INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2026, VOL. 72, NO. 1, PP. 1–8
Manuscript received September 17, 2025; revised February, 2026. doi: 10.24425/ijet.2026.157870

Application of Large Language Models to automatic classification of vulnerabilities according to the CVSS 3.1 standard

Michal Walkowski, Nikita Zhukov, and Slawomir Sujecki

***Abstract*—We evaluated three chatbot models (ChatGPT-4o-mini, Gemini 2.0 Flash, Deepseek Chat) to automate CVSS 3.1 vulnerability scoring using 4,459 CVE records. Gemini achieved the highest accuracy across prompt strategies, while ChatGPT showed vector-score inconsistencies, and Deepseek underestimated severity. Results suggest that chatbots can support analysts but require validation mechanisms.**

***Keywords*—Security; CVSS Score; Well-known Vulnerabilities; Large Language Model**

I. Introduction

CYBERSECURITY is increasingly recognized as a critical factor for the stability of societies, economies, and everyday digital interactions [1], [2]. The number of disclosed software vulnerabilities continues to grow year by year, creating a significant challenge for organizations that rely on digital infrastructure. According to the National Vulnerability Database (NVD), more than 39,000 new Common Vulnerabilities and Exposures (CVEs) were registered in 2024, surpassing previous record years and illustrating the accelerating pace of security disclosures [3]. This continuous increase is driven not only by the rapid expansion of software ecosystems but also by the adoption of automated vulnerability discovery tools and the growing role of coordinated vulnerability disclosure programs. Consequently, the identification, evaluation, and prioritization of vulnerabilities have become a critical part of security operations for governments, enterprises, and end users [4], [5].

The Common Vulnerability Scoring System (CVSS) is the most widely adopted framework for assessing vulnerability severity. CVSS 4.0, released in late 2023, is the most recent version [6]; however, it is not yet widely used in practice, with CVSS 3.1 still being the dominant standard applied across the industry [7]. One of the persistent challenges is the lack of complete CVSS 3.1 coverage for all vulnerabilities. Many older CVEs are available only with CVSS 2.0 scores, requiring conversion to CVSS 3.1 in order to be properly integrated into modern vulnerability management workflows [8]–[10]. Manual calculation of CVSS vectors remains time-consuming and requires expert knowledge, leading to delays, inconsistencies, and potential misclassifications [11]. Furthermore, different vendors apply proprietary vulnerability prioritization methods [12]–[15], which lack transparency and may result in decisions based on undisclosed heuristics rather than standardized scoring [16].

The introduction of machine learning and natural language processing further advanced this field. Shahid and Debar proposed CVSS-BERT [17], using deep learning to predict CVSS metrics from vulnerability descriptions. Kekül et al. [18] addressed the issue of missing vectors in the NVD by applying ML classifiers, while Nowak et al. [9] developed models to convert CVSS 2.0 to 3.x scores, highlighting the ongoing transition challenges. These works illustrate the potential of AI but focus primarily on structured ML rather than generative models.

With the rise of Large Language Models (LLMs) [19], recent works have explored their applicability to CVSS scoring. Turtiainen et al. [20] evaluated ChatGPT in a zero-shot setting on 113,000 CVEs, achieving 65% correctness in severity categories but only 20% exact vector matches. McClanahan et al. [21] analyzed practical advantages and risks of ChatGPT in vulnerability management workflows, emphasizing hallucinations and inconsistency issues. Marchiori et al. [22] compared LLM-based methods with hybrid embedding approaches, concluding that hybrid models achieve higher reliability. Liu et al. [23] confirmed that prompt design significantly influences LLM performance, while Chopra et al. [24] proposed ChatNVD, an LLM-powered interface for interactive vulnerability exploration.

Compared to these studies, the work presented in this paper introduces several novel contributions. First, unlike prior evaluations that used naturally skewed datasets, this study applies a balanced dataset across the full CVSS 3.1 spectrum (0.0-10.0), ensuring fairness across severity levels. Second, while most prior research examined a single chatbot model (e.g., GPT), here we perform a multi-chatbot comparative analysis including ChatGPT-4o-mini, Gemini 2.0 Flash, and Deepseek Chat, revealing model-specific strengths and weak-

M. Walkowski and N. Zhukov are with the Department of Telecommunications and Teleinformatics, Wroclaw University of Science and Technolgy, Poland (e-mail: michal.walkowski@pwr.edu.pl, ORCID: https://orcid.org/0000-0003-4887-6098; 262922@student.pwr.edu.pl).

S. Sujecki is with the Department of Telecommunications and Teleinformatics, Wroclaw University of Science and Technology, Poland, and also with the Faculty of Electronics, Military University of Technology, Poland (e-mail: slawomir.sujecki@pwr.edu.pl, ORCID: https://orcid.org/0000-0003-4588-6741).

nesses. Third, we extend previous work by systematically analyzing prompt engineering strategies, demonstrating that different chatbots react inconsistently to augmented or restrictive prompts. Finally, our study provides an in-depth error analysis highlighting systematic biases, such as ChatGPT's inconsistency between vectors and scores, and Deepseek's severe underestimation of critical vulnerabilities.

Thus, this contribution complements the existing literature by expanding the scope of evaluation, proposing a methodology for fair benchmarking, and identifying new pitfalls in chatbot-based CVSS scoring. These findings are of practical relevance for organizations considering the integration of conversational AI into vulnerability management processes.

In this contribution, we present a systematic evaluation of chatbot-based automation for CVSS 3.1 scoring. Building on the balanced dataset of 4,459 CVEs spanning the full range of base scores, we compare three modern chatbot platforms: ChatGPT-4o-mini (OpenAI), Gemini 2.0 Flash (Google), and Deepseek Chat. Our study examines the impact of prompt engineering on scoring accuracy, identifies systematic biases across models, and provides a quantitative and qualitative error analysis.

The remainder of this paper is organized as follows:

- Background - introduces CVSS 3.1 and the fundamentals of LLM-based chatbots.
- Related Work - summarizes existing approaches to automated vulnerability scoring and chatbot-assisted security tasks.
- Methodology - describes the dataset, chatbot models, and evaluation metrics.
- Results - presents experimental findings, including accuracy, F1-scores, and error analysis.
- Discussion - critically analyzes the results in comparison with the literature, highlighting practical implications.
- Conclusions - provide a summary of findings and outline directions for future research.

II. Methodology

This section describes the methodology adopted in order to evaluate the applicability of modern chatbot systems for automated CVSS 3.1 scoring. The research design follows a structured approach consisting of dataset preparation, chatbot model selection, prompt engineering, and evaluation procedures.

The dataset used in this study was collected from the National Vulnerability Database (NVD). A total of 4,459 CVE entries with official CVSS 3.1 vectors were retrieved. In order to minimize bias introduced by the naturally skewed distribution of vulnerabilities, the dataset was balanced across all eleven CVSS base score intervals (0.0-1.0, 1.0-2.0, ..., 9.0-10.0). For each interval, up to 500 CVEs were randomly selected. This ensured that the evaluation would equally represent vulnerabilities from low to critical severity, rather than reflecting the over representation of medium and high scores observed in real-world distributions.

Each CVE entry included the vulnerability description, the official CVSS 3.1 vector, and the corresponding base score. These records were used as the reference dataset for evaluating chatbot outputs.

The evaluation of chatbot suitability for CVSS 3.1 scoring was conducted across three models provided by different vendors. The primary selection criteria included the lowest cost per token and the shortest response time, as these factors directly influence the feasibility of large-scale automated assessments. Initially, four models were considered: *ChatGPT-4o-mini* (OpenAI), *deepseek-chat* (DeepSeek), *gemini-2.0-flash* (Google), and *grok-3-mini* (X.ai).

- ChatGPT-4o-mini (OpenAI) - a compact version of GPT-4 optimized for efficiency and cost-effectiveness, widely available through OpenAI's API. It was selected for its balance between performance and low pricing, making it suitable for high-volume experiments.
- Gemini 2.0 Flash (Google) - the most advanced real-time chatbot available at the time of writing, offering multimodal capabilities. Its main advantage is very low latency, combined with competitive pricing and strong reliability in structured tasks.
- Deepseek Chat (DeepSeek) - an open-source large language model aimed at general-purpose reasoning. Although slower than Gemini and ChatGPT, it offered flexibility and transparency, making it an interesting candidate for evaluation.

API keys were obtained for all platforms, and preliminary response time measurements were conducted. The average generation time for *grok-3-mini* was 15 seconds per query, which implies a runtime of approximately 18.5 hours for a single test round (4,459 queries). The faster variant, *grok-3-mini-fast*, reduced the average time to 9 seconds, but the total runtime (around 11 hours) was still considerably longer than that of the other models, and the associated cost was significantly higher [25]–[28]

Due to these limitations, the Grok model family was excluded from further testing. As a result, the final evaluation focused exclusively on ChatGPT-4o-mini, Gemini 2.0 Flash, and Deepseek Chat.

The experiment was designed to test three different prompting strategies, reflecting the impact of input formulation on chatbot performance:

- Test 1 - Baseline Prompt: Each CVE description was provided to the chatbot with the instruction: "Provide the CVSS 3.1 vector and the corresponding base score." This represented a zero-shot scenario without additional guidance.
- Test 2 - Augmented Prompt: In addition to the CVE description, the chatbot was provided with a structured explanation of CVSS 3.1 metric categories and expected output format. This aimed to reduce syntactic errors and improve adherence to the CVSS standard.
- Test 3 - Restrictive Prompt: The CVE description was provided together with explicit instructions prohibiting the chatbot from using external knowledge sources (e.g., databases or training memory), requiring it to rely solely

on the given description. This tested the chatbot's ability to reason without leveraging latent memorization.

In order to assess chatbot performance, several evaluation metrics were defined:

- Accuracy: The percentage of cases where the chatbot's predicted base score exactly matched the official score, as well as accuracy within a ± 1.0 tolerance range.
- F1-Score: Computed for each severity class (Low, Medium, High, Critical) based on precision and recall, in order to evaluate classification balance.
- Consistency: The degree to which the generated CVSS vector corresponded to the declared base score when recalculated using the official CVSS formula.
- Error Analysis: Systematic categorization of common mistakes, including underestimation, overestimation, invalid vectors, and contradictions between vector and score.

For each chatbot, the 4,459 CVE descriptions were submitted under each of the three prompt variants, yielding a total of 13,377 queries per model. The chatbot responses were captured and automatically parsed using Python scripts with regular expressions, ensuring that both vectors and scores were extracted. Responses failing to comply with the expected format were flagged as syntactic errors.

Subsequently, each predicted score was compared against the reference CVSS 3.1 score from NVD. Accuracy and F1-scores were calculated, while inconsistencies between vectors and declared scores were separately logged.

All experiments were performed using publicly accessible chatbot APIs. Data preprocessing and evaluation scripts were implemented in Python, with Pandas used for dataset handling and Matplotlib for visualization. Experiments were executed on a Linux-based environment with sufficient memory and processing power to handle multiple API requests in parallel.

III. Experiment

This section presents the findings of the three experiments described in methodology section. Results are structured by prompt type and chatbot model.

A. Baseline Test

The baseline test applied the zero-shot instruction. This experiment served as a reference point for subsequent tests. The prompt required the chatbot to generate only the Base Score value and the corresponding CVSS 3.1 vector. The objective of this test was to obtain reference results against which deviations in the two subsequent experiments could be evaluated, as well as to compare the efficiency of the models in terms of usage cost and total runtime.

1) ChatGPT-4o-mini: : The baseline run lasted 1h 32m 58s at a total cost of $0.20 [25]. Table I shows the distribution of predicted and vector-based scores compared to the Actual. Predictions were concentrated in the 5-8 range (82.3%), while vector-based recalculations produced a more balanced spread. Eight malformed vectors were recorded.

TABLE I
DISTRIBUTION OF PREDICTED AND VECTOR-BASED SCORES FOR CHATGPT-4O-MINI IN THE BASELINE TEST

Score Interval	Actual	Predicted	Vector-Based
0–2	37	2	15
3–5	1000	156	682
6–8	2634	3686	2801
9–10	788	615	953

The evaluation of *ChatGPT-4o-mini* under the baseline prompt produced two distinct result sets depending on whether directly predicted scores or vector-based recalculations were considered.

For the direct predictions, overall accuracy reached 42.4%, with a weighted precision of 0.58, recall of 0.42, and an F1-score of 0.36.

By contrast, when recalculated from the generated CVSS vectors, performance improved modestly. Accuracy increased to 45.2%, with weighted precision of 0.53, recall of 0.45, and F1-score of 0.44. This improvement stemmed primarily from better handling of the *Critical [9-10]* class, where recall rose from 0.19 to 0.51, yielding an F1-score of 0.56. Nevertheless, the model still struggled with *Low [0-2]*-severity vulnerabilities (recall of 0.20) and continued to misclassify a large fraction of them as *Medium [3-5]* or *High [6-8]*. In summary, ChatGPT-4o-mini showed a bias toward medium severity, frequently downgrading critical vulnerabilities. Vector-based outputs improved alignment but introduced inconsistencies between scores and vectors.

2) Gemini 2.0 Flash: : The run with achieved the lowest cost ($0.151) [26] and shortest runtime 46m 42s.

Table II compares the distribution of predicted and vector-based scores generated by *Gemini 2.0 Flash* against the ground truth values. The direct predictions were heavily skewed towards the *Medium* [6-8] and *Critical* [9-10] ranges, with 3310 and 942 cases assigned, respectively. This resulted in a substantial underestimation of the *Low* class [0-2], where only 202 predictions were made compared to 1000 actual cases.

TABLE II
DISTRIBUTION OF PREDICTED AND VECTOR-BASED SCORES FOR GEMINI 2.0 FLASH IN THE BASELINE TEST

Score Interval	Actual	Predicted	Vector-Based
0–2	37	5	12
3–5	1000	202	734
6–8	2634	3310	2491
9–10	788	942	1222

The baseline prompt achieved an overall accuracy of 47.6% for both direct predictions and vector-based recalculations. The weighted precision was comparable in both cases (0.56 vs. 0.56), as were recall (0.48 vs. 0.48) and F1-score (0.44 vs. 0.44). These values demonstrate stable performance across both evaluation modes.

Vector-based results were nearly identical, with slight improvements in the stability of classification for *Low [0-2]* and

Critical [9-10] classes. Importantly, the vector-based approach did not introduce additional inconsistencies, and the misclassification patterns remained the same as for direct predictions.

In summary, *Gemini 2.0 Flash* produced the most balanced performance among the tested chatbots in the baseline setting. It showed consistent recognition of *Critical [9-10]* vulnerabilities and reliable handling of *Medium [3-5]* ones, albeit at the expense of very poor treatment of the *Info* and *Low* categories. Nevertheless, compared to ChatGPT-4o-mini and Deepseek Chat, Gemini achieved the best overall trade-off between precision and recall.

3) Deepseek Chat: : The run lasted 7h 54m 17s at a cost of $0.28 [27]. Table III presents the distribution of predicted and vector-based scores generated by *Deepseek Chat* in comparison with the actual ground truth values. The model exhibited a strong bias towards higher severity classes. In the case of direct predictions, 2800 vulnerabilities were classified as *Medium* [3-5] and as many as 1528 as *Critical* [9–10], compared to the reference values of 2634 and 788, respectively. This represents a near doubling of the *Critical* class, clearly illustrating Deepseek's tendency to overestimate severity.

TABLE III
DISTRIBUTION OF PREDICTED AND VECTOR-BASED SCORES FOR DEEPSEEK CHAT IN THE BASELINE TEST

Score Interval	Actual	Predicted	Vector-Based
0–2	37	1	5
3–5	1000	130	620
6–8	2634	2800	2400
9–10	788	1528	1434

The baseline test produced an overall accuracy of 47.1% for direct predictions and 46.8% for vector-based recalculations. Weighted precision remained in the range of 0.54–0.55, with recall around 0.47 and F1-scores of 0.44 (predicted) and 0.43 (vector-based). This indicates a consistent but modest performance across both evaluation modes.

For direct predictions revealed that the model handled *Critical* vulnerabilities relatively well, with a recall of 0.78, but frequently misclassified *Low [0-2]*-severity cases as either *Medium [3-5]* or *High [7-8]*, yielding a very low recall of 0.14 despite high precision (0.84). The *Medium* class achieved balanced results with an F1-score of 0.54, while the *Info* class remained poorly represented, with only two correct identifications out of 15.

Vector-based recalculations produced nearly identical results, with minor fluctuations: recall for the *Critical [9-10]* class dropped slightly from 0.78 to 0.77, while precision for the *Low [0-2]* class improved marginally. However, these changes were within the margin of noise and did not alter the overall performance profile of the model.

In summary, *Deepseek Chat* demonstrated a strong bias towards high-severity categories, particularly inflating the number of *Critical [9-10]* vulnerabilities, while severely underestimating *Low [0-2]* and *Info* classes. Compared to ChatGPT-4o-mini, Deepseek achieved better recall for critical vulnerabilities but remained less balanced overall than Gemini 2.0 Flash.

B. Extended Prompt Test

Similar to the baseline test, the discussed prompt required the chatbot to generate only the CVSS 3.1 Base Score together with the corresponding vector. However, in this case the full structure of the vector and all possible parameter values for the Base Score were explicitly provided. The objective of this procedure was to completely eliminate the problem of malformed vectors and to identify potential shifts in scoring tendencies once supplementary information was supplied.

It was observed that under this prompt, the *ChatGPT-4o-mini* model periodically returned outputs in a deviating format, such as inserting the word "Vector" or introducing a space between the "CVSS" prefix and the rest of the vector (e.g., "CVSS 3.1/…"). Such anomalies occurred exclusively in this test and may be attributed to ambiguities in the response format implied by the prompt design.

1) ChatGPT-4o-mini: : With the extended prompt, *ChatGPT-4o-mini* completed the task in 1h 35m 02s at a total cost of $0.37. The use of additional instructions successfully eliminated malformed vectors, which were present in the baseline test. However, this improvement in syntactic compliance came at the expense of overall classification performance. Accuracy and F1-scores both declined compared to baseline values, indicating that the extended prompt did not translate into better semantic understanding of CVSS scoring.

Table IV shows the distribution of predicted and vector-based scores. As in the baseline test, the model remained heavily biased toward the *Medium* severity range (6–8), with 3610 predictions versus the actual 2634 cases. This over-concentration occurred at the expense of *Low* - severity vulnerabilities (3–5), which were severely underrepresented in direct predictions (200 vs. 1000). Vector-based recalculations partially corrected the imbalance, raising the *Low* class to 742 cases and aligning the *Critical* range (9–10) closer to the ground truth, but still inflating its frequency (1005 vs. 788). The *Info/None* category (0–2) remained almost entirely neglected.

TABLE IV
DISTRIBUTION OF PREDICTED AND VECTOR-BASED SCORES FOR CHATGPT-4O-MINI IN THE EXTENDED TEST

Score Interval	Actual	Predicted	Vector-Based
0–2	37	4	12
3–5	1000	200	742
6–8	2634	3610	2700
9–10	788	645	1005

Evaluation metrics in Table V confirm this degradation. Direct predictions yielded a precision of 0.39, recall of 0.38, and an F1-score of 0.39. Vector-based scores showed slightly higher recall (0.41) and F1 (0.42), but precision dropped further to 0.37. Compared to baseline results, both configurations represent a noticeable regression in performance.

TABLE V
EVALUATION METRICS FOR CHATGPT-4O-MINI IN THE EXTENDED TEST

Metric	Predicted	Vector-Based
Precision	0.39	0.37
Recall	0.38	0.41
F1-score	0.39	0.42

In summary, while the extended prompt successfully improved syntactic correctness by eliminating malformed vectors, it also reduced the overall classification reliability of *ChatGPT-4o-mini*. The model continued to overestimate medium-severity vulnerabilities and failed to improve detection of low- and information-level cases, confirming its limited ability to leverage detailed instructions in this task.

2) Gemini 2.0 Flash: : With the extended prompt, *Gemini 2.0 Flash* completed the evaluation in 46m 16s at a cost of $0.263. The introduction of detailed instructions not only eliminated malformed vectors but also improved the balance of predictions across severity classes compared to the baseline test.

Table VI shows that direct predictions were still slightly biased toward the *Medium* (6–8) and *Critical* (9–10) ranges, but the overall distribution more closely approximated the ground truth than in the baseline. For instance, *Low [0-2]*-severity vulnerabilities (3–5) rose to 210 predictions versus only 202 in the baseline, and vector-based recalculations further improved this figure to 760, aligning closely with the reference 1000 cases. Similarly, the *Critical [9-10]* class expanded to 1042 direct predictions and 1225 vector-based scores, both closer to the target 788 than the baseline results.

TABLE VI
DISTRIBUTION OF PREDICTED AND VECTOR-BASED SCORES FOR GEMINI 2.0 FLASH IN THE EXTENDED TEST

Score Interval	Actual	Predicted	Vector-Based
0–2	37	7	14
3–5	1000	210	760
6–8	2634	3200	2460
9–10	788	1042	1225

Evaluation metrics presented in Table VII confirm this improvement. Direct predictions achieved a precision of 0.46, recall of 0.49, and F1-score of 0.52, while vector-based results raised recall to 0.52 and the F1-score to 0.55. Both sets of results represent a measurable increase over baseline performance, particularly in terms of F1, which reflects a more balanced handling of precision and recall.

TABLE VII
EVALUATION METRICS FOR GEMINI 2.0 FLASH IN THE EXTENDED TEST

Metric	Predicted	Vector-Based
Precision	0.46	0.45
Recall	0.49	0.52
F1-score	0.52	0.55

In summary, *Gemini 2.0 Flash* demonstrated the most substantial improvement under the extended prompt, achieving the best overall accuracy and F1 among the tested models. The elimination of malformed vectors and the more balanced class distribution confirms that Gemini was uniquely capable of leveraging extended instructions to enhance CVSS 3.1 scoring reliability.

3) Deepseek Chat: : With the extended prompt, *Deepseek Chat* required 8h 00m 03s to complete the evaluation, at a cost of $0.38. While malformed vectors were successfully eliminated, overall classification performance degraded compared to the baseline test.

Table VIII illustrates the distribution of predicted and vector-based scores. Direct predictions remained skewed toward the *Medium* (6–8) and *Critical* (9–10) classes, with 2900 and 1387 cases assigned, respectively, compared to the actual 2634 and 788. This overestimation came at the expense of *Low [0-2]*-severity vulnerabilities (3–5), which were severely underrepresented (170 vs. 1000). Vector-based recalculations improved the balance slightly, raising the *Low* class to 680 and reducing the inflation of the *Medium [6-8]* and *Critical [9-10]* categories, but the distributions still diverged substantially from the reference.

TABLE VIII
DISTRIBUTION OF PREDICTED AND VECTOR-BASED SCORES FOR DEEPSEEK CHAT IN THE EXTENDED TEST

Score Interval	Actual	Predicted	Vector-Based
0–2	37	2	8
3–5	1000	170	680
6–8	2634	2900	2440
9–10	788	1387	1331

Evaluation metrics in Table IX confirm this decline in performance. Direct predictions achieved precision of 0.37, recall of 0.42, and an F1-score of 0.37. Vector-based scores showed a marginal increase in recall (0.43) and F1 (0.39), but precision fell further to 0.35. These values are consistently lower than those obtained in the baseline test, particularly for the *Medium [6-8]* class, which experienced the sharpest loss in accuracy.

TABLE IX
EVALUATION METRICS FOR DEEPSEEK CHAT IN THE EXTENDED TEST

Metric	Predicted	Vector-Based
Precision	0.37	0.35
Recall	0.42	0.43
F1-score	0.37	0.39

In summary, although the extended prompt eliminated malformed vectors, it did not improve semantic reliability for *Deepseek Chat*. On the contrary, both predicted and vector-based outputs exhibited worsened performance, reinforcing the model's tendency to overestimate higher-severity vulnerabilities and confirming its limited ability to benefit from extended instruction design.

C. Restricted Prompt Test

In the third test, an experiment was conducted by extending the baseline prompt with an additional clause explicitly prohibiting the chatbot from using external vulnerability databases when assigning severity scores.

This approach, however, is subject to inherent limitations. A language model that has been trained on information originating from prohibited databases (e.g., CVE) does not necessarily have a mechanism to fully disentangle such knowledge from the rest of its training corpus [29]. Consequently, completely disabling the influence of these sources may be either impossible or difficult to verify in practice. Despite this uncertainty, the test was carried out because similar restrictions were applied in the study by [30], discussed in methodology section, and the results were intended to provide a comparative verification of the observations reported there.

1) ChatGPT-4o-mini: : With the restricted prompt, *ChatGPT-4o-mini* completed the evaluation in 1h 31m 34s. Predictions once again concentrated in the *Medium* (6–8) severity range, while the number of malformed vectors increased to eleven, representing a regression compared to both the baseline and extended tests.

Table X shows that the distribution remained highly imbalanced. Direct predictions heavily favored the *Medium* category (3700 vs. 2634 actual cases), while underestimating *Low*-severity vulnerabilities (140 vs. 1000) and largely neglecting the *Info/None* range (3 vs. 37). Vector-based recalculations partially corrected the imbalance, raising the *Low* class to 700 and increasing the *Critical* category (1000 vs. 788 actual), but systematic biases persisted.

TABLE X
DISTRIBUTION OF PREDICTED AND VECTOR-BASED SCORES FOR CHATGPT-4O-MINI IN THE RESTRICTED TEST

Score Interval	Actual	Predicted	Vector-Based
0–2	37	3	9
3–5	1000	140	700
6–8	2634	3700	2750
9–10	788	616	1000

Evaluation metrics in Table XI confirm the performance decline. Direct predictions achieved a precision of 0.38, recall of 0.37, and F1-score of 0.38. Vector-based results slightly improved recall (0.40) and F1 (0.41), but precision dropped further to 0.36. These results are consistently lower than those observed in the extended test, demonstrating that the restrictive instructions did not enhance scoring reliability.

TABLE XI
EVALUATION METRICS FOR CHATGPT-4O-MINI IN THE RESTRICTED TEST

Metric	Predicted	Vector-Based
Precision	0.38	0.36
Recall	0.37	0.40
F1-score	0.38	0.41

In summary, the restrictive test not only degraded classification performance but also introduced more malformed outputs. *ChatGPT-4o-mini* showed limited ability to adapt to prohibitive instructions, reinforcing its tendency to over-concentrate predictions in the medium-severity range and confirming its instability when deprived of broader contextual cues.

2) Gemini 2.0 Flash: : With the restricted prompt, *Gemini 2.0 Flash* completed the evaluation in 00:48:53. A total of four malformed vectors were recorded, which represent a minor formatting issue compared to the extended test. Nevertheless, overall accuracy declined slightly, particularly in the *Medium* severity class, where misclassifications increased.

Table XII presents the distribution of predicted and vector-based scores compared to the ground truth. Direct predictions again showed a strong bias toward the *Medium* range (3320 vs. 2634 actual), while underestimating the *Low* class (198 vs. 1000). Vector-based recalculations improved balance slightly, recovering 740 cases in the *Low* range and increasing the *Critical* category (1204 vs. 788 actual), though systematic deviations persisted.

TABLE XII
DISTRIBUTION OF PREDICTED AND VECTOR-BASED SCORES FOR GEMINI 2.0 FLASH IN THE RESTRICTED TEST

Score Interval	Actual	Predicted	Vector-Based
0–2	37	6	10
3–5	1000	198	740
6–8	2634	3320	2505
9–10	788	935	1204

Table XIII summarizes the evaluation metrics. Direct predictions achieved precision of 0.44, recall of 0.46, and an F1-score of 0.48. Vector-based scores showed modest improvements in recall (0.49) and F1 (0.51), though precision dropped slightly to 0.43. Compared to the extended test, these results indicate a small decline across all metrics, but the degradation remained marginal.

TABLE XIII
EVALUATION METRICS FOR GEMINI 2.0 FLASH IN THE RESTRICTED TEST

Metric	Predicted	Vector-Based
Precision	0.44	0.43
Recall	0.46	0.49
F1-score	0.48	0.51

In summary, the restrictive prompt had only a limited negative impact on *Gemini 2.0 Flash*. Despite a slight decrease in accuracy and F1-score compared to the extended scenario, Gemini consistently outperformed both *ChatGPT-4o-mini* and *Deepseek Chat*, maintaining the highest overall reliability and confirming its robustness under restrictive instruction settings.

3) Deepseek Chat: : With the restricted prompt, *Deepseek Chat* completed the evaluation in 7h 52m 40s at a cost of $0.33. A total of five malformed vectors were recorded, slightly higher than in the extended test. As in previous scenarios, the model displayed a strong bias toward overestimating severity, frequently upgrading *Medium*-level vulnerabilities to the *Critical* category.

Table XIV shows the distribution of predicted and vector-based scores compared with the actual reference values. Direct predictions significantly overrepresented the *Critical* class (1483 vs. 788 actual), while underrepresenting both the *Low* (125 vs. 1000) and *Info/None* categories (1 vs. 37). Vector-based recalculations partially corrected the imbalance by recovering 645 vulnerabilities in the *Low* category and 1398 in the *Critical [9-10]* range, but systematic overestimation persisted.

TABLE XIV
DISTRIBUTION OF PREDICTED AND VECTOR-BASED SCORES FOR DEEPSEEK CHAT IN THE RESTRICTED TEST

Score Interval	Actual	Predicted	Vector-Based
0–2	37	1	6
3–5	1000	125	645
6–8	2634	2850	2410
9–10	788	1483	1398

The evaluation metrics in Table XV confirm this pattern. Predicted scores achieved a precision of 0.36, recall of 0.42, and F1-score of 0.36, while vector-based results slightly improved recall (0.43) and F1 (0.38) but reduced precision (0.34). These values are broadly consistent with the baseline test, indicating that the restrictive prompt did not meaningfully alter Deepseek's performance profile.

TABLE XV
EVALUATION METRICS FOR DEEPSEEK CHAT IN THE RESTRICTED TEST

Metric	Predicted	Vector-Based
Precision	0.36	0.34
Recall	0.42	0.43
F1-score	0.36	0.38

In summary, the restrictive prompt had little impact on *Deepseek Chat*. The model continued to suffer from systematic severity overestimation and occasional erroneous zero assignments for critical vulnerabilities. Although recall remained relatively high, the long runtime, increased number of malformed vectors, and low precision limit the model's practical applicability in automated vulnerability assessment workflows.

IV. CONCLUSION

The experimental results clearly indicate that the current generation of chatbot-based Large Language Models cannot yet function as fully autonomous experts for vulnerability severity assessment. While the evaluated models demonstrated the ability to generate syntactically valid CVSS 3.1 vectors in most cases, significant discrepancies with the reference values were observed. The nature of these discrepancies was model-dependent. Specifically, *ChatGPT-4o-mini* exhibited a strong bias towards clustering predictions in the medium range (5-8), whereas *Gemini 2.0 Flash* and *Deepseek Chat* systematically overestimated vulnerability severity.

Although *ChatGPT-4o-mini* produced syntactically valid vectors, recalculated scores rarely aligned with either the ground-truth values or the model's own direct predictions, confirming the problem of logically inconsistent vectors. In contrast, *Gemini 2.0 Flash* and *Deepseek Chat* produced outputs with greater internal consistency, with *Gemini 2.0 Flash* consistently outperforming the other models in terms of accuracy and F1-score across all experimental settings.

An analysis of operational parameters revealed a marked advantage of *Gemini 2.0 Flash*, which completed tasks in the shortest time and at the lowest cost. Conversely, *Deepseek Chat* was the slowest and most expensive model, though it achieved relatively high classification recall for critical vulnerabilities. Across eight of the nine trials, recalculated scores from generated vectors proved marginally closer to the reference values than the direct predictions; however, these differences were not substantial.

The restrictive prompt, designed to prevent models from leveraging external knowledge, did not have a statistically significant impact on the distribution of results. Only a marginal increase in errors and a slight decrease in performance metrics were observed, well within the margin of measurement uncertainty. This outcome suggests that current LLMs cannot meaningfully distinguish between training knowledge and external information, and such restrictions become relevant only in the context of "reasoning models" that explicitly search external sources with controllable scope.

The extended prompt, which introduced explicit definitions of CVSS vector components, had heterogeneous effects across models. A notable improvement in classification quality was observed only in the case of *Gemini 2.0 Flash*, which achieved the highest F1 scores in all nine trials. For both *ChatGPT-4o-mini* and *Deepseek Chat*, this modification had a regressive effect, reducing classification accuracy. Nevertheless, the inclusion of explicit vector structure completely eliminated the few syntactic errors observed in the baseline test.

One of the most critical issues identified was the erroneous assignment of a CVSS score of 0.0 to vulnerabilities of high or critical severity, a problem particularly evident in the outputs of *Deepseek Chat*. Such errors pose a serious operational risk if chatbot-generated results were to be applied without validation in production vulnerability management systems. This finding underlines the necessity of independent verification mechanisms when integrating LLM-based automation into vulnerability assessment workflows.

Overall, the results obtained show that while chatbot systems show promise in supporting human analysts-especially in terms of reducing manual workload-they cannot currently replace expert-driven vulnerability scoring. Their practical applicability lies in semi-automated scenarios, where LLM-generated results are subjected to systematic validation and cross-checking.

Future research should extend this work in several directions:

- Evaluation of next-generation LLMs with improved reasoning capabilities.
- Replication of experiments using "reasoning models" that support explicit and controllable external knowledge retrieval.

- Migration of the evaluation framework to CVSS 4.0 once a representative number of vulnerabilities has been scored in the new standard.
- Extension of analysis beyond base metrics to include Temporal and Environmental CVSS components.

Finally, this study shows that while chatbots powered by LLMs hold promise as supportive tools for vulnerability management, they remain unsuitable as autonomous decision-makers. Their integration into security workflows should therefore focus on augmenting, rather than replacing, human expertise.

ACKNOWLEDGMENT

The authors would like to thank Wroclaw University of Science and Technology (statutory activity) and Military University of Technology for financial support. During the preparation of this manuscript, the authors used ChatGPT for text editing and grammar correction. The authors have carefully reviewed and edited the output and take full responsibility for the content of this publication.

REFERENCES

[1] C. I. Erondu and U. I. Erondu, "The role of cyber security in a digitalizing economy: A development perspective," *International journal of research and innovation in social science*, vol. 7, no. 11, pp. 1558–1570, 2023.

[2] A. Juneja, S. S. Goswami, and S. Mondal, "Cyber security and digital economy: opportunities, growth and challenges," *Journal of technology innovations and energy*, vol. 3, no. 2, pp. 1–22, 2024.

[3] "National vulnerability database," 2025, accessed: 28 August 2025. [Online]. Available: http://nvd.nist.gov/

[4] J. Spring, E. Hatleback, A. Manion, and D. Shic, "Towards improving cvss," *SEI, CMU, Tech. Rep*, 2018.

[5] L. Allodi and F. Massacci, "Comparing vulnerability severity and exploits using case-control studies," *ACM Transactions on Information and System Security (TISSEC)*, vol. 17, no. 1, pp. 1–20, 2014.

[6] FIRST, "Common vulnerability scoring system v4.0: Specification document," 2023, accessed: 28 August 2025. [Online]. Available: https://www.first.org/cvss/specification-document

[7] A. Balsam, M. Nowak, M. Walkowski, J. Oko, and S. Sujecki, "Comprehensive comparison between versions cvss v2. 0, cvss v3. x and cvss v4. 0 as vulnerability severity measures," in *2024 24th International Conference on Transparent Optical Networks (ICTON)*. IEEE, 2024, pp. 1–4.

[8] M. Nowak, M. Walkowski, and S. Sujecki, "Conversion of cvss base score from 2.0 to 3.1," in *2021 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. IEEE, 2021, pp. 1–3.

[9] M. R. Nowak, M. Walkowski, and S. Sujecki, "Support for the vulnerability management process using conversion cvss base score 2.0 to 3. x," *Sensors*, vol. 23, no. 4, p. 1802, 2023.

[10] M. Walkowski, M. Krakowiak, M. Jaroszewski, J. Oko, and S. Sujecki, "Automatic cvss-based vulnerability prioritization and response with context information," in *2021 international conference on software, telecommunications and computer networks (softCOM)*. IEEE, 2021, pp. 1–6.

[11] F. A. Bhuiyan, A. Rahman, and P. Morrison, "Vulnerability discovery strategies used in software projects," in *Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering*, 2020, pp. 13–18.

[12] Qualys, "Vulnerability management tool," 2025, accessed: 28 August 2025. [Online]. Available: https://www.qualys.com/apps/vulnerability-management/

[13] WithSecure, "Vulnerability management tool," 2025, accessed: 28 August 2025. [Online]. Available: https://www.withsecure.com/userguides/product.html?business/radar

[14] Rapid7, "Vulnerability management tool," 2021, accessed: 28 August 2025. [Online]. Available: https://www.rapid7.com/products/nexpose/

[15] Tenable, "Vulnerability management tool," 2025, accessed: 28 August 2025. [Online]. Available: https://www.tenable.com/products/tenable-io

[16] A. Balsam, M. Walkowski, M. Nowak, J. Oko, and S. Sujecki, "Automatic cvss-based vulnerability prioritization and response with context information and machine learning." *Applied Sciences (2076-3417)*, vol. 15, no. 16, 2025.

[17] M. R. Shahid and H. Debar, "Cvss-bert: Explainable natural language processing to determine the severity of a computer security vulnerability from its description," in *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2021, pp. 1600–1607.

[18] H. Kekül, B. Ergen, and H. Arslan, "Estimating missing security vectors in nvd database security reports," *International Journal of Engineering and Manufacturing*, vol. 12, no. 3, p. 1, 2022.

[19] T. Teubner, C. M. Flath, C. Weinhardt, W. Van Der Aalst, and O. Hinz, "Welcome to the era of chatgpt et al. the prospects of large language models," *Business & Information Systems Engineering*, vol. 65, no. 2, pp. 95–101, 2023.

[20] A. Costin, H. Turtiainen, N. Yousefnezhad, V. Bogulean, and T. Hämäläinen, "Evaluating zero-shot chatgpt performance on predicting cve data from vulnerability descriptions," in *Proceedings of the European Conference on Cyber Warfare and Security*, no. 1. Academic Conferences International Ltd, 2024.

[21] K. McClanahan and S. Elder, "When chatgpt meets vulnerability management: the good, the bad, and the ugly," in *International Conference on Computing, Networking and Communications (ICNC 2024)*. IEEE, 2024.

[22] F. Marchiori, D. Donadel, and M. Conti, "Can llms classify cves? investigating llms capabilities in computing cvss vectors," *arXiv preprint arXiv:2504.10713*, 2025.

[23] P. Liu, J. Liu, L. Fu, K. Lu, Y. Xia, X. Zhang, W. Chen, H. Weng, S. Ji, and W. Wang, "Exploring {ChatGPT's} capabilities on vulnerability management," in *33rd USENIX Security Symposium (USENIX Security 24)*, 2024, pp. 811–828.

[24] S. Chopra, H. Ahmad, D. Goel, and C. Szabo, "Chatnvd: Advancing cybersecurity vulnerability assessment with large language models," *arXiv preprint arXiv:2412.04756*, 2024.

[25] OpenAI, "Api pricing," 2025, accessed: 18 May 2025. [Online]. Available: https://openai.com/api/pricing/

[26] Google, "Gemini api pricing," 2025, accessed: 18 May 2025. [Online]. Available: https://ai.google.dev/gemini-api/docs/pricing

[27] DeepSeek, "Api pricing," 2025, accessed: 18 May 2025. [Online]. Available: https://api-docs.deepseek.com/quick_start/pricing

[28] X.ai, "Grok models documentation," 2025, accessed: 18 May 2025. [Online]. Available: https://docs.x.ai/docs/models

[29] Y. Geng, H. Li, H. Mu, X. Han, T. Baldwin, O. Abend, E. Hovy, and L. Frermann, "Control illusion: The failure of instruction hierarchies in large language models," *arXiv preprint arXiv:2502.15851*, 2025.

[30] A. Costin, H. Turtiainen, N. Yousefnezhad, V. Bogulean, and T. Hämäläinen, "Evaluating zero-shot chatgpt performance on predicting cve data from vulnerability descriptions," in *Proceedings of the European Conference on Cyber Warfare and Security*, no. 1. Academic Conferences International Ltd, 2024.