Tên đề tài: “Nghiên cứu đề xuất phương pháp phát hiện lỗ hổng phần mềm sử dụng kỹ thuật SMOTE theo tiếp cận học sâu”

Giảng viên hướng dẫn:
Bùi Văn Công

Thành Viên Nhóm

Họ và tên	Mã sinh viên	Lớp	Chức danh thực hiện đề tài
Nguyễn Xuân Phượng	21103100946	DHTI15A3HN	Chủ nhiệm đề tài
Vũ Nguyễn Dạ Uyên	21103100291	DHTI15A4HN	Thành viên
Nguyễn Trung Hiếu	21103100097	DHTI15A4HN	Thành viên
Phạm Văn Dũng	21103100057	DHTI15A3HN	Thành viên

Quay lại Trang Chương trình

DANH MỤC

PHẦN MỞ ĐẦU

PHẦN

01. Giới thiệu ý tưởng nghiên cứu

PHẦN

02. Mục tiêu, đối tượng, phạm vi của đề tài

PHẦN

03.Cơ sở lý thuyết và lịch sử nghiên cứu

PHẦN

04.Phương pháp nghiên cứu

PHẦN MỞ ĐẦU

Giới thiệu ý tưởng nghiên cứu

Giới thiệu ý tưởng nghiên cứu

1.1. Bối cảnh và động cơ nghiên cứu

Trong bối cảnh an ninh mạng ngày càng trở nên quan trọng, việc phát hiện lỗ hổng phần mềm là một yếu tố quan trọng để bảo vệ hệ thống. Các phương pháp truyền thống thường gặp khó khăn trong việc phát hiện các lỗ hổng vì dữ liệu lỗ hổng thường không cân bằng. Đề tài này tập trung vào việc cải thiện khả năng phát hiện lỗ hổng bằng cách kết hợp kỹ thuật SMOTE và học sâu.

1.2. Mục tiêu nghiên cứu

- Đề xuất phương pháp phát hiện lỗ hổng phần mềm hiệu quả bằng cách sử dụng kỹ thuật SMOTE.
- Áp dụng học sâu để nâng cao hiệu suất phát hiện lỗ hổng.

PHẦN 02

Mục tiêu, đối tượng, phạm vi của đề tài

Mục tiêu, đối tượng, phạm vi của đề tài

2.1 Mục tiêu nghiên cứu:

Việc sử dụng kết hợp các phương pháp trong phân tích và phân loại lỗ hổng mã nguồn luôn là vấn đề cần thiết bởi việc kết hợp đó sẽ giúp việc phân loại mã nguồn được chính xác và hiệu quả hơn. Đồng thời sẽ giúp tránh được các vấn đề như phát hiện nhầm, tình trạng bị overfitting. Đồng thời việc sử dụng các phương pháp nhằm chuẩn hoá đồ thị theo cạnh và đỉnh cũng là các cách kết hợp nhằm giúp cho việc phân tích các đặc trưng được tốt để quá trình phân loại đạt được độ chính xác cao.

2.2 Đối tượng, phạm vi của đề tài:

Trong phạm vi của đề tài, nội dung nghiên cứu được thực hiện gồm:

- Tổng quan về học sâu
- Nghiên cứu phương pháp phân tích mã nguồn.
- Tìm hiểu CNN, GCN, SMOTE
- Cài đặt và huấn luyện mô hình phát hiện lỗ hổng phần mềm

PHẦN 03

Cơ sở lý thuyết và lịch sử nghiên cứu

Cơ sở lý thuyết và lịch sử nghiên cứu

- Cơ sở lý thuyết: lập trình Python, các giải thuật CNN, RNN,SMOTE...
- Lịch sử nghiên cứu:
  - 1. SMOTE và Học Sâu:
  - SMOTE (2002): Được giới thiệu bởi Chawla et al., SMOTE là kỹ thuật tạo mẫu giả cho lớp thiểu số nhằm giải quyết vấn đề không cân bằng lớp trong học máy.
  - Học Sâu: Phát triển từ những năm 2000, học sâu với các mô hình như mạng nơ-ron tích chập (CNN) và mạng nơ-ron hồi tiếp (RNN) đã chứng minh hiệu quả trong nhiều ứng dụng học máy.
  - 2. Phát Hiện Lỗ Hổng Phần Mềm:
  - Cây Quyết Định và Random Forests: Cây quyết định được phát triển từ những năm 1980, với Random forests (2001) của Breiman sử dụng nhiều cây để cải thiện độ chính xác.
  - Ứng Dụng Trong Phát Hiện Lỗ Hổng: Kỹ thuật này đã được áp dụng để phân loại mã nguồn và phát hiện lỗ hổng bảo mật.
  - 3. Kết Hợp SMOTE và Học Sâu:
  - Tích Hợp SMOTE với Học Sâu: Nghiên cứu gần đây tích hợp SMOTE với các mô hình học sâu nhằm cải thiện phát hiện lỗ hổng phần mềm bằng cách giải quyết vấn đề không cân bằng lớp trong dữ liệu.
  - 4. Định Hướng Nghiên Cứu Hiện Tại:
  - Nghiên Cứu Gần Đây: Tích hợp SMOTE và học sâu đang được nghiên cứu để nâng cao hiệu quả phát hiện lỗ hổng phần mềm.
  - Hướng Phát Triển: Tương lai có thể tập trung vào cải tiến các kỹ thuật SMOTE và mô hình học sâu mới để cải thiện độ chính xác và hiệu suất.

PHẦN 04

Phương pháp nghiên cứu

Phương pháp nghiên cứu

Thu thập và chuẩn bị dữ liệu

- Dữ liệu lỗ hổng phần mềm: Thu thập dữ liệu từ các nguồn như cơ sở dữ liệu lỗ hổng công khai.
- Tiền xử lý dữ liệu: Làm sạch và chuẩn bị dữ liệu, bao gồm việc phân loại dữ liệu và xử lý dữ liệu không cân bằng bằng SMOTE.

Áp dụng kỹ thuật SMOTE

- Tạo mẫu giả: Sử dụng SMOTE để tăng cường dữ liệu lớp thiểu số nhằm cải thiện cân bằng lớp.

Xây dựng và huấn luyện mô hình học sâu

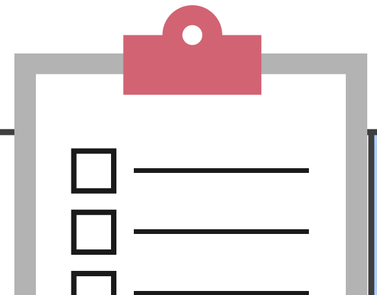
- Chọn mô hình: Lựa chọn các mô hình học sâu như CNN hoặc RNN.
- Huấn luyện mô hình: Sử dụng dữ liệu đã được tăng cường để huấn luyện mô hình và điều chỉnh các tham số.

Đánh giá mô hình

- Chỉ số đánh giá: Sử dụng các chỉ số như độ chính xác, độ nhạy, và độ đặc hiệu để đánh giá hiệu suất của mô hình.

Tài liệu t ham khảo

[1] “CVE,” 2021, http://cve.mitre.org.
[2] CWE TOP25, https://cwe.mitre.org/top25/archive/2021/2021_cwe_top25.html.
[3] G. Lin, S. Wen, Q. -L. Han, J. Zhang and Y. Xiang, "Software Vulnerability Detection Using Deep Neural Networks: A Survey," in Proceedings of the IEEE, vol. 108, no. 10, pp. 1825-1848, Oct. 2020, doi: 10.1109/JPROC.2020.2993293.
[4] Zeng, G. Lin, L. Pan, Y. Tai and J. Zhang, "Software Vulnerability Analysis and Discovery Using Deep Learning Techniques: A Survey," in IEEE Access, vol. 8, pp. 197158-197172, 2020, doi: 10.1109/ACCESS.2020.3034766.
[5] H. Wang et al., "Combining Graph-Based Learning With Automated Data Collection for Code Vulnerability Detection," in IEEE Transactions on Information Forensics and Security, vol. 16, pp. 1943-1958, 2021, doi: 10.1109/TIFS.2020.3044773.
[6] Li X, Wang L, Xin Y, Yang Y, Tang Q, Chen Y. Automated Software Vulnerability Detection Based on Hybrid Neural Network. Applied Sciences. 2021; 11(7):3201. https://doi.org/10.3390/app11073201.
[7] H. Wei, M. Li, “Supervised deep features for software functional clone detection by exploiting lexical and syntactical information in source code,” in Proceedings of the TwentySixth International Joint Conference on Artificial Intelligence, pp. 3034–3040, Melbourne, Australia, August 2017.
[8] G. Siewruk and W. Mazurczyk, "Context-Aware Software Vulnerability Classification Using Machine Learning," in IEEE Access, vol. 9, pp. 88852-88867, 2021, doi: 10.1109/ACCESS.2021.3075385.
[9] Jinchang Hu; Jinfu Chen; Lin Zhang; Yisong Liu; Qihao Bao; Hilary AckahArthur, "A memory-related vulnerability detection approach based on vulnerability features," Tsinghua Science and Technology , vol. 25, no. 5, pp. 604 - 613, 2020.



LOGO CỦA BẠN

Thank you