CẢI TIẾN BỘ DỮ LIỆU PHÂN LOẠI MÃ ĐỘC TRONG TỆP THỰC THI WINDOWS BẰNG THUẬT TOÁN DI TRUYỀN

Phan Ngọc Vũ - 210201023

Tóm tắt

- Lớp: CS2205.APR 2024
- Link Github: https://github.com/PhanNgocVuUIT/CS2205.MAR2024.git
- Link YouTube video: https://youtu.be/I2Wf5yIsETU
- Họ tên: Phan Ngọc Vũ

Giới thiệu

- Làm thế nào để giảm số lượng đặc trưng, dung lượng ?
- Cải tiến: Áp dụng thuật toán di truyền để cải tiến quy trình trích xuất đặc trưng tập tin thực thi (PE).

Bảng thống kê thông tin về EMBER, SOREL-20M và BODMAS

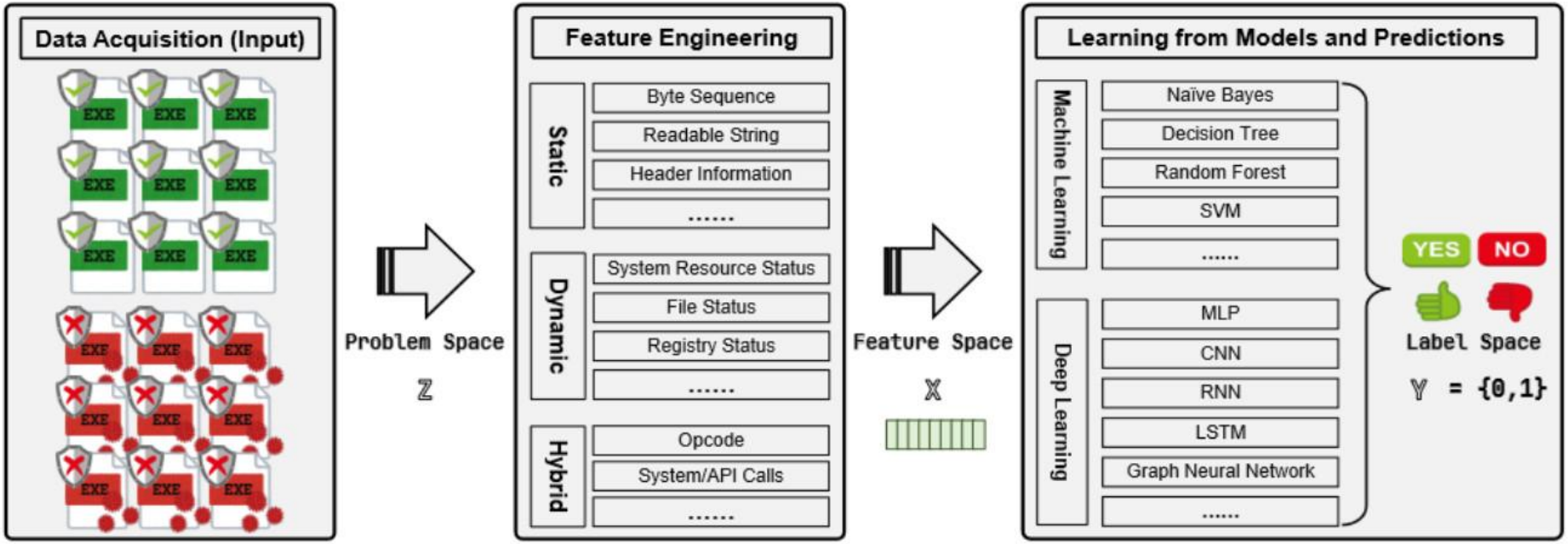
[1,2,3] là 3 bộ dữ liệu mở phổ biến nhất dùng trong PLMĐ PE

Bộ dữ liệu	Số đặc trưng	Dung lượng
EMBER	25.000	1.2 GB
SOREL-20M	20.000	2.3 GB
BODMAS	30.000	4.1 GB

Mục tiêu

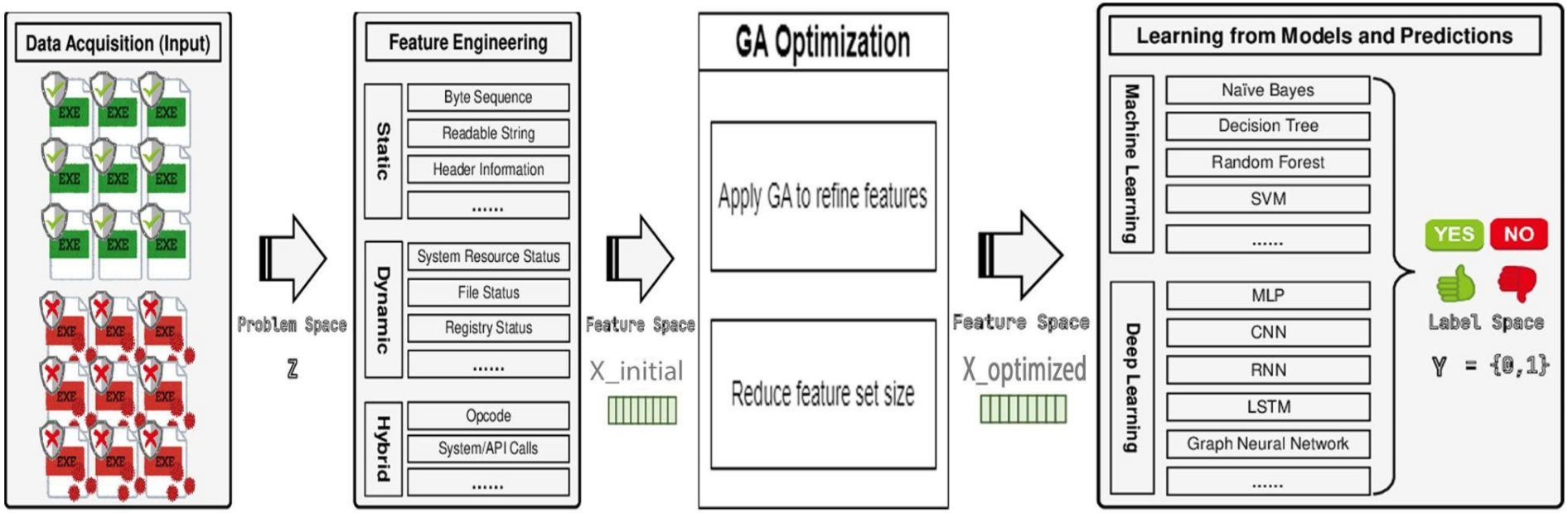
Mục tiêu	Chi tiết
Tạo bộ dữ liệu UIT_PE gọn nhẹ	- Áp dụng thuật toán GA để giảm 50% đặc trưng được trích xuất từ file PE
Huấn luyện và đánh giá UIT_PE trên RFC, GBC, LR	- Cải thiện độ chính xác ít nhất 5% so với bộ dữ liệu gốc. - Giảm thời gian huấn luyện xuống 20%.

Nội dung và Phương pháp



Quy trình hiện tại dùng để trích xuất đặc trưng và dùng thuật toán học máy để phân loại mã độc [4]

Nội dung và Phương pháp (tt)



Mô hình bổ sung thuật toán GA để cải tiến quy trình

Nội dung và Phương pháp (tt)

Nội dung thực hiện:

1. Thu thập và chuẩn bị dữ liệu:
- Thu thập malware và benign từ Bodmas và Windows 11.
2. Trích xuất đặc trưng từ file PE:
- Phân tích cấu trúc của file PE để hiểu headers, sections, ...
- Trích xuất các đặc trưng tĩnh như header, sections, imports từ các tệp PE.
3. Lựa chọn đặc trưng bằng thuật toán di truyền (GA):
- Áp dụng GA để lựa chọn đặc trưng thu được bộ dữ liệu UIT_PE
4. Huấn luyện và đánh giá bộ dữ liệu UIT_PE trên các mô hình học máy:
- Sử dụng RandomForest, LogisticRegression, và GradientBoostingClassifier để huấn luyện bộ dữ liệu.
- Đánh giá hiệu suất dựa trên độ chính xác, điểm F1, và ROC AUC

Nội dung và Phương pháp (tt)

Phương pháp thực hiện:

- Phân tích và tiền xử lý dữ liệu: Sử dụng thư viện trong Python: pefile và LIEF để phân tích cấu trúc và trích xuất dữ liệu từ file PE.
- Lựa chọn đặc trưng bằng thuật toán di truyền (GA): Để tìm ra tập hợp các đặc trưng tối ưu ảnh hưởng tới việc 1 tập tin là mã độc hay không.
- Huấn luyện và đánh giá mô hình: Sử dụng Jupyter Notebook trên môi trường Anaconda để thực hiện quá trình huấn luyện và đánh giá mô hình trên dữ liệu đã được tối ưu hóa.

Kết quả dự kiến

- Giảm kích thước bộ dữ liệu: Thông qua áp dụng thuật toán GA, dự kiến sẽ giảm được 50% số lượng đặc trưng, từ đó giảm 30% dung lượng lưu trữ và rút ngắn 20% thời gian xử lý cho việc phân loại.
- Hiệu suất: Các mô hình học máy, sau khi được huấn luyện trên bộ dữ liệu UIT_PE, dự kiến sẽ cho độ chính xác cao hơn 5% so với trước.

Tài liệu tham khảo

- [1] Hyrum S. Anderson, Phil Roth: EMBER: An Open Dataset for Training Static PE Malware Machine Learning Models. ArXiv 2018: arxiv.org/abs/1804.04637.
- [2] Richard Harang, Ethan M. Rudd: SOREL-20M: A Large Scale Benchmark Dataset for Malicious PE Detection. Conference on Applied Machine Learning for Information Security 2021.
- [3] Limin Yang, Arridhana Ciptadi, Ihar Laziuk, Ali Ahmadzadeh, Gang Wang: BODMAS: An Open Dataset for Learning based Temporal Analysis of PE Malware. University of Illinois at Urbana-Champaign and Blue Hexagon.
- [4] Xiang Ling et al.:Adversarial Attacks against Windows PE Malware Detection: A Survey of the State-of-the-Art. Computers & Security, December 23, 2021. DOI:10.1016/j.cose.2023.103134