Dataset Alignment Effects on Codestral False Positive Rates in SWCC Vulnerability Severity Prediction

Assignee Research

May 30, 2026

Abstract

This report synthesises findings from 15 peer-reviewed papers addressing the following research question: What is the impact of dataset alignment on the false positive rate of Codestral when evaluating vulnerability severity predictions on the SWCC benchmark. Static Application Security Testing (SAST) tools play a vital role in modern software development by automatically detecting potential vulnerabilities in source code. However, their effectiveness is often limited by a high rate of false positives, which wastes developer's effort. 16 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: FP-Predictor - False Positive Prediction for Static Analysis Reports. Research question: What is the impact of dataset alignment on the false positive rate of Codestral when evaluating vulnerability severity predictions on the SWCC benchmark?.

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 3.2/10.

3 Results

15 papers retrieved. 16 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
FPPredictor was trained on the CamBenchCAP dataset.	×	0.09
FPPredictor was evaluated on the CryptoAPI-Bench benchmark.	×	0.07
FPPredictor achieved an overall accuracy of up to 96.6% on the CryptoAPI-Bench benchmark.	×	0.14
The CryptoAPI-Bench dataset contained 91 vulnerable (True Positive) cases according to ground truth.	×	0.05
The CryptoAPI-Bench dataset contained 27 secure (False Positive) cases according to ground truth.	×	0.07
FPPredictor predicted 2 out of 91 vulnerable cases as false positives.	×	0.06
FPPredictor predicted 1 out of 27 secure cases as a false positive.	×	0.07
FPPredictor's accuracy on the false positive subset (secure files) was approximately 3.7% based on automated predictions	×	0.07
FPPredictor's accuracy on the true positive subset (vulnerable files) was approximately 97.8%.	×	0.04
Manual inspection revealed that 22 out of 26 cases initially marked as incorrectly predicted actually exhibited security	×	0.06
The test case 'CredentialInString-Corrected.java' is labeled as non-vulnerable in CryptoAPI-Bench.	×	0.04
The file 'CredentialInString-Corrected.java' uses an unprotected key in line 17.	×	0.01
The file 'CredentialInString-Corrected.java' initializes the AES cipher in CBC mode in lines 21 and 22.	×	0.01
AES-CBC mode is discouraged due to susceptibility to padding oracle attacks.	×	0.01
Four test cases containing conditional ('if') statements or control-flow dependencies were identified as challenging for	×	0.08
CogniCrypt was used as the static analysis tool to provide input for FPPredictor.	×	0.07

References

- http://arxiv.org/abs/2305.16615v1
- http://arxiv.org/abs/2505.12925v2
- http://arxiv.org/abs/2603.10558v1