PHIẾU ĐĂNG KÝ ĐỀ TÀI *(Dành cho sinh viên)*

1. Đơn vị chủ trì: **Khoa Công nghệ Thông tin**	**2. Cơ quan chủ quản:** **Trường Đại học Kinh tế - Kỹ thuật Công nghiệp**
3. 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"	
4. Mã số thuế:	
5. Chủ nhiệm đề tài: Nguyễn Xuân Phượng	
Họ và tên: Nguyễn Xuân Phượng Khoa: Công nghệ thông tin Điện thoại: 0365921103	Mã số sinh viên: 21103100946 Lớp: DHTI15A3HN Email: phuong0961070156@gmail.com
6. Giảng viên hướng dẫn: Bùi Văn Công	
Học vị: ThS. Khoa học máy tính Điện thoại: 0983978015	Chức vụ, đơn vị: Giảng viên, Khoa CNTT Email: bvcong@uneti.edu.vn
7. Tóm tắt nội dung đề tài: - Nghiên cứu phương pháp phân tích mã nguồn để tìm kiếm lỗ hổng bảo mật theo tiếp cận học sâu; Nghiên cứu lập trình Python; Nghiên cứu một số mô hình đa kết hợp sử dụng học sâu ứng dụng cho bài toán phát hiện lỗ hổng phần mềm . - Xây dựng mô hình huấn luyện, thực nghiệm và đánh giá hiệu quả của mô hình. - Báo cáo tổng kết đề tài	
8. Thời gian thực hiện đề tài: 10 tháng (Từ tháng 7 năm 2024 đến tháng 4 năm 2025)	
9. Khối lượng và kinh phí: - Khối lượng nghiên cứu khoa học: 65 giờ chuẩn - Chi phí phục vụ đề tài: Theo quy chế chi tiêu nội bộ	
10. Loại hình nghiên cứu: N/C cơ bản ☐ N/C ứng dụng ☒ Triển khai thực nghiệm ☒	**11. Lĩnh vực khoa học:** Tự nhiên ☐ Kỹ thuật công nghệ ☒ Nông nghiệp ☐ Y học ☐ Xã hội nhân văn ☐

Ngày tháng năm 2024

Giảng viên hướng dẫn

Bùi Văn Công
Khoa CNTT

Chủ nhiệm đề tài

Nguyễn Xuân Phượng
Phòng KHCN

Nguyễn Hoàng Chiến

Ngày ….tháng ….năm….
HIỆU TRƯỞNG
Trường Đại học Kinh tế - Kỹ thuật Công nghiệp

THUYẾT MINH ĐỀ TÀI NGHIÊN CỨU KHOA HỌC

(Dành cho sinh viên)

1. 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" | **2. Mã số:**

3. Thời gian thực hiện: 10 tháng (Từ tháng 7 năm 2024 đến tháng 4 năm 2025)

4. Khối lượng và kinh phí:

- Khối lượng nghiên cứu khoa học: 65 giờ chuẩn
- Chi phí phục vụ đề tài: Theo quy chế chi tiêu nội bộ

5. Họ tên Chủ nhiệm đề tài (trưởng nhóm sinh viên): **Nguyễn Xuân Phượng**

Khoa: Công nghệ thông tin Lớp: DHTI15A3HN

Điện thoại: 0365921103 Email: phuong0961070156@gmail.com

6. Giảng viên hướng dẫn: Bùi Văn Công

Học hàm, học vị, chuyên môn: Thạc sĩ Khoa học máy tính

Chức vụ: Giảng viên Đơn vị công tác: Khoa Công nghệ thông tin

Điện thoại: 0983978015 Email: bvcong@uneti.edu.vn

7. Cơ quan chủ quản: Trường Đại học Kinh tế Kỹ thuật Công nghiệp

Địa chỉ: 456 Minh Khai - Hai Bà Trưng - Hà Nội

Điện thoại: 024.38621504; Fax: 024.38623938;

8. Đơn vị phối hợp chính: Khoa Công nghệ thông tin

9. Danh sách những sinh viên thực hiện chính:

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

10. Giới thiệu ý tưởng nghiên cứu:

Với sự phát triển mạnh mẽ của thời đại công nghệ số hiện nay, việc ứng dụng công nghệ thông tin ngày càng mạnh mẽ vào mọi lĩnh vực của đời sống xã hội. Điều này dẫn đến vấn đề đảm bảo an toàn thông tin, kiểm soát thông tin đang là một thách thức lớn. Theo thống kê tại [1] có thể thấy được sự gia tăng đột biến về số lượng lỗ hổng và mức độ thiệt hại của chúng. Báo cáo tại [2] cho thấy ngày càng có nhiều lỗ hổng được phát hiện và công bố. Đặc biệt mức độ nghiêm trọng của các lỗ hổng là rất lớn. Chính vì vậy việc phân tích mã nguồn để tìm kiếm lỗ hổng bảo mật đang là thách thức lớn của các tổ chức phát triển phần mềm hiện nay. Chính vì vậy nhóm tác giả đã nghiên cứu học sâu sử dụng các thuật toán tái cân bằng dữ liệu cũng như mạng biến đổi đồ thị áp dụng trong bài toán phân loại lỗ hổng phần mềm đã được nhóm tác giả đề xuất nhằm giải quyết các vấn đề trên.

11. Mục tiêu, đối tượng, phạm vi của đề tài:

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.

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
- Cài đặt và huấn luyện mô hình phát hiện lỗ hổng phần mềm

12. Cơ sở lý thuyết và lịch sử nghiên cứu:

- Cơ sở lý thuyết: lập trình Python, các giải thuật CNN, GCN, …
- Lịch sử nghiên cứu:

Python là một ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng, do Guido van Rossum tạo ra và lần đầu ra mắt vào năm 1991. Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình và là ngôn ngữ lập trình dễ học, được dùng rộng rãi trong phát triển trí tuệ nhân tạo [10].

Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu. Python hoàn toàn tạo kiểu động và dùng cơ chế cấp phát bộ nhớ tự động; do vậy nó tương tự như Perl, Ruby, Scheme, Smalltalk, và Tcl. Python được phát triển trong một dự án mã mở, do tổ chức phi lợi nhuận Python Software Foundation quản lý.

Trong lĩnh vực học máy, cây quyết định là một kiểu mô hình dự báo (predictive model), nghĩa là một ánh xạ từ các quan sát về một sự vật/hiện tượng tới các kết luận về giá trị mục tiêu của sự vật/hiện tượng. Mỗi một nút trong (internal node) tương ứng với một biến; đường nối giữa nó với nút con của nó thể hiện một giá trị cụ thể cho biến đó. Mỗi nút lá đại diện cho giá trị dự đoán của biến mục tiêu, cho trước các giá trị của các biến được biểu diễn bởi đường đi từ nút gốc tới nút lá đó. Kỹ thuật học máy dùng trong cây quyết định được gọi là học bằng cây quyết định, hay chỉ gọi với cái tên ngắn gọn là cây quyết định. Học bằng cây quyết định cũng là một phương pháp thông dụng trong trí tuệ nhân tạo. Khi đó, cây quyết định mô tả một cấu trúc cây, trong đó, các lá đại diện cho các phân loại còn cành đại diện cho các kết hợp của các thuộc tính dẫn tới phân loại đó. Một cây quyết định có thể được học bằng cách chia tập hợp nguồn thành các tập con dựa theo một kiểm tra giá trị thuộc tính. Quá trình này được lặp lại một cách đệ quy cho mỗi tập con dẫn xuất. Quá trình đệ quy hoàn thành khi không thể tiếp tục thực hiện việc chia tách được nữa, hay khi một phân loại đơn có thể áp dụng cho từng phần tử của tập con dẫn xuất. Một bộ phân loại rừng ngẫu nhiên (random forest) sử dụng một số cây quyết định để có thể cải thiện tỉ lệ phân loại.

Random forests tạo ra cây quyết định trên các mẫu dữ liệu được chọn ngẫu nhiên, được dự đoán từ mỗi cây và chọn giải pháp tốt nhất bằng cách bỏ phiếu. Nó cũng cung cấp một chỉ báo khá tốt về tầm quan trọng của tính năng. Random forests có nhiều ứng dụng, chẳng hạn như công cụ đề xuất, phân loại lỗ hổng phần mềm dựa trên đặc trưng của mã nguồn và đưa ra có lỗ hổng hay không [11, 12].

Bài toán phân loại lỗ hổng phần mềm đang là một trong các bài toán được nhiều tổ chức, doanh nghiệp quan tâm khi ngày càng nhiều ứng dụng công nghệ thông tin được ứng dụng rộng rãi sâu rộng thì yếu tố bảo mật càng được coi trọng và quan tâm.

13. Phương pháp nghiên cứu:

- Phương pháp nghiên cứu lý thuyết
- Đề xuất giải pháp
- Demo ứng dụng phân loại lỗ hổng phần mềm
- Đánh giá, phân tích kết quả nghiên cứu

14. Nội dung nghiên cứu:

- Nghiên cứu phương pháp phân tích mã nguồn để tìm kiếm lỗ hổng phần mềm.
- Nghiên cứu lập trình Python
- Nghiên cứu một số mô hình học sâu cho bài toán phát hiện lỗ hổng phần mềm.
- Xây dựng ứng dụng phát hiện lỗ hổng phần mềm, thực nghiệm và đánh giá hiệu năng của mô hình.
- Báo cáo tổng kết đề tài

15. Dạng sản phẩm, kết quả tạo ra:

- Báo cáo khoa học
- Bài viết đăng kỷ yếu hội nghị sinh viên NCKH
- Xây dựng ứng dụng phát hiện lỗ hổng phần mềm, thực nghiệm và kết quả

16. Yêu cầu khoa học đối với sản phẩm kết quả tạo ra

STT	Tên sản phẩm	Số lượng	Mô tả đặc điểm	Yêu cầu khoa học	Địa chỉ áp dụng
1	Báo cáo khoa học	01	Báo cáo	Đúng quy cách	Sử dụng cho sinh viên chuyên ngành CNTT của Nhà trường trong đồ án môn học, học phần trí tuệ nhân tạo, học phần toán rời rạc, mạng máy tính, an toàn thông tin và an ninh mạng.
2	Tóm tắt thông tin đề tài theo mẫu NCKH-04 để đăng tạp chí hoặc kỷ yếu	01	Báo cáo	Số hiệu chính xác, đảm bảo tính mới, không sao chép	Khoa CNTT
3	Ứng dụng phân loại lỗ hổng phần mềm	01	Báo cáo	Đúng quy định	Khoa CNTT

17. Tiến độ thực hiện:

Stt	Nội dung công việc	Kết quả đạt được	Thời gian bắt đầu, kết thúc	Người, đơn vị thực hiện
1	- Nghiên cứu tài liệu - Xây dựng đề cương sơ bộ	- Đề cương sơ bộ	07/2024 - 08/2024	Nguyễn Xuân Phượng
2	- Xây dựng đề cương chi tiết - Nghiên cứu một số phương pháp phân tích mã nguồn	-Đề cương chi tiết -Những phương pháp hiện nay ưu và nhược điểm	09/2024 - 10/2024	Nguyễn Vũ Dạ Uyên Nguyễn Trung Hiếu

3	-Nghiên cứu về đặc điểm, quy trình phân tích mã nguồn -Nghiên cứu thuật toán Random Forest và Decision Tree - Xây dựng mô hình	- Bài viết	11/2024 - 12/2024	Nguyễn Xuân Phượng Nguyễn Vũ Dạ Uyên Nguyễn Trung Hiếu Phạm Văn Dũng
4	Cài đặt chương trình	- Ứng dụng, kết quả thực nghiệm	01/2024 - 02/2024	Nguyễn Xuân Phượng
6	- Viết báo cáo, bài viết đăng kỷ yếu.	- Viết báo cáo, bài viết đăng kỷ yếu.	03/2024 - 04/2024	Nguyễn Vũ Dạ Uyên

18. Dự toán kinh phí thực hiện đề tài

Mục	Nội dung	Số tiền/ giờ chuẩn	Giải trình chi tiết	Ghi chú
1. Công nghiên cứu				
	Chuyên đề 1: - Tiếp cận phương pháp **Chuyên đề 2:** - Xây dựng ứng dụng, thực nghiệm và kết quả	30 giờ chuẩn 35 giờ chuẩn	Thu thập thông tin, nghiên cứu lý thuyết. Nghiên cứu lý thuyết, xây dựng và cài đặt Demo, viết báo cáo - tài liệu, in ấn.	
2. Chi khác				
	Chi phí phục vụ đề tài		Theo qui chế chi tiêu nội bộ	

Ngày ….tháng ….năm 2024

Giảng viên hướng dẫn

Chủ nhiệm đề tài

Bùi Văn Công

Nguyễn Xuân Phượng

Khoa CNTT

Phòng KHCN

Nguyễn Hoàng Chiến

Ngày ….tháng ….năm….

HIỆU TRƯỞNG

Trường Đại học Kinh tế - Kỹ thuật Công nghiệp

TÀI LIỆU THAM 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.
[10] Nguyễn Tất Bảo Thiện, Nguyễn Quốc Huy, *Trí Tuệ Nhân Tạo Học Máy Và Ứng Dụng,* NXB Thanh niên, 2022
[11] Trần Hoàng Bình, (2015). Ứng dụng Random Forest để tư vấn chọn lộ trình học trong học chế tín chỉ. Luận văn Thạc sĩ chuyên ngành Khoa học máy tính. Mã số: 60.48.01.01, Đại học Đà Nẵng.
[12]. Breiman L. and Cutler A., (2007). Random Forests. [Online]. Available: https://www.stat.berkeley.edu/~breiman/RandomForests/ [Accessed: 08-Aug-2017].
[13] Breiman L., (2001). Random Forests. Machine Learning Journal Paper, vol. 45, (no.1), p. 5-32. Oct. 2001.