**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA CÔNG NGHỆ PHẦN MỀM**

**BÙI NGỌC VIỆT**

**LỮ HOÀNG THÀNH**

**KHÓA LUẬN TỐT NGHIỆP**

**NGHIÊN CỨU MỘT SỐ THUẬT TOÁN MÁY HỌC KẾT HỢP VỚI HỆ MIỄN DỊCH NHÂN TẠO   
ĐỂ PHÁT HIỆN VÀ PHÂN LỚP VIRUS MÁY TÍNH**

**KỸ SƯ NGÀNH KĨ THUẬT PHẦN MỀM**

**TP. HỒ CHÍ MINH, THÁNG 07 NĂM 2016**

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA CÔNG NGHỆ PHẦN MỀM**

**BÙI NGỌC VIỆT - 11520477**

**LỮ HOÀNG THÀNH - 11520372**

**KHÓA LUẬN TỐT NGHIỆP**

**NGHIÊN CỨU MỘT SỐ THUẬT TOÁN MÁY HỌC KẾT HỢP VỚI HỆ MIỄN DỊCH NHÂN TẠO   
ĐỂ PHÁT HIỆN VÀ PHÂN LỚP VIRUS MÁY TÍNH**

**KỸ SƯ NGÀNH KĨ THUẬT PHẦN MỀM**

**GIẢNG VIÊN HƯỚNG DẪN**

**PGS.TS. VŨ THANH NGUYÊN**

**TP. HỒ CHÍ MINH, THÁNG 07 NĂM 2016**

DANH SÁCH HỘI ĐỒNG BẢO VỆ KHÓA LUẬN

Hội đồng chấm khóa luận tốt nghiệp, thành lập theo Quyết định số …………………… ngày ………………….. của Hiệu trưởng Trường Đại học Công nghệ Thông tin.

* 1. …………………………………………. – Chủ tịch.
  2. …………………………………………. – Thư ký.
  3. …………………………………………. – Ủy viên.
  4. …………………………………………. – Ủy viên.

|  |  |
| --- | --- |
| ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH  **TRƯỜNG ĐẠI HỌC**  **CÔNG NGHỆ THÔNG TIN \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA**  **VIỆT NAM**  **Độc Lập – Tự Do – Hạnh Phúc \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |

*TP.HCM, ngày…..tháng…..năm……….*

NHẬN XÉT KHÓA LUẬN TỐT NGHIỆP

(CỦA CÁN BỘ HƯỚNG DẪN)

**Tên khóa luận :**

**NGHIÊN CỨU MỘT SỐ THUẬT TOÁN MÁY HỌC KẾT HỢP VỚI HỆ MIỄN DỊCH NHÂN TẠO ĐỂ PHÁT HIỆN VÀ PHÂN LỚP VIRUS MÁY TÍNH**

**Nhóm SV thực hiện: Cán bộ hướng dẫn:**

**Bùi Ngọc Việt 11520477** **PGS.TS. Vũ Thanh Nguyên**

**Lữ Hoàng Thành 11520372**

**Đánh giá khóa luận:**

1. Về cuốn báo cáo:

Số trang 98 Số chương 07

Số bảng số liệu 14 Số hình vẽ 31

Số tài liệu tham khảo 14 Sản phẩm 01

Một số nhận xét về hình thức cuốn báo cáo: ………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

2. Về nội dung nghiên cứu:

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

3. Về chương trình ứng dụng:

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

4. Về thái độ làm việc của sinh viên:

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

**Đánh giá chung:**

………………………………………………………………………………………………………………………………………………………………………………

**Điểm từng sinh viên:**

Bùi Ngọc Việt **:** **………/10**

Lữ Hoàng Thành **:** **………/10**

**Người nhận xét**

(Ký tên và ghi rõ họ tên)

|  |  |
| --- | --- |
| ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH  **TRƯỜNG ĐẠI HỌC**  **CÔNG NGHỆ THÔNG TIN \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA**  **VIỆT NAM**  **Độc Lập – Tự Do – Hạnh Phúc \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |

*TP.HCM, ngày…..tháng…..năm……….*

NHẬN XÉT KHÓA LUẬN TỐT NGHIỆP

(CỦA CÁN BỘ PHẢN BIỆN)

**Tên khóa luận :**

**NGHIÊN CỨU MỘT SỐ THUẬT TOÁN MÁY HỌC KẾT HỢP VỚI HỆ MIỄN DỊCH NHÂN TẠO ĐỂ PHÁT HIỆN VÀ PHÂN LỚP VIRUS MÁY TÍNH**

**Nhóm SV thực hiện: Cán bộ phản biện:**

**Bùi Ngọc Việt 11520477 ThS. Phạm Thi Vương**

**Lữ Hoàng Thành 11520372**

**Đánh giá khóa luận:**

1. Về cuốn báo cáo:

Số trang 98 Số chương 07

Số bảng số liệu 14 Số hình vẽ 31

Số tài liệu tham khảo 14 Sản phẩm 01

Một số nhận xét về hình thức cuốn báo cáo: ………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

2. Về nội dung nghiên cứu:

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

3. Về chương trình ứng dụng:

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

4. Về thái độ làm việc của sinh viên:

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

**Đánh giá chung:**

………………………………………………………………………………………………………………………………………………………………………………

**Điểm từng sinh viên:**

Bùi Ngọc Việt **:** **………/10**

Lữ Hoàng Thành **:** **………/10**

**Người nhận xét**

(Ký tên và ghi rõ họ tên)

|  |  |
| --- | --- |
| ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH  **TRƯỜNG ĐẠI HỌC**  **CÔNG NGHỆ THÔNG TIN \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA**  **VIỆT NAM**  **Độc Lập – Tự Do – Hạnh Phúc \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |

**ĐỀ CƯƠNG CHI TIẾT**

|  |
| --- |
| **Tên đề tài : NGHIÊN CỨU MỘT SỐ THUẬT TOÁN MÁY HỌC KẾT HỢP VỚI HỆ MIỄN DỊCH NHÂN TẠO ĐỂ PHÁT HIỆN VÀ PHÂN LỚP VIRUS MÁY TÍNH** |
| **Cán bộ hướng dẫn : PGS.TS. Vũ Thanh Nguyên** |
| **Thời gian thực hiện :** Từ ngày 01/03/2016 đến ngày 01/06/2016 |
| **Sinh viên thực hiện:**  **Bùi Ngọc Việt - 11520477**  **Lữ Hoàng Thành – 11520372** |
| **Nội dung đề tài :** NGHIÊN CỨU MỘT SỐ THUẬT TOÁN MÁY HỌC VÀ KẾT HỢP VỚI HỆ MIỄN DỊCH NHÂN TẠO ĐỂ PHÁT HIỆN VÀ PHÂN LỚP VIRUS MÁY TÍNH |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Kế hoạch thực hiện:**  *Khóa luận của nhóm được thực hiện trong vòng 16 tuần, nhóm em chia nhỏ công việc theo tuần để dễ quản lý, cụ thể như sau:*   |  |  |  | | --- | --- | --- | | Tuần | Công việc | Sinh viên thực hiện | | 1-2 (2 tuần) | Khảo sát và xác định ứng dụng cụ thể sẽ làm.  Phác thảo các nội dung cần tìm hiểu. | Việt, Thành. | | 3 (1 tuần) | Tìm hiểu tình hình an ninh thông tin, và bảo mật dữ liệu hiện nay. | Việt. | | Nghiên cứu về virus và các nghiên cứu liên quan. | Thành. | | 4-5 (2 tuần) | Tìm hiểu tổng quan Data Mining. | Thành. | | Tìm hiểu tổng quan Machine Learning. Một số Machine Learning tiêu biểu. | | Nghiên cứu về miễn dịch con người và hệ miễn dịch nhân tạo. | Việt. | | 6-8 (3 tuần) | Tìm hiểu mạng noron Kohonen. | Việt. | | Tìm hiểu Support Vector Machine. | Việt, Thành. | | 9-12 (3 tuần) | Thiết kế giao diện ứng dụng.  Tiến hành viết báo cáo từng phần. | Thành. | | Cài đặt chương trình. | Việt. | | 13-14 (2 tuần) | Kiểm tra lỗi hệ thống và sửa chữa nếu có.  Test và nâng cấp ứng dụng. | Việt, Thành. | | 15-16 (2 tuần) | Thời gian dự trữ để tiến hoàn chỉnh sửa và tất báo cáo. | Việt, Thành. | | |
| **Xác nhận của CBHD**  (Ký tên và ghi rõ họ tên) | **TP.HCM, ngày….. tháng….. năm…..**  **Sinh viên**  (Ký tên và ghi rõ họ tên) |

**LỜI CẢM ƠN**

Để có thể hoàn thành khóa luận một cách hoàn chỉnh, chúng em đã nhận được rất nhiều sự hướng dẫn nhiệt tình của quý Thầy Cô. Vì vậy, chúng em xin phép được gửi những lời cảm ơn chân thành nhất.

Đầu tiên, chúng em xin chân thành cảm ơn và bày tỏ lòng biết ơn sâu sắc nhất đến **PGS.TS Vũ Thanh Nguyên**, thầy đã tận tình chỉ bảo, hướng dẫn, truyền đạt những kinh nghiệm quý báu và động viên nhóm chúng em để chúng em có thể hoàn thành khóa luận một cách tốt đẹp nhất. Nhóm chúng em xin gởi đến thầy những lời tri ân nhất đối với những điều mà thầy đã dành cho chúng em.

Chúng em xin chân thành cảm ơn anh **Nguyễn Tấn Toàn** giảng viên khoa Công Nghệ Phần Mềm đã kề vai, sát cánh hướng dẫn và chia sẻ những kinh nghiệm quý báu, kiến thức chuyên môn và các tài liệu liên quan để chúng em có thể hoàn thành tốt.

Chúng em cũng xin chân thành cảm ơn toàn thể quý thầy cô trong khoa Công Nghệ Phần Mềm và toàn thể quý thầy cô của Trường Đại Học Công Nghệ Thông Tin TP.HCM. Quý thầy cô đã luôn tận tình truyền đạt những kiến thức, kinh nghiệm quý báu cũng như tạo mọi điều kiện thuận lợi nhất cho chúng em trong suốt quá trình học tập tại trường và đặc biệt hơn là trong suốt quá trình thực hiện khóa luận này.

Sau cùng, chúng em xin kính chúc toàn thể quý thầy cô trong khoa Công Nghệ Phần Mềm và **PGS.TS Vũ Thanh Nguyên** luôn dồi dào sức khỏe, sự nhiệt huyết và thành công trong cuộc sống cũng như trong sứ mệnh truyền đạt kiến thức cho các thế hệ trẻ mai sau.

*TP. Hồ Chí Minh, ngày 01 tháng 07 năm 2016*

*Nhóm thực hiện*

Bùi Ngọc Việt – Lữ Hoàng Thành

MỤC LỤC

[Chương 1. TỔNG QUAN ĐỀ TÀI 2](#_Toc455888234)

[1.1. Giới thiệu đề tài 2](#_Toc455888235)

[1.2. Mục tiêu đề tài 3](#_Toc455888236)

[1.3. Đối tượng phạm vi nghiên cứu 3](#_Toc455888237)

[1.3.1. Đối tượng nghiên cứu 3](#_Toc455888238)

[1.3.2. Phạm vi nghiên cứu. 4](#_Toc455888239)

[1.4. Nội dung 4](#_Toc455888240)

[Chương 2. TỔNG QUAN VIRUS MÁY TÍNH 5](#_Toc455888241)

[2.1. Tổng quan 5](#_Toc455888242)

[2.1.1. Định nghĩa 5](#_Toc455888243)

[2.1.2. Một số loại phổ biến 6](#_Toc455888244)

[2.2. Hình thức lây nhiễm 10](#_Toc455888245)

[2.2.1. Cách cổ điển 11](#_Toc455888246)

[2.2.2. Qua e-mail 11](#_Toc455888247)

[2.2.3. Qua mạng Internet 13](#_Toc455888248)

[2.3. Cơ chế chuẩn đoán 14](#_Toc455888249)

[2.3.1. Phát hiện virus dựa vào chuỗi nhận dạng 14](#_Toc455888250)

[2.3.2. Phát hiện virus dựa vào hành vi 15](#_Toc455888251)

[2.3.3. Phát hiện virus dựa vào ý định 15](#_Toc455888252)

[Chương 3. HỆ MIỄN DỊCH NHÂN TẠO 17](#_Toc455888253)

[3.1. Khái niệm 17](#_Toc455888254)

[3.2. Cấu trúc 17](#_Toc455888255)

[3.3. Các mô hình trừu tượng của hệ miễn dịch và tương tác giữa chúng 19](#_Toc455888256)

[3.3.1. Tổng quát hoá các thành phần sinh học của hệ miễn dịch 19](#_Toc455888257)

[3.3.2. Không gian hình dạng (Shape - Space) 19](#_Toc455888258)

[3.3.3. Đánh giá tương tác giữa các phần tử 21](#_Toc455888259)

[3.3.3.1. Không gian Euclid và không gian Mahattan 21](#_Toc455888260)

[3.3.3.2. Không gian Hamming 21](#_Toc455888261)

[3.4. Một số thuật toán trong hệ miễn dịch nhân tạo 22](#_Toc455888262)

[3.4.1. Thuật toán chọn lọc âm tính 22](#_Toc455888263)

[3.4.2. Thuật toán chọn lọc âm tính 24](#_Toc455888264)

[3.5. Ứng dụng 25](#_Toc455888265)

[3.5.1. Điều khiển 26](#_Toc455888266)

[3.5.2. An ninh máy tính 26](#_Toc455888267)

[3.5.3. Phát hiện lỗi 27](#_Toc455888268)

[3.5.4. Phát hiện bất thường trong hệ thống 27](#_Toc455888269)

[3.5.5. Tối ưu hóa 27](#_Toc455888270)

[3.5.6. Khai phá dữ liệu 27](#_Toc455888271)

[Chương 4. MẠNG NƠ-RON NHÂN TẠO 28](#_Toc455888272)

[4.1. Tiếp cận mạng nơ-ron sinh học 28](#_Toc455888273)

[4.2. Mạng nơ-ron nhân tạo 29](#_Toc455888274)

[4.2.1. Cấu trúc một nơ-ron nhân tạo 29](#_Toc455888275)

[4.2.2. Cấu trúc mạng nơ-ron nhân tạo 31](#_Toc455888276)

[4.3. Tìm hiểu mạng nơ-ron Kohonen 32](#_Toc455888277)

[4.3.1. Giới thiệu 32](#_Toc455888278)

[4.3.2. Cầu trúc của mạng nơ-ron Kohonen 32](#_Toc455888279)

[4.3.3. Bài toán phân cụm và mạng Kohonen 33](#_Toc455888280)

[4.4. Một vài ứng dụng của SOM 33](#_Toc455888281)

[Chương 5. DATA MINING VÀ MACHINE LEARNING 34](#_Toc455888282)

[5.1. Data Mining 34](#_Toc455888283)

[5.1.1. Tổng quan 34](#_Toc455888284)

[5.1.1.1. Quá trình khám phá tri thức 34](#_Toc455888285)

[5.1.1.2. Khai phá dữ liệu 36](#_Toc455888286)

[5.1.2. Các loại dữ liệu có thể khai phá 37](#_Toc455888287)

[5.1.3. Các phương pháp, kỹ thuật chính trong khai phá dữ liệu 38](#_Toc455888288)

[5.1.3.1. Phân lớp và dự đoán (Classification & Prediction) 39](#_Toc455888289)

[5.1.3.2. Luật kết hợp (Association Rules) 40](#_Toc455888290)

[5.1.3.3. Khai thác mẫu tuần tự (Sequential / Temporal patterns) 40](#_Toc455888291)

[5.1.3.4. Phân nhóm - đoạn (Clustering / Segmentation) 40](#_Toc455888292)

[5.1.3.5. Hồi quy (Regression) 41](#_Toc455888293)

[5.1.3.6. Tổng hợp hóa (Summarization) 41](#_Toc455888294)

[5.1.3.7. Mô hình hóa sự phụ thuộc (dependency modeling) 42](#_Toc455888295)

[5.1.3.8. Phát hiện sự biến đổi và độ lệch (Change and deviation detection) 42](#_Toc455888296)

[5.1.4. Ứng dụng của khai phá dữ liệu 42](#_Toc455888297)

[5.1.5. Những thách thức trong khai phá dữ liệu 43](#_Toc455888298)

[5.2. Machine Learning 44](#_Toc455888299)

[5.2.1. Giới thiệu về máy học 44](#_Toc455888300)

[5.2.2. Phân loại 44](#_Toc455888301)

[5.2.2.1. Học có giám sát 45](#_Toc455888302)

[5.2.2.2. Học bán giám sát 46](#_Toc455888303)

[5.2.2.3. Học không giám sát 47](#_Toc455888304)

[5.3. Suport Vector Machine 47](#_Toc455888305)

[5.3.1. Phân lớp dữ liệu 47](#_Toc455888306)

[5.3.2. Bài toán phân lớp dữ liệu 47](#_Toc455888307)

[5.3.3. Quá trình phân lớp dữ liệu 49](#_Toc455888308)

[5.4. Sử dụng SVM 50](#_Toc455888309)

[5.4.1. Giới thiệu 50](#_Toc455888310)

[5.4.2. Ý tưởng của phương pháp 51](#_Toc455888311)

[5.4.3. Không gian đặc trưng 51](#_Toc455888312)

[5.4.4. Cơ sở lý thuyết của phương pháp SVM 52](#_Toc455888313)

[5.4.5. Các bước chính của phương pháp SVM 52](#_Toc455888314)

[5.4.6. Đánh giá độ chính xác của hệ thống 52](#_Toc455888315)

[Chương 6. XÂY DỰNG HỆ THỐNG PHÁT HIỆN VIRUS 55](#_Toc455888316)

[6.1. Ý tưởng 55](#_Toc455888317)

[6.1.1. Xây dựng dữ liệu 55](#_Toc455888318)

[6.1.1.1. Dữ liệu 56](#_Toc455888319)

[6.1.1.2. Giải pháp 56](#_Toc455888320)

[6.1.2. Phân lớp các chuỗi 57](#_Toc455888321)

[6.1.2.1. Dữ liệu 57](#_Toc455888322)

[6.1.2.2. Hướng giải quyết 58](#_Toc455888323)

[6.1.3. Phân lớp tập tin 59](#_Toc455888324)

[6.1.3.1. Dữ liệu 59](#_Toc455888325)

[6.1.3.2. Hướng giải quyết 59](#_Toc455888326)

[6.2. Cài đặt 61](#_Toc455888327)

[6.2.1. Pre-Process 61](#_Toc455888328)

[6.2.2. Detector 63](#_Toc455888329)

[6.2.3. Training 66](#_Toc455888330)

[6.2.4. Scanner 68](#_Toc455888331)

[6.3. Kiểm thử 69](#_Toc455888332)

[6.3.1. Dữ liệu 69](#_Toc455888333)

[6.3.2. Kết quả 69](#_Toc455888334)

[6.3.2.1. Kết quả thử nghiệm trên bộ data 1 70](#_Toc455888335)

[6.3.2.2. Kết quả thử nghiệm trên bộ data 2 70](#_Toc455888336)

[6.3.2.3. Kết quả thử nghiệm trên bộ data 3 71](#_Toc455888337)

[6.3.2.4. Kết quả thử nghiệm trên bộ data 4 71](#_Toc455888338)

[Chương 7. KẾT LUẬN 72](#_Toc455888339)

[7.1. Kết quả đạt được 72](#_Toc455888340)

[7.2. Hướng phát triển 73](#_Toc455888341)

[TÀI LIỆU THAM KHẢO 75](#_Toc455888342)

DANH MỤC HÌNH VẼ

[Hình 2‑1. Một số loại virus máy tính 6](#_Toc456047746)

[Hình 2‑2. Các hình thức lây nhiễm 10](#_Toc456047747)

[Hình 2‑3. Lây nhiễm qua E-Mail 12](#_Toc456047748)

[Hình 2‑4. Lây nhiễm qua Internet 13](#_Toc456047749)

[Hình 2‑5. Cơ chế chuẩn đoán 14](#_Toc456047750)

[Hình 3‑1. Cấu trúc phân tầng của AIS 18](#_Toc456047751)

[Hình 3‑2. Kháng thể nhận diện kháng nguyên dựa vào phần bù 19](#_Toc456047752)

[Hình 3‑3. Thuật toán miễn dịch nhân tạo 22](#_Toc456047753)

[Hình 3‑4. Sơ đồ thuật toán chọn lọc tích cực 24](#_Toc456047754)

[Hình 3‑5. Mô hình thuật toán chọn lọc âm tính. 25](#_Toc456047755)

[Hình 3‑6. Các ứng dụng của hệ AIS 26](#_Toc456047756)

[Hình 4‑1. Cấu tạo nơ-ron sinh học. 29](#_Toc456047757)

[Hình 4‑2. Cấu tạo nơ-ron nhân tạo. 29](#_Toc456047758)

[Hình 4‑3. Cấu trúc mạng noron nhân tạo 31](#_Toc456047759)

[Hình 4‑4. Một số mạng nơ-ron nhân tạo được sử dụng phổ biến hiện nay 31](#_Toc456047760)

[Hình 5‑1. Các bước trong KDD 35](#_Toc456047761)

[Hình 5‑2. Khai phá dữ liệu – các loại dữ liệu 37](#_Toc456047762)

[Hình 5‑3. Khai phá dữ liệu - Phương pháp và kỹ thuật 39](#_Toc456047763)

[Hình 5‑4. Phân loại máy học 45](#_Toc456047764)

[Hình 5‑5. Ánh xạ vào không gian đặc trưng 51](#_Toc456047765)

[Hình 6‑1. Cơ chế rút trích chuỗi nhị phân 57](#_Toc456047766)

[Hình 6‑2. Sơ đồ quy trình xây dựng dữ liệu huấn luyện 57](#_Toc456047767)

[Hình 6‑3. Các bước chuẩn bị dữ liêu cho quá trình training mạng nơ-ron 58](#_Toc456047768)

[Hình 6‑4. Quá trình tính độ nguy hiểm của tệp tin 60](#_Toc456047769)

[Hình 6‑5. Màn hình setting tham số cho module Pre-Process 62](#_Toc456047770)

[Hình 6‑6. Màn hình kết quả của module Pre-Process 63](#_Toc456047771)

[Hình 6‑7. Màn hình setting cho module detector 65](#_Toc456047772)

[Hình 6‑8. Màn hình kết quả của Module Detector 65](#_Toc456047773)

[Hình 6‑9. Màn hình setting tham số cho module Training 67](#_Toc456047774)

[Hình 6‑10. Màn hình kết quả của Module Training 67](#_Toc456047775)

[Hình 6‑11. Màn hình setting cho module Scanner 68](#_Toc456047776)

DANH MỤC BẢNG

[Bảng 5‑1. Đánh giá độ phức tạp hệ thống SVM 53](#_Toc456047693)

[Bảng 6‑1. Các đối tượng chính của giai đoạn xây dựng dữ liệu huấn luyện 61](#_Toc456047694)

[Bảng 6‑2. Các phương thức chính của giai đoạn xây dựng dữ liệu huấn luyện 62](#_Toc456047695)

[Bảng 6‑3. Một số hàm, lớp trong module Detector 63](#_Toc456047696)

[Bảng 6‑4. Tham số trong module Detector 64](#_Toc456047697)

[Bảng 6‑5. Các phương thức trong trong module Training 66](#_Toc456047698)

[Bảng 6‑6. Các tham số trong module Training 66](#_Toc456047699)

[Bảng 6‑7. Các phương thức trong trong module Scanner 68](#_Toc456047700)

[Bảng 6‑8. Dữ liệu đầu vào 69](#_Toc456047701)

[Bảng 6‑9. Sự tương quan giữa giá trị r và quá trình xây dựng dữ liệu huấn luyện 70](#_Toc456047702)

[Bảng 6‑10. Kết quả thử nghiệm 1 70](#_Toc456047703)

[Bảng 6‑11. Kết quả thử nghiệm 2 71](#_Toc456047704)

[Bảng 6‑12. Kết quả thử nghiệm 3 71](#_Toc456047705)

[Bảng 6‑13. Kết quả thử nghiệm 3 72](#_Toc456047706)

DANH MỤC TỪ VIẾT TẮT

|  |  |
| --- | --- |
| Công nghệ thông tin | CNTT |
| Artificial Immune System | AIS |
| Negative Selection Algorythm | NSA |
| Clonal Selection Algorithm | CLONALG |
| Cơ sở dữ liệu | CSDL |
| Khai phá dữ liệu | KPDL |
| Anti-Virus | AV |
| Personal computer | PC |
| Deoxyribonucleic Acid | DNA |
| Artificial Neural Network | ANN |
| Self-Organizing Map | SOM |
| Support Vector Machine | SVM |

DANH MỤC TỪ ANH – VIỆT

|  |  |
| --- | --- |
| Computer virus | Virus máy tính |
| Artificial Immune System | Hệ miễn dịch nhân tạo |
| Shape – Space | Không gian hình dạng |
| Real-Shape-Space | Không gian thực |
| Integer-Shape-Space | Không gian nguyên |
| Hamming-Shape-Space | Không gian Hamming |
| Symbolic-Shape-Space | Không gian ký hiệu |
| Positive Selection | Chọn lọc dương tính |
| Negative Selection | Chọn lọc âm tính |
| Clonal Selection | Chọn lọc nhân bản |
| Machine Learning | Học máy |
| Classification | Phân lớp |
| Artificial Neural Network | Mạng thần kinh nhân tạo |
| Self-Organizing Map | Mạng tự tổ chức |

**MỞ ĐẦU**

Đi kèm với sự phát triển của phần mềm máy tính mang lại những lợi ích trong tất cả các lĩnh vực trong cuộc sống là sự xuất hiện của nhiều chương trình và mã nguồn độc (virus) nhằm các mục đích khác nhau như: đánh cắp thông tin, phá hoại, làm đảo lộn môi trường mạng Internet hoặc một số đích quân sự, chống phá nhà nước của các thế lực thù địch. Các chương trình độc hại có thể xâm nhập vào hệ thống máy tính của người dùng bằng nhiều cách như qua email, trao đổi tập tin trên thiết bị lưu trữ bên ngoài (USB)... Thông thường, người dùng khó có thể nhận biết được sự xuất hiện của chương trình virus cho đến khi chúng gây ra các hỏng hóc cho hệ thống máy tính và các phần mềm ứng dụng.

Để phát hiện virus nhằm ngăn chặn những tác hại do chúng mang lại, các nhà nghiên cứu khoa học đã và đang ngày càng cho ra nhiều công nghệ và phương pháp mới. Một trong những phướng pháp đó là 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. Vì được xây dựng dựa trên hệ miễn dịch sinh học của con người nên ta có thể 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.

Xuất phát từ các yếu tố trên, nhóm đã tập trung nghiên cứu đề tài:

***“Nghiên cứu một số thuật toán máy học và kết hợp với hệ miễn dịch nhân tạo để phát hiện và phân lớp virus máy tính”***

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.

# TỔNG QUAN ĐỀ TÀI

## Giới thiệu đề tài

Với sự bùng nổ và phát triển mạnh mẽ của CNTT (Công nghệ thông tin) trong thời gian qua, CNTT ngày càng phổ biến và giữ vai trò đặc biệt quan trọng trong việc phát triển kinh tế và đất nước. Giúp xử lý tính toán dữ liệu, kết nối thông tin liên lạc của các đơn vị tổ chức trong và ngoài nước. CNTT vừa là công cụ, vừa là động lực thúc đẩy quá trình công nghiệp hóa, hiện đại hóa đất nước. Kéo theo đó 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 virus máy tính ngày nay xuất hiện các biến thể mới từng giờ từng phút. Khi một máy tính được đưa ra sử dụng sẽ khó tránh khỏi vấn đề về virus. Virus có thể lây lan qua bất kỳ con đường nào: Qua Internet, email, chat, qua các đường truyền gián tiếp như ổ đĩa USB, ổ cứng, đĩa CD…

Sự phát triển của Internet làm bùng nổ sự lây lan của virus trên phạm vi toàn cầu với hậu quả cực kỳ nghiêm trọng. Các hệ thống máy tính thường xuyên phải đối phó với các đợt tấn công, dẫn đến nguy cơ bị đình trệ, tắc nghẽn và tê liệt. Việc sử dụng virus lấy cắp tài khoản ngân hàng hay thông tin cá nhân quan trọng, mở cửa sau cho tin tặc đột nhập chiếm quyền điều khiển và các hành động khác nhằm có lợi cho người phát tán trở nên phổ biến.

Ngày nay, việc phòng chống virus trở thành vấn đề cấp bách và sống còn của mọi tổ chức, mọi doanh nghiệp lớn và nhỏ cũng như của từng các nhân. Trong bối cảnh đó đề tài “*Nghiên cứu một số thuật toán máy học và kết hợp với hệ miễn dịch nhân tạo để phát hiện và phân lớp virus máy tính*” được tiến hành nhằm nghiên cứu các thuật toán máy học (cụ thể là nhóm sẽ sử dụng SVM – Support Vector Machines) cũng như 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 đề tài

Trong phạm vi nghiên cứu của đề tài, nhóm sẽ tìm hiểu về hệ miễn dịch nhân tạo trên máy tính và kết hợp với một số thuật toán máy học khác trong việc phát hiện virus, với các tiêu chí cơ bản:

* Đưa ra hướng tiếp cận từ hệ miễn dịch con người sau đó áp dụng vào miễn dịch nhân tạo trên máy tính, khắc phục nhược điểm của các hướng tiếp cận truyền thống và góp phần tăng khả năng phát hiện, nhận biết các loại virus mới.
* Đề tài không hướng đến việc nhận dạng xác định tên của virus máy tính, nhưng 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 có khả năng xác định một tập tin nào đó là virus hay chương trình sạch, không phải là một chương trình duyệt virus như các phần mềm diệt virus hiện nay.
* 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 nhằm giúp ích cho mọi người trong việc bảo mật thông tin.

## Đối tượng phạm vi nghiên cứu

### Đối tượng nghiên cứu

* Virus và phát hiện virus
* Hệ miễn dịch nhân tạo
* Khai phá dữ liệu và máy học
* Phương pháp phân lớp tài liệu SVM

### Phạm vi nghiên cứu.

## Nội dung

Nội dung của luận văn được chia thành 6 chương:

**Chương 1:** Dẫn nhập và giới thiệu chung về đề tài.

**Chương 2:** Tìm hiểu về tổng quan về các loại virus phổ biến cũng như các hình thức tấn công, hình thức lây nhiễm và cơ chế chuẩn đoán virus.

**Chương 3:** Cung cấp cái nhìn tổng quan hề hệ miễn dịch nhân tạo cũng như ứng dụng của chúng.

**Chương 4:** Tìm hiểu về khai phá dữ liệu các phương pháp, ứng dụng … Tiếp đến là giới thiệu và phân loại các lại máy học.

**Chương 5:** Giới thiệu về bài toán phân lớp và áp dụng SVM để phân lớp dữ liệu.

**Chương 6:** Xây dựng hệ thông phát hiện virus.

**Chương 7:** Kết quả đạt được và hướng phát triển.

# TỔNG QUAN VIRUS MÁY TÍNH

## Tổng quan

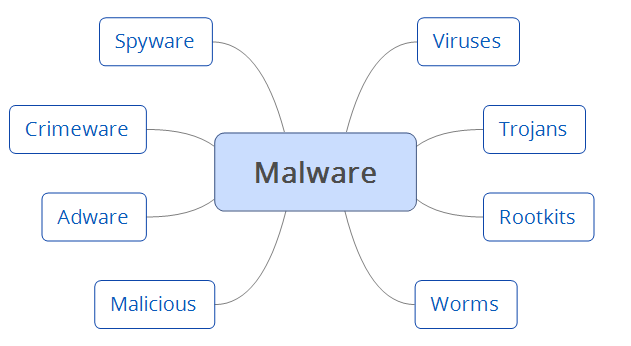
### Định nghĩa

Virus máy tính (computer virus) là chương trình phần mềm, và chương trình này được thiêt kế để có thể lây lan ra các file chương trình hoặc các khu vực hệ thống của đĩa cứng hay đĩa mềm [1]. Đặc điểm đặc chưng của chúng là khả năng tự nhân bản. Virus luôn luôn có thể tự mình làm mọi việc mà không cần có sự cho phép của người sử dụng, nhưng tất nhiên là lần đầu tiên phải có một thao tác nào đó của người dùng kích hoạt nó lên (do không biết đó là virus).

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” [2].

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.

### Một số loại phổ biến



Hình ‑. Một số loại virus máy tính

**Compiled Virus** là virus mà mã thực thi của nó đã được dịch hoàn chỉnh bởi một trình biên dịch để nó có thể thực thi trực tiếp từ hệ điều hành [1] [3]. Các loại boot virus như (Michelangelo và Stoned), file virus (như Jerusalem) rất phổ biến trong những năm 80 là virus thuộc nhóm này, compiled virus cũng có thể là pha trộn bởi cả boot virus va file virus trong cùng một phiên bản.

**Interpreted Virus** là một tổ hợp của mã nguồn mã chỉ thực thi được dưới sự hỗ trợ của một ứng dụng cụ thể hoặc một dịch vụ cụ thể trong hệ thống [1] [3]. Một cách đơn giản, virus kiểu này chỉ là một tập lệnh, cho đến khi ứng dụng gọi thì nó mới được thực thi. Macro virus, scripting virus là các virus nằm trong dạng này. Macro virus rất phổ biến trong các ứng dụng Microsoft Office khi tận dụng khả năng kiểm soát việc tạo và mở file để thực thi và lây nhiễm. Sự khác nhau giữa macro virus và scripting virus là: macro virus là tập lệnh thực thi bởi một ứng dụng cụ thể, còn scripting virus là tập lệnh chạy bằng một service của hệ điều hành. Melisa là một ví dụ xuất sắc về macro virus, Love Stages là ví dụ cho Scripting Virus.

**Worm** cũng là một chương trình có khả năng tự nhân bản và tự lây nhiễm trong hệ thống tuy nhiên nó có khả năng “tự đóng gói”, điều đó có nghĩa là worm không cần phải có “file chủ” để mang nó khi nhiễm vào hệ thống. Như vậy, có thể thấy rằng chỉ dùng các chương trình quét file sẽ không diệt được worm trong hệ thống vì worm không “bám” vào một file hoặc một vùng nào đó trên đĩa cứng. Mục tiêu của worm bao gồm cả làm lãng phí nguồn lực băng thông của mạng và phá hoại hệ thống như xoá file, tạo backdoor, thả keylogger,... Tấn công của worm có đặc trưng là lan rộng cực kỳ nhanh chóng do không cần tác động của con người (như khởi động máy, copy file hay đóng/mở file) [1] [3].

**Trojan Horse** là loại mã độc hại được đặt theo sự tích “Ngựa thành Troa”. Trojan horse không tự nhân bản tuy nhiên nó lây vào hệ thống với biểu hiện rất ôn hoà nhưng thực chất bên trong có ẩn chữa các đoạn mã với mục đích gây hại. [1]

Trojan có thể lựa chọn một trong 3 phương thức để gây hại:

* Tiếp tục thực thi các chức năng của chương trình mà nó bám vào, bên cạnh đó thực thi các hoạt động gây hại một cách riêng biệt (ví dụ như gửi một trò chơi dụ cho người dùng sử dụng, bên cạnh đó là một chương trình đánh cắp password).
* Tiếp tục thực thi các chức năng của chương trình mà nó bám vào, nhưng sửa đổi một số chức năng để gây tổn hại (ví dụ như một trojan giả lập một cửa sổ login để lấy password) hoặc che dấu các hành động phá hoại khác (ví dụ như trojan che dấu cho các tiến trình độc hại khác bằng cách tắt các hiển thị của hệ thống).
* Thực thi luôn một chương trình gây hại bằng cách núp dưới danh một chương trình không có hại (ví dụ như một trojan được giới thiệu như là một chò chơi hoặc một tool trên mạng, người dùng chỉ cần kích hoạt file này là lập tức dữ liệu trên PC sẽ bị xoá hết).

**Malicious Mobile** Code là một dạng mã phần mềm có thể được gửi từ xa vào để chạy trên một hệ thống mà không cần đến lời gọi thực hiện của người dùng hệ thống đó. Malicious Mobile Code được coi là khác với virus, worm ở đặc tính là nó không nhiễm vào file và không tìm cách tự phát tán. Thay vì khai thác một điểm yếu bảo mật xác định nào đó, kiểu tấn công này thường tác động đến hệ thống bằng cách tận dụng các quyền ưu tiên ngầm định để chạy mã từ xa. Các công cụ lập trình như Java, ActiveX, JavaScript, VBScript là môi trường tốt cho malicious mobile code. Một trong những ví dụ nổi tiếng của kiểu tấn công này là Nimda, sử dụng JavaScript. [3]

**Virus Hoax** là các cảnh báo giả về virus. Các cảnh bảo giả này thường núp dưới dạng một yêu cầu khẩn cấp để bảo vệ hệ thống. Mục tiêu của cảnh báo virus giả là cố gắng lôi kéo mọi người gửi cảnh báo càng nhiều càng tốt qua email. Bản thân cảnh báo giả là không gây nguy hiểm trực tiếp nhưng những thư gửi để cảnh báo có thể chữa mã độc hại hoặc trong cảnh báo giả có chứa các chỉ dẫn về thiết lập lại hệ điều hành, xoá file làm nguy hại tới hệ thống. Kiểu cảnh báo giả này cũng gây tốn thời gian và quấy rối bộ phận hỗ trợ kỹ thuật khi có quá nhiều người gọi đến và yêu cầu dịch vụ. [1] [3]

**Tracking Cookie** là một dạng lạm dụng cookie để theo dõi một số hành động duyệt web của người sử dụng một cách bất hợp pháp. Cookie là một file dữ liệu chứa thông tin về việc sử dụng một trang web cụ thể nào đó của web-client. Mục tiêu của việc duy trì các cookie trong hệ thống máy tính nhằm căn cứ vào đó để tạo ra giao diện, hành vi của trang web sao cho thích hợp và tương ứng với từng web-client. Tuy nhiên tính năng này lại bị lạm dụng để tạo thành các phần mềm gián điệp (spyware) nhằm thu thập thông tin riêng tư về hành vi duyệt web của cá nhân. [1] [3]

**Attacker Tool** là những bộ công cụ tấn công có thể sử dụng để đẩy các phần mềm độc hại vào trong hệ thống. Các bộ công cụ này có khả năng giúp cho kẻ tấn công có thể truy nhập bất hợp pháp vào hệ thống hoặc làm cho hệ thống bị lây nhiễm mã độc hại. Khi được tải vào trong hệ thống bằng các đoạn mã độc hai, attacker tool có thể chính là một phần của đoạn mã độc đó (ví dụ như trong một trojan) hoặc nó sẽ được tải vào hệ thống sau khi nhiễm. Ví dụ như một hệ thống đã bị nhiễm một loại worm, worm này có thể điều khiển hệ thống tự động kết nối đến một web-site nào đó, tải attacker tool từ site đó và cài đặt attacker tool vào hệ thống. Attacker tool thường gặp là backdoor và keylogger. [1] [3]

**Rootkits** là tập hợp của các file được cài đặt lên hệ thống nhằm biến đổi các chức năng chuẩn của hệ thống thành các chức năng tiềm ẩn các tấn công nguy hiểm. Ví dụ như trong hệ thống Windows, rootkit có thể sửa đổi, thay thế file, hoặc thường trú trong bộ nhớ nhằm thay thế, sửa đổi các lời gọi hàm của hệ điều hành. Rootkit thường được dùng để cài đặt các công cụ tấn công như cài backdoor, cài keylogger. Ví dụ về rootkit là: LRK5, Knark, Adore, Hack Defender. [3]

**Web Browser Plug-in** là phương thức cài mã độc hại thực thi cùng với trình duyệt web. Khi được cài đặt, kiểu mã độc hại này sẽ theo dõi tất cả các hành vi duyệt web của người dùng (ví dụ như tên web site đã truy nhập) sau đó gửi thông tin ra ngoài. Một dạng khác là phần mềm gián điệp có chức năng quay số điện thoại tự động, nó sẽ tự động kích hoạt modem và kết nối đến một số điện thoại ngầm định mặc dù không được phép của chủ nhân. [1] [3]

**Phishing** là một hình thức tấn công thường có thể xem là kết hợp với mã độc hại. Phishing là phương thức dụ người dùng kết nối và sử dụng một hệ thống máy tính giả mạo nhằm làm cho người dùng tiết lộ các thông tin bí mật về danh tính (ví dụ như mật khẩu, số tài khoản, thông tin cá nhân,...). Kẻ tấn công phishing thường tạo ra trang web hoặc email có hình thức giống hệt như các trang web hoặc email mà nạn nhân thường hay sử dụng như trang của Ngân hàng, của công ty phát hành thẻ tín dụng,... Email hoặc trang web giả mạo này sẽ đề nghị nạn nhân thay đổi hoặc cung cấp các thông tin bí mật về tài khoản, về mật khẩu,... Các thông tin này sẽ được sử dụng để trộm tiền trực tiếp trong tài khoản hoặc được sử dụng vào các mục đích bất hợp pháp khác. [1]

**Email Generator** là những chương trình cho phép tạo ra và gửi đi một số lượng lớn các email. Mã độc hại có thể gieo rắc các email generator vào trong hệ thống. Các chương trình gián điệp, spam, mã độc hại có thể được đính kèm vào các email được sinh là từ email generator và gửi tới các địa chỉ có trong sổ địa chỉ của máy bị nhiễm. Email Generator là những chương trình cho phép tạo ra và gửi đi một số lượng lớn các email. Mã độc hại có thể gieo rắc các email generator vào trong hệ thống. Các chương trình gián điệp, spam, mã độc hại có thể được đính kèm vào các email được sinh là từ email generator và gửi tới các địa chỉ có trong sổ địa chỉ của máy bị nhiễm. [1]

**Attacker Toolkit** là các bộ công cụ có thể được tải xuống và cài vào hệ thống khi hệ thống đã bị khống chế bởi phần mềm độc hại. Các công cụ kiểu như các bộ dò quét cổng (port scanner), bộ phá mật khẩu (password cracker), bộ dò quét gói tin (Packet Sniffer) chính là các Attacker Toolkit thường hay được sử dụng.

## Hình thức lây nhiễm

Lây nhiễm là đặc điểm cơ bản của virus máy tính. Chúng ta cần tìm hiểu cách mà virus lây lan như thế nào. Virus lây nhiểm qua ba cách sau:

* Cách cổ điển
* Qua e-mail
* Qua mạng Internet



Hình ‑. Các hình thức lây nhiễm

### Cách cổ điển

Cách cổ điển nhất của sự lây nhiễm 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 ổ cứng di động hoặc các thiết bị giải trí kỹ thuật số.

### Qua e-mail

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:

Hình ‑. Lây nhiễm qua E-Mail

* 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.

### Qua mạng Internet

Ngày nay Internet phát triển mạnh mẽ trên toàn thế giới nên các hình thức lây nhiễm virus qua Internet trở thành phương pháp lây nhiễm chủ yếu của virus.

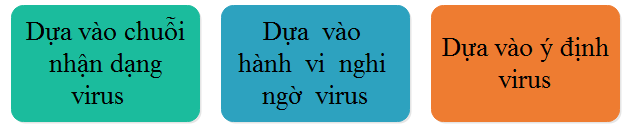
Có các hình thức lây nhiễm virus và phần mềm độc hại thông qua Internet như sau:

Hình ‑. Lây nhiễm qua Internet

* 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 chuẩn đoán

Có ba kỹ thuật nhận dạng virus máy tính đã được áp dụng:



Hình ‑. Cơ chế chuẩn đoán

### 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.

# HỆ MIỄN DỊCH NHÂN TẠO

## Khái niệm

Hệ miễn dịch nhân tạo (Artificial Immune System - AIS) là 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. [4]

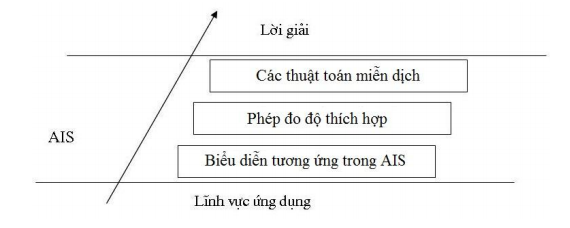
Có rất nhiều định nghĩa về hệ miễn dịch nhân tạo nhưng có thể đưa ra bốn định nghĩa cơ bản sau:

* **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)

## Cấu trúc

**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 ‑. 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

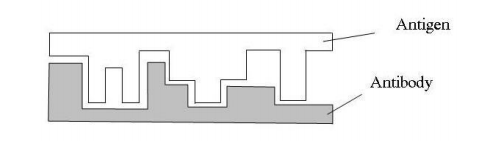
### Tổng quát hoá các thành phần sinh học của hệ miễn dịch

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. [5] [6]



Hình ‑. 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... [4] [5] [6] Trong phạm vi của đề tài, nhóm thực hiện tập trung vào hai thuật toán được trình bày sau đây.

Hình ‑. Thuật toán miễn dịch nhân tạo

### Thuật toán chọn lọc âm tính

Phép chọn lọc tích cực đối với các lympho bào (T-cell và B-cell) nhằm mục đích tránh sự góp mặt của những lympho bào không có ích. Những lympho bào mà cơ quan thụ cảm chúng không có khả năng nhận diện được kháng nguyên. Kết quả cuối cùng của phép chọn lọc tích cực là những lympho bào tồn tại qua phép chọn sẽ là những lympho bào có khả năng nhận diện được kháng nguyên.

Tất cả T-cell có cơ quan thụ cảm nhận diện được một phần của kháng nguyên (peptide), mà peptide này đã kết hợp với MHC tạo thành phức chất MHC/peptide, cần được chọn vào quần thể có giá trị, và giải phóng khỏi tuyến ức để nó thực hiện nhiệm vụ miễn dịch của mình. Quá trình chọn lọc tích cực đảm bảo chỉ những T- cell mà cơ quan thụ cảm của nó có thể nhận diện kháng nguyên (nhận diện được phức hợp MHC/ peptide) mới được đi ra khỏi tuyến ức, lưu thông trong cơ thể. Quá trình này xảy ra trong tuyến ức, do đó nó được gọi là phép chọn lọc tích cực trung tâm, hay phép chọn lọc tích cực trong tuyến ức. Ở quá trình chọn lọc tích cực này độ thích hợp của kháng thể và kháng nguyên không đủ lớn để tạo ra quá trình chọn lọc Clone (clonal selection) nó chỉ là tiền đề cho quá trình chọn lọc nhân bản.

Trong quá trình sản sinh B-cell, nhiều B-cell sau khi đột biến cơ quan thụ cảm của nó không thể kết hợp được với kháng nguyên sẽ bị loại bỏ. Một số B-cell có cơ quan thụ cảm (BCR hay Antibody) sau khi đột biến kết hợp được với kháng nguyên sẽ được duy trì, một số B-cell tốt (B-cell có độ thích hợp cao) sẽ trở thành B-cell ghi nhớ sống lâu trong cơ thể.

Ý tưởng của thuật toán khá đơn giản, đó là chọn ra những T-cell có thể nhận diện các peptide (một phần của kháng nguyên xâm nhập vào cơ thể), mà peptide này đã kết hợp với MHC để tạo thành MHC/peptide. Giả thiết có tập MHC/peptide, được gọi là tập S, các cơ quan thụ cảm T-cell sẽ phải được kiểm tra về khả năng kết hợp với các phần tử thuộc tập S này. Nếu một T-cell không nhận diện được bất kì phần tử nào, nó sẽ bị loại bỏ. Trái lại nó được chọn như một tế bào có khả năng miễn dịch và bổ sung vào quần thể A.

Thuật toán chọn lọc tích cực được minh họa như trong hình 4.3 và nó có thể tóm tắt như sau:

**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 được biểu diễn như các chuỗi nhị phân có cùng độ dài L, 2L phần tử khác nhau được tạo ra.

**Bước 2.** **Đánh giá độ thích hợp**: Xác định độ thích hợp của một tất cả phần tử trong quần thể P với tất cả phần tử trong tập S.

**Bước 3.** **Tạo một quần thể có giá trị**: Nếu độ thích hợp của một phần tử trong P với một phần tử trong S lớn hơn hoặc bằng một ngưỡng tương tác chéo ε nào đó thì T-cell có khả năng nhận diện kháng nguyên, sẽ được chọn vào quần thể giá trị A trái lại T-cell bị loại bỏ.



Hình ‑. Sơ đồ thuật toán chọn lọc tích cực

### 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.

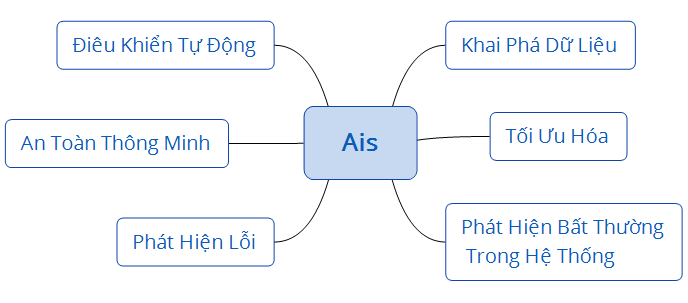
Mô hình thuật toán:



Hình ‑. Mô hình thuật toán chọn lọc âm tính.

## 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.



Hình ‑. Các ứng dụng của hệ AIS

### 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).

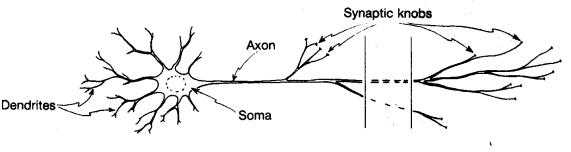
# MẠNG NƠ-RON NHÂN TẠO

## Tiếp cận mạng nơ-ron sinh học

Hệ thống bộ não con người gồm khoảng 100 tỷ tế bào thần kinh, thường được gọi là nơ-ron. Mỗi nơ-ron hoạt động như một bộ xử lý đơn giản. Mỗi tế bào nơron gồm ba phần:

* Thân nơ-ron với nhân bên trong (gọi là soma), là nơi tiếp nhận hay phát ra các xung động thần kinh.
* Một hệ thống dạng cây các dây thần kinh vào (gọi là dendrite) để đưa tín hiệu tới nhân nơ-ron. Các dây thần kinh vào tạo thành một lưới dày đặc xung quanh thân nơ-ron, chiếm diện tích khoảng 0,25 mm2.
* Đầu dây thần kinh ra (gọi là sợi trục axon) phân nhánh dạng hình cây, có thể dài từ một cm đến hàng mét. Chúng nối với các dây thần kinh vào hoặc trực tiếp với nhân tế bào của các nơ-ron khác thông qua các khớp nối (gọi là synapse). Thông thường mỗi nơ-ron có thể có từ vài chục cho tới hàng trăm ngàn khớp nối để nối với các nơ-ron khác. Có hai loại khớp nối, khớp nối kích thích (excitatory) sẽ cho tín hiệu qua nó để tới nơ-ron còn khớp nối ức chế (inhibitory) có tác dụng làm cản tín hiệu tới nơ-ron. Người ta ước tính mỗi nơ-ron trong bộ não của con người có khoảng 104 khớp nối.

Chức năng cơ bản của các tế bào nơ-ron là liên kết với nhau để tạo nên hệ thống thần kinh điều khiển hoạt động của cơ thể sống. Các tế bào nơ-ron truyền tín hiệu cho nhau thông qua các dây thần kinh vào và ra, các tín hiệu đó có dạng xung điện và được tạo ra từ các quá trình phản ứng hoá học phức tạp. Tại nhân tế bào, khi điện thế của tín hiệu vào đạt tới một ngưỡng nào đó thì nó sẽ tạo ra một xung điện dẫn tới trục dây thần kinh ra. Xung này truyền theo trục ra tới các nhánh rẽ và tiếp tục truyền tới các nơ-ron khác.

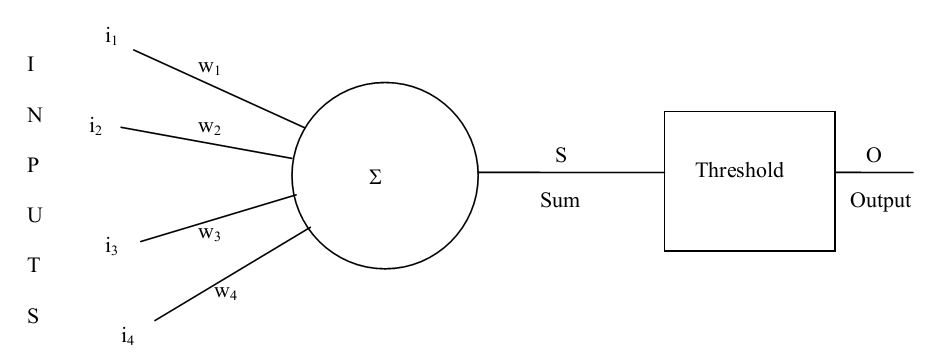


Hình ‑. Cấu tạo nơ-ron sinh học.

## Mạng nơ-ron nhân tạo

### Cấu trúc 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 ‑. 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.

 Threshold set at 0.5

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)=

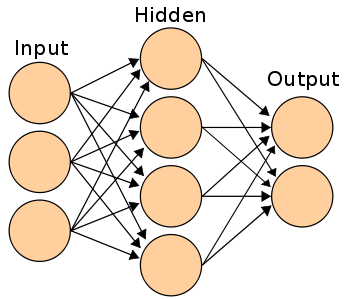
* Hàm sigmoid (logistic)

f (u) =

* Hàm tang- hyperbol

f (u) = tang (u) =

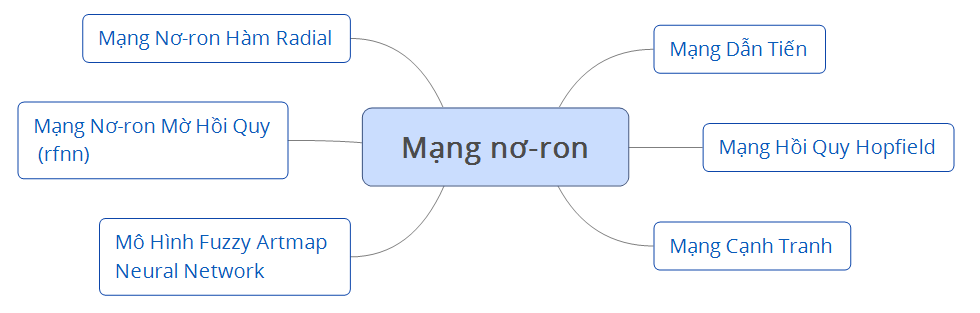
### Cấu trúc mạng nơ-ron nhân tạo



Hình ‑. Cấu trúc mạng noron nhân tạo

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.



Hình ‑. Một số mạng nơ-ron nhân tạo được sử dụng phổ biến hiện nay

## Tìm hiểu mạng nơ-ron Kohonen

### Giới thiệu

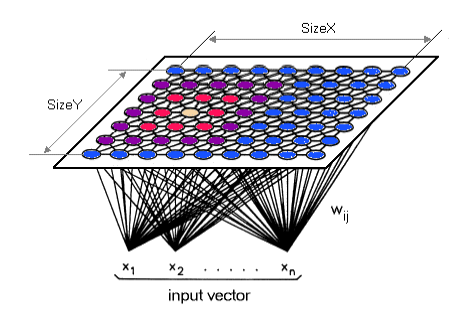
Ý tưởng về SOM lần đầu tiên được giới thiệu bởi C.Von der Malsburg năm 1973 và được phát triển bởi T.Kohonen (Finland) năm 1982 nên còn được gọi là Kohonen Network, nó giúp giảm số chiều (kích thước) của dữ liệu, qua đó giúp cho chúng ta hiểu được dữ liệu với kích thước lớn đó. Điểm nổi bật của thuật toán SOM là nó sẽ chuyển các dạng quan hệ phức tạp, thống kê rời rạc giữa các đối tượng dữ liệu đa chiều (n chiều) về các quan hệ hình học trực quan đơn giản với số chiều được rút gọn đi nhiều lần - thường là một hoặc hai chiều. Về bản chất giải thuật được biết đến như là kỹ thuật nén dữ liệu dựa trên véc tơ trọng số [6].

### Cầu trúc của mạng nơ-ron Kohonen

Mạng nơ-ron Kohonen chỉ bao gồm một lớp dữ liệu đầu vào và một lớp dữ liệu đầu ra của các nơ-ron và nó không chưa lớp ẩn.Lớp dữ liệu đầu vào đối với mạng nơ-ron Kohonen là các nơ-ron đầu vào. Các nơ-ron đầu vào này tạo thành mẫu dữ liệu đầu vào của mạng, các nơ-ron của lớp ra được sắp xếp trên một mảng 2 chiều, mảng này được gọi là lớp ra Kohonen. Tất cả các nơ-ron của lớp vào đều được nối với các nơ-ron trên lớp ra Kohonen. Mỗi liên kết giữa đầu vào và đầu ra của mạng Kohonen tương ứng với một trọng số. Kích thước của mỗi vetor trọng số bằng kích thước của lớp vào.

SOM gồm các thành phần sau:

* Output Layer**:** Gồm các node (nơ-ron) được bố trí trên một lưới (bản đồ) kích thước (X x Y). Mỗi nơ-ron có vị trí xác định trên lưới, tại mỗi neuron lưu giữ một vector trọng số (weight vector) có số chiều bằng với số chiều của input vector.
* Input vector**:** Là các training sample có kích thước n.
* Ma trận trọng số (weight matrix) wijkết nối giữa input vector và các nơ-ron.



Hình 4‑3. Cấu trúc mạng Kohone

### Bài toán phân cụm và mạng Kohonen

## Một vài ứng dụng của SOM

Thuật toán SOM đã được sử dụng trong nhiều lĩnh vực khác nhau với trên 5000 ứng dụng, SOM đã khẳng định được các ưu điểm sau:

SOM rất có hiệu quả trong quá trình phân tích đòi hỏi trí thông minh để đưa ra quyết định nhanh chóng trên thị trường. Nó giúp cho người phân tích hiểu vấn đề hơn trên một tập dữ liệu tương đối lớn.

Có khả năng biểu diễn dữ liệu đa chiều dùng trong trình bày và làm báo cáo.

Xác định các cụm dữ liệu (ví dụ các nhóm khách hàng) giúp cho việc tối ưu phân bố nguồn lực (quảng cáo, tìm kiếm sản phẩm,...).

Có thể dùng để phát hiện sự gian lận trong thẻ tín dụng, và các lỗi dữ liệu.

# DATA MINING VÀ MACHINE LEARNING

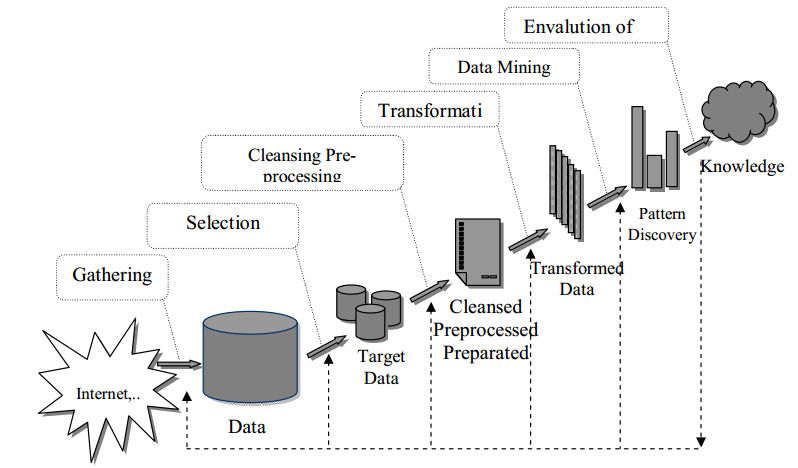
## Data Mining

### Tổng quan

#### Quá trình khám phá tri thức

Khám phá tri thức là quá trình tìm ra những tri thức, đó là những mẫu tìm ẩn,  
trước đó chưa biết và là thông tin hữu ích đáng tin cậy.

Quá trình khám phá tri thức từ CSDL là một quá trình có sử dụng nhiều phương pháp và công cụ tin học nhưng vẫn là một quá trình mà trong đó con người là trung tâm. Do đó, nó không phải là một hệ thống phân tích tự động mà là một hệ thống bao gồm nhiều hoạt động tương tác thường xuyên giữa con người và CSDL, tất nhiên là với sự hỗ trợ của các công cụ tin học [7]. Người sử dụng hệ thống ở đây phải là người có kiến thức cơ bản về lĩnh vực cần phát hiện tri thức để có thể chọn được đúng các tập con dữ liệu, các lớp mẫu phù hợp và đạt tiêu chuẩn quan tâm so với mục đích. Tri thức mà ta nói ở đây là các tri thức rút ra từ các CSDL, thường để phục vụ cho việc giải quyết một loạt nhiệm vụ nhất định trong một lĩnh vực nhất định. Do đó, quá trình phát hiện tri thức cũng mang tính chất hướng nhiệm vụ, không phải là phát hiện mọi tri thức bất kỳ mà là phát hiện tri thức nhằm giải quyết tốt nhiệm vụ đề ra.



Hình ‑. Các bước trong KDD

##### Gom dữ liệu (Gathering)

Tập hợp dữ liệu là bước đầu tiên trong quá trình khai phá dữ liệu. Đây là bước được khai thác trong một CSDL, một kho dữ liệu và thậm chí các dữ liệu từ các nguồn ứng dụng Web.

##### Trích lọc dữ liệu (Selection)

Ở giai đoạn này lựa chọn những dữ liệu phù hợp với nhiệm vụ phân tích trích rút từ CSDL.

##### Làm sạch, tiền xử lý và chuẩn bị trước dữ liệu (Cleansing, Pre-processing and Preparation)

Giai đoạn thứ ba này là giai đoạn hay bị sao lãng, nhưng thực tế nó là một bước rất quan trọng trong quá trình khai phá dữ liệu. Một số lỗi thường mắc phải trong khi gom dữ liệu là tính không đủ chặt chẽ, logic. Vì vậy, dữ liệu thường chứa các giá trị vô nghĩa và không có khả năng kết nối dữ liệu, ví dụ: điểm = -1. Giai đoạn này sẽ tiến hành xử lý những dạng dữ liệu không chặt chẽ nói trên. Những dữ liệu dạng này được xem như thông tin dư thừa, không có giá trị. Bởi vậy, đây là một quá trình rất quan trọng vì dữ liệu này nếu không được “làm sạch” sẽ gây nên những kết quả sai lệch nghiêm trọng.

##### Chuyển đổi dữ liệu (Transformation)

Tiếp theo là giai đoạn chuyển đổi dữ liệu, dữ liệu được chuyển đổi hay được hợp nhất về dạng thích hợp cho việc khai phá.

##### Khai phá dữ liệu (Data Mining)

Đây là một tiến trình cốt yếu. Ở giai đoạn này nhiều thuật toán khác nhau đã được sử dụng một cách phù hợp để trích xuất thông tin có ích hoặc cá mẫu điển hình trong dữ liệu.

##### Đánh giá kết quả mẫu (Evaluation of Result)

Đây là giai đoạn cuối trong quá trình khai phá dữ liệu. Ở giai đoạn này, các mẫu dữ liệu được chiết xuất, không phải bất cứ mẫu dữ liệu nào cũng đều hữu ích, đôi khi nó còn bị sai lệch. Vì vậy, cần phải ưu tiên những tiêu chuẩn đánh giá để chiết xuất ra các tri thức cần thiết. Từ quá trình khám phá tri thức trên chúng ta thấy được sự khác biệt giữa khám phá tri thức và khai phá dữ liệu. Trong khi khám phá tri thức là nói đến quá trình tổng thể phát hiện tri thức hữu ích từ dữ liệu. Còn KPDL chỉ là một bước trong quá trình khám phá tri thức, các công việc chủ yếu là xác định được bài toán khai phá, tiến hành lựa chọn phương pháp KPDL phù hợp với dữ liệu có được và tách ra các tri thức cần thiết.

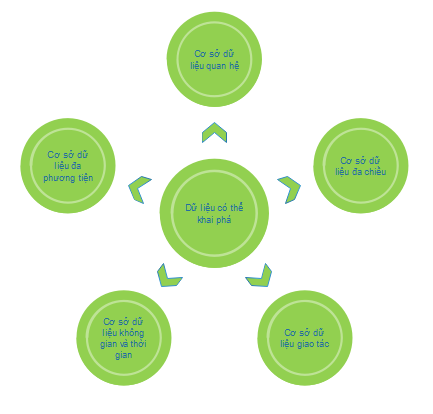
#### Khai phá dữ liệu

Khai phá dữ liệu (*Data Mining*) là một bước quan trọng trong quá trình khám phá tri thức.

Khai phá dữ liệu được định nghĩa như là một quá trình chắt lọc hay khai phá tri thức từ một lượng lớn dữ liệu. Một ví dụ hay được sử dụng là là việc khai thác vàng từ đá và cát, Dataming được ví như công việc "Đãi cát tìm vàng" trong một tập hợp lớn các dữ liệu cho trước. Thuật ngữ Dataming ám chỉ việc tìm kiếm một tập hợp nhỏ có giá trị từ một số lượng lớn các dữ liệu thô. Có nhiều thuật ngữ hiện được dùng cũng có nghĩa tương tự với từ Datamining như Knowledge Mining (khai phá tri thức), knowledge extraction (chắt lọc tri thức), data/patern analysis (phân tích dữ liệu/mẫu), data archaeoloogy (khảo cổ dữ liệu), data dredging (nạo vét dữ liệu),...

**Định nghĩa:** Khai phá dữ liệu là một tập hợp các kỹ thuật được sử dụng để tự động khai thác và tìm ra các mối quan hệ lẫn nhau của dữ liệu trong một tập hợp dữ liệu khổng lồ và phức tạp, đồng thời cũng tìm ra các mẫu tiềm ẩn trong tập dữ liệu đó. [7]

### Các loại dữ liệu có thể khai phá



Hình ‑. Khai phá dữ liệu – các loại dữ liệu

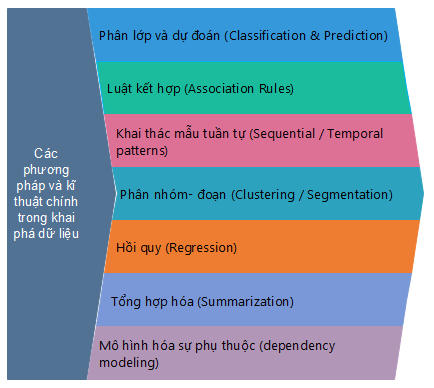
Các loại dữ liệu có thể được khai phá như sau:

* Cơ sở dữ liệu quan hệ (Relational databases): là những CSDL được tổ chức theo mô hình quan hệ. Hiện nay, các hệ quản trị CSDL đều hỗ trợ mô hình này như: MS Access, MS SQL Server, Oracle, IBM DB2,...
* Cơ sở dữ liệu đa chiều (Multidimention structures, data warehouse, data mart): còn được gọi là nhà kho dữ liệu, trong đó dữ liệu được chọn từ nhiều nguồn khác nhau và chứa những đặc tính lịch sử thông qua thuộc tính thời gian tường minh hoặc ngầm định.
* Cơ sở dữ liệu giao tác (Transaction databases): là loại dữ liệu được sử dụng nhiều trong siêu thị, thương mại, ngân hàng,...
* Cơ sở dữ liệu quan hệ – hướng đối tượng (Object relational databases): mô hình CSDL này là lai giữa mô hình hướng đối tượng và mô hình CSDL quan hệ.
* Cơ sở dữ liệu không gian và thời gian (Spatial, temporal, and time – series data): chứa những thông tin về không gian địa lý hoặc thông tin theo thời gian.
* Cơ sở dữ liệu đa phương tiện (Multimedia database): là loại dữ liệu có nhiều trên mạng, bao gồm các loại như âm thanh, hình ảnh, video, văn bản và nhiều kiểu dữ liệu định dạng khác.

### Các phương pháp, kỹ thuật chính trong khai phá dữ liệu

Các kỹ thuật KPDL được có thể chia làm 2 nhóm chính:

* *Kỹ thuật KPDL mô tả*: có nhiệm vụ mô tả về các tính chất hoặc các đặc tính chung của dữ liệu trong CSDL hiện có. Nhóm kỹ thuật này gồm các phương pháp: phân nhóm (*Clustering)*, tổng hợp hóa (*Summerization)*, phát hiện sự biến đổi và độ lệch (*Change* *and deviation detection*)*,* phân tích luật kết hợp (*Association Rules),...*
* *Kỹ thuật KPDL dự đoán:* có nhiệm vụ đưa ra các dự đoán dựa vào các suy diễn trên dữ liệu hiện thời. Nhóm kỹ thuật này gồm các phương pháp: phân lớp (*Classification),* hồi quy (*Regression),...*



Hình ‑. Khai phá dữ liệu - Phương pháp và kỹ thuật

#### Phân lớp và dự đoán (Classification & Prediction)

Là đặt các mẫu vào các lớp được xác định trước. Nhiệm vụ chính là tìm các hàm ánh xạ các mẫu dữ liệu một cách chính xác vào trong các lớp.Ví dụ một ngân hàng muốn phân loại các khách hành của họ vào trong hai nhóm có nợ hay không nợ, từ đó giúp họ ra quyết định cho vay hay không cho vay. Quá trình phân lớp dữ liệu thường gồm 2 bước: xây dựng mô hình và sử dụng mô hình để phân lớp dữ liệu.

***Bước 1*:** Một mô hình sẽ được xây dựng dựa trên việc phân tích các mẫu dữ liệu sẵn có. Mỗi mẫu tương ứng với một lớp, được quyết định bởi một thuộc tính gọi là thuộc tính lớp. Các mẫu dữ liệu này còn được gọi là *tập dữ liệu huấn luyện (training* *data set)*. Các nhãn lớp của tập dữ liệu huấn luyện đều phải được xác định trước khi xây dựng mô hình, vì vậy phương pháp này còn được gọi là *học có giám sát* *(supervised learning)* khác với phân nhóm dữ liệu là *học không có giám sát* *(unsupervised learning).*

***Bước 2*:** Sử dụng mô hình để phân lớp dữ liệu. Trước hết chúng ta phải tính độ chính xác của mô hình. Nếu độ chính xác là chấp nhận được, mô hình sẽ được sử dụng để dự đoán nhãn lớp cho các mẫu dữ liệu khác trong tương lai.

Trong kỹ thuật phân lớp chúng ta có thể sử dụng các phương pháp như: Cây quyết định (*Decision Tree*), K-Láng giềng gần nhất (*k-Nearest Neighbor*), Mạng Nơron (*Neural networks*), Giải thuật di truyền (*Genetic algorithms*), Mạng Bayesian (*Bayesian networks*), Tập mờ và tập thô (*Rough and Fuzzy Sets*).

#### Luật kết hợp (Association Rules)

Luật kết hợp là dạng luật biểu diễn tri thức ở dạng tương đối đơn giản. Mục tiêu của phương pháp này là phát hiện và đưa ra các mối liên hệ giữa các giá trị dữ liệu trong CSDL. Mẫu đầu ra của giải thuật KPDL là tập luật kết hợp tìm được. Tuy luật kết hợp là một dạng luật khá đơn giản nhưng lại mang rất nhiều ý nghĩa. Thông tin mà dạng luật này đem lại rất có lợi trong các hệ hỗ trợ ra quyết định. Tìm kiếm được những luật kết hợp đặc trưng và mang nhiều thông tin từ CSDL tác nghiệp là một trong những hướng tiếp cận chính của lĩnh vực khai phá dữ liệu.

#### Khai thác mẫu tuần tự (Sequential / Temporal patterns)

Tương tự như khai thác luật kết hợp nhưng có thêm tính thứ tự và tính thời gian. Một luật mô tả mẫu tuần tự có dạng tiêu biểu X -> Y phản ánh sự xuất hiện của biến cố X sẽ dẫn đến việc xuất hiện kế tiếp biến cố Y. Hướng tiếp cận này có tính dự báo cáo.

#### Phân nhóm - đoạn (Clustering / Segmentation)

Mục tiêu chính của việc phân nhóm dữ liệu là nhóm các đối tượng tương tự nhau trong tập dữ liệu vào các *nhóm* sao cho mức độ tương tự giữa các đối tượng trong cùng một nhóm là lớn nhất và mức độ tương tự giữa các đối tượng nằm trong các nhóm khác nhau là nhỏ nhất. Các nhóm có thể tách nhau hoặc phân cấp gối lên nhau và số lượng các nhóm là chưa biết trước. Một đối tượng có thể vừa thuộc nhóm này, nhưng cũng có thể vừa thuộc nhóm khác. Không giống như phân lớp dữ liệu, phân nhóm dữ liệu không đòi hỏi phải định nghĩa trước các *mẫu dữ liệu huấn luyện*. Vì thế, có thể coi phân nhóm dữ liệu là một cách học bằng quan sát (*learning by observation)*, trong khi phân lớp dữ liệu là học bằng ví dụ (*learning by example*). Trong phương pháp này bạn sẽ không thể biết kết quả các nhóm thu được sẽ như thế nào khi bắt đầu quá trình.

Vì vậy, thông thường cần có một chuyên gia về lĩnh vực đó để đánh giá các nhóm thu được. Phân nhóm còn được gọi là học không có giám sát (unsupervised learning). Phân nhóm dữ liệu được sử dụng nhiều trong các ứng dụng về phân đoạn thị trường, phân đoạn khách hàng, nhận dạng mẫu, phân loại trang Web, …

Ngoài ra phân nhóm dữ liệu còn có thể được sử dụng như một bước tiền xử lý cho các thuật toán KPDL khác.

#### Hồi quy (Regression)

Hồi quy là việc học một hàm ánh xạ từ một mẫu dữ liệu thành một biến dự đoán có giá trị thực. Nhiệm vụ của hồi quy tương tự như phân lớp, điểm khác nhau chính là ở chỗ thuộc tính để dự báo là liên tục chứ không rời rạc. Việc dự báo các giá trị số thường được làm bởi các phương pháp thống kê cổ điển chẳng hạn như hồi quy tuyến tính. Tuy nhiên phương pháp mô hình hóa cũng có thể được sử dụng như cây quyết định.

#### Tổng hợp hóa (Summarization)

Là công việc liên quan đến các phương pháp tìm kiếm một mô tả tập con dữ liệu. Kỹ thuật mô tả khái niệm và tổng hợp hóa thường áp dụng trong việc phân tích dữ liệu có tính thăm dò và báo cáo tự động. Nhiệm vụ chính là sản sinh ra các mô tả đặc trưng cho một lớp. Mô tả loại này là một kiểu tổng hợp, tóm tắt các đặc tính chung của tất cả hay hầu hết các mục của một lớp. Các mô tả đặc trưng thể hiện theo luật có dạng sau: “*Nếu một mục thuộc về lớp đã chỉ trong tiền đề thì mục đó có tất cả các thuộc tính đã nêu trong kết luận*”.

#### Mô hình hóa sự phụ thuộc (dependency modeling)

Là việc tìm kiếm một mô hình mô tả sự phụ thuộc giữa các biến, thuộc tính theo hai mức. Mức cấu trúc của mô hình mô tả (thường dưới dạng đồ thị), trong đó, các biến phụ thuộc bộ phận vào các biến khác. Và mức định lượng mô hình mô tả mức độ phụ thuộc. Những phụ thuộc này thường được biểu thị dưới dạng theo luật “nếu - thì” - nếu tiền đề đúng thì kết luận đúng. Về nguyên tắc, cả tiền đề và kết luận đều có thể là sự kết hợp logic của các giá trị thuộc tính. Trên thực tế, tiền đề thường là nhóm các giá trị thuộc tính và kết luận chỉ là một thuộc tính. Hơn nữa, hệ thống có thể phát hiện các luật phân lớp trong đó tất cả các luật cần phải có cùng một thuộc tính do người dùng chỉ ra trong kết luận.

Quan hệ phụ thuộc cũng có thể biểu diễn dưới dạng mạng tin cậy Bayes. Đó là đồ thị có hướng, không chu trình. Các nút biểu diễn thuộc tính và trọng số của liên kết phụ thuộc giữa các nút đó.

#### Phát hiện sự biến đổi và độ lệch (Change and deviation detection)

Tập trung vào khám phá hầu hết sự thay đổi có nghĩa dưới dạng độ đo đã biết  
trước hoặc giá trị chuẩn, phát hiện độ lệch đáng kể giữa nội dung của tập con dữ liệu thực và nội dung mong đợi. Hai mô hình độ lệch hay dùng là lệch theo thời gian và lệch theo nhóm. Độ lệch theo thời gian là sự thay đổi có ý nghĩa của dữ liệu thời gian. Độ lệch theo nhóm là sự khác nhau của dữ liệu trong hai tập con dữ liệu, ở đây xét cả trường hợp tập con dữ liệu này thuộc tập con kia. Nghĩa xác định dữ liệu trong một nhóm con của đối tượng có khác đáng kể so với toàn bộ đối tượng không? Theo cách này, sai sót dữ liệu hay sai lệch so với giá trị thông thường sẽ được phát hiện.

### Ứng dụng của khai phá dữ liệu

Khai phá dữ liệu có nhiều ứng dụng trong thực tế, một số ứng dụng điển hình như:

*Bảo hiểm, tài chính và thị trường chứng khoán*: phân tích tình hình tài chính và dự báo giá của các loại cổ phiếu trong thị trường chứng khoán. Danh mục vốn và giá, lãi suất, dữ liệu thẻ tín dụng, phát hiện gian lận,…

*Điều trị y học và chăm sóc y tế*: một số thông tin về chẩn đoán bệnh lưu trong các hệ thống quản lý bệnh viện. Phân tích mối liên hệ giữa triệu chứng bệnh, chẩn đoán và phương pháp điều trị (chế độ dinh dưỡng, thuốc,…).

*Sản xuất và chế biến*: qui trình, phương pháp chế biến và xử lý xử cố.

*Text mining & Web mining*: phân lớp văn bản và các trang web, tóm tắt văn bản,...

*Lĩnh vực khoa học*: quan sát thiên văn, dữ liệu gene, dữ liệu sinh vật học, tìm kiếm, so sánh các hệ gene và thông tin di truyền, mối liên hệ gene và các bệnh di truyền,…

*Lĩnh vực khác*: viễn thông, môi trường, thể thao, âm nhạc, giáo dục,…

### Những thách thức trong khai phá dữ liệu

Mức độ nhiễu cao trong dữ liệu của KPDL. Tiêu chuẩn mạnh mẽ của giải thuật đối với nhiễu trở nên quan trọng hơn trong khi các tiêu chuẩn khác phần nào có thể giảm bớt.

Kích thước lớn của các tập dữ liệu cần xử lý. Các tập dữ liệu trong KPDL thường có kích thước cực kỳ lớn. Trong thực tế, kích thước của các tập dữ liệu trong KPDL thường ở mức tera-byte. Với kích thước như thế, thời gian xử lý thường cực kỳ dài. Thêm vào đó, các giải thuật học truyền thống thường yêu cầu tập dữ liệu được tải toàn bộ lên trên bộ nhớ để xử lý. Mặc dù kích thước bộ nhớ trong của máy tính đã gia tăng đáng kể trong thời gian gần đây, việc gia tăng này cũng không thể đáp ứng kịp với việc tăng kích thước dữ liệu. Vì vậy, việc vận dụng các kỹ thuật xác suất, lấy mẫu, đệm, song song và tăng dần vào các giải thuật để tạo ra các phiên bản phù hợp với yêu cầu của KPDL trở nên ngày càng quan trọng.

Các kỹ thuật trong KPDL là hướng tác vụ và hướng dữ liệu. Thay vì tập trung vào xử lý tri thức dạng kí hiệu và khái niệm như trong máy học, mọi phát triển trong KPDL thì kết chặt vào các ứng dụng thực tế và đặc tính dữ liệu cụ thể trong các ứng dụng đó. Ví dụ, luật kết hợp là kỹ thuật KPDL nhằm tìm kiếm những mối liên kết giữa các món hàng trong các hóa đơn ở siêu thị. Giải thuật học trong kỹ thuật này được phát triển dựa trên đặc tính về dữ liệu rất đặc thù là ở dạng nhị phân.

## Machine Learning

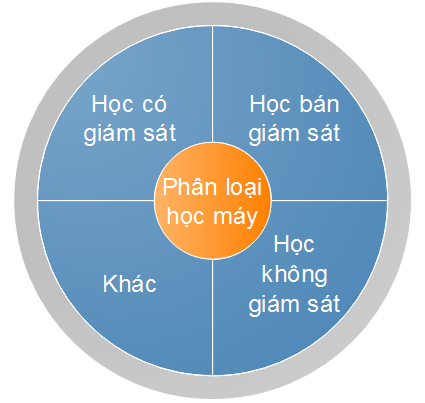
### Giới thiệu về máy học

Học máy (Machine Learning) là một ngành khoa học nghiên cứu các thuật toán cho phép máy tính có thể học được các khái niệm (concept). Cụ thể hơn, học máy là một phương pháp để tạo ra các chương trình máy tính bằng việc phân tích các tập dữ liệu. Học máy có liên quan mật thiết đến thống kê, vì cả hai lĩnh vực đều nghiên cứu việc phân tích dữ liệu. Tuy nhiên khác với thống kê, học máy tập trung vào sự phức tạp của các giải thuật trong việc thực thi tính toán.

### Phân loại

Trong lĩnh vực học máy có nhiều phương pháp học khác nhau, trong phần này giới thiệu về ba phương pháp học được sử dụng phổ biến nhất, gồm có:

* Học có giám sát (supervised learning).
* Học bán/ nửa giám sát (semi-supervised learning).
* Học không giám sát (unsupervised learning).



Hình ‑. Phân loại máy học

#### Học có giám sát

Học có giám sát là học với tập dữ liệu huấn luyện ban đầu hoàn toàn được gán nhãn từ trước. Học có giám sát là phương pháp học sử dụng cho lớp bài toán phân lớp, phân loại (Classification).

*Học có giám sát* là một kỹ thuật của ngành học máy để xây dựng một hàm từ dữ liệu huấn luyện. Dữ liệu huấn luyện bao gồm các cặp đối tượng đầu vào (thường dạng vector) và đầu ra thực sự. Đầu ra của một hàm có thể là một giá trị liên tục (gọi là hồi quy), hay có thể là dự đoán một nhãn phân lớp cho một đối tượng đầu vào (gọi là phân lớp). Nhiệm vụ của chương trình học có giám sát là dự đoán giá trị của hàm cho một đối tượng bất kỳ là đầu vào hợp lệ, sau khi đã xem xét một số ví dụ huấn luyện (nghĩa là, các cặp đầu vào và đầu ra tương ứng). Để đạt được điều này, chương trình học phải tổng quát hoá từ các dữ liệu sẵn có để dự đoán những tình huống chưa gặp phải theo một cách hợp lý.

Một số thuật toán thường được lựa chọn khi xây dựng bộ phân lớp gồm có:

* Máy vector hỗ trợ (Support Vector Machine – SVM).
* K láng giềng gần nhất (K Nearest Neighbours – KNN).
* Tiếp cận xác suất thống kê (Naïve Bayes – NB).
* Cây quyết định (Decision Tree – DT).
* Sử dụng mạng nơron (Neural Network – Nnet).
* Dựa trên vector trọng tâm (Centroid–base vector).
* Tuyến tính bình phương nhỏ nhất (Linear Least Square Fit – LLSF).

#### Học bán giám sát

Theo Xiaojin Zhu, khái niệm học bán giám sát được đưa ra năm 1970 khi bài toán đánh giá quy tắc Linear Discrimination Fisher cùng với dữ liệu chưa gán nhãn được nhiều sự quan tâm của các nhà khoa học trên thế giới.

Trong khoa học máy tính, *học bán giám sát* là một phương thức của ngành học máy sử dụng cả dữ liệu gán nhãn và chưa gán nhãn, nhiều nghiên cứu của ngành học máy có thể tìm ra được dữ liệu chưa gán nhãn khi sử dụng với một số lượng nhỏ dữ liệu gán nhãn. Công việc thu được kết quả của dữ liệu gán nhãn thường đòi hỏi ở trình độ tư duy và khả năng của con người, công việc này tốn nhiều thời gian và chi phí, do vậy dữ liệu gán nhãn thường rất hiếm và đắt, trong khi dữ liệu chưa gán nhãn thì lại rất phong phú. Trong trường hợp đó, chúng ta có thể sử dụng học bán giám sát để thi hành các công việc ở quy mô lớn.

Học bán giám sát bao gồm dữ liệu gán nhãn và chưa gán nhãn. Học bán giám sát có thể được áp dụng vào việc phân lớp và phân cụm. Mục tiêu của học bán giám sát là huấn luyện tập phân lớp tốt hơn học có giám sát từ dữ liệu gán nhãn và chưa gán nhãn.

Như vậy, có thể nói học bán giám sát là phương pháp học có giám sát kết hợp với việc tận dụng các dữ liệu chưa gán nhãn. Trong phần bổ sung thêm vào cho dữ liệu gán nhãn, thuật toán cung cấp một vài thông tin giám sát, việc này không cần thiết cho tất cả các mẫu huấn luyện. Thông thường thông tin này sẽ được kết hợp với một vài mâu cho trước.

Các dữ liệu gán nhãn thường hiếm, đắt và rất mất thời gian, đòi hỏi sự nỗ lực của con người, trong khi đó dữ liệu chưa gán nhãn thì vô vàn nhưng để sử dụng vào mục đích cụ thể của chúng ta thì rất khó, vì vậy ý tưởng kết hợp giữa dữ liệu chưa gán nhãn và dữ liệu đã gán nhãn để xây dựng một tập phân lớp tốt hơn là nội dung chính của học bán giám sát. Bởi vậy học bán giám sát là một ý tưởng tốt để giảm bớt công việc của con người và cải thiện độ chính xác lên mức cao hơn.

Một số thuật toán thường được sử dụng gồm có: thuật toán Cực đại kỳ vọng (EM - Expectation Maximization), SVM truyền dẫn (TSVM - Transductive Support Vector Machine), Self-training, Co-training và các phương pháp dựa trên đồ thị (graph-based).

#### Học không giám sát

Học không giám sát là học với tập dữ liệu huấn luyện ban đầu hoàn toàn chưa được gán nhãn. Học không giám sát là phương pháp học sử dụng cho lớp bài toán gom cụm, phân cụm (Clustering).

Có rất nhiều thuật toán học không giám sát được ra đời và phát triển nhằm giải quyết bài toán phân cụm phục vụ khai thác hiệu quả nguồn dữ liệu chưa gán nhãn nhiều và rất đa dạng. Việc lựa chọn sử dụng thuật toán nào tuỳ thuộc vào dữ liệu và mục đích của từng bài toán. Trong đó các thuật toán thường được sử dụng như: k-means, SOM (Self-Organizing Map), HAC (Hierarchical Agglomerative Clustering),DBSCAN, FCM,...

## Suport Vector Machine

### Phân lớp dữ liệu

### Bài toán phân lớp dữ liệu

Phân lớp (Classification) là một tiến trình xử lý nhằm xếp các mẫu dữ liệu hay các đối tượng vào một trong các lớp đã được định nghĩa trước. Các mẫu dữ liệu hay các đối tượng được xếp vào các lớp dựa vào giá trị của các thuộc tính (attributes) cho một mẫu dữ liệu hay đối tượng. Sau khi đã xếp tất cả các đối tượng đã biết trước vào các lớp tương ứng thì mỗi lớp được đặc trưng bởi tập các thuộc tính của các đối tượng chứa trong lớp đó.

Quá trình phân lớp còn được gọi là qui trình gán nhãn cho các tập dữ liệu. Nhiệm vụ của bài toán phân lớp dữ liệu là cần xây dựng mô hình (bộ) phân lớp để khi có một dữ liệu mới vào thì mô hình phân lớp sẽ cho biết dữ liệu đó thuộc lớp nào. Có nhiều cách để biểu diễn một mô hình phân lớp và có rất nhiều thuật toán giải quyết nó. Các thuật toán phân lớp tiêu biểu bao gồm như mạng neural, cây quyết định, suy luận quy nạp, mạng Beyesian, Support Vector Machine (SVM),… Trong các kỹ thuật đó, SVM được coi là công cụ mạnh, phổ biến và đặc biệt thích hợp cho phân lớp dữ liệu lớn và nhiều chiều.

Như vậy, nhiệm vụ của bài toán phân lớp dữ liệu là cần xây dựng mô hình (bộ) phân lớp để khi có một dữ liệu mới vào thì mô hình phân lớp sẽ cho biết dữ liệu đó thuộc lớp nào.

Có nhiều bài toán phân lớp dữ liệu, như phân lớp nhị phân, phân lớp đa lớp, phân lớp đa trị,…

Phân lớp nhị phân là quá trình tiến hành việc phân lớp dữ liệu vào một trong hai lớp khác nhau dựa vào việc dữ liệu đó có hay không một số đặc tính theo quy định của bộ phân lớp.

Phân lớp đa lớp là quá trình phân lớp với số lượng lớp lớn hơn hai. Như vậy, tập hợp dữ liệu trong miền xem xét được phân chia thành nhiều lớp chứ không đơn thuần chỉ là hai lớp như trong bài toán phân lớp nhị phân. Về bản chất, bài toán phân lớp nhị phân là trường hợp riêng của bài toán phân lớp đa lớp.

Trong phân lớp đa trị, mỗi đối tượng dữ liệu trong tập huấn luyện cũng như các đối tượng mới sau khi được phân lớp có thể thuộc vào từ hai lớp trở lên. Ví dụ như trang web về việc bùng phát bệnh cúm gia cầm, thủy cầm tại một số tính phía Bắc vừa thuộc về lĩnh vực y tế liên quan đến lây bệnh sang người nhưng cũng thuộc về lĩnh vực kinh tế lien quan đến ngành chăn nuôi… Trong những trường hợp như vậy, việc sắp xếp một tài liệu vào nhiều hơn một lớp là phù hợp với yêu cầu thực tế.

Sau đây chúng ta sẽ tìm hiểu khái quát về quá trình phân lớp dữ liệu và sơ bộ về phương pháp phân lớp dữ liệu.

### Quá trình phân lớp dữ liệu

Quá trình phân lớp dữ liệu thường gồm hai bước: xây dựng mô hình (tạo bộ phân lớp) và sử dụng mô hình đó để phân lớp dữ liệu.

* **Bước 1:** xây dựng mô hình
* Một mô hình sẽ được xây dựng dựa trên việc phân tích các đối tượng dữ liệu đã được gán nhãn từ trước. Tập các mẫu dữ liệu này còn được gọi là tập dữ liệu huấn luyện (training data set). Các nhãn lớp của tập dữ liệu huấn luyện được xác định bởi con người trước khi xây dựng mô hình, vì vậy phương pháp này còn được gọi là *học có giám sát (supervised learning).* Trong bước này, chúng ta còn phải tính độ chính xác của mô hình, mà cần phải sử dụng một tập dữ liệu kiểm tra (test data set). Nếu độ chính xác là chấp nhận được (tức là cao), mô hình sẽ được sử dụng để xác định nhãn lớp cho các dữ liệu khác mới trong tương lai.
* Tồn tại nhiều phương pháp phân lớp dữ liệu để giải quyết bài toán phân lớp tùy thuộc vào cách thức xây dựng mô hình phân lớp như phương pháp Bayes, phương pháp cây quyết định, phương pháp k-người láng giềng gần nhất, phương pháp máy hỗ trợ vector.... Các phương pháp phân lớp khác nhau chủ yếu về mô hình phân lớp. Mô hình phân lớp còn được gọi là thuật toán phân lớp.
* **Bước 2:** Sử dụng mô hình đã được xây dựng ở bước 1 để phân lớp dữ liệu mới.

## Sử dụng SVM

Bài toán phân lớp (*Classification*) và dự đoán (*Prediction*) là hai bài toán cơ bản và có rất nhiều ứng dụng trong tất cả các lĩnh vực như: học máy, nhận dạng, trí tuệ nhân tạo,... Trong khóa luận này, chúng em sẽ đi sâu nghiên cứu phương pháp Support Vector Machines (SVM), một phương pháp rất hiệu quả hiện nay.

### Giới thiệu

Phương pháp SVM được coi là công cụ mạnh cho những bài toán phân lớp phi tuyến tính được các tác giả Vapnik và Chervonenkis phát triển mạnh mẽ năm 1995. Phương pháp này thực hiện phân lớp dựa trên nguyên lý Cực tiểu hóa Rủi ro có Cấu trúc SRM (*Structural Risk Minimization*), được xem là một trong các phương pháp phân lớp giám sát không tham số tinh vi nhất cho đến nay. Các hàm công cụ đa dạng của SVM cho phép tạo không gian chuyên đổi để xây dựng mặt phẳng phân lớp [8].

SVM sử dụng thuật toán học nhằm xây dựng một siêu phẳng làm cực tiểu hoá độ phân lớp sai của một đối tượng dữ liệu mới. Độ phân lớp sai của một siêu phẳng được đặc trưng bởi khoảng cách bé nhất tới siêu phẳng đấy [8].

Đặc trưng cơ bản quyết định khả năng phân lớp là khả năng phân lớp những dữ liệu mới dựa vào những tri thức đã tích luỹ được trong quá trình huấn luyện. Sau quá trình huấn luyện nếu hiệu suất tổng quát hoá của bộ phân lớp cao thì thuật toán huấn luyện được đánh giá là tốt. Hiệu suất tổng quát hoá phụ thuộc vào hai tham số là *sai số* *huấn luyện* hay và *năng lực* của máy học. Trong đó sai số huấn luyện là tỷ lệ lỗi phân lớp trên tập dữ liệu huấn luyện. Còn năng lực của máy học được xác định bằng kích thước Vapnik-Chervonenkis (kích thước VC). Kích thước VC là một khái niệm quan trọng đối với một họ hàm phân tách (hay là tập phân lớp). Đại lượng này được xác định bằng số điểm cực đại mà họ hàm có thể phân tách hoàn toàn trong không gian đối tượng. Một tập phân lớp tốt là tập phân lớp có năng lực thấp nhất (có nghĩa là đơn giản nhất) và đảm bảo sai số huấn luyện nhỏ. Phương pháp SVM được xây dựng trên ý tưởng này.

### Ý tưởng của phương pháp

Cho trước một tập huấn luyện được biểu diễn trong không gian vector, trong đó mỗi tài liệu là một điểm, phương pháp này tìm ra một siêu phẳng quyết định tốt nhất có thể chia các điểm trên không gian này thành hai lớp riêng biệt tương ứng là lớp “+” và lớp “-”. Chất lượng của siêu phẳng này được quyết định bởi khoảng cách của điểm dữ liệu gần nhất của mỗi lớp đến mặt phẳng này. Khi đó khoảng cách biên càng lớn thì việc phân loại càng chính xác.

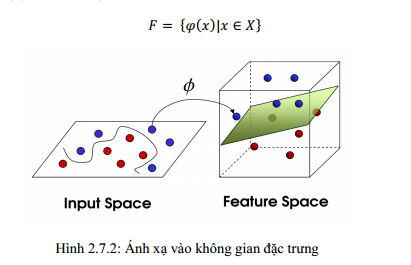
### Không gian đặc trưng

Sự hiệu quả của việc huấn luyện phụ thuộc nhiều vào cách dữ liệu được diễn tả. Khi dữ liệu được diễn tả phù hợp, việc huấn luyện sẽ dễ dàng và hiệu quả hơn. Khi dữ liệu được diễn tả phù hợp, việc huấn luyện sẽ dễ dàng và hiệu quả hơn. Vì vậy, việc chuyển đổi dữ liệu từ không gian đầu vào sang một không gian đặc trung là rất cần thiết và là một bước quan trọng giúp dữ liệu được phân cụm tốt hơn. Ánh xạ vector:



Trong đó n là số chiều của đầu vào (hay còn gọi là số thuộc tính) và N là số chiều của không gian đặc trưng. Dự liệu sẽ được chuyển vào không gian đặc trung với N > n.

Không gian đặc trưng kí hiệu là F:



Hình ‑. Ánh xạ vào không gian đặc trưng

### Cơ sở lý thuyết của phương pháp SVM

SVM thực chất là một bài toán tối ưu, mục tiêu của thuật toán này là tìm được một không gian F và siêu phẳng quyết định f’ trên F sao cho sai số phân loại là thấp nhất.

Cho tập mẫu  với , thuộc vào hai lớp nhãn  là nhãn lớp tương ứng của các 

Phương trình siêu phẳng chứa vector  trong không gian:



Đặt 

Như vậy,  biểu diện sự phân lớp của  vào hai lớp đã nêu.

### Các bước chính của phương pháp SVM

* Tiền xử lý dữ liệu: Xử lý các dữ liệu đầu vào nếu chúng chưa phải là số thực, đưa chúng về dạng số của SVM.
* Chọn hàm hạt nhân: Cần chọn hàm hạt nhân phù hợp tương ứng cho từng.bài toán cụ thể để đạt được độ chính xác phân lớp cao nhất.
* Thực hiện việc kiểm tra chéo để xác định các tham số phù hợp nhất.
* Sử dụng các tham số cho việc huấn luyện tập mẫu.
* Kiểm thử két quả phân lớp với tập dữ liệu Test.

### Đánh giá độ chính xác của hệ thống

Đánh giá kết quả phương pháp phân lớp có thể được tính toán theo nhiều cách khác nhau. Theo khảo sát của Sebastiani [Fabrizio Sebastiani], độ đo phổ biến nhất được sử dụng để đánh giá kết quả phân lớp là độ hồi tưởng và độ chính xác [8]:

Bảng ‑. Đánh giá độ phức tạp hệ thống SVM

|  |  |  |  |
| --- | --- | --- | --- |
| Lớp Ci | | Dữ liệu thực | |
| Thuộc lớp Ci | Không Thuộc lớp Ci |
| Dự đoán | Thuộc lớp Ci | TPi | TNi |
| Không Thuộc lớp Ci | FPi | FNi |

Các kí hiệu sử dụng trong đánh giá độ chính xác.

Trong đó:

**TPi** (true positives): số lượng ví dụ dương được thuật toán phân đúng vào Ci

**TNi** (true negatives): số lượng ví dụ âm được thuật toán phân đúng vào Ci

**FPi** (false positives): số lượng ví dụ dương được thuật toán phân sai vào Ci

**FNi** (false negatives): số lượng ví dụ âm được thuật toán phân sai vào Ci

Độ chính xác Pri của lớp Ci là tỷ số ví dụ dương được thuật toán phân lớp, cho giá trị đúng trên tổng số ví dụ được thuật toán phân lớp vào Ci là:



Độ hồi tưởng  của lớp Ci là tỷ lệ số lượng ví dụ dương được thuật toán phân lớp, cho giá trị đúng trên tổng số ví dụ dương thực sự thuộc lớp Ci là:



Do nhóm thực hiện phân lớp tập dữ liệu thành 2 lớp, nên độ chính xác và độ hồi tưởng cần được tính cho toàn bộ tập 2 lớp. Nhóm sử dụng 3 độ đo là F1-Score, Precision và Recall để đánh giá độ chính xác của hệ thống với D là tập nhãn sẽ được phân lớp:

Precision:



Recall:



F1-Score:

x

# XÂY DỰNG HỆ THỐNG PHÁT HIỆN VIRUS

## Ý tưởng

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*” [9] 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.

Như đã trình bày bài toán phòng chống virus máy tính nói chung và bài toán phát hiện máy tính nói riêng là một bài toán rất khó vì vậy nhóm đã quyết định chia bài toán này thành nhiều bài toán nhỏ để giải quyết. 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.

* **Xây dựng dữ liệu** : từ các tập tin đầu vào xây dựng tập hợp các chuỗi nhị phân được gán nhãn, chuẩn bị cho quá trình huấn luyện.
* **Phân lớp các chuỗi**: từ các chuỗi đã được gán nhãn, dữ liệu của bài toán 1, xây dựng hệ thống mạng no-ron để đánh giá mức độ nguy hiểm của một chuỗi.
* **Phân lớp tập tin**: từ mạng noron kết quả của bài toán 2, độ nguy hiểm của tập tin và xây dựng SVM để phân lớp tập tin đó.

### Xây dựng dữ liệu

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.

Kết quả của bài toán là 2 tập gồm các chuỗi virus là chuỗi sạch. Đây chính là bước chuẩn bị dữ liệu cho bài toán 2 (bài toán trọng tâm).

#### Dữ liệu

* Input:
  + Thư mục tập tin virus.
  + Thư mục tập tin sạch.
* Output: Một cơ sở dữ liệu bao gồm các chuỗi nhị phân đã được gán nhãn là virus hoặc benign

#### Giải pháp

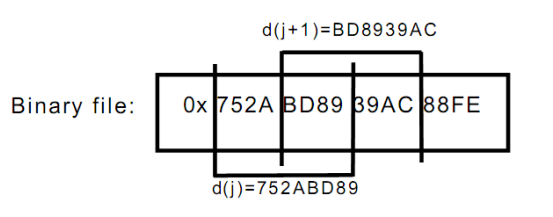
Mặc dù bản chất virus cũng là một chương trình máy tính nhưng do kích thước tập tin virus thường phải rất nhỏ nên chúng thường chứa các chỉ dẫn thực thi ngắn mà ít xuất hiện trong các chương trình sạch. Mục tiêu của ta là hướng đến nhận dạng các đoạn mã này.

Nhóm thực hiện khai thác ý tưởng được nêu ra ở trên kết hợp sử dụng một giải thuật nổi tiếng trong Miễn dịch nhân tạo là giải thuật **chọn lọc âm tính** để loại ra các chuỗi nhị phân thường gặp trong các tập tin sạch trong tập chuỗi virus.

**Giải thuật chọn lọc âm tính:**

Giải thuật chọn lọc âm tính được sử dụng để loại trừ các chuỗi sạch các đoạn mã tồn tại trong cả các tập tin virus và tập tin chương trình sạch. Mục đích cuối cùng là chỉ giữ lại các chuỗi virus thật sự.

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 ‑. 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, R-Contiguous.



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

### Phân lớp các chuỗi

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

#### Dữ liệu

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

#### Hướng giải quyết

Xây dựng mạng Kohonen, với đầu vào là các bit nhị phân (32 bits). Đầu ra sẽ là số lượng các detector nhận diện chuỗi này là virus. Từ đó tính toán được độ nguy hiểm của chuỗi virus này.

Tuy nhiên, số lượng chuỗi virus là rất ít so với chuỗi sạch, nên chúng ta chỉ chọn ra một số lượng chuỗi sạch vừa đủ để thực hiện train mạng nơ-ron. Từ đó phát sinh ra bước chuẩn bị dữ liệu.

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. Vì vậy nhóm đã thực hiện mapping chuỗi 32 bits thành đối tượng gồm 4 số thập phân. Việc làm này sẽ giảm số lượng nơ-ron đầu vào của mạng xuống rất nhiều. Ngoài ra nhãn của chuỗi (được gán từ bước 1), cũng được phản ánh trong đối tượng này.



Hình ‑. Các bước chuẩn bị dữ liêu cho quá trình training mạng nơ-ron

Sau khi đã chuẩn bị dữ liệu xong thì tiến hành training mạng nơ-ron.

### Phân lớp tập tin

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

Độ nguy hiểm giữa tệp sạch và tệp virus chênh lệch nhau khá rõ ràng. Dựa vào giá trị mức độ nguy hiểm này, ta có thể chọn một giá trị ngưỡng (Threshold) 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.

#### Dữ liệu

* Input:
  + Thư mục tập tin virus.
  + Thư mục tập tin thực thi sạch.
* Output :
  + Các tệp tin phân đã được gán nhãn là virus hoặc benign.

#### Hướng giải quyết

1. Tiền xử lý

Từ kết quả của bước 2, chúng ta đã có được mạng nơ-ron để xác định độ nguy hiểm của các chuỗi. Để xác định được độ nguy hiểm của 1 file, đơn giản chúng ta chỉ cần tính trung bình độ nguy hiểm của các fragment (chuỗi 32 bits) trong tệp đó.

Nhóm nghiên cứu nhận thấy độ nguy hiểm của các chuỗi không thể xử lý san bằng như thế được, bởi vì một tệp có 1 chuỗi có dộ nguy hiểm cao sẽ có khả năng là tệp virus cao hơn so với tệp có nhiều chuối có độ nguy hiểm thấp. Vì thế chúng ta cần nhân thêm hệ số đối với các độ nguy hiểm này.

Ngoài ra các fragment có độ nguy hiểm gần nhau sẽ được gom lại thành nhóm để tăng tốc độ xử lý.



Hình ‑. Quá trình tính độ nguy hiểm của tệp tin

##### **Xử lý dữ liệu**

Sau khi đã tính được độ nguy hiểm của các tệp tin. Chúng ta sử dụng chúng như đầu vào để training SVM.

Vì dữ liệu đầu vào là các số cụ thể nên nhóm quyết định sử dụng SVM tuyến tính để phân lớp dữ liệu.

## Cài đặt

Trong phần này sẽ trình bày về cách nhóm đã hiện thực hóa các cách giải quyết nêu ở phần trên.

Chương trình cài đặt trên môi trường như sau:

* + .Net Framework 4.5
  + DevExpress
  + Một số thư viện Aforge 2.2.5.0, Accord.NET 3.0.2.0

Như ở phần ý tưởng đã trình bày, chia nhỏ bài toán để thực hiện. Vì vậy phần cài đặt sẽ tiến hành chia nhỏ phần mềm thành cách module tương ứng với mỗi vài toàn. Mỗi module có dữ liêu input, output, các kỹ thuật xử lý riêng. Tuy nhiên giữa các module vẫn có liện hệ chặt chẽ về thời gian, vì bước sau sử dụng data của bước trước làm input.

### Pre-Process

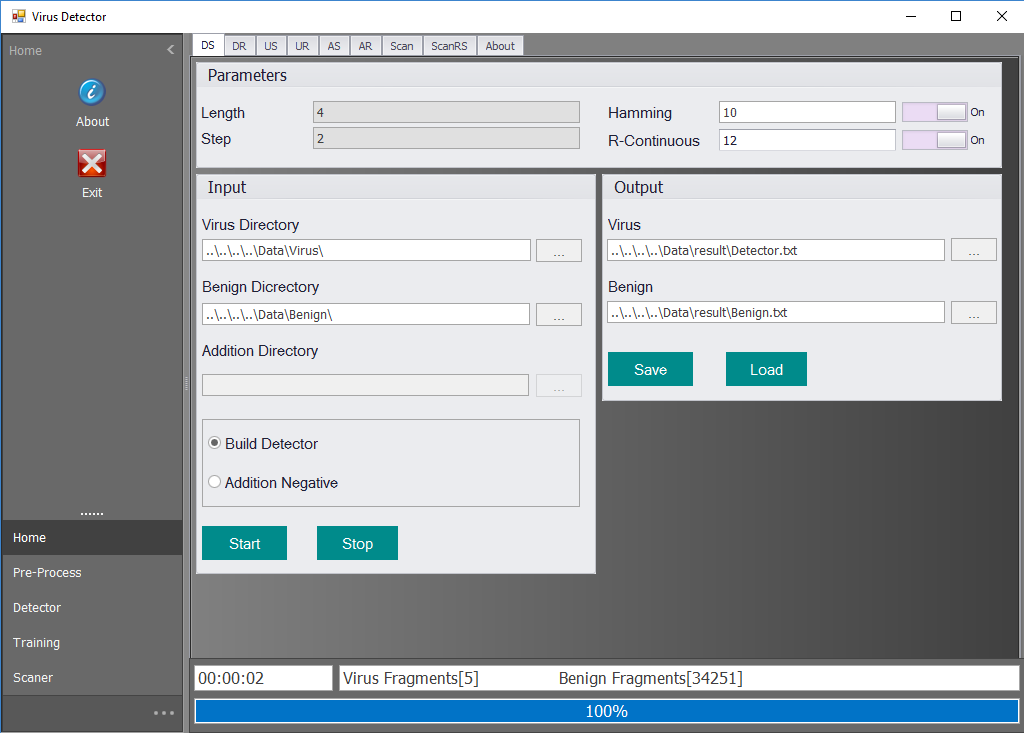
Kết quả output của bước này là 2 tập <String> : tập virus và tập sạch.

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

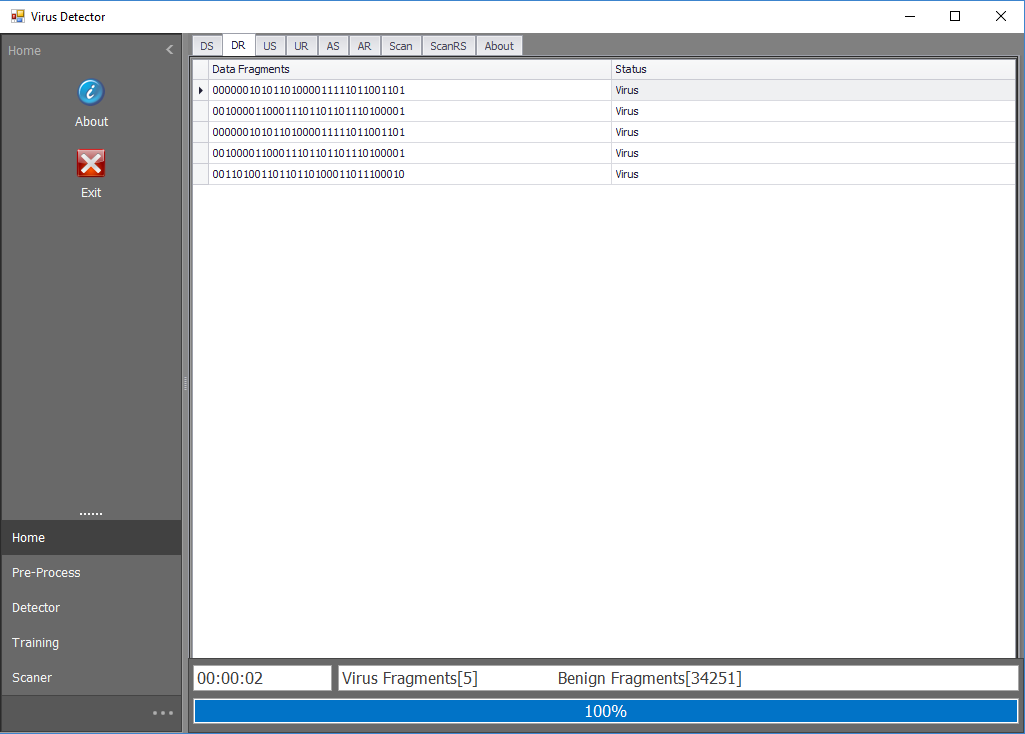
|  |  |
| --- | --- |
| **Đối tượng** | **Mô tả** |
| Matching | Cung cấp các phương thức xác định mức độ tương đồng (hoặc sai khác) giữa các chuỗi nhị phân dựa trên các luật so khớp Hamming hoặc R –contiguous. |
| TrainingData | Mỗi đối tượng trainingData là một danh sách các đoạn mã nhị phân trích xuất từ các tập tin. |
| DataGeneration | Cung cấp các phương thức chính để đọc nguồn dữ liệu cũng như cài đặt thuật toán NSA. |

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

|  |  |  |
| --- | --- | --- |
| **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). |



Hình ‑. Màn hình setting tham số cho module Pre-Process



Hình ‑. Màn hình kết quả của module Pre-Process

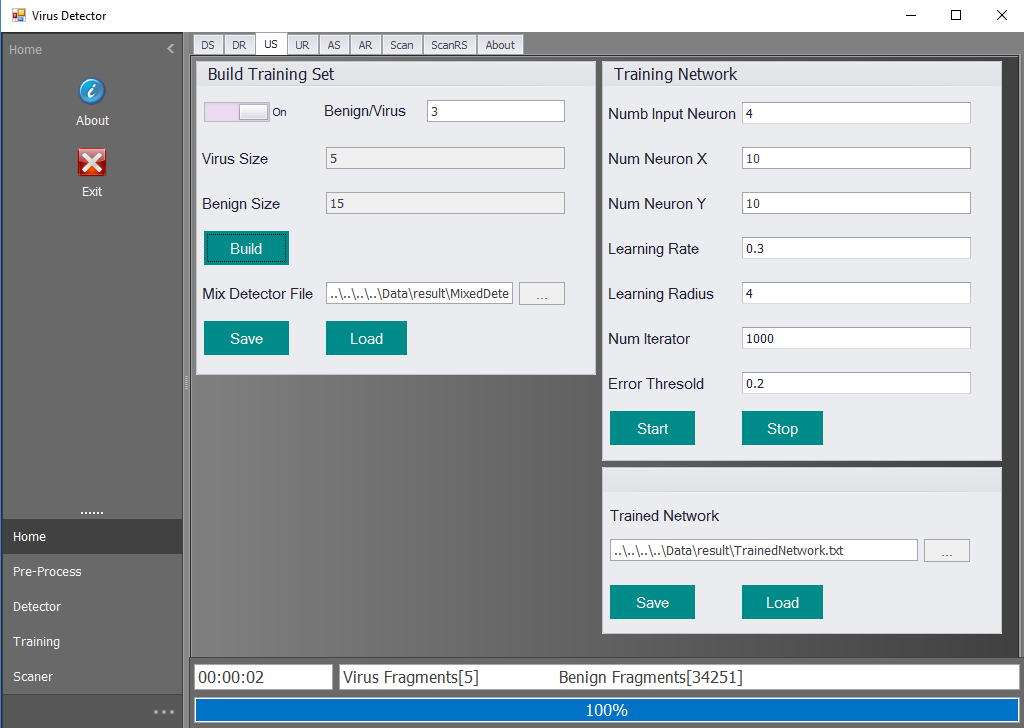
### Detector

Bảng ‑. Một số hàm, lớp trong module Detector

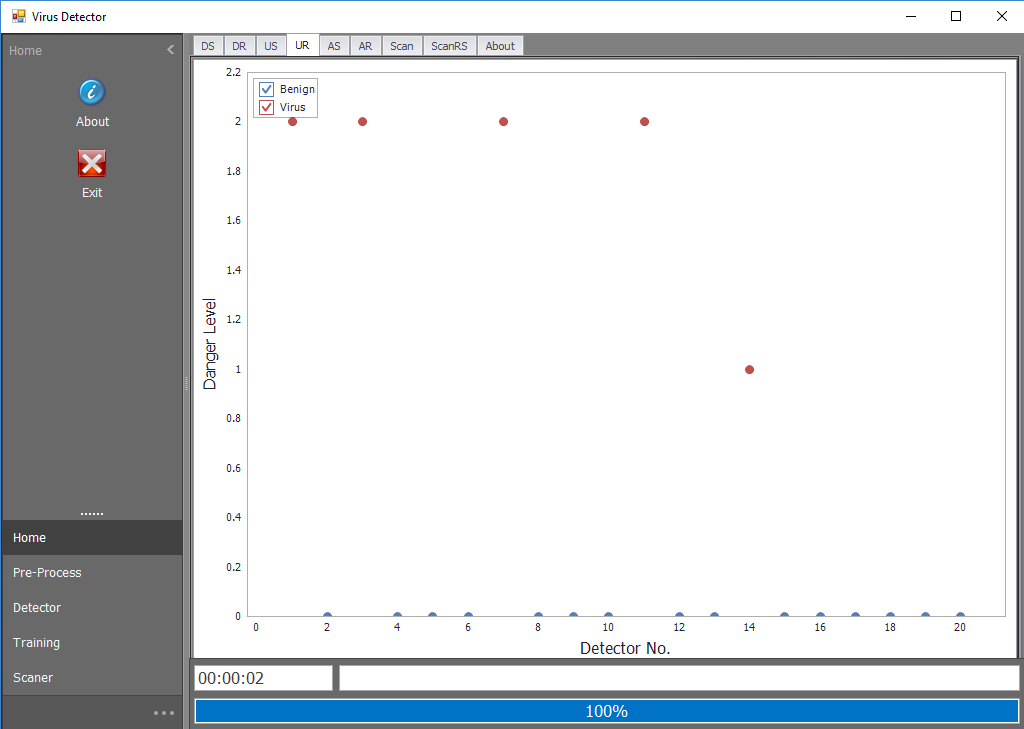
|  |  |  |
| --- | --- | --- |
| **STT** | **Function** | **Mô tả** |
| 1 | ClusteringManager | Quản lí cho thuật toán phân cụm chuỗi Virus bằng mạng Kohonen, thực hiện việc nhận input đầu vào, tính toán thông số cho mạng Kohonen và luyện mạng Kohonen. |
| 2 | trainNetwork | Huấn luyện mạng Kohonen. |
| 3 | saveNetwork | Lưu mạng Kohonen xuống file. |
| 4 | loadNetwork | Đọc lại mạng Kohonen từ dữ liệu đã lưu |
| 5 | LKDistanceNeuron | Là các lớp kế thừa từ thư viện AForge.NET, kế thừa mạng Kohonen có sẵn nhưng tùy chỉnh cho phù hợp với thuật toán nhận biết Virus. Cụ thể là tùy chỉnh mạng Kohonen để có thể truyền vào input Vector có số lượng thuộc tính > số lượng vector đầu vào (Hàm trong thư viện chuẩn, khi input Vector có số lượng thuộc tính khác với số lượng vector đầu vào thì sẽ báo lỗi Exception). Mục đích là để thuận tiện hơn cho việc lập trình mapping các nơ ron với mức độ nguy hiểm của chuỗi 32 đầu vào. |
| 6 | LKSOMLearning | Xử lí thuật toán huấn luyện mạng Kohonen, kế thừa từ lớp SOMLearning.cs của thư viện AForge.NET và thêm thuật toán mapping các nơ ron với mức độ nguy hiểm của chuỗi 32 đầu vào. |

Bảng ‑. Tham số trong module Detector

|  |  |  |
| --- | --- | --- |
| **STT** | **Tham số** | **Mô tả** |
| 1 | Input | Tập dữ liệu huấn luyện, là mảng các chuỗi 33 bit (mỗi chuỗi 33 bit là một mảng 33 phần tử, 32 phần tử đầu là các bit dữ liệu, phần tử cuối đánh dấu chuỗi là chuỗi virus hay chuỗi sạch). |
| 2 | mappingDataIndex | Vị trí của bit dữ liệu label trong các vector đầu vào. Khi giá trị bằng -1, có nghĩa là bit dữ liệu label nằm tại vị trí cuối cùng của vector đầu vào. |



Hình ‑. Màn hình setting cho module detector



Hình ‑. Màn hình kết quả của Module Detector

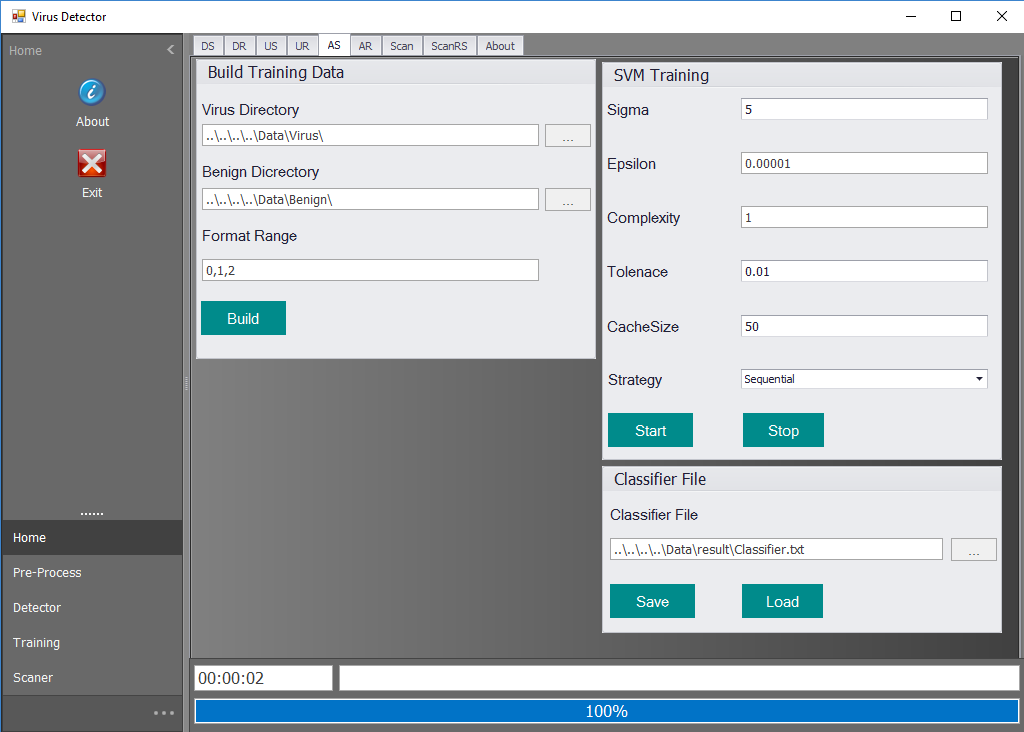
### Training

Bảng ‑. Các phương thức trong trong module Training

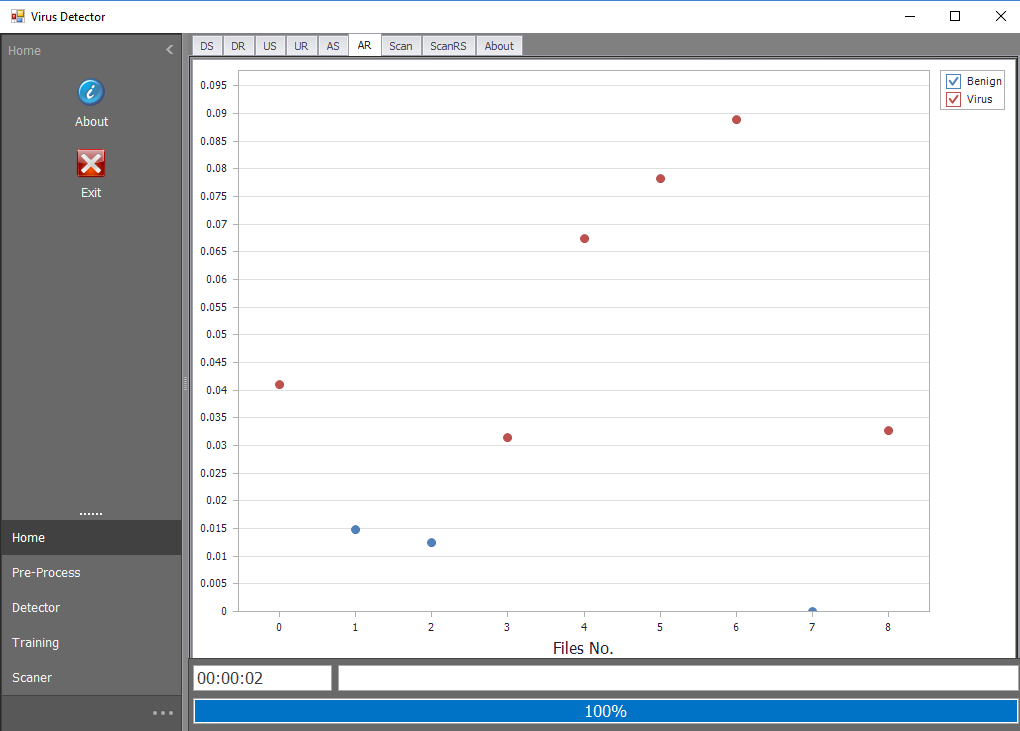
|  |  |  |
| --- | --- | --- |
| **STT** | **Tên** | **Mô tả** |
| 1 | buildTrainingSet | Chuẩn hóa dữ liệu. |
| 2 | Train | Huấn luyện mạng nơ-ron. |
| 3 | Compute | Tính toán mức độ nguy hiểm của tập tin. |
| 4 | getFormatData | Lấy mức độ nguy hiểm của tập tin đã tính toán được. |

Bảng ‑. Các tham số trong module Training

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên** | **Mô tả** |
| 1 | numOfHiddenNeuron | Số nơ ron lớp ẩn |
| 2 | numOfIterator | Số vòng lặp huấn luyện |
| 3 | errorThresold | ngưỡng lỗi |
| 4 | Network | Mạng Kohonen để phục vụ quá trình preprocess data cho mạng nơ ron nhân tạo |
| 5 | formatRange | Các mức độ nguy hiểm |



Hình ‑. Màn hình setting tham số cho module Training



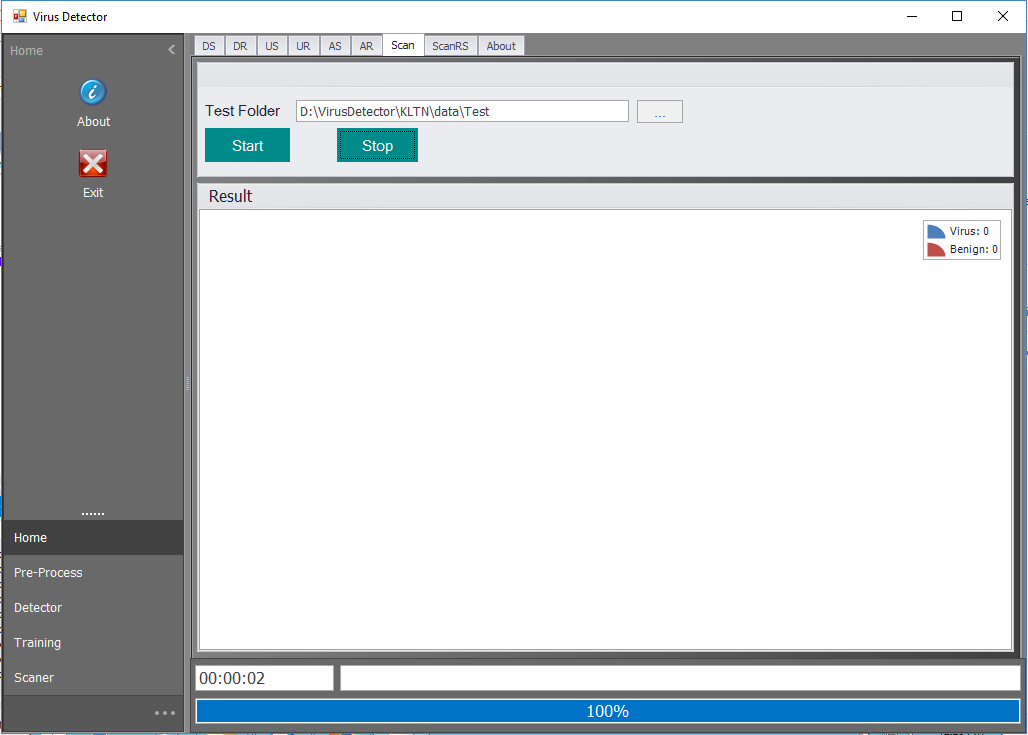
Hình ‑. Màn hình kết quả của Module Training

### Scanner

Module Scanner cho thực hiện thao tác cuối cùng, sử dụng kết quả của các module trước để kiểm tra một (hoặc nhiều) tệp tin là virus hay sạch.

Bảng ‑. Các phương thức trong trong module Scanner

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên** | **Mô tả** |
| 1 | ScanFileSvm | Thực hiện scan một file. |
| 2 | compute | Tính toán mức độ nguy hiểm của tập tin |
| 3 | getFormatData | Lấy mức độ nguy hiểm của tập tin đã tính toán được |



Hình ‑. Màn hình setting cho module Scanner

## Kiểm thử

### Dữ liệu

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.

Bảng ‑. Dữ liệu đầu vào

|  |  |  |
| --- | --- | --- |
| **Bộ dữ liệu** | **Số tập tin virus** | **Số tập tin sạch** |
| Data 1 | 50 | 50 |
| Data 2 | 100 | 50 |
| Data 3 | 200 | 50 |
| Data 4 | 300 | 50 |

### Kết quả

Như đã trình bày ở chương trước, ta sử dụng cơ chế so khớp trùng chuỗi R-contiguous và Hamming để làm điều kiện lọc âm tính. Đây là hai 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.

Bảng ‑. Sự tương quan giữa giá trị r và quá trình xây dựng dữ liệu huấn luyện

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tên bộ dữ liệu** | **Giá trị r** | **Giá trị h** | **Thời gian** | **Số chuỗi virus** | **Số chuỗi  sạch** |
| Data 1 | 8 | 14 | 00:00:19 | 0 | 1404991 |
|  | 14 | 10 | 00:09:48 | 1 | 1404991 |
|  | 17 | 8 | 00:12:47 | 18 | 1404991 |
| Data 2 | 8 | 14 | 00:00:37 | 0 | 1404991 |
|  | 14 | 10 | 00:02:47 | 3 | 1404991 |
|  | 17 | 8 | 00:31:02 | 65 | 1404991 |

Để đơn giản, chúng ta sử dụng chính các bộ data trên để làm data cho tất các các trường hợp.

#### Kết quả thử nghiệm trên bộ data 1

Với dữ liệu đầu vào : r = 17, h = 9, có được kết quả của bước 1 (pre-process) là 12 chuỗi virus. Nhóm nghiên cứu nhận thấy rằng đây là kết quả cho ra khả năng nhận diện virus tốt nhất.

Bảng ‑. Kết quả thử nghiệm 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Virus** | **Phát hiện** | **File Virus** | **File Benign** | **Tỉ lệ** |
| Data 1 | 48 | 50 | 0 | 0.96 |
| Data 2 | 53 | 100 | 0 | 0.53 |
| Data 3 | 61 | 200 | 0 | 0.31 |
| Data 4 | 69 | 300 | 0 | 0.23 |
| Data Test 1 | 0 | 0 | 50 | N/A |
| Data Test 2 | 3 | 0 | 100 | N/A |
| Data Test 3 | TODO | 50 | 50 | TODO |

Tỉ lệ chính xác của chương trình giảm dần khi tiếp xúc với các bộ dữ liệu lớn, nguyên nhân là vì dữ liệu training nhỏ, nên không thể nhận diện được các virus chưa xuất hiện trong dữ liệu training mà chỉ xuất hiện trong dữ liệu kiểm thử.

Tỉ lệ nhận dạng nhầm là thấp, tức khả năng nhận dạng nhầm của chương trình là rất thấp. Điều này rất cần thiết vì khi đi vào thực tế, chương trình sẽ hạn chế khả năng nhận dạng nhầm các tập tin hệ thống, các tập tin sạch.

#### Kết quả thử nghiệm trên bộ data 2

Với dữ liệu đầu vào : r = 17, h = 9, có được kết quả của bước 1 (pre-process) là 12 chuỗi virus. Nhóm nghiên cứu nhận thấy rằng đây là kết quả cho ra khả năng nhận diện virus tốt nhất.

Bảng ‑. Kết quả thử nghiệm 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Virus** | **Phát hiện** | **File Virus** | **File Benign** | **Tỉ lệ** |
| Data 1 | 50 | 50 | 0 | 0.96 |
| Data 2 | 100 | 100 | 0 | 1 |
| Data 3 | 199 | 200 | 0 | 0.995 |
| Data 4 | 298 | 300 | 0 | 0.995 |
| Data Test 1 | 1 | 0 | 50 | N/A |
| Data Test 2 | 7 | 0 | 100 | N/A |
| Data Test 3 | TODO | 50 | 50 | TODO |

#### Kết quả thử nghiệm trên bộ data 3

Bảng ‑. Kết quả thử nghiệm 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Virus** | **Phát hiện** | **File Virus** | **File Benign** | **Tỉ lệ** |
| Data 1 | 50 | 50 | 0 | 0.96 |
| Data 2 | 100 | 100 | 0 | 1 |
| Data 3 | 199 | 200 | 0 | 0.995 |
| Data 4 | 298 | 300 | 0 | 0.995 |
| Data Test 1 | 1 | 0 | 50 | N/A |
| Data Test 2 | 7 | 0 | 100 | N/A |
| Data Test 3 | TODO | 50 | 50 | TODO |

#### Kết quả thử nghiệm trên bộ data 4

Bảng ‑. Kết quả thử nghiệm 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Virus** | **Phát hiện** | **File Virus** | **File Benign** | **Tỉ lệ** |
| Data 1 | 50 | 50 | 0 | 0.96 |
| Data 2 | 100 | 100 | 0 | 1 |
| Data 3 | 199 | 200 | 0 | 0.995 |
| Data 4 | 298 | 300 | 0 | 0.995 |
| Data Test 1 | 1 | 0 | 50 | N/A |
| Data Test 2 | 7 | 0 | 100 | N/A |
| Data Test 3 | TODO | 50 | 50 | TODO |

# KẾT LUẬN

## Kết quả đạt được

Về mặt lý thuyết, sau một khoảng thời gian khá dài thu thập và nghiên cứu tài liệu từ nhiều nguồn khác nhau, những kết quả chính mà nhóm đã đạt được khi thực hiện đề tài này bao gồm:

* Có 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.
* Các tri thức tổng quan sơ lược về hệ miễn dịch nhân tạo, mạng nơ ron nhân tạo cùng một số thuật toán máy học hỗ trợ, nắm vững về khái niệm, các thuật toán chính, cách tổ chức xây dựng mô hình…
* Tìm hiểu hệ miễn dịch của con người và cơ chế đặc hiệu miễn dịch: một số khái niệm về miễn dịch sinh học, các thành phần chủ yếu của hệ miễn dịch, các cơ chế kích hoạt và nhận diện miễn dịch cơ bản, chức năng sinh học của hệ miễn dịch (các cơ quan lymphoid tiên phát, các cơ quan lymphoid thứ phát, các tầng bảo vệ của hệ miễn dịch), hệ miễn dịch thích nghi và bẩm sinh (hệ miễn dịch bẩm sinh, hệ miễn dịch thích nghi), nhận dạng mẫu, phân biệt Self và Nonself (phép chọn lọc tích cực, chọn lọc tiêu cực, chọn lọc nhân bản). Trình bày một cách tổng quát về mô hình mạng nơron và ứng dụng mạng nơron trong máy học.
* Trình bày một cách hệ thống về Data Mining và các phương pháp khái phá dữ liệu.
* Nghiên cứu, phân tích và sử dụng thuật toán SVM giải quyết bài toán phân lớp dữ liệu.
* Bên cạnh đó, nâng cao khả năng tìm hiểu, phân tích, nghiên cứu các kỹ thuật, cũng như thiết kế hoàn thiện hệ thống. Nhằm giải quyết các bài toán, các vấn đề.

Về mặt thực nghiệm kết quả mà nhóm thực hiện được bao gồm:

* Cài đặt thành công một chương trình minh họa cho hướng tiếp cận được nêu ra.
* Cài đặt các thuật toán hệ miễn dịch nhân tạo như sử dụng khoảng cách Hamming, Rcontiguos, giải thuật chọn lọc âm tính (NSA), thuật toán phân cụm Kmeans.
* Sử dụng mạng nơ-ron Kohonen (SOM) trong việc phân lớp, phân cụm.
* 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 sử dụng.
* Cơ sở dữ liệu huấn luyện được nạp vào chương trình dưới dạng đường dẫn đến thư mục, cho phép ta dễ dàng thay đổi dữ liệu huấn luyện và kiểm tra theo từng tình huống cụ thể.

## Hướng phát triển

Do mức độ phức tạp cũng như hạn chế về mặt thời gian và kiến thức nên khóa luận vẫn còn tồn tại một số hạn chế, một số kiến thức chuyên môn về nghiên cứu chỉ mức khái quát, chưa đi sâu chi tiết nhiều thuật toán liên quan. Hướng nghiên cứu mới nên nhóm chưa có nhiều kinh nghiệm trong việc tìm hiểu, tìm các tài liệu liên quan.

Nhóm sẽ định hướng một số nghiên cứu tiếp theo:

* Viết bài báo khoa học.
* Nghiên cứu mở rộng 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.

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | Peter Mell, Karen Kent, Joseph Nusbaum, Guide to Malware Incident Prevention and Handling, National Institute of Standards and Technology, 2005. |
| [2] | Mark R. Colombell, "The Legislative Response to the Evolution of Computer Viruses," 2002. |
| [3] | Nam, Trần Hải, "Các phương pháp lây lan và phá hoại của virus," in *luận văn thạc sĩ khoa học*, ĐH Bách Khoa Hà Nội, 2007. |
| [4] | Dipankar Dasgupta, An Overview of Artificial Immune Systems and Their Applications, Springer Berlin Heidelberg, 1999. |
| [5] | Nguyên, PGS.TS Vũ Thanh, "Nghiên cứu một số phương pháp máy học nhằm tăng cường khả năng bảo mật thông tin và an toàn máy tính". |
| [6] | Mai Trọng Khang, Nguyễn Hoàng Ngân, "Nghiên cứu một số thuật toán máy học và hệ miễn dịch nhân tạo trong phát hiện virus máy tính," Đại học Công nghệ thông tin, Khóa Luận Tốt Nghiệp Đại Học, 2013. |
| [7] | Jiawei Han,Jian Pei,Micheline Kamber, Data Mining: Concepts and Techniques (3rd Edition), 2012. |
| [8] | Jason Weston, "Support Vector Machine and Statistical Learning Theory," NEC Labs America. |
| [9] | Jeffrey Horton, "Computer Viruses: an Introduction," 1997. |
| [10] | Nguyễn Xuân Hoài, Nguyễn Văn Trường, Vũ Mạnh Xuân, "Hệ miễn dịch nhân tạo và ứng dụng," *Tạp chí Khoa học và Công nghê Đại học Thái Nguyên,* 2007. |
| [11] | Chin-Teng Lin, CSG Lee, "Neural fuzzy systems: a neuro-fuzzy synergism to intelligent systems," *Google, Inc. (search). SIGN IN SIGN UP.,* no. Prentice-Hall, Inc., 1996. |
| [12] | A. Ng, "Support Vector Machines," in *CS229 Lecture notes*, Stanford University. |
| [13] | Paulo Gaspar, Jaime Carbonell, José Luís Oliveira, "On the parameter optimization of Support Vector Machines for binary classification," Journal of Integrative Bioinformatics, 2012. |
| [14] | Thorsten Joachims, "SVM light - Support Vector Machine," Cornell University, 2008. |