DeepSeek R1 Computational Efficiency for Vulnerability Detection in High-Complexity Code

Assignee Research

June 4, 2026

Abstract

This report synthesises findings from 3 peer-reviewed papers addressing the following research question: What is the computational efficiency (inference time and memory usage) of Deepseek R1 when detecting vulnerabilities in high-cyclomatic-complexity code versus low-complexity code, as measured on the. 7 claims were extracted from source literature; 7 were independently verified against retrieved documents. An automated multi-reviewer quality assessment produced a score of 8.1/10. This report is a machine-generated literature synthesis and does not constitute original research.

1 Introduction

This paper examines: LLMs in Code Vulnerability Analysis: A Proof of Concept. Research question: What is the computational efficiency (inference time and memory usage) of Deepseek R1 when detecting vulnerabilities in high-cyclomatic-complexity code versus low-complexity code, as measured on the Big-Vul benchmark?.

2 Methodology

Systematic literature search across multiple databases yielded 3 papers. Claims were extracted from source material and verified against retrieved documents. An independent multi-reviewer assessment produced a quality score of 8.1/10.

3 Results

3 papers retrieved. 7 claims extracted; 7 independently verified. Quality review score: 8.1/10.

4 Limitations

This report is a machine-generated literature synthesis and does not constitute original research. Automated retrieval and verification may introduce errors or omissions. Review scores reflect automated assessment, not human peer review. Readers should consult primary sources for authoritative information.

5 Extracted Claims

Claim	Verified	Confidence
Traditional software security analysis methods struggle to keep pace with the scale and complexity of modern codebases.	✓	0.29
The paper explores the incorporation of code-specific and general-purpose Large Language Models (LLMs) to automate criti	✓	0.36
The study evaluates five pairs of recent LLMs, including both code-based and general-purpose open-source models, on two	✓	0.36
Fine-tuning uniformly outperforms both zero-shot and few-shot approaches across all tasks and models.	✓	0.33
Code-specialized models excel in zero-shot and few-shot settings on complex tasks, while general-purpose models remain n	✓	0.37
Discrepancies among CodeBLEU, CodeBERTScore, BLEU, and ChrF highlight the inadequacy of current metrics for measuring re	✓	0.28
This study contributes to the software security community by investigating the potential of advanced LLMs to improve vul	✓	0.32

References

- https://openalex.org/W7162149865
- https://doi.org/10.1186/s42400-025-00518-7
- https://openalex.org/W7124227854