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

AI-BASED BUG DETECTION AND FIXING IN SOFTWARE DEVELOPMENT

Rationale/Introduction

Software bugs remain a persistent challenge in software development, leading to security vulnerabilities, performance issues, and increased development costs. Traditional debugging processes often rely on manual code inspection and extensive testing, which are time-consuming and prone to human error. The integration of Artificial Intelligence (AI) into bug detection and fixing offers a promising solution by automating the identification and correction of code anomalies. Machine learning models, deep learning techniques, and Natural Language Processing (NLP) are being increasingly leveraged to analyze source code patterns, predict potential errors, and even generate fixes (Gupta et al., 2021).

AI-powered bug detection and fixing systems aim to enhance software reliability by identifying defects early in the development lifecycle, reducing debugging time, and improving code quality. Despite these advantages, challenges such as false positives, contextual understanding of code logic, and adaptability across different programming languages still exist. This study seeks to explore the effectiveness, limitations, and future prospects of AI-driven bug detection and fixing methodologies in modern software development.

Significance of the Study

This study is significant as it contributes to the advancement of automated debugging technologies, reducing human effort and improving software quality. By investigating AI-based bug detection and fixing, the research provides valuable insights into how software development teams can streamline debugging processes and reduce production failures. Furthermore, the study will benefit organizations, developers, and software engineers seeking to integrate AI-driven solutions into their development pipelines to enhance efficiency and security.

Scope and Limitations of the Study

This research focuses on AI-driven techniques for bug detection and automatic code correction. It will analyze various machine learning and deep learning models used in identifying software vulnerabilities and suggest ways AI can enhance debugging accuracy. However, the study will be limited to static code analysis and will not cover runtime debugging or performance optimization. Additionally, the implementation of AI models in large-scale software projects will be discussed at a conceptual level rather than through real-world deployment.

Objectives of the Study

The primary objective of this study is to comprehensively explore and evaluate the effectiveness of AI-based bug detection and automated fixing techniques in software development. By assessing the challenges and opportunities associated with AI-driven debugging, this research aims to provide a structured and detailed understanding of how artificial intelligence can significantly contribute to improving software quality, development efficiency, and overall software reliability. Furthermore, the study seeks to analyze the impact of AI on the software lifecycle, considering aspects such as accuracy, adaptability, and integration with existing software engineering practices.

- To examine the role of AI in detecting and fixing software bugs across various programming languages and frameworks.
- To evaluate the accuracy, performance, and efficiency of machine learning models in identifying and resolving software defects.
- To identify the limitations and challenges associated with current AI-based debugging systems and propose potential improvements.
- To analyze how AI-driven debugging compares to traditional manual debugging methods in terms of speed, accuracy, and overall effectiveness.
- To propose a structured framework for seamlessly integrating AI-based bug detection and fixing solutions into modern software development workflows to enhance productivity and software security.

Expected Outputs

This study is expected to provide a comprehensive evaluation of AI-driven bug detection and fixing methodologies, offering an in-depth understanding of their strengths and limitations. The research will include a conceptual model or prototype that demonstrates the practical application of AI-based debugging tools. Additionally, the study will present comparative insights between AI-assisted and manual debugging techniques, outlining key differences in efficiency, accuracy, and scalability. The findings will serve as a valuable resource for software engineers, AI researchers, and technology companies seeking to integrate AI into software development for enhanced reliability and streamlined debugging processes.

REFERENCES

Gupta, R., Jain, A., & Kumar, P. (2021). "Machine learning approaches for software bug detection and classification: A systematic review." *Journal of Software Engineering Research and Development, 9*(1), 24-39.

Pradel, M., & Sen, K. (2018). "DeepBugs: A learning approach to name-based bug detection." *Proceedings of the ACM on Programming Languages, 2*(OOPSLA), 147.