Pretraining Objectives and CodeT5 Performance on CWE-200 Vulnerability Detection

Assignee Research

June 7, 2026

Abstract

This report synthesises findings from 16 peer-reviewed papers addressing the following research question: How does the choice of pretraining objective (e.g., masked language modeling vs. contrastive learning) affect CodeT5's performance on the CWE-200 benchmark when fine-tuned for vulnerability detection. 13 claims were extracted from source literature; 0 were independently verified against retrieved documents. An automated multi-reviewer quality assessment produced a score of 3.2/10. This report is a machine-generated literature synthesis and does not constitute original research.

1 Introduction

This paper examines: Detecting Vulnerabilities from Issue Reports for Internet-of-Things. Research question: How does the choice of pretraining objective (e.g., masked language modeling vs. contrastive learning) affect CodeT5's performance on the CWE-200 benchmark when fine-tuned for vulnerability detection?.

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 3.2/10.

3 Results

16 papers retrieved. 13 claims extracted; 0 independently verified. Quality review score: 3.2/10.

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
The study uses BoW, BERT, GloVe, and W2V NLP features for classification.	×	0.07
The corpus is re-sampled to balance the quantities of data classes by increasing the population of vulnerability-indicat	×	0.07
GPT-4o is used to classify vulnerability-indicating issues with handcrafted prompts and a constant Seed decoding paramet	×	0.05
The model's performance is measured using pass@3, which is the upmost correct responses given three generation attempts	×	0.05
The performance of the models is evaluated against a random guesser with an AUC of 0.50.	×	0.05
GitHub resolved issues are mined if they are not a pull request, if both title and submission description are non-empty,	×	0.03
Issues tagged as 'security' are included as vulnerability-indicating (6,696 issues).	×	0.07
Issues tagged as 'bug' but not as 'security' are included as non-vulnerability-indicating (528,494 issues).	×	0.07
A corpus of 11,000 issues is randomly sampled with stratification (95% CL and $\pm$5% CI).	×	0.05
Surrogates for vulnerability-indicating labels are: [CVE, GitHub, version, use, vulnerability, issue, security, severity	×	0.06
Surrogates for non-vulnerability-indicating labels are: [data, file, foundry, type, filter, fail, error, debug, default,	×	0.04
A BERT MLM is fine-tuned by hiding occurrences of the Surrogates in the corpus using the [MASK] token.	×	0.07
10-CV is used with training epochs=2 and batch size=5.	×	0.05

References

- http://arxiv.org/abs/2407.05862v1
- http://arxiv.org/abs/2511.01941v1

- http://arxiv.org/abs/2204.03214v2