

Original Article

Benchmarking Generative AI Models for Cybersecurity Vulnerability Identification Using 50 Curated Vulnerability Cases: An Exploratory Single-Pass Comparative Study of ChatGPT, Gemini, and Claude

Emanuel Dungon ^{1,*}, Mark Jan Irog ¹, Hans Luther Salva ¹,
Julse Lorenz Alvior ¹, Rosa Mae Bascar ¹, Rod Albert Aspera ¹,
Serafin Palmares ¹, Kristine Soberano ¹

Received: 07 August 2026; Revised: 21 September 2026;
Accepted: 29 September 2026; Published: 05 October 2026

DOI: <https://doi.org/10.66074/R9L7P4J8W>

Abstract

LLMs are being rapidly examined for cybersecurity applications, including identifying software vulnerabilities, conducting secure code reviews, and performing security analysis. Despite recent developments in Generative Artificial Intelligence (GenAI), there is a lack of empirical evidence on the relative performance of state-of-the-art commercial LLMs on discovering and categorizing software vulnerabilities. In this paper, we benchmarked the cybersecurity vulnerability-identification capabilities of ChatGPT, Gemini, and Claude on a dataset of fifty curated vulnerability cases from various Common Weakness Enumeration (CWE) categories. All the models were given the same prompts in a comparative benchmarking setup in a standardized testing environment. Responses were scored using a CWE-based scoring structure that includes exact matches, related matches based on established MITRE CWE links, and incorrect classifications. Descriptive statistics and the Friedman test were used to analyze performance. Descriptive analyses showed that ChatGPT obtained the highest weighted score (72/100), followed by Claude (66/100) and Gemini (64/100). ChatGPT also achieved the highest exact classification rate and the shortest observed response time under the experimental conditions. However, the inferential analysis indicated no statistically significant differences among the evaluated models

¹ State University of
Northern Negros
Sagay City, Negros
Occidental, Philippines
*emanueldungon23@
gmail.com

(Friedman $\chi^2(2) = 3.170$, $p = .205$). These findings suggest broadly comparable vulnerability identification across the three LLMs under the exploratory, single-iteration benchmark conditions employed in the study. All evaluated models demonstrated meaningful capability for cybersecurity vulnerability identification. Although ChatGPT obtained the highest descriptive benchmark scores, inferential statistical analysis indicated no statistically significant performance differences among the evaluated models.

Keywords: artificial intelligence, benchmarking, cybersecurity, large language models, vulnerability detection

1. Introduction

Detecting cybersecurity vulnerabilities is an important aspect of information security, since vulnerabilities in software systems can be exploited to undermine the confidentiality, integrity, and availability of information assets. Today's software systems include large amounts of code, numerous services and configurations tied together in ways that make security vulnerabilities far more likely. The Open Worldwide Application Security Project (OWASP) defines the most severe online application security risks affecting enterprises globally as broken access control, security misconfiguration, cryptography failure, injection vulnerabilities, and authentication failure. (OWASP, 2025) Therefore, detecting vulnerabilities is imperative for countering cyber-attacks and mitigating security threats.

Recent studies have also identified important limitations of Large Language Models in software-security tasks, including hallucinated security findings, inconsistent vulnerability classifications, susceptibility to prompt variations, limited reasoning transparency, and challenges in distinguishing among closely related weakness categories (Ji et al., 2023; Schulhoff et al., 2024). These limitations highlight the need for rigorous benchmarking before operational deployment in cybersecurity environments.

In general, the traditional approaches for vulnerability identification are manual code review, Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST), and automated vulnerability scanning. These methodologies have played a vital role in software security assurance by discovering known vulnerabilities and security bugs. But existing solutions generally grapple with issues such as scalability, detection accuracy, contextual awareness, and the interpretation of sophisticated program logic. Static analysis techniques can yield many false positives while dynamic testing approaches require working execution environments and may

miss vulnerabilities that are not triggered during testing (Goseva-Popstojanova & Perhinschi, 2015). These constraints have provided an incentive for academics to explore alternative approaches to improving vulnerability assessments and reducing human effort.

Despite growing interest in applying LLMs to cybersecurity, comprehensive comparative evaluations remain limited. Recent systematic reviews have highlighted the need for rigorous benchmarking studies that assess vulnerability detection, secure code analysis, and cybersecurity decision-support capabilities across multiple models (Xu et al., 2024; Zhang et al., 2025).

Despite their potential, rigorous evaluation remains necessary before LLMs can be relied upon in cybersecurity decision-making processes. Security-critical environments require accurate and consistent vulnerability identification, making empirical benchmarking essential for understanding model strengths, limitations, and practical applicability (Xu et al., 2024; Zhang et al., 2025). Careful assessment enables researchers and practitioners to identify both the benefits and limitations of LLMs and to determine appropriate roles for these systems within cybersecurity workflows.

To fill these gaps, this study benchmarks the cybersecurity vulnerability identification capabilities of ChatGPT, Gemini, and Claude on a dataset of 50 standardized vulnerability cases across a range of Common Weakness Enumeration (CWE) categories and known software security flaws. All models were tested under controlled testing conditions to ensure consistency and comparability. All models were given the same instructions. We computed the Weighted Performance Score, vulnerability recognition rate, Exact Classification Rate, and response efficiency using a CWE-based scoring framework in which responses were classified as a perfect match, a related classification, or an incorrect classification.

In contrast to earlier work that has examined single LLMs or generic software engineering tasks, this paper proposes a unified cybersecurity benchmark that includes standardized prompts, a curated vulnerability dataset, the MITRE CWE taxonomy, and inferential statistical analysis.

This paper empirically compares the capabilities of modern Generative AI models for identifying and classifying software vulnerabilities, providing evidence of their relative strengths and limits in cybersecurity applications. Moreover, the study contributes to the expanding literature on AI-assisted vulnerability analysis by examining the trade-offs between vulnerability-identification performance, exact classification capability, vulnerability recognition, and response characteristics.

The study aimed to determine how effectively ChatGPT, Gemini, and Claude can identify cybersecurity vulnerabilities from standardized vulnerability cases. It also evaluated the vulnerability classification performance of each generative AI model in terms of exact CWE classifications, related CWE classifications, misclassifications, and weighted vulnerability identification performance. In addition, the study compared the three models based on exact classification rate, vulnerability recognition rate, and response-time efficiency. Finally, it examined which model provided the

most effective balance among vulnerability-identification performance, exact classification rate, vulnerability recognition rate, and response latency.

2. Methodology

2.1 Research Design

This study employed an experimental comparative benchmarking design to assess the cybersecurity vulnerability discovery capabilities of 3 major Generative Artificial Intelligence (GenAI) models: ChatGPT, Gemini, and Claude. Experimental comparative research is suitable for evaluating the performance of different intelligent systems under controlled, homogeneous testing conditions and for objective comparison against given evaluation criteria (Creswell & Creswell, 2023).

The independent variable in this study was the Generative AI model used to identify vulnerabilities. The dependent variables included the exact classification rate, related classification rate, weighted performance score, vulnerability recognition rate, and response time.

To ensure fairness, consistency, and reproducibility, all models were evaluated using the same benchmark dataset, standardized prompts, testing procedures, and scoring framework. This controlled experimental design minimized methodological variation and enabled direct comparison of model performance. The study was designed as a benchmarking investigation to compare observed performance differences under standardized conditions, rather than to generalize the findings to a broader population.

2.2 Benchmark Dataset Development

In this study, we generated a benchmark dataset of 50 curated cybersecurity vulnerability cases. Each benchmark represents a unique software security issue and includes vulnerable source code derived from publicly available educational and research sites.

The benchmark cases were classified according to the Common Weakness Enumeration (CWE) framework, a set of common identifiers for software security problems developed by MITRE. Benchmark instances were additionally mapped to the OWASP Top 10 Web Application Security Risks, where relevant, to ensure representation of commonly exploited web application vulnerabilities (MITRE, 2024; OWASP, 2025)

The benchmark dataset was compiled from multiple publicly available cybersecurity resources, including:

- MITRE Common Weakness Enumeration (CWE) Database

- OWASP Vulnerability Documentation
- Juliet Test Suite
- OWASP WebGoat
- Damn Vulnerable Web Application (DVWA)
- Secure coding repositories
- Academic cybersecurity datasets
- Vulnerable open-source software projects

To mitigate potential dataset selection bias and enhance the diversity of the benchmarks, we collected vulnerability instances from several internationally recognized cybersecurity resources. These include the MITRE Common Weakness Enumeration (CWE) database, OWASP vulnerability documentation, the Juliet Test Suite, WebGoat, Damn Vulnerable Web Application (DVWA), academic cybersecurity datasets, secure coding repositories, and vulnerable open-source software projects. The use of numerous independent sources was intended to obtain a broad representation of software security flaws and to reduce reliance on any single benchmark repository.

The benchmark dataset was designed for controlled, comparative evaluation of model behavior under standardized conditions, rather than for statistical generalization to all software-security vulnerabilities. Although the dataset includes diverse categories and vulnerability sources, larger benchmark collections may provide broader coverage of real-world security weaknesses.

2.3 Benchmark Dataset Composition

The final benchmark dataset comprised 50 vulnerability cases representing a diverse range of software security weaknesses. While most benchmark cases originated from distinct vulnerability instances, a small number were derived from the same source-code sample but evaluated separately because they represented different CWE categories.

The dataset included vulnerabilities such as:

- SQL Injection (CWE-89)
- Cross-Site Scripting (CWE-79)
- Hardcoded Credentials (CWE-798)
- Authentication Vulnerabilities (CWE-287, CWE-306)
- Authorization Vulnerabilities (CWE-639, CWE-862)
- Client-Side Security Decision Weaknesses (CWE-565, CWE-807)
- Buffer Overflow (CWE-120, CWE-121)

- Integer Overflow (CWE-190)
- Cryptographic Weaknesses (CWE-327)
- Path Traversal (CWE-22)
- OS Command Injection (CWE-78)
- Unrestricted File Upload (CWE-434)
- Cross-Site Request Forgery (CWE-352)
- Server-Side Request Forgery (CWE-918)
- Out-of-Bounds Read (CWE-125)
- Resource Management and Other Security Weaknesses

Each benchmark case contained:

- Case identifier
- Programming language
- Vulnerable source code
- Ground-truth vulnerability
- CWE classification
- Severity level
- Benchmark reference explanation
- Recommended mitigation

During benchmark construction, one source code sample contained three independent vulnerabilities (CWE-190, CWE-120, and CWE-639). The evaluation of the three vulnerabilities was performed separately as three distinct benchmark scenarios. This ensures that there is a single ground-truth vulnerability for each dataset observation and that the evaluation is consistent.

The benchmark dataset was carefully curated to encompass a broad spectrum of vulnerability categories, including authentication failures, authorization failures, injection vulnerabilities, memory-safety vulnerabilities, cryptographic vulnerabilities, resource-management vulnerabilities, and web-application security risks. Instead, this diversity was intended to test model performance across a wide range of cybersecurity domains, not just a narrow selection of vulnerability classes.

2.4 Benchmark Validation

The benchmark dataset was developed using publicly available cybersecurity resources, including the MITRE Common Weakness Enumeration (CWE) database, OWASP documentation, the Juliet Test Suite, WebGoat, Damn Vulnerable Web Application (DVWA), secure-coding repositories, and academic cybersecurity

datasets. Vulnerability labels and CWE classifications were cross-referenced against official MITRE CWE definitions and associated documentation to improve consistency and accuracy (MITRE, 2024).

The benchmark reference answers served as the evaluation standard throughout the study. Although the dataset was carefully curated and reviewed by the researchers, no formal independent expert-validation procedure was conducted. Future studies may strengthen benchmark reliability through expert review and inter-rater reliability assessment.

2.5 Experimental Procedure

2.5.1 Prompt Standardization

Prompt standardization was implemented to reduce experimental bias and improve reproducibility. Previous studies have shown that prompting strategies can substantially influence model outputs and evaluation outcomes (White et al., 2023; Schulhoff et al., 2024; Sahoo et al., 2024).

Standardized Prompt:

- *The prompt used during experimentation was:*
- *You are a cybersecurity analyst.*
- *Analyze the following source code.*
- *Identify all cybersecurity vulnerabilities.*

For each vulnerability, provide:

- *Vulnerability Name*
- *CWE Classification*
- *Severity Level*
- *Technical Explanation*
- *Recommended Mitigation*

The use of a similar prompt also confirmed that the differences in performance were due to differences in the AI models' capabilities, not to changes in prompt creation. To mitigate heterogeneity due to prompt engineering, we deliberately used a single, uniform prompt so that performance differences were largely attributable to model capabilities rather than to prompt crafting. Standardization of prompts is a well-recognized need for fair benchmarking, as it allows consistency, reproducibility, and direct comparability between the systems under evaluation. Other prompting

approaches may influence model performance. A common prompt was used to have a controlled basis of evaluation.

2.5.2 AI Evaluation

Each of the 50 benchmark cases was independently submitted to ChatGPT, Gemini, and Claude. For every benchmark case, the following information was recorded:

- identified vulnerability name,
- assigned CWE classification,
- severity level,
- technical explanation,
- recommended mitigation, and
- response completion time.

A total of 150 AI-generated responses (50 benchmark cases $\times$ 3 AI models) were collected and analyzed.

Table 1. Model configuration.

Model	Version	Platform	Testing Period
ChatGPT	GPT-4o	ChatGPT (OpenAI)	July 10-24, 2026
Gemini	Gemini 2.5 Pro	Google Gemini	July 10-24, 2026
Claude	Claude Sonnet 4	Anthropic Claude	July 10-24, 2026

All models were examined in the same evaluation period, using the same benchmark instances and uniform prompts. We also documented model versions to enable transparency and repeatability. All model configurations were kept at their defaults during experimentation, with the platform settings available.

2.5.3 Experimental Environment

The benchmarking experiments were conducted using commercially available web-based versions of ChatGPT (GPT-4o), Gemini 2.5 Pro, and Claude Sonnet 4. All evaluations were performed between July 10 and July 24, 2026, using the same workstation and internet connection to minimize environmental variability.

Hardware Specifications:

- Processor: Intel Core i7 or equivalent
- RAM: 16 GB

- Operating System: Windows 11 64-bit
- Browser: Google Chrome (latest version)

Model Parameters:

- Temperature: Default platform setting
- Top-p: Default platform setting
- Maximum output length: Platform default
- No additional system prompts applied.

All models received identical benchmark inputs and standardized prompts throughout the experiment.

2.6 Benchmark Evaluation Framework

Every AI response is compared to the benchmark reference answer. Instead of utilizing typical binary classification criteria, the replies were scored using a benchmark scoring mechanism based on the validity of the identified vulnerability and the Common Weakness Enumeration (CWE) categorization.

Table 2. Benchmark scoring framework.

Score	Interpretation
2	Exact Match: Correct vulnerability identification and exact CWE classification.
1	Related Match: Correct vulnerability identification and a CWE classification explicitly related to the benchmark CWE within the MITRE CWE hierarchy.
0	Incorrect Match: Incorrect vulnerability identification or a CWE classification not explicitly related to the benchmark CWE.

The scoring framework was meant to assess both the correct classification of vulnerabilities and the overall recognition of related software problems. Exact CWE identification is the maximum level of classification accuracy. However, some credit was given if the model recognized a CWE explicitly associated with the benchmark CWE in the MITRE Common Weakness Enumeration taxonomy.

The relatedness was determined only based on verified ties in the MITRE CWE to minimize subjectivity. Partial credit was only awarded when there was an explicitly defined link between the projected CWE and benchmark CWE (e.g., ParentOf, ChildOf, or PeerOf) as described in the CWE Relationships section and the associated CWE views. Classifications not documented as hierarchical or peer relationships were considered unconnected and received no credit. This approach enabled the determination of scores not at the evaluator's discretion but based on an official, standardized cybersecurity taxonomy. In this paradigm, an exact CWE match was awarded 2 points, an explicitly connected CWE classification was awarded 1 point,

and an unrelated classification was awarded 0 points. Each AI model could receive up to 100 points for each of the fifty benchmark cases used in the study.

To further reduce evaluator subjectivity, related classifications were credited only when explicit ParentOf, ChildOf, or PeerOf relationships were documented within the MITRE CWE taxonomy. No discretionary scoring decisions were applied beyond these predefined criteria. All classifications were evaluated using the same scoring rules across all three models.

Scoring was based exclusively on the benchmark ground-truth vulnerability assigned to each benchmark case. Additional vulnerabilities reported by a model neither increased nor reduced the assigned score. Evaluation focused solely on the correctness of the benchmark vulnerability classification and its corresponding CWE assignment.

The goal of this test was to evaluate how well Generative AI systems can identify software vulnerabilities and map them appropriately to known CWE classes. The framework integrates the stated CWE linkages into the scoring process, therefore accounting for the hierarchical nature of the CWE taxonomy while ensuring a rigorous and reproducible evaluation technique.

2.7 Evaluation Metrics

The performance was measured with benchmark-based vulnerability-classification criteria from the Common Weakness Enumeration (CWE) scoring framework used in this work. Each AI-generated response was compared to the benchmark reference answer and classified as an exact, related, or incorrect match.

2.7.1 Exact Match Rate

The Exact Match Rate represents the percentage of benchmark cases for which the AI model correctly identified both the vulnerability and its corresponding CWE classification.

$$EMR = \frac{Exact\ Matches}{Total\ Cases} \times 100$$

A higher Exact Match Rate indicates greater accuracy in classifying vulnerabilities and CWEs.

2.7.2 Related Match Rate

The Related Match Rate is the percentage of benchmark cases for which the AI model assigned a CWE classification with an explicit documented link to the benchmark CWE in the MITRE Common Weakness Enumeration taxonomy.

$$\text{Related Match Rate (\%)} = \frac{\text{Number of Related Matches}}{\text{Total Benchmark Cases}} \times 100$$

Related match occurred when the expected CWE and the benchmark CWE demonstrated a ParentOf, ChildOf, or PeerOf link in the MITRE CWE hierarchy. This metric allows responses that produce the exact benchmark CWE, as well as those that correctly identify closely related flaws.

2.7.3 Weighted Identification Score

Overall vulnerability-identification performance was measured using a weighted scoring framework that assigned:

- 2 points for an exact match;
- 1 point for a related match; and
- 0 points for an incorrect match.

The weighted score was computed as:

$$\text{Weighted Score} = (2 \times \text{Exact Matches}) + (1 \times \text{Related Matches})$$

The maximum potential weighted score was 100 points, based on the fifty benchmark instances employed in this study.

2.7.4 Weighted Performance Score

The Weighted Performance Score was used to measure overall model performance, considering both exact and related classifications.

$$\text{Weighted Performance Score (\%)} = \frac{\text{Weighted Score}}{100} \times 100$$

Higher numbers indicate greater overall vulnerability identification capabilities.

2.7.5 Exact Classification Rate

Exact Classification Rate measures a model's ability to assign the exact benchmark CWE classification.

$$\text{Exact Classification Rate (\%)} = \frac{\text{Exact Matches}}{\text{Total Benchmark Cases}} \times 100$$

The higher the Exact Classification Rate, the more accurately a model can distinguish among closely related vulnerability categories.

2.7.6 Vulnerability Recognition Rate

Vulnerability Recognition Rate was used to evaluate the model's ability to correctly recognize either the benchmark vulnerability or a documented related vulnerability within the MITRE CWE hierarchy.

$$\text{Recognition Rate (\%)} = \frac{\text{Exact Matches} + \text{Related Matches}}{\text{Total Benchmark Cases}} \times 100$$

Higher values indicate stronger vulnerability-recognition performance.

2.7.7 Mean Response Time

Response efficiency was assessed using mean response time.

$$\bar{x} = \frac{\sum t}{n}$$

where:

- t = individual response completion times;
- n = number of benchmark cases.

Lower values indicate lower observed response latency under the experimental conditions used in the study.

2.7.8 Ranking

Ranking was employed to determine the relative performance of ChatGPT, Gemini, and Claude across the evaluated metrics. Models were ranked according to:

1. Weighted Performance Score;
2. Exact Classification Rate;

3. Vulnerability Recognition Rate;
4. Mean Response Time; and

Collectively, these metrics provided complementary descriptive indicators of benchmark performance, exact classification capability, vulnerability recognition, and response latency.

2.8 Statistical Treatment of Data

2.8.1 Descriptive statistics

Descriptive statistics were used to summarize, analyze, and compare the performance of ChatGPT, Gemini, and Claude in identifying cybersecurity vulnerabilities. Descriptive statistics are well suited for benchmarking studies, as they provide a systematic approach to organizing, summarizing, and interpreting model performance across multiple evaluation criteria. The following statistical measures were calculated and analyzed:

- Frequency
- Percentage
- Mean
- Exact Match Rate
- Related Match Rate
- Weighted Identification Score
- Weighted Performance Score
- Exact Classification Rate
- Vulnerability Recognition Rate
- Mean Response Time
- Ranking

Statistical analyses were performed using JASP. Descriptive statistics, repeated-measures analysis, Friedman tests, Kendall's coefficient of concordance (W), and post hoc pairwise comparisons were computed using the software's default procedures.

2.8.2 Inferential Statistical Analysis

In addition to descriptive statistics, inferential analyses were conducted to determine whether the performance differences among ChatGPT, Gemini, and Claude were statistically significant.

The Friedman Test was selected as the major nonparametric statistical approach to compare overall model performance, given that all three AI models were tested over the identical benchmark instances, and the obtained scores were ordinal (0, 1, or 2).

Where necessary, pairwise comparisons were conducted using the Wilcoxon Signed-Rank Test to determine differences in performance between pairs of models. Effect size was assessed using Kendall's W to estimate the practical size of the observed changes. Statistical significance was determined at the 0.05 level.

2.9 Ethical Considerations

The work was conducted using well-established ethical criteria for cybersecurity and artificial intelligence research.

All benchmark instances were taken from publicly available educational, research, and cybersecurity sites. No testing was performed against live systems, production environments, proprietary software, or unauthorized targets. The study made sure that:

- The vulnerability information was utilized for academic and research reasons only.
- No exploit development or actual attacks were done.
- Benchmark cases were curated and reviewed by the researchers before experimentation.
- Benchmarking of AI-generated outputs against benchmark reference responses.
- Results were reported in an objective and open manner.
- Generative AI systems were evaluated as decision-support tools, not as alternatives for cybersecurity specialists.

3. Results

3.1 Vulnerability classification performance

The performance of the three AI systems (ChatGPT, Gemini, and Claude) in identifying cybersecurity vulnerabilities was evaluated using the benchmark scoring framework described in Section 2.6. Within this framework, an exact CWE match was scored 2, a CWE classification with an explicitly documented relationship to the benchmark CWE within the MITRE Common Weakness Enumeration hierarchy was scored 1, and an inaccurate or irrelevant CWE classification was scored 0. All models were evaluated on fifty (50) benchmark vulnerability instances.

Table 3. Distribution of vulnerability classification outcomes.

Classification Outcome	ChatGPT	Gemini	Claude
Full Credit (Score = 2)	30	25	29
Related Credit (Score = 1)	12	14	8
No Credit (Score = 0)	8	11	13
Total Cases	50	50	50

ChatGPT had the most full-credit classifications, with 30 exact CWE matches, or 60.0% of all benchmark cases, as shown in Table 3. Claude made 29 exact classifications (58.0%), and Gemini made 25 exact classifications (50.0%). Descriptive results indicate that ChatGPT achieved the highest number of exact benchmark CWE classifications among the evaluated models. However, subsequent inferential analysis did not identify statistically significant performance differences among the evaluated models.

Gemini had the most related-credit classifications, with 14 cases in which the predicted CWE was explicitly documented as related to the benchmark CWE. ChatGPT generated 12 related classifications, and Claude produced 8. This means that Gemini was more likely to discover vulnerabilities closely related to the benchmark CWE, even if it did not assign the same classification.

If we look at misclassifications, ChatGPT had 8 no-credit responses, Gemini had 11, and Claude had 13. While Claude showed a higher number of exact classifications, it also produced more incorrect classifications than ChatGPT under the strict MITRE relationship-based scoring framework.

Overall, ChatGPT achieved the highest number of exact classifications and the lowest number of incorrect classifications in the descriptive analysis. However, these observed differences should be interpreted with caution, as subsequent inferential analyses did not demonstrate statistically significant performance differences among the evaluated models.

3.2 Overall vulnerability identification performance

To evaluate overall vulnerability-identification capability, classification outcomes were converted into weighted scores using the scoring framework:

Weighted Score = $(2 \times \text{Full Credit}) + (1 \times \text{Related Credit})$

The maximum achievable score was 100 points.

Table 4. Overall vulnerability identification performance.

Model	Weighted Score	Maximum Score	Weighted Performance Score (%)	Rank
ChatGPT	72	100	72.00	1
Claude	66	100	66.00	2
Gemini	64	100	64.00	3

As shown in Table 4, ChatGPT obtained the highest weighted score of 72 points, corresponding to a Weighted Performance Score of 72.00%. Claude ranked second with a weighted score of 66 points (66.00%), while Gemini obtained 64 points (64.00%).

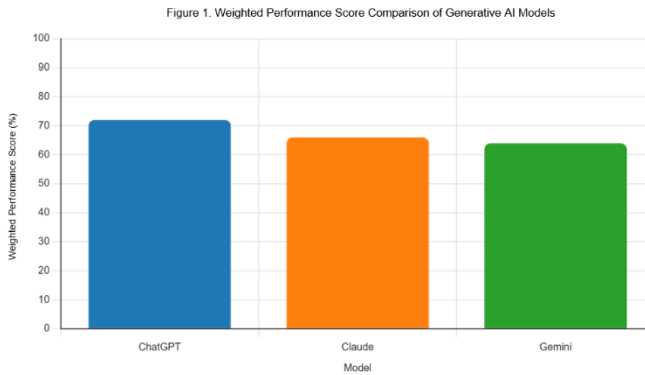


Figure 1. Weighted performance score comparison of generative AI models.

ChatGPT achieved the highest Weighted Performance Score (72.00%), followed by Claude (66.00%) and Gemini (64.00%). As illustrated in Figure 1, ChatGPT achieved the highest Weighted Performance Score among the evaluated models. Although all three models demonstrated effective vulnerability-identification capability, ChatGPT consistently achieved the highest benchmark score, indicating the highest descriptive performance observed in this benchmark. However, inferential analysis did not demonstrate statistically significant differences among the evaluated models.

3.3 Analysis of exact classification rate, vulnerability recognition rate, and speed

To provide a descriptive benchmarking assessment of model performance, four evaluation dimensions were analyzed: Exact Classification Rate, Vulnerability Recognition Rate, and Speed.

Table 5. Comparative performance metrics.

Metric	ChatGPT	Gemini	Claude
Weighted Performance Score (%)	72.00	64.00	66.00
Exact Classification Rate (%)	60.00	50.00	58.00
Vulnerability Recognition Rate (%)	84.00	78.00	74.00
Average Score per Case	1.44	1.28	1.32
Average Response Time (s)	3.76	8.27	5.59

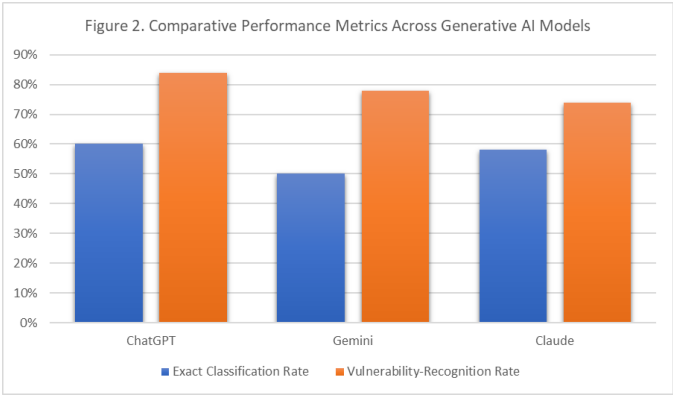


Figure 2. Comparative performance metrics across generative AI models

The figure compares the exact classification rate and vulnerability-recognition rate for ChatGPT, Gemini, and Claude. ChatGPT achieved the highest exact classification rate (60.00%) and vulnerability-recognition rate (84.00%). Figure 2 provides a graphical comparison of the principal performance metrics evaluated in this study. ChatGPT achieved the highest exact classification rate (60.00%) and vulnerability-recognition rate (84.00%). Claude demonstrated a comparable Exact Classification Rate (58.00%). Gemini achieved the second-highest vulnerability recognition rate (78.00%) but showed a lower Exact Classification Rate (50.00%). These findings further support the conclusion that ChatGPT provided the highest descriptive values across several benchmark metrics evaluated in this study. Nevertheless, the inferential results suggested that the observed differences were within the expected variation among the evaluated models.

3.3.1 Exact Classification Rate

Exact Classification Rate is the capacity of a model to identify the exact benchmark CWE classification correctly. The highest Exact Classification Rate was

achieved by ChatGPT at 60.00%, followed by 58.00% for Claude and 50.00% for Gemini. Descriptive results indicate that ChatGPT achieved the highest Exact Classification Rate under the benchmark conditions employed in this study.

3.3.2 Vulnerability Recognition Rate

Vulnerability Recognition Rate was measured using exact classifications and related classifications recognized within the MITRE CWE hierarchy. ChatGPT achieved the highest Vulnerability Recognition Rate at 84.00%, followed by Gemini at 78.00% and Claude at 74.00%. Descriptive results indicate that ChatGPT achieved the highest Vulnerability Recognition Rate among the evaluated models.

3.3.3 Speed

Response efficiency was determined by average response time. ChatGPT recorded the shortest observed response time, with an average latency of 3.76 seconds per benchmark instance. On average, Claude took 5.59 seconds to answer, while Gemini took 8.27 seconds.

ChatGPT achieved the highest descriptive benchmark scores and the shortest observed response latency among the evaluated models. Response-time measurements should be interpreted as observed latency under the experimental conditions rather than direct measures of intrinsic computational efficiency.

3.4 Summary of Results

Table 6. Overall performance ranking.

Rank	Model	Weighted Score	Weighted Performance Score (%)	Average Score per Case	Average Response Time (s)
1	ChatGPT	72	72.00	1.44	3.76
2	Claude	66	66.00	1.32	5.59
3	Gemini	64	64.00	1.28	8.27

Descriptive benchmark results indicated that ChatGPT achieved the highest weighted score, Exact Classification Rate, and Vulnerability Recognition Rate, and the lowest average response time among the evaluated models. Claude ranked second on several descriptive metrics, while Gemini achieved comparable overall performance and demonstrated the highest proportion of related CWE classifications. However, statistical testing indicated that the observed differences among the evaluated models were not statistically significant. Overall, the findings demonstrate that all three evaluated Generative AI systems can identify cybersecurity

vulnerabilities and map them to Common Weakness Enumeration (CWE) classifications.

ChatGPT achieved the highest descriptive benchmark scores and the lowest observed response latency; however, these observed differences should be interpreted in light of the non-significant inferential statistical results.

3.5 Inferential Statistical Analysis

Inferential statistical analyses were conducted to determine whether the performance differences observed among ChatGPT, Gemini, and Claude were statistically significant.

A Friedman test was performed because all three models were evaluated using the same benchmark dataset and generated ordinal performance scores ranging from 0 to 2. The analysis indicated that the observed differences were not statistically significant, $\chi^2(2) = 3.170$, $p = .205$. Since the p-value exceeded the predetermined significance level of .05, the null hypothesis could not be rejected.

To assess practical significance, Kendall's coefficient of concordance (W) was calculated and yielded a value of 0.032, indicating a negligible effect size. Pairwise post hoc comparisons further revealed no statistically significant differences between ChatGPT and Gemini ($p = .559$), ChatGPT and Claude ($p = .082$), or Gemini and Claude ($p = .244$).

These findings suggest that although descriptive performance differences were observed, all three Generative AI models demonstrated broadly comparable cybersecurity vulnerability-identification capability under the benchmark conditions employed in this study.

Table 7. Inferential statistical analysis results.

Test	Statistic	p-value	Interpretation
Friedman Test	$\chi^2(2) = 3.170$	.205	Not Significant
Kendall's W	0.032	—	Negligible Effect
ChatGPT vs Gemini	—	.559	Not Significant
ChatGPT vs Claude	—	.082	Not Significant
Gemini vs Claude	—	.244	Not Significant

4. Discussion

The results demonstrate that all evaluated Generative AI models possess meaningful capability for identifying cybersecurity vulnerabilities. Across the 50 tests, ChatGPT, Gemini, and Claude identified the correct problem or a similar one in most cases. These findings suggest that contemporary Large Language Models possess

substantial cybersecurity knowledge and demonstrate meaningful capability to identify software security weaknesses across multiple vulnerability categories.

Although ChatGPT achieved the highest descriptive scores on several benchmark metrics, inferential analysis did not identify statistically significant performance differences among the evaluated models. It achieved the highest Weighted Performance Score (72.00%), the highest exact classification rate (60.00%), the highest vulnerability recognition rate (84.00%), and the fastest average response time. Despite all that, the results should be interpreted as evidence of broadly comparable vulnerability-identification capability rather than definitive superiority of any single model.

The findings of this study are consistent with recent cybersecurity literature demonstrating that Large Language Models can effectively support vulnerability detection, secure code review, malware analysis, and software-security assessment tasks (Xu et al., 2024; Zhang et al., 2025). Prior studies have further shown that model performance is influenced by architecture, training methodology, and prompting strategies (OpenAI, 2023; Schulhoff et al., 2024). The observed differences in classification accuracy among ChatGPT, Gemini, and Claude are therefore consistent with broader findings reported in contemporary LLM literature.

Claude ranked second overall, achieving a Weighted Performance Score of 66.00% and an exact classification rate of 58.00%. Although Claude achieved a high exact classification rate, it generated more misclassifications than ChatGPT, thereby reducing its overall performance.

Gemini achieved a vulnerability-recognition rate of 78.00%, indicating a strong ability to identify cybersecurity weaknesses. However, its Exact Classification Rate was lower than that of ChatGPT and Claude, suggesting greater difficulty in assigning the exact benchmark CWE classifications.

A notable observation was that many inexact classifications corresponded to closely related vulnerability categories rather than being entirely incorrect. This suggests that the evaluated models generally recognized the underlying security weakness but occasionally struggled to distinguish between closely related CWE categories.

This shows that the models understood the type of security problem but sometimes had trouble distinguishing between names in the MITRE CWE list. So, the main problem for these models seems to be giving the right name for the problem, not finding the problem itself.

From a practical perspective, the results suggest that Generative AI may support vulnerability identification and CWE classification tasks like those evaluated in this study. Additional research is required before broader conclusions can be drawn regarding remediation support, explanation quality, professional usability, or cybersecurity decision-making.

There are some things to keep in mind when looking at the results. The tests used fifty problems, which might not cover all the different security issues that exist in real life. Also, three company models were tested using the same method to ask the question. Using different prompting methods, different model versions, or other models designed for security might yield different results. Also, the study mainly focused on identifying problems and naming them. Did not check how good the explanations were, how well solutions were given, or how the models changed over time.

Future studies should investigate the influence of advanced prompting techniques, including role-based, chain-of-thought, and few-shot prompting, which have been shown to improve task performance across various LLM applications (White et al., 2023; Sahoo et al., 2024).

Future benchmarking studies should also examine how prompt-engineering strategies affect cybersecurity vulnerability-identification performance across different LLM architectures (Schulhoff et al., 2024; Sahoo et al., 2024).

Overall, the findings indicate that Generative AI systems may provide useful support for vulnerability identification and CWE classification tasks similar to those evaluated in this study. Although descriptive performance differences were observed, inferential statistical analysis identified no statistically significant differences among the evaluated models.

5. Conclusion

The study evaluated the cybersecurity vulnerability identification performance of ChatGPT, Gemini, and Claude using 50 benchmark cases across different CWE categories. All three models demonstrated meaningful capability to recognize vulnerabilities and associate them with relevant CWE classifications. Although ChatGPT obtained the highest descriptive scores in weighted performance, exact classification, and vulnerability recognition, the Friedman Test showed that the differences among the models were not statistically significant. Some errors involved assigning related CWE categories rather than completely failing to identify the vulnerability, indicating difficulty in distinguishing closely related weaknesses. Overall, the models showed broadly comparable performance and potential as supplementary tools for cybersecurity analysis. However, their outputs should undergo expert validation, and Generative AI should be used as a decision-support tool rather than as a standalone security solution. Future studies should use larger, more diverse datasets, include additional models, examine prompting strategies and different programming environments, assess the quality of responses and mitigation, and explore integration with established static and dynamic analysis tools.

6. Limitation of the Study

Several limitations should be considered when interpreting the findings. The study used only 50 curated vulnerability cases, which provided controlled comparative benchmarking but may not represent the full range of real-world cybersecurity vulnerabilities. Although reference classifications were based on authoritative sources such as MITRE CWE, no formal independent expert validation or inter-rater reliability assessment was conducted, and some overlap among benchmark cases may remain.

The study evaluated only ChatGPT, Gemini, and Claude using a single standardized prompt. Other models, prompting strategies, and model-specific configurations may produce different results. The assessment was also limited to vulnerability identification and CWE classification and did not examine explanation quality, mitigation quality, false positives, reasoning consistency, or professional usability.

Because the benchmark cases were drawn from publicly available cybersecurity resources, the model's familiarity with similar examples cannot be ruled out. Each case was also tested only once per model using default settings, so run-to-run variability was not assessed. The findings should therefore be interpreted as an exploratory snapshot rather than a definitive measure of model performance. Response-time results may also have been affected by network and platform conditions. Finally, because Large Language Models are continuously updated, the results apply only to the model versions evaluated during the study period.

Acknowledgment

The authors sincerely thank the peer reviewers for their valuable comments and constructive suggestions, which contributed to the improvement of this manuscript.

Conflict of Interest Statement

The authors declare no conflict of interest.

References

- Creswell, J. W., & Creswell, J. D. (2022). *Research design: Qualitative, quantitative, and mixed methods approaches* (6th ed.). SAGE Publications. <https://www.sagepub.com/shop/buy-a-book/research-design-6-270550>
- Goseva-Popstojanova, K., & Perhinschi, A. (2015). On the capability of static code analysis to detect security vulnerabilities. *Information and Software Technology*, 68, 18–33. <https://doi.org/10.1016/j.infsof.2015.08.002>

- Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y., Madotto, A., & Fung, P. (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12), Article 248, 1–38. <https://doi.org/10.1145/3571730>
- MITRE. (2024). *Common Weakness Enumeration (CWE)*. <https://cwe.mitre.org/>
- OpenAI. (2023). *GPT-4 technical report*. arXiv. <https://arxiv.org/abs/2303.08774>
- OWASP Foundation. (2025). *OWASP Top 10: The ten most critical web application security risks*. <https://top10.owasp.org/2025/>
- Pearce, H., Ahmad, B., Tan, B., Dolan-Gavitt, B., & Karri, R. (2022). Asleep at the keyboard? Assessing the security of GitHub Copilot's code contributions. In *2022 IEEE Symposium on Security and Privacy Workshops (SPW)* (pp. 754–768). IEEE. <https://doi.org/10.1109/SP46214.2022.9833571>
- Sahoo, P. K., Singh, A. K., Saha, S., Jain, V., Mondal, S., & Chadha, A. (2024). A systematic survey of prompt engineering in large language models: Techniques and applications. *ACM Computing Surveys*. <https://doi.org/10.1145/3691210>
- Schulhoff, S., Ilie, M., Balepur, N., Kahou, S. E., Michalski, V., & Pal, C. (2024). *The prompt report: A systematic survey of prompting techniques*. arXiv. <https://arxiv.org/abs/2406.06608>
- White, J., Hays, S., Fu, Q., Spencer-Smith, J., & Schmidt, D. C. (2023). *A prompt pattern catalog to enhance prompt engineering with ChatGPT*. arXiv. <https://arxiv.org/abs/2302.11382>
- Xu, H., Wang, S., Li, N., Wang, K., Zhao, Y., Chen, K., Yu, T., Liu, Y., & Wang, H. (2024). *Large language models for cyber security: A systematic literature review*. arXiv. <https://arxiv.org/abs/2405.04760>
- Zhang, J., Bu, H., Wen, H., Liu, Y., Fei, H., Xi, R., Li, L., Yang, Y., Zhu, H., & Meng, D. (2025). When LLMs meet cybersecurity: A systematic literature review. *Cybersecurity*, 8(1), Article 55. <https://doi.org/10.1186/s42400-025-00361-w>

Author Contributions: Dungon, E. R.; Irog, M. J.; Salva, H. L.; Alvior, J. L.; Bascar, R. M.; Aspera, R. A.; Palmares, S.; Soberano, K.; Study design, method conception, data collection, data analysis, and manuscript writing. All authors have read and approved the final version of the manuscript and agreed to be accountable for the integrity and scholarly content of the work.