DeepSeek-R1 Context Window Scaling for Security Vulnerability Detection in Code

Assignee Research

May 30, 2026

Abstract

This report synthesises findings from 16 peer-reviewed papers addressing the following research question: What is the impact of context window scaling on the security vulnerability detection performance of DeepSeek-R1 compared to other models across different code lengths and complexity levels. Many studies have developed Machine Learning (ML) approaches to detect Software Vulnerabilities (SVs) in functions and fine-grained code statements that cause such SVs. However, there is little work on leveraging such detection outputs for data-driven SV assessment to give. 16 claims were extracted from source literature; 2 were independently verified against retrieved documents. An automated multi-reviewer quality assessment produced a score of 4.5/10. This report is a machine-generated literature synthesis and does not constitute original research.

1 Introduction

This paper examines: On the Use of Fine-grained Vulnerable Code Statements for Software Vulnerability Assessment Models. Research question: What is the impact of context window scaling on the security vulnerability detection performance of DeepSeek-R1 compared to other models across different code lengths and complexity levels?.

2 Methodology

Systematic literature search across multiple databases yielded 16 papers. Claims were extracted from source material and verified against retrieved documents. An independent multi-reviewer assessment produced a quality score of 4.5/10.

3 Results

16 papers retrieved. 16 claims extracted; 2 independently verified. Quality review score: 4.5/10.

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 study is the first to leverage data-driven models for automating function-level software vulnerability (SV) assessme	✓	0.21
The study empirically demonstrates that incorporating the context of vulnerable statements improves SV assessment perfor	✓	0.20
The Common Vulnerability Scoring System (CVSS) has two main versions, version 2 and version 3.	×	0.09
CVSS version 3 came into effect in 2015.	×	0.01
The vulnerability CVE-2004-0113, first found in 2004, was exploited in 2018.	×	0.02
In the dataset analyzed, 95.3% of vulnerabilities have a Network Access Vector.	×	0.02
In the dataset analyzed, 74.5% of vulnerabilities have Low Access Complexity.	×	0.05
In the dataset analyzed, 78.2% of vulnerabilities require no Authentication.	×	0.02
In the dataset analyzed, 67.9% of vulnerabilities are classified with Medium Severity.	×	0.04
Using Program Slicing (PS) context with vulnerable statements yields an average assessment score improvement of 5.2 comp	×	0.10
Using Surrounding context with vulnerable statements yields an average assessment score improvement of 3.6 compared to t	×	0.11
Using Function context with vulnerable statements yields an average assessment score improvement of 5.9 compared to the	×	0.13
Removing Program Slicing (PS) context results in an average assessment score decrease of 3.8.	×	0.03
Removing Surrounding context results in an average assessment score decrease of 5.5.	×	0.03
The vulnerability CVE-2017-1000487 exists in the Plexus-utils project.	×	0.02
In the fixing commit b38a1b3 for CVE-2017-1000487, line 8 containing 'unifyQuotes(dir)' is the vulnerable statement.	×	0.02

References

- http://arxiv.org/abs/2203.08417v1
- http://arxiv.org/abs/1002.1154v1
- http://arxiv.org/abs/2506.11022v2