Scientific Journal of Faculty of Education, Misurata University, Libya
Volume (14), Issue (1)
ISSN: 2710-4141

https://doi.org/10.36602/sjfe.2026.v14.i01.06

Received: 06/10/2026 Accepted: 27/10/2026 Published: 29/10/2026

An AHP Framework for Evaluating Large Language Models in Software Vulnerability Assessment

Mohamed Abugharsa[1]*
[1] Department of Information Systems, Faculty of Information Technology, Misurata University, Libya
m.abugharsa@it.misuratau.edu.ly

Ali M. Abdulshahed[2]
[2] Department of Electrical and Electronic Engineering, Faculty of Engineering, Misurata University, Libya
a.abdulshahed@eng.misuratau.edu.ly

Abstract

Large language models (LLMs) can assist software security review, but detecting more weaknesses may increase false alarms, unreliable patches, or higher costs. This methodological paper specifies a reproducible Analytic Hierarchy Process (AHP) framework for evaluating LLM-assisted vulnerability assessment in a synthetic banking-information application. Applying the framework to a local operational banking system was not feasible because obtaining the required security approvals was difficult; no production banking infrastructure was tested. Six objectives are linked to measurable indicators covering ground-truth completeness, detection, classification and severity assessment, secure remediation, reliability and resource use, and decision sensitivity. The protocol specifies 12 vulnerable and 12 corrected variants, four scenarios, repeated runs, blinded adjudication, CVSS v4.0 assessment, regression tests, and paired analysis. A comparison with SecurityEval, CyberSecEval, CyberSecEval 2, and BountyBench positions the proposed contribution as linking verified outcomes to stakeholder priorities rather than demonstrating superior benchmark performance. An exact reciprocal AHP example uses weights of 0.40, 0.30, 0.20, and 0.10. A separate simulation yields group weights of 0.3565, 0.2920, 0.2187, and 0.1328, with a consistency ratio of 0.0008. These are constructed examples, not elicited expert judgments. No LLM benchmarking was performed, and no model ranking is claimed. Empirical validation requires archived runs, independently assessed outcomes, and human stakeholder judgments; production transfer additionally requires institutional authorization and service-specific validation.

Keywords: Large language models; Software vulnerability assessment; Analytic hierarchy process; Secure code review; CVSS v4.0; Banking web application

Copyright: ©2026 by the authors. This work is licensed under a (CC BY-4.0).

Scientific Journal of Faculty of Education, Misurata University, Libya
Volume (14), Issue (1)
ISSN: 2710-4141

https://doi.org/10.36602/sjfe.2026.v14.i01.06

Received: 06/10/2026 Accepted: 27/10/2026 Published: 29/10/2026

إطار قائم على عملية التحليل الهرمي لتقييم نماذج اللغة الكبيرة في تقييم ثغرات البرمجيات

محمد بشير ابوغرسة[1]*
[1] قسم نظم المعلومات، كلية تقنية المعلومات، جامعة مصراتة، ليبيا

علي محمد عبد الشاهد[2]
[2] قسم الهندسة الكهربائية والإلكترونية، كلية الهندسة، جامعة مصراتة، ليبيا

ملخص البحث

يمكن لنماذج اللغة الكبيرة أن تساعد في مراجعة أمن البرمجيات، لكن زيادة كشف الثغرات قد تقترن بإنذارات كاذبة أو إصلاحات غير موثوقة أو تكلفة أعلى. تقدم هذه الورقة المنهجية إطارًا قابلًا لإعادة التطبيق قائمًا على عملية التحليل الهرمي لتقييم هذه النماذج في تطبيق اصطناعي للمعلومات المصرفية. تعذر التطبيق على منظومة مصرفية محلية تشغيلية بسبب صعوبة الحصول على الموافقات الأمنية اللازمة؛ ولم تُختبر أي بنية مصرفية إنتاجية. ويربط الإطار ستة أهداف بمؤشرات قابلة للقياس تشمل اكتمال البيانات المرجعية، والكشف، والتصنيف والخطورة، والإصلاح الآمن، والموثوقية والموارد، وحساسية القرار للأوزان. يحدد البروتوكول اثنتي عشرة حالة معيبة ونظائرها المصححة، وأربعة سيناريوهات، وتكرارات مستقلة، وتقييمًا محجوب الهوية، واختبارات أمنية ووظيفية، وتحليلًا إحصائيًا مقترنًا. وتوضح المقارنة بالدراسات السابقة أن الإسهام المقترح هو ربط النتائج المتحقق منها بأولويات أصحاب المصلحة، دون ادعاء تفوق تجريبي. يستخدم المثال الحسابي الأول أوزانًا تبلغ 0.40 و0.30 و0.20 و0.10؛ وتنتج المحاكاة المنفصلة أوزانًا جماعية تبلغ 0.3565 و0.2920 و0.2187 و0.1328، بنسبة اتساق 0.0008. هذه أمثلة منشأة وليست أحكام خبراء مشاركين. لم تُجر اختبارات مقارنة فعلية للنماذج، ولم يُستنتج ترتيب لأدائها. يتطلب التحقق التجريبي حفظ التشغيلات، والتحكيم المستقل، وجمع أحكام بشرية، بينما يتطلب الانتقال إلى الأنظمة التشغيلية تصريحًا مؤسسيًا وتحققًا خاصًا بالخدمة.

الكلمات المفتاحية: أمن البرمجيات، التطبيقات المصرفية، عملية التحليل الهرمي، كشف الثغرات، نماذج اللغة الكبيرة

Copyright: ©2026 by the authors. This work is licensed under a (CC BY-4.0).

Scientific Journal of Faculty of Education, Misurata University, Libya
Volume (14), Issue (1)
ISSN: 2710-4141

https://doi.org/10.36602/sjfe.2026.v14.i01.06

Received: 06/10/2026 Accepted: 27/10/2026 Published: 29/10/2026

1 Introduction

Software security evaluation is a multi-objective problem. A model may locate a vulnerable function but misclassify its root cause; another may produce plausible patches that fail security regression tests. Selection by a single accuracy score obscures these trade-offs. The OWASP Top 10:2025 organizes prevalent web application risk classes, MITRE CWE provides a weakness vocabulary, and the FIRST CVSS v4.0 defines a severity framework. These resources describe different aspects of a finding and should not be treated as interchangeable labels (OWASP Foundation, 2025; MITRE, 2026; FIRST, 2023).

The Libyan context makes this question timely. The Central Bank of Libya (CBL) reported that, during 1–23 January 2025, LYPay had 1,364,284 individual subscribers and processed 56,636 purchase and transfer transactions worth LYD 507.6 million; the same bulletin recorded 219,676 OnePay transactions worth LYD 1.134 billion (Central Bank of Libya, 2025a). These are service-specific administrative figures for a short window, not estimates of national coverage or security outcomes. In July 2025, the CBL reported that electronic transactions in the first half of 2025 had increased by 75% and point-of-sale terminals increased by 92% relative to the first half of 2024 (Central Bank of Libya, 2025c). The two measures have different denominators and should not be combined into a single adoption index.

The CBL announced a National Financial Inclusion Strategy for 2025–2029 and outlined initiatives involving digital government payments, data protection, digital banking identity, and domestic transfers (Central Bank of Libya, 2025b). These announcements set policy direction; they do not demonstrate that any particular application is secure or that every initiative has been deployed. As electronic payment channels become more salient, defects in transaction-facing software can affect confidentiality, integrity, availability, fraud exposure, and public confidence. The present synthetic web application is an intentionally narrower research instrument, not a replica of payment infrastructure.

1.1 Statement of the Problem

The research question is: How can a controlled evaluation combine detection, classification, severity, patch quality, reliability, and resource use into a defensible choice of an LLM-assisted security-review system? The contribution is a specified evaluation design and a reproducible AHP illustration. Applying the framework to an operational local banking system was not feasible because obtaining the required security approvals was difficult. The alternative is a researcher-controlled synthetic application inspired by publicly observable banking-information services, with no access to production code, customer records, or banking endpoints.

1.2 Aim and Measurable Objectives

The main objective is to specify a reproducible AHP framework that links independently verifiable security outcomes to stakeholder priorities. Table 1 translates this aim into six measurable objectives for an empirical follow-up. Counts and coverage requirements are protocol targets, not observed achievements or institutionally approved performance thresholds.

Copyright: ©2026 by the authors. This work is licensed under a (CC BY-4.0).

Scientific Journal of Faculty of Education, Misurata University, Libya
Volume (14), Issue (1)
ISSN: 2710-4141

https://doi.org/10.36602/sjfe.2026.v14.i01.06

Received: 06/10/2026 Accepted: 27/10/2026 Published: 29/10/2026

Table 1: Objectives linked to indicators and verification

Objective	Indicator	Planned verification target	Evidence
O1 Validate the test corpus	Validated pairs / 12; complete artifact records / 24.	12 vulnerable–corrected pairs; 100% with hashes, oracle tests, functional tests, and independent review.	Ground-truth manifest and test logs; Section 3.2.
O2 Compare detection	TP, FP, FN; precision, recall, F1 and paired differences.	720 scheduled evaluations per LLM; report denominators, failed tasks, and 95% intervals. No superiority threshold is assumed.	Blinded adjudication; Sections 3.3–3.6.
O3 Assess classification and severity	CWE and OWASP accuracy; valid-vector rate; CVSS score MAE.	Adjudicate 100% of findings; report matched-finding and end-to-end denominators and all invalid vectors.	Two assessors and resolution records; Section 3.5.
O4 Verify secure remediation	Secure-patch pass rate; functional regression rate.	Test 100% of attempted patches. A successful patch must remove the defect, preserve functionality, and pass review.	Attack and functional regression logs; Appendix C.
O5 Measure reliability and resources	Run agreement; evidence score; median and 95th-percentile latency; tokens and API cost.	Five runs per condition; archive status and available usage for 100% of attempts, with missing data identified.	Run archive, scoring rubric, and dated pricing; Section 3.5.
O6 Evaluate decision robustness	AHP weights, CR, weighted scores, and rank acceptability.	At least five human specialists in follow-up; review CR > 0.10; ±50% weight variation and 1,000 seeded weight draws.	Individual matrices and sensitivity script; Section 4 and Appendix B.

This article completes the protocol specification and demonstrates AHP arithmetic with constructed judgments. O1–O5 and the human-panel and model-ranking components of O6 remain prospective. Evidence of meeting a protocol target must be derived from the specified records. Model acceptance floors for recall, cost, latency, and critical regressions must be agreed upon with the adopting institution before outputs are examined.

1.3 Significance of the Study

The argument follows three linked propositions. First, the CBL's reported transaction activity and digital-inclusion plans establish a concrete setting in which software quality matters to more users and merchants; however, these statistics do not measure cyber risk. Second, a security-review assistant is useful only when its findings are correctly located, its repairs withstand independent testing, and its false alarms and operating demands are manageable. Third, organizations value these outcomes differently; therefore, a transparent preference model is needed in addition to technical metrics. The planned synthetic experiment will test the second proposition, and AHP operationalizes the third. The first proposition provides policy relevance without turning public payment statistics into evidence about vulnerabilities.

The intended beneficiaries are application-security reviewers, software developers, and banking technology decision-makers who need an auditable protocol for comparing security-review assistants.

Copyright: ©2026 by the authors. This work is licensed under a (CC BY-4.0).

Scientific Journal of Faculty of Education, Misurata University, Libya
Volume (14), Issue (1)
ISSN: 2710-4141

https://doi.org/10.36602/sjfe.2026.v14.i01.06

Received: 06/10/2026 Accepted: 27/10/2026 Published: 29/10/2026

2 Literature Review and Related Studies

SecurityEval provides CWE-linked examples for assessing the security of generated code; its original paper reports 130 samples covering 75 weakness types (Siddiq & Santos, 2022). CyberSecEval examines insecure code generation and assistance with cyberattacks (Bhatt et al., 2023). CyberSecEval 2 extends assessment to prompt injection, code-interpreter abuse, false refusals, and exploit-generation tasks (Bhatt et al., 2024). BountyBench evaluates detection, exploitation, and patching in real-world codebases using verifiable task outcomes and bug-bounty context (Zhang et al., 2025). These studies motivate controlled security evaluation, but they address different tasks, datasets, and success definitions; their scores cannot serve as baseline results for the present application.

Table 2: Comparison with previous security evaluation studies

Study	Task and evidence	Relationship to this framework	Comparison limitation
SecurityEval Siddiq and Santos (2022)	Generated-code security; CWE-linked examples.	Informs weakness mapping and comparison with static analysis.	Code generation differs from reviewing and repairing an existing application.
CyberSecEval Bhatt et al. (2023)	Insecure code generation and cyberattack assistance.	Supports a controlled, repeatable evaluation pipeline.	Generation and safety outcomes do not measure this protocol's localized detection or repairs.
CyberSecEval 2 Bhatt et al. (2024)	Prompt injection, interpreter abuse, false refusal, and exploitation.	Shows the need for several security and utility outcomes.	Its tasks do not establish performance on the planned banking-information corpus.
BountyBench Zhang et al. (2025)	Detection, exploitation, and patching in real-world codebases.	Motivates localized adjudication and verifiable patch outcomes.	Its real-world task setting is more externally realistic than the synthetic design here.
Present Study	Planned paired synthetic cases; verified findings and repairs; AHP decision layer.	Links detection, repair, reliability, and cost to elicited priorities and sensitivity.	Only protocol and simulated AHP calculations are demonstrated; no measured benchmark superiority.

2.1 Security taxonomies and scoring

A detected defect is first matched to an affected code location and behavior, then to a specific CWE entry and an OWASP Top 10:2025 category. OWASP is an awareness taxonomy of broad risks, while CWE identifies underlying weakness types. CVSS v4.0 vectors are assigned by two trained assessors using the FIRST specification and the official calculator. The vector, score, and disagreement resolution are archived. An LLM-produced number without a valid vector is not accepted as a scored severity assessment.

2.2 Decision analysis

The Analytic Hierarchy Process decomposes a decision into criteria and pairwise comparisons (Saaty, 1980, 2008). Expert-derived priorities represent stakeholder preferences; measured model performance represents alternatives. Keeping these layers separate prevents

Copyright: ©2026 by the authors. This work is licensed under a (CC BY-4.0).

Scientific Journal of Faculty of Education, Misurata University, Libya
Volume (14), Issue (1)
ISSN: 2710-4141

https://doi.org/10.36602/sjfe.2026.v14.i01.06

Received: 06/10/2026 Accepted: 27/10/2026 Published: 29/10/2026

an arbitrary preference matrix from masquerading as a security experiment. A criterion whose score is unavailable cannot silently receive a favorable value; the decision is postponed or reported on a stated subset of criteria.

2.3 Research Gap and Hypotheses

The contribution of this study, relative to these studies, is the explicit connection between auditable task outcomes and a stakeholder-weighted selection decision in a constrained banking-information context. It is not a claim that detection, patch testing, or multi-metric evaluation is new. The prospective hypotheses are as follows: H1: precision and recall differ between systems on paired tasks; H2: structured review changes secure-patch success relative to code-only review under equal patch-task budgets; and H3: cost-sensitive AHP priorities can change rankings relative to F1 alone. These questions require observed outcomes and cannot be tested using the simulated preference matrices.

3 Methodology, Research Tools and Procedures

3.1 Study design and ethical boundary

Build a locally hosted, explicitly non-official banking-information application with exchange rates, announcements, reports, search, downloads, contact, a dummy user portal, and a simulated content-management function. Use invented records only. Local containers have outbound network access disabled during security scans and execute with restricted privileges. No real CBL credentials, data, source code, API, host, or endpoint is placed in the experimental environment. Public website content may establish high-level service categories but cannot establish internal architecture or vulnerability status.

Application to a local operational banking system was not feasible because obtaining the required security approvals was difficult. This access constraint determined the synthetic case design; it is not evidence that a particular institution refused a documented request. The application therefore models selected information-service functions rather than a core banking or payment-processing system. A field extension would require written authorization defining the permitted assets and tests, an institution-approved isolated environment, and agreed handling of any confidential material before data access or execution.

3.2 Ground truth construction

Create 12 isolated defect variants, each paired with a corrected variant and a passing functional test. Define one canonical vulnerability ID per distinct root cause and sink; multiple mentions of the same root cause count once. An independent assessor validates each seed by reading source and running local tests, then freezes file hashes before model assessment. The proposed twelve classes are SQL injection, reflected XSS, stored XSS, path traversal, object authorization failure, missing CSRF protection, weak password storage, hard-coded secret, missing rate control in the simulated login, unsafe debug configuration, insufficient audit logging, and insecure file upload. This selection is a planned design and not evidence that these defects have been implemented or verified.

Table 3: Candidate vulnerability injection plan

ID	Module	Weakness	CWE candidate
V01	Search	SQL injection	CWE-89
V02	News	Reflected XSS	CWE-79
V03	Contact	Stored XSS	CWE-79
V04	Downloads	Path traversal	CWE-22

Copyright: ©2026 by the authors. This work is licensed under a (CC BY-4.0).

Scientific Journal of Faculty of Education, Misurata University, Libya
Volume (14), Issue (1)
ISSN: 2710-4141

https://doi.org/10.36602/sjfe.2026.v14.i01.06

Received: 06/10/2026 Accepted: 27/10/2026 Published: 29/10/2026

ID	Module	Weakness	CWE candidate
V05	Portal	Object authorization	CWE-639
V06	Admin	CSRF	CWE-352
V07	Portal	Weak password hashing	CWE-916
V08	Configuration	Hardcoded secret	CWE-798
V09	Portal	Unrestricted login attempts	CWE-307
V10	Configuration	Debug exposure	CWE-489
V11	Admin	Missing security audit	CWE-778
V12	Uploads	Unrestricted file upload	CWE-434

Table 3 is a candidate injection plan. Final CWE and OWASP mappings require verification against the specific implemented behavior; no CVSS vector or score can be defensibly assigned before implementation and deployment assumptions are fixed.

3.3 Systems and scenarios

Candidate API systems are GPT-6 Astra (gpt-6-astra), GPT-6.1 Sol (gpt-6.1-sol), and GPT-6 Luna (gpt-6-luna), subject to access and API validation on the actual run date (OpenAI, n.d.). Log exact model identifier, version or snapshot if exposed, request parameters, reasoning configuration, prompts, timestamps, and usage. Compare Bandit and Semgrep under pinned rules and versions; any OWASP ZAP scan targets localhost only. No comparative result is inferred from vendor descriptions. Bandit and Semgrep provide detection baselines (Bandit Developers, n.d.; Semgrep, n.d.); ZAP provides a separately reported local dynamic-scan baseline (ZAP Development Team, n.d.). Compare tools only on outcomes they support, rather than assigning invented patch or explanation scores to unavailable capabilities.

Four scenarios are planned: S1 selected-file review, S2 repository review, S3 sandboxed inspect-and-patch review, and S4 a prompt-robustness version of S1. S1–S3 each use one fixed phrasing; S4 uses three prespecified phrasings. Each of 12 defect pairs contributes a vulnerable and a corrected code state, with five independent runs per state, scenario, and phrasing. The total is $12 \times 2 \times (1 + 1 + 1 + 3) \times 5 = 720$ scheduled task evaluations per LLM, or 2,160 across three LLMs. These are planned evaluations, not observed calls; agent tasks may require multiple API requests. Additional patch tasks, retries, and baseline scans are counted separately. Randomize execution order, fix the time and token budgets, as well as the tool permissions, and record all settings. Where sampling cannot be deterministic, retain the available API configuration and variability.

3.4 Prompt and response schema

A common instruction requests an array of findings containing repository-relative path, line span, vulnerable operation, evidence, CWE, OWASP category, CVSS v4.0 vector, confidence, impact, and a minimal repair. A separate patch task receives the same code context. Responses must cite concrete code evidence; unsupported allegations are adjudicated as false positives. The model may abstain. Reviewers blinded to model identity match outputs to the frozen ground truth, with disagreements resolved by a third reviewer and raw decisions retained.

3.5 Outcomes and adjudication

A true positive requires the correct defective operation and an overlapping location; a correct general class without localization is a classification observation but not a detection true positive. Precision = TP/(TP+FP), recall = TP/(TP+FN), and F1 = 2PR/(P+R); zero-denominator cases are reported as undefined with counts, never silently set to zero. Report per-variant macro

Copyright: ©2026 by the authors. This work is licensed under a (CC BY-4.0).

Scientific Journal of Faculty of Education, Misurata University, Libya
Volume (14), Issue (1)
ISSN: 2710-4141

https://doi.org/10.36602/sjfe.2026.v14.i01.06

Received: 06/10/2026 Accepted: 27/10/2026 Published: 29/10/2026

averages and pooled micro counts. CWE and OWASP accuracy are conditional on correctly matched findings, alongside unconditional end-to-end accuracy to avoid favorable selection. Severity error is defined as the mean absolute difference from adjudicated CVSS v4.0 scores; validity of vector syntax is separate. Patch success requires an attack regression to fail, existing functionality to pass, and independent code review. Timing is wall clock at the client; cost uses actual recorded tokens multiplied by the contemporaneous published API price and excludes unmeasured labor.

For O4, the secure-patch denominator includes all assigned patch attempts, including invalid and incomplete patches; functional regression is also reported separately. For O5, run agreement is the mean pairwise Jaccard overlap of detected canonical defect-ID sets over the five runs; report both-empty comparisons separately rather than assigning perfect agreement. Evidence quality uses three binary checks per finding: identifiable location, named defective operation, and a supported causal link to the claimed weakness; report the mean score divided by three. Optional calibration uses the Brier score against adjudicated finding correctness. Failure rates, refusals, and missing usage are reported alongside these measures so conditional averages do not conceal unsuccessful tasks.

3.6 Statistical analysis

Pair systems on the same frozen defect variant, code state, and task context. Report 95% bootstrap confidence intervals by resampling the 12 defect pairs as clusters, while keeping repeated runs and both code states within each cluster (Efron & Tibshirani, 1993). Use 10,000 replicates with an archived seed and interval method. For per-variant F1 across three systems within a prespecified scenario, use Friedman's test only with complete paired blocks. Where warranted, follow this with Wilcoxon signed-rank comparisons and Holm adjustment, checking distributional assumptions; otherwise use an exact paired procedure (Demšar, 2006). Designate one primary comparison and label other scenarios exploratory or include them in the prespecified multiplicity family. Report effect sizes and raw distributions. With only 12 independent defect pairs, power and bootstrap coverage can be limited; repeated runs do not increase the independent sample size.

4 AHP Decision Framework

The top level comprises detection quality D, remediation quality R, reliability and interpretability I, and operating efficiency E. D combines recall, precision, CWE mapping, and severity error; R combines secure-patch pass rate and regression-free rate; I combines repeated-run agreement, evidence quality, and abstention calibration; E combines latency and actual API cost. To avoid double counting, F1 is displayed as a descriptive outcome and excluded when precision and recall are already criteria. Each subcriterion is prespecified and scored on a bounded 0–1 scale. For a benefit measure (x), normalize to $(x-L)/(U-L)$; for a cost measure, use $(U-x)/(U-L)$, clipping only values beyond preregistered feasible bounds and reporting the clipping rate. Bounds come from requirements established before observing alternatives, not sample extrema.

Recruit at least five independent specialists across application security, software engineering, and banking technology governance. Each supplies pairwise judgments on Saaty's 1–9 scale, a disclosure of expertise, and justification for extreme choices. Preserve individual matrices. Reciprocal entry $a_{ij} = 1/a_{ji}$ and $a_{ii} = 1$. For group aggregation, take the geometric mean of judgments cell by cell and derive the normalized principal eigenvector w. Calculate $CI = (\lambda max - n)/(n-1)$ and $CR = CI/RI$; investigate CR above 0.10, recording any revision rather than

Copyright: ©2026 by the authors. This work is licensed under a (CC BY-4.0).

Scientific Journal of Faculty of Education, Misurata University, Libya
Volume (14), Issue (1)
ISSN: 2710-4141

https://doi.org/10.36602/sjfe.2026.v14.i01.06

Received: 06/10/2026 Accepted: 27/10/2026 Published: 29/10/2026

automatically discarding participants. The weighted score for alternative m is $S_m = \Sigma_j w_j z_{mj}$. No human expert responses have been collected; Appendix F provides a separate, explicitly simulated role-based illustration.

An exact-consistency teaching example uses w = (0.40, 0.30, 0.20, 0.10). $A_{ij} = w_i/w_j$ produces the following reciprocal matrix. It has $\lambda max = 4$, CI = 0, and CR = 0 by construction; these are algebraic properties, not expert estimates.

Table 4: Exact-consistency illustrative AHP matrix

Criterion	D	R	I	E
D	1	4/3	2	4
R	3/4	1	3/2	3
I	1/2	2/3	1	2
E	1/4	1/3	1/2	1

Sensitivity analysis varies each top-level weight from 0.5 to 1.5 times its elicited value while proportionally renormalizing the remaining weights. In addition, generate 1,000 weight vectors by independently drawing each multiplier from Uniform (0.5, 1.5), multiplying the group weights, and renormalizing, using archived seed 20261020. Recompute rankings of eligible systems and report each system's first-place share, ties under a prespecified tolerance, and the smallest tested perturbation that reverses the leading pair. These shares describe the disclosed sampling rule, not population probabilities. This is a planned procedure: illustrative weights alone cannot identify a winning model without a measured performance matrix.

5 Results of Framework Validation

5.1 Numerical results of the simulated AHP exercise

The five explicitly simulated role profiles yield a group priority vector of detection 0.3565, remediation 0.2920, reliability and interpretability 0.2187, and operational efficiency 0.1328. The priorities sum to 1.0000 within rounding. Detection exceeds remediation by 0.0645 and efficiency by 0.2236; these differences describe the constructed judgments, not estimated preferences in a population. Individual consistency ratios range from 0.0038 to 0.0225, and the group ratio is 0.0008, all below the prespecified 0.10 review threshold. The underlying matrices, prompts, rules, and charts appear in Appendix F. Low inconsistency follows partly from construction using coherent seed vectors and therefore cannot be used as evidence that five independent human specialists agree.

The roles are not identical: the DevSecOps profile assigns comparatively high priority to efficiency, whereas the application-security and independent-risk profiles emphasize detection. This deliberate spread shows why publishing only one aggregate weight vector would conceal differences in decision objectives. No alternative-level performance values were generated, so neither the group weights nor their differences produce a score or ranking for Astra, Sol, Luna, Bandit, Semgrep, or ZAP. Similarly, there is no measured precision, recall, patch success, latency, or cost result in this article.

The demonstrated outputs are the structural validity of the illustrative matrix and the separate simulated priorities in Appendix F. The exact matrix is positive and reciprocal; its priorities sum to one, and CR is zero by construction. No LLM calls, scans, expert surveys, CVSS adjudications, patches, or comparative statistical tests were performed. Consequently, Table 1 presents an evaluation plan rather than ~~a table of~~ achieved performance; observed rates, confidence intervals, costs, and model rankings are unavailable.

Copyright: ©2026 by the authors. This work is licensed under a (CC BY-4.0).

Scientific Journal of Faculty of Education, Misurata University, Libya
Volume (14), Issue (1)
ISSN: 2710-4141

https://doi.org/10.36602/sjfe.2026.v14.i01.06

Received: 06/10/2026 Accepted: 27/10/2026 Published: 29/10/2026

The protocol permits complete reporting after data acquisition: a flow count showing attempted and completed tasks; TP, FP, and FN by system and weakness class; precision, recall, F1, mapping accuracy, CVSS vector validity and error; patch and regression outcomes; latency, tokens, and cost; expert matrix consistency; final weighted scores and sensitivity. Raw records and adjudication decisions must accompany aggregate tables.

6 Discussion

6.1 Why the Libyan Payment Transition Changes the Research Priority

The public evidence establishes a direction and scale of use rather than a technical diagnosis. LYPay and OnePay activity recorded in a defined January 2025 interval, the reported first-half growth of electronic transactions and POS terminals, and the 2025–2029 inclusion strategy together indicate that digital interfaces are becoming a more consequential part of Libyan financial service delivery (Central Bank of Libya, 2025a, 2025b, 2025c). More participation may enlarge the number of users affected by a service failure or a defective transaction path. It does not imply that any live service has a particular weakness, nor that the CBL website processes the cited transactions. This distinction is essential to the validity of a banking-oriented case study.

The security question is therefore operational: which review assistant can reliably help developers prevent or repair faults before deployment under the resources available to them? A single recall measure answers only whether seeded defects were mentioned. Financial services also require control of false accusations, evidence that reviewers can trace back to code, repairs that preserve transaction and account invariants, and predictable effort. NIST SSDF supports embedding verification and remediation within the development process (Souppaya et al., 2022); the Basel Committee places cyber and technology disruptions within a wider resilience framework (Basel Committee on Banking Supervision, 2021). Neither source establishes the performance of the LLMs considered here, but both explain why a multi-criterion evaluation is a relevant research contribution.

The AHP layer has a precise, limited role. Detection, repair, consistency, and efficiency are measured on the same frozen tasks; AHP then makes a stakeholder's trade-offs explicit. It must not compensate for unsafe patches with a low price without showing that trade-off explicit in the component scores. A practical decision report should include minimum safety gates—such as no critical regression and a specified recall floor—before applying the weighted score for systems that pass. These thresholds are institutional choices to be registered before evaluation. Reporting both gate failures and weighted rankings avoids presenting a high aggregate score as proof of acceptable security.

The design separates evidence of a vulnerability from a plausible explanation. This matters because confident, incorrectly localized findings consume reviewer time and can distort apparent recall. Secure patch quality is assessed through both a targeted attack regression and preserved functionality; a patch that merely passes existing unit tests may leave the root cause intact. CVSS v4.0 assessment is a structured vectoring task rather than free-form prediction of a scalar. Expert weights encode decision priorities and are neither objectively "correct" nor transferable unchanged to another institution.

6.2 Interpretation of the simulated AHP result

The numerical exercise supports a narrow methodological claim: the proposed hierarchy can convert reciprocal role judgments into normalized, sufficiently consistent priorities and display disagreements across roles. The highest group weight on detection is coherent with the study's

Copyright: ©2026 by the authors. This work is licensed under a (CC BY-4.0).

Scientific Journal of Faculty of Education, Misurata University, Libya
Volume (14), Issue (1)
ISSN: 2710-4141

https://doi.org/10.36602/sjfe.2026.v14.i01.06

Received: 06/10/2026 Accepted: 27/10/2026 Published: 29/10/2026

concern about missed defects, but it is not a finding that actual banks prefer detection over all other objectives. The separate repair weight is substantial because discovering a flaw without an effective correction may leave risk unchanged. The reliability weight recognizes that inconsistent or weakly evidenced findings can consume scarce expert review capacity. The lower efficiency weight should not be interpreted as a license to ignore cost or latency; it reflects only the five scripted profiles and would change under different institutional priorities.

A consistency ratio indicates whether a respondent's pairwise statements are internally coherent. It does not establish that criteria are complete, that the chosen priorities are ethically or operationally appropriate, or that the evaluator agrees with any observed model outcome. In this simulation, CR values are especially small because the entries were derived from ordered seed intensities. A real panel could exhibit more conflict, and its subgroup differences would be substantively informative. The aggregation should therefore be accompanied by individual matrices and sensitivity analysis, as the figures do, rather than presented as an objective security score.

The decisive empirical gap is the alternative-performance matrix. Once an authorized local experiment supplies independently verified metrics, safety gates should be checked before weighted scoring. A system with a severe patch regression or an unacceptable missed-critical-defect rate should not be selected merely because it is inexpensive. Among systems that meet those constraints, AHP can make the remaining trade-offs explicit; rank reversals under plausible weights should be reported as decision uncertainty. These conclusions are conditional design implications, not observed differences among the named models.

6.3 Comparison with previous studies

The framework agrees with SecurityEval and CyberSecEval that security needs explicit evaluation beyond functional code correctness (Siddiq & Santos, 2022; Bhatt et al., 2023). CyberSecEval 2 broadens the range of security and utility outcomes; this protocol instead focuses on code review, verified remediation, and resources within a fixed application (Bhatt et al., 2024). It shares BountyBench's emphasis on verifiable detection and patching, while BountyBench has the stronger realism of real-world codebases (Zhang et al., 2025). The proposed addition is a transparent AHP layer that separates measured outcomes from stakeholder preferences and examines ranking sensitivity. This is a comparison of evaluation designs, not an empirical comparison of scores. The simulated CR values cannot establish improved detection or repair, and the absence of local banking authorization further limits external validity.

7 Threats to Validity

Construct validity is threatened by matching rules, ambiguous defect boundaries, and subjective evidence-quality ratings. Use a frozen codebook and blinded dual adjudication. Internal validity is affected by prompt wording, API updates, time-dependent model behavior, nondeterministic output, tool budgets, and reviewer expectations; pin settings and preserve complete transcripts. External validity is limited by the small synthetic application, artificial seeds, the Flask or FastAPI stack, and lack of production data. Conclusion validity is limited by only 12 planned variants, dependence among repeated runs, multiplicity, and potentially unstable AHP judgments. These limitations preclude inference about CBL production security or financial software in general. The difficulty of obtaining security approvals prevented an operational local banking case. Although synthetic controls improve auditability, they omit

Copyright: ©2026 by the authors. This work is licensed under a (CC BY-4.0).

Scientific Journal of Faculty of Education, Misurata University, Libya
Volume (14), Issue (1)
ISSN: 2710-4141

https://doi.org/10.36602/sjfe.2026.v14.i01.06

Received: 06/10/2026 Accepted: 27/10/2026 Published: 29/10/2026

production transaction workflows, legacy integration, real user behavior, and institutional controls; they cannot establish effectiveness in a local bank.

The CBL figures are secondary contextual data with reporting periods and definitions set by the source. They cannot be used as exposure denominators for the synthetic benchmark. Transferring the proposed findings to live Libyan payment software would require a separate authorized evaluation, service-specific threat model, independent ground truth, and assessment of operational and third-party dependencies. Published policy announcements should not be read as evidence of implementation completion.

8 Ethics and Reproducibility

No unauthorized penetration testing was conducted. No real customer data were collected, no accounts were accessed, and no real vulnerabilities were exploited. The CBL public site serves solely as contextual inspiration; neither its internal implementation nor its security posture is inferred. The synthetic application must state that it is not affiliated with or endorsed by the Central Bank of Libya. Publication of deliberately vulnerable code should include local-only default binding, fictitious data, clear labeling, and safe deployment instructions. The security-approval constraint explains the absence of field application; any later banking evaluation must obtain written institutional authorization before access, testing, or use of confidential records.

The reproducibility archive should contain /digital_twin, /ground_truth, /prompts, /model_outputs, /security_tools, /tests, /patches, /ahp, /statistics, /raw_results, /figures, and /documentation, with dependency locks, hashes, container definitions, execution dates, redacted API configuration, adjudication records, and analysis scripts. This manuscript specifies the archive but does not claim that the implementation or archive has already been built.

9 Conclusion

The expansion of electronic payments in Libya makes secure software review a consequential research topic. Published Central Bank of Libya indicators show active instant-payment services and growth in electronic transactions and point-of-sale infrastructure, while the national financial-inclusion strategy signals continued digital development (Central Bank of Libya, 2025a, 2025b, 2025c). These facts motivate scrutiny of software quality; they do not imply any vulnerability in the CBL website, LYPay, OnePay, or another production service.

The paper specifies an authorized synthetic benchmark and an AHP framework covering detection, classification, CVSS v4.0 assessment, secure remediation, consistency, false-positive control, latency, and cost. In the explicitly simulated expert-role exercise, group priorities were 0.3565 for detection, 0.2920 for remediation, 0.2187 for reliability, and 0.1328 for efficiency; the group consistency ratio was 0.0008. These are reproducible calculations on constructed judgments, not elicited human preferences. They illustrate the mechanics of a transparent decision rule but cannot establish a preferred LLM.

The study supports a reproducible evaluation process rather than a model ranking. Six objectives have explicit indicators, planned verification targets, and evidence sources. A comparative conclusion requires actual model and tool runs, independently verified findings and repairs, measured resource use, and judgments from human stakeholders. Comparison with previous benchmarks establishes the methodological position of the AHP layer without demonstrating superior performance. Difficulty obtaining the required security approvals prevented application to a local operational banking system. A subsequent authorized

Copyright: ©2026 by the authors. This work is licensed under a (CC BY-4.0).

Scientific Journal of Faculty of Education, Misurata University, Libya
Volume (14), Issue (1)
ISSN: 2710-4141

https://doi.org/10.36602/sjfe.2026.v14.i01.06

Received: 06/10/2026 Accepted: 27/10/2026 Published: 29/10/2026

institutional pilot is therefore needed to test transferability beyond the synthetic information-service setting.

Recommendations

- Freeze the local application and the independently verified ground truth before comparative evaluation.
- Validate findings and repairs through blinded adjudication, functional tests, and security regression tests.
- Collect actual model outputs, resource usage, and judgments from qualified human stakeholders before ranking systems.
- Apply minimum security gates and report sensitivity to stakeholder priorities before operational adoption.
- Obtain written institutional security approval and an agreed-upon isolated testing scope before extending the evaluation to a local banking system.

Declarations

Data availability: The manuscript contains the simulated AHP matrices and their calculated outputs. No empirical model-run dataset is reported. A follow-up release should include implementation code, frozen test artifacts, raw outputs, adjudication records, and analysis scripts in a versioned archive.

Funding: This research received no external funding.

Author contributions and approval: All authors contributed to the development, review, and preparation of the manuscript and approved its final version for submission. The authors agreed on their respective contributions in accordance with the CRediT (Contributor Roles Taxonomy) framework and consented to authorship.

Appendix A

Data dictionary and audit trail

Every run receives a unique run_id, immutable source commit hash, defect_variant_id, scenario_id, prompt_id, system_id, exact model identifier, UTC request and response timestamps, request parameters, input and output token counts, response status, tool transcript hash, and raw response path. Every finding receives a unique finding_id linked to run_id, code span, claimed weakness, evidence quotation, CVSS vector, proposed patch identifier, and adjudication record. The audit table stores reviewer identifiers, independent decisions, resolution reason, and date. Keep diagnostic logs and the original model text even when parsing fails. Remove secrets from archived client configuration and record any redaction in a manifest. Dataset versions are frozen before scoring.

The unit of a defect is a specific root cause in a concrete code state, not a general weakness category. A defect may occur in one or more nearby lines; matching tolerances must be fixed by the two assessors before model outputs are opened. When two model findings point to the same root cause, the first eligible finding can be a true positive, and duplicates are tracked separately. Reports outside the seeded ground truth are not automatically false positives: assessors first determine whether they expose a newly verified defect in the synthetic application, revise the dataset under a logged protocol deviation, and rerun all systems on the amended denominator if necessary.

Appendix B Expert instrument and aggregation

The survey preface defines the deployment decision as choosing a secondary reviewer for a small public banking-information application. Experts are asked to compare detection quality

Copyright: ©2026 by the authors. This work is licensed under a (CC BY-4.0).

Scientific Journal of Faculty of Education, Misurata University, Libya
Volume (14), Issue (1)
ISSN: 2710-4141

https://doi.org/10.36602/sjfe.2026.v14.i01.06

Received: 06/10/2026 Accepted: 27/10/2026 Published: 29/10/2026

against remediation quality, detection against reliability, detection against efficiency, remediation against reliability, remediation against efficiency, and reliability against efficiency. For each pair, the respondent selects which criterion matters more and one of Saaty's intensities 1, 3, 5, 7, or 9, with 2, 4, 6, or 8 allowed as intermediate choices. The form provides plain descriptions, asks for role and years of experience, and includes free text explaining strong judgments. The same process is repeated within each main criterion for subcriteria.

Pairwise judgments are elicited without showing the names of the evaluated models or their scores. Individual consistency is reviewed, and a participant may clarify a mistaken entry once; both versions are retained. Aggregate judgments using the cellwise geometric mean. Because the resulting group matrix remains reciprocal, compute the dominant eigenvector and normalize it. The 4×4 random index used for the top level is 0.90 in the conventional Saaty table; the threshold is $CR \leq 0.10$. Do not claim that a mathematically consistent matrix establishes consensus. Report between-expert dispersion, criteria definitions, participation rate, and a sensitivity analysis retaining each expert's own weights.

Appendix C

Predeclared Evaluation Decisions

Before opening model output, register the security test oracle for each variant: a test that demonstrates the vulnerable behavior in the seeded version, a security regression that passes in the fixed version, and a functionality invariant that must remain true. Reviewers run patch tests in a disposable container with a strict execution deadline and no external network. A syntactically invalid patch is counted as a failed patch attempt. A patch that suppresses the exposed endpoint while breaking a legitimate function is a regression. A patch that handles only the supplied demonstration payload but leaves the root cause is unsuccessful after independent review.

A failure to return within the time budget is a task failure and is included in the intention-to-evaluate denominator. API transport failures are distinguished from model refusals and task failures; a predefined retry rule handles transient transport errors. Costs are measured for all attempted calls, including unsuccessful ones. No post hoc prompt repair is allowed in the principal comparison. Additional exploratory prompts are separately labeled. Report all excluded runs and exact reasons.

Appendix D

Reporting and Release Checklist

The empirical follow-up should provide a study flow diagram; version and access logs for each model and baseline tool; full prompts and model parameters; ground-truth codebook; independently reviewed CWE and OWASP mappings; CVSS v4.0 vectors and scoring rationale; confusion counts by defect; patch test outcomes and reviewer decisions; distribution of latency, token usage and cost; individual anonymized AHP matrices and group weights; sensitivity settings and random seeds; statistical scripts; and a machine-readable manifest of file hashes. Every plotted aggregate must be reproducible from the same raw records that populate the tables.

The manuscript must be revised from a methodological article to an empirical article only after those records exist. Results should state measured values with denominators and uncertainty, Discussion should interpret observed failure modes, and the abstract should report measured findings without extrapolation to the live CBL site. The author declarations, journal

Copyright: ©2026 by the authors. This work is licensed under a (CC BY-4.0).

Scientific Journal of Faculty of Education, Misurata University, Libya
Volume (14), Issue (1)
ISSN: 2710-4141

https://doi.org/10.36602/sjfe.2026.v14.i01.06

Received: 06/10/2026 Accepted: 27/10/2026 Published: 29/10/2026

choice, originality status, and Turnitin similarity check require the authors' own verification before sending to the journal.

Appendix E

Interpretation Guide for an Empirical Follow-up

The primary comparison is a paired performance comparison across the same defect variants. Report each model's observed counts with its denominator, including abstentions and incomplete calls. A high precision score on only a few attempted findings may coexist with low recall; both values and the number of attempts are required. Conversely, broad detection with many false alarms may impose more triage work than a small gain in recall justifies. Define reviewer minutes per confirmed finding as an optional operational measure only if actual human timing is collected. Do not impute time from token usage.

AHP results should be interpreted alongside the unweighted measurement table. The weighted score compresses several objectives into one number under a particular elicited preference profile; it cannot certify that a model is safe for autonomous security decisions. Present the full range of individual expert rankings, the group ranking, and the sensitivity threshold for reversal. If a leading model changes when a modest weight shifts, describe the alternatives as close and preference-dependent. If one model dominates every measured criterion, a weighted ranking adds little information but still records stakeholder priorities.

For CVSS v4.0, report vector validity, exact agreement on each Base metric, score absolute error, and disagreements attributable to missing deployment context. A scalar difference alone can conceal materially different assumptions about attack prerequisites or impacts. Classification reporting should similarly distinguish the broad OWASP category from a specific CWE. Findings in classes that are difficult to expose through source inspection, such as logging gaps or insecure design, should be discussed as separate strata rather than forced into the same interpretation as a syntactically obvious injection sink.

The conclusion of an empirical extension should state the context in which its results hold: a frozen synthetic banking-information application, selected defects, defined model snapshots, specified prompts, time and tool budgets, and one expert panel. Any larger claim requires replication on additional independently developed applications and defect distributions. At no stage should the description imply that the Central Bank of Libya endorsed the work or that its live infrastructure shares the synthetic defects.

Appendix F

Simulated Expert Judgments and AHP Analysis

This section is a transparent role-play simulation generated for methodological illustration. S1–S5 are simulated personas, not recruited human experts, and their judgments do not establish stakeholder preferences. No independent LLM expert was queried; the following role prompts and deterministic response rules produce the complete synthetic dataset. The analysis demonstrates AHP arithmetic and charts only. It does not rank Astra, Sol, or Luna because no performance matrix has been measured.

F.1 Expert role prompts and response rule

Shared prompt supplied to each simulated persona: "Act as the stated professional role selecting a secondary security-review assistant for a synthetic banking-information web application. Compare four criteria: D detection quality, R secure remediation, I consistency and

Copyright: ©2026 by the authors. This work is licensed under a (CC BY-4.0).

Scientific Journal of Faculty of Education, Misurata University, Libya
Volume (14), Issue (1)
ISSN: 2710-4141

https://doi.org/10.36602/sjfe.2026.v14.i01.06

Received: 06/10/2026 Accepted: 27/10/2026 Published: 29/10/2026

evidence quality, and E latency and cost. Give six pairwise judgments using the 1–9 AHP scale, explain each priority briefly, and do not claim any real survey participation or model-performance evidence." The role-specific sentence for each S1–S5 is reported in Table 5. For reproducibility, this illustrative run encodes the profile intensities as the integer seed vector shown there, converts each ratio to the closest Saaty integer on a logarithmic distance scale, and sets the reciprocal entry exactly. This is a deterministic simulation of potential responses, not measured output of five independent model calls.

Table 5: Simulated expert roles and seed profiles

ID and simulated role	Prompt-specific priority statement	Seed D:R:I:E
S1 Application security reviewer	Detection and verified repairs matter most because missed weaknesses remain exposed.	5:4:2:1
S2 Secure software engineer	A change must repair the root cause while preserving legitimate behavior.	4:5:3:1
S3 Banking technology governance	Traceability and stable evidence matter alongside defect discovery.	5:4:3:2
S4 DevSecOps lead	At scale, time and cost affect whether secondary review is deployed.	4:3:3:5
S5 Independent risk assessor	High-impact omissions dominate; inconsistent findings also burden triage.	6:4:4:2

F.2 Pairwise matrices and consistency

The column order in every matrix is D, R, I, E. Entries are the synthetic pairwise judgments after discretization. Each matrix is reciprocal, with ones on its diagonal. The group matrix is the cellwise geometric mean of five matrices. Principal eigenvectors are normalized to sum to one; λmax, CI = (λmax−4)/3, and CR = CI/0.90 are calculated from unrounded matrices.

Table 6: Simulated persona S1 pairwise matrix (CR = 0.0057)

	D	R	I	E
D	1.000	1.000	3.000	5.000
R	1.000	1.000	2.000	4.000
I	0.333	0.500	1.000	2.000
E	0.200	0.250	0.500	1.000

Table 7: Simulated persona S2 pairwise matrix (CR = 0.0151)

	D	R	I	E
D	1.000	1.000	1.000	4.000
R	1.000	1.000	2.000	5.000
I	1.000	0.500	1.000	3.000
E	0.250	0.200	0.333	1.000

Table 8: Simulated persona S3 pairwise matrix (CR = 0.0170)

	D	R	I	E
D	1.000	1.000	2.000	3.000
R	1.000	1.000	1.000	2.000
I	0.500	1.000	1.000	2.000
E	0.333	0.500	0.500	1.000

Copyright: ©2026 by the authors. This work is licensed under a (CC BY-4.0).

Scientific Journal of Faculty of Education, Misurata University, Libya
Volume (14), Issue (1)
ISSN: 2710-4141

https://doi.org/10.36602/sjfe.2026.v14.i01.06

Received: 06/10/2026 Accepted: 27/10/2026 Published: 29/10/2026

Table 9: Simulated persona S4 pairwise matrix (CR = 0.0225)

	D	R	I	E
D	1.000	1.000	1.000	1.000
R	1.000	1.000	1.000	0.500
I	1.000	1.000	1.000	0.500
E	1.000	2.000	2.000	1.000

Table 10: Simulated persona S5 pairwise matrix (CR = 0.0038)

	D	R	I	E
D	1.000	2.000	2.000	3.000
R	0.500	1.000	1.000	2.000
I	0.500	1.000	1.000	2.000
E	0.333	0.500	0.500	1.000

Table 11: Group geometric-mean pairwise matrix (λmax = 4.0023, CR = 0.0008)

	D	R	I	E
D	1.000	1.149	1.644	2.825
R	0.871	1.000	1.320	2.091
I	0.608	0.758	1.000	1.644
E	0.354	0.478	0.608	1.000

F.3 Calculated priorities

Table 12: Simulated individual and group priorities and consistency ratios

Simulated judgment set	D	R	I	E	CR
S1	0.410	0.350	0.158	0.082	0.0057
S2	0.304	0.382	0.238	0.076	0.0151
S3	0.366	0.278	0.233	0.124	0.0170
S4	0.246	0.204	0.204	0.347	0.0225
S5	0.424	0.227	0.227	0.122	0.0038
Group	0.356	0.292	0.219	0.133	0.0008

The group weights above are computed from aggregated judgments rather than obtained by averaging the rounded individual priorities. All consistency ratios remain below the prespecified 0.10 review threshold. This reflects the use of deterministic seed construction and should not be taken as evidence of real-world expert agreement. The figures visualize only synthetic AHP weights and consistency; there is no model performance or a final AHP score to chart.

Copyright: ©2026 by the authors. This work is licensed under a (CC BY-4.0).

Scientific Journal of Faculty of Education, Misurata University, Libya
Volume (14), Issue (1)
ISSN: 2710-4141

https://doi.org/10.36602/sjfe.2026.v14.i01.06

Received: 06/10/2026 Accepted: 27/10/2026 Published: 29/10/2026

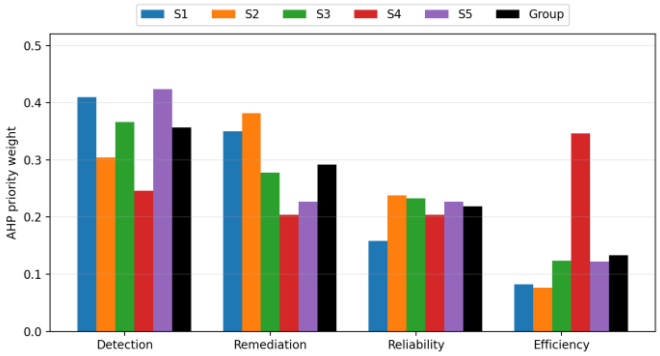


Figure 1. Simulated individual and group AHP weights.

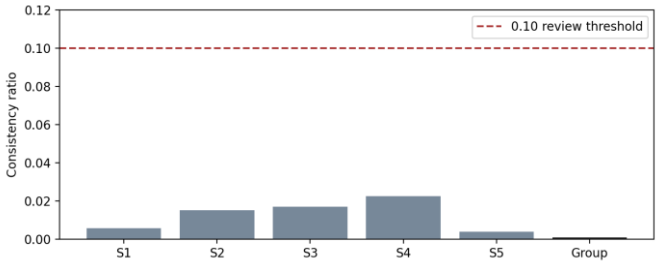


Figure 2. Consistency ratios of simulated pairwise judgments.

F.4 Conclusions

The simulated group gives D the highest priority (0.356) and E the lowest priority (0.133). The DevSecOps role raises efficiency relative to the other personas by design. These numbers illustrate how different stakeholder priorities affect a decision rule, but they cannot be interpreted as views of actual Libyan banking specialists. A ranking would require a measured 0–1 performance vector for each evaluated system and multiplication by these weights. Using synthetic weights for a real deployment would require explicit stakeholder acceptance and sensitivity analysis.

Conflict of Interest

The authors declare there is no conflict of interest associated with this study.

References

Bandit Developers. (n.d.). *Bandit documentation*. Retrieved October 20, 2026, from https://bandit.readthedocs.io/en/latest/

Basel Committee on Banking Supervision. (2021). *Principles for operational resilience*. Bank for International Settlements. https://www.bis.org/bcbs/publ/d516.pdf

Bhatt, M., Chennabasappa, S., Li, Y., Nikolaidis, C., Song, D., Wan, S., Ahmad, F., Aschermann, C., Chen, Y., Kapil, D., Molnar, D., Whitman, S., & Saxe, J. (2024). *CyberSecEval 2: A wide-ranging cybersecurity evaluation suite for large language models*. arXiv. https://doi.org/10.48550/arXiv.2404.13161

Copyright: ©2026 by the authors. This work is licensed under a (CC BY-4.0).

Scientific Journal of Faculty of Education, Misurata University, Libya
Volume (14), Issue (1)
ISSN: 2710-4141

https://doi.org/10.36602/sjfe.2026.v14.i01.06

Received: 06/10/2026 Accepted: 27/10/2026 Published: 29/10/2026

Bhatt, M., Chennabasappa, S., Nikolaidis, C., Wan, S., Evtimov, I., Gabi, D., Song, D., Ahmad, F., Aschermann, C., Fontana, L., Frolov, S., Giri, R. P., Kapil, D., Kozyrakis, Y., LeBlanc, D., Milazzo, J., Straumann, A., Synnaeve, G., Vontimitta, V., . . . Saxe, J. (2023). *Purple Llama CyberSecEval: A secure coding benchmark for language models*. arXiv. https://doi.org/10.48550/arXiv.2312.04724

Central Bank of Libya. (2025a, January 23). *Instant payment services statistics*. https://cbl.gov.ly/en/instant-payment-services-statistics-2/

Central Bank of Libya. (2025b, June 15). *Governor of the Central Bank of Libya announces from Tripoli the launch of the National Financial Inclusion Strategy and a number of transformative projects during the opening of the Electronic Payment Forum and Exhibition*. https://cbl.gov.ly/en/governor-of-the-central-bank-of-libya-announces-from-tripoli-the-launch-of-the-national-financial-inclusion-strategy-and-a-number-of-transformative-projects-during-the-opening-of-the-electronic-paymen/

Central Bank of Libya. (2025c, July 15). *The Governor of the Central Bank of Libya discusses performance indicators with banks and electronic payment companies, and directs the expansion of instant payment services as part of the digital transformation plan*. https://cbl.gov.ly/en/the-governor-of-the-central-bank-of-libya-discusses-performance-indicators-with-banks-and-electronic-payment-companies-and-directs-the-expansion-of-instant-payment-services-as-part-of-the-digital-tra/

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research, 7*, 1–30. https://jmlr.org/papers/v7/demsar06a.html

Efron, B., & Tibshirani, R. J. (1993). *An introduction to the bootstrap*. Chapman & Hall. https://doi.org/10.1201/9780429246593

FIRST. (2023). *Common Vulnerability Scoring System version 4.0: Specification document*. https://www.first.org/cvss/v4.0/specification-document

MITRE. (2026). *Common Weakness Enumeration (Version 4.20)*. https://cwe.mitre.org/data/index.html

OpenAI. (n.d.). *Models*. Retrieved October 20, 2026, from https://developers.openai.com/api/docs/models

OWASP Foundation. (2025). *OWASP Top 10:2025*. https://top10.owasp.org/2025/

Saaty, T. L. (1980). *The analytic hierarchy process: Planning, priority setting, resource allocation*. McGraw-Hill International Book Company.

Saaty, T. L. (2008). Decision making with the analytic hierarchy process. *International Journal of Services Sciences, 1*(1), 83–98. https://doi.org/10.1504/IJSSCI.2008.017590

Semgrep. (n.d.). *Semgrep docs*. Retrieved October 20, 2026, from https://docs.semgrep.dev/

Siddiq, M. L., & Santos, J. C. S. (2022). SecurityEval dataset: Mining vulnerability examples to evaluate machine learning-based code generation techniques. In *Proceedings of the 1st International Workshop on Mining Software Repositories Applications for Privacy and Security* (pp. 29–33). Association for Computing Machinery. https://doi.org/10.1145/3549035.3561184

Souppaya, M., Scarfone, K., & Dodson, D. (2022). *Secure software development framework (SSDF) version 1.1: Recommendations for mitigating the risk of software vulnerabilities (NIST SP 800-218)*. National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-218

Copyright: ©2026 by the authors. This work is licensed under a (CC BY-4.0).

Scientific Journal of Faculty of Education, Misurata University, Libya
Volume (14), Issue (1)
ISSN: 2710-4141

https://doi.org/10.36602/sjfe.2026.v14.i01.06

Received: 06/10/2026 Accepted: 27/10/2026 Published: 29/10/2026

ZAP Development Team. (n.d.). *ZAP documentation*. Retrieved October 20, 2026, from https://www.zaproxy.org/docs/

Zhang, A. K., Ji, J., Menders, C., Dulepet, R., Qin, T., Wang, R. Y., Wu, J., Liao, K., Li, J., Hu, J., Hong, S., Demilew, N., Murgai, S., Tran, J., Kacheria, N., Ho, E., Liu, D., McLane, L., Bruvik, O., . . . Liang, P. (2025). *BountyBench: Dollar impact of AI agent attackers and defenders on real-world cybersecurity systems*. arXiv. https://doi.org/10.48550/arXiv.2505.15216

Copyright: ©2026 by the authors. This work is licensed under a (CC BY-4.0).