Scaling Effects on Vulnerability Classification Accuracy in Llama3, Codestral, and DeepSeek R1

Assignee Research

May 30, 2026

Abstract

This report synthesises findings from 14 peer-reviewed papers addressing the following research question: What is the impact of model size scaling on the pass@1 accuracy of Llama3, Codestral, and Deepseek R1 when evaluating vulnerability classification on the Big-Vul dataset. Recent advancements in generative AI have led to the widespread adoption of large language models (LLMs) in software engineering, addressing numerous long-standing challenges. However, a comprehensive study examining the capabilities of LLMs in software vulnerability detection. 14 claims were extracted from source literature; 0 were independently verified against retrieved documents. An automated multi-reviewer quality assessment produced a score of 2.8/10. This report is a machine-generated literature synthesis and does not constitute original research.

1 Introduction

This paper examines: Benchmarking Large Language Models for Multi-Language Software Vulnerability Detection. Research question: What is the impact of model size scaling on the pass@1 accuracy of Llama3, Codestral, and Deepseek R1 when evaluating vulnerability classification on the Big-Vul dataset?.

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 2.8/10.

3 Results

14 papers retrieved. 14 claims extracted; 0 independently verified. Quality review score: 2.8/10.

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
The top three programming languages in 2023 were Python, Java, and JavaScript according to the IEEE Spectrum survey.	×	0.05
Vulnerability data was collected from the National Vulnerability Database (NVD) up to March 28, 2024.	×	0.04
Vulnerability-Fixing Commits (VFCs) were identified by searching for references in CVE entries that include links contai	×	0.02
VFCs from Python, Java, and JavaScript were considered if they modify at least one file written in the corresponding lan	×	0.04
Pre-commit versions of changed functions were labeled as vulnerable, while post-commit versions and all unchanged functi	×	0.02
Tree-sitter was used to extract changed and unchanged functions from VFCs.	×	0.01
Function de-duplication was performed using MD5 hashes without applying code normalization to prevent data leakage.	×	0.02
A time-aware setting was implemented with June 1, 2023, as the cutoff date to split training and testing data.	×	0.02
VFCs submitted after June 1, 2023, comprise the test set, while those submitted before are used for training.	×	0.02
The most recent 10% of training commits form the validation set, while the earlier 90% constitute the training set.	×	0.04
The effectiveness of LLMs varies across programming languages, with the best LLM achieving F1 scores of 0.200 for Python	×	0.06
LLMs consistently perform better on the JavaScript dataset, with the instruction-tuned StarCoder-2 achieving the highest	×	0.05
CodeQwen1.5, DeepSeek-Coder, CodeGemma, Starcoder-2, and CodeLlama were selected as the five best-performing open-source	×	0.08
SLMs such as CodeBERT, GraphCodeBERT, UniXcoder, CodeT5, and CodeT5+ were considered for their proven effectiveness in s	×	0.06

References

- http://arxiv.org/abs/2503.01449v1
- http://arxiv.org/abs/2511.01941v1
- http://arxiv.org/abs/2503.10486v2