Republic of the Philippines
CAVITE STATE UNIVERSITY
Don Severino de las Alas Campus
Indang, Cavite

AUTOMATED BUG DETECTION AND FIXING IN SOFTWARE DEVELOPMENT

Rationale/Introduction

Software reliability and security are crucial concerns in modern software development, as undetected bugs can lead to severe vulnerabilities, performance issues, and financial losses. Traditional debugging methods rely heavily on manual code reviews and extensive testing, which can be time-consuming, error-prone, and inefficient in large-scale software projects. With the increasing complexity of software systems, there is a pressing need for automated solutions that can identify and rectify bugs efficiently to enhance software quality and security.

This research explores the implementation of an automated bug detection and fixing system using advanced machine learning techniques and static code analysis. By leveraging deep learning models, natural language processing (NLP), and program analysis, the study aims to develop a system capable of identifying common software defects, providing recommendations for fixes, and even implementing corrective measures automatically. The integration of such an intelligent system into the software development lifecycle can significantly reduce debugging time, improve code quality, and enhance the overall efficiency of software engineering processes.

Significance of the Study

The significance of this research lies in its potential to revolutionize the software debugging process. Automated bug detection and fixing can drastically reduce development costs, minimize security vulnerabilities, and streamline software maintenance. By incorporating artificial intelligence and machine learning into the debugging workflow, software engineers can focus on innovation and feature development rather than spending extensive time on troubleshooting and debugging.

Furthermore, this research benefits various stakeholders, including software developers, quality assurance teams, and software companies, by reducing human effort in identifying and fixing errors. The implementation of an automated debugging system can lead to improved software reliability, reduced system downtime, and enhanced user satisfaction. Additionally, the study contributes to the broader field of artificial intelligence by showcasing practical applications of machine learning in software development and maintenance.

Scope and Limitations of the Study

This study focuses on developing an automated system for detecting and fixing software bugs using machine learning techniques. It will explore various approaches, including static and dynamic code analysis, NLP-based defect prediction, and reinforcement learning for automated bug fixing. The research will assess the effectiveness of different models in identifying common software issues such as syntax errors, security vulnerabilities, and logical inconsistencies.

However, the study has certain limitations. It will primarily target high-level programming languages such as Python, Java, and C++ and may not cover all programming paradigms or languages comprehensively. Additionally, while the system will provide automated recommendations and potential fixes, it may not be able to handle complex, context-dependent bugs that require deep domain knowledge. Another limitation is that the research will focus on controlled experiments and simulated environments rather than real-world deployment scenarios in large-scale enterprise software.

Objectives of the Study

The primary objective of this study is to design and develop an automated bug detection and fixing system that enhances software quality and reliability.

- To analyze common software bugs and defects across different programming languages and categorize their impact on software quality.
- To develop a machine learning-based model that can efficiently detect bugs in source code using static and dynamic analysis techniques.
- To implement a recommendation system that suggests potential fixes for identified bugs based on past debugging patterns and learned models.
- To evaluate the effectiveness and accuracy of the automated bug detection and fixing system in comparison to traditional debugging methods.
- To provide insights into the integration of automated bug detection tools into existing software development workflows.

Expected Outputs

This research is expected to produce a functional prototype of an automated bug detection and fixing system that utilizes machine learning techniques for enhanced software quality assurance. The system will demonstrate the ability to detect various types of software bugs, categorize them, and suggest appropriate solutions, potentially automating corrective actions where feasible.

Additionally, the study will generate a comprehensive analysis of different machine learning models and their effectiveness in software defect detection. The findings will include performance evaluations, accuracy assessments, and comparisons between traditional debugging approaches and the proposed automated system. The research will also provide guidelines for integrating such systems into real-world software development environments, offering insights for software engineers, researchers, and industry practitioners interested in leveraging artificial intelligence for software maintenance and debugging.

REFERENCES

Bird, C., Zimmermann, T., Nachiappan, N., & Murphy, B. (2015). The practice of bug fixing. *IEEE Software, 32*(2), 52-59.

Gupta, A., Pal, S., & Agrawal, A. (2021). Automated bug detection using deep learning: A survey. *ACM Computing Surveys, 54*(3), 1-29.

Li, X., Xia, X., Lo, D., & Jin, Z. (2019). Deep learning-based automated program repair: Are we there yet? *Proceedings of the 41st International Conference on Software Engineering*, 1-12.

Zhang, F., Li, L., Ma, X., & Hu, S. (2020). Machine learning for software bug detection: A systematic review. *Journal of Software: Evolution and Process, 32*(6), e2265.