Scaling Model Size and Training Objectives in Code Vulnerability Detection

Assignee Research

June 8, 2026

Abstract

This report synthesises findings from 16 peer-reviewed papers addressing the following research question: What is the impact of model size scaling (e.g., CodeT5-110M vs. CodeT5-220M) on vulnerability detection accuracy when trained with contrastive vs. MLM objectives. 15 claims were extracted from source literature; 1 was independently verified against retrieved documents. An automated multi-reviewer quality assessment produced a score of 4.6/10. This report is a machine-generated literature synthesis and does not constitute original research.

1 Introduction

This paper examines: Automated Code-centric Software Vulnerability Assessment: How Far Are We? An Empirical Study in C/C++. Research question: What is the impact of model size scaling (e.g., CodeT5-110M vs. CodeT5-220M) on vulnerability detection accuracy when trained with contrastive vs. MLM objectives?.

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 4.6/10.

3 Results

16 papers retrieved. 15 claims extracted; 1 independently verified. Quality review score: 4.6/10.

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
Employing multi-task learning allows deep learning models to achieve an average 8–22% increase in Matthews Correlation C	✓	0.18
The Big-Vul dataset consists of 188,636 functions written in C/C++.	×	0.04
The Big-Vul dataset covers 91 different software vulnerability types.	×	0.07
The Big-Vul dataset originates from 348 different projects on GitHub.	×	0.04
The study extracted seven CVSS metrics as output labels for the models.	×	0.04
The CVSS metric 'Access Vector' describes the medium or technique required to attack or penetrate a system.	×	0.01
The CVSS metric 'Access Complexity' describes the complexity required to exploit the software vulnerability and initiate	×	0.08
The CVSS metric 'Confidentiality' measures the extent to which an attack allows unauthorized access to system sensitive	×	0.02
The CVSS metric 'Integrity' measures the extent to which an attack allows unauthorized modifications to system data.	×	0.02
The CVSS metric 'Availability' measures the extent to which an attack restricts accessibility to system data, resources,	×	0.03
In the dataset distribution, the 'Availability' metric has the highest count of instances at 2,280.	×	0.02
In the dataset distribution, the 'Authentication' metric has the lowest count of instances at 566.	×	0.02
The total number of instances in the dataset used for the study is 12,063.	×	0.03
The CodeBERT model required 9,235 seconds of training time.	×	0.04
The LGBM + BoST model required 347 seconds of training time for the Access Vector metric.	×	0.03

References

- http://arxiv.org/abs/2106.05967v3
- http://arxiv.org/abs/2304.06875v4
- http://arxiv.org/abs/2407.17053v4