BÁO CÁO TỔNG KẾT ĐỒ ÁN MÔN HỌC

**Môn học: Cơ chế hoạt động của mã độc**

**Tên chủ đề: Fileless malware detection**

*Mã nhóm:* ***G13*** *Mã đề tài:* ***S15***

**Lớp**: NT230.N22.ATCL

1. **THÔNG TIN THÀNH VIÊN NHÓM:**

*(Sinh viên liệt kê tất cả các thành viên trong nhóm)*

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Email** |
| 1 | Trần Quốc Đạt | 20521179 | 20521179@gm.uit.edu.vn |
| 2 | Nguyễn Minh Huy | 20520545 | 20520545@gm.uit.edu.vn |

1. **TÓM TẮT NỘI DUNG THỰC HIỆN:[[1]](#footnote-1)**
   1. **Chủ đề nghiên cứu trong lĩnh vực Mã độc:** *(chọn nội dung tương ứng bên dưới)*

🗹 Phát hiện mã độc

🞎 Đột biến mã độc

🞎 Khác: …………………………………

* 1. **Liên kết lưu trữ mã nguồn của nhóm:**

Mã nguồn của đề tài đồ án được lưu tại: https://drive.google.com/drive/folders/1QS6Sz\_KPS90RcMHyx1dRE2huUhiqW\_1r?usp=sharing

*(Lưu ý: GV phụ trách phải có quyền truy cập nội dung trong Link)*

* 1. **Tên bài báo tham khảo chính:**

|  |
| --- |
| Fang, Yong, Xiangyu Zhou, and Cheng Huang. "Effective method for detecting malicious PowerShell scripts based on hybrid features☆." Neurocomputing 448 (2021): 30-39. |

* 1. **Dịch tên Tiếng Việt cho bài báo:**

|  |
| --- |
| Phương pháp hiệu quả để phát hiện các tập lệnh PowerShell độc hại dựa trên các tính năng lai |

* 1. **Tóm tắt nội dung chính:**

|  |
| --- |
| Nội dung bài báo đề cập đến **03** kỹ thuật chính bao gồm:  **Kỹ thuật 1 (Trích xuất thuộc tính thủ công):** Nhóm tác giả thấy rằng không có nghiên cứu nào về phân tích mức chức năng tập lệnh để phát hiện các tập lệnh PowerShell, mà là mức chức năng của PowerShell cần phải hữu ích để phát hiện các tập lệnh PowerShell độc hại. Vì vậy, nhóm tác giả sẽ trình bày chi tiết phân tích của nhóm tác giả về các tập lệnh PowerShell độc hại và lành tính. Trong phần trích xuất thuộc tính thủ công, framework sẽ trích xuất ***entropy, phân phối ký tự và sử dụng hàm, v.v***. Tập lệnh sẽ được phân tích cú pháp thành cây cú pháp trừu tượng, sau đó mô hình sẽ tính toán phân phối của 23 loại nút từ cây cú pháp trừu tượng (AST), độ sâu của AST, v.v.  **Kỹ thuật 2 (Trích xuất thuộc tính tự động):** Có **2** bước chính trong kỹ thuật này  **- Scripts preprocessing**: tất cả các comment sẽ bị xóa, bao gồm comment một dòng và comment nhiều dòng. Vì mã PowerShell không phân biệt chữ hoa chữ thường, tất cả các mã thông báo đã được chuẩn hóa thành chữ thường. Hơn nữa, để tránh sự mở rộng của vectơ từ gây ra bởi các số, tất cả các số đã được thay thế bằng '\*'. Sau đó, tất cả các ký tự không liên quan được thay thế bằng khoảng trắng (bao gồm ngắt dòng, tab, ký hiệu đặc biệt, v.v. Nhóm tác giả sử dụng biểu thức chính quy **1/2 ^ a zA Z $** để tìm các ký tự có liên quan này). Cuối cùng, tất cả các khoảng trắng liên tiếp được thay thế bằng một khoảng trắng để đảm bảo rằng các mã thông báo sẽ được phân đoạn chính xác.  **- Training mô hình FastText:**  Nhóm tác giả sử dụng những đoạn code sau khi xử lý về dạng text để đào tạo trước mô hình FastText và kích thước của vectơ được tạo là 300. Output của mô hình FastText này là nhãn dự đoán (***Predicted\_label***) và Độ tin cậy (***Confidence***)  **Kỹ thuật 3 (Lai thuộc tính):**  Kết hợp nhóm thuộc tính đã trích xuất từ kỹ thuật 1 và output của kỹ thuật 2 thành 1 nhóm thuộc tính lai (hybrid features), sau đó đưa vào mô hình FastText để phân loại. |

* 1. **Tóm tắt các kỹ thuật chính được mô tả sử dụng trong bài báo:**

|  |
| --- |
| VD: Bài báo sử dụng 3 kỹ thuật cho 3 giai đoạn chính trong phương pháp đề xuất, bao gồm:   * Kỹ thuật 01: Trích xuất thuộc tính tĩnh (**Manual features extraction**)   Trong kỹ thuật này, nhóm tác giả sẽ sử dụng các thư viện có sẵn của powershell để trích xuất các thuộc tính như **URL/IP, Entropy, NumString, MaxLength, AvgLength, TotalUsageTimes, Shellcode, SpecialVariables**, ...     * Kỹ thuật 02: Trích xuất thuộc tính động (**Automatic features extraction**)   Trong kỹ thuật này, nhóm tác giả xử lý code và biến đổi thành dạng text sau đó sử dụng mô hình fasttext phân loại benign hay malicious     * Kỹ thuật 03: Kết hợp lại kết quả của 2 kỹ thuật trên làm input cho mô hình phân loại Random Forest |

* 1. **Môi trường thực nghiệm của bài báo:**

|  |
| --- |
| * Cấu hình máy tính: **Intel i5-6700 K CPU and 24 GB of memory** * Các công cụ hỗ trợ sẵn có: **Windows Powershell, Scikit-learn and Keras** * Ngôn ngữ lập trình để hiện thực phương pháp: **Powershell, Python** * Đối tượng nghiên cứu (chương trình phần mềm dùng để kiểm tra tính khả thi của phương pháp/tập dữ liệu – nếu có): tác giả dùng tập dữ liệu do nhóm tác giả tự thu thập từ nhiều nguồn khác nhau (chủ yếu lấy trên github) bao gồm **4316** benign scripts, **4202** malicious scripts và **4202** mix\_malicious scripts (mix\_malicious scripts được tạo ra bằng cách chèn ngẫu nhiên đoạn code độc hại vào chương trình lành tính)   https://github.com/das-lab/mpsd   * Tiêu chí đánh giá tính hiệu quả của phương pháp: **Precision Recall F1-score Accuracy Loss, Confusion Matrix, ROC curve** trong phân loại giữa benign và mix\_malicious scripts |

* 1. **Kết quả thực nghiệm của bài báo:**

|  |
| --- |
| Kết quả thực nghiệm cho thấy hiệu quả của việc áp dụng hybrid features trong việc phân loại giữa benign và mix scripts: **Precision Recall F1-score Accuracy Loss** lần lượt là **0.9779 0.9767 0.9773 0.9776 0.0224**, **ROC\_AUC=0.9931**  Ưu điểm: Cho độ chính xác cũng như Precision Recall F1-score TPR cao trong khi Loss và FPR ở mức thấp  Nhược điểm: Quá trình trích xuất thuộc tính phải được triển khai trên Windows. Độ chính xác phát hiện của mô hình được đề xuất bị giới hạn bởi hiệu quả của các thuộc tính được trích xuất thủ công và kết quả phân loại dựa trên ngữ cảnh |

* 1. **Công việc/tính năng/kỹ thuật mà nhóm thực hiện lập trình và triển khai cho demo:**

|  |
| --- |
| - Dùng powershell trích xuất các thuộc tính tĩnh  - Tiền xử lý code trong các file dataset thành dạng text đưa vào mô hình phân loại FastText  - Kết hợp các kết quả từ 2 giai đoạn trên làm input cho mô hình Random Forest  <liệt kê các công việc đã thực hiện+ tóm tắt kết quả của công việc này trong việc thực hiện phương pháp/demo>  - Dùng powershell trích xuất các thuộc tính tĩnh: Trích xuất thành công một số thuộc tính tĩnh và lưu kết quả trích xuất vào 3 file **benign\_features.csv, malicious\_features.csv, mix\_malicious\_features.csv**  - Tiền xử lý code trong các file dataset thành dạng text đưa vào mô hình phân loại FastText: Tiền xử lý code và lưu kết quả vào 3 thư mục **preprocess.**.. Tạo các file text **Train\_Benign.txt, Train\_Malicious.txt, Train\_Mix\_Malicious.txt** phục vụ quá trình training FastText model. Lưu kết quả phân loại vào 2 file **semantic\_features\_1.csv, semantic\_features\_2.csv.**  - Kết hợp các kết quả từ 2 giai đoạn trên làm input cho mô hình Random Forest: Kết hợp thành công và đưa vào mô hình phân loại Random Forest |

* 1. **Các khó khăn, thách thức hiện tại khi thực hiện:**

|  |
| --- |
|  |

1. **TỰ ĐÁNH GIÁ MỨC ĐỘ HOÀN THÀNH SO VỚI KẾ HOẠCH THỰC HIỆN:**

|  |
| --- |
| 90% |

1. **NHẬT KÝ PHÂN CÔNG NHIỆM VỤ:**

|  |  |  |
| --- | --- | --- |
| **STT** | **Công việc** | **Phân công nhiệm vụ** |
| 1 | Dùng powershell trích xuất các thuộc tính tĩnh | Huy |
| 2 | Tiền xử lý code trong các file dataset thành dạng text đưa vào mô hình phân loại FastText | Đạt |
| 3 | Kết hợp các kết quả từ 2 giai đoạn trên làm input cho mô hình Random Forest | Đạt |

BÁO CÁO TỔNG KẾT CHI TIẾT

**Phần bên dưới của báo cáo này là tài liệu báo cáo tổng kết - chi tiết của nhóm thực hiện cho đề tài này.**

*Qui định: Mô tả các bước thực hiện/ Phương pháp thực hiện/Nội dung tìm hiểu (Ảnh chụp màn hình, số liệu thống kê trong bảng biểu, có giải thích)*

* 1. **Phương pháp thực hiện**

*<Trình bày kiến trúc, thành phần của hệ thống trong bài báo>*

*A diagram of a flowchart

Description automatically generated with low confidence*

Mô hình trong bài báo gồm 2 framework chính là **Manual feature extraction** (Trích xuất thuộc tính thủ công) và **Automatic feature extraction** (Trích xuất thuộc tính tự động)**.** Cụ thể từng phương pháp như sau:

**A.1. Manual feature extraction**

Sau khi tóm tắt các phương pháp phát hiện các tập lệnh PowerShell độc hại của các nhà nghiên cứu trong những năm gần đây, nhóm tác giảthấy rằng không có nghiên cứu nào về phân tích mức chức năng tập lệnh để phát hiện các tập lệnh PowerShell, mà là mức chức năng của PowerShell cần phải hữu ích để phát hiện các tập lệnh PowerShell độc hại. Vì vậy, trong chương này, nhóm tác giả sẽ trình bày chi tiết phân tích của nhóm tác giả về các tập lệnh PowerShell độc hại và lành tính. Phân tích của nhóm tác giả chủ yếu bao gồm ba cấp độ: **cấp độ văn bản, mức chức năng và cấp cây cú pháp trừu tượng.**

Trong phần trích xuất tính năng thủ công, framework sẽ trích xuất entropy thông tin, phân phối ký tự và sử dụng hàm, v.v. Tập lệnh sẽ được phân tích cú pháp thành cây cú pháp trừu tượng, sau đó mô hình sẽ tính toán phân phối của 23 loại nút từ cây cú pháp trừu tượng (AST), độ sâu của AST, v.v.

- **Shellcode** (các tập lệnh PowerShell độc hại thường được trộn lẫn với mã Shellcode)

- **Entropy** (phân tích sự phân bố của các ký tự khác nhau. Tin tặc thường làm xáo trộn các tập lệnh PowerShell độc hại để tránh sự phát hiện của phần mềm chống vi-rút. Nói cách khác, entropy của kịch bản lành tính cao hơn chữ viết bị xáo trộn)

- **Top 5 characters** (có một khoảng cách giữa sự phân bố ký tự trong kịch bản độc hại và các tập lệnh lành tính. Nhóm tác giảchuyển đổi năm ký tự thành ASCII, là năm ký tự nhiều nhất trong tập lệnh PowerShell, sau đó sắp xếp chúng theo số lượng của từng ký tự trong tập lệnh, nhóm tác giảsử dụng 5 ASCII hàng đầu được sắp xếp làm tính năng cuối cùng)

- **Độ dài ký tự** (số lượng chuỗi trong tập lệnh độc hại dài hơn các mẫu lành tính.

- **URL hoặc IP** (các tập lệnh PowerShell độc hại thường tải xuống phần mềm độc hại hoặc mã độc từ các trang web bên ngoài để tấn công thêm vào máy tính)

- **Special variable names** (các tập lệnh PowerShell độc hại luôn được đặt tên là "cmd", "Shell", "c", sử dụng tập lệnh PowerShell để gọi các dòng lệnh trong hack)

- **Malicious function using** (đếm thời gian sử dụng của các chức năng, thông số và mã thông báo này. Sau đó, nhóm tác giảtiếp tục theo dõi 200 chức năng hàng đầu để xác định chức năng nào là độc hại.)

**A.2. Semantics based features**

Nhóm tác giả đã phân tích các tính năng cấp ký tự khác nhau của mã PowerShell và trích xuất các tính năng từ phân phối nút và phân phối mã thông báo của cây cú pháp trừu tượng. Tiếp theo, chúng ta cần sử dụng FastText để thực hiện phân tích ngữ nghĩa văn bản trên các tập lệnh PowerShell, điều này sẽ giúp mô hình có được các tính năng lai.

***- Scripts preprocessing***: tất cả các comment đã bị xóa, bao gồm comment một dòng và comment nhiều dòng. Vì mã PowerShell không phân biệt chữ hoa chữ thường, tất cả các mã thông báo đã được chuẩn hóa thành chữ thường. Hơn nữa, để tránh sự mở rộng của vectơ từ gây ra bởi các số, tất cả các số đã được thay thế bằng '\*'. Sau đó, tất cả các ký tự không liên quan được thay thế bằng khoảng trắng (bao gồm ngắt dòng, tab, ký hiệu đặc biệt, v.v. Nhóm tác giả sử dụng biểu thức chính quy **1/2 ^ a zA Z $** để tìm các ký tự có liên quan này). Cuối cùng, tất cả các khoảng trắng liên tiếp được thay thế bằng một khoảng trắng để đảm bảo rằng các mã thông báo sẽ được phân đoạn chính xác.

***- Training mô hình FastText:***

Nhóm tác giả sử dụng bộ đào tạo để đào tạo trước mô hình FastText và kích thước của vectơ được tạo là 300

*<Trình bày kiến trúc, thành phần đã thực hiện (nội dung mà nhóm đã thực hiện)>*

**A.3. Manual feature extraction**

Dùng code powershell để parsed các file raw (.ps1) trong các thư mục (benign, malicious, mix-malicious) thành AST Nodes. Từ AST Nodes sẽ trích xuất ra các thuộc tính như URL/IP; Entropy; Số chuỗi, độ dài tối đa của chuỗi và độ dài trung bình của chuỗi; Tổng số lần sử dụng của hàm, biến, từ; Số lượng Shellcode; Số lượng các ký tự đặc biệt....

**A.4. Semantics based features**

Tiền xử lý dữ liệu trong các file raw (.ps1) thành dạng text theo phương pháp của tác giả. Sau đó đưa vào mô hình fasttext dự đoán kết quả benign hay malicious.

* 1. **Chi tiết cài đặt, hiện thực**

*<cách cài đặt, lập trình trên máy tính, cấu hình máy tính sử dụng, chuẩn bị dữ liệu, v.v>*

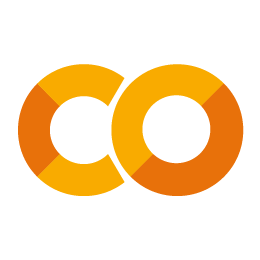
***- Môi trường thực hiện (Máy ảo Windows 10, Google Colab)***

+ Máy ảo Windows 10 (Cài đặt trên VMWare Pro 17, 2.4G RAM, 4 Processor, 80GB Disk)

A screenshot of a computer

Description automatically generated

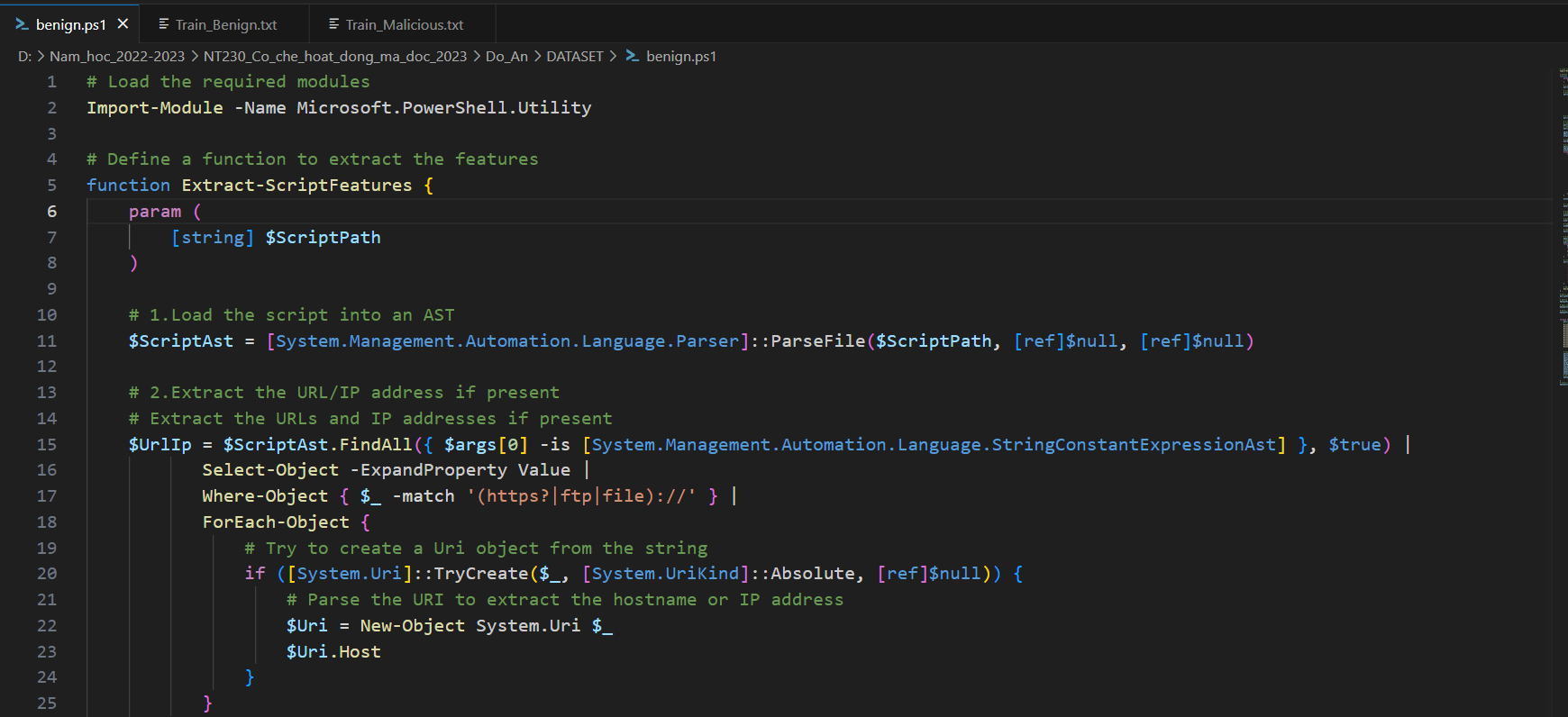
+ Google Colab (12GB RAM)



***- Chi tiết thực hiện:***

Trên máy ảo Windows 10. Chủ yếu để chạy **3** file powershell **benign.ps1, malicious.ps1, mix\_malicious.ps1** mà nhóm đã code. Các file này có nhiệm vụ trích xuất ra các thông tin **Manual features** như đã đề cập ở trên bằng cách đọc lần lượt từng file trong **3** thư mục dataset (benign, malicious, mix-malicious) ghi lần lượt các thông tin trích xuất vào 3 file csv **benign\_features, malicious\_features, mixmalicious\_features**.

* File **benign.ps1** đọc lần lượt từng file trong thư mục **benign dataset** và ghi các thông tin trích xuất vào file **benign\_features.csv**



A screen shot of a computer code

Description automatically generated with low confidence

A screen shot of a computer code

Description automatically generated with low confidence

A picture containing text, screenshot, software

Description automatically generated

A screen shot of a computer code

Description automatically generated with low confidence

A screen shot of a computer program

Description automatically generated with low confidence

File **benign\_features.csv** sau khi trích xuất có các thông tin như sau và được gắn thêm cột **Class = 0**

A screenshot of a computer

Description automatically generated with medium confidence

• Tương tự, file **malicious.ps1, mix\_malicious.ps1** đọc lần lượt từng file trong thư mục **malicious dataset, mixmalicious dataset** và ghi các thông tin trích xuất vào file **malicious\_features.csv, mix\_malicious\_features.csv**

File **malicious\_features.csv, mix\_malicious\_features.csv** sau khi trích xuất có các thông tin như sau và được gắn thêm cột **Class = 1**

A screenshot of a computer

Description automatically generated with low confidence

A picture containing text, number, screenshot, font

Description automatically generated

Tiền xử lý dữ liệu trước khi đưa vào model FastText: Ở bước này ta sẽ biến đổi code trong các file (.ps1) trong 3 thư mục dataset gốc thành dạng mong muốn và thêm các file (.ps1) sau khi biến đổi vào 3 thư mục dataset mới **preprocess\_powershell\_benign\_dataset, preprocess\_malicious\_pure, preprocess\_mixed\_malicious** (Tên các file trong các thư mục dataset mới vẫn giữ nguyên và chỉ khác nội dung code bên trong)

*A screenshot of a computer

Description automatically generated with medium confidence*

*A screenshot of a computer

Description automatically generated with medium confidence*

Code trước khi biến đổi:

A screen shot of a computer program

Description automatically generated with medium confidence

Code sau khi biến đổi chỉ nằm trên 1 dòng

A screen shot of a computer

Description automatically generated with low confidence

Tạo tệp dữ liệu cho FastText từ các tệp đã được tiền xử lý. Bước này sẽ tao ra **3** file text là **Train\_Benign.txt, Train\_Malicious.txt, Train\_Mixmalicious.txt**. Cụ thể ta sẽ gán nhãn **\_\_label\_\_benign, \_\_label\_\_malicious, \_\_label\_\_malicious** lần lượtcho từng file trong thư mục **preprocess\_powershell\_benign\_dataset, preprocess\_malicious\_pure, preprocess\_mixed\_malicious,** sau đó lần lượt ghi nội dung vào **3** file **Train\_Benign.txt, Train\_Malicious.txt, Train\_Mixmalicious.txt**

A screen shot of a computer program

Description automatically generated with low confidence

*A screenshot of a computer program

Description automatically generated with low confidence*

*File Train\_Benign.txt*

*A picture containing screenshot, text, font

Description automatically generated*

*File Train\_Malicious.txt*

*A picture containing text, screenshot, font, information

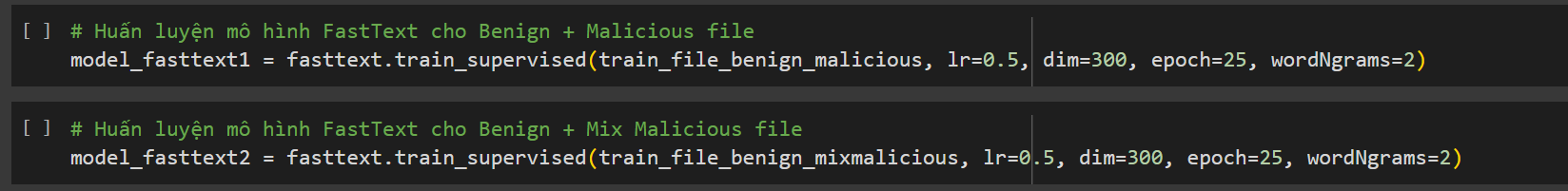
Description automatically generated*

Gộp các file chuẩn bị train cho 2 model

A screenshot of a computer program

Description automatically generated with medium confidence

Huấn luyện 2 mô hình fasttext (**model\_fasttext1, model\_fasttext2**) với tập dữ liệu tương ứng với các tham số **Learning\_rate=0.5, dim=300, epoch=25, ngrams=2**)

**

Dùng **model\_fasttext1** đưa ra dự đoán là **\_\_label\_\_benign** hay **\_\_label\_\_malicious** cho từng file trong 2 thư mục **preprocess\_powershell\_benign\_dataset, preprocess\_malicious\_pure.** Kết quả dự đoán lưu vào file **semantic\_features\_1.csv** A picture containing text, electronics, screenshot, software

Description automatically generated

*A picture containing text, screenshot, number, font

Description automatically generated*

Dùng **model\_fasttext2** đưa ra dự đoán là **\_\_label\_\_benign** hay **\_\_label\_\_malicious** cho từng file trong 2 thư mục **preprocess\_powershell\_benign\_dataset, preprocess\_mixed\_malicious.** Kết quả dự đoán lưu vào file **semantic\_features\_2.csv**

*A screen shot of a computer code

Description automatically generated with low confidence*

*A picture containing text, screenshot, font, number

Description automatically generated*

***Random Forest for Manual Extraction***

*Phân loại benign và malicious, tương tự cho phân loại benign và mix-malicious*

*A screenshot of a computer

Description automatically generated*

*A screenshot of a computer

Description automatically generated with medium confidence*

*A screenshot of a computer

Description automatically generated with medium confidence*

*A screen shot of a computer

Description automatically generated with medium confidence*

***Fast Text for Semantic Features Extraction***

*Phân loại benign và malicious, tương tự cho phân loại benign và mix-malicious*

*A screenshot of a computer program

Description automatically generated with low confidence*

*A screenshot of a computer program

Description automatically generated with low confidence*

***Random Forest for Hybrid Features (Manual+Semantic)***

*A screenshot of a computer

Description automatically generated with medium confidence*

*A screenshot of a computer

Description automatically generated with medium confidence*

*A screenshot of a computer

Description automatically generated with medium confidence*

*Phân loại benign và malicious, tương tự cho phân loại benign và mix-malicious*

*A screenshot of a computer program

Description automatically generated with medium confidence*

*A screenshot of a computer

Description automatically generated*

*A screenshot of a computer

Description automatically generated with medium confidence*

*A screenshot of a computer

Description automatically generated with medium confidence*

*A screenshot of a computer

Description automatically generated with medium confidence*

*A screenshot of a computer

Description automatically generated with medium confidence*

*A screenshot of a computer

Description automatically generated with medium confidence*

*A screenshot of a computer program

Description automatically generated with medium confidence*

*A screen shot of a computer program

Description automatically generated with low confidence*

* 1. **Kết quả thực nghiệm**

*<mô tả hình ảnh về thực nghiệm, bảng biểu số liệu thống kê từ thực nghiệm, nhận xét về kết quả thu được.>*

***\*Accuracy, Precision, Recall, F1-Score***

*Random forest for Manual Extraction (Ký hiệu* ***Manual****)*

*- Phân loại benign và malicious: Cho kết quả cao (>99%)*

*A screenshot of a computer screen

Description automatically generated with medium confidence*

*- Phân loại benign và mix-malicious: Cho kết quả tương đối cao (>93%)*

*A screenshot of a computer screen

Description automatically generated with low confidence*

*Fast Text for Semantic Features Extraction (Ký hiệu* ***FastText-2****)*

*- Phân loại benign và malicious: Cho kết quả cao (>99%)*

*A picture containing text, screenshot, font

Description automatically generated*

*- Phân loại benign và mix-malicious: Cho kết quả tương đối thấp (>85%)*

*A picture containing text, font, screenshot

Description automatically generated*

*Random Forest for Hybrid Features (Ký hiệu* ***M-FastText-2****)*

*- Phân loại benign và malicious: Cho kết quả cao (>99%)*

*A screenshot of a computer program

Description automatically generated with low confidence*

*- Phân loại benign và mix-malicious: Kết quả cao hơn một chút so với phương pháp* ***Manual (93%)*** *hoặc* ***FastText-2(85%)***

*A screenshot of a computer

Description automatically generated with medium confidence*

***Confusion Matrix***

*- Phân loại benign và malicious: Dự đoán sai đều ở mức thấp ở cả 3 model*

*A picture containing screenshot, rectangle, diagram, text

Description automatically generated*

*- Phân loại benign và mix-malicious: Số dự đoán sai thấp ở 2 model* ***Manual*** *và* ***M-FastText-2*** *và tương đối nhiều ở model* ***FastText-2***

*A picture containing screenshot, text, rectangle, diagram

Description automatically generated*

***ROC-Curve***

*- Phân loại benign và malicious: Tỷ lệ TPR cao và FPR thấp cho thấp cả 3 model đều có khả năng phân loại tốt*

*A picture containing text, line, screenshot, plot

Description automatically generated*

*- Phân loại benign và mix-malicious: model* ***Manual*** *và* ***M-FastText-2*** *cho kết quả khá cao (AUC=0.93, 0.95) và model FastText-2 tỷ lệ FPR còn tương đối cao (0.3)*

*A picture containing text, line, screenshot, plot

Description automatically generated*

* 1. **Hướng phát triển**

*<Nêu hướng phát triển tiềm năng của đề tài này trong tương lai. Nhận xét về tính ứng dụng của đề tài>.*

*Đề tài tận dụng kết quả từ nhiều phương pháp phân loại Fileless Malware khác nhau sau đó kết hợp lại cho kết quả phân loại cuối cùng. Đề tài có tính ứng dụng thực tế cao, tuy nhiên cần đòi hỏi các kiến thức của các chuyên gia trong lĩnh vực cũng như dữ liệu phải được xử lý qua nhiều giai đoạn (trích xuất manual features phải dùng thư viện powershell,....) gây đôi chút khó khăn.*

---

**HẾT**

1. Ghi nội dung tương ứng theo mô tả [↑](#footnote-ref-1)