tiến hành các phép đột biến trên kháng thể, tỉ lệ và số lượng đột biến tỉ lệ nghịch với giá trị ái lực kháng nguyên f. Sau bước này ta có tập trưởng thành maturedSet.

4) Siêu biến đổi Metadynamics:

Bước siêu biến đổi metadynamics bao gồm 2 quá trình nhỏ hơn: cập nhật tập kháng thể nhớ memorySet và cập nhật tập kháng thể population.

Các kháng thể với ái lực kháng nguyên cao nhất trong tập maturedSet được chọn để làm ứng viên có thể trở thành kháng thể ghi nhớ để đưa vào tập memorySet. Nếu độ ái lực của một ứng viên này cao hơn một kháng thể đang được kích hoạt trong tập m thì nó sẽ được dùng để thay thế vị trí của kháng thể này.

Việc cập nhật tập kháng thể population được thực hiện bằng cách tái khởi tạo ngẫu nhiên một số kháng thể mới để thay thế cho các kháng thể có ái lực thấp nhất trong tập population.

Như đã nói, số lượng nhân bản của từng phần tử được chọn ra từ quần thể totalSet tỉ lệ thuận với ái lực kháng nguyên của chúng. Để xác định số lượng nhân bản cho mỗi phần tử, ta thực hiện sắp xếp các kháng thể theo thứ tự giảm dần ái lực đối với kháng nguyên. Ta duyệt xuôi dãy các kháng thể đã được sắp xếp, số lượng tập nhân bản của mỗi kháng nguyên được tính theo công thức sau:

numClones =[ +0.5]

Trong đó β là hệ số nhân bản, N là độ lớn của tập kháng thể, và i là chỉ số của kháng thể hiện thời, i.

Như vậy, tổng số các nhân bản được sinh ra từ n kháng thể được chọn sẽ là:

Nc=

Mỗi nhân bản sẽ có một độ đột biến α nhất định, xác định khả năng biến đổi của nhân bản. α được tính như sau:

α = [e (-p\*f)]

Trong đó, ρ là hệ số đột biến, f là ái lực của kháng thể gốc với kháng nguyên. β thường được chọn trong khoảng (0, 1] còn ρ nằm trong khoảng [1, 10].

Thuật giải:

NO

YES

Evaluation

Proliferation

Affinity maturation

Metadynamics

Stop

Present new antigen

Initialization

Hình 4.3. Sơ đồ giải thuật chọn lọc nhân bản

Bước 1:

Khởi tạo: Khởi tạo ngẫu nhiên một quần thể population với số lượng kháng thể được xác định trước.

Bước 2:

Trình diện từng kháng nguyên trong môi trường, thực hiện các thao tác:

* Antigenic Affinity Evaluation
* Proliferation
* Affinity Maturation
* Metadynamics

Bước 3: Khi giá trị f đạt đến ngưỡng xác định trước hoặc số kỷ nguyên đã đạt giới hạn, dừng thuật giải. Nếu không bắt đầu một kỷ nguyên mới, quay lại bước 2. Điều kiện dừng của thuật toán sẽ được trình bày chi tiết trong phần cài đặt thử nghiệm ở chương 5.

##### Artificial Neural Networks

Như đã trình bày trong phần giải thuật chọn lọc nhân bản CLONALG, mỗi kháng thể chính là một mạng nơ-ron được mã hóa ở dạng chuỗi bao gồm 1 chuỗi cấu trúc mạng và 1 chuỗi trọng số mạng. Phần này trình bày chi tiết cơ chế mã hóa một mạng nơ-ron dưới dạng chuỗi để xây dựng các kháng thể.

Các mạng nơ-ron nhân tạo là loại mạng dẫn tiến đa lớp với hàm tổ hợp tuyến tính và hàm kích hoạt là hàm lưỡng cực.

*f(x) = -1 + 2 / [1 + e-αx] với α =2*

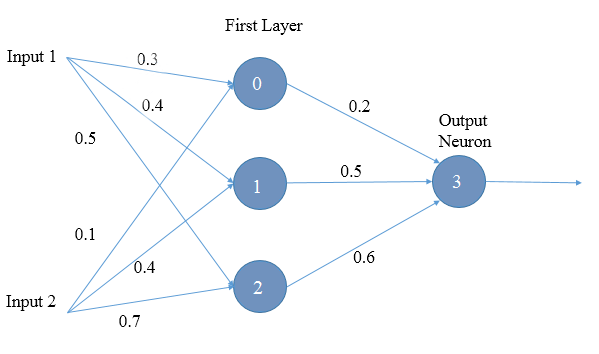
Hàm lưỡng cực cho ra giá trị output trong đoạn [-1, 1], thích hợp với bài toán ta đang cần giải quyết, phân loại một chuỗi nhị phân vào 1 trong 2 lớp virus hoặc sạch. Giá trị nhãn cho mỗi lớp được xác định là 0.5 tương ứng với virus và -0.5 tương ứng với sạch. Như vậy, khi cần xác định một chuỗi nhị phân có mang tính chất virus hay không, ta kiểm tra đầu ra của mạng khi truyền chuỗi này vào và tính toán độ lệch với đầu ra mong muốn là 0.5. Nếu độ lệch này nhỏ hơn hoặc bằng 0.5, ta kết luận chuỗi này mang tính chất virus, nếu độ lệch lớn hơn 0.5, ta kết luận chuỗi nhị phân là sạch.

Ngược lại, khi cần xác định một chuỗi có là sạch thì ta đưa vào giá trị đầu ra mong muốn là -0.5. Cả hai trường hợp này là ngược nhau nhưng mang ý nghĩa thông tin như nhau và thông thường, ta chỉ cần sử dụng một trường hợp với đầu ra mong muốn là 0.5 để xác định tính chất virus hay không của chuỗi cần kiểm tra.

Chuỗi biểu diễn trọng số của mạng nơ-ron được xây dựng như sau từ chuỗi trong số của mỗi nơ-ron có trong mạng. Mỗi nơ ron này có một tập các trọng số W: w0 w1…wm wm+1với wj (0≤j≤m) là trọng số của liên kết thứ j từ nơ ron lớp phía trước kết nối đến nơ ron lớp hiện tại, wm+1 là giá trị ngưỡng của nơ ron.Từ đó, chuỗi trọng số của mạng nơ ron được biểu diễn W0 W1 W2 … Wn với n là số nơ ron trong mạng. Cần lưu ý là số nơ-ron được xác định từ lớp ẩn đầu tiên của mạng nơ-ron, tức là các nơ-ron ở lớp đầu vào sẽ được bỏ qua.

Kế tiếp, ta định nghĩa một chuỗi cấu trúc mô tả cấu trúc của mạng nơ-ron bao gồm: Số nơ-ron mỗi lớp ẩn, số nơ ron ở lớp đầu vào và lớp xuất. Ở đây, kích thước chuỗi cấu trúc được cố định bằng 4, cho phép mạng có tối đa hai lớp ẩn. Giả sử một cấu trúc mạng được mô tả bởi một chuỗi gồm 4 phần tử A B C D, ta có:

* A chỉ ra số đơn vị đầu vào của mạng nơ-ron. Ở đây, A được xác định trước và bằng giá trị với L, tức là bằng với chiều dài của các chuỗi nhị phân.
* B, C chỉ ra số nơ-ron ở các lớp ẩn 1, 2 tương ứng. Cần lưu ý là C có thể mang giá trị 0, khi đó ta bỏ qua lớp ẩn thứ hai.
* D là số nơ-ron xuất, được cố định bằng 1.



Hình 4.4. Mạng nơ-ron nhân tạo

Structure string:

2 3 0 1

Chromosome string:

0.3 0.1 0.0 0.4 0.4 0.0 0.5 0.7 0.0 0.2 0.5 0.6 0.0

## Bài toán Xây dựng bộ phân lớp tập tin

### Phát biểu bài toán

Cho một tập tin thực thi trong hệ điều hành Windows, tìm cách phát hiện tính chất tập tin này là virus hay chương trình sạch.

Phân tích:

Như đã được đề cập trong phần phân tích của bài toán 2, bài toán 3 là vấn đề đặt ra khi sử dụng các bộ phát hiện trong môi trường thực tiễn, mà ở đây là việc xác định tính chất virus của một tập tin chứa mã thực thi (ta gọi tắt là tập tin).

### Cơ chế xác định mức độ nguy hiểm của một chuỗi nhị phân

* Input: Một chuỗi nhị phân l với chiều dài L
* Output: Giá trị thực biểu diễn mức độ nguy hiểm của l

Do các bộ phát hiện được xây dựng dựa trên việc nhận dạng chuỗi nên trước khi xác định một tập tin nào đó, ta cần xây dựng cơ chế xác định tính chất virus của các chuỗi nhị phân trích xuất từ tập tin này.

Ý tưởng đơn thuần sẽ cho rằng, nếu có ít nhất 1 trong số các bộ phát hiện đưa ra cảnh báo về chuỗi mã nào đó, chuỗi đó sẽ được gán là virus. Ý tưởng này nhìn qua có lẽ rất hợp lý do ta khai thác toàn bộ khả năng hỗ trợ giữa các bộ phát hiện, khi chuỗi mã thoát khỏi sự kiểm tra của bộ này thì sẽ bị bộ phát hiện khác bắt giữ. Điều này đảm bảo tỉ lệ nhận dạng chuỗi là virus rất cao, tuy nhiên, đồng thời tỉ lệ cảnh báo nhầm cũng cao không kém. Lý do là bởi vì, để kết luận một chuỗi là sạch đòi hỏi tất cả các bộ phát hiện đều không đưa ra cảnh báo. Để sử dụng ý tưởng này mà vẫn đảm bảo tỉ lệ cảnh báo nhầm thấp đòi hỏi quá trình huấn luyện phải rất lâu để đạt sao cho giá trị fpr trung bình của tất cả kháng thể trong quần thể nhớ phải càng thấp càng tốt, mặc dù vậy, đôi lúc ta không thể đạt được giá trị fpr lý tưởng do độ thích nghi của các kháng thể còn phụ thuộc vào giá trị error, một khi fpr quá thấp sẽ dẫn đến xu hướng gia tăng của giá trị error này.

Thay vì kết luận tính chất virus của một chuỗi dựa trên cảnh báo của bất kì bộ phát hiện nào, ta tính toán tỉ lệ bộ phát hiện đưa ra cảnh báo trên tổng số bộ phát hiện, và gọi giá trị này là giá trị xác định mức độ nguy hiểm của chuỗi. Như vậy, nếu quá trình huấn luyện cho ra các bộ phát hiện tốt thì ta có thể kết luận mức độ nguy hiểm của chuỗi nhị phân càng cao thì khả năng là chuỗi virus càng lớn và ngược lại, chuỗi sạch sẽ có mức độ nguy hiểm rất thấp.

dl(s) = (1)

Trong đó dl(s) là hàm giá trị mức độ nguy hiểm của chuỗi s, n là số bộ phát hiện, detect (i, s) xác định cảnh báo của bộ phát hiện thứ i đối với chuỗi s với 1 đưa ra cảnh báo, 0 không đưa ra cảnh báo.

### Cơ chế xác định mức độ nguy hiểm của một tập tin

Kế đến là cơ chế xác định mức độ nguy hiểm của một tập tin dựa trên kết quả của quá trình xác định mức độ nguy hiểm của các đoạn mã trích xuất từ nó. Điều này có thể thực hiện dễ dàng bằng cách tính giá trị mức độ nguy hiểm trung bình của tất cả các đoạn mã trích xuất từ tập tin.

dl(file) = (2)

Trong đó dl(file) là hàm tính mức độ nguy hiểm của tập tin file, m tổng số đoạn mã trích xuất từ tập tin, sj là chuỗi mã thứ j trong các đoạn mã được trích xuất.

Ta có thể kết hợp (1) và (2) trong một giải thuật như sau:

Giải thuật:

Đặt dataFragment là tập các đoạn mã trích xuất từ tập tin

detectorSet là tập các bộ phát hiện.

dl = 0; dl là giá trị mức độ nguy hiểm của tập tin.

**for** every string s in dataFragment do

count =0;

**for** every detector d in detectorSet do

count += detect(d,s);

**end for**

dl+= count/ |detectorSet|;

**end for**

dl/=|dataFragment|;

### Xây dựng bộ phân lớp xác định nhãn của một tập tin

Việc xác định thành công mức độ nguy hiểm của một tập tin chỉ là bước đầu trong việc xác định tập tin đó có là tập tin virus hay không.

Bởi vì một tập tin sạch sẽ có mức độ nguy hiểm thấp trong khi một tập tin virus thì ngược lại, mức độ nguy hiểm càng cao thể hiện tính chất độc hại của nó càng rõ. Dựa vào giá trị mức độ nguy hiểm này, ta có thể chọn một giá trị ngưỡng nào đó, sao cho nếu mức độ nguy hiểm của tập tin cao hơn giá trị ngưỡng này thì ta khẳng định đó là tập tin virus.

Tuy nhiên, việc xác định giá trị ngưỡng này không phải dễ dàng, ta cần thực nghiệm trên một số lượng tập tin lớn, bao gồm cả virus và chương trình sạch, từ đó lựa chọn giá trị ngưỡng phù hợp.

Nhóm thực hiện sử dụng một giải pháp hiệu quả hơn nhiều bằng việc xây dựng một bộ phân lớp, học từ dữ liệu thực nghiệm từ đó xác định nhãn (virus hoặc sạch) cho các tập tin mới.

#### 4.4.4.1. Xây dựng dữ liệu huấn luyện cho bộ phân lớp

Quá trình xây dựng dữ liệu huấn luyện cho bộ phân lớp được thực hiện bằng cách chọn ngẫu nhiên 1 tập tin nào đó trong tập các tập tin mà ta đã biết chính xác là virus hay chương trình sạch. Tính mức độ nguy hiểm của tập tin này đồng thường gán nhãn tương ứng cho nó. Mỗi bộ <dl, label> với dl là mức độ nguy hiểm của một tập tin và label là nhãn của tập tin đó, chính là một bộ dữ liệu huấn luyện.

Điều ta cần làm tiếp theo là xây dựng một tập các bộ huấn luyện như thế.

#### 4.4.4.2. Xây dựng bộ phân lớp

Bởi vì cấu trúc mỗi bộ dữ liệu huấn luyện khá đơn giản nên ta có thể sử dụng một mạng nơ-ron nhân tạo với giải thuật học lan truyền ngược để học các dữ liệu huấn luyện này.

Tương tự như các mạng nơ-ron có chức năng bộ phát hiện, bộ phân lớp mạng nơ ron này cũng có 1 nơ ron xuất và hàm kích hoạt là xích ma lưỡng cực. Giá trị nhãn được gán 0.5 cho virus và -0.5 cho chương trình sạch. Số đơn vị đầu vào là 1 và không có lớp ẩn.

Sau quá trình huấn luyện, để kiểm tra một tập tin thực thi có là virus hay không, ta đưa giá trị mức độ nguy hiểm của tập tin vào bộ phân lớp và tính giá trị xuất ra.

Ta sử dụng cơ chế xác định nhãn cho tập tin tương tự như cơ chế xác định nhãn cho chuỗi nhị phân đã được trình bày ở phần trước.

Chương 5 CÀI ĐẶT VÀ THỬ NGHIỆM

Trong chương này, nhóm trình bày phần cài đặt hiện thực hóa hướng tiếp cận đã được nêu ở chương trước, bên cạnh đó là trình bày một số kết quả thực nghiệm cũng như những phân tích trên kết quả này.

Chương trình được xây dựng bằng ngôn ngữ lập trình C# trên nền dotNet 4.0

## Cài đặt

### Hiện thực hóa hướng tiếp cận

Như đã trình bày ở chương trước, hướng tiếp cận được xây dựng từ giải pháp cho các bài toán nhỏ. Để hiện thực hóa hướng tiếp cận đã đề ra, nhóm xây dựng riêng thành từng giai đoạn tương ứng với mỗi giải pháp. Mỗi giai đoạn bao gồm các đối tượng và phương thức riêng, nhằm thực hiện một chức nhất định của mình. Giữa các giai đoạn có sự liên hệ nhất định về mặt thời gian, vì kết quả của giai đoạn này chính là dữ liệu đầu vào cho giai đoạn tiếp sau đó.

#### Giai đoạn Xây dựng dữ liệu huấn luyện

Giai đoạn Xây dựng dữ liệu huấn luyện được thiết kế nhằm hiện thực hóa giải pháp cho bài toán 1 đã được đặt ra trong chương 4.

Quá trình này bao gồm việc xây dựng các đối tượng và phương thức hỗ trợ cho việc đọc các tập tin, trích xuất thành các chuỗi nhị phân và tiến hành các thao tác tính toán như NSA hay K-Means:

Các đối tượng chính được xây dựng:

|  |  |
| --- | --- |
| Đối tượng | Mô tả |
| trainingData | Mổi đối tượng trainingData là một danh sách các đoạn mã nhị phân trích xuất từ các tập tin |
| Cluster | Mỗi đối tượng Cluster là một nhóm các đoạn mã được cho là tương tự nhau |
| Matching | Cung cấp các phương thức xác định mức độ tương đồng (hoặc sai khác) giữa các chuỗi nhị phân dựa trên các luật so khớp Hamming hoặc R –contiguous. |
| DataGeneration | Cung cấp các phương thức chính để đọc nguồn dữ liệu cũng như cài đặt thuật toán NSA. |
| KMeans | Cung cấp chức năng gom nhóm các đoạn mã nhị phân theo k nhóm. Đối tượng này được xây dựng trong bộ thư viễn hỗ trợ AForge.Net |

Bảng 5.1. Các đối tượng chính của giai đoạn xây dựng dữ liệu huấn luyện

Các phương thức chính được xây dựng:

|  |  |  |
| --- | --- | --- |
| Tên phương thức | Đối số | Mô tả |
| Match | Hai chuỗi nhị phân s1,s2 | Trả về true nếu s1 khớp s2, và ngược lại |
| Readfile | Đường dẫn đến tập tin | Đọc luồng dữ liệu từ tập tin theo từng byte |
| ReadDirectory | Đường dẫn thư mục | Đọc luồng dữ liệu từ các tập tin thực thi trong thư mục. |
| NegativeSelection |  | Thực hiện quá trình chọn lọc âm tính để xây dựng các đối tượng trainingData (virus, sạch) |

Bảng 5.2 Các phương thức chính của giai đoạn xây dựng dữ liệu huấn luyện

Kết quả của quá trình xây dựng dữ liệu huấn luyện là các bộ <string, label> với string là chuỗi nhị phân còn label là nhãn tương ứng với nó. Các bộ dữ liệu huấn luyện có label là virus sẽ được gom thành các nhóm khác nhau, mỗi nhóm mô tả một kháng nguyên. Còn các bộ dữ liệu huấn luyện sạch sẽ được gom vào cùng một nhóm, mô tả không gian các tế bào cơ thể.

#### 5.1.1.2. Giai đoạn Xây dựng các bộ phát hiện

Sau khi hoàn tất quá trình hiện thực hóa giai đoạn đầu tiên, giả sử ta đã có được dữ liệu huấn luyện cần thiết cho quá trình học, chúng ta tiến hành xây dựng một hệ học có khả năng học các dữ liệu huấn luyện này và cho ra các bộ phát hiện tốt.

Như đã đề cập trong phần giải pháp ở chương [], hai thành phần chính của giai đoạn này cần hiện thực trước tiên là giải thuật chọn lọc nhân bản CLONALG và mạng nơ-ron nhân tạo.

Mỗi mạng nơ-ron nhân tạo bao gồm một tập các lớp, mỗi lớp lại bao gồm một tập các nơ-ron cùng các phương thức tính toán ở mỗi lớp. Cuối cùng, mỗi nơ-ron, đơn vị cơ bản của mạng nơ-ron có thuộc tính chính là một chuỗi các trọng số từ các nơ-ron lớp phía trước kết nối đến nó.

Đối tượng mạng nơ-ron được xây dựng từ dưới lên, trước tiên là các nơ-ron với các thuộc tính và phương thức tính toán cho mỗi nơ-ron, tiếp theo là các lớp, rồi đến việc xây dựng đối tượng mạng.

Ngoài chuỗi trọng số, mỗi nơ-ron còn có một thuộc tính quan trọng là hàm kích hoạt được sử dụng thao tác xác định đầu ra cho mỗi nơ ron.

Sau khi đã cài đặt đối tượng mạng nơ-ron, ta tiến đến việc hiện thực hóa giải thuật chọn lọc nhân bản CLONALG.

Dưới đây là danh sách các đối tượng và phương thức chính được cài đặt nhằm mục đích hiện thức hóa CLONALG:

|  |  |
| --- | --- |
| Tên đối tượng | Mô tả |
| Kháng thể | Biểu diễn dạng mã hóa của một mạng nơ ron, bao gồm một mảng kiểu thực chứa danh sách trọng số của mạng và một mảng kiểu số nguyên chứa các thông số cấu trúc của mạng nơ ron. |
| Kháng nguyên | Là một mảng hai chiều, được xây dựng từ đối tượng Cluster. Mỗi phần tử là một chuỗi mã nhị phân. |
| Môi trường | Mô tả dữ liệu huấn luyện với các kháng nguyên và chuỗi sạch. |
| Hàm kích hoạt fitnessFunction | Cung cấp các phương thức tính toán giá trị thích nghi của kháng thể với mội trường. |

Bảng 5.3. Các đối tượng chính hiện thực hóa CLONALG

Các đối tượng memorySet, population, totalSet, clonalAndMaturedSet là danh sách các kháng thể.

|  |  |
| --- | --- |
| Tên phương thức | Mô tả |
| Initialisation | Khởi tạo các kháng thể trong tập population. Mỗi kháng thể được khởi tạo ngẫu nhiên cả về mảng trọng số và mảng cấu trúc. |
| Evaluation | Gọi đối tượng fitnessFunction để tính độ thích nghi của các kháng thể với kháng nguyên đang được trình diện và dữ liệu huấn luyện sạch. |
| CloneAndMutate | Thực hiện thao tác nhân bản và độ biến tùy theo thứ tự của kháng thể trong tập totalSet. |
| Metadynamics | Chọn lọc kháng thể vào quần thể nhớ. Khởi tạo các kháng thể mới thay thế các kháng thể có độ thích nghi thấp nhất trong tập population. |
| RunEpochs | Thực hiện một ký nguyên của CLONALG, trình diện lần lượt các kháng nguyên và thực hiện các thao tác tương ứng. |
| Learning | Thực hiện toàn bộ quá trình huấn luyện theo từng kỷ nguyên. Số kỷ nguyên tối đa do người dùng xác định và cho phép quá trình huấn luyện dừng lại theo yêu cầu. |
| GetNetworkDetectors | Từ quần thể nhớ các kháng thể, ta xây dựng tập hợp các mạng nơ ron tương ứng sau khi quá trình học kết thúc. |

Bảng 5.4. Các phương thức chính trong giai đoạn xây dựng các bộ phát hiện

Quá trình huấn luyện kết thúc khi số kỷ nguyên đạt tới một giới hạn nào đó hoặc do người sử dụng chủ động gọi lệnh dừng. Kết thúc quá trình này, ta được một tập các bộ phát hiện chính là các mạng nơ-ron được tạo ra từ thông tin mã hóa chứa trong các kháng thể của quần thể nhớ memorySet.

#### 5.1.1.3. Giai đoạn xây dựng bộ phân lớp

Trong giai đoạn này, ta hiện thực hóa quá trình tính toán giá trị mức độ nguy hiểm của chuỗi mã nhị phân dựa trên các bộ phát hiện, từ đó xác định giá trị mức độ nguy hiểm của tập tin. Kế tiếp, ta xây dựng dữ liệu huấn luyện cho bộ phân lớp bằng cách tính mức độ nguy hiểm của các tập tin virus và tập tin sạch cùng giá trị nhãn tương ứng.

Một bộ phân lớp là một mạng nơ-ron với các đặc trưng được mô tả trong phần hướng tiếp cận.

Dưới đây là danh sách các đối tượng và phương thức chính được cài đặt trong giai đoạn xây dựng bộ phân lớp:

|  |  |
| --- | --- |
| Tên | Mô tả |
| AffinityVector | Là đối tượng mộ tả thông tin về một tập tin với các thuộc tính chính bao gồm tên, giá trị nguy hiểm và nhãn xác định virus hay chương trình sạch |
| AffinityVectorGeneration | Cung cấp các phương thức trong việc đọc luồng dữ liệu cũng như xác định mức độ nguy hiểm của tập tin. |
| dangerLevelOfFragment | Phương thức hỗ trợ việc xác định mức độ nguy hiểm của đoạn mã nhị phân |
| dangerLevelOfaFile | Phương thức hỗ trợ việc tính mức độ nguy hiểm của một tập tin |
| BackPropagation | Đối tượng cung cấp các phương thức huấn luyện bộ phân lớp bằng giải thuật lan truyền ngược. |

Bảng 5.5. Các đối tượng và phương thức chính trong giai đoạn xây dựng các bộ phân lớp

Kết thúc giai đoạn này, ta được một bộ phân lớp là một mạng nơ ron có khả năng xác định nhãn của một tập tin dựa trên giá trị mức độ nguy hiểm của tập tin đó.

**Mô hình tổng thể hệ thống:**

Input files

Training data generation

Detectors generation

Classifier generation

Detection

Detected file/files

Results

Training Data

Detectors

Files

Files

Detectors + classifier

Hình 5.1. Mô hình tổng thể của hệ thống

### Mô-đun hóa các giai đoạn

Sau khi hướng tiếp cận đã được hiện thực hóa thành các giai đoạn, ta tiến hành bước mô-đun hóa các giai đoạn này.

Ở đây, hướng tiếp cận được mô hình hóa, mỗi giai đoạn được cài đặt trong chương trình dưới dạng mô đun, cho phép người sử dụng tương tác với hệ thống thông qua giao diện và lựa chọn thông số hệ thống và theo dõi, kiểm soát hoạt động của các quá trình .

Các mô đun chính bao gồm:

* Mô đun xây dựng dữ liệu huấn luyện (training Data Generation)
* Mô đun xây dựng các bộ phát hiện (detectors generation)
* Mô đun xây dựng bộ phân lớp (classifier generation)
* Mô đun kiểm tra tập tin, thư mục (detecting)

Trước khi các quá trình trong mỗi mô đun được thực hiện, các thông số cần thiết sẽ được cập nhật trong trường hợp có bất kì thay đổi nào từ phía người sử dụng. Trong hình vẽ mô phỏng mô hình tổng thể của hệ thống cũng như của các mô đun, hình elip biểu thị các trạng thái bắt đầu hay kết thúc, hình vuông biểu thị các quá trình con cũng như các thao tác tính toán cần thiết, còn hình bình hành biểu thị các đầu vào, đầu ra tương ứng cho mỗi quá trình, thao tác. Dòng dữ liệu được định hướng theo các dấu mũi tên.

#### 5.1.2.1. Xây dựng bộ tạo dữ liệu học

Input : Các tập tin virus và chương trình sạch được lưu trữ vào hai thư mục riêng biệt

Output: Dữ liệu huấn luyện dưới dạng các chuỗi nhị phân, được chia thành hai nhóm chính. Nhóm virus gọi là nhóm các kháng nguyên, mỗi kháng nguyên là một cụm các chuỗi mã nhị phân được cho là tương đồng nhau. Nhóm sạch bao gồm các chuỗi mã trích xuất từ các tập tin sạch.

##### Mô hình tổng quát của mô đun:

Strings

Virus strings

Virus files

Benign files

Extraction

Negative Selection

Clustering

Antigens

Benign strings

Hình 5.2. Mô hình tổng quát của mô-đun xây dựng dữ liệu huấn luyện

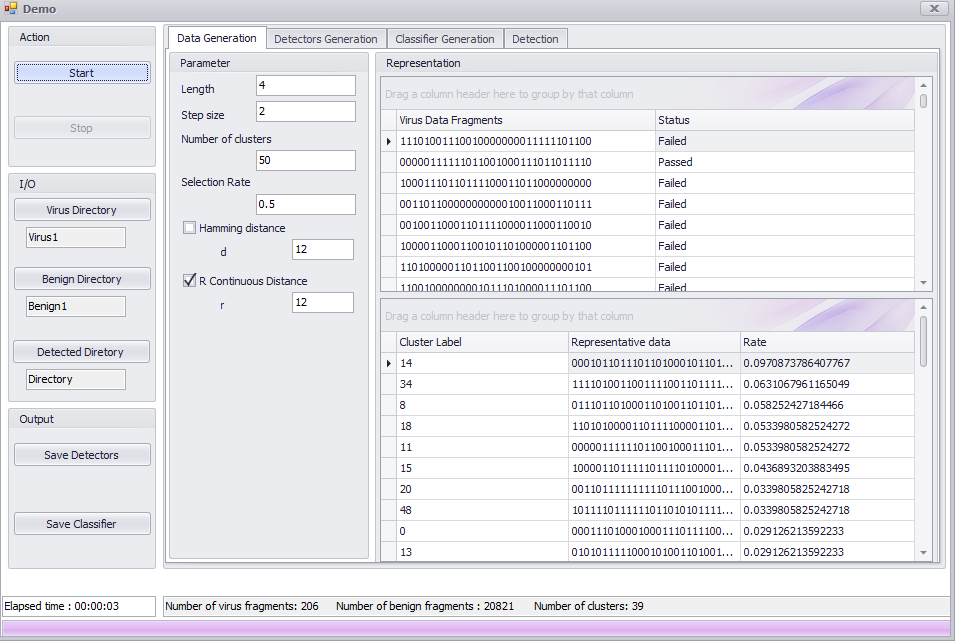
Mô đun xây dựng dữ liệu huấn luyện gồm có 3 thành phần chính: Đầu tiên là giai đoạn đọc luồng dữ liệu từ tập tin rồi phân tách thành các chuỗi mã nhị phân, ta gọi là Extraction hay quá trình trích xuất dữ liệu. Negative Selection là bước áp dụng giải thuật chọn lọc âm tính để loại các chuỗi mã thông thường mà tồn tại trong cả virus và chương trình sạch ra khỏi nhóm các chuỗi mã virus. Bước gom cụm clustering tiến hành thuật toán K-Means để gom các chuỗi nhị phân virus vào các nhóm mà ta gọi là kháng nguyên với số nhóm k do người dùng xác định.

##### Danh sách thông số:

|  |  |  |
| --- | --- | --- |
| Tên | Kiểu giá trị | Mô tả |
| Virus Directory  Benign Directory | Chuỗi kí tự | Đường dẫn đến thư mục chứa các tập tin virus và thư mục chương trình sạch. |
| Length | Kiểu số nguyên | Kích thước của chuỗi nhị phân tính bằng đơn vị byte. Giá trị mặc định là 4. |
| Step size | Kiểu số nguyên | Kích thước bước trích xuất chuỗi mã nhị phân từ luồng dữ liệu, cũng chính là số bytes chung giữa hai chuỗi nhị phân liên tiếp. Giá trị mặc định là bằng ½ chiều dài chuỗi nhị phân. |
| Number of clusters | Kiểu số nguyên | Thông số số nhóm k khi áp dụng thuật toán K-Means. Ở đây k chỉ mang ý nghĩa tối đa vì số nhóm thực tế có thể thấp hơn tùy vào trạng thái của dữ liệu. |
| Selection Rate | Kiểu Thực | Tỉ lệ xác định các nhóm có mật độ phần tử đông nhất được chọn để trở thành các kháng nguyên sử dụng trong quá trình huấn luyện. |
| d | Kiểu số nguyên | Giá trị ngưỡng sử dụng trong luật so khớp Hamming. |
| r | Kiểu số nguyên | Giá trị ngưỡng sử dụng trong luật so khớp R-Contiguous |

Bảng 5.6. Danh sách thông số của mô-đun xây dựng dữ liệu huấn luyện

##### Giao diện thông số và kết quả



Hình 5.3. Giao diện mô-đun xây dựng dữ liệu huấn luyện

#### 5.1.2.2. Xây dựng mô-đun phát sinh các bộ phát hiện

Input: Dữ liệu huấn luyện (Các kháng nguyên và tập chuỗi nhị phân sạch)

Output: Các bộ phát hiện tốt

##### Mô hình tổng quát của mô đun:

Training Data

CLONALG

Decoding

Detectors

Antibodies

Hình 5.4. Mô hình tổng quát của mô đun xây dựng các bộ phát hiện

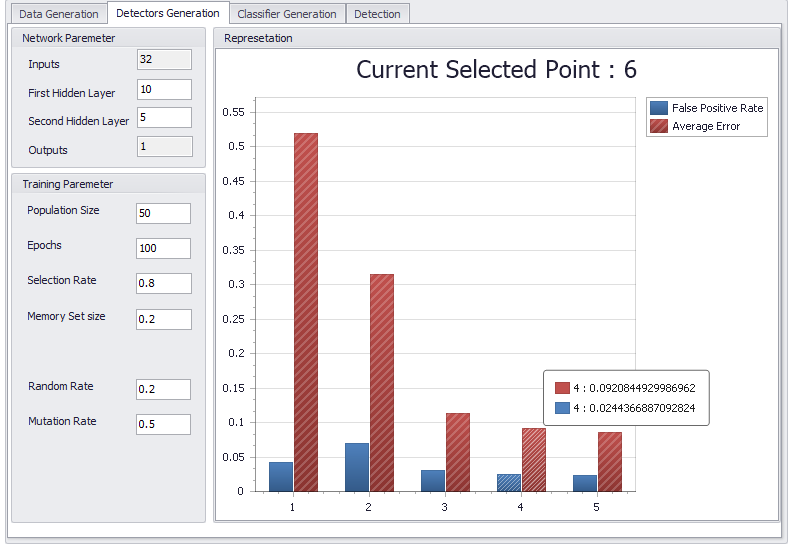
Mô đun xây dựng bộ phát hiện gồm hai bộ phận chính, trong đó CLONALG đóng vai trò là người huấn luyện còn Decoding có chức năng giải mã thông tin từ các kháng thể nhớ thành các mạng nơ ron tương ứng.

##### Danh sách các thông số:

|  |  |  |
| --- | --- | --- |
| Tên | Kiểu giá trị | Mô tả |
| Network Parameter | Kiểu nguyên | Mô tả rang buộc cấu trúc của các mạng nơ ron mà thông tin được mã hóa trong các kháng thể. Trong đó có số đơn vị đầu vào bằng với kích thước các chuỗi nhị phân tính bằng đơn vị bit, số nơ ron xuất là 1 và hai thông số còn lại quy định số nơ ron tối đa ở hai lớp ẩn. |
| Population size | Kiểu nguyên | Kích thước quần thể kháng thể population |
| Memory Set size | Kiểu thực | Kích thước quần thể nhớ theo tỉ lệ với quần thể population. |
| Epochs | Kiểu nguyên | Số kỷ nguyên tối đa trong quá trình huấn luyện |
| Selection Rate | Kiểu thực | Tỉ lệ chọn kháng thể để nhân bản và độ biến |
| Random Rate | Kiểu thực | Tỉ lệ kháng thể được tạo mới trong mỗi kỷ nguyên. |
| Mutation Rate | Kiểu thực | Giá trị kiểm soát xác suất một kháng thể xảy ra độ biến. |

Bảng 5.7. Danh sách các thông số của mô đun xây dựng các bộ phát hiện

##### Giao diện thông số và mô tả kết quả



Hình 5.5. Giao diện mô đun xây dựng các bộ phát hiện

#### 5.1.2.3. Xây dựng bộ phân lớp

Input:

* + Thư mục các tập tin virus và chương trình sạch
  + Các bộ phát hiện

Output: Một bộ phân lớp có khả năng xác định nhãn của một tập tin dựa trên giá trị mức độ nguy hiểm của nó.

##### Mô hình tổng quát của mô đun:

Detectors

Input files

Affinity Vector Generation

Classifier Training

Classifier

Hình 5.6. Mô hình tổng quát của mô-đun xây dựng bộ phân lớp

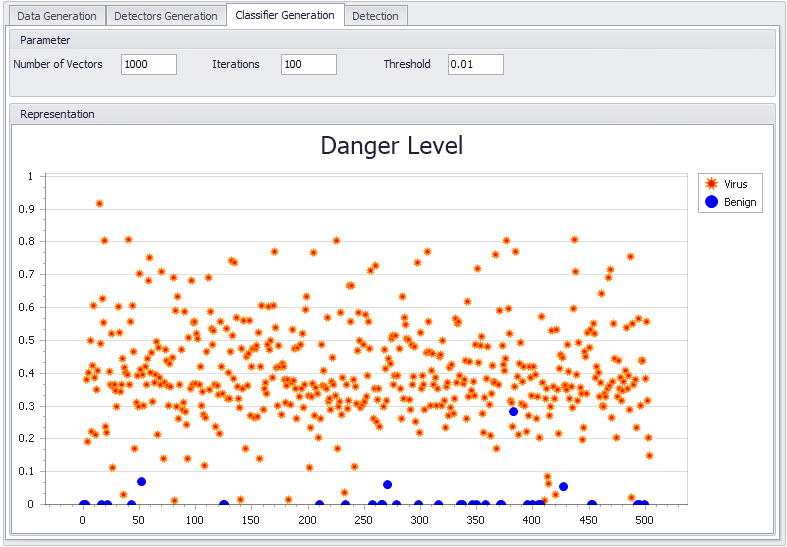
Hai thành phần chính của mô đun xây dựng bộ phát lớp bao gồm bộ xác định mức độ nguy hiểm của tập tin (hay xác định vec-tơ ái lực của tập tin) và bộ huấn luyện bộ phân lớp classifier.

##### Danh sách thông số

|  |  |  |
| --- | --- | --- |
| Tên | Kiểu giá trị | Mô tả |
| Number of Vectors | Kiểu nguyên | Số vec-tơ ái lực sẽ được tạo ra làm dữ liệu cho việc huấn luyện bộ phân lớp. |
| Iterations | Kiểu nguyên | Số vòng lặp tối đa khi huấn luyện bộ phân lớp bằng giải thuật lan truyền ngược. |
| Threshold | Kiểu thực | Ngưỡng sai số để dừng quá trình huấn luyện. |

Bảng 5.8. Danh sách thông số của mô-đun xây dựng bộ phân lớp

##### Giao diện thông số và kết quả



Hình 5.7. Giao diện mô-đun xây dựng bộ phân lớp

#### 5.1.2.4. Xây dựng bộ kiểm tra tập tin

Input: Tập tin hay thư mục cần kiểm tra

Output: Nhãn xác định tập tin (hay các tập tin trong thư mục) là virus hay chương trình sạch.

##### Mô hình tổng quát của mô đun:

Detectors

Input file/files

Affinity Vector Generation

Detection

Results

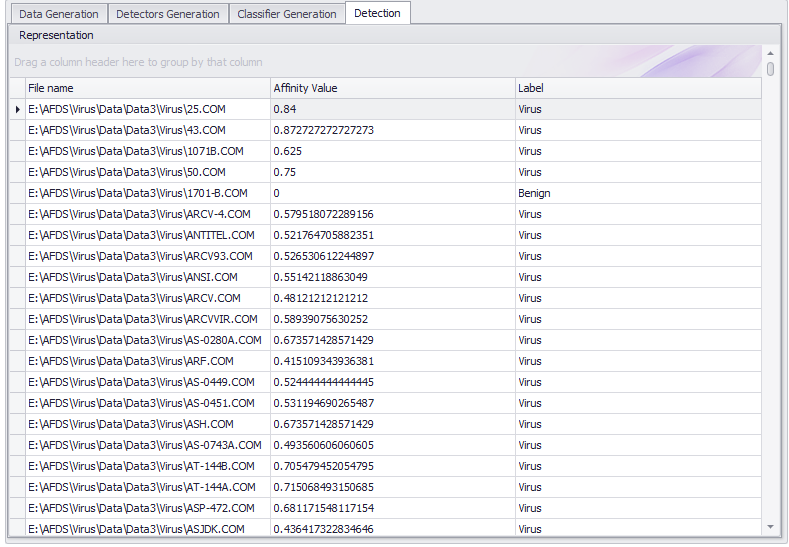
Classifier

Hình 5.8. Mô hình tổng quát mô-đun kiểm tra tập tin

Ở đây, quá trình kiểm tra một tập tin gồm giai đoạn xác định mức độ nguy hiểm của tập tin, tương tự như cách mà ta phát sinh ra các vec-tơ ái lực làm dữ liệu cho quá trình huấn luyện bộ phát hiện. Sau khi đã có được giá trị mức độ nguy hiểm của tập tin, ta sử dụng bộ phân lớp để xác định nhãn của tập tin dựa trên mức độ nguy hiểm của nó.

Thông số chính của mô đun là đường dẫn đến thư mục chứa các tập tin cần kiểm tra.

##### Giao diện thông số và kết quả



Hình 5.9. Giao diện mô đun kiểm tra tập tin

## Kết quả thử nghiệm

### Dữ liệu đầu vào

Đặc trưng của đề tài hướng đến việc xây dựng hướng tiếp cận giải quyết bài toán phát hiện virus máy tính, mà đặc biệt là các virus thực thi trong hệ điều hành Windows, do đó, việc thao tác trên các tập tin virus là điều không thể tránh khỏi.

Do chính sách bảo mật từ các trang web nghiên cứu virus, để tránh việc người sử dụng lạm dụng các virus này gây hại cho cộng đồng, nhiều bước đăng kí và kiểm soát chặt chẽ được thiết lập, việc thu thập các tập tin virus này cũng gặp rất nhiều khó khăn. Tuy nhiên, số lượng các nhà nghiên cứu cũng như các bạn trẻ đam mê về bảo mật khá lớn, nhiều người trong số đó xây dựng những kho lưu trữ tập tin virus riêng cho mình và sẵn sàng chia sẻ cho mọi người.

Nguồn tập tin virus mà nhóm nghiên cứu thu thập được bao gồm 5923 virus với nhiều loại và thời gian xuất hiện khác nhau, được thu thập từ các trang web liên quan đến bảo mật nổi tiếng của Việt Nam. Đây quả là một số lượng rất lớn cho việc nghiên cứu. Từ đây, nhóm tiến hành phân loại cũng như xây dựng các bộ dữ liệu riêng, phù hợp với chương trình được cài đặt và quá trình thử nghiệm của mình. Dưới đây là bảng mô tả các bộ dữ liệu này với các tập tin chương trình sạch được trích xuất trực tiếp từ máy tính của nhóm thực hiện.

|  |  |  |
| --- | --- | --- |
| Tên | Số tập tin virus | Số tập tin sạch |
| Data 1 | 100 | 50 |
| Data 2 | 200 | 50 |
| Data 3 | 300 | 50 |
| Data 4 | 400 | 50 |
| Data 5 | 500 | 50 |

Bảng 5.9. Dữ liệu huấn luyện và kiểm tra

Ở đây, số lượng tập tin virus trong mỗi bộ tập tin huấn luyện được chọn với số lượng khác nhau. Bên cạnh đó, cơ chế chọn ra các virus cho mỗi bộ dữ liệu tập tin huấn luyện này từ bộ dữ liệu ban đầu cũng khác nhau, ví dụ như Data 4 bao gồm các virus được chọn theo giá trị Date modified của tập tin thì Data 5 được chọn theo giá trị kích thước của tập tin, Data 3 thì được chọn theo tên…. Mỗi bộ dữ liệu có một tập các tập tin sạch khác nhau và cùng một số lượng là 50.

### Xác định thông số

#### Lựa chọn ngưỡng so khớp trong giai đoạn xây dựng dữ liệu huấn luyện

Như đã trình bày ở chương trước, ta sử dụng cơ chế so khớp trùng chuỗi R-contiguous để xác định một chuỗi nào đó trong tập chuỗi virus sẽ bị loại ra hay được giữ lại. Ở đây, r là thông số quan trọng nhất ảnh hưởng trực tiếp đến kết quả của giai đoạn này như số lượng chuỗi dùng làm dữ liệu huấn luyện, mức độ tách biệt giữa các chuỗi thuộc các phân lớp khác nhau, thời gian xây dựng dữ liệu huấn luyện,… từ đó ảnh hưởng đến kết quả của các giai đoạn theo sau.

Thử nghiệm được tiến hành trên hai bộ tập tin đầu vào là Data 1, Data 2 và Data 3 ta cho thông số r nhận lần lượt các giá trị từ thấp đến cao.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên bộ dữ liệu | Giá trị r | Thời gian | Số chuỗi virus | Số chuỗi sạch |
| Data 1 | 8 |  | 0 | 74349 |
|  | 12 | 0:00:11 | 4 |  |
|  | 16 | 00:03:47 | 5261 |  |
| Data 2 | 8 |  | 0 | 76918 |
|  | 12 | 00:01:00 | 40 |  |
|  | 16 | 0:17:39 | 22008 |  |
| Data 3 | 8 | 0:00:11 | 0 | 150833 |
|  | 12 | 0:06:11 | 186 |  |
|  | 16 | 1:30:36 | 108810 |  |

Bảng 5.10. Thử nghiệm sự tương quan giữa giá trị r và quá trinh xây dựng dữ liệu huấn luyện trên Data 1, Data 2, Data 3.

Từ bảng trên, ta có thể dễ dàng nhận thấy rằng, khi giá trị r gia tăng, thời gian thực thi tăng theo, đồng thời số lượng chuỗi trong tập chuỗi virus cũng tăng lên rất nhiều.

Ta giải thích cho điều này như sau: khi giá trị r nhỏ, ngưỡng so khớp nhỏ từ đó một chuỗi trong tập virus dễ dàng được gán là khớp với một chuỗi trong tập chuỗi sạch. Khi r =8, ta thấy toàn bộ chuỗi nhị phân trong tập chuỗi virus đều bị loại ra, chứng tỏ giá trị r này quá khắt khe. Ngược lại, khi r tăng dần, độ khắt khe để giữ lại một chuỗi nào đó trong tập chuỗi virus cũng giảm xuống, nhiều chuỗi nhị phân được giữ lại hơn. Tuy nhiên, khi r quá lớn sẽ dẫn tới số lượng chuỗi nhị phân này gia tăng đột biến và mất đi sự tách biệt giữa một chuỗi thuộc tập chuỗi virus với một chuỗi thuộc tập chuỗi sạch. (khi r=16, tồn tại hai chuỗi giống nhau liên tiếp 15 bit nhưng lại thuộc về hai tập phân loại khác nhau.)

Điều quan trọng là tùy thuộc vào bộ dữ liệu tập tin đầu vào, ta tiến hành thực hiện thao tác xây dựng dữ liệu huấn luyện này nhiều lần để lựa chọn giá trị r sao cho số lượng chuỗi nhị phân trong tập chuỗi virus thích hợp nhất.

#### Lựa chọn kích thước quần thể:

Ở giai đoạn huấn luyện các mạng nơ-ron để phát sinh các bộ phát hiện, chúng ta có rất nhiều thông số cần được quan tâm như số nơ ron tối đa có thể có ở các lớp ẩn, các thông số huấn luyện như số kỷ nguyên, tỉ lệ đột biến, tỉ lệ chọn lọc, tỉ lệ phát sinh,….Ở đây, ta tập trung vào thông số kích thước quần thể population và thông số kích thước quần thể nhớ memorySet, đây là những thông số rất quan trọng, không chỉ ảnh hưởng đến thời gian huấn luyện mà còn cả độ chính xác của các bộ phát hiện.

Ở đây, việc lựa chọn giá trị cho hai thông số kích thước quần thể và kích thước quần thể nhớ sẽ ảnh hưởng trực tiếp đến số lượng các bộ phát hiện được xây dựng. Số lượng các bộ phát hiện có ảnh hưởng chủ yếu đến kích thước mỗi mạng nơ-ron và chi phí huấn luyện. Tức là, số lượng các bộ phát hiện càng lớn thì cấu trúc mạng nơ-ron càng bớt phức tạp, tỉ lệ chồng lấn giữa các bộ phát hiện càng lớn, chi phí huấn luyện càng lớn. Trong khi đó, khi số lượng detectors nhỏ, cấu trúc mạng sẽ khá phức tạp (số lớp ẩn và số nơ-ron lớp ẩn) do giải thuật huấn luyện buộc phải trích xuất những đặc trưng nhất để cập nhật cấu trúc và trọng số của mạng nơ-ron. Ở đây ta chọn số lượng các cá thể nhớ từ 10 % đến 20 % số lượng cá thể trong quần thể. Tốt nhất là ta cần lựa chọn các thông số kích thước quần thể population và kích thước quần thể nhớ memorySet sao cho số lượng các bộ phát hiện gần với số lượng các kháng nguyên mà ta cần huấn luyện.

Trong thử nghiệm của mình, nhóm chọn giá trị mặc định cho thông số kích thước quần thể là 100 và kích thước quần thể nhớ là 0.2 (20 % của giá trị kích thước quần thể)

#### Lựa chọn số nơ-ron lớp ẩn:

Trong quá trình thực hiện của mình, quá trình xây dựng các bộ phát hiện sẽ xây dựng các mạng nơ-ron tốt cả về trọng số lẫn cấu trúc mạng. Ở đây, ta cố định số lớp ẩn là 2 song lại cho phép linh hoạt số nơ-ron của mỗi lớp ẩn này sao cho dưới giá trị tối đa mà ta quy định.

Việc lựa chọn giới hạn trên cho số lượng nơ-ron của mỗi lớp ẩn này cũng rất quan trọng, ảnh hưởng trực tiếp đến độ phức tạp của cấu trúc mạng, từ đó ảnh hưởng đến thời gian và độ tốt của quá trình huấn luyện.

Việc chọn giá trị quá cao sẽ gây lãng phí tài nguyên nhưng quá thấp sẽ khiến hệ thống khó học được chức năng mong muốn do không đủ tài nguyên cung cấp cho nó. Trong thực nghiệm của mình, nhóm lựa chọn giới hạn cho lớp ẩn đầu tiên là 16 và lớp ẩn thứ hai là 8.

#### Lựa chọn điều kiện dừng:

Hai thông số chính quyết định đến điều kiện dừng của quá trình huấn luyện (ở giai đoạn xây dựng các bộ phát hiện) là số kỉ nguyên và tỉ lệ lỗi.

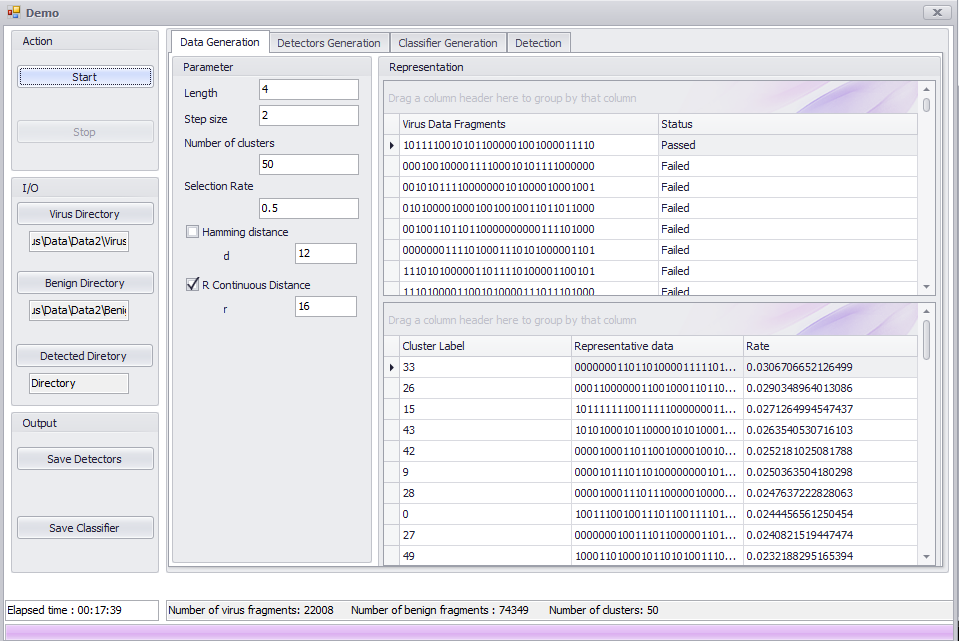
Số kỉ nguyên chính là số vòng lặp mà sau mỗi kỷ nguyên, hệ thống biến đổi từ trạng thái hiện tại sang trạng thái mới. Số kỷ nguyên tối đa được lựa chọn trong thực nghiệm là 100, tức là quá trình huấn luyện sẽ bị bắt buộc dừng lại sau 100 vòng lặp.

Tỉ lệ lỗi là một giá trị thường được sử dụng để làm điều kiện dừng trong các hệ thống học. Ở đây, mục đích của ta không phải tìm ra một mạng nơ-ron tốt nhất mà là một tập hợp các mạng nơ-ron tốt từ các kháng thể trong quần thể nhớ, do đó tỉ lệ lỗi được tính là giá trị lỗi trung bình của tất cả các kháng thể trong quần thể nhớ. Tuy nhiên, chúng ta cũng cần lưu ý rằng, giá trị này không phản ánh chính xác mức độ tốt của cả quần thể nhớ mà chỉ là giá trị được tính toán nhằm thuận lợi cho việc xác định điều kiện dừng. Bởi vì ta sử dụng giải thuật chọn lọc nhân bản để chọn ra một tập hợp các mạng nơ ron, ta không hướng đến việc mỗi mạng nơ ron trong tập hợp này sẽ phải học được tất cả các mẫu (các kháng nguyên) được đưa ra mà chỉ tập trung vào những mẫu nào mà nó có khả năng học tốt nhất.

Thông thường, chúng ta thường lựa chọn một giá trị ngưỡng để dừng quá trình huấn luyện khi tỉ lệ lỗi đạt tới ngưỡng này, tuy nhiên, việc xác định giá trị ngưỡng này không hề dễ dàng do còn phụ thuộc vào dữ liệu đầu vào và có thể có trường hợp hệ thống hội tụ đến một giá trị mà tỉ lệ lỗi không bao giờ đạt tới ngưỡng dừng. Trong thử nghiệm của mình, nhóm không xác định cụ thể một giá trị ngưỡng lỗi để dừng quá trình huấn luyện mà thay vào đó, độ tốt huấn luyện của mỗi ký nguyên sẽ được thể hiện trên một biểu đồ cột. Việc chúng ta cần làm là quan sát biểu đồ cột này và lựa chọn thời điểm mà ta cho là hệ thống đạt được trạng thái tốt nhất.

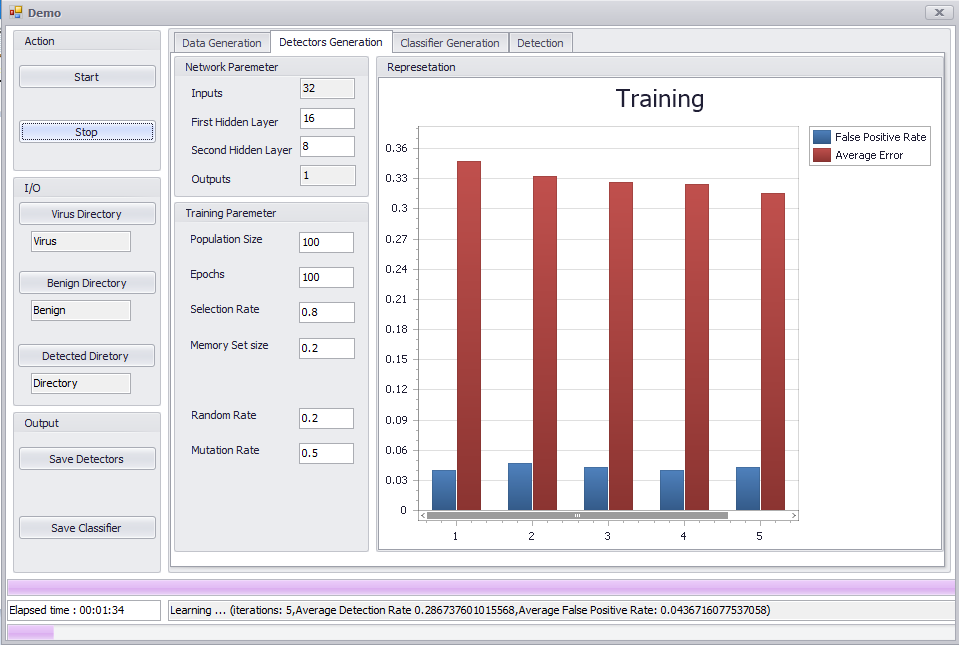
### Trình bày kết quả

#### Kết quả huấn luyện



Hình 5.10. Kết quả thử nghiệm giai đoạn xây dựng dữ liệu huấn luyện

Từ hình 5.10, ta thấy số lượng chuỗi sạch rất lớn (> 50000) và số lượng chuỗi trong tập virus cũng không hề nhỏ (>20000). Bên cạnh đó, ta dễ dàng biết được nếu một chuỗi nào đó trong tập chuỗi virus bị loại ra hay được giữ lại. Mặt khác, số lượng các kháng nguyên cũng như tỉ lệ số lượng các chuỗi nhị phân của nó cũng được trình bày ở bảng phía dưới.



Hình 5.11. Kết quả thử nghiệm giai đoạn xây dựng các bộ phát hiện

Từ hình 5.11, ta quan sát được từng kỷ nguyên của quá trình huấn luyện, từ thông tin số kháng nguyên hiện tại đang được trình diện, giá trị lỗi đạt tới sau mỗi kỷ nguyên cũng như thời gian thực hiện của toàn bộ quá trình.



Hình 5.12. Kết quả thử nghiệm giai đoạn xây dựng bộ phân lớp

Từ hình 5.12, ta thấy có sự tách biệt rõ ràng giữa giá trị mức độ nguy hiểm của một tập tin virus và 1 tập tin sạch. Mặc dù vậy, do quá trình huấn luyện trước đó chọn điểm dừng có giá trị ngưỡng lỗi khoảng 0.169 và tỉ lệ cảnh báo nhầm >0.1 nên dẫn đến kết quả của giai đoạn xây dựng bộ phân lớp chưa thật sự tốt, ta thấy vẫn còn những phần tử không nằm ở vị trí mà ta mong muốn.

#### Kết quả kiểm tra

##### Thử nghiệm 1

|  |  |
| --- | --- |
| Dữ liệu huấn luyện | Dữ liệu kiểm tra |
| Data 1 | Data 1, Data 2, Data 3, Data 4, Data 5 |

Bảng 5.11. Dữ liệu huấn luyện và kiểm tra trong thử nghiệm 1

|  |  |  |  |
| --- | --- | --- | --- |
| Virus | Phát hiện | Tổng số | Tỉ lệ |
| Data 1 | 88 | 100 | 0.88 |
| Data 2 | 198 | 200 | 0.99 |
| Data 3 | 273 | 300 | 0.91 |
| Data 4 | 379 | 400 | 0.9475 |
| Data 5 | 482 | 500 | 0.964 |

Bảng 5.12. Kết quả kiểm tra trên dữ liệu virus trong thử nghiệm 1

|  |  |  |  |
| --- | --- | --- | --- |
| Sạch | Phát hiện | Tổng số | Tỉ lệ |
| Data 1 | 10 | 50 | 0.2 |
| Data 2 | 11 | 50 | 0.22 |
| Data 3 | 16 | 50 | 0.32 |
| Data 4 | 13 | 50 | 0.26 |
| Data 5 | 8 | 50 | 0.16 |

Bảng 5.13. Kết quả kiểm tra trên dữ liệu sạch trong thử nghiệm 1

Ở đây, chúng ta sử dụng các tập tin trong bộ dữ liệu Data 1 để huấn luyện và tiến hành kiểm tra trên các bộ dữ liệu Data 1, 2, 3,4,5.

Khi tiến hành kiểm tra trên các bộ dữ liệu, kết quả nhận được khá tốt khi tỉ lệ lỗi dưới 30 % và tỉ lệ phát hiện khá cao (> 85 %). Mặc dù vậy, tỉ lệ cảnh báo nhầm trong trường hợp kiểm tra với bộ dữ liệu Data 3 khá cao, lên tới 32 %.

Từ đây, ta thấy rằng khả năng phát hiện virus của hướng tiếp cận đề ra khá tốt trong khi cần phải có nhiều cải tiến hơn nữa để giảm tỉ lệ cảnh báo nhầm.

##### Thử nghiệm 2

|  |  |
| --- | --- |
| Dữ liệu huấn luyện | Dữ liệu kiểm tra |
| Data 2 | Data 1, Data 2, Data 3, Data 4, Data 5 |

Bảng 5.14. Dữ liệu huấn luyện và dữ liệu kiểm tra trong thử nghiệm 2

|  |  |  |  |
| --- | --- | --- | --- |
| Virus | Phát hiện | Tổng số | Tỉ lệ |
| Data 1 | 90 | 100 | 0.9 |
| Data 2 | 199 | 200 | 0.995 |
| Data 3 | 280 | 300 | 0.933 |
| Data 4 | 384 | 400 | 0.96 |
| Data 5 | 488 | 500 | 0.976 |

Bảng 5.15. Kết quả kiểm tra trên dữ liệu virus trong thử nghiệm 2

|  |  |  |  |
| --- | --- | --- | --- |
| Sạch | Phát hiện | Tổng số | Tỉ lệ |
| Data 1 | 10 | 50 | 0.2 |
| Data 2 | 11 | 50 | 0.22 |
| Data 3 | 16 | 50 | 0.32 |
| Data 4 | 13 | 50 | 0.26 |
| Data 5 | 9 | 50 | 0.18 |

Bảng 5.16. Kết quả kiểm tra trên dữ liệu sạch trong thử nghiệm 2

Tương tự như trong thừ nghiệm 1, ta tiến hành huấn luyện hệ thống trên bộ dữ liệu Data 2 và sau đó tiến hành kiểm tra trên các bộ dữ liệu còn lại.

Bảng 5.15 và 5.16 cho ta kết quả của quá trình kiểm tra này, ta có thể thấy ra kết quả kiểm tra không khác quá nhiều so với ở thử nghiệm 1. Mặc khác, tỉ lệ phát hiện đúng trong thử nghiệm 2 có phần cao hơn tỉ lệ 1. Sự chênh lêch được xem xét xuất phát từ các nguyên nhân:

* Quá trình huấn luyện trong thử nghiệm 1 được dừng lại khi hệ thống hội tụ đến tỉ lệ lỗi 0.209 (tỉ lệ lỗi không đổi sau nhiều vòng lặp) trong khi ở thử nghiệm 2, quá trình huấn luyện được dừng lại ở giá trị lỗi tốt hơn (0.169).
* Số tập tin virus ở bộ dữ liệu 2 nhiều gấp đôi số tập tin virus ở bộ dữ liệu 1.

Từ đó, ta nhận thấy rằng kết quả phát hiện virus của hệ thống phụ thuộc rất nhiều vào việc lựa chọn thông số của quá trình huấn luyện và dữ liệu huấn liệu đầu vào.

Chương 6 KẾT LUẬN

## 6.1. Các kết quả đạt được

Về mặt lý thuyết, sau một khoảng thời gian khá dài thu thập tài liệu liên quan, đọc hiểu, nghiên cứu từ nhiều nguồn khác nhau, nhóm cơ bản đã trang bị cho mình thật nhiều kiến thức quan trọng không chỉ trực tiếp liên quan đến đề tài, mà còn hữu ích cho nguồn tri thức bản thân. Những kết quả chính mà nhóm đã đạt được khi thực hiện đề tài này bao gồm:

* Nghiên cứu, nắm được các kiến thức cần thiết về virus máy tính, bao gồm khái niệm, phân loại cũng như phương thức mà một chương trình virus lây nhiễm và xâm hại hệ thống. Bên cạnh đó, nhóm còn nghiên cứu các phương pháp chính trong nhận dạng, phát hiện virus được sử dụng trong các chương trình Antivirus thông dụng.
* Các tri thức về hệ miễn dịch nhân tạo, mạng nơ ron nhân tạo cùng một số thuật toán máy học hỗ trợ được nhóm nghiên cứu khá kĩ, nắm vững về khái niệm, các thuật toán chính, cách tổ chức xây dựng mô hình… từ đó quan sát được khả năng kết hợp giữa AIS và ANN để hình thành ý tưởng giải quyết bài toán đã đặt ra.
* Thành công trong việc xây dựng một hướng tiếp cận dựa trên sự kết hợp giữa AIS và ANN để giải quyết bài toán phát hiện virus máy tính. Hướng tiếp cận được trình bày rõ ràng, chi tiết thông qua nhiều bài toán con. Mỗi bài toán này được giải quyết bởi các thuật toán thích hợp và xây dựng thành một mô hình nhỏ. Các bài toán có sự liên hệ thống nhất với nhau về thứ tự thực hiện cũng như ảnh hưởng lẫn nhau về kết quả giữa các quá trình.
* Việc kết hợp giữa hệ miễn dịch nhân tạo và một số thuật toán máy học như mạng nơ-ron, K-Means để giải quyết bài toán phát hiện virus máy tính là một hướng tiếp cận mới, có những ưu điểm nhất định so với các hướng tiếp cận truyền thống. Đầu tiên, việc lưu trữ một cơ sở dữ liệu mã nhận dạng virus là không cần thiết do đó khắc phục được tính cồng kềnh của các AV, đồng thời giảm thời gian kiểm tra tập tin. Hướng tiếp cận cho phép phát hiện được các virus mới do ta hướng đến việc nhận dạng các chuỗi mã đặc trưng cho một chương trình virus. Tuy nhiên, nhược điểm chính của hướng tiếp cận này là chỉ có khả năng phát hiện virus, không thể cho biết tên của nó nên giảm đi độ tin cậy đối với người dùng do thói quen sử dụng các AV sử dụng chuỗi mã nhận dạng.
* Một số bài báo khoa học đã được xây dựng và nhận đăng, cho thấy được tính mới và tính khoa học cao của đề tài, được trình bày chi tiết trong phần Phụ lục 2 của báo cáo.

Về mặt thực nghiệm, sau khi đã xây dựng được hướng tiếp cận giải quyết bài toán, nhóm đã tiến hành hiện thực hóa hướng tiếp cận này thành một chương trình minh họa nhằm kiểm tra khả năng ứng dụng thực nghiệm của hướng tiếp cận cũng như so sánh giữa mô hình thực tế và mô hình lý thuyết. Các kết quả mà nhóm thực hiện được bao gồm:

* Cài đặt thành công một chương trình nhỏ nhằm minh họa cho hướng tiếp cận được nêu ra. Bên cạnh đó, các kết quả thực nghiệm cũng cho thấy khả năng ứng dụng vào thực tiễn cao.
* Mỗi bài toán được đặt ra trong hướng tiếp cận được cài đặt thành một mô-đun riêng, có giao diện người dùng riêng và hoàn toàn có thể thực hiện độc lập với các mô-đun khác trong trường hợp cần thiết.
* Giao diện người dùng thân thiện, dễ dàng thay đổi các thông số huấn luyện cũng như quan sát hoạt động của các quá trình.
* Cơ sở dữ liệu huấn luyện được xây dựng dưới dạng thư mục, được tải vào chương trình dưới dạng đường dẫn, từ đó cho phép ta dễ dàng thay đổi dữ liệu huấn luyện và kiểm tra theo từng tình huống cụ thể.
* Nhiều kĩ thuật lập trình được áp dụng như Serialization, Threading, hướng đối tượng … được áp dụng để nâng cao hiệu quả huấn luyện.

## 6.2. Hạn chế

Bên cạnh những kết quả đã đạt được, vẫn còn rất nhiều hạn chế tồn tại trong cả hướng tiếp cận đề ra và quá trình cài đặt thử nghiệm, đòi hỏi có những nghiên cứu xa hơn để giải quyết chúng và phát triển đề tài lên một mức mới. Một số hạn chế chính có thể được kể ra như sau:

* Đề tài chỉ mới tập trung vào loại virus thực thi trên hệ điều hành Windows.
* Việc lựa chọn các thông số chủ yếu dựa vào thực nghiệm và các thông số mặc định, chưa có một nghiên cứu tổng quát trên việc lựa chọn tất cả thông số cũng như mức độ ảnh hưởng của mỗi thông số đến kết quả sau cùng.
* Thời gian các quá trình huấn luyện khá lâu mặc dù đã được ứng dụng nhiều kĩ thuật lập trình để rút ngắn thời gian thực hiện.
* Do sử dụng kĩ thuật lập trình đa luồng với ThreadPool nên chương trình sử dụng rất nhiều tài nguyên CPU khi hoạt động. Do nó sẽ tạo ra số tiểu trình tối đa mà CPU có thể quản lý để chạy chương trình nên với các máy tính có bộ vi xử lý càng cao thì tốc độ thực hiện chương trình càng nhanh.
* Kết quả chạy thực nghiệm chương trình chưa thực sự thuyết phục do các bộ dữ liệu chưa đủ lớn, kết quả kiểm tra còn phụ thuộc nhiều vào dữ liệu huấn luyện. Khả năng phát hiện các virus mới là hoàn toàn khả thi song vấn đề cảnh báo nhầm cũng là một trở ngại rất lớn.
* Chương trình thao tác trực tiếp trên các tập tin độc hại nên đòi hỏi phải sử dụng máy ảo trong quá trình thực thi của chương trình, do đó chưa thực sự đảm bảo an toàn cho người tiến hành các thực nghiệm kiểm tra chương trình.

## 6.3. Hướng phát triển

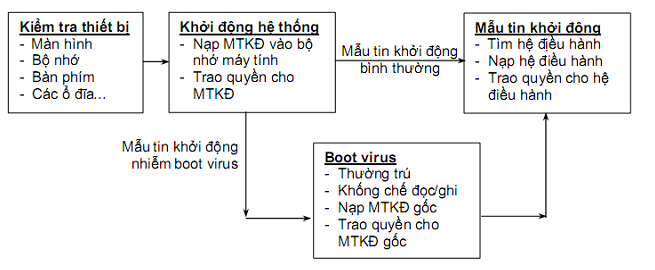
Ngay từ những giai đoạn đầu thực hiện đề tài, nhóm định hướng tập trung vào nghiên cứu giải quyết một bài toán nhỏ của bài toán phòng chống virus máy tính. Điều này tạo thuận lợi cho việc xây dựng hướng tiếp cận và cài đặt thử nghiệm, phù hợp với năng lực của người thực hiện ở mức độ đại học. Mặc dù vậy, đề tài cho phép ta mở rộng và phát triển nhằm giải quyết những bài toán bảo mật cao hơn, đặc biệt mở ra các hướng tiếp cận mới sử dụng các thuật toán mô phỏng sinh học vào giải quyết các vấn đề của con người. Dưới đây là một số các cải tiến cần thiết nhằm khắc phục những hạn chế kể trên đồng thời phát triển, mở rộng mô hình:

* Nghiên cứu việc áp dụng hướng tiếp cận để phát hiện các loại virus máy tính khác nhau.
* Nghiên cứu việc mở rộng hướng tiếp cận để giải quyết các bài toán khác nhau nhằm bảo vệ máy tính như hệ thống phát hiện xâm nhập, hệ thống phát hiện virus…
* Nghiên cứu việc kết hợp hướng tiếp cận được nêu ra với các hướng tiếp cận truyền thống nhằm phát huy hiệu quả nhận dạng, phát hiện virus máy tính.
* Tiếp tục vận dụng các kĩ thuật lập trình mới nhằm rút ngắn thời gian huấn luyện.
* Tỉ lệ cảnh báo nhầm cao cũng là một hạn chế rất lớn, do đó cần xây dựng những giải pháp để giảm bớt tỉ lệ cảnh báo nhầm mà không làm giảm tỉ lệ phát hiện virus.
* Số lượng chuỗi nhị phân sạch trong quá trình huấn luyện đôi khi rất lớn do được rút trích trực tiếp từ các tập tin sạch, điều này đòi hỏi ta phải tìm cách giảm số lượng của chuỗi sạch bằng cách xây dựng các bộ lọc, chỉ quan tâm đến những chuỗi nhị phân phân bố rộng rãi thay vì tất cả các chuỗi sạch.

PHỤ LỤC 1 CÁC LOẠI VIRUS MÁY TÍNH

## Boot virus

Là thuỷ tổ của virus máy tính, boot virus lây vào các mẫu tin khởi động (MTKĐ, boot record) trên hệ thống đĩa. Đối với đĩa mềm, MTKĐ nằm trên sector đầu tiên thuộc track 0, side 0 của đĩa (boot sector). Khi khởi động máy bằng đĩa mềm, sau quá trình tự kiểm tra (POST - Power On Self Test), thường trình khởi động máy trong ROM-BIOS sẽ đọc và nạp 512 byte trên boot sector vào bộ nhớ rồi trao quyền, không cần biết đoạn mã này sẽ thực hiện những công việc gì.



Hình P1.1: Nguyên tắc hoạt động của boot virus

Nguyên tắc của boot virus là thay thế MTKĐ bằng đoạn mã của nó để giành quyền khởi động máy. Khi được trao quyền, boot virus sẽ thường trú, khống chế các tác vụ đọc/ghi đĩa rồi nạp tiếp phần khởi động gốc (Hình P1.1). Quá trình lây nhiễm của boot virus trên đĩa mềm được minh họa như sau:

Đĩa mềm sạch: POST→ MTKĐ → HĐH

Đĩa mềm nhiễm: POST→ Boot virus → MTKĐ → HĐH

Đối với đĩa cứng, có 2 MTKĐ: master boot nằm ở sector đầu tiên thuộc track 0, side 0 và boot sector nằm ở sector đầu tiên của track 0, side 1. Trên đĩa cứng, boot virus có thể lây vào master boot hoặc boot sector. Quá trình minh họa như sau:

Đĩa cứng sạch: POST→ Master boot→ Boot sector → HĐH

Đĩa cứng nhiễm: POST → Boot virus→Master boot→ Boot sector→ HĐH, hoặc POST → Master boot → Boot virus → Boot sector → HĐH.

Do được kích hoạt trước HĐH nên boot virus có thể lây nhiều loại HĐH.

Tuy nhiênchúng cũng gặp nhiều trở ngại như:

* + Cơ hội kích hoạt: người dùng ít khi khởi động máy từ đĩa mềm.
  + Cơ hội lây lan: phần lớn các chip ROM được trang bị chức năng phát hiện tác vụ ghi vào boot sector hoặc cấm khởi động từ đĩa mềm.
  + Cơ hội tồn tại: do dung lượng nhỏ nên đĩa mềm ngày càng ít được sử dụng.

## File virus

Khác với boot virus, file virus lây vào tập tin chương trình của hệ điều hành. Tùy theo hệ điều hành và cơ chế hoạt động của vật chủ, file virus có cách lây nhiễm khác nhau (Hình P1.2a và P1.2b). Cơ chế hoạt động của file virus như sau:

* + Xác định điểm vào lệnh (IP-Instruction Pointer) gốc của vật chủ.
  + Ghép mã lệnh virus vào vật chủ. Tùy theo định dạng/tình trạng của vật chủ, vị trí ghép có thể ở đầu, giữa hoặc ở cuối vật chủ.
  + Thay điểm vào lệnh của vật chủ trỏ đến địa chỉ mã lệnh của virus.

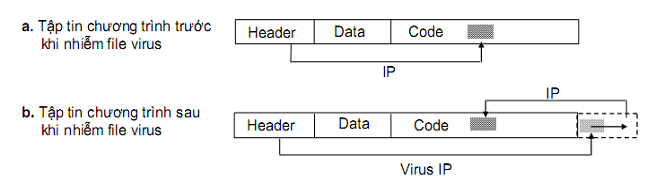
Khi vật chủ thi hành, file virus sẽ thực hiện các thao tác:

- Kiểm tra sự có mặt của virus trong vùng nhớ. Nếu chưa thường trú, chúng tiế hành sao chép vào vùng nhớ đã được hệ điều hành cấp phát.

- Tìm kiếm các vật chủ khác để lây vào

- Thực hiện các hoạt động đã được lập trình (phá hoại, do thám, định thời…)

- Phục hồi điểm vào lệnh gốc và trao quyền cho vật chủ.



Hình P1.2: Kỹ thuật lây nhiễm của file virus

Nhược điểm của file virus là dễ bị phát hiện do kích thước file thay đổi và chỉ lây mạnh trên các HĐH DOS và Windows 9x/Me. Trên Windows NT/2000/XP, chúng thường vấp phải hàng rào bảo vệ hạt nhân hệ thống Kernel32. Vì vậy file virus dần được hacker thay bằng các hình thức khác như worm, trojan horse…

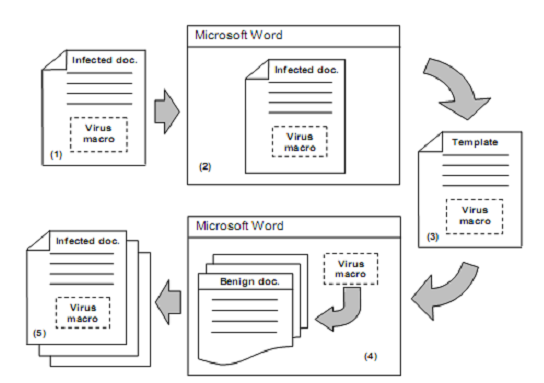
## Macro virus

Để tự động hóa thao tác sử dụng của người dùng, hãng Microsoft đã tích hợp tập lệnh VBA (Visual Basic Application) vào bộ công cụ Microsoft Office. Theo CSI (Cisco Safe Implementation), macro virus được định nghĩa như sau:

“Macro virus là loại virus máy tính được viết bằng ngôn ngữ VBA, thực thi trong môi trường của Microsoft Word hoặc Excel. Macro là tập lệnh giúp tự động hóa tác vụ của người dùng”

Các macro virus lợi dụng các template trong môi trường tương tác VBA làm nơi lưu trú. Khi người dùng mở tư liệu mới, các thủ tục của virus trong template sẽkết nhập vào tư liệu hiện thời dưới dạng các thủ tục VBA đã biên dịch (Hình P1.3).

Tùy theo phiên bản Microsoft Office, tập mã thủ tục VBA sẽ được chuyển dạng tương thích đi lên, làm biến đổi mã lệnh của macro virus.



Hình P1.3: Quy trình lây nhiễm của macro virus

1. Tập tin tư liệu chứa mã lệnh virus macro

2. Ứng dụng Office mở tư liệu nhiễm virus macro

3. Tập thủ tục VBA của virus kết nhập vào template

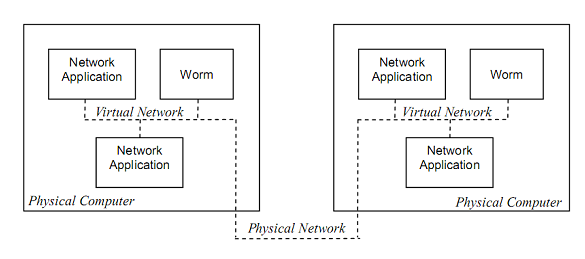
4. Virus macro trong template kết nhập vào các đối tượng mới

5. Hàng loạt tư liệu khác bị nhiễm virus macro

## Worm, sâu trình

Theo Vesselin Bonchev, worm (tạm dịch sâu trình) là:

“Các chương trình có khả năng tạo bản sao chính nó (thường là qua mạng máy tính) như một chương trình (hoặc tập các chương trình) độc lập, không phụ thuộc vào sự tồn tại của ứng dụng chủ”.

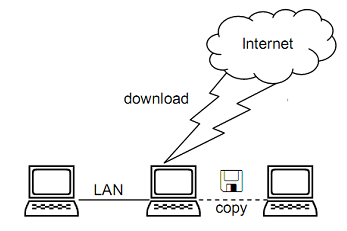


Hình P1.4: Worm lan truyền trên hệ thống mạng

Khai thác các dịch vụ vận chuyển dữ liệu như HTTP (HyperText Transfer Protocol), FTP (File Transfer Protocol), IRC (Internet Relay Chat), DNS (Domain Name Service), Drive Sharing, Email, Packet Routing…, sâu trình len lõi vào các hệ thống đích để tìm cơ hội do thám, đánh cắp, phá hoại dữ liệu. Khác với virus, sâu trình không đính vào các ứng dụng khác mà tồn tại trên hệ thống như một ứng dụng độc lập (hình P1.4). Sâu trình khó phát hiện vì chúng có định dạng giống các ứng dụng hợp thức. Tuy nhiên sâu trình cũng có nhược điểm. Do đứng độc lập nên khả năng tự kích hoạt bị hạn chế và dễ bị tiêu diệt. Kịch bản chung của sâu trình là đăng ký kích hoạt khi hệ điều hành hoàn tất quá trình khởi động.

## Trojan horse

Thuật ngữ “trojan horse” (mượn từ điển tích “ngựa gỗ thành Troie” trong thần thoại Hy Lạp, gọi tắt là trojan) là tên gọi các loại mã độc do người lập trình cài sẵn vào ứng dụng. Ban đầu, trojan là công cụ của người lập trình. Khi phát hiện các trường hợp không hợp thức (như sao chép lậu, bẻ khóa…), các đoạn mã này sẽ tự kích hoạt để trừng trị người dùng như xóa ứng dụng, làm treo máy… Về sau ý nghĩa của trojan càng mờ nhạt. Lợi dụng Internet, các hacker tung lên mạng các “tiện ích” (có cài sẵn trojan) để lừa người dùng download về sử dụng (ví dụ các phần mềm bẻ khóa, kích hoạt tài khoản xem phim trực tuyến…). Khi chạy loại “ứng dụng” này, các đoạn mã trojan sẽ thực thi các lệnh của hacker. CSI định nghĩa trojan horse là “loại chương trình được viết như tiện ích hữu hiệu, nhưng thực tế lại là công cụ tấn công”.



Hình P1.5: Cơ chế lan truyền của trojan horse

Trojan không tự lây mà lan truyền nhờ thao tác download và sao chép của người dùng (hình P1.5). Giống sâu trình, trojan có định dạng ứng dụng nên việc phát hiện gặp nhiều trở ngại. Trong một họ (family), trojan khởi thủy được gọi là mầm độc (germs). Thông thường, mã của mầm độc dùng để lắp ghép cho con cháu.

Đôi khi trojan còn chứa trong lòng nó một loại mã độc khác (như sâu trình chẳng hạn). Khi thực thi, loại trojan này (gọi là dropper) sẽ kết xuất mã độc thứ cấp ra hệ thống đích (tái sinh) mỗi khi một trong các bản sao của chúng bị tiêu diệt. Injector làmột dạng cải tiến của dropper: thay vì kết xuất ra bộ nhớ ngoài, mã độc được cài vào bộ nhớ trong để tránh bị phát hiện. Kết hợp giữa trojan và sâu trình, dropper và injector có đặc tính lây lan nên gần với virus hơn.

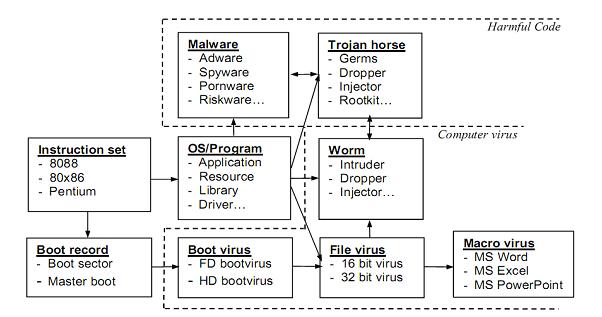
Trapdoor (hay backdoor) rất được giới hacker ưa chuộng. Khi lây vào hệ thống, backdoor có nhiệm vụ mở cổng (port, điểm truy cập ứng dụng), làm nội gián chờ đáp ứng thao tác quét cổng của hacker. Khi nhận được tín hiệu, backdoor khống chế hàng rào bảo vệ hệ thống, dọn đường đón các đợt thâm nhập từ bên ngoài.

Gần đây xuất hiện loại trojan đặc biệt nguy hiểm gọi là rootkit. Ban đầu, rootkit là tên gọi các bộ công cụ (kit) giúp người quản trị nắm quyền điều khiển hệ thống ở mức cao nhất (root). Trong tay hacker, rootkit trở thành công cụ đắc lực để đánh cắp mật khẩu truy nhập, thu thập thông tin trên máy nạn nhân hoặc che đậy các hoạt động thâm nhập bất hợp pháp. Hacktool là một dạng rootkit sơ cấp. Cao cấp hơn có các loại rootkit thám báo như keylogger (theo dõi hoạt động bàn phím), sniffer (theo dõi gói tin qua mạng), filehooker (theo dõi truy nhập tập tin)…

## Malware

Malware là tên gọi chung các loại phần mềm độc hại như:

* Adware: loại phần mềm tự động bật (popup) các cửa sổ quảng cáo, thay đổi các thiết lập hệ thống…, gây phiền phức cho người sử dụng.
* Spyware: loại phần mềm gián điệp, săn lùng thông tin thương mại, đánh cắp thông tin cá nhân như địa chỉ e-mail, độ tuổi, giới tính, thói quen mua sắm…
* Pornware: loại phần mềm đồi trụy phát tán từ các trang web khiêu dâm, tự động bật lên các hình ảnh gợi dục, phim sex... Pornware rất nguy hiểm cho trẻ em và thanh thiếu niên, làm suy đồi đạo đức xã hội.
* Riskware: phần mềm trôi nổi, không được kiểm định chất lượng. Riskware tiềm ẩn nhiều lỗi nghiêm trọng, làm suy giảm chất lượng phục vụ của hệ thống, có nguy cơ ảnh hưởng dữ liệu của người dùng.



Hình P1.6: Nguồn gốc mã độc và phân loại mã độc

Trong lĩnh vực anti-virus, nghiên cứu malware là bài toán sôi nổi và khá lý thú. Hình P1.6 mô tả nguồn gốc, quá trình phát triển của các loại virus máy tính và mã độc hiện nay.

PHỤ LỤC 2 CÔNG TRÌNH ĐÃ CÔNG BỐ

# DANH MỤC TÀI LIỆU THAM KHẢO

## TÀI LIỆU TIẾNG VIỆT

[1]. Nguyễn Phương Anh, Nguyễn Vĩnh Kha, “Nghiên cứu một số thuật toán thông minh trong máy học và hệ miễn dịch nhân tạo. Ứng dụng xây dựng hệ thống phát hiện virus máy tính”, khóa luận tốt nghiệp đại học, ĐH CNTT, ĐHQG Tp.HCM, 2011.

[2]. Nguyễn Xuân Hoài, Nguyễn Văn Trường, Vũ Mạnh Xuân, “Hệ miễn dịch nhân tạo và ứng dụng”, Tạp chí Khoa học và Công nghệ Đại học Thái Nguyên, 2007.

[3]. Trương Minh Nhật Quang, “Tiếp cận máy học và hệ chuyên gia để nhận dạng, phát hiện virus máy tính”, Luận án Tiến sĩ Toán học, Đại học KHTN, ĐHQG Tp.HCM, 2009.

## TÀI LIỆU TIẾNG ANH

[4]. Anastasia Doumas, Konstantinos Mavroudakis, Dimitris Gritzalis, Sokratis Katsikas, “Design of a neural network for recognition and classification of computer viruses, Computers & Security”, vol. 14, no. 5, 1995.

[5]. Dipankar Dasgupta and Luis Fernando Niño, “Immunological Computation - Theory and Applications”, pp. 27-29.

[6]. Farmer, J.D., Packard, N.H., Perelson, A.S.: “The immune system, adaptation, and machine learning, Physica D 22,pp. 187–204, 1986.

[7]. Forrest, S., Perelson, A., Allen, L., Cherukuri, R.: “Self-nonself discrimination in a computer”, Proceedings IEEE Symposium on Research in Security and Privacy, Los Alamitos, CA, pp. 202–212. IEEE Computer Society Press, Los Alamitos, 1994.

[8]. Fred Cohen, “Computer Viruses: Theory and Experiments. Computer and Security 6”, pp. 22-35, 1987.

[9]. Gerald Tesauro, Jeffrey O. Kephart, Gregory B. Sorkin, “Neural Networks for Computer Virus Recognition”, IEEE Expert, vol. 11, no. 4, pp. 5-6, 1996.

[10]. [Hans-Georg Beyer](http://link.springer.com/search?facet-author=%22Hans-Georg+Beyer%22), [Hans-Paul Schwefel](http://link.springer.com/search?facet-author=%22Hans-Paul+Schwefel%22), “Evolution strategies – A comprehensive introduction, [Natural Computing](http://link.springer.com/journal/11047)”, vol. 1, [Issue 1](http://link.springer.com/journal/11047/1/1/page/1), pp. 3-52, 2002.

[11]. Hazem M. El-Bakry, Nikos Mastorakis, “Fast virus detection by using high speed time delay neural networks”, Journal in Computer Virology, vol. 6, no. 2, pp. 115-122, 2010.

[12]. J.W.Kappler, N.Roehm, P.Marrack, “T cell tolerance by clonal elimination in the thymus.” in Cell, pp. 273–280, 1987.

[13]. Leandro N. de Castro and Fernando J. Von Zuben, ”Learning and Optimization Using the Clonal Selection Principle”, IEEE Transactions on Evolutionary Computation, Special Issue on Artificial Immune Systems, vol. 6, n. 3, pp. 239-251, 2002.

[14]. Minoru Fukumi, Yasue Mitsukura, Norio Akamatsu, "A New Rule Generation Method from Neural Networks Formed Using a Genetic Algorithm with Virus Infection", IEEE-INNS-ENNS International Joint Conference on Neural Networks, vol. 3, pp. 3413, 2000.

[15]. Rui Chao and Ying Tan, “A Virus Detection System Based on Artificial Immune System”, CIS '09 Proceedings of the 2009 International Conference on Computational Intelligence and Security, vol. 01, pp. 6-10, 2009.

[16]. [SuhaAfaneh](http://www.sciencedirect.com/science/article/pii/S1568494612003833) and [Raed Abu Zita](http://www.sciencedirect.com/science/article/pii/S1568494612003833), “Virus detection using clonal selection algorithm with Genetic Algorithm”, [Applied Soft Computing](http://www.sciencedirect.com/science/journal/15684946) ,[vol. 13, Issue 1](http://www.sciencedirect.com/science/journal/15684946/13/1), pp. 239–246, 2013.

[17]. The National Academy of Engineering Committee on Engineering's Grand Challenges (USA, 2008): http://www.engineeringchallenges.org

[18]. Vladimir Golovko, MyroslavKomar, Anatoly Sachenko, “Principles of Neural Network Artificial Immune System Design to Detect Attacks on Computers”, Modern Problems of Radio Engineering, Telecommunications and Computer Science, 2010.

[19]. Nguyen Phuong Anh, Mai Trong Khang, Nguyen Hoang Ngan, Vu Thanh Nguyen , “A combination of Immune Algorithm and Evolutionary Artificial Neural Network for Virus Detection”.

[20]. Warren S. McCulloch, and Walter H.Pitts, "A logical calculus of the ideas immanent in nervous activity", Bulletin of Mathematical Biophysics, vol. 5, pp. 115-133, 1943.

[21]. W.E.Paul, Ed.” Fundamental Immunology”, Raven Press Ltd. New York, pp.88–90, 1989.