SUMMARY

Relevance of the issue: As software systems become larger and more complex, code quality is a critical concern. Poor code quality affects system maintainability and stability. It can also cause serious security issues. Traditional rule-based detection methods have limitations. They cannot handle complex business logic and large codebases well. These methods fail to understand the code's semantic structure deeply. They also miss potential logical errors and code defects. So, detecting and fixing code errors efficiently is urgent. Improving code quality is an important problem to solve.

Objectives: This study aims to propose an intelligent system. The system can detect and repair code defects automatically. It combines code semantic embedding and deep learning models. The goal is to improve the accuracy of code error detection. It also aims to make detection more efficient. This will help improve code quality and lower maintenance costs in software development..

Subject: The study focuses on hidden defects in the code. These defects lead to code quality problems. The main focus is on complex logical errors. Traditional rules cannot detect these errors. The research will explore the semantic representation of code. It will also study path extraction from Abstract Syntax Trees. Another focus is using deep learning for code error detection.

Research Theme: The main theme of this research is to create an automated code defect detection system. It uses semantic embedding and deep learning

models. The system extracts syntactic paths and semantic representations of the code. It combines these with attention mechanisms and graph neural networks. This improves code error detection accuracy. It also boosts cross-project adaptability.

Tasks to achieve the goal:

1) Create a code semantic embedding model using Abstract Syntax Tree (AST).

2) Use deep learning to improve the accuracy of error detection.

3) Build an automated system to detect errors in complex code. The system should identify potential errors accurately.

4) Test the model across different projects to check its stability. Make sure it works well in various codebases.

Scientific Novelty:This study introduces code semantic embedding technology as its main innovation. By using Abstract Syntax Tree (AST) to extract syntax paths from code and combining deep learning and attention mechanisms for in-depth analysis, this method can accurately capture complex dependency relationships and logical structures in code. Compared with traditional rule-based static analysis tools, it can not only recognize more detailed deep semantics, but also insight into hidden complex logic, thus demonstrating unparalleled advantages in improving the accuracy and efficiency of code error detection.

Practical significance: This study provides developers with an effective means to detect and fix code defects in a timely manner during the early

stages of software development, greatly reducing the negative impact of potential errors on software systems. Through this system, the efficiency of code review has been significantly improved, the cost of software maintenance has been effectively reduced, and the quality of project development has also skyrocketed. Especially in the current software industry, facing increasingly stringent requirements for code quality and security testing, this method undoubtedly has significant application value and practical significance.

Keywords: Code semantic embedding, error detection, deep learning, neural networks, Abstract syntax tree (AST).