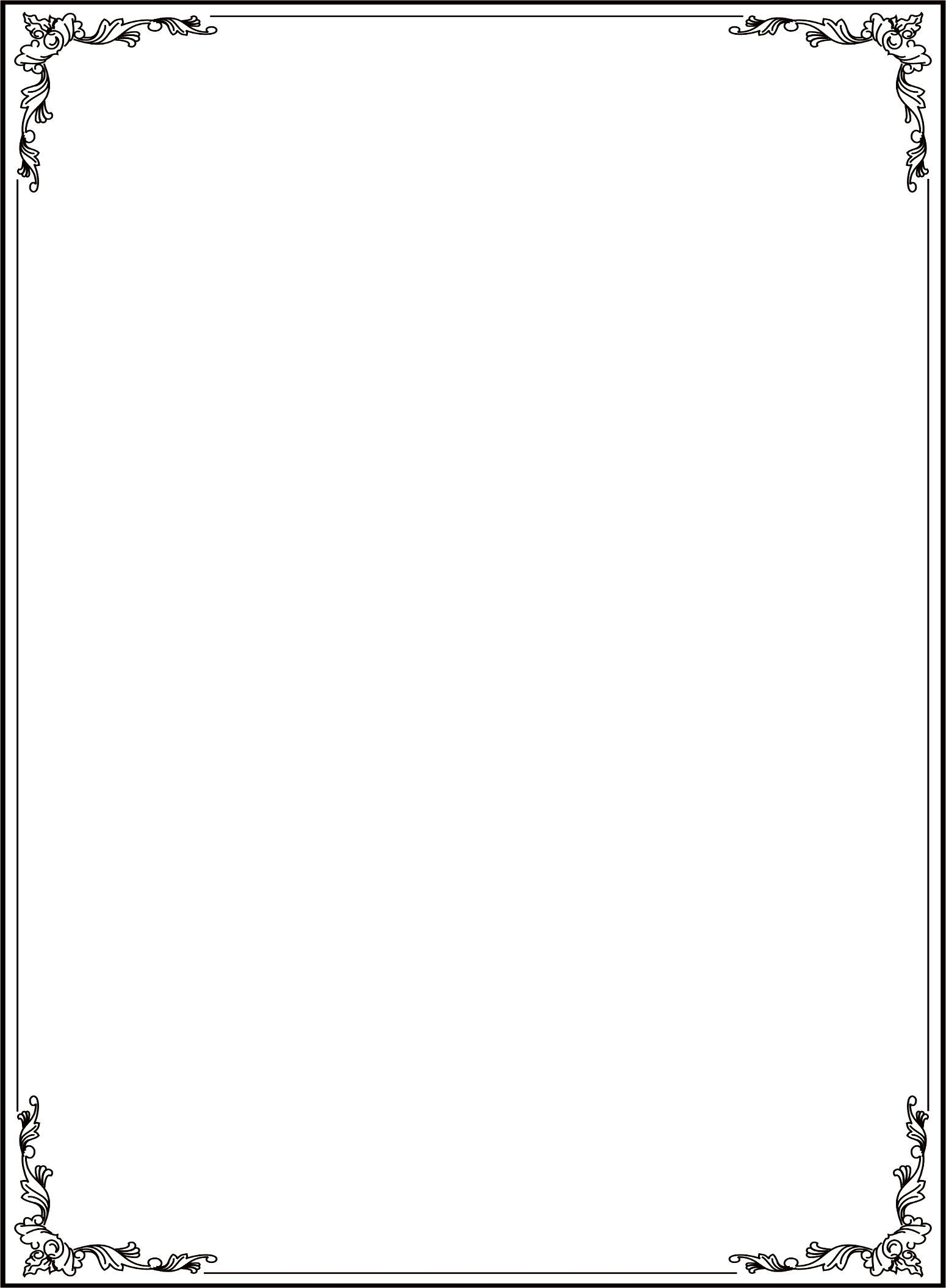
**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA HỆ THỐNG THÔNG TIN**



**BÁO CÁO ĐỒ ÁN**

**AN TOÀN VÀ BẢO MẬT HỆ THỐNG THÔNG TIN**

**ĐỀ TÀI:**

**MALWARE DETECTION**

**Giảng viên hướng dẫn:** Hà Lê Hoài Trung

**Lớp:** IS335.O11.HTCL

**Sinh viên thực hiện:**

| Hồ Thị Hằng | 20521285 |
| --- | --- |
| Dương Nhật Minh | 20520242 |
| Nguyễn Tấn Phát | 20521736 |
| Nguyễn Anh Nguyên | 20521674 |

***TP.Hồ Chí Minh, tháng 12 năm 2023***

**MỤC LỤC**

[**CHƯƠNG 1: TÌM HIỂU MALWARE** 3](#_gjdgxs)

[**1.**](#_30j0zll) **Mô tả Malware** 3

[**1.1.**](#_1fob9te) **Khái niệm** 3

[**1.2.**](#_3znysh7) **Cơ chế hoạt động của Malware** 3

[**1.3.**](#_2et92p0) **Dấu hiệu nhận biết Malware** 4

[**2.**](#_tyjcwt) **Các loại Malware** 5

[**2.1.**](#_3dy6vkm) **Phân loại theo cách Malware phát tán** 5

[**2.1.1.**](#_1t3h5sf) **Virus** 5

[**2.1.2.**](#_4d34og8) **Worm** 6

[**2.1.3.**](#_2s8eyo1) **Trojan** 7

[**2.2.**](#_17dp8vu) **Phân loại theo hành động malware sau khi lây nhiễm vào máy tính** 8

[**2.2.1.**](#_3rdcrjn) **Spyware** 8

[**2.2.2.**](#_26in1rg) **Rootkit** 9

[**2.2.3.**](#_lnxbz9) **Adware** 10

[**2.2.4.**](#_35nkun2) **Ransomeware** 11

[**2.2.5.**](#_1ksv4uv) **Cryptojacking** 12

[**2.2.6.**](#_44sinio) **Malvertising** 13

[**3.**](#_2jxsxqh) **Cách phòng chống Malware** 14

[**CHƯƠNG 2: CÁC MÔ HÌNH HỌC MÁY SỬ DỤNG** 16](#_z337ya)

[**1.**](#_3j2qqm3) **Feedforward Neural Networks (FNN)** 16

[**2.**](#_1y810tw) **Random Forest** 17

[**3.**](#_4i7ojhp) **Logistic Regression** 18

[**4.**](#_2xcytpi) **Gated Recurrent Units ( GRU )** 19

[**5.**](#_1ci93xb) **Recurrent Neural Network ( RNN )** 20

[**CHƯƠNG 3: TRỰC QUAN KẾT QUẢ THỰC HIỆN** 21](#_3whwml4)

[**1.**](#_2bn6wsx) **Giới thiệu dataset** 21

[**2.**](#_qsh70q) **Quá trình thực hiện** 23

[**2.1.**](#_3as4poj) **Nguồn dữ liệu** 23

[**2.2.**](#_1pxezwc) **Tiền xử lý dữ liệu** 25

[**2.3.**](#_49x2ik5) **Chuẩn bị dữ liệu** 32

[**2.4.**](#_2p2csry) **Xây Dựng Và Đánh Giá Mô Hình** 33

[**2.4.1.**](#_147n2zr) **Mô hình Feedforward Neural Networks (FNN)** 33

[**2.4.2.**](#_3o7alnk) **Mô hình Random Forest** 39

[**2.4.3.**](#_23ckvvd) **Mô hình Logistic Regression** 45

[**2.4.4.**](#_ihv636) **Mô hình Gated Recurrent Unit (GRU)** 50

[**2.4.5.**](#_32hioqz) **Mô hình Recurrent Neural Network (RNN)** 56

[**2.5.**](#_1hmsyys) **So sánh độ chính xác giữa các mô hình** 62

# **CHƯƠNG 1: TÌM HIỂU MALWARE**

****

1. **Mô tả Malware**
   1. **Khái niệm**

***Malware*** là từ viết tắt của cụm Malicious Software và nó có nghĩa là phần mềm độc hại. Malware thường được dùng để chỉ các chương trình hoặc các tệp tin do Hacker tạo ra có khả năng làm cản trở hoạt động bình thường của hệ thống, gây hại cho người sử dụng internet.

Malware có khả năng xâm nhập trái phép vào hệ thống, thiết bị; kiểm soát và làm hỏng hoặc vô hiệu hóa hệ thống mạng; gây cản trở hoạt động của Website; cài cắm và theo dõi thiết bị.

* 1. **Cơ chế hoạt động của Malware**

Các malware thường lây nhiễm vào máy tính bằng cách lừa người dùng click và cài đặt một chương trình nào đó. Sau khi cài đặt xong, malware sẽ thực hiện các hành động mà người dùng không lường trước được, chẳng hạn như:

* Tự sao chép các phần khác nhau trong file system.
* Cài đặt các ứng dụng chiếm keystroke hay tài nguyên commandeer system. Thường chạy ngầm mà người dùng không biết, đồng thời ảnh hưởng xấu đến tốc độ của hệ thống.
* Chặn quyền truy cập vào các file, chương trình hoặc thậm chí là chính hệ thống. Đôi khi, các hacker yêu cầu người dùng phải trả tiền để có thể lấy lại quyền truy cập.
* Bombard trình duyệt hay desktop bằng quảng cáo.
* Phá vỡ các thành phần thiết yếu của hệ thống, làm thiết bị không thể hoạt động được.

Việc trigger tấn công phát tán malware có thể do hành động của người dùng. Nhưng cách trigger phổ biến nhất chính là thông qua việc click, thường là click vào các link hay pop-up. Trong đó, chúng có thể hiện lên các thông báo như “Nhận giải thưởng của bạn ngay tại đây”, “Tài khoản đã bị xâm phạm. Vui lòng đăng nhập và xác minh các thanh toán gần đây”. Hay cũng có thể là “Hệ thống đã bị nhiễm virus! Hãy click vào đây để quét”,… Sau đó, lần click tiếp theo sẽ trigger trình download các phần mềm độc hại, ngay cả khi người dùng không cố ý chọn.

Bên cạnh đó, malware cũng có thể được ngụy trang dưới dạng các chương trình hay ứng dụng chuyển đổi PDF, giải nén file,… Nhưng khi chương trình được tải xuống, nó sẽ bắt đầu thực hiện các thay đổi trái phép trên hệ thống. Chẳng hạn như theo dõi người dùng, hiển thị các pop-up, thay đổi kết quả của công cụ tìm kiếm.

* 1. **Dấu hiệu nhận biết Malware**

Khi thiết bị nhiễm Malware, bạn có thể nhận thấy các dấu hiệu sau:

* Máy tính chạy chậm, tốc độ xử lý của hệ điều hành giảm cho dù bạn đang điều hướng Internet hay chỉ sử dụng các ứng dụng cục bộ.
* Bạn bị làm phiền bởi quảng cáo pop-up, mà cụ thể hơn là Adware.
* Hệ thống liên tục gặp sự cố, bị đóng băng hoặc hiển thị BSOD – màn hình xanh (đối với Windows).
* Dung lượng ổ cứng giảm bất thường.
* Hoạt động Internet của hệ thống tăng cao không rõ nguyên nhân.
* Tài nguyên hệ thống tiêu hao bất thường, quạt máy tính hoạt động hết công suất.
* Trang chủ của trình duyệt mặc định thay đổi mà không có sự cho phép của bạn. Các liên kết bạn nhấp vào sẽ chuyển hướng bạn đến các trang không mong muốn.
* Các thanh công cụ, tiện ích mở rộng hoặc plugin mới được thêm vào trình duyệt.
* Các chương trình anti-virus ngừng hoạt động và không cập nhật được.
* Bạn nhận được thông báo đòi tiền chuộc từ Malware, nếu không dữ liệu của bạn sẽ bị xóa.

1. **Các loại Malware**

Hiện nay có rất nhiều Malware khác nhau và một vài Malware phổ biến và gây hậu quả nghiêm trọng có thể nói đến như sau:

* 1. **Phân loại theo cách Malware phát tán**
     1. **Virus**

***Virus*** là một đoạn code máy tính, nó tự chèn mình vào trong code của một chương trình độc lập khác. Sau đó buộc chương trình thực hiện hành động độc hại và tự phát tán. Loại chương trình này vô cùng nguy hiểm vì có khả năng sinh sôi, lây lan ra khắp hệ thống phần mềm, gây thiệt hại phần cứng với tốc độ rất nhanh. Nếu không khắc phục kịp thời, mọi thông tin, dữ liệu, thậm chí là thiết bị đều sẽ mất kiểm soát.

**Ví dụ:**

***Tên:*** Melissa

***Loại:*** Macro Virus (virus sử dụng ngôn ngữ macro trong các ứng dụng văn bản)

***Thời gian xuất hiện:*** Năm 1999

***Mô tả:*** Melissa là một trong những virus máy tính đầu tiên mà tình trạng lây nhiễm trên Internet lan rộng nhanh chóng. Nó thường được phát tán qua email, nơi mà nó kèm theo một tài liệu văn bản Microsoft Word có chứa mã độc hại. Khi người dùng mở tài liệu, virus sẽ tự động gửi bản sao của chính nó đến 50 địa chỉ email trong danh bạ của người dùng.

***Hậu quả:*** Melissa gây ra sự cố nghiêm trọng cho hàng nghìn tổ chức và cá nhân bởi việc làm quá tải hệ thống email và gây mất ổn định cho một số mạng máy tính. Nó đã gây tổn thất lớn và làm tăng sự chú ý đối với nguy cơ từ các loại virus và malware khác.***Phòng tránh và loại bỏ:*** Để bảo vệ khỏi Melissa và các loại virus tương tự, người dùng được không nên mở các tệp tin đính kèm không mong muốn trong email và cập nhật đều đặn phần mềm antivirus để phát hiện và loại bỏ virus từ hệ thống.

* + 1. **Worm**

***Worm (Sâu máy tính)*** là một loại Malware còn độc hại hơn cả Virus, nó có khả năng lây lan và nhân bản cực kì nhanh khi đã lây nhiễm vào máy tính.

Không giống như Virus vốn cần có vật chủ, Worm lây nhiễm vào một thiết bị thông qua các file tải về hoặc một kết nối mạng trước khi nhân bản và phát tán. Worm có thể khiến cho thiết bị hoạt động không ổn định, mất dữ liệu cá nhân v.v...

Worm rất khó khắc phục và giải quyết các hậu quả. Bởi Worm có thể tự sinh sôi, hoạt động mà không chịu bất kỳ sự tác động, điều khiển nào đến từ con người cả. Thậm chỉ khi đã bị tiêu diệt rồi thì vẫn có khả năng tự tái tạo, hoạt động lại như bình thường.

**Ví dụ:**

***Tên:*** ILOVEYOU

***Loại:*** Worm

***Thời gian xuất hiện:*** Năm 2000

***Mô tả:*** ILOVEYOU là một loại worm phổ biến được phát tán qua email. Nó đi kèm với một tệp tin đính kèm có tên "LOVE-LETTER-FOR-YOU.txt.vbs." Khi người dùng mở tệp tin, nó sẽ tự động gửi bản sao của chính nó đến tất cả địa chỉ email trong danh bạ và cố gắng xóa các tệp tin quan trọng trên máy tính.

***Hậu quả:*** ILOVEYOU đã lan rộng rất nhanh, gây tổn thất lớn cho doanh nghiệp và người dùng cá nhân. Nó làm chậm đáng kể các hệ thống email và gây thiệt hại nặng nề cho dữ liệu cá nhân.

***Phòng tránh và loại bỏ:*** Để tránh ILOVEYOU và các loại worm khác, người dùng cần cẩn thận với các tệp tin đính kèm không mong muốn và duy trì phần mềm antivirus cập nhật. Điều này giúp phát hiện và ngăn chặn sự lây nhiễm từ các loại malware.

* + 1. **Trojan**

***Trojan*** thường được ngụy trang thành các phần mềm hữu ích cho máy tính. Một khi người dùng dính bẫy và tải những phần mềm có chứa Trojan về, chúng có thể truy cập vào các dữ liệu nhạy cảm rồi chỉnh sửa - ngăn chặn hoặc tệ hơn là xóa luôn các dữ liệu đó... vậy nên hiệu năng hoạt động của thiết bị đã nhiễm Trojan sẽ cực kì tệ hại.

Thực chất Trojan giống như một cánh cổng mở ra và cho phép hàng triệu loại Virus khác nhau tiến công, gây hại cho máy tính.

Khác với Virus và Worm, Trojan không có chức năng sao chép dữ liệu những lại có khả năng hủy diệt rất kinh khủng.

**Ví dụ:**

***Tên:*** Zeus (còn được biết đến với tên gọi Zbot)

***Loại:*** Trojan Horse

***Thời gian xuất hiện:*** Đầu những năm 2000

***Mô tả:*** Zeus là một Trojan Horse phổ biến được sử dụng để đánh cắp thông tin ngân hàng và thông tin đăng nhập từ người dùng máy tính. Nó thường được phát tán qua các email rất giống với email chính thức từ tổ chức tài chính hoặc các trang web ngân hàng. Khi người dùng mở tệp tin đính kèm hoặc truy cập liên kết trong email, Zeus sẽ cài đặt trên máy tính và bắt đầu theo dõi và gửi thông tin cá nhân về một máy chủ từ xa.

***Hậu quả:*** Zeus đã gây ra nhiều vụ mất cắp tài khoản ngân hàng và thông tin cá nhân. Nó đã tiến xa bằng cách sử dụng phương thức nguy hiểm như keylogging để ghi lại thông tin đăng nhập và mật khẩu của người dùng.

***Phòng tránh và loại bỏ:*** Để tránh Zeus và các Trojan Horse khác, người dùng cần tránh mở các tệp tin và liên kết không xác định từ email hoặc nguồn không đáng tin cậy. Cũng quan trọng là cập nhật phần mềm antivirus và thực hiện quét máy tính đều đặn để phát hiện và loại bỏ malware.

* 1. **Phân loại theo hành động malware sau khi lây nhiễm vào máy tính**
     1. **Spyware**

***Spyware*** là một dạng Malware được thiết kế để bí mật chạy ngầm trên máy tính và gửi tất cả các loại dữ liệu mà nó thu thập được về cho người đã tạo ra nó.

Spyware không chỉ làm suy giảm hiệu năng hoạt động của thiết bị, nó còn nhắm đến rất nhiều thông tin nhạy cảm và có thể cho phép người tạo ra nó truy cập từ xa đến thiết bị đã bị lây nhiễm.

Spyware thường được sử dụng để ăn cắp các thông tin tài chính hoặc cá nhân. Một trong những loại Spyware phổ biến nhất là Keylogger và nó có thể ghi lại các thao tác bàn phím của bạn để lấy mật khẩu và các thông tin cá nhân khác.

**Ví dụ:**

***Tên:*** FinFisher (còn được biết đến với tên gọi FinSpy)

***Loại:*** Spyware

***Thời gian xuất hiện:*** Cuối những năm 2000

***Mô tả:*** FinFisher là một phần mềm gián điệp được thiết kế để thu thập thông tin từ máy tính và thiết bị di động mà nó lây nhiễm. Phần mềm này thường được phân phối thông qua các chiến dịch phishing hoặc tận dụng các lỗ hổng bảo mật trong hệ điều hành. Khi cài đặt, nó có thể theo dõi và ghi lại mọi hoạt động của người dùng, bao gồm cả thông tin đăng nhập, vị trí địa lý, và thậm chí là màn hình máy tính.

***Hậu quả:*** FinFisher thường được sử dụng cho mục đích gián điệp quốc gia và có thể gây nghiêm trọng cho quyền riêng tư và an ninh cá nhân. Các tổ chức và cá nhân có thể trở thành mục tiêu của spyware này mà họ có thể không hề biết.

***Phòng tránh và loại bỏ:*** Để bảo vệ khỏi spyware như FinFisher, người dùng cần duy trì phần mềm antivirus và anti-spyware được cập nhật đều đặn. Họ cũng nên tránh mở các tệp tin hoặc liên kết không xác định, duyệt web an toàn, và kiểm tra định kỳ các ứng dụng và hệ điều hành để cập nhật các bản vá bảo mật mới.

* + 1. **Rootkit**

***Rookit*** là một chương trình, thường là một tập hợp các công cụ phần mềm, cho phép Hacker truy cập từ xa và kiểm soát máy tính hay hệ thống khác. Rootkit bao gồm các công cụ (thường là bất hợp pháp) có quyền truy cập root với hệ thống đích, và Rootkit cũng dùng nó để che giấu sự hiện diện của mình.

Kể từ khi người dùng cài đặt phần mềm này vào thiết bị, Rootkit ngay lập tức tấn công và tước quyền quản trị. Khi này các tin tặc có thể tự do truy cập trái phép, vượt qua được bất cứ bức tường bảo vệ nào một cách dễ dàng. Đánh cấp dữ liệu, theo dõi hành vi người dùng một cách ung dung mà không có bất kỳ cảnh báo lỗi hệ thống

**Ví dụ:**

***Tên:*** Sony BMG Copy Protection Rootkit

***Loại:*** Rootkit

***Thời gian xuất hiện:*** Năm 2005

***Mô tả:*** Sony BMG, một hãng đĩa nhạc lớn, đã sử dụng rootkit như một phần của biện pháp bảo vệ bản quyền trong một số đĩa CD của họ. Rootkit này được cài đặt mà không thông báo cho người dùng khi họ chèn đĩa CD vào máy tính. Mục tiêu của nó là ẩn đi một số thay đổi không mong muốn trên hệ thống máy tính để bảo vệ các tệp nhạc khỏi việc sao chép bất hợp pháp.

***Hậu quả:*** Cộng đồng an ninh mạng phát hiện ra rằng rootkit của Sony BMG tạo ra lỗ hổng bảo mật nghiêm trọng có thể bị tận dụng bởi phần mềm độc hại khác. Ngoài ra, việc cài đặt rootkit mà không thông báo đã gây nhiều tranh cãi và dẫn đến sự phản đối mạnh mẽ từ cộng đồng người dùng.

***Phòng tránh và loại bỏ:*** Để ngăn chặn rootkit và bảo vệ hệ thống, người dùng cần sử dụng các công cụ chống rootkit, duy trì phần mềm bảo mật được cập nhật và kiểm tra định kỳ các phần mềm có thể có lỗ hổng bảo mật. Cũng quan trọng là luôn kiểm tra thông báo và điều khoản khi cài đặt phần mềm từ các nguồn không rõ.nào diễn ra.

* + 1. **Adware**

***Adware*** là loại Malware được dùng để thu thập thông tin cá nhân lưu trữ trên máy tính rồi hiển thị nhiều loại quảng cáo trên máy tính đã bị lây nhiễm. Adwarebuộc trình duyệt phải chuyển hướng đến các quảng cáo trên trang web. Các quảng cáo này cũng thường buộc người dùng download thêm các phần mềm độc hại khác.

Adware còn có thể chứa cả Trojan và Spyware. Thêm vào đó, Adware còn làm chậm máy tính của bạn rất rõ ràng.

**Ví dụ:**

***Tên:*** Superfish

***Loại:*** Adware

***Thời gian xuất hiện:*** Năm 2014

***Mô tả:*** Superfish là một phần mềm adware mà Lenovo đã cài đặt trên một số máy tính xách tay của họ từ năm 2014 đến 2015. Mục tiêu của Superfish là hiển thị quảng cáo và giúp người tiêu dùng tìm kiếm và mua sắm trực tuyến. Tuy nhiên, phần mềm này thực hiện điều này bằng cách chèn quảng cáo vào trình duyệt web mà người dùng sử dụng, thậm chí cả những trang web mà họ đang truy cập.

***Hậu quả:*** Superfish gặp phải sự phản đối mạnh mẽ từ cộng đồng người dùng vì cách nó can thiệp vào trải nghiệm duyệt web và có nguy cơ gây lỗ hổng bảo mật. Quảng cáo không mong muốn không chỉ làm phiền người dùng mà còn tăng nguy cơ cho các cuộc tấn công giả mạo và lừa đảo.

***Phòng tránh và loại bỏ:*** Để ngăn chặn adware như Superfish, người dùng cần cài đặt và duy trì một phần mềm chống malware và adware. Ngoài ra, khi cài đặt phần mềm mới, họ nên kiểm tra kỹ và từ chối cài đặt bất kỳ phần mềm kèm theo không mong muốn hoặc không cần thiết.

* + 1. **Ransomeware**

***Ransomeware*** là một loại malware có khả năng mã hóa các file trong ổ cứng và yêu cầu thanh toán (thường là bằng Bitcoin) để đổi lấy key giải mã.

Ransomware đang bùng phát mạnh mẽ trong những năm gần đây. Nếu không có được key giải mã, thì về mặt toán học, nạn nhân không thể nào lấy lại quyền truy cập vào file. Ransomware được xem là vũ khí của tội phạm mạng vì nó thường dùng các phương thức thanh toán nhanh chóng bằng tiền điện tử.

**Ví dụ:**

***Tên:*** WannaCry

***Loại:*** Ransomware

***Thời gian xuất hiện:*** Tháng 5 năm 2017

***Mô tả:*** WannaCry là một loại ransomware lan truyền chủ yếu thông qua lỗ hổng bảo mật trong hệ điều hành Windows. Nó mã hóa dữ liệu trên máy tính của người dùng và yêu cầu một khoản tiền chuộc (ransom) để cung cấp khóa giải mã. WannaCry đã sử dụng một công cụ bí mật NSA bị rò rỉ có tên là EternalBlue để lan truyền qua các máy tính chưa được cập nhật.

***Hậu quả:*** WannaCry gây ra sự cố toàn cầu với hàng trăm nghìn máy tính ở hơn 150 quốc gia bị ảnh hưởng. Các tổ chức, bao gồm cả các bệnh viện và doanh nghiệp, đã bị tắt máy tính và mất dữ liệu quan trọng. Nhiều người dùng cá nhân và doanh nghiệp đã phải đối mặt với quyết định trả tiền hoặc mất quyền truy cập vào dữ liệu của họ.

***Phòng tránh và loại bỏ:*** Để bảo vệ khỏi ransomware như WannaCry, người dùng cần thường xuyên cập nhật hệ điều hành và phần mềm antivirus của họ. Việc sao lưu dữ liệu thường xuyên cũng là một biện pháp quan trọng để phục hồi dữ liệu mà không phải trả tiền chuộc. Ngoài ra, cần hạn chế việc kích hoạt các liên kết hay tệp tin từ nguồn không rõ và kiểm tra tính bảo mật định kỳ.

* + 1. **Cryptojacking**

***Cryptojacking*** là malware nhằm mục đích yêu cầu nạn nhân cung cấp Bitcoin cho các hacker, nó cũng thường hoạt động ngầm khiến nạn nhân không thể biết được.

Cryptojacking lây nhiễm vào trong máy tính và sử dụng các chu kỳ CPU để có thể đào Bitcoin. Phần mềm này cũng có thể chạy nền trên nhiều hệ điều hành, thậm chí là dưới dạng JavaScript trong cửa sổ trình duyệt.

**Ví dụ:**

***Tên:*** Coinhive

***Loại:*** Cryptojacking

***Thời gian xuất hiện:*** Năm 2017

***Mô tả:*** Coinhive là một ví dụ điển hình về cryptojacking, một kỹ thuật mà kẻ tấn công sử dụng để lợi dụng nguồn lực máy tính của người khác để đào tiền điện tử mà không cần sự đồng ý của họ. Coinhive thường được tích hợp vào trang web hoặc ứng dụng mà người dùng truy cập mà không biết. Khi họ truy cập trang web đó, máy tính của họ sẽ bắt đầu đào tiền điện tử, đóng góp sức mạnh xử lý vào mạng blockchain của người tấn công.

***Hậu quả:*** Cryptojacking có thể dẫn đến tăng nhanh của việc tiêu tốn tài nguyên máy tính, gây giảm hiệu suất và kéo dài thời gian sử dụng pin trên các thiết bị di động. Trong môi trường doanh nghiệp, nó có thể tăng chi phí điện năng và gây ảnh hưởng đến khả năng làm việc của các hệ thống.

***Phòng tránh và loại bỏ:*** Để ngăn chặn cryptojacking, người dùng cần cập nhật và duy trì phần mềm bảo mật, sử dụng tiện ích chặn quảng cáo và mã độc hại, và tránh truy cập các trang web không an toàn. Trong môi trường doanh nghiệp, các biện pháp bảo mật mạng cũng cần được triển khai để ngăn chặn việc truy cập và hoạt động của cryptojacking.

* + 1. **Malvertising**

***Malvertising*** được biết đến là một loại quảng cáo độc hại. Loại malware này sử dụng các quảng cáo hoặc mạng quảng cáo hợp pháp, nhằm âm thầm phát tán malware đến máy tính của người dùng. Khi người dùng click vào quảng cáo đó, hacker sẽ được chuyển hướng đến một trang web độc hại, hoặc tự cài đặt malware trên máy của người dùng.

Trong một số trường hợp, malware còn có thể được nhúng vào trong quảng cáo để tự động thực thi mà không cần nạn nhân thực hiện bất kỳ hành động nào. Kỹ thuật này còn được biết đến với cái tên “drive-by download.”

**Ví dụ:**

***Tên:*** The Angler Exploit Kit

***Loại:*** Malvertising

***Thời gian xảy ra:*** Năm 2015

***Mô tả:*** Các kẻ tấn công đã sử dụng mạng quảng cáo để phát tán exploit kit này thông qua các trang web phổ biến. Khi người dùng truy cập những trang web này và tương tác với quảng cáo bị nhiễm malware, Angler Exploit Kit được triển khai để tận dụng các lỗ hổng trong trình duyệt hoặc các plugin trình duyệt chưa được cập nhật.

***Hậu quả:*** Sau khi bị nhiễm, máy tính của người dùng có thể trở thành một phần của botnet hoặc bị kiểm soát từ xa để thực hiện các hành động độc hại khác nhau, bao gồm cả việc lấy cắp thông tin cá nhân, triển khai ransomware, hoặc thậm chí tham gia vào các cuộc tấn công phân phối malware khác.

***Phòng tránh và loại bỏ:*** Để ngăn chặn malvertising, người dùng cần duy trì phần mềm antivirus cập nhật và tránh tương tác với các quảng cáo không đáng tin cậy. Cũng quan trọng là thường xuyên cập nhật trình duyệt và plugin trình duyệt để đóng các lỗ hổng bảo mật. Người quản trị hệ thống và chủ sở hữu trang web cũng cần theo dõi và kiểm soát các quảng cáo xuất hiện trên trang web của họ để ngăn chặn malvertising.

1. **Cách phòng chống Malware**

**Thứ nhất: Cập nhật thường xuyên hệ điều hành và ứng dụng**

Nếu bạn hiểu rõ về Malware là gì và cách thức tấn công của Malware là gì, ta sẽ dễ dàng nhận ra một trong những cách phổ biến mà tội phạm mạng tấn công vào thiết bị và hệ thống của người dùng chính là việc lợi dụng những lỗ hổng trong các phần mềm cũ hoặc lỗi thời.

Vì vậy hãy đảm bảo chúng ta cần thường xuyên cài đặt các bản cập nhật ngay khi chúng có sẵn, vì chúng luôn được các nhà cung cấp xá những lỗi và lỗ hổng bảo mật mà phiên bản trước còn tồn tại.

**Thứ hai: Cài đặt các phần mềm diệt virus**

Một trong những cách được coi là cơ bản nhưng không kém phần hiệu quả để bảo vệ khỏi mã độc Malware là gì? Điều đó chính là cài đặt phần mềm chống virus. Phần mềm chống virus sẽ có nhiệm vụ bảo vệ thiết bị của chúng ta khỏi các phần mềm độc hại gây nguy hiểm cho hệ thống. Nó sẽ quét máy tính của chúng ta để phát hiện và làm sạch các phần mềm gây hại, đồng thời cung cấp các bản cập nhật tự động để cung cấp khả năng bảo vệ nâng cao chống lại các virus mới được tạo ra.

**Thứ ba: Sử dụng tường lửa**

Một cách khác để bảo vệ thiết bị của chúng ta khỏi Malware chính là sử dụng tường lửa. Lợi ích của việc sử dụng tường lửa để chống lại virus Malware là gì? Đây là giải pháp giúp ngăn chặn các cuộc tấn công độc hại bằng cách chặn tất cả các truy cập trái phép hoặc từ một mạng máy tính riêng.

Ngoài phần mềm chống virus, tường lửa cũng sẽ được coi là “một tấm khiên” kiên cố để chống lại Malware xâm nhập.

**Thứ tư: Tránh nhấp vào những liên kết không xác định**

Lừa đảo vẫn là cách dễ nhất để Malware xâm nhập vào thiết bị của người dùng. Lúc này, họ có thể lừa người dùng bằng cách gửi các email hoặc một đường liên kết có thể đến từ một doanh nghiệp hợp pháp hoặc nguồn có uy tín. Chính những liên kết này có thể hướng ta đến một trang web giả mạo - nơi ta được nhắc nhập các thông tin cá nhân của mình hoặc đưa chúng ta đến một trang web lây nhiễm phần mềm độc hại trực tiếp vào máy tính của ta.

Do đó chỉ cần ta không nhấp vào các đường liên kết không xác định là cách tốt để ta không cần quan tâm đến hậu quả của Malware là gì.

**Thứ năm: Chỉ tải những ứng dụng từ nguồn đáng tin cậy**

Mua ứng dụng từ các nguồn đáng tin cậy cũng là cách làm giảm khả năng thiết bị của ta bị nhiễm Malware.

Trên thực tế, các thương hiệu lớn trên thế giới sẽ rất cẩn trọng để đảm bảo họ không làm hỏng đến danh tiếng của chính mình. Do đó, để kiểm tra tính xác thực của nguồn, người dùng có thể kiểm tra tên đầy đủ, danh sách các ứng dụng đã từng xuất bản và chi tiết liên hệ ở phần mô tả ứng dụng trong các cửa hàng ứng dụng Google Play hoặc Apple.

**Thứ sáu: Sao lưu dữ liệu thường xuyên**

Để có thể truy xuất dữ liệu và khôi phục nhanh nhất hệ thống của mình trở lại hoạt động bình thường sau khi bị Malware tấn công thì việc sao lưu thường xuyên là điều mà ta nên đặc biệt quan tâm.

Điều này sẽ giúp giảm thiểu tối đa mọi thiệt hại và đảm bảo rằng ta không phải là nạn nhân của một cuộc tấn công Ransomware.

# **CHƯƠNG 2: CÁC MÔ HÌNH HỌC MÁY SỬ DỤNG**

1. **Feedforward Neural Networks (FNN)**

* **Tác giả và Năm Xuất Bản:**

Thuật toán FNN được phát triển bởi nhóm nghiên cứu trong lĩnh vực trí tuệ nhân tạo, nhưng không có một tác giả cụ thể do nó là một khái niệm tổng quát trong machine learning và deep learning.

Feedforward Neural Networks đã được đề cập trong nhiều bài báo và sách giáo trình, ví dụ như "Deep Learning" của Ian Goodfellow, Yoshua Bengio, và Aaron Courville, xuất bản năm 2016.

* **Đối tượng bài toán:**

FNN là một kiến trúc cơ bản trong deep learning, thường được sử dụng cho các bài toán như phân loại hình ảnh, dự đoán chuỗi thời gian, và nhiều ứng dụng khác trong lĩnh vực machine learning.

* **Ý tưởng cơ bản của thuật toán FNN:**

FNN là một mô hình mạng nơ-ron có cấu trúc từ lớp đầu vào đến lớp đầu ra mà không có các chu kỳ phản hồi. Dữ liệu di chuyển qua mạng một chiều từ lớp đầu vào đến lớp đầu ra mà không có các kết nối phản hồi. Các lớp trong FNN được chia thành các tầng, bao gồm tầng đầu vào, các tầng ẩn (nếu có), và tầng đầu ra.

* **Độ phức tạp:**

Độ phức tạp của FNN phụ thuộc vào số lượng nơ-ron và số lớp trong mô hình. Đối với một FNN có kích thước lớn, độ phức tạp có thể tăng lên đáng kể, đặc biệt là trong quá trình huấn luyện.

* **Ứng dụng của FNN:**

FNN được sử dụng rộng rãi trong nhiều lĩnh vực ứng dụng. Ví dụ, trong lĩnh vực xử lý ảnh, FNN thường được sử dụng để nhận diện đối tượng và phân loại ảnh. Trong tài chính, FNN được áp dụng để dự đoán giá cổ phiếu và phân tích dữ liệu tài chính. Ngoài ra, FNN cũng được sử dụng trong ngôn ngữ tự nhiên xử lý và nhiều ứng dụng khác.

* **FNN trong phòng chống Malware:**

FNN có thể được sử dụng để xây dựng các mô hình phân loại malware.

Dữ liệu đầu vào có thể bao gồm các đặc điểm của các tập tin hoặc quá trình, và mô hình có thể học từ dữ liệu để phân loại chúng là độc hại hay không.

1. **Random Forest**

* **Tác giả và Năm Xuất Bản:**

Thuật toán Random Forest (RF) không có một tác giả cụ thể do đây là một thuật toán tập hợp được phát triển từ sự kết hợp của nhiều cây quyết định (decision trees). Khái niệm về Random Forest đã được giới thiệu bởi Leo Breiman trong bài báo "Random Forests" vào năm 2001.

* **Dùng cho bài toán:**

Random Forest là một thuật toán rất linh hoạt và được sử dụng rộng rãi cho các bài toán phân loại và hồi quy trong machine learning. Nó thường được áp dụng trong các ứng dụng như dự đoán giá cổ phiếu, phân loại ảnh y tế, và nhiều lĩnh vực khác.

* **Ý tưởng cơ bản của thuật toán Random Forest:**

Random Forest là một tập hợp của nhiều cây quyết định. Mỗi cây quyết định được xây dựng dựa trên một tập dữ liệu con được chọn ngẫu nhiên từ tập dữ liệu huấn luyện. Sau đó, kết quả từ nhiều cây được kết hợp để tạo ra một dự đoán cuối cùng. Quyết định được đưa ra bằng cách bình chọn (đối với phân loại) hoặc lấy trung bình (đối với hồi quy) từ các cây con.

* **Độ phức tạp:**

Độ phức tạp của Random Forest phụ thuộc vào số lượng cây trong tập hợp và số lượng thuộc tính được sử dụng cho mỗi cây. Tuy nhiên, nhờ tính chất của thuật toán, RF thường có hiệu suất tốt và ít yêu cầu tinh chỉnh**.**

* **Ứng dụng của Random Forest:**

RF được sử dụng rộng rãi trong nhiều lĩnh vực, chẳng hạn như trong y học để dự đoán bệnh lý, trong tài chính để đưa ra quyết định đầu tư, và trong marketing để phân loại khách hàng. Điều này là do khả năng mạnh mẽ của RF trong việc xử lý dữ liệu phức tạp và khả năng làm việc tốt với các tập dữ liệu lớn.

* **Random Forest trong phòng chống Malware:**

Random Forest là một thuật toán học máy ensemble, tức là nó kết hợp nhiều decision trees để đưa ra dự đoán chung.

Random Forest có thể phân loại mẫu dữ liệu thành các lớp khác nhau, như là malware hoặc không phải malware, dựa trên các đặc trưng của chúng.

1. **Logistic Regression**

* **Tác giả và Năm Xuất Bản:**

Thuật toán Logistic Regression (LR) được phát triển trong thống kê và machine learning, không có một tác giả cụ thể cho thuật toán này.

Nguyên tắc cơ bản của Logistic Regression đã xuất hiện từ những nghiên cứu về phân tích hồi quy trong thống kê từ thập kỷ thứ 18 và sau đó được phát triển để áp dụng trong machine learning.

* **Dùng cho bài toán:**

Logistic Regression thường được sử dụng cho các bài toán phân loại, đặc biệt là trong các tình huống khi biến phụ thuộc là một biến nhị phân, tức chỉ có hai lớp.

* **Ý tưởng cơ bản của thuật toán Logistic Regression:**

Logistic Regression là một thuật toán phân loại dựa trên khả năng của mô hình hồi quy tuyến tính. Nó sử dụng hàm logistic để chuyển đổi đầu ra của mô hình từ một dạng liên tục thành xác suất, giúp dự đoán xác suất rơi vào một lớp cụ thể.

* **Độ phức tạp:**

Độ phức tạp của Logistic Regression là tương đối thấp. Quá trình huấn luyện nhanh chóng với dữ liệu lớn và không đòi hỏi nhiều tài nguyên tính toán.

* **Ứng dụng của Logistic Regression:**

LR thường được sử dụng trong các lĩnh vực như y học để dự đoán xác suất mắc bệnh, trong tài chính để đánh giá rủi ro tín dụng, và trong marketing để dự đoán xác suất mua hàng của khách hàng. Điều này là do khả năng LR giúp giải quyết các vấn đề phân loại một cách hiệu quả và hiểu quả.

* **Logistic Regression trong phòng chống Malware:**

Logistic Regression là một thuật toán học máy thường được sử dụng trong phân loại và dự đoán, thường áp dụng cho bài toán phân loại hai lớp

Trong phòng chống malware, Logistic Regression để phân loại mẫu dữ liệu thành malware hoặc không phải malware, dựa trên đặc trưng hoặc thuộc tính của chúng.

1. **Gated Recurrent Units ( GRU )**

* **Tác giả và Năm Xuất Bản:**

Thuật toán Gated Recurrent Unit (GRU) đã được phát triển bởi Chung et al. và công bố trong bài báo có tựa đề "Empirical Evaluation of Gated Recurrent Networks on Sequence Modeling" vào năm 2014.

Thuật toán Gated Recurrent Unit (GRU) đã được phát triển bởi Chung et al. và công bố trong bài báo có tựa đề "Empirical Evaluation of Gated Recurrent Networks on Sequence Modeling" vào năm 2014.

* **Dùng cho bài toán:**

GRU thường được sử dụng trong các bài toán liên quan đến xử lý dãy thời gian, như dự đoán chuỗi thời gian, ngôn ngữ tự nhiên, và nhiều ứng dụng khác trong lĩnh vực machine learning.

* **Ý tưởng cơ bản của thuật toán GRU:**

GRU là một kiến trúc mạng nơ-ron hồi quy (RNN) cải tiến với cổng (gate) để kiểm soát việc lưu trữ thông tin. Nó giúp giải quyết vấn đề biến mất gradient trong RNN thông thường bằng cách duy trì một cơ chế tự động kiểm soát việc cập nhật thông tin từ quá khứ.

* **Độ phức tạp:**

Độ phức tạp của GRU tương đối cao, đặc biệt là khi áp dụng trong các mô hình có kích thước lớn và với dữ liệu lớn. Tuy nhiên, nó thường hiệu quả hơn so với RNN truyền thống.

* **Ứng dụng của GRU:**

GRU thường được sử dụng trong ngôn ngữ tự nhiên xử lý, máy dịch, và dự đoán chuỗi thời gian như dự báo thời tiết. Nó cũng có ứng dụng trong nhiều lĩnh vực khác yêu cầu mô hình hiểu và dự đoán dữ liệu theo thời gian.

* **GRU trong phòng chống Malware:**

Trong phòng chống malware GRU dùng để phân loại và dự đoán hành vi độc hại dựa trên chuỗi hành vi.

GRU giúp giải quyết vấn đề biến mất gradient, có khả năng nhớ thông tin từ các bước trước, từ đó cung cấp cảnh báo sớm và ngăn chặn các hoạt động độc hại.

1. **Recurrent Neural Network ( RNN )**

* **Tác giả và Năm Xuất Bản:**

Thuật toán Recurrent Neural Network (RNN) có xuất xứ từ công trình của Jordan và Rumelhart đã xuất bản trong bài báo "Forward models: Supervised learning with a distal teacher" vào năm 1986. Tuy nhiên, thuật toán RNN đã được phát triển và mở rộng từ đó đến nay bởi nhiều nhà nghiên cứu.

* **Dùng cho bài toán:**

RNN thường được sử dụng trong các bài toán liên quan đến dãy thời gian, như dự đoán chuỗi, ngôn ngữ tự nhiên, và nhiều ứng dụng khác trong lĩnh vực machine learning.

* **Ý tưởng cơ bản của thuật toán RNN:**

RNN là một kiến trúc mạng nơ-ron có khả năng lưu trữ thông tin từ quá khứ và sử dụng nó trong tương lai. Mỗi đơn vị trong RNN có trạng thái ẩn, giúp nó "nhớ" thông tin trước đó và chuyển tiếp thông tin qua các bước thời gian.

* **Độ phức tạp:**

Độ phức tạp của RNN có thể tăng lên đáng kể khi mô hình đối mặt với các chuỗi dữ liệu dài. Nó cũng gặp vấn đề biến mất gradient khi xử lý các chuỗi dài, làm giảm hiệu suất.

* **Ứng dụng của RNN:**

RNN được sử dụng trong nhiều lĩnh vực như dự đoán chuỗi thời gian (ví dụ: dự báo tình trạng thời tiết), ngôn ngữ tự nhiên xử lý (ví dụ: dịch máy), và trong robot học để mô phỏng hành vi dựa trên chuỗi thời gian.

Tuy nhiên, do vấn đề biến mất gradient, nhiều biến thể của RNN, như Long Short-Term Memory (LSTM) và Gated Recurrent Unit (GRU), đã được phát triển để cải thiện hiệu suất.

* **RNN trong phòng chống Malware:**

Trong phòng chống malware, RNN được sử dụng để phân loại và dự đoán hành vi độc hại dựa trên dữ liệu chuỗi như chuỗi hành vi ứng dụng hoặc tệp tin. Theo dõi các mẫu hành vi, phân tích ngôn ngữ và giao tiếp của malware, dự đoán hành vi tương lai, phân loại ngữ cảnh, và cung cấp khả năng dự đoán tiến triển của cuộc tấn công.

# **CHƯƠNG 3: TRỰC QUAN KẾT QUẢ THỰC HIỆN**

1. **Giới thiệu dataset**

***Tên dataset:*** Malware Detection

***Mô tả:*** Tập dữ liệu được sử dụng để xây dựng và phát triển các mô hình máy học với mục đích cảnh báo khi trong file hoặc chương trình xuất hiện malware.

Với 100.000 dòng dữ liệu và 35 cột thuộc tính mô tả các đặc điểm đầu vào và đầu ra để nhận biết một file có xuất hiện malware hay không.

***Link dataset:*** <https://www.kaggle.com/datasets/nsaravana/malware-detection>

***Mô tả thuộc tính:***

| Thuộc tính | Mô tả |
| --- | --- |
| hash | Tên tập tin |
| millisecond | Thời gian |
| classification | Phân loại: malware, benign |
| state | Trạng thái |
| usage\_counter | Bộ đếm sử dụng cấu trúc nhiệm vụ |
| prio | Mức độ ưu tiên động của một quá trình |
| static\_prio | Mức độ ưu tiên tĩnh của một quá trình |
| normal\_prio | Ưu tiên mà không tính đến tính kế thừa RT |
| policy | Chính sách lập kế hoạch của quá trình |
| vm\_pgoff | Phần bù của vùng trong tệp, tính bằng trang |
| vm\_truncate\_count | Được sử dụng để đánh dấu một vma đã được xử lý |
| task\_size | Kích thước của nhiệm vụ hiện tại |
| cached\_hole\_size | Kích thước của lỗ không gian địa chỉ miễn phí |
| free\_area\_cache | Lỗ không gian địa chỉ đầu tiên |
| mm\_users | Người dùng không gian địa chỉ |
| map\_count | Số vùng nhớ |
| hiwater\_rss | Đỉnh của kích thước tập hợp thường trú |
| total\_vm | Tổng số trang |
| shared\_vm | Số trang được chia sẻ |
| exec\_vm | Số trang thực thi được |
| reserved\_vm | Số trang dành riêng |
| nr\_ptes | Số mục trong bảng trang |
| end\_data | Địa chỉ cuối của thành phần mã |
| last\_interval | Khoảng thời gian cuối cùng trước khi đập |
| nvcsw | Số lượng chuyển đổi bối cảnh tình nguyện |
| nivcsw | Số lượng chuyển đổi bối cảnh tự nguyện |
| min\_flt | Lỗi trang nhỏ |
| maj\_flt | Lỗi trang lớn |
| fs\_excl\_counter | Chứa các tài nguyên độc quyền của hệ thống tập tin. |
| lock | Khóa đồng bộ hóa read-write được sử dụng để truy cập hệ thống tệp |
| utime | Thời gian của người dùng |
| stime | Thời gian hệ thống |
| gtime | Thời gian của khách |
| cgtime | Thời gian nhóm tích lũy. Bộ đếm tài nguyên tích lũy |
| signal\_nvcsw | Bộ đếm tài nguyên tích lũy |

1. **Quá trình thực hiện** 
   1. **Nguồn dữ liệu**

* **Import các thư viện cần thiết**

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

* **Đọc dữ liệu vào Dataframe**

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

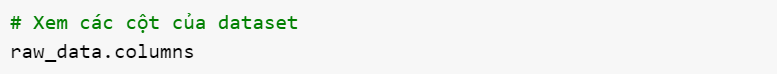
Mô tả được tạo tự động

Kết quả:

Ảnh có chứa văn bản, Phông chữ, hàng, số

Mô tả được tạo tự động

* **Xem các cột của Dataset**



Kết quả:

Ảnh có chứa văn bản, Phông chữ, biên lai, màu trắng

Mô tả được tạo tự động

* **Thống kê các độ đo trên tất cả các cột**

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình, màu trắng

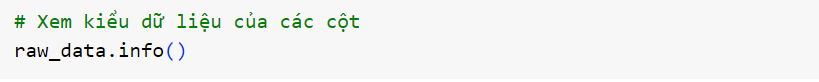
Mô tả được tạo tự động

Kết quả:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

* **Xem kiểu dữ liệu của các cột**



Kết quả:

Ảnh có chứa văn bản, thực đơn, tài liệu, ảnh chụp màn hình

Mô tả được tạo tự động

* 1. **Tiền xử lý dữ liệu**

1. **Xem sự tương quan giữa 2 giá trị trong cột kết quả "classification"**

* **Đếm số lượng dòng ứng với mỗi giá trị của cột "classification"**

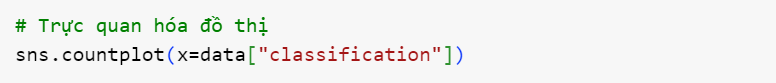
Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình, màu trắng

Mô tả được tạo tự động

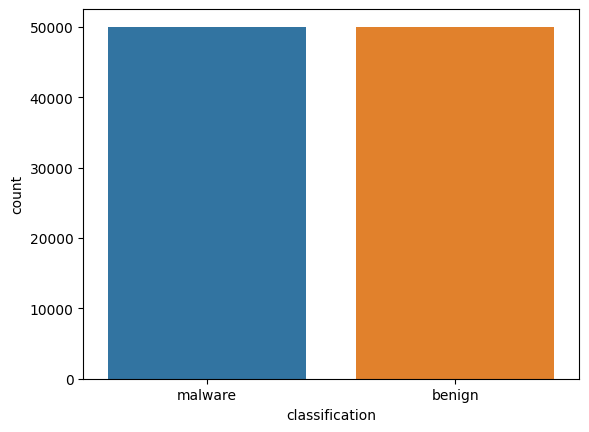
Kết quả:



* **Trực quan hóa đồ thị**



Kết quả:

****

Dựa vào đồ thị, ta thấy được số lượng dòng dữ liệu cân bằng giữa hai nhóm "malware" và "begin" là quan trọng trong phát triển mô hình máy học để tránh tình trạng chệch (bias) và để mô hình có khả năng tổng quát hóa tốt hơn trên cả hai loại nhãn => Từ đó có thể giúp mô hình học hiệu quả từ cả hai loại nhãn trên

1. **Xem xét các giá trị bị Null**



Kết quả:

Ảnh có chứa văn bản, ảnh chụp màn hình, thực đơn, Phông chữ

Mô tả được tạo tự động

Không xuất hiện giá trị NULL trong tất cả cột dữ liệu

1. **Xem xét các giá trị bị Duplicate**



Kết quả:

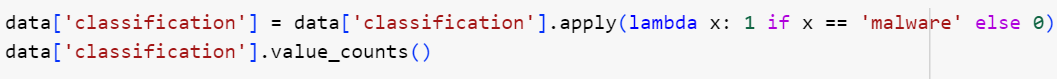


Không có dòng dữ liệu nào bị duplicate trong tập dữ liệu

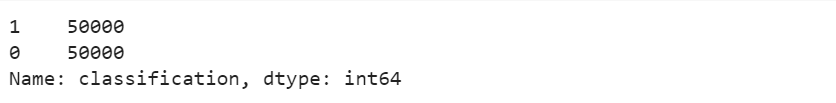
1. **Chuẩn hóa dữ liệu cột "classification"**

Thay thế giá trị "malware" thành 1

Thay thế giá trị "begin" thành 0



Kết quả:



1. **Mức độ tương quan giữa các cột dữ liệu**

Việc tìm mức độ tương quan giữa các cột dùng để xác định mối quan hệ và khả năng ảnh hưởng lẫn nhau giữa chúng

* **Trực quan hóa biểu đồ sự tương quan giữa 35 cột**

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Mô tả được tạo tự động

Kết quả:

Ảnh có chứa văn bản, ảnh chụp màn hình, hình vuông, Song song

Mô tả được tạo tự động

Dựa vào đồ thị heatmap, ta thấy được rằng có 11 cột không có sự tương quan với các cột dữ liệu còn lại

* **Xóa những cột dữ liệu không có sự tương quan**

Ảnh có chứa văn bản, Phông chữ, hàng, ảnh chụp màn hình

Mô tả được tạo tự động

* **Kiểm tra lại số cột sau khi xóa**



Kết quả:



* **Trực quan hóa biểu đồ sự tương quan giữa 23 cột**

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

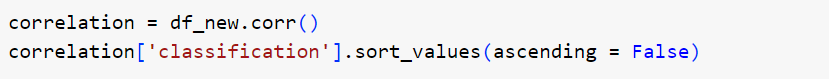
Mô tả được tạo tự động

Kết quả:

Ảnh có chứa mẫu, ảnh chụp màn hình, hình vuông, Hình chữ nhật

Mô tả được tạo tự động

* **Sự tương quan giữa các cột so với cột kết quả "classification"**



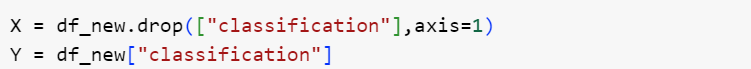
Kết quả:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, thực đơn

Mô tả được tạo tự động

* 1. **Chuẩn bị dữ liệu**

1. **Tách dữ liệu thành các tập X (đầu vào) và tập Y(đầu ra)**



Kết quả:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động



Kết quả:

Ảnh có chứa văn bản, biên lai, Phông chữ, màu trắng

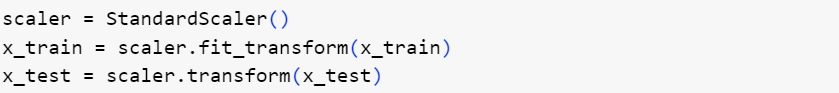
Mô tả được tạo tự động

1. **Chia dữ liệu thành tập Train (huấn luyện mô hình) và Test (để kiểm tra hiệu suất mô hình)**

Tỷ lệ: Train (80%) - Test (20%)



1. **Chuẩn hóa dữ liệu các tập đầu vào x\_train và x\_test**



* 1. **Xây Dựng Và Đánh Giá Mô Hình**
     1. **Mô hình Feedforward Neural Networks (FNN)**

Tiến hành thực hiện train model với 5 kịch bản khác nhau

| Scenarios | Scenarios 1 | Scenarios 2 | Scenarios 3 | Scenarios 4 | Scenarios 5 |
| --- | --- | --- | --- | --- | --- |
| Learning rate | 0.1 | 0.01 | 0.01 | 0.01 | 0.01 |
| Activation Function | Sigmoid | Softmax | Sigmoid | Sigmoid | Sigmoid |
| Loss function | binary\_  crossentropy | binary\_  crossentropy | binary\_  crossentropy | binary\_  crossentropy | binary\_  crossentropy |
| Epoch | 10 | 20 | 10 | 10 | 20 |
| Batch Size | 32 | 32 | 32 | 32 | 32 |
| Dataset Split | 80% - 20% | 80% - 20% | 80% - 20% | 90% - 10% | 90% - 10% |
| Dense Layer | 256 | 64 | 64 | 64 | 64 |
| Output Execution Time | 1.3036s | 0.8213s | 1.3630s | 1.0761s | 1.1324s |
| Training Accuracy | 99.75 % | 50.11 % | 99.96 % | 99.95 % | 99.89 % |
| Validation Accuracy | 99.98 % | 49.76 % | 99.93 % | 99.88 % | 99.96 % |
| Testing Accuracy | 99.95 % | 49.82 % | 99.94 % | 99.86 % | 99.95 % |

Dựa vào các chỉ số đánh giá độ chính xác của mô hình trên tập train, test và validation. Có thể thấy được, mô hình FNN với kịch bản 1 (Scenarios 1) cho ra kết quả với độ chính xác tốt nhất.

* **Kịch bản 1 ( Kịch bản có Testing Accuracy cao nhất )**
* **Xây dựng mô hình**

A screen shot of a computer

Description automatically generated

Kết quả:

Ảnh có chứa văn bản, ảnh chụp màn hình, số, Phông chữ

Mô tả được tạo tự động

* **Cấu hình mô hình**

**A screenshot of a computer

Description automatically generated**

* **Huấn luyện mô hình**

Ảnh có chứa văn bản, Phông chữ, đại số, ảnh chụp màn hình

Mô tả được tạo tự động

Kết quả:

Ảnh có chứa văn bản, biên lai, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

* **Trực quan hóa kết quả của quá trình huấn luyện**

**A screenshot of a computer code

Description automatically generated**

Kết quả:

A screenshot of a graph

Description automatically generated

* **Trực quan hóa mối quan hệ giữa độ chính xác và các epoch**

**A screenshot of a computer

Description automatically generated**

Kết quả:

A graph with blue lines and numbers

Description automatically generated

* **Ma trận nhầm lẫn (confusion)**

**A computer code on a white background

Description automatically generated**

Kết quả:

A blue and white graph

Description automatically generated

* Dựa vào ma trận nhầm lẫn, ta có thể thấy được mô hình FNN với tỉ lệ train (80%) và test (20%) cho ra kết quả dự đoán chính xác khoảng 50% trên hai nhãn là không bị malware (0) và bị malware (1).
* Số lượng kết quả dự đoán đúng là không bị malware (0) và kết quả thực tế cũng là không bị malware (0) là 10038 giá trị.
* Tuy nhiên, số lượng kết quả dự đoán sai là không bị malware (0) nhưng kết quả thực tế là bị malware (1) là 9962 giá trị.
* **Trực quan kết quả trên đường AUC, ROC**

**A computer code on a white background

Description automatically generated**

Kết quả:

A line graph with blue and red dots

Description automatically generated

* Đường cong ROC đi qua trung tâm của biểu đồ. Điều này cho thấy mô hình có khả năng phân loại đúng và sai các trường hợp positive và negative như nhau nhưng với độ chính xác không cao.
* TPR = 0,5 => Điều này có nghĩa là mô hình phát hiện đúng được 50% các trường hợp positive.
* FPR = 0,5 => Điều này có nghĩa là mô hình có 50% các trường hợp negative bị phân loại sai thành positive.
  + 1. **Mô hình Random Forest**
* **Chia tỷ lệ:**

**+**  train(80%) - test (20%)

**+**  train(70%) - test (30%)

* **Xây dựng mô hình**

**A close-up of a computer code

Description automatically generated**

* **Huấn luyện mô hình**

**A screenshot of a computer

Description automatically generated**

* **Đánh giá mô hình với train (80%) - test (20%)**

**A close-up of a computer code

Description automatically generated**

Kết quả:



* **Đánh giá mô hình với train (70%) - test (30%)**

A screenshot of a computer code

Description automatically generated

* **Ma trận nhầm lẫn (confusion) với train (80%) - test (20%)**

**A computer code on a white background

Description automatically generated**

A blue squares with white numbers

Description automatically generated

* Dựa vào ma trận nhầm lẫn, ta có thể thấy được mô hình Random Forest với tỉ lệ train (80%) và test (20%) cho ra kết quả dự đoán chính xác 100% trên hai nhãn là không bị malware (0) và bị malware (1).
* Số lượng kết quả dự đoán đúng là bị malware (1) và kết quả thực tế cũng là bị malware (1) là 9962 giá trị.
* Số lượng kết quả dự đoán đúng là không bị malware (0) và kết quả thực tế cũng là không bị malware (0) là 10038 giá trị.
* **Ma trận nhầm lẫn (confusion) với train (70%) - test (30%)**

**A computer screen shot of a code

Description automatically generated**

A blue squares with white numbers

Description automatically generated

* Dựa vào ma trận nhầm lẫn, ta có thể thấy được mô hình Random Forest với tỉ lệ train (70%) và test (30%) cho ra kết quả dự đoán gần như chính xác 100% trên hai nhãn là không bị malware (0) và bị malware (1).
* Số lượng kết quả dự đoán đúng là bị malware (1) và kết quả thực tế cũng là bị malware (1) là 14947 giá trị.
* Số lượng kết quả dự đoán đúng là không bị malware (0) và kết quả thực tế cũng là không bị malware (0) là 15050 giá trị.
* Tuy nhiên, số lượng kết quả dự đoán sai là không bị malware (0) và kết quả thực tế là bị malware (1) là 3 giá trị.
* **Trực quan kết quả trên đường AUC, ROC với train (80%) - test (20%)**

**A computer screen shot of a computer code

Description automatically generated**

Kết quả:

A graph with a line and a point

Description automatically generated with medium confidence

* Dựa vào biểu đồ, ta thấy đường cong ROC nằm gần góc trên bên trái của biểu đồ. Điều này cho thấy mô hình có thể phân loại chính xác hầu hết các trường hợp dương tính và âm tính, bất kể ngưỡng phân loại.
* TPR = 1 => Điều này có nghĩa là mô hình có thể phát hiện đúng tất cả các trường hợp là positive.
* FPR = 0 => Điều này có nghĩa là mô hình không có trường hợp positive nào là sai.
* **Trực quan kết quả trên đường AUC, ROC với train (70%) - test (30%)**

A computer screen shot of a program

Description automatically generated

Kết quả:

A graph with a line and a point

Description automatically generated with medium confidence

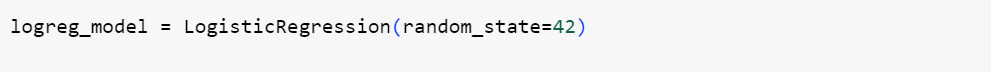
Tương tự như biểu đồ ROC và AUC của **train (80%) - test (20%)**, mô hình có thể phân loại chính xác hầu hết các trường hợp dương tính và âm tính, bất kể ngưỡng phân loại.

* + 1. **Mô hình Logistic Regression**
* **Chia tỷ lệ:**

+ train(80%) - test (20%)

+ train(70%) - test (30%)

* **Xây dựng mô hình**



* **Huấn luyện mô hình**

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Mô tả được tạo tự động

* **Đánh giá mô hình với train (80%) - test (20%)**

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình, hàng

Mô tả được tạo tự động

Kết quả:



* **Đánh giá mô hình với train (70%) - test (30%)**

A screenshot of a computer code

Description automatically generated

Kết quả:

A screenshot of a computer code

Description automatically generated

* **Ma trận nhầm lẫn (confusion) với train (80%) - test (20%)**

**A computer code with text

Description automatically generated with medium confidence**

Kết quả:

A blue squares with white squares

Description automatically generated

* Dựa vào ma trận nhầm lẫn, ta có thể thấy được mô hình Logistic Regression với tỉ lệ train (80%) và test (20%) cho ra kết quả dự đoán trên hai nhãn là không bị malware (0) và bị malware (1).
* **Số lượng kết quả dự đoán đúng** là bị malware (1) và kết quả thực tế cũng là bị malware (1) là 9503 giá trị. Tuy nhiên, **số lượng kết quả dự đoán sai** là bị malware (1) nhưng kết quả thực tế cũng là không bị malware (0) là 710 giá trị.
* **Số lượng kết quả dự đoán đúng** là không bị malware (0) và kết quả thực tế cũng là không bị malware (0) là 9328 giá trị. Tuy nhiên, **số lượng kết quả dự đoán sai** là không bị malware (0) nhưng kết quả thực tế cũng là bị malware (1) là 459 giá trị.
* **Ma trận nhầm lẫn (confusion) với train (70%) - test (30%)**

**A computer screen shot of a code

Description automatically generated**

Kết quả:

A blue squares with white numbers

Description automatically generated

* Dựa vào ma trận nhầm lẫn, ta có thể thấy được mô hình Logistic Regression với tỉ lệ train (70%) và test (30%) cho ra kết quả dự đoán trên hai nhãn là không bị malware (0) và bị malware (1).
* **Số lượng kết quả dự đoán đúng** là bị malware (1) và kết quả thực tế cũng là bị malware (1) là 14272 giá trị. Tuy nhiên, **số lượng kết quả dự đoán sai** là bị malware (1) nhưng kết quả thực tế cũng là không bị malware (0) là 1099 giá trị.
* **Số lượng kết quả dự đoán đúng** là không bị malware (0) và kết quả thực tế cũng là không bị malware (0) là 13951 giá trị. Tuy nhiên, **số lượng kết quả dự đoán sai** là không bị malware (0) nhưng kết quả thực tế cũng là bị malware (1) là 678 giá trị.
* **Trực quan kết quả trên đường AUC, ROC với train (80%) - test (20%)**

**A screenshot of a computer program

Description automatically generated**

Kết quả:

A graph of a function

Description automatically generated with medium confidence

* Đường cong ROC trong hình cho thấy mô hình phân loại có hiệu suất tốt, với diện tích dưới đường cong (AUC) bằng 0,98. Nghĩa là mô hình có khả năng phân loại chính xác các trường hợp dương tính và âm tính tốt hơn nhiều so với ngẫu nhiên.
* TPR = 0,98 => mô hình phát hiện được 98% các trường hợp positive.
* FPR = 0,02. Điều này có nghĩa là mô hình chỉ có 2% các trường hợp negative bị phân loại sai thành positive.
* **Trực quan kết quả trên đường AUC, ROC với train (70%) - test (30%)**

A screenshot of a computer code

Description automatically generated

Kết quả:

A graph of a function

Description automatically generated with medium confidence

* Đường cong ROC trong hình cho thấy mô hình phân loại có hiệu suất tốt, với diện tích dưới đường cong (AUC) bằng 0,98. Điều này có nghĩa là mô hình có khả năng phân loại chính xác các trường hợp dương tính và âm tính tốt hơn nhiều so với ngẫu nhiên.
* TPR = 0,98 => mô hình phát hiện được 98% các trường hợp positive.
* FPR = 0,02. Điều này có nghĩa là mô hình chỉ có 2% các trường hợp negative bị phân loại sai thành positive.
  + 1. **Mô hình Gated Recurrent Unit (GRU)**

Tiến hành thực hiện train model với 5 kịch bảng khác nhau.

| Scenarios | Scenarios 1 | Scenarios 2 | Scenarios 3 | Scenarios 4 | Scenarios 5 |
| --- | --- | --- | --- | --- | --- |
| Activation Function | Sigmoid | Softmax | Sigmoid | Sigmoid | Sigmoid |
| Loss function | binary\_  crossentropy | binary\_  crossentropy | binary\_  crossentropy | binary\_  crossentropy | binary\_  crossentropy |
| Epoch | 10 | 20 | 10 | 10 | 20 |
| Batch Size | 32 | 32 | 32 | 32 | 32 |
| Dataset Split | 80% - 20% | 80% - 20% | 80% - 20% | 90% - 10% | 90% - 10% |
| Dense Layer | 256 | 64 | 64 | 64 | 64 |
| Output Execution Time | 3.64 | 5.24 | 2.20 | 1.28 | 1.98 |
| Training Accuracy | 0.999 | 0.501 | 0.999 | 0.999 | 0.999 |
| Validation Accuracy | 0.998 | 0.497 | 0.963 | 0.999 | 0.998 |
| Testing Accuracy | 99.77% | 49.81% | 96.06% | 100% | 99.91% |

Dựa vào các chỉ số đánh giá độ chính xác của mô hình trên tập train, test và validation. Có thể thấy được, mô hình GRU với kịch bản 4 **(Scenarios 4)** cho ra kết quả với độ chính xác tốt nhất.

* **Kịch bản 4 ( Kịch bản có Testing Accuracy cao nhất )**
* **Xây dụng mô hình**

**A screenshot of a computer program

Description automatically generated**

Kết quả:

**A screenshot of a computer program

Description automatically generated**

* **Cấu hình mô hình**

**A computer code with black text

Description automatically generated with medium confidence**

* **Huấn luyện mô hình**

**A computer code with black text

Description automatically generated**

Kết quả:

**A close-up of a number

Description automatically generated**

* **Trực quan hóa quá trính huấn luyện**

**A computer screen shot of a program

Description automatically generated**

Kết quả:

**A graph of a train history and a train history

Description automatically generated**

* **Trực quan hóa mối quan hệ giữa độ chính xác và các epoch**

**A screenshot of a computer code

Description automatically generated**

Kết quả:

**A graph of a line

Description automatically generated**

* **Ma trận nhầm lẫn (Confusion)**

**A computer screen shot of a code

Description automatically generated**

Kết quả:

**A screenshot of a graph

Description automatically generated**

* **Trực quan hóa kết quả trên đường AUC, ROC**

**A screenshot of a computer code

Description automatically generated**

Kết quả:

**A graph of a line

Description automatically generated**

* + 1. **Mô hình Recurrent Neural Network (RNN)**

Tiến hành thực hiện train model với 5 kịch bảng khác nhau:

| Scenarios | Scenarios 1 | Scenarios 2 | Scenarios 3 | Scenarios 4 | Scenarios 5 |
| --- | --- | --- | --- | --- | --- |
| Activation Function | Sigmoid | Softmax | Sigmoid | Sigmoid | Sigmoid |
| Loss function | binary\_  crossentropy | binary\_  crossentropy | binary\_  crossentropy | binary\_  crossentropy | binary\_  crossentropy |
| Epoch | 10 | 20 | 10 | 10 | 20 |
| Batch Size | 32 | 32 | 32 | 32 | 32 |
| Dataset Split | 80% - 20% | 80% - 20% | 80% - 20% | 90% - 10% | 90% - 10% |
| Dense Layer | 256 | 64 | 64 | 64 | 64 |
| Output Execution Time | 2.81 | 2.95 | 1.27 | 1.63 | 0.9 |
| Training Accuracy | 0.99 | 0.50 | 1 | 0.999 | 0.999 |
| Validation Accuracy | 0.99 | 0.497 | 0.99 | 0.999 | 0.998 |
| Testing Accuracy | 99.98% | 49.72% | 99.99% | 100% | 99.86% |

Kết quả từ bảng trên, ta thấy kịch bản 4 cho ra kết quả tốt nhất

* **Kịch bản 4 (kịch bản có Testing Accuracy cao nhất)**
* **Xây dụng mô hình**

**A screenshot of a computer

Description automatically generated**

Kết quả:

**A screenshot of a computer

Description automatically generated**

* **Cấu hình mô hình**

Ảnh có chứa văn bản, Phông chữ, hàng, ảnh chụp màn hình

Mô tả được tạo tự động

* **Huấn luyện mô hình**

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

Kết quả:

**A close-up of a table

Description automatically generated**

* **Trực quan hóa kết quả của quá trính huấn luyện**

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

Kết quả:

**A graph of a train history

Description automatically generated with medium confidence**

* **Trực quan hóa mối quan hệ giữa độ chính xác và các epoch**

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

Kết quả:

**A graph of a line

Description automatically generated**

* **Ma trận nhầm lẫn (Confusion)**

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

Kết quả:

**A blue and white graph

Description automatically generated**

* **Trực quan hóa kết quả trên đường AUC, ROC**

**A computer code with text

Description automatically generated**

Kết quả:

**A graph of a line

Description automatically generated**

* 1. **So sánh độ chính xác giữa các mô hình**

A screenshot of a computer code

Description automatically generated

Kết quả:

A bar chart with different colored rectangles

Description automatically generated

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

Kết quả:

Ảnh có chứa văn bản, ảnh chụp màn hình, Hình chữ nhật, biểu đồ

Mô tả được tạo tự động

* Dựa vào 2 biểu đồ, ta có thể thấy được với tỉ lệ train (80%) và test (20%)
* Mô hình Random Forest, GRU, RNN cho ra độ chính xác trên tập test là cao nhất với 100%
* Tiếp theo đó là mô hình FNN cho ra độ chính xác trên tập test là 99.95%
* Cuối cùng là mô hình Logistic Regression cho ra độ chính xác trên tập test là 94.16%
* Nhìn chung, cả năm mô hình đều cho ra kết quả với độ chính xác khá cao, đều trên 90%.