Automated Code Quality and Security Auditor (A-CQSA) Using Large Language Models with Composite Risk Scoring

Abhinav Singh
Department of Computing Technologies SRM IST, Chennai, India
Cherry Jain
Department of Computing Technologies SRM IST, Chennai, India

Abstract

Ensuring software security and maintaining high code quality are critical challenges in modern software development. Traditional static analysis tools effectively identify known vulnerability patterns but rely on rule-based detection, lacking contextual understanding and producing high false-positive rates that burden developers. This paper proposes the Automated Code Quality and Security Auditor (A-CQSA), a hybrid framework integrating static analysis tools with Large Language Models (LLMs) to deliver context-aware, explainable code auditing. A-CQSA introduces three novel contributions: (1) an LLM-based false positive filtering pipeline that achieved a 40.3% reduction in false positives across 15 evaluated repositories; (2) a Composite Security Risk Scoring Engine that aggregates CVSS severity, CWE exploitability, code reachability, file exposure, and historical false-positive propensity into a normalized 0–100 score; and (3) a Semantic Caching mechanism that reduced LLM API latency by 62% on repeated vulnerability patterns. Experimental evaluation on open-source repositories demonstrates that A-CQSA achieves a precision of 0.87, recall of 0.91, and F1-score of 0.89, outperforming standalone static analysis tools and providing actionable remediation suggestions with developer-friendly explanations.

Index Terms

Software Security, Static Analysis, Large Lan-guage Models, False Positive Reduction, DevSecOps, Explainable AI, Vulnerability Detection, Semantic Caching, Risk Scoring