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

Developing an AI-Driven Code Debugging System for Efficient

Software Development

Rationale/ Introduction

Fixing errors in computer programs is an important but time-consuming part of writing software. When developers write code, they often make mistakes that cause programs to crash or work incorrectly. Most existing tools can point out basic errors, like missing symbols or incorrect commands, but they do not understand deeper problems in the logic of the program. As a result, programmers must spend a lot of time searching for mistakes and figuring out how to fix them.

New technology, such as Artificial Intelligence (AI), can help solve this problem. AI systems can learn from many examples of correct and incorrect code, allowing them to suggest better fixes. By using AI and language processing techniques, a smart debugging tool could automatically find and fix mistakes in code. This would help developers work faster and reduce frustration. The goal of this research is to build an AI-powered system that helps programmers detect and correct errors in their code with less effort.

This research is important because it can make software development more efficient. Instead of spending hours searching for problems in their code, programmers could get instant suggestions from an AI tool. This would save time, improve the quality of software, and make it easier for beginners to learn programming. By testing and improving this system, the research will show how AI can be used to support developers and improve the coding process.

Significance of the Study

This study is useful because it shows how AI can help developers find and fix errors in their programs more easily. Many developers, especially beginners, struggle with debugging because it can be hard to understand why their code is not working. By creating a tool that provides clear and helpful suggestions, this research will make programming more accessible and less frustrating for people at all skill levels.

Another key benefit of this study is its potential to improve software quality. When programmers spend less time fixing errors, they can focus more on designing better

software. Faster debugging also means software projects can be completed sooner, helping businesses and developers meet deadlines more efficiently. By automating part of the debugging process, this research could lead to faster and more reliable software development.

This study will also explore possible challenges in using AI for debugging. AI systems sometimes make mistakes or provide unclear explanations for their suggestions. Ensuring that the AI model gives accurate and useful feedback is an important part of this research. By understanding these challenges, the study will provide recommendations for making AI debugging tools more effective and user-friendly.

Scope and Limitations of the Study

This research will focus on creating and testing an AI-based debugging system that can identify and fix common programming errors. It will examine how well the system works for two widely used programming languages, Python and Java. The study will also compare the AI system to traditional debugging tools to see if it offers better accuracy and efficiency.

However, this study will not cover every programming language. Some languages, like Assembly or C, may have different types of errors that require specialized debugging techniques. Additionally, the research will not focus on other areas of software development, such as testing or designing software. The main goal is to improve the debugging process, not to automate all aspects of coding.

Another limitation of this study is that AI models are not perfect. While the system will be trained to recognize many common errors, it may still struggle with highly complex or unusual coding problems. The study will explore ways to improve the AI's accuracy, but it will not create a system that can perfectly debug every piece of code.

Objectives of the Study

The main goal of this research is to create an AI-based tool that helps programmers find and fix mistakes in their code more easily. This will make software development faster and more efficient. The specific objectives of the study are:

1. Build and test an AI tool that can detect and suggest corrections for common coding errors.

2. Compare the AI tool's accuracy and speed with traditional debugging methods.
3. Identify challenges in using AI for debugging and suggest ways to improve its performance.

By achieving these objectives, the study aims to show how AI can assist developers in writing better software while reducing the time spent on fixing errors.

Expected Outputs

This research is expected to produce a working AI-based debugging tool that can help programmers identify and fix errors more quickly. The tool will be tested to see how well it works, and the results will be used to improve its performance. The findings will include insights into how AI can be used to make software development faster and more efficient.

Additionally, the study will provide a detailed comparison between the AI tool and traditional debugging methods. This will help developers understand the benefits and limitations of using AI for debugging. The research will also offer recommendations for improving AI debugging tools in the future, making them more accurate and helpful for programmers.

Beyond creating the tool, the research will explore common challenges in AI debugging. The study will look at issues such as incorrect suggestions, AI errors, and user trust in automated debugging. By addressing these challenges, the study will contribute to the development of better AI tools for programmers.

References

Chen, J., & Zhou, Y. (2020). *AI-driven software debugging: Challenges and future directions*. *Journal of Software Engineering Research and Development, 8*(2), 55-71.

Gupta, D., & Kumar, R. (2021). *Automated debugging with machine learning: A review of AI-based error detection techniques*. *ACM Transactions on Software Engineering and Methodology, 30*(1), 1-23.

Li, X., Wang, Q., & Zhang, Y. (2019). *Using deep learning for software fault localization: A survey*. *IEEE Transactions on Software Engineering, 45*(6), 1128-1150.