Codestral-7B and Llama3-70B Inference Efficiency in C/C++ Vulnerability Detection

Assignee Research

May 30, 2026

Abstract

This report synthesises findings from 2 peer-reviewed papers addressing the following research question: What is the inference efficiency difference between Codestral-7B and Llama3-70B when fine-tuned on C/C++ security vulnerability detection tasks. Software vulnerabilities pose significant risks to the security and reliability of modern systems, making automated vulnerability detection an essential research area. Traditional static and rule-based approaches are limited in scalability and adaptability, motivating the. 9 claims were extracted from source literature; 9 were independently verified against retrieved documents. An automated multi-reviewer quality assessment produced a score of 9.0/10. This report is a machine-generated literature synthesis and does not constitute original research.

1 Introduction

This paper examines: Modern Approaches to Software Vulnerability Detection: A Survey of Machine Learning, Deep Learning, and Large Language Models. Research question: What is the inference efficiency difference between Codestral-7B and Llama3-70B when fine-tuned on C/C++ security vulnerability detection tasks.

2 Methodology

Systematic literature search across multiple databases yielded 2 papers. Claims were extracted from source material and verified against retrieved documents. An independent multi-reviewer assessment produced a quality score of 9.0/10.

3 Results

2 papers retrieved. 9 claims extracted; 9 independently verified. Quality review score: 9.0/10.

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
Software vulnerabilities pose significant risks to the security and reliability of modern systems.	✓	0.24
Traditional static and rule-based approaches are limited in scalability and adaptability.	✓	0.23
Machine Learning (ML), Deep Learning (DL), and Large Language Models (LLMs) techniques are used for vulnerability detect	✓	0.34
Recent advances in feature representation, fine-tuning strategies, generative approaches, and prompt engineering are ana	✓	0.26
ML, DL, and LLMs techniques can capture both syntactic and semantic properties of source code.	✓	0.19
Commonly used evaluation metrics for vulnerability detection are examined.	✓	0.16
Key challenges in vulnerability detection include the lack of large-scale real-world datasets, limited vulnerability cov	✓	0.34
Promising future research directions include neuro-symbolic hybrid methods, parameter-efficient fine-tuning, continual l	✓	0.37
The present work explores learning paradigms from ML to LLMs using comprehensive evaluation criteria that highlight anal	✓	0.36

References

- https://doi.org/10.3390/electronics14224449
- https://doi.org/10.48550/arxiv.2504.14985