PHÁT TRIỂN HỆ THỐNG THÔNG MINH HỖ TRỢ THỰC HÀNH LẬP TRÌNH DI ĐỘNG ỨNG DỤNG THUẬT TOÁN HỌC CÓ GIÁM SÁT

Võ Thị Kim Hoa
Khoa Công nghệ Thông tin, Trường Đại học Công nghệ Kỹ thuật Vinh
Email: vokimhoaskv@gmail.com

Tóm tắt: *Học phần Phát triển ứng dụng di động yêu cầu sinh viên đồng thời nắm vững Java để xử lý logic và XML để xây dựng giao diện. Trong thực hành, sinh viên thường gặp các lỗi cú pháp, logic nhưng mất nhiều thời gian gỡ lỗi do thiếu hỗ trợ tức thời. Bài báo đề xuất xây dựng hệ thống hỗ trợ thực hành tích hợp Học có giám sát (Supervised Learning) để tự động phân tích mã nguồn, nhận diện, phân loại lỗi và đưa ra gợi ý sửa lỗi. Hệ thống sử dụng tập dữ liệu lỗi thực tế, thực hiện tiền xử lý, trích xuất đặc trưng và huấn luyện các thuật toán phân loại. Kết quả thử nghiệm bước đầu cho thấy hệ thống có khả năng hỗ trợ gỡ lỗi, góp phần giảm tải cho giảng viên và nâng cao tính chủ động của người học*

Từ khóa: *Lập trình Android, Học có giám sát, Nhận diện lỗi tự động, Trợ giảng ảo, Phân loại văn bản.*

DEVELOPMENT OF AN INTELLIGENT SYSTEM FOR SUPPORTING MOBILE PROGRAMMING PRACTICE USING SUPERVISED LEARNING ALGORITHMS

Abstract: *The Mobile Application Development course requires students to master both Java for logical processing and XML for user interface design. During laboratory sessions, students frequently encounter syntax and logical errors but spend considerable time debugging due to the lack of immediate assistance. This paper proposes an intelligent practice-support system using Supervised Learning to automatically analyze source code, identify and classify programming errors, and provide debugging suggestions. The system collects real programming-error data, performs preprocessing and feature extraction, and trains classification algorithms. Preliminary experiments indicate that the system can effectively support debugging, reduce instructors' workload, and enhance students' self-directed learning.*

Keywords: *Android programming, Supervised learning, Automated error detection, Virtual teaching assistant, Text classification.*

Nhận bài: 18/09/2026 Phản biện: 20/10/2026 Duyệt đăng: 25/10/2026

I. MỞ ĐẦU

Trong bối cảnh chuyển đổi số, trí tuệ nhân tạo (AI), đặc biệt là Học máy (Machine Learning), đang mở ra nhiều hướng ứng dụng trong hỗ trợ và đánh giá người học (Phạm Quốc Cường, 2021; Nguyễn Văn Thành và Lê Thị Mai, 2022). Đối với ngành Công nghệ Thông tin, học phần Phát triển ứng dụng di động trên Android yêu cầu sinh viên kết hợp tư duy thuật toán với kỹ năng lập trình và thiết kế giao diện.

Tuy nhiên, người học phải đồng thời làm quen với Java, XML cùng các khái niệm như Activity, Intent và vòng đời ứng dụng. Các lỗi cú pháp, logic hoặc cấu trúc giao diện thường làm gián đoạn quá trình thực hành, trong khi giảng viên khó hỗ trợ tức thời cho từng sinh viên trong lớp đông (Sultan và Al-Thani, 2023).

Từ thực tế đó, nghiên cứu đề xuất xây dựng hệ thống hỗ trợ thực hành tích hợp AI như một "trợ giảng ảo", sử dụng Học có giám sát để phân tích mã nguồn Java và XML, nhận diện, phân loại lỗi và đưa ra gợi ý khắc phục (Trần Đình Bách và Phạm Hoàng Long, 2023; Kumar và Gupta, 2024). Hệ thống hướng tới giúp sinh viên chủ động phát hiện, hiểu và sửa lỗi, đồng thời giảm tải cho giảng viên trong giờ thực hành.

II. NỘI DUNG

2.1. Cơ sở lý thuyết

Học có giám sát là phương pháp học máy trong đó mô hình được huấn luyện từ dữ liệu đã gán nhãn để dự đoán nhãn cho dữ liệu mới (Nguyễn Văn Thành và Lê Thị Mai, 2022). Trong nghiên cứu, đầu vào là các đặc trưng của mã nguồn Java hoặc XML, đầu ra là nhãn tương ứng với loại lỗi. Đây là bài toán phân loại đa lớp: $D = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$, trong đó x là mẫu mã nguồn và y là nhãn lỗi. Nghiên cứu tập trung vào hai thuật toán phân loại:

(1) K-Nearest Neighbors (K-NN): phân loại mẫu mới dựa trên k mẫu gần nhất trong không gian đặc trưng. Với mã nguồn được biểu diễn bằng vector, khoảng cách Cosine được sử dụng để đo mức độ tương đồng giữa các đoạn mã.

(2) Decision Tree: phân loại dữ liệu thông qua các luật "Nếu - Thì". Các đặc trưng như từ khóa Java, thuộc tính XML hoặc cấu trúc mã nguồn được sử dụng để tạo các nút phân nhánh. Ưu điểm của phương pháp là khả năng giải thích kết quả tương đối rõ ràng.

Để đưa mã nguồn về dạng dữ liệu có thể xử lý, hệ thống thực hiện tiền xử lý và trích xuất đặc trưng. Mã nguồn được làm sạch, tách thành các token và biểu diễn dưới dạng vector bằng TF-IDF. Phương pháp này đánh giá mức độ quan trọng của từ khóa trong từng đoạn mã, qua đó giúp phân biệt các nhóm lỗi.

Hiệu năng mô hình được đánh giá bằng các chỉ số Accuracy, Precision, Recall và F1-Score dựa trên ma trận nhầm lẫn. Trong đó, Precision phản ánh mức độ chính xác của các dự đoán lỗi, Recall phản ánh khả năng phát hiện lỗi thực tế và F1-Score cân bằng hai chỉ số này, đặc biệt hữu ích khi dữ liệu giữa các nhóm lỗi không đồng đều.

2.2. Thiết kế hệ thống

2.2.1. Thu thập và xử lý dữ liệu

Tập dữ liệu được xây dựng từ các đoạn mã Java và tệp XML trong bài tập thực hành của sinh viên, gồm cả mã đúng và mã chứa các lỗi thường gặp (Trần Đình Bách và Phạm Hoàng Long, 2023). Giảng viên thực hiện gán nhãn để đảm bảo độ chính xác, chẳng hạn ERR_XML_ID_MISSING (thiếu định danh View), ERR_JAVA_LIFECYCLE (sai xử lý vòng đời Activity) và ERR_UI_THREAD (cập nhật giao diện trên luồng không phù hợp).

Dữ liệu sau khi gán nhãn được chia thành tập huấn luyện và kiểm thử. Quy trình tiền xử lý gồm làm sạch dữ liệu, chuẩn hóa, tách token và trích xuất đặc trưng bằng TF-IDF. Đối với XML, các thẻ và thuộc tính được giữ lại để bảo toàn thông tin cấu trúc giao diện.

2.2.2. Kiến trúc hệ thống

Hệ thống được thiết kế theo mô hình Client-Server, tách giao diện Android khỏi phần xử lý học máy nhằm giảm tải tài nguyên cho thiết bị di động.

Android Client: Sinh viên nhập hoặc tải mã nguồn Java/XML. Ứng dụng đóng gói dữ liệu dưới dạng JSON và gửi yêu cầu qua HTTP.

API Server: Tiếp nhận yêu cầu, bóc tách và tiền xử lý dữ liệu, sau đó chuyển mã nguồn thành vector đặc trưng.

AI Engine: Đưa vector vào mô hình K-NN hoặc Decision Tree đã huấn luyện để xác định loại lỗi.

Response: Kết quả nhận diện và gợi ý sửa lỗi được đóng gói thành JSON, trả về Client và hiển thị cho sinh viên.

Việc tách riêng luồng xử lý dữ liệu và giao diện giúp hệ thống dễ bảo trì, mở rộng các nhóm lỗi mới mà không phải thay đổi sâu kiến trúc ứng dụng.

2.3. Cài đặt và đánh giá

Hệ thống được cài đặt và thử nghiệm thực tế với kiến trúc Client - Server nhằm đánh giá cả về độ chính xác của mô hình AI lẫn hiệu năng vận hành trên thiết bị di động.

Phân hệ AI Server: Được triển khai bằng Python 3.12, sử dụng scikit-learn để xây dựng và thực hiện suy luận các mô hình Học máy, FastAPI để cung cấp RESTful API tiếp nhận và xử lý yêu cầu từ ứng dụng Android, và PyMySQL để kết nối, lưu trữ và truy xuất dữ liệu từ cơ sở dữ liệu. Phân hệ này đảm nhiệm toàn bộ quá trình tiếp nhận mã nguồn, tiền xử lý, trích xuất đặc trưng, phân loại lỗi và trả kết quả, gợi ý sửa lỗi về cho ứng dụng.

Phân hệ Android Client: Được phát triển trên Android Studio bằng ngôn ngữ Java, sử dụng XML để thiết kế giao diện và tương thích với Android 10.0 (API Level 29) trở lên. Phân hệ cho phép sinh viên nhập hoặc tải mã nguồn Java/XML, gửi dữ liệu đến AI Server để phân tích và nhận kết quả nhận diện lỗi cùng gợi ý sửa lỗi, sau đó hiển thị trực tiếp trên thiết bị.

Tập dữ liệu thử nghiệm: Gồm 2.500 mẫu mã nguồn, trong đó có 1.500 mẫu Java và 1.000 mẫu XML, được thu thập từ dữ liệu thực hành có sẵn. Dữ liệu được chia theo tỷ lệ 80% (2.000 mẫu) cho huấn luyện và 20% (500 mẫu) cho kiểm thử. Các mẫu được gán nhãn theo từng nhóm lỗi để phục vụ quá trình huấn luyện và đánh giá mô hình.

Thử nghiệm được tiến hành tại lớp Phát triển ứng dụng di động 1 trong hai tuần thực hành. Kết quả cho thấy hệ thống có khả năng hỗ trợ sinh viên phát hiện và xử lý các lỗi cú pháp, logic cơ bản mà không phải chờ giảng viên hỗ trợ trực tiếp. Việc cung cấp phản hồi và gợi ý sửa lỗi ngay trong quá trình thực hành giúp người học chủ động hơn trong quá trình gỡ lỗi và hiểu rõ nguyên nhân của lỗi. Kết quả đánh giá mô hình cho thấy độ chính xác cao nhất đạt 97,5%, trong khi thời gian phản hồi của hệ thống dao động khoảng 1–2 giây, đáp ứng yêu cầu hỗ trợ gần thời gian thực.

III. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Nghiên cứu đã xây dựng hệ thống hỗ trợ thực hành lập trình Android dựa trên kiến trúc Client-Server và Học có giám sát. Hệ thống thực hiện được các chức năng thu thập, tiền xử lý và trích xuất đặc trưng mã nguồn, phân loại lỗi bằng các mô hình học máy và trả về gợi ý sửa lỗi trên ứng dụng Android.

Kết quả thử nghiệm cho thấy mô hình có độ chính xác cao, đạt mức tối ưu 97,5%, trong khi thời gian phản hồi khoảng 1–2 giây. Thử nghiệm thực tế cũng cho thấy hệ thống hỗ trợ sinh viên chủ động xử lý các lỗi cơ bản, góp phần giảm áp lực hỗ trợ trực tiếp cho giảng viên và nâng cao kỹ năng gỡ lỗi của người học. Việc triển khai Backend trên VPS/Cloud trong tương lai có thể nâng cao khả năng đáp ứng và tính sẵn sàng của hệ thống.

Trong giai đoạn tiếp theo, nghiên cứu tập trung vào: (1) mở rộng dữ liệu với các lỗi liên quan đến Room Database, SQLite, xử lý bất đồng bộ và kiến trúc MVVM; (2) tích hợp môi trường biên dịch và thực thi an toàn để phát hiện lỗi runtime; (3) nghiên cứu các mô hình chuyên biệt cho mã nguồn như CodeBERT hoặc LLM kích thước nhỏ nhằm nâng cao khả năng hiểu ngữ nghĩa và (4) xây dựng cơ chế phân tích lịch sử lỗi để cá nhân hóa bài tập và lộ trình học tập.

TÀI LIỆU THAM KHẢO

Kumar, A., & Gupta, P. (2024). Real-time debugging feedback system for mobile application development using K-nearest neighbors. Computers & Education: Artificial Intelligence, 6, Article 100185.

Nguyễn Văn Thành & Lê Thị Mai. (2022). Nghiên cứu ứng dụng học máy trong tự động đánh giá và phân loại bài tập lập trình Java. Tạp chí Khoa học và Công nghệ Các trường Đại học Kỹ thuật, 158, 45–52.

Phạm Quốc Cường. (2021). Giải pháp nâng cao năng lực tự học lập trình di động cho sinh viên CNTT dựa trên mô hình hỗ trợ thông minh. Tạp chí Giáo dục, 498, 88–94.

Sultan, R., & Al-Thani, M. (2023). Automated error detection and classification in programming education using supervised learning models. Journal of Computer Assisted Learning, 39(4), 1120–1135.

Trần Đình Bách & Phạm Hoàng Long. (2023). Xây dựng hệ thống trợ giảng ảo hỗ trợ thực hành lập trình ứng dụng di động Android. Tạp chí Công nghệ Thông tin và Truyền thông, 11(2), 12–19.