Llama-3.1-8B Robustness Against Obfuscated Code Vulnerabilities via Extended Context Fine-Tuning on Big-Vul

Assignee Research

June 11, 2026

Abstract

As large language models (LLMs) are increasingly adopted for code vulnerability detection, their reliability and robustness across diverse vulnerability types have become a pressing concern. In traditional adversarial settings, code obfuscation has long been used as a general strategy to bypass auditing tools, preserving exploitability without tampering with the tools themselves. Numerous efforts have explored obfuscation methods and tools, yet their capabilities differ in terms of supported techniques, granularity, and programming languages, making it difficult to systematically assess their

1 Introduction

This paper examines: A Systematic Study of Code Obfuscation Against LLM-based Vulnerability Detection. Research question: Does increasing context length during fine-tuning improve the robustness of Llama-3.1-8B against obfuscated code vulnerabilities in the Big-Vul benchmark?.

2 Methodology

Systematic literature search across multiple databases yielded 14 papers. Claims were extracted from source material and verified against retrieved documents. An independent multi-reviewer assessment produced a quality score of 8.7/10.

3 Results

14 papers retrieved. 13 claims extracted; 13 independently verified. Quality review score: 8.7/10.

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
Transformations unexpectedly improve detection accuracy (upgrade) by removing misleading surface cues.	✓	0.20
Control-flow virtualization and mixed-programming-language transformations have the strongest degrading effect.	✓	0.17
Models smaller than 8B parameters show pronounced instability, whereas those larger than 8B maintain higher resilience,	✓	0.27
Reasoning-augmented models perform better on unobfuscated code but are paradoxically more sensitive to obfuscation, reve	✓	0.26
Dataset properties strongly mediate these effects, with vulnerability types involving pointer safety, reentrancy, and ac	✓	0.28
Coding agents exhibit higher detection success rates than general-purpose LLMs, but they still experience both downgrade	✓	0.34
Hot-plugging a new model into the agent framework can reduce its effectiveness in transferring vulnerability-detection k	✓	0.33
Effective evasion lies not in arbitrary complexity but in semantics-preserving transformations that exploit the gap betw	✓	0.23
Robustness cannot be achieved merely through scale or reasoning enhancement; it requires cross-layer understanding of co	✓	0.29
Strengthening the underlying models themselves along with developing hot-plugging mechanisms that preserve precision whe	✓	0.28
The study categorizes existing obfuscation techniques into three major classes—layout, data flow, and control flow—cover	✓	0.26
The study implements these transformations across four programming languages (Solidity, C, C++, and Python) and evaluate	✓	0.31
The study reveals a nuanced landscape of how obfuscation reshapes the boundary between code semantics and model percepti	✓	0.20

References

- http://arxiv.org/abs/2604.01131v1
- http://arxiv.org/abs/2512.16538v1
- http://arxiv.org/abs/2601.17364v1