# Context Window Size and False Positive Rates in Small Language Models for Code Defect Detection

Assignee Research

June 7, 2026

**Abstract**

This report synthesises findings from 13 peer-reviewed papers addressing the following research question: What is the correlation between context window size and false positive rates in small language models evaluated on the CodeXGLUE defect detection task. 18 claims were extracted from source literature; 4 were independently verified against retrieved documents. An automated multi-reviewer quality assessment produced a score of 5.2/10. This report is a machine-generated literature synthesis and does not constitute original research.

## 1 Introduction

This paper examines: An Empirical Evaluation of Locally Deployed LLMs for Bug Detection in Python Code. Research question: What is the correlation between context window size and false positive rates in small language models evaluated on the CodeXGLUE defect detection task?.

## 2 Methodology

Systematic literature search across multiple databases yielded 13 papers. Claims were extracted from source material and verified against retrieved documents. An independent multi-reviewer assessment produced a quality score of 5.2/10.

## 3 Results

13 papers retrieved. 18 claims extracted; 4 independently verified. Quality review score: 5.2/10.

## 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 |
|---|---|---|
| Locally executed models achieve accuracy between 43% and 45% in bug detection. | ✓ | 0.26 |
| Locally executed models produce a large proportion of partially correct responses that identify problematic code regions | ✓ | 0.32 |
| Performance varies significantly across projects, highlighting the importance of codebase characteristics. | ✓ | 0.24 |
| Local models can identify a meaningful share of bugs, though precise localization remains difficult for locally executed | ✓ | 0.33 |
| Performance degrades when bugs require cross-function reasoning. | × | 0.04 |
| Defect complexity is the primary factor governing detection accuracy. | × | 0.05 |
| Model performance drops systematically as the number of co-occurring defects increases. | × | 0.03 |
| Open-weight models can approach proprietary system performance on converting unstructured bug reports into structured fo | × | 0.04 |
| The availability of open-weight models such as LLaMA and Mistral has made local deployment a practical option, eliminati | × | 0.06 |
| Prior work has largely evaluated large cloud-hosted models. | × | 0.07 |
| BugsInPy is a curated set of real Python bugs with reproducible test cases drawn from well-known open-source projects. | × | 0.06 |
| There were reproducibility issues in the BugsInPy dataset, and revisions were proposed to support more reliable evaluati | × | 0.07 |
| Software bugs remain a major challenge in modern software development. | × | 0.05 |
| Identifying and fixing bugs consumes a substantial portion of development time. | × | 0.04 |
| Undetected errors can lead to serious failures in production systems. | × | 0.07 |
| A software bug can be defined as an unintended error in source code that results in incorrect or unexpected behavior. | × | 0.05 |
| Recent advances in large language models (LLMs) have demonstrated promising capabilities in code understanding and gener | × | 0.11 |
| Models such as GPT-4 and Claude have shown strong performance on tasks related to bug detection. | × | 0.09 |

## References


- http://arxiv.org/abs/2508.15478v2
- http://arxiv.org/abs/2601.08844v1
- http://arxiv.org/abs/2604.23361v1