Evaluating Zero-Shot Chatgpt Performance on Predicting CVE Data From Vulnerability Descriptions

Hannu Turtiainen[1], Narges Yousefnezhad[2], Vadim Bogulean[2], Andrei Costin[1], and Timo Hämäläinen[1]

[1] Faculty of Information Technology, University of Jyväskylä, Jyväskylä, Finland

[2] Binare Oy, Jyväskylä, Finland

hannu.ht.turtiainen@jyu.fi
narges.yousefnezhad@binare.io
vadim.bogulean@binare.io
andrei.costin@jyu.fi
timoh@jyu.fi

Abstract: Vulnerability management is a critical industry activity driven by compliance and regulations aiming to allocate best-fitted resources to address vulnerabilities efficiently. The increasing number of vulnerabilities reported and discovered by a diverse community results in varying quality of the reports and differing perspectives. To tackle this, machine learning (ML) has shown promise in automating vulnerability assessments. While some existing ML approaches have demonstrated feasibility, there is room for improvement. Additionally, gaps remain in the literature to understand how the specific terminology used in vulnerability databases and reports influences ML interpretation. Large Language Model (LLM) systems, such as ChatGPT, are praised for their versatility and high applicability to any domain. However, how well or poorly a state-of-the-art LLM system performs on existing vulnerability datasets at a large scale and across different scoring metrics needs to be clarified or well-researched. This paper aims to close several such gaps and present a more precise and comprehensive picture of how ChatGPT performs on predicting vulnerability metrics based on NVD's CVE vulnerability database. We analyze the responses from ChatGPT on a set of 113,228 (~50% out of all NVD vulnerabilities) CVE vulnerability descriptions and measure its performance against NVD-CVE as ground truth. We measure and analyze the predictions for several vulnerabilities in metadata and calculate performance statistics.

Keywords: AI, ChatGPT, CVE, ML, NVD, Vulnerability Management

1. Introduction

Gathering vulnerability information is crucial for cybersecurity as more attacks occur yearly. Automation is required as the attacks have also become automated; thus, a robust, up-to-date, and information-rich database is mandatory for automated tools to function. The National Vulnerability Database (NVD) is the de facto database in the industry, and it does an avid job of disseminating and cataloging information about software vulnerabilities. The more common knowledge about security vulnerabilities, the greater the chance of mitigating potential security risks. The rapid evolution of the cybersecurity landscape has launched demands for robust vulnerability tracking systems; thus, the ground truth data for these systems must be as good as possible.

NVD CVE database is generally of high quality and acts as a ground truth in our (and many others) studies. However, it still suffers from inherent data quality issues, as highlighted by recent works by Anwar et al. (2022), Dong et al. (2019), and Kuehn et al. (2021), and this may indirectly affect any studies and comparisons to where it is used as a ground truth. At the same time, to the best of our knowledge, there is no better and more curated ground truth for vulnerability information than NVD CVE.

Large language models (LLM), such as GPT-4 by OpenAI, can sift through large text datasets quickly and efficiently. By leveraging machine learning algorithms, LLMs can analyze patterns and extract insights from various sources, contributing to a more comprehensive and accurate vulnerability database or accelerating the attempts to improve the database manually. Despite the potential benefits, it is essential to approach the integration of LLMs cautiously. However, it should be acknowledged that the quality of LLM results can vary, as inaccuracies, biases, limited data, and data poisoning during the training may impact the model's performance. Therefore, it is crucial to implement mechanisms for validation and verification to ensure that the information provided by LLMs aligns with established cybersecurity standards and experts' opinions. Enhancing the quality of the NVD requires effort. Incorporating LLMs to harness the power of advanced language processing may be the way forward for the NVD dataset.

Our primary focus is to address critical gaps and comprehensively assess ChatGPT's performance using NVD's widely recognized CVE vulnerability database. In our experiment, we test how well ChatGPT predicts Common Weakness Enumeration (CWE), Common Vulnerability Scoring System (CVSS), and Common Platform

Enumeration (CPE) based on the vulnerability description from the NVD database. We compare the predictions against the ground truth from the NVD database.

In summary, we attempt to bridge some research field gaps and provide insights into qualitative and quantitative performance metrics of ChatGPT when applied to CVE and vulnerability reporting. To the best of our knowledge, this work is the first of its kind and of this scale of ChatGPT applied to CVEs.

The findings indicate that ChatGPT could have performed more robustly in certain aspects, raising questions about its suitability for fully automated and autonomous AI-based vulnerability management systems. As we discuss the challenges inherent in this research, we underscore the importance of addressing these limitations.

Our main contributions to this work are as follows:

1. We collect and evaluate ChatGPT GPT-4 qualitative and quantitative performance metrics over 33,197 NVD CVE entries.
2. We process the results using multi-metrics analysis.
3. We provide key insights and takeaways on what it means for automated cybersecurity and where ChatGPT's blind spots for CVEs.
4. We release the finalized dataset as open data under FAIR Data Principles and permissive licensing.

2. Experiment and Data Setup

In this section, we briefly present our experimental and data setup. The main components of our experiments are NVD CVE recent database subset (Section 2.1), ChatGPT queries and responses (Section 2.2), and statistical metrics for qualitative performance analysis (Section 2.3).

2.1 CVE Dataset Setup

The NVD CVE database is the standard vulnerability database used by organizations and individuals worldwide. It is also open-source and free of cost. To fetch the latest and complete NVD CVE database, we employed FastCVE by "Binaré Oy"' (2023). FastCVE is an open-source, highly reliable, and efficient alternative to well-known cve-search, which did not support the latest NVD API usage shift when our experiments started.

The official NVD CVE database contains 222,982 entries as of the submission of this paper. For our NVD dataset snapshot, we queried FastCVE for "7,000 random entries" for each year between 2002–2023 and limited the total to 115,000. The choice of 7,000 entries/year comes from practical considerations such as budgeting (e.g., cumulative costs of ChatGPT API queries) and timing (e.g., getting ChatGPT results for a query take up to 60 seconds, queries from the same account cannot be parallelized, ChatGPT service downtime). Since not all earlier years had at least 7,000 CVE entries and we removed "rejected" entries before submitting them, we ended up with 113,228 entries to query ChatGPT, more than 50% of the entire NVD CVE database.

However, not all 113,228 entries are fully specified in the NVD CVE database. For example, 32,616 entries lacked a proper and relevant CWE value. Instead, they had "NVD-CWE-Other" or "NVD-CWE-noinfo", which we did not match against the ChatGPT data. Moreover, "NVD-CWE-Other" or "NVD CWE-noinfo" are generally useless noise as they are irrelevant for any meaningful comparison, as they convey no additional information compared to entries missing CWE altogether. Also, entries typically have either CVSS v2, v3, or v3.1 information and not necessarily more than one of them. However, there are some exceptions, of course. Nevertheless, we ended up with 107,378 entries with CVSS v2 score, 47,200 entries with CVSS v3 (i.e., v3.0 and v3.1 combined) metrics, and 80,612 with properly assigned CWE.

2.2 ChatGPT Setup

For this experiment, we have enabled a paid ChatGPT-4 API key. All results were served by the ChatGPT model version gpt-4-0613. In API query terms, we set ChatGPT with the following prompt:

```
messages = [
        {
        "role": "system",
        "content": "You are a cybersecurity expert. Do not look up any vulnerability databases such as NVD.
        You will be provided with a bug description. Provide a fitting CWE, CVSS version 2, and version 3 with
```

exact numbers and EPSS score as a response without any other text. Please also try to give a matching CPE to this bug. Parse the result as a JSON.",

},

{"role": "user", "content": prompt},

]

Subsequently, we iterated over each CVE entry in our snapshot dataset and supplied the CVE description text and the prompt above to the ChatGPT APIs. Naturally, we collected and stored the response for each query for later offline analysis and processing.

2.3 Statistical Metrics

We rely on statistical metrics to compare the qualitative performance of ChatGPT versus the NVD CVE database (NVD as ground truth). Since these metrics are tailored to evaluate distinct aspects of the data-model relationship Hodson (2022), narrowing the comparison to only one or two metrics diminishes the breadth of insights that can be extracted from the analysis. Such limitations on several metrics may constrain the potential discoveries derived from modeling studies. As recommended by several studies (e.g., Jackson et al. (2019)), employing a diverse set of metrics enables acquiring a broader range of insights and understanding of the data relation and observable phenomena Liemohn et al. (2021). We employed core Python, scikit-learn Pedregosa et al. (2011), and NumPy Harris et al. (2020) libraries for all our metric calculations.

– MAE (Mean Absolute Error) measures the average skew of the predictions to the actual values using the same units and dimensions as the original values.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|GT_i - Pred_i|$$

– MSE (Mean Squared Error) measures average error but penalizes more significant errors than smaller ones. It uses squared units and squared dimensions of the target variables.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(GT_i - Pred_i)^2$$

– RMSE (Root Mean Squared Error) RMSE (Root Mean Squared Error) measures the average error of the predictions, but it penalizes significant errors more than smaller ones. It uses the same units and dimensions as the original values.

$$RMSE = \sqrt{MSE}$$

– MAPE (Mean Absolute Percentage Error) provides the average percentage error of the predictions as a relative measure.

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{|GT_i - Pred_i|}{GT_i}\right) \times 100$$

– R-squared (Coefficient of Determination), in our case, is a metric to see how well ChatGPT predicts the required values (dependent variable) based on the vulnerability description (feature vectors from the text are independent variable). In the context of model performance, R-squared has its limitations and should be considered alongside other evaluation metrics.

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(GT_i - Pred_i)^2}{\sum_{i=1}^{N}(GT_i - MeanGT)^2}$$

– The Pearson Correlation Coefficient measures the linear relationship between the ground truth and the predictions. A score of one means perfect positive correlation, and zero indicates no linear correlation.

$$Correlation = \frac{Covariance(GT, Pred)}{StdDev(GT) \times StdDev(Pred)}$$

Together, these metrics provide a comprehensive evaluation of a model's performance from different perspectives, and each metric comes with its own set of advantages and disadvantages Mamun et al. (2020). MAE, RMSE, and MAPE are generally more interpretable and easy to communicate to non-technical stakeholders, making them suitable for practical applications. Pearson Correlation measures the association between variables, while MAE, MSE, RMSE, and MAPE quantify the error between predicted and actual values. Pearson Correlation is not sensitive to outliers, while MAE and MAPE are less sensitive than MSE and RMSE. R-squared provides insights into the model's overall goodness of fit, with higher values indicating better explanatory power Chicco et al. (2021). Researchers and analysts often use a combination of these metrics to understand a model's strengths and weaknesses comprehensively.

3. Results and Analysis

In this section, we present our main results and their analysis. Figures 1 and 2 summarize the ground truth data from the NVD database for CVSS v2 and v3, respectively. These Figures show that not all entries have all metrics assigned, which is a problem in data quality. Of course, older entries do not have the newer CVSSv3 metrics, but all should have at least one or the other. Figures 3 and 4 disclose the results from ChatGPT predictions for CVSS. We also highlight the number of entries that have both ground truth and prediction data available. Figure 5 showcases the prediction results for CWE and CPE data. We analyze and interpret the statistics of ChatGPT's prediction of CVSSv2, CVSSv3, CWE, and CPE separately and independently below. Figure 6 includes the results of our statistics, explained in Section 2.3, for CVSSv2 and CVSSv3 score, severity, and vector values.

	All NVD entries	**with** CVSSv2 Score	without CVSSv2 Score	**with** CVSSv2 Severity	without CVSSv2 Severity	**with** CVSSv2 Vector	without CVSSv2 Vector
Count (%)	113,228 (100%)	107,378 (95%)	5,850 (5%)	107,381 (95%)	5,847 (5%)	107,381 (95%)	5,847 (5%)
Total	113,228	113,228		113,228		113,228	

Figure 1: NVD: CVSSv2 ground truth

	All NVD entries	**with** CVSSv3 Score	without CVSSv3 Score	**with** CVSSv3 Severity	without CVSSv3 Severity	**with** CVSSv3 Vector	without CVSSv3 Vector
Count (%)	113,228 (100%)	47,200 (42%)	66,028 (58%)	47,200 (42%)	66,028 (58%)	47,200 (42%)	66,028 (58%)
Total	113,228	113,228		113,228		113,228	

Figure 2: NVD: CVSSv3 ground truth

	GPT and NVD Both CVSSv2 Score	GPT **true** CVSSv2 Score	GPT false CVSSv2 Score	GPT and NVD Both CVSSv2 Severity	GPT **true** CVSSv2 Severity	GPT false CVSSv2 Severity	GPT and NVD Both CVSSv2 Vector	GPT **true** CVSSv2 Vector	GPT false CVSSv2 Vector
Count (%)	103,703 (92% of our NVD)	35,996 (35%)	67,707 (65%)	103,705 (92%)	62,782 (61%)	40,923 (39%)	52,656 (49%)	23,151(44%)	29,505 (56%)
Total	103,703	103,703		103,705	103,705		52,656	52,656	

Figure 3: GPT vs. NVD: CVSSv2 results

	GPT and NVD Both CVSSv3 Score	GPT **true** CVSSv3 Score	GPT false CVSSv3 Score	GPT and NVD Both CVSSv3 Severity	GPT **true** CVSSv3 Severity	GPT false CVSSv3 Severity	GPT and NVD Both CVSSv3 Vector	GPT **true** CVSSv3 Vector	GPT false CVSSv3 Vector
Count (%)	44,401 (39% of our NVD)	17,494 (39%)	26,907 (61%)	44,395 (39%)	28,697 (65%)	15,698 (35%)	18,550 (16%)	3,645(20%)	14,905 (80%)
Total	44,401	44,401		44,395	44,395		18,550	18,550	

Figure 4: GPT vs. NVD: CVSSv3 results

	All NVD entries	NVD **with** CWE	NVD without CWE (or CWE-OTHER)	GPT and NVD both CWE	GPT **true** CWE	GPT false CWE	GPT and NVD both CPE	GPT **true** CPE	GPT false CPE
Count (%)	113,228 (100%)	80,612 (71%)	32,616 (29%)	80,612 (71%)	45,326 (56%)	35,286 (44%)	66,204 (58%)	46,886 (71%)	19,318 (29%)
Total	113,228	113,228		80,612	80,612		66,204	66,204	

Figure 5: GPT vs. NVD: CWE and CPE results

	MAE	MAPE	MSE	R^2	RMSE	Pearson's	Total items	Comment
CVSSv2 Score	1.29	28.13	3.35	0.18	1.83	0.60	103,703	Table 3 / Column 2
CVSSv3 Score	0.94	15.08	2.02	0.27	1.42	0.64	44,401	Table 4 / Column 2
CVSSv2 Severity	0.41	25.23	0.45	-0.21	0.67	0.49	103,705	Table 3 / Column 5
CVSSv3 Severity	0.39	16.37	0.45	0.17	0.67	0.61	44,395	Table 4 / Column 5
CVSSv2 Vector-Access	0.11	x	x	x	x	x	52,656	Table 3 / Column 8
CVSSv2 Vector-Authentication	0.07	x	x	x	x	x	52,656	– // –
CVSSv2 Vector-Availability	0.30	x	x	x	x	x	52,656	– // –
CVSSv2 Vector-Complexity	0.30	x	x	x	x	x	52,656	– // –
CVSSv2 Vector-Confidentiality	0.25	x	x	x	x	x	52,656	– // –
CVSSv2 Vector-Integrity	0.23	x	x	x	x	x	52,656	– // –
CVSSv3 Vector-Attack Complexity	0.12	x	x	x	x	x	18,550	Table 4 / Column 8
CVSSv3 Vector-Attack Vector	0.27	x	x	x	x	x	18,550	– // –
CVSSv3 Vector-Availability	0.20	x	x	x	x	x	18,550	– // –
CVSSv3 Vector-Confidentiality	0.21	x	x	x	x	x	18,550	– // –
CVSSv3 Vector-Integrity	0.20	x	x	x	x	x	18,550	– // –
CVSSv3 Vector-Privileges	0.17	x	x	x	x	x	18,550	– // –
CVSSv3 Vector-Scope Changes	0.08	x	x	x	x	x	18,550	– // –
CVSSv3 Vector-User Interaction	0.18	x	x	x	x	x	18,550	– // –

Figure 6: Summary of statistics

For the CVSS scores, we aimed to match the exact score (i.e., 0.0 – 10.0). Only then would the prediction be positive. The severity metric would better indicate "close enough" scores, as all the severity values are a range of scores. We got the severity values based on the scores from the ground truth and the ChatGPT predictions and translated the severities from words to a "value" from 0–2 for CVSSv2 (low, medium, high) and 0–3 for CVSSv3 (low, medium, high, critical), respectively. With these values, we can calculate statistical metrics and errors. For the CVSS vectors, we also checked for an exact match (case we marked as true directly). However, we also studied the accuracy of ChatGPT to predict individual metrics for each scoring system to see which vector metric caused ChatGPT the most issues. We used a "value" system (0...N), similar to the severity statistics above.

The CVSSv2 results (Figure 3) show that ChatGPT predicted the exact score 35% of the time, which is not ideal but surprisingly good because of the exact score requirement (i.e., 101 distinct possible values, 0.0 – 10.0). However, the severity was correct only 61% of the time, which is far from being qualitatively acceptable as only three values in the range in CVSS v2 (low, medium, high). CVSSv2 vectors were matched identically 44% of the time, which seems promising. The base score is calculated from the vector; however, we have far fewer predictions for the vectors than the scores. Thus, at this point, we cannot draw many conclusions on whether ChatGPT has any correlation between the predictions. As for the individual CVSSv2 vector values (Figure 6: CVSSv2 Vector), our statistics would indicate that the *Access Complexity* and *Authentication* metrics are usually correct. These metrics are far more frequently correct than the others, while the *Availability* impact score causes the most problems for ChatGPT.

On average, the CVSSv2 score predictions (Figure 6) were off by 1.29 points (MAE), which is certainly big enough to cause severity to be wrong. CVSS v2 MSE and RMSE were 3.35 and 1.83, respectively, indicating that some prediction errors were way off. The MAPE value of 28.1% means that, on average, the predictions are off by approximately 28.1% in percentage terms, which is relatively high. The R-squared value of 0.18 is ideally not great; however, the Pearson Correlation Coefficient is better at 0.60. Nevertheless, the ChatGPT performs better

than the random mean but does not fit well. For CVSSv2 severity, ChatGPT was off by 0.41 points (MAE) on average, which is quite a decent result. The statistics show that ChatGPT's performance is moderate; however, nowhere near precise enough for cybersecurity applications similar to the test.

With CVSSv3 (Figure 4), considerably less completely valid data was available than with CVSSv2. This is because we have taken entries from each year since 2002, and the effort to add v3 metrics to older entries seems non-existent; thus, NVD-CVE ground truth misses a lot of CVSSv3 data. ChatGPT CVSSv3 predictions were similar to CVSSv2 with a 39% correct score, and the event was slightly better than v2 for the v3 severity at 65% correct predictions. Predicting CVSSv3 vectors proved problematic for ChatGPT, with only 20% being entirely correct. However, v3 has more metrics in the base score vector than v2. Therefore, this lower performance in v3 vector prediction is intuitively expected but discouraging. As for vectors (Figure 6), scope and attack complexity performed the best. This is expected as they only have two values to set. The user interaction metric also has two values but still performs worse. The attack vector exploitability metric was the worst metric from the predictions.

The CVSSv3 score predictions (Figure 6) were off by 0.94 points (MAE), which is considerably better than the v2 scores. MSE and RMSE were 2.02 and 1.42, respectively, significantly better than the v2 statistics, indicating that the false predictions were closer to the ground truth overall. The MAPE value of 15.0% is also almost half of the v2 score. The R-squared value of 0.27 is still relatively low, although also better than v2, while the Pearson Correlation Coefficient was raised to 0.67. Based on the data, **ChatGPT is fitting predictions better for the v3 than the v2 scores**. Nevertheless, even the v3 results are far from acceptable as is. For CVSSv3 severity, ChatGPT was off by 0.41 points (MAE) on average.

The overall better performance for v3 severity over v2 predictions could be interpreted to be because of any of the following reasons: either due to a more extensive, more convergent set of 103,703 (CVSSv2) vs. 44,401 (CVSSv3) or due to better finetune of ChatGPT towards predicting CVSSv3 scores as this CVSS version is being more actively used in reports over the last decade.

For the CPE metrics (Figure 5), we parse the CPE result of ChatGPT. If the CPE is valid, we query FastCVE for CVEs present in that CPE. The result is true if the CVE we are processing is on that list. For CWE metrics, we check if the ground truth and the ChatGPT results match, and we do this only if the ground truth has a valid value, i.e., values that are not NVD-CWE-other nor NVD-CWE-noinfo. The results indicate that ChatGPT produced the same CWE as the ground truth in 56% of the valid cases and a valid-and-correct CPE in 71% of the entries. It is good to note that the CPE predictions also suffered from varying responses and non-valid predictions, as presented in Section 3.1. The results are not exceptional, but a solid foundation for improvements. We must remember that the ground truth descriptions may not always be fully representative as they seem to add more information to the metadata rather than a complete description of the vulnerability. Nevertheless, we leave exploring the quality of the descriptions as future work.

3.1 Samples of Chatgpt Responses Posing Challenges

CVE databases, including NVD, have many quality and completeness issues (Section 2.1). ChatGPT responses are no exception to this rule, despite the intuitive expectation that the platform would reply in a more uniform and standardized manner.

Below, we provide the most commonly seen variations of the output format response from ChatGPT for the requested CVSSv2 information. Such variations pose challenges for automated data processing and data normalization. Therefore, handling and normalizing such responses carefully and on a case-by-case basis is required.

1. "CVSSv2": "AV:N/AC:L/Au:N/C:P/I:P/A:P (7.5)" → "vector (score)"
2. "CVSSv2": "7.5 (AV:N/AC:L/Au:N/C:P/I:P/A:P)" → "score (vector)"
3. "CVSSv2": "AV:N/AC:L/Au:N/C:P/I:P/A:P (7.5 – High)" → "vector (score - severity)
4. "CVSSv2": {"Score": 7.2,"Vector": "AV:L/AC:L/Au:N/C:C/I:C/A:C"} → { "Score": <float>, "Vector": vector}

   NOTE: first-capitalized sub-keys

   NOTE: "Score" as a raw/native FLOAT 7.2

5. "CVSSv2": {"score": 4.3,"vector": "AV:N/AC:M/Au:N/C:N/I:P/A:N"} → { score: score, "vector": vector}

NOTE: lowercase sub-keys

NOTE: "score" as a raw/native FLOAT 7.2

6. "CVSSv2": "AV:N/AC:L/Au:N/C:P/I:N/A:N" → "vector", "CVSSv2_Score": "5.0" → "score"

NOTE: separate main keys

7. "CVSSv2": {"Base Score": "7.8","Vector": "(AV:N/AC:L/Au:N/C:N/I:N/A:C)"} → { "Base Score": STRING, "Vector": vector},

8. "CVSSv2": {"Base Score": 7.8,"Vector": "(AV:N/AC:L/Au:N/C:N/I:N/A:C)"} → { "Base Score": FLOAT, "Vector": vector},

NOTE: "Base Score" as a raw/native <float> 7.8 as opposed to string variant "7.8" in most other cases

9. "CVSSv2": "AV:N/AC:L/Au:N/C:C/I:C/A:C" → "vector"

NOTE: no score

10. "CVSSv2": {"Base Score": "7.5", "Impact Subscore": "6.4", "Exploitability Subscore": "10.0",

"Access Vector": "Network," "Access Complexity": "Low," "Authentication": "None," "Confidentiality Impact": "Partial," "Integrity Impact": "Partial," "Availability Impact": "Partial"} → {"Base Score": score}

NOTE: vector required to be constructed (we do not)

11. "CVSSv2": {"Base Score": "5.0", "Impact Subscore": "2.9", "Exploitability Subscore": "10.0", "Vector": "AV:N/AC:L/Au:N/C:N/I:N/A:P"} → { "Base Score": score, "Vector": vector},

NOTE: more data than expected (sometimes contradictory!)

12. "CVSSv2": "7.5", -> "score"

NOTE: no vector

4. Related Work

Several papers about the applicability of ChatGPT for cybersecurity have been released since the release of ChatGPT in 2023.

Liu et al. (2023) evaluate the performance of ChatGPT in Vulnerability Description Mapping (VDM) tasks and investigate the application of closed-source LLMs in real-world security management scenarios – ChatGPT shows promising results. However, it does not threaten the crucial role of security engineers in the domain of vulnerability analysis. Naito et al. (2023), ChatGPT, when provided with vulnerability-related information and asset management data, can generate an effective high-threat test attack path, including specific vulnerability details like CVEs. Our work builds upon the efforts of Liu et al. (2023) and Naito et al. (2023) by investigating if ChatGPT can enrich the vulnerability databases for a more complete cataloging of known vulnerabilities.

The use of LLMs, such as ChatGPT, has also raised concerns. Kalla & Kuraku (2023) verify the potential benefits of utilizing ChatGPT in the context of cybersecurity by discussing the advantages, disadvantages, and risks associated with ChatGPT and artificial intelligence (AI) in cybersecurity. Hu & Chen (2023) also perform an analysis from a dimensional perspective on the cybersecurity opportunities (e.g., malware protection) and risks (e.g., privacy breaches) associated with ChatGPT-like information systems. Thoughtful consideration of its limitations and associated risks, including limited contextual understanding, generation of misleading information, and malicious exploitation, is essential in its implementation Kalla & Kuraku (2023). Potential misuse of generative AI tools by cyber offenders is also discussed in Gupta et al. (2023), outlining scenarios where adversaries could employ ChatGPT for various cyber attacks, including social engineering, phishing, automated hacking, and developing malware.

Al-Hawawreh et al. (2023) demonstrate how ChatGPT can be exploited to design and execute false data injection attacks on critical infrastructure, such as industrial control systems. Scanlon et al. (2023) evaluate the positive and negative effects of ChatGPT, specifically GPT-4, within the context of digital forensics. Scanlon et al. (2023) experimentally demonstrated that although ChatGPT holds considerable potential in digital forensic investigation, not all outputs from ChatGPT can be considered reliable. Thus, the essential role of human expertise remains crucial in the process. As the quality and accuracy of the data in the vulnerability databases are paramount, the concerns raised by the authors are of concern. The use of LLMs regarding cybersecurity must be explicitly justified.

Another avenue for cybersecurity enhancement by LLMs is to aid programmers by detecting possible vulnerabilities during code creation. Nair et al. (2023) investigated the essential strategies a designer must employ to leverage ChatGPT to produce secure hardware code. To conduct this investigation, the authors instruct ChatGPT to generate code scenarios corresponding to 10 common CWEs within the hardware design (CWE-1194) framework. Espinha Gasiba et al. (2023) conducted a brief experiment involving five distinct vulnerable code snippets derived from C/C++ and sourced from the Sifu platform. In over 60% of the provided code snippets, ChatGPT accurately identified issues in the source code and suggested effective solutions. Pearce et al. (2023) investigate the performance of five commonly used "off-the-shelf" LLMs in real-world security bug scenarios. Their experiments show off-the-shelf models can create security fixes in simple scenarios without extra training but face challenges in real-world situations.

It is clear that LLMs, such as ChatGPT, could help eliminate security vulnerabilities at their source (programming); however, they could also disseminate these bugs if the training data had issues at the time of training. Furthermore, LLMs could help cybersecurity experts find bugs in software and, crucially, help aid in vulnerability database entry creation.

5. Conclusion

To combat cyber threats, disseminating information on known vulnerabilities is crucial; thus, robust and available databases are required with as much metadata as possible for automated tools to function. As of early 2024, the NVD database is still lacking in metadata and suffers from data quality issues. LLMs, such as ChatGPT, are a quick way to enrich the data and, thus, provide more means to combat the threats. Our first-of-its-kind exploration of ChatGPT's performance on the NVD CVE vulnerability database has provided valuable insights into the capabilities and limitations of LLMs in cybersecurity. While ChatGPT exhibited a level of helpfulness, it became apparent that as is, it needs to improve to trust the output rigorously. We have also discussed how to interpret our experiments' results in detail. For example, based on current data, ChatGPT is fitting predictions better for v3 than v2 scores.

At this point, it is clear that LLMs such as ChatGPT cannot fully replace expert-based efforts such as NVD-CVE. And we do not recommend it’s use on this type of cyberseurity work without rigorous scrutiny of the output. Perhaps more research is required in the area of Small Language Models (SMLs) that are tailored as Domain Specific Languages (DSLs), for example, in areas such as technical cybersecurity, and we leave such explorations as immediate future work.

Acknowledgements

Hannu Turtiainen thanks the Finnish Cultural Foundation / Suomen Kulttuurirahasto (www.skr.fi) for supporting his Ph.D. dissertation work and research (grant decision no. 00231412).

(Part of) This work was supported by the European Commission under the Horizon Europe Programme, as part of the project LAZARUS (https://lazarus-he.eu/) (Grant Agreement no. 101070303). The content of this article does not reflect the official opinion of the European Union. Responsibility for the information and views expressed therein lies entirely with the authors.

References

Al-Hawawreh, M. et al. (2023). 'Chatgpt for cybersecurity: practical applications, challenges, and future directions,' Cluster Computing.

Anwar, A. et al. (2022). 'Cleaning the NVD: comprehensive quality assessment, improvements, and analyses', Transactions on Dependable and Secure Computing.

"Binaré Oy"' (2023), 'FastCVE - fast, rich, API-based search for CVE and more. (CPE, CWE, CAPEC)', https://github.com/binareio/FastCVE.

Chen, Y. et al. (2023). Diversevul: A new vulnerable source code dataset for deep learning based vulnerability detection, in '26th International Symposium on Research in Attacks, Intrusions and Defenses', ACM.

Chicco, D. et al. (2021). 'The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation', PeerJ Computer Science.

Dasbach-Prisk, A. et al. (2023). Sensorloader: Bridging the gap in cyber-physical reverse engineering across embedded peripheral devices, in '1st International Workshop on Security and Privacy of Sensing Systems'.

Dong, Y. et al. (2019). Towards the detection of inconsistencies in public security vulnerability reports, in '28th USENIX Security Symposium', USENIX Association.

Espinha Gasiba, T. et al. (2023). I'm sorry dave, I'm afraid I can't fix your code: On chatgpt, cybersecurity, and secure coding, in '4th International Computer Program ming Education Conference'.
Gadyatskaya, O. & Papuc, D. (2023). Chatgpt knows your attacks: Synthesizing at tack trees using llms, in 'International Conference on Data Science and Artificial Intelligence,' Springer.
Gupta, M. et al. (2023). 'From chatgpt to threatgpt: Impact of generative AI in cyber security and privacy,' IEEE Access.
Harris, C. R. et al. (2020). 'Array programming with NumPy,' Nature. Hodson, T. O. (2022), 'Root-mean-square error (rmse) or mean absolute error (mae): When to use them or not,' Geoscientific Model Development.
Hu, C. & Chen, J. (2023). A dimensional perspective analysis on the cybersecurity risks and opportunities of chatgpt-like information systems, in 'International Conference on Networking and Network Applications', IEEE.
Jackson, E. K. et al. (2019). 'Introductory overview: Error metrics for hydrologic modmodeling - A review of common practices and an open source library to facilitate use and adoption', Environmental Modelling & Software.
Kalla, D. & Kuraku, S. (2023). 'Advantages, disadvantages and risks associated with chatgpt and ai on cybersecurity,ä Journal of Emerging Technologies and Innovative Research.
Kuehn, P. et al. (2021). OVANA: an approach to analyze and improve the information quality of vulnerability databases, in D. Reinhardt & T. Müller, eds, '16th International Conference on Availability, Reliability and Security', ACM.
Liemohn, M. W. et al. (2021). 'Rmse is not enough: Guidelines to robust data model comparisons for magnetospheric physics,' Journal of Atmospheric and Solar Terrestrial Physics.
Liu, X. et al. (2023). Not the end of story: An evaluation of chatgpt-driven vulnerability description mappings, in 'Findings of the Association for Computational Linguistics: ACL,' Association for Computational Linguistics.
Mamun, A. A. et al. (2020). 'A comprehensive review of the load forecasting techniques using single and hybrid predictive models,' IEEE Access.
Nair, M. et al. (2023). How hardened is your hardware? guiding chatgpt to generate secure hardware resistant to cwes, in 'Cyber Security, Cryptology, and Machine Learning - 7th International Symposium', Springer.
Naito, T. et al. (2023). Llm-based attack scenarios generator with IT asset management and vulnerability information, in '6th International Conference on Signal Processing and Information Security', IEEE.
Ozturk, O. S. et al. (2023). New tricks to old codes: Can AI chatbots replace static code analysis tools?, in 'European Interdisciplinary Cybersecurity Conference,' ACM.
Pearce, H. et al. (2023). Examining zero-shot vulnerability repair with large language models, in '44th IEEE Symposium on Security and Privacy', IEEE.
Pedregosa, F. et al. (2011), ‘Scikit-learn: Machine learning in Python’, Journal of Ma chine Learning Research.
Scanlon, M. et al. (2023). 'Chatgpt for digital forensic investigation: The good, the bad, and the unknown,' Forensic Science International: Digital Investigation.
Szabó, Z. & Bilicki, V. (2023). 'A new approach to web application security: Utilizing GPT language models for source code inspection,' Future Internet.
Tihanyi, N. et al. (2023). The formai dataset: Generative AI in software security through the lens of formal verification, in '19th International Conference on Predictive Models and Data Analytics in Software Engineering', ACM.
Waskito, S. A. O. et al. (2023). Otter: Simplifying embedded sensor data collection and analysis using large language models, in '29th Annual International Conference on Mobile Computing and Networking', ACM.