KHOA KỸ THUẬT VÀ CÔNG NGHỆ **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN** **Độc lập – Tự do – Hạnh phúc**

**ĐỀ CƯƠNG CHI TIẾT**

**KHÓA LUẬN TỐT NGHIỆP NGÀNH CÔNG NGHỆ THÔNG TIN**

Họ tên sinh viên: **Phạm Hữu Lộc** MSSV: 110121055

Lớp: DA21TTB Khóa: 2021

Tên đề tài: XÂY DỰNG CƠ CHẾ PHÁT HIỆN AI-GENERATED CODE TRONG BÀI TẬP LẬP TRÌNH CỦA SINH VIÊN

**Mục tiêu của đề tài**

Đề tài "Xây dựng cơ chế phát hiện AI-generated code trong bài tập lập trình của sinh viên" hướng đến việc nghiên cứu, xây dựng và đánh giá một hệ thống có khả năng phát hiện mã nguồn được tạo ra bởi các công cụ AI trong bài tập lập trình của sinh viên. Với sự phát triển mạnh mẽ của các mô hình ngôn ngữ lớn và công cụ hỗ trợ lập trình như ChatGPT, Github Copilot, Deepseek hay Remini,… Việc sinh viên sử dụng các công cụ này trong quá trình học tập đang trở nên phổ biến. Điều này tạo ra thách thức trong việc đánh giá năng lực thực sự của sinh viên, đặc biệt là trong các môn học cơ bản như Kỹ thuật lập trình hay Cấu trúc dữ liệu. Đề tài này phát triển một công cụ với quy trình cụ thể nhằm hỗ trợ việc phát hiện mã nguồn được tạo bởi AI thông qua việc phân tích các đặc trưng của mã nguồn mà không cần sử dụng các mô hình học sâu phức tạp.

Mục tiêu nghiên cứu của đề tài tập trung vào việc tìm hiểu sâu về các đặc trưng của mã nguồn do AI tạo ra, từ đó phát triển các phương pháp phân tích và phát hiện hiệu quả. Quá trình thực hiện đề tài giúp củng cố kiến thức về phân tích mã nguồn, hiểu rõ về cấu trúc và đặc điểm của mã do con người và AI tạo ra. Việc thu thập và xây dựng tập dữ liệu đối chiếu giúp phát triển kỹ năng xử lý dữ liệu thực tế, trong khi việc trích xuất các đặc trưng từ mã nguồn như AST (Abstract Syntax Tree), độ phức tạp, độ lặp lại và các đặc điểm về cách đặt tên biến/hàm giúp nâng cao hiểu biết về kỹ thuật phân tích mã nguồn.

Việc xây dựng mô hình phát hiện bằng phương pháp rule-based và học máy nhẹ cũng là cơ hội để áp dụng kiến thức về thuật toán vào giải quyết vấn đề thực tế, cũng như rèn luyện kỹ năng đánh giá hiệu quả của các phương pháp khác nhau. Quá trình này không chỉ giúp hoàn thiện kỹ năng lập trình và xử lý dữ liệu mà còn phát triển tư duy phân tích, tổng hợp và đánh giá trong nghiên cứu khoa học.

Về mặt ứng dụng, đề tài hướng đến việc xây dựng một công cụ thực tế có thể hỗ trợ giảng viên trong việc đánh giá bài tập của sinh viên, đảm bảo tính công bằng trong quá trình học tập. Công cụ này có thể được tích hợp vào hệ thống quản lý học tập, hỗ trợ việc phát hiện tự động mã nguồn nghi ngờ, từ đó giúp giảng viên tập trung vào việc đánh giá chính xác hơn các trường hợp đặc biệt.

**Nội dung thực hiện:**

Đề tài sẽ bắt đầu với việc nghiên cứu các khái niệm và công nghệ liên quan đến các công cụ AI hỗ trợ lập trình. Việc tìm hiểu cách thức hoạt động và đặc điểm mã nguồn từ các công cụ như ChatGPT, Github Copilot, Deepseek và Remini cung cấp nền tảng lý thuyết vững chắc để xác định các đặc trưng tiềm năng giúp phân biệt mã nguồn do AI tạo ra và mã nguồn do sinh viên viết.

Sau đó, đề tài sẽ tiến hành thu thập và xây dựng tập dữ liệu mẫu bao gồm hai nhóm: mã nguồn thực tế từ bài tập của sinh viên và mã nguồn được tạo bởi các công cụ AI với cùng yêu cầu bài tập. Quá trình này đòi hỏi sự phối hợp với giảng viên để có được bài tập và mã nguồn của sinh viên, đồng thời cần thực hiện việc sử dụng các công cụ AI để tạo ra mã nguồn tương ứng. Dữ liệu thu thập sẽ được phân loại và gán nhãn thủ công, tạo thành tập dữ liệu chuẩn để phục vụ cho quá trình phân tích và đánh giá sau này.

Tiếp theo, đề tài sẽ tập trung vào việc phân tích đặc trưng mã nguồn thông qua việc xây dựng các công cụ phân tích tự động. Các phương pháp phân tích và trích xuất đặc trưng sẽ được nghiên cứu và triển khai, bao gồm việc sử dụng AST để phân tích cấu trúc mã nguồn, tính toán đo lường độ phức tạp (cyclomatic complexity), phân tích độ lặp lại của mã (code redundancy) và cách đặt tên biến/hàm. Ngoài ra, đề tài còn xem xét các đặc trưng khác như tỷ lệ comment, biến đặt tên chung chung và các mẫu lập trình đặc trưng.

Dựa trên kết quả phân tích đặc trưng, đề tài sẽ tiến hành xây dựng các phương pháp phát hiện mã nguồn do AI tạo ra. Hai hướng tiếp cận chính sẽ được áp dụng: phương pháp dựa trên luật (rule-based) với việc xác định các ngưỡng thủ công cho các đặc trưng đã trích xuất, và phương pháp học máy nhẹ nếu có đủ dữ liệu. Việc kết hợp hai phương pháp này giúp tận dụng ưu điểm của mỗi phương pháp, đồng thời cho phép đánh giá hiệu quả của từng cách tiếp cận.

Cuối cùng, đề tài sẽ phát triển một ứng dụng thử nghiệm với giao diện người dùng thân thiện, cho phép người dùng nhập mã nguồn và nhận kết quả phân tích, cùng với các gợi ý và cảnh báo khi phát hiện mã nguồn có khả năng được tạo bởi AI. Ứng dụng này có thể được phát triển dưới dạng công cụ CLI hoặc web app đơn giản sử dụng các công nghệ như FastAPI và Next.js.

**Phương pháp thực hiện:**

Để đạt được mục tiêu của đề tài, quá trình thực hiện sẽ được chia thành các bước cụ thể, tuần tự theo quy trình phát triển phần mềm và phân tích dữ liệu. Cụ thể, các phương pháp được áp dụng bao gồm:

**Nghiên cứu và tổng quan lý thuyết:** Tìm hiểu về các công cụ AI hỗ trợ lập trình phổ biến, cách thức hoạt động và đặc điểm mã nguồn chúng tạo ra. Nghiên cứu các phương pháp phân tích mã nguồn như AST, cyclomatic complexity, code similarity. Tìm hiểu các công trình nghiên cứu trước đó liên quan đến phát hiện mã nguồn do AI tạo ra.

**Thu thập và xử lý dữ liệu:** Thu thập bài tập và mã nguồn của sinh viên. Sử dụng các công cụ AI để tạo mã nguồn với cùng yêu cầu bài tập. Phân loại và gán nhãn dữ liệu để tạo tập dữ liệu chuẩn. Tiền xử lý dữ liệu bao gồm chuẩn hóa định dạng, loại bỏ các yếu tố không liên quan và chuẩn bị dữ liệu cho quá trình phân tích.

**Phân tích và trích xuất đặc trưng:** Xây dựng công cụ phân tích AST cho các ngôn ngữ lập trình cần xét (C, Java, Python, JavaScript,...). Tính toán các chỉ số về độ phức tạp của mã nguồn. Phát triển phương pháp đo lường độ lặp lại của mã ở mức token và cấu trúc hay phân tích cách đặt tên biến, hàm và tỉ lệ comment trong mã nguồn.

**Xây dựng mô hình phát hiện:** Thiết kế các luật và ngưỡng dựa trên phân tích đặc trưng đã trích xuất cho phương pháp rule-based. Huấn luyện các mô hình học máy đơn giản như nếu có đủ dữ liệu. Kết hợp hai phương pháp để tạo mô hình tối ưu.

**Đánh giá và tinh chỉnh:** Chia dữ liệu thành các tập huấn luyện và kiểm thử theo tỉ lệ phù hợp. Tinh chỉnh các tham số của mô hình để cải thiện kết quả phát hiện, đặc biệt là giảm tỷ lệ false positive.

**Phát triển ứng dụng thử nghiệm:** Thiết kế và xây dựng giao diện người dùng đơn giản cho phép nhập mã nguồn và hiển thị kết quả phân tích.

**Đánh giá tổng thể và đề xuất cải tiến:** Thử nghiệm ứng dụng với các trường hợp thực tế để đánh giá hiệu quả tổng thể. Phân tích các hạn chế và đề xuất hướng cải tiến trong tương lai. Tổng hợp kết quả và rút ra bài học từ quá trình nghiên cứu.

**Bố cục đề tài:**

**Chương 1. Giới thiệu đề tài:** Phần này trình bày tổng quan về đề tài, bao gồm lý do chọn đề tài, tầm quan trọng của vấn đề phát hiện mã nguồn do AI tạo ra trong môi trường học tập. Chương này cũng xác định rõ mục tiêu, phạm vi và giới hạn của đề tài, cũng như phương pháp nghiên cứu tổng quát và cấu trúc của toàn bộ khóa luận.

**Chương 2. Cơ sở lý thuyết:** Chương này cung cấp nền tảng lý thuyết cần thiết cho đề tài, bao gồm tổng quan về các công cụ AI hỗ trợ lập trình như ChatGPT, Github Copilot, Deepseek và Remini. Phần này cũng giới thiệu về các khái niệm và phương pháp phân tích mã nguồn như AST, cyclomatic complexity, code clone detection, cùng với tổng quan về các kỹ thuật phát hiện mã nguồn do AI tạo ra đã được nghiên cứu trước đây.

**Chương 3. Phân tích và thiết kế hệ thống:** Chương này tập trung vào việc phân tích bài toán phát hiện mã nguồn do AI tạo ra, xác định các yêu cầu chức năng và phi chức năng của hệ thống. Phần này cũng trình bày quy trình thu thập và xử lý dữ liệu, thiết kế kiến trúc tổng thể của hệ thống phát hiện, bao gồm các thành phần chính và mối quan hệ giữa chúng, cũng như mô hình luồng dữ liệu và các công nghệ được lựa chọn để triển khai.

**Chương 4. Thu thập và phân tích dữ liệu:** Chương này mô tả chi tiết quá trình thu thập dữ liệu mã nguồn từ sinh viên và tạo mã nguồn bằng các công cụ AI. Phần này trình bày phương pháp phân loại và gán nhãn dữ liệu, cùng với các kỹ thuật tiền xử lý được áp dụng. Ngoài ra, chương này cũng trình bày quá trình phân tích và trích xuất đặc trưng từ mã nguồn, bao gồm việc sử dụng AST, tính toán độ phức tạp và phân tích các đặc trưng khác. Cuối cùng, phần này cung cấp các phân tích thống kê về đặc trưng của hai nhóm mã nguồn, từ đó rút ra những đặc điểm giúp phân biệt mã nguồn do AI tạo ra.

**Chương 5. Xây dựng và đánh giá mô hình phát hiện và ứng dụng:** Chương này trình bày chi tiết về việc xây dựng các mô hình phát hiện mã nguồn do AI tạo ra, bao gồm phương pháp rule-based và phương pháp học máy. Phần này mô tả quá trình thiết kế luật và xác định ngưỡng, cũng như việc huấn luyện và tinh chỉnh các mô hình học máy. Chương này cũng trình bày kết quả đánh giá hiệu quả của các mô hình. Ngoài ra, phần này trình bày các tính năng chính của ứng dụng, cách sử dụng và các ví dụ minh họa.

**Chương 6. Kết luận và hướng phát triển:** Chương cuối cùng tổng kết những kết quả đạt được của đề tài, đánh giá những đóng góp chính và hạn chế còn tồn tại. Phần này cũng đề xuất các hướng phát triển tiếp theo, bao gồm việc mở rộng phạm vi ngôn ngữ lập trình, cải thiện độ chính xác của mô hình phát hiện, và tích hợp vào các hệ thống quản lý học tập.

**Tài liệu tham khảo, Phụ lục.**

1. **Tài liệu tham khảo:**

Các tài liệu tham khảo sử dụng trong đề tài được lựa chọn từ nhiều nguồn uy tín, bao gồm bài báo khoa học quốc tế, tài liệu nghiên cứu trong nước, sách chuyên ngành và các nguồn dữ liệu, công cụ thực tế. Những tài liệu này đóng vai trò quan trọng trong việc xây dựng cơ sở lý thuyết, lựa chọn phương pháp phân tích và định hướng triển khai hệ thống. Một số tài liệu tham khảo hiện tại:

|  |  |
| --- | --- |
| [1] | K. Niyazmetova, "Zenodo," 2025. [Online]. Available: https://zenodo.org/records/14826525. |

**2. Kế hoạch thực hiện đề tài**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tuần** | **Từ ngày - đến ngày** | **Công việc thực hiện** | **Ghi chú** |
| **Tuần 1** | 8/04 – 13/04 | **Phân tích đề tài** - Lựa chọn công nghệ - Lập kế hoạch - Viết đề cương chi tiết khóa luận |  |
| **Tuần 2** | 14/04 – 20/04 | **Nghiên cứu cơ bản** - Khảo sát các công cụ AI phổ biến (ChatGPT, Copilot, Deepseek, Remini, …) - Thu thập mẫu mã nguồn sinh từ AI - Tìm hiểu các phương pháp phân tích đặc trưng code (code fingerprint, syntax tree, cyclomatic complexity, clone detection) - Trao đổi với GVHD |  |
| **Tuần 3** | 21/04 – 27/04 | **Xây dựng tập dữ liệu mẫu** - Thu thập bài tập từ sinh viên - Sinh mẫu code từ AI với cùng đề bài - Phân loại thủ công mã AI vs mã con người - Báo cáo tiến độ tuần 2 với GVHD |  |
| **Tuần 4** | 28/04 – 04/05 | **Phân tích đặc trưng mã nguồn** - Viết tool Python để đo: + Độ phức tạp (radon) + Độ lặp lại (token-level) + Tỉ lệ comment, biến đặt tên chung chung - Định nghĩa bộ đặc trưng đầu tiên - Báo cáo tiến độ tuần 3 với GVHD |  |
| **Tuần 5** | 05/05 – 11/05 | **Phân tích và thống kê dữ liệu** - Chạy phân tích trên tập dữ liệu - Thống kê đặc trưng nào để phân biệt tốt giữa AI và người - Lập bảng, biểu đồ so sánh, nhận xét - Báo cáo tiến độ tuần 4 với GVHD |  |
| **Tuần 6** | 12/05 – 18/05 | **Xây dựng prototype cơ bản** - Viết tool CLI/Web nhỏ nhận input mã nguồn - Tính toán đặc trưng và xuất kết quả - Gợi ý cảnh báo nếu nghi ngờ là mã AI-generated - Báo cáo tiến độ tuần 5 với GVHD |  |
| **Tuần 7** | 19/05 – 25/05 | **Kiểm thử – tinh chỉnh thuật toán** - Thử nghiệm với nhiều mẫu code - Tối ưu ngưỡng cảnh báo - Loại bỏ false positive nhiều nhất có thể - Báo cáo tiến độ tuần 6 với GVHD |  |
| **Tuần 8** | 26/05 – 01/06 | **Phát triển giao diện người dùng**- Web app nhỏ để nhập mã và hiển thị kết quả - Giao diện đơn giản (Next.js/FastAPI) - Hoàn thiện chức năng toàn hệ thống - Báo cáo tiến độ tuần 7 GVHD |  |
| **Tuần 9** | 02/06 – 08/06 | **Kiểm thử toàn hệ thống + viết khóa luận** - Test thử toàn diện - Viết các phần chính của khóa luận: Cơ sở lý thuyết, Phương pháp tiếp cận, Phân tích dữ liệu - Báo cáo tiến độ tuần 8 với GVHD |  |
| **Tuần 10** | 09/06 – 11/06 | **Chỉnh sửa – Viết báo cáo hoàn chỉnh** - Viết phần đánh giá, kết luận, hướng phát triển - Gửi GVHD xem bản nháp |  |
| **Tuần 11** | 12/06 – 14/06 | **Sửa báo cáo theo phản hồi GVHD** - Hoàn thiện quyển khóa luận, chuẩn bị slide báo cáo - Gửi khóa luận hoàn chỉnh |  |
| **Tuần 12** | 15/06 – 16/06 | **Bảo vệ, nộp đồ khóa luận** - Chuẩn bị trình bày - Nộp source code, tài liệu, slide |  |

*Trà Vinh, ngày tháng 4 năm 2025*

**GIẢNG VIÊN HƯỚNG DẪN SINH VIÊN THỰC HIỆN  
  
  
  
  
 Nguyễn Bảo Ân Phạm Hữu Lộc**