Scalability of CodeT5-Based Vulnerability Detection in IDEs for Incremental Code Changes

Assignee Research

June 1, 2026

Abstract

This report synthesises findings from 15 peer-reviewed papers addressing the following research question: What is the scalability of CodeT5-based vulnerability detection in IDE environments when processing incremental code changes versus full-file analysis, as measured by latency per code edit and. In the rapidly evolving software development landscape, Python stands out for its simplicity, versatility, and extensive ecosystem. Python packages, as units of organization, reusability, and distribution, have become a pressing concern, highlighted by the considerable number of. 19 claims were extracted from source literature; 1 was independently verified against retrieved documents. An automated multi-reviewer quality assessment produced a score of 3.8/10. This report is a machine-generated literature synthesis and does not constitute original research.

1 Introduction

This paper examines: An Empirical Study of Vulnerabilities in Python Packages and Their Detection. Research question: What is the scalability of CodeT5-based vulnerability detection in IDE environments when processing incremental code changes versus full-file analysis, as measured by latency per code edit and detection accuracy on a modified CodeT5-Python dataset with injected vulnerabilities?.

2 Methodology

Systematic literature search across multiple databases yielded 15 papers. Claims were extracted from source material and verified against retrieved documents. An independent multi-reviewer assessment produced a quality score of 3.8/10.

3 Results

15 papers retrieved. 19 claims extracted; 1 independently verified. Quality review score: 3.8/10.

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 study collected vulnerability reports from GitHub Advisories, Snyk, and Huntr.	×	0.03
The study gathered 2,379 vulnerability reports from GitHub Advisories.	×	0.04
The study gathered 930 vulnerability reports from Snyk.	×	0.04
The study gathered 321 vulnerability reports from Huntr.	×	0.04
The total number of initially gathered vulnerability reports was 3,630.	×	0.04
The initial screening criteria required reports to include fix commits addressing the corresponding vulnerabilities.	×	0.03
The initial screening criteria required repositories to be accessible and commits not rolled back or deleted at the time	×	0.03
The initial screening criteria required reports to not be duplicates of other reports.	×	0.01
The initial collection resulted in 1,767 unique vulnerability reports after screening.	×	0.03
Rule-based static analysis methods such as CodeQL, PySA, and Bandit typically operate at the project level.	×	0.06
ML-based static analysis methods generally work at the function level.	×	0.06
The PyVul benchmark accommodates vulnerabilities at both the commit and function levels.	✓	0.16
For the commit-level benchmark, the 1,767 collected commits were used as patched, non-vulnerability samples.	×	0.06
For the commit-level benchmark, the direct parent version of the collected commits on the main branch were used as vulne	×	0.05
The study utilized LLM in-context learning capabilities due to insufficient fine-tuning data for the annotation task.	×	0.03
The study implemented prompt engineering practices including system role definition, few-shot learning, and chain-of-tho	×	0.02
The system role for the LLMs was designated as a security expert.	×	0.03
The LLMs were instructed to state reasons before yielding the final answer.	×	0.03
The benchmark construction retained only commits with at least one relevant function change.	×	0.03

References

- http://arxiv.org/abs/2201.08441v1
- http://arxiv.org/abs/2509.04260v1
- http://arxiv.org/abs/2604.21917v1