CTIBench: A Benchmark for Evaluating LLMs in Cyber Threat Intelligence

Md Tanvirul Alam
Rochester Institute of Technology
Rochester, NY, USA

Dipkamal Bhusal
Rochester Institute of Technology
Rochester, NY, USA

Le Nguyen
Rochester Institute of Technology
Rochester, NY, USA

Nidhi Rastogi
Rochester Institute of Technology
Rochester, NY, USA

Abstract

Cyber threat intelligence (CTI) is crucial in today's cybersecurity landscape, providing essential insights to understand and mitigate the ever-evolving cyber threats. The recent rise of Large Language Models (LLMs) have shown potential in this domain, but concerns about their reliability, accuracy, and hallucinations persist. While existing benchmarks provide general evaluations of LLMs, there are no benchmarks that address the practical and applied aspects of CTI-specific tasks. To bridge this gap, we introduce CTIBench, a benchmark designed to assess LLMs' performance in CTI applications. CTIBench includes multiple datasets focused on evaluating knowledge acquired by LLMs in the cyber-threat landscape. Our evaluation of several state-of-the-art models on these tasks provides insights into their strengths and weaknesses in CTI contexts, contributing to a better understanding of LLM capabilities in CTI. Code and dataset available at `https://github.com/xashru/cti-bench`.

1 Introduction

The evolving landscape of the digital world has led to an unprecedented growth in cyber attacks, posing significant challenges for many organizations. Cyber Threat Intelligence (CTI), which involves the collection, analysis, and dissemination of information about potential or current threats to an organization's cyber systems [35], can provide actionable insights to help organizations defend against these attacks. Large Language Models (LLMs) have the potential to revolutionize the field of CTI by enhancing the ability to process and analyze vast amounts of unstructured threat and attack data; allowing security analysts to utilize more intelligence sources than ever before. However, LLMs are prone to hallucinations [52] and misunderstandings of text, especially in specific technical domains, that can lead to a lack of truthfulness from the model [34]. This necessitates the careful consideration of using LLMs in CTI as their limitations can lead to them producing false or unreliable intelligence which could be disastrous if used to address real cyber threats.

The lack of proper benchmark tasks and datasets to evaluate LLM capabilities in CTI leaves their reliability and usefulness an open research question. Without standardized benchmarks, it is difficult to objectively measure and compare how effectively LLMs handle CTI tasks and generally understand the domain. General benchmarks like GLUE [51], SuperGLUE [50], MMLU [26], HELM [31], KOLA [53], and various domain-specific benchmarks [25, 54, 55] provide datasets and frameworks for evaluating LLMs in terms of general language understanding or domain-specific capabilities. However, these benchmarks fail to capture the practical aspects of cybersecurity. Limited works on LLM evaluations in cybersecurity [14, 15, 30, 32, 33] have primarily catered to specific cybersecurity

Preprint. Under review.

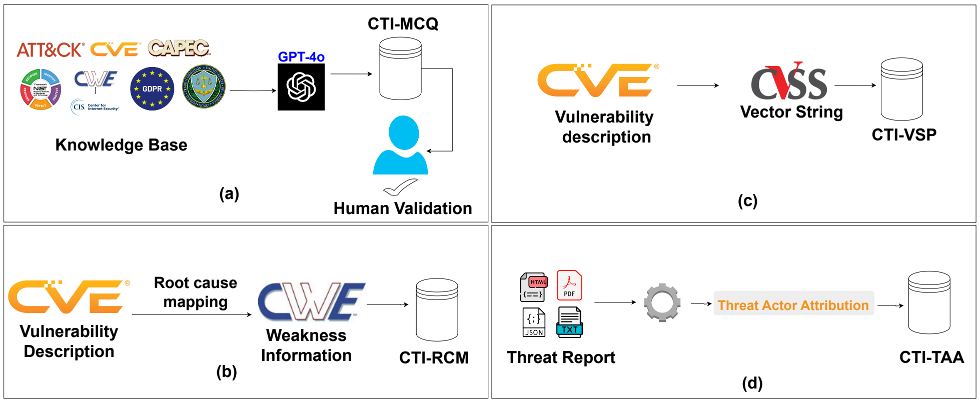


Figure 1: Overview of CTIBench.

industry or, focused on designing tasks that evaluate the memorization ability of LLMs, failing to capture the comprehension and problem-solving capabilities in the broad domain of CTI.

Addressing this gap, we propose **CTIBench**, a novel suite of benchmark tasks and datasets to evaluate LLMs in cyber threat intelligence. To this end, we first create a knowledge evaluation dataset (*CTI-MCQ*) with multiple-choice questions to assess the LLMs' understanding of CTI standards, threats, detection strategies, mitigation plans, and best practices. We utilize various authoritative sources and standards within CTI domain such as NIST [28], MITRE [9], GDPR [1] to craft this dataset. Next, we propose two practical CTI tasks that evaluate LLMs' reasoning and problem-solving skills: mapping Common Vulnerabilities and Exposures (CVE) descriptions [39] to Common Weakness Enumeration (CWE) categories [40] (*CTI-RCM*) and calculating the Common Vulnerability Scoring System (CVSS) scores (*CTI-VSP*). Both tasks assess LLMs' ability to understand and evaluate cyber threats and vulnerabilities. Additionally, *we create a task* requiring LLMs to analyze publicly available threat reports and attribute the reports to specific threat actors or malware families (*CTI-TAA*). This task necessitates a thorough understanding of how different cyber threats or malware families have behaved in the past to identify meaningful correlations. Together, these tasks provide a robust framework for assessing LLMs in CTI. Figure 1 provides an overview of our benchmark. We *evaluate* five state-of-the-art LLMs—three commercial and two open-source—on these tasks. Our results and analysis provide important insights into the LLMs' capabilities in CTI analysis and highlight future avenues for research. We make the datasets and code publicly available.

Through CTIBench, we provide the research community with a robust tool to accelerate incident response by automating the triage and analysis of security alerts, enabling them to focus on critical threats and reducing response time. To the best of our knowledge, CTIBench is the first benchmark specifically designed to evaluate LLMs' comprehension, reasoning, and problem-solving abilities in the broad domain of CTI, addressing the limitations of existing benchmarks that either focus on general language understanding or specific cybersecurity tasks.

2 Related Work

Large Language Models in Cybersecurity. The advent of large language models (LLMs) like ChatGPT-3 [16], ChatGPT-4 [12], LLama models [48], Gemini [47] has enabled a wide range of applications across different domains, including cybersecurity. For example: code-based LLMs like CodeLlama [42] that can generate secure code are already integral parts of various industries. LLMs have also found applications in several other cybersecurity tasks like vulnerability detection [43, 21, 38], incidence response [13], program repair [44], IT operations [24], and cybersecurity knowledge assistance [46]. Despite these advancements of LLMs, there is a significant gap in their evaluation in the cybersecurity domain.

Evaluating Large Language Models. General evaluations benchmarks like GLUE [51], SuperGLUE [50], MMLU [26], HELM [31] and KOLA [53] assess general understanding capabilities of LLMs.

Existing evaluation works in cybersecurity are either limited by their lack of comprehensiveness or being too narrow in their domain adaptation. For example: SECURE [15] proposes benchmarks for ICS industries, Purple Llama CyberSecEval [14], and SecLLMHolmes [49] evaluates LLMs' propensity to generate insecure code. SevenLLM [27] design tasks focused on extracting entities, relationships, and generating summaries categorized into understanding and generation tasks, and lack practical problem-solving evaluation of LLMs in the CTI domain. Other cybersecurity benchmarks [33] [30] [32] only evaluate the memorization and information extraction capabilities of LLMs.

3 CTIBench: A Benchmark for Evaluating LLMs in CTI

We are motivated by the need to create knowledge-intensive tasks to evaluate the cognitive capabilities of LLMs in Cyber Threat Intelligence (CTI) [29]. Our benchmark aims to verify that LLMs have the ability to understand, investigate, and analyze cyber threat reports. To this end, we have designed tasks and datasets that emphasize four fundamental cognitive capabilities of LLMs in CTI: *memorization* (ability to recall and utilize previously learned information), *understanding* (ability to comprehend the content and context), *problem-solving* (ability to apply knowledge and reasoning to address specific challenges), and *reasoning* (ability to draw logical conclusions and make informed decisions based on available information) [53].

3.1 CTI-MCQ: Cyber Threat Intelligence Multiple Choice Questions

Data Collection. To generate the CTI-MCQ dataset, we utilize various authoritative sources within the CTI domain. These sources include, among others, CTI frameworks such as NIST [28], the Diamond Model of Intrusion Detection [17], and regulations like GDPR [1]. Additionally, we incorporate CTI sharing standards such as STIX and TAXII [11] to formulate questions on fundamental cybersecurity and CTI knowledge. We leveraged the MITRE ATT&CK framework [45], CWE database [19], and CAPEC [18] to develop questions on attack patterns, threat actors, APT campaigns, detection methods, mitigation strategies, common software vulnerabilities and attack pattern enumeration. Finally, we supplement our dataset by manually collecting and curating questions from publicly available CTI quizzes, ensuring their relevance and accuracy by referencing established CTI resources. While these quizzes *may* introduce some bias, as they may not fully represent the entire spectrum of CTI knowledge, we ensured our diverse range of sources and manual curation process mitigated this potential limitation.

Generating Questions Using GPT-4. We utilize GPT-4o [5] to prepare the MCQs. Initially, we preprocess the content from the webpage to remove sections inappropriate for MCQ generation (such as page creation or update dates, references, etc.). We then optimize the prompt for creating questions that are challenging enough to test the knowledge of LLMs in cybersecurity. An example prompt is shown in Prompt A. We vary the number of questions we generate based on document length; we create more questions as the length of the document increases. We then randomly sample approximately 3000 questions for manual validation. This approach ensures a variety of questions while retaining quality

Dataset Validation. We manually analyze the quality of MCQs extracted from ChatGPT-4o to ensure the quality of the CTI-MCQ dataset. The human annotators were given access to the source URLs when analyzing the questions. We identified *two major issues from LLM-generated questions*: some questions had multiple correct options in the answers, and sometimes the LLM-given answer was incorrect. We removed the questions that were unanswerable from the given context or questions that included multiple correct options. We fixed the responses that had incorrect annotations provided by ChatGPT-4o.

Our final dataset consists of 2500 questions, out of which, 1578 questions are collected from MITRE, 750 from CWE, 40 from the manual collection, and 32 from standards and frameworks. This approach ensures a variety of questions while retaining quality.

3.2 CTI-RCM: Cyber Threat Intelligence Root Cause Mapping

Root cause mapping (RCM) identifies the underlying cause(s) of a vulnerability by correlating CVE records and bug tickets with CWE entries [19]. Accurate RCM is crucial for guides investments, policies, and practices to address and eliminate the root causes of vulnerabilities, benefiting both industry and government decision-makers. However, the current vulnerability management ecosystem *does not* perform this task accurately at scale [37], due to the complexity and nuance of CVE descriptions, the vast number of CWE categories (over 900 as of May 2024), and the need for domain expertise to interpret the relationships between them. To address this challenge, we designed the CTI-RCM task.Here is an example of RCM from MITRE [37]. The description for CVE-2018-15506 is as follows:

> In BubbleUPnP 0.9 update 30, the XML parsing engine for SSDP/UPnP functionality is vulnerable to an XML External Entity Processing (XXE) attack. Remote, unauthenticated attackers can use this vulnerability to: (1) Access arbitrary files from the filesystem with the same permission as the user account running BubbleUPnP, (2) Initiate SMB connections to capture a NetNTLM challenge/response and crack the cleartext password, or (3) Initiate SMB connections to relay a NetNTLM challenge/response and achieve Remote Command Execution in Windows domains.

The CWE is mapped to CWE-611: Improper Restriction of XML External Entity Reference. Here is the reasoning excerpt as shown in the reference link:

> Reasoning: Description says “vulnerable to an XML External Entity Processing (XXE) attack.” There is additional information that focuses on technical impact, for instance, “attackers can do [X]”, which is rarely useful for weaknesses, so that can be ignored. When you perform a text search on CWE for “XML External Entity Processing (XXE) attack” and “XXE,” it returns CWE-611. When you click the entry, you see that the entry lists XXE as an “Alternate Term.” Therefore, CWE-611 is the right mapping for this CVE.

As can be seen, this is a very nuanced task and requires a deep understanding of both the CVE descriptions and the CWE taxonomy to make accurate mappings.

Data Collection. For this task, we collect data from the National Vulnerability Database (NVD) [10]. The NVD database provides descriptions of past vulnerabilities identified by CVE, along with their associated mappings to Common Weakness Enumeration (CWE) entries. For our study, we specifically focus on vulnerabilities reported in the year 2024 that include associated CWE mappings. We then randomly sample 1,000 vulnerabilities to include in our dataset.

3.3 CTI-VSP: Cyber Threat Intelligence Vulnerability Severity Prediction

The Vulnerability Severity Prediction task involves predicting the Common Vulnerability Scoring System (CVSS) vector string from a given vulnerability description. The CVSS is a standardized framework used to assess the severity of security vulnerabilities. It is composed of three metric groups: Base, Temporal, and Environmental. The Base metric group, which we focus on in our study, reflects the severity of a vulnerability based on its intrinsic characteristics. These characteristics are constant over time and assume a reasonable worst-case impact across different deployed environments. More details can be found in Appendix B.

While accurately calculating an accurate CVSS score requires additional detailed information such as original bug identification, third-party exploit analysis, or technical documentation for the vulnerable software, an approximation can be made using the initial CVE description[1]. The CVSS v3 Base Score is derived from the following eight metrics: *Attack Vector (AV)*, *Attack Complexity (AC)*, *Privileges Required (PR)*, *User Interaction (UI)*, *Scope (S)*, *Confidentiality Impact (C)*, *Integrity Impact (I)*, and *Availability Impact (A)*. Accurately calculating the CVSS score necessitates correctly mapping the vulnerability description to the appropriate severity levels for each metric. This task is inherently challenging due to the need for precise interpretation and understanding of technical language and context. Consequently, it serves as a robust benchmark for evaluating the capability of Large Language Models (LLMs) in understanding and processing cybersecurity-related information. Note that while there is a newer CVSS standard, CVSS 4.0, it was standardized in November 2023, and not all models might have adequate knowledge of the standard. Therefore, we focus on CVSS v3.

Data Collection. We use the same data source as the CTI-RCM for this task. Specifically, we collect 1,000 vulnerability descriptions from 2024 and their associated CVSS v3 strings.

[1] `https://www.first.org/cvss/v3.0/examples`

3.4 CTI-TAA: Cyber Threat Intelligence Threat Actor Attribution

Threat actor attribution is a crucial process of identifying the individuals, groups, or organizations responsible for a cyberattack or malicious activity. This is usually done by analyzing various indicators of compromise (IOCs), such as malware signatures, attack vectors, infrastructure, tactics, techniques, procedures (TTPs), and other contextual information like geopolitical motives or previous attack patterns. This is a challenging task because threat actors often use sophisticated evasion tactics, shared tools and techniques, limited and incomplete data, rapidly changing TTPs, and inherent biases in analysis [36]. This task exemplifies abductive reasoning, which involves forming plausible conclusions from incomplete information, often seeking the best explanation [20]. By evaluating LLMs on this task, we aim to benchmark their capability to perform complex reasoning and analysis in the context of CTI.

Data Collection We create a small-scale dataset to enable LLMs to reason about this intricate CTI concept. To this end, we collect 50 threat reports from reputed vendors that have an Advanced Persistent Threat (APT) group attributed to the reports [23]. The reports vary in the amount of detail provided about the threat actor.

To create a controlled evaluation setting, we remove all mentions of the threat actors and their associated malware campaign names, replacing them with placeholders. We then task the LLMs with attributing the reports to known threat actors. To further ensure accuracy, we manually verify each LLM response to account for the multiple aliases that threat actors often use.

4 Experiments and Results

4.1 Experimental Settings

We evaluate ChatGPT-3.5 (`gpt-3.5-turbo`) [3], ChatGPT-4 (`gpt-4-turbo`) [4], Gemini-1.5 [2], LLAMA3-70B [7] and LLAMA3-8B [8] on our benchmark tasks. We set the temperature of LLMs at 0 and $top\_p = 1$ to obtain more deterministic responses. Each task is evaluated on a zero-shot prompt template with instruction of LLMs to act as a cybersecurity expert. Below, we show a prompt template used for the vulnerability root cause mapping. Please see Appendix C for evaluation prompts used for all tasks.

> You are a cybersecurity expert specializing in cyber threat intelligence. Analyze the following CVE description and map it to the appropriate CWE. Provide a brief justification for your choice. Ensure the last line of your response contains only the CWE ID.
>
> CVE Description:
> $\{description\}$

4.2 Evaluation Metrics

We use accuracy to evaluate the CTI-MCQ and CTT-RCM tasks, as both tasks are equivalent to multi-class classification. For the CTI-VSP task, we compute the mean absolute deviation (MAD) between the CVSS v3.1 scores of the ground truth and the model's predictions. Although we ask the model to predict a vector string, the CVSS score can be deterministically derived from it. We utilize the Python library *cvss* [41] to compute the CVSS score (a numerical value in the range of 0-10 that determines the overall severity of a vulnerability) from the predicted string, ensuring that any potential errors from the LLM performing the computation are eliminated. This approach focuses solely on assessing the LLM's reasoning ability regarding vulnerability.

For the CTI-TAA task, we categorize the predictions into three types: *correct answer* (when the LLM accurately identifies the threat actor or one of its aliases), *plausible answer* (when the threat report lacks sufficient details to pinpoint the answer, but the LLM provides a plausible or related threat actor within a similar group), and *incorrect answer* (when the LLM misattributes the threat actor due to misjudgment, hallucination, or spurious correlation). Based on these categories, we compute two types of accuracy: Correct Accuracy, which is the fraction of correct answers, and Plausible Accuracy, which is the fraction of correct and plausible answers combined. Details on generating the ground truth and evaluation for CTI-TAA are provided in Appendix D.

Model	**CTI-MCQ (Acc)**	**CTI-RCM (Acc)**	**CTI-VSP (MAD)**	**CTI-TAA (Acc)**	
				Correct	**Plausible**
ChatGPT-4	**71.0**	**72.0**	1.31	**52**	**86**
ChatGPT-3.5	54.1	67.2	1.57	44	62
Gemini-1.5	65.4	66.6	**1.09**	38	74
LLAMA3-70B	65.7	65.9	1.83	52	80
LLAMA3-8B	61.3	44.7	1.91	28	36

Table 1: Results of different LLMs on the benchmark datasets (bold indicates the best performing model, lower is better for MAD)

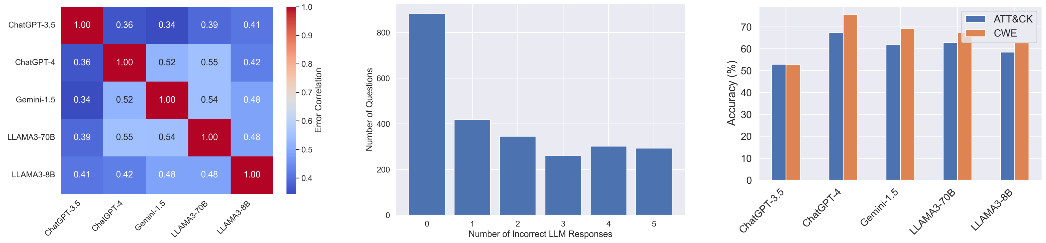
4.3 Results Summary

Table 1 presents the performance of various LLMs on our benchmark CTI tasks. The results indicate that GPT-4 outperforms other models across all tasks except for CTI-VSP, where Gemini-1.5 takes the lead. Despite being open-source, LLAMA3-70B performs comparably to Gemini-1.5 and even outperforms it on two tasks, though it struggles with the CTI-VSP task. ChatGPT-3.5 exceeds the performance of LLAMA3-8B but is generally outperformed by the other models across most tasks. LAMA3-8B, being a smaller model, fails to match the performance of larger models on tasks requiring more nuanced understanding and reasoning. However, it performs decently on the CTI-MCQ task.

5 Analysis

5.1 CTI-MCQ

The heatmap analysis of error correlations, in Figure 2(a), shows that the larger models exhibit higher error correlations. This trend suggests that these models, such as ChatGPT-4, Gemini-1.5, and LLAMA3-70B, are likely to make similar mistakes when answering the CTI-MCQ. For instance, ChatGPT-4 shows error correlations of 0.52 with Gemini-1.5 and 0.55 with LLAMA3-70B, while Gemini-1.5 and LLAMA3-70B have a correlation of 0.54. In Figure 2(b), we show the number of questions incorrectly answered by a number of LLMs. Overall, 293 questions were answered incorrectly by all the models (5) in the evaluation. Upon inspection, we found these questions to be related to mitigation plans and tools and techniques used by adversaries. We show a sample of such questions in Appendix Table 2.



(a) Error correlation

(b) Error frequency

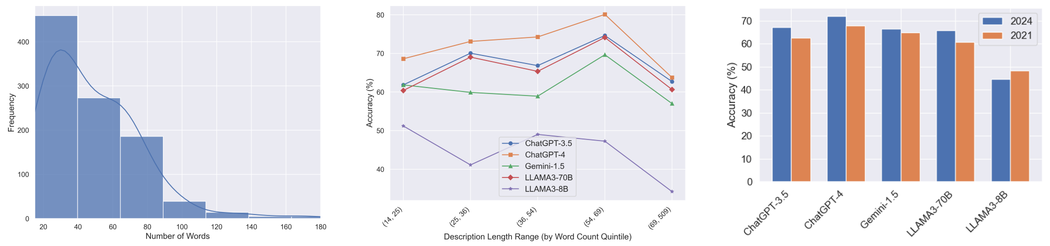
(c) Acc on ATT&CK & CWE

Figure 2: Error analysis on the CTI-MCQ tasks

Figure 2(c) displays the accuracy of different LLMs on multiple-choice questions (MCQs) from two primary sources: MITRE ATT&CK and CWE. Given that MITRE ATT&CK information is more volatile compared to the more stable nature of CWEs, models generally perform better on questions sourced from CWE. However, even the best-performing model, ChatGPT-4, achieves an accuracy of 75.65%, indicating that there is still further room for improvement.

5.2 CTI-RCM

In the CTI-RCM task, LLMs are assigned to identify the underlying cause(s) of a vulnerability by correlating CVE Records with CWE entries. Figure 3(a) shows the frequency distribution of word counts across the CVE descriptions in our dataset, revealing a right-skewed distribution where most descriptions have a lower word count. Figure 3(b) demonstrates that all models, except LLAMA3-8B, improve accuracy with longer descriptions, peaking in the 54-69 word range. This trend suggests that longer descriptions offer more context, aiding the models in accurately mapping CVE records to CWE entries. However, performance declines for the longest CVE descriptions. The most likely reason is that as description length increases, the potential for matching multiple weaknesses increases, causing LLMs to struggle to identify the most appropriate one.



(a) Distribution of word counts in CVE descriptions

(b) Accuracy vs. description length for different LLMs for CTI-RCM

(c) Accuracy for the CTI-ROM task for 2 different years

Figure 3: CTI-RCM and CTI-VSP analysis

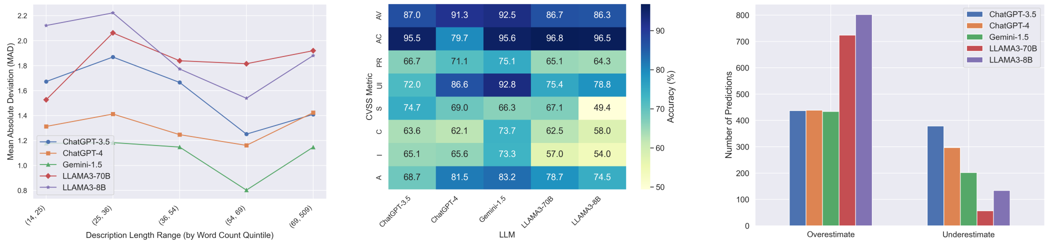
When creating the CTI-RCM dataset, we gathered CVE descriptions exclusively from 2024, which is beyond the training cut-off date for the models we evaluated. To investigate model performance further, we conducted an additional experiment using CVE descriptions and their associated CWE mappings from 2021. The results, presented in Figure 3 (c), show that four out of five models perform slightly worse on the 2021 dataset. This suggests the task is inherently challenging and could serve as a robust evaluation benchmark for future LLMs.

5.3 CTI-VSP

In the CTI-VSP task, LLMs are tasked with predicting the CVSS v3 Base String based on the CVE description, which is then converted to a CVSS score using a predefined formula. We evaluate the performance of the LLMs by computing the Mean Absolute Deviation (MAD) from the ground truth in the dataset. Like the CTI-RCM task, models tend to perform better with longer descriptions, as indicated by a decrease in MAD in Figure 3(c). However, there is a noticeable performance drop for all the models, as evidenced by the sharp increase in MAD for the last quintiles. This pattern suggests that while longer descriptions provide more context and generally improve performance, they can also introduce complexity that leads to misattribution of the severity of the threat. This combined finding from the CTI-VSP and CTI-RCM tasks indicates that additional description length does not necessarily equate to better performance and may hinder the models' ability to assess the threat accurately.

We illustrate the accuracy of LLMs in predicting the CVSS v3 Base Metrics in Figure 4(a). The CVSS Base Metrics include Attack Vector (AV), Attack Complexity (AC), Privileges Required (PR), User Interaction (UI), Scope (S), Confidentiality Impact (C), Integrity Impact (I), and Availability Impact (A) (detailed in Appendix B). The heatmap shows that all models perform relatively well predicting AV, AC, and UI. However, they struggle with PR, S, C, and I, suggesting that CVE descriptions often lack sufficient detail to infer these metrics accurately.

Figure 4(b) illustrates the number of overestimations and underestimations made by LLMs when predicting the CVSS base score from a vulnerability description. Overestimation occurs when the predicted score is higher than the actual score, while underestimation occurs when the predicted score is lower. All models exhibit a higher frequency of overestimation compared to underestimation, with this tendency being particularly pronounced in the two open-source LLAMA models. This suggests that LLMs may need calibration to improve their accuracy in threat severity prediction.



(a) MAD vs. description length for different LLMs for CTI-VSP

(b) Accuracy with respect to base metrics

(c) Error analysis of severity score prediction

Figure 4: CTI-VSP analysis

5.4 CTI-TAA

The results of the threat actor attribution tasks (Table 1) indicate that LLMs can perform nuanced analyses of the information presented in threat reports and make insightful correlations. While smaller models like LLAMA3-8B struggle with more complex reasoning tasks, larger models demonstrate reasonable performance. Our analysis of the reasoning provided by LLMs suggests that they possess a general understanding of the cyber threat landscape, though they may occasionally misattribute information. Below, we present representative examples of threat-attribution predictions made by ChatGPT-4.

Correct response: We task the LLM to predict the threat actor CHRYSENE given a threat report by replacing the mention of the threat actor and its campaign with [PlaceHolder].[2] CHRYSENE is an Iranian cyberespionage group active since 2014, targeting Middle Eastern governments and various industries. ChatGPT-4 predicted the threat actor as OilRig, which is an alias of CHRYSENE.

Plausible response: We task the LLM to predict the threat actor MuddyWater given a threat report by replacing the mention of the threat actor and its campaign with [PlaceHolder].[3] MuddyWater is a cyber espionage group linked to Iran's Ministry of Intelligence and Security (MOIS) that targets government and private sectors across the Middle East, Asia, Africa, Europe, and North America. ChatGPT-4 predicted the threat actor as APT35, which is not the alias of MuddyWater but shares some common attack patterns like originating from Iran, targeting the Middle East and North America, and using multi-stage attacks.

Incorrect response: We task the LLM to predict the threat actor APT41 given a threat report by replacing the mention of the threat actor and its campaign with [PlaceHolder].[4] APT41 is an espionage group from China that has been active since 2012 and involved in financially motivated operations, targeting healthcare, telecom, technology, and video game industries. ChatGPT-4 incorrectly predicted the threat actor as APT29, based on the fact that APT29 has been active since 2012 and its state-sponsored espionage operations.

5.5 Compute Cost

Appendix F presents the detailed token counts for each task. GPT-4 generated significantly longer responses across all tasks than the other models.

6 Ethical Concerns

All the evaluation tasks in our proposed benchmark utilize publicly available threat information from reputable sources such as NIST, MITRE, CVE, CWE, and EU. None of the datasets include personal information or make sensitive judgments related to social issues, bias, deception, or discrimination.

[2] https://www.welivesecurity.com/en/eset-research/oilrigs-outer-space-juicy-mix-same-ol-rig-new-drill-pipes/

[3] https://www.deepinstinct.com/blog/darkbeatc2-the-latest-muddywater-attack-framework

[4] https://www.trendmicro.com/en_us/research/24/d/earth-freybug.html

7 Limitations

While our evaluation of LLMs for Cyber Threat Intelligence (CTI) tasks provides valuable insights, it is important to acknowledge certain limitations. First, the extensive range of CTI tasks poses a significant challenge, and our study has only selected a limited subset of these tasks to measure the LLM's capabilities. This selection may not fully represent the comprehensive functionality required for effective CTI operations. We will broaden the scope of evaluated tasks to encompass a wider array of CTI activities, ensuring a more holistic assessment of LLM performance in our future works. Second, our evaluation is restricted to English-language CTI techniques. This limitation overlooks the global nature of cyber threats, which often involve multiple languages. Future studies will consider incorporating multilingual CTI evaluations to better reflect the diverse linguistic landscape of cyber threats and enhance the applicability of LLMs in international contexts.

8 Conclusion

The emergence of LLMs has opened up new possibilities in cybersecurity, especially in Cyber Threat Intelligence (CTI). However, their capabilities and limitations in this domain remain unclear. In this paper, we introduce CTIBench, a benchmark designed to evaluate LLM performance in various CTI tasks. Our evaluation offers valuable insights into the knowledge and capabilities of LLMs across various CTI aspects, as well as their limitations. We aim for our benchmark to help researchers understand the practical applications of LLMs in CTI and to pave the way for their reliable use and the effective detection and mitigation of cyberthreats.

References

[1] 2024. *European Union, GDPR*. Available at https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679.

[2] 2024. *Gemini Models*. Available at https://deepmind.google/technologies/gemini/pro/.

[3] 2024. *GPT-3.5 Turbo Model OpenAI*. Available at https://platform.openai.com/docs/models/gpt-3-5-turbo.

[4] 2024. *GPT-4-Turbo and GPT-4 Model OpenAI*. Available at https://platform.openai.com/docs/models/gpt-4-turbo-and-gpt-4.

[5] 2024. GPT-4o Model OpenAI. Available at https://platform.openai.com/docs/models/gpt-4o.

[6] 2024. *Malpedia*. Available at https://malpedia.caad.fkie.fraunhofer.de/.

[7] 2024. *Meta Llama3-70b*. Available at https://huggingface.co/meta-llama/Meta-Llama-3-70B.

[8] 2024. *Meta Llama3-8b*. Available at https://huggingface.co/meta-llama/Meta-Llama-3-8B.

[9] 2024. *MITRE ATT&CK*. Available at https://attack.mitre.org/.

[10] 2024. *National Vulnerability Database*. Available at https://nvd.nist.gov/.

[11] 2024. *Threat Intelligence Standards*. Available at https://oasis-open.github.io/cti-documentation/.

[12] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[13] Toufique Ahmed, Supriyo Ghosh, Chetan Bansal, Thomas Zimmermann, Xuchao Zhang, and Saravan Rajmohan. 2023. Recommending root-cause and mitigation steps for cloud incidents using large language models. In *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*. IEEE, 1737–1749.

[14] Manish Bhatt, Sahana Chennabasappa, Cyrus Nikolaidis, Shengye Wan, Ivan Evtimov, Dominik Gabi, Daniel Song, Faizan Ahmad, Cornelius Aschermann, Lorenzo Fontana, et al. 2023. Purple llama cyberseceval: A secure coding benchmark for language models. *arXiv preprint arXiv:2312.04724* (2023).

[15] Dipkamal Bhusal, Md Tanvirul Alam, Le Nguyen, Ashim Mahara, Zachary Lightcap, Rodney Frazier, Romy Fieblinger, Grace Long Torales, and Nidhi Rastogi. 2024. SECURE: Benchmarking Generative Large Language Models for Cybersecurity Advisory. arXiv:2405.20441 [cs.CR]

[16] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[17] Sergio Caltagirone, Andrew Pendergast, and Christopher Betz. 2013. The diamond model of intrusion analysis. *Threat Connect* 298, 0704 (2013), 1–61.

[18] CAPEC. 2024. Common Attack Pattern Enumerations and Classifications. Available at `https://capec.mitre.org/`.

[19] Steve Christey, J Kenderdine, J Mazella, and B Miles. 2013. Common weakness enumeration. *Mitre Corporation* (2013).

[20] Igor Douven. 2021. Abduction. In *The Stanford Encyclopedia of Philosophy* (Summer 2021 ed.), Edward N. Zalta (Ed.). Metaphysics Research Lab, Stanford University.

[21] Mohamed Amine Ferrag, Ammar Battah, Norbert Tihanyi, Merouane Debbah, Thierry Lestable, and Lucas C Cordeiro. 2023. Securefalcon: The next cyber reasoning system for cyber security. *arXiv preprint arXiv:2307.06616* (2023).

[22] FIRST.Org, Inc. 2019. Common Vulnerability Scoring System v3.1: Specification Document. `https://www.first.org/cvss/v3.1/specification-document` Accessed: 2024-06-05.

[23] Fraunhofer FKIE. 2024. Threat Actors (powered by MISP). `https://malpedia.caad.fkie.fraunhofer.de/actors`. `https://malpedia.caad.fkie.fraunhofer.de/actors` Accessed: 2024-06-05.

[24] Hongcheng Guo, Jian Yang, Jiaheng Liu, Liqun Yang, Linzheng Chai, Jiaqi Bai, Junran Peng, Xiaorong Hu, Chao Chen, Dongfeng Zhang, et al. 2023. Owl: A large language model for it operations. *arXiv preprint arXiv:2309.09298* (2023).

[25] Taicheng Guo, Bozhao Nan, Zhenwen Liang, Zhichun Guo, Nitesh Chawla, Olaf Wiest, Xiangliang Zhang, et al. 2023. What can large language models do in chemistry? a comprehensive benchmark on eight tasks. *Advances in Neural Information Processing Systems* 36 (2023), 59662–59688.

[26] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring Massive Multitask Language Understanding. In *International Conference on Learning Representations*. `https://openreview.net/forum?id=d7KBjmI3GmQ`

[27] Hangyuan Ji, Jian Yang, Linzheng Chai, Chaoren Wei, Liqun Yang, Yunlong Duan, Yunli Wang, Tianzhen Sun, Hongcheng Guo, Tongliang Li, et al. 2024. SEvenLLM: Benchmarking, Eliciting, and Enhancing Abilities of Large Language Models in Cyber Threat Intelligence. *arXiv preprint arXiv:2405.03446* (2024).

[28] Chris Johnson, Lee Badger, David Waltermire, Julie Snyder, Clem Skorupka, et al. 2016. Guide to cyber threat information sharing. *NIST special publication* 800, 150 (2016), 35.

[29] David R Krathwohl. 2002. A revision of Bloom's taxonomy: An overview. *Theory into practice* 41, 4 (2002), 212–218.

[30] Guancheng Li, Yifeng Li, Wang Guannan, Haoyu Yang, and Yang Yu. 2023. SecEval: A Comprehensive Benchmark for Evaluating Cybersecurity Knowledge of Foundation Models. https://github.com/XuanwuAI/SecEval.

[31] Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Alexander Cosgrove, Christopher D Manning, Christopher Re, Diana Acosta-Navas, Drew Arad Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue WANG, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri S. Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Andrew Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. 2023. Holistic Evaluation of Language Models. *Transactions on Machine Learning Research* (2023). `https://openreview.net/forum?id=iO4LZibEqW` Featured Certification, Expert Certification.

[32] Zefang Liu. 2023. SecQA: A Concise Question-Answering Dataset for Evaluating Large Language Models in Computer Security. *arXiv preprint arXiv:2312.15838* (2023).

[33] Zefang Liu, Jialei Shi, and John F Buford. 2024. CyberBench: A Multi-Task Benchmark for Evaluating Large Language Models in Cybersecurity. (2024).

[34] Aman Madaan and Amir Yazdanbakhsh. 2022. Text and patterns: For effective chain of thought, it takes two to tango. *arXiv preprint arXiv:2209.07686* (2022).

[35] Rob McMillan. 2013. Definition: threat intelligence. *Gartner. com* 5 (2013).

[36] Yangyang Mei, Weihong Han, Shudong Li, Xiaobo Wu, KaiHan Lin, and Yulu Qi. 2022. A Review of Attribution Technical for APT Attacks. In *2022 7th IEEE International Conference on Data Science in Cyberspace (DSC)*. IEEE, 512–518.

[37] MITRE. 2024. CVE → CWE Mapping "Root Cause Mapping" Guidance. `https://cwe.mitre.org/documents/cwe_usage/guidance.html#rcm` Accessed: 2024-06-05.

[38] Aleena Noor. [n. d.]. Large Language Models in Cybersecurity: Upcoming AI Trends in 2023-24. ([n. d.]).

[39] CVE Project. 2024. CVEs published in 2024. `https://cve.mitre.org/`. Accessed: 2024-05-24.

[40] CWE Project. 2024. CWE: Common Weakness Enumeration. Available at `https://cwe.mitre.org/index.html`.

[41] Red Hat Product Security. 2023. CVSS2/3/4 library with interactive calculator for Python 2 and Python 3. `https://pypi.org/project/cvss/` Accessed: 2024-06-05.

[42] Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, et al. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950* (2023).

[43] Alexey Shestov, Anton Cheshkov, Rodion Levichev, Ravil Mussabayev, Pavel Zadorozhny, Evgeny Maslov, Chibirev Vadim, and Egor Bulychev. 2024. Finetuning Large Language Models for Vulnerability Detection. *arXiv preprint arXiv:2401.17010* (2024).

[44] André Silva, Sen Fang, and Martin Monperrus. 2023. RepairLLaMA: Efficient Representations and Fine-Tuned Adapters for Program Repair. *arXiv preprint arXiv:2312.15698* (2023).

[45] Blake E Strom, Andy Applebaum, Doug P Miller, Kathryn C Nickels, Adam G Pennington, and Cody B Thomas. 2018. Mitre ATT&CK: Design and philosophy. In *Technical report*. The MITRE Corporation.

[46] Madeena Sultana, Adrian Taylor, Li Li, and Suryadipta Majumdar. 2023. Towards Evaluation and Understanding of Large Language Models for Cyber Operation Automation. In *2023 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 1–6.

[47] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295* (2024).

[48] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).

[49] Saad Ullah, Mingji Han, Saurabh Pujar, Hammond Pearce, Ayse Coskun, and Gianluca Stringhini. 2024. LLMs Cannot Reliably Identify and Reason About Security Vulnerabilities (Yet?): A Comprehensive Evaluation, Framework, and Benchmarks. In *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 199–199.

[50] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems* 32 (2019).

[51] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461* (2018).

[52] Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, et al. 2021. Ethical and social risks of harm from language models. *arXiv preprint arXiv:2112.04359* (2021).

[53] Jifan Yu, Xiaozhi Wang, Shangqing Tu, Shulin Cao, Daniel Zhang-Li, Xin Lv, Hao Peng, Zijun Yao, Xiaohan Zhang, Hanming Li, et al. 2023. KoLA: Carefully Benchmarking World Knowledge of Large Language Models. In *The Twelfth International Conference on Learning Representations*.

[54] Juanma Zambrano Chaves, Nandita Bhaskhar, Maayane Attias, Jean-Benoit Delbrouck, Daniel Rubin, Andreas Loening, Curtis Langlotz, and Akshay Chaudhari. 2023. RaLEs: a Benchmark for Radiology Language Evaluations. *Advances in Neural Information Processing Systems* 36 (2023), 74429–74454.

[55] Wenxuan Zhang, Mahani Aljunied, Chang Gao, Yew Ken Chia, and Lidong Bing. 2024. M3exam: A multilingual, multimodal, multilevel benchmark for examining large language models. *Advances in Neural Information Processing Systems* 36 (2024).

Checklist

1. For all authors...

   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]

   (b) Did you describe the limitations of your work? [Yes]

   (c) Did you discuss any potential negative societal impacts of your work? [N/A]

   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...

   (a) Did you state the full set of assumptions of all theoretical results? [N/A]

   (b) Did you include complete proofs of all theoretical results? [N/A]

3. If you ran experiments (e.g. for benchmarks)...

   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]

   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]

   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]

   (d) Did you include the total compute time and the resource type used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

   (a) If your work uses existing assets, did you cite the creators? [Yes]

   (b) Did you mention the license of the assets? [N/A]

   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]

   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes]

5. If you used crowdsourcing or conducted research with human subjects...

   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Appendix

A Prompt for generating CTI-MCQ

> **Prompt for generating MCQs with ChatGPT-4o from MITRE ATT&CK**
> (Italized parts are unique to MITRE ATT&CK)
>
> You are a cybersecurity expert specializing in cyber threat intelligence. Given the text below, please generate a maximum of 5 multiple-choice questions with four possible options each.
>
> Follow these requirements:
> 1. Question Format: Each question must have four options. The options should be challenging and require careful consideration. Avoid creating options that could be interpreted as correct under different circumstances.
> 2. Target Audience: The questions should be suitable for security professionals with three to five years of experience in cyber threat intelligence. *Avoid generic questions such as "What is the objective?", "Which operating system can be targeted?".*
> 3. Content Coverage: Aim to cover various sections of the document to ensure a comprehensive evaluation of the candidate's knowledge. Include context-specific questions that require an understanding of the document's content.
> 4. Technical Precision: Use precise terminology and concepts relevant to cyber threat intelligence. Incorporate situational or scenario-based questions where applicable.
> 5. *Include Technique IDs and Names: Ensure that all questions, where applicable, mention both the ID and the full name of the MITRE ATT&CK pattern technique.*
> 6. *Premise Inclusion: Each question should include a premise indicating it pertains to MITRE ATT&CK, specifying the relevant platform (Enterprise, ICS, or Mobile) where necessary.*
> 7. Output Format: Return the output in TSV format (must be tab-separated) with the following columns: Question, Option A, Option B, Option C, Option D, Correct Answer (A, B, C, D), and Explanation.
>
> **Important:** Only return the TSV content as specified. Do not include any additional text or commentary outside the TSV format.
>
> Text:
> ...

B CVSS Base Metric

The Base Metrics group in CVSS encapsulates a vulnerability's fundamental and immutable properties and is crucial for understanding the potential impact and exploitability of a vulnerability [22]. It consists of the following metrics:

1. **Attack Vector (AV):** The Attack Vector metric evaluates the proximity an attacker must have to the vulnerable component to exploit it. Possible values:
   - *Network (N):* The attacker can exploit the vulnerability remotely over a network. This typically indicates the highest risk, as remote exploitation can be conducted without physical access.
   - *Adjacent (A):* The attacker requires access to the local network or adjacent hardware. Examples include attacks over Bluetooth or local subnet.
   - *Local (L):* The attacker needs local access to the system, either physically or via a local account. This increases the difficulty compared to network attacks.
   - *Physical (P):* The attacker must have physical contact or access to the target system, which makes this the most challenging attack vector.
2. **Attack Complexity (AC):** This metric measures the conditions beyond the attacker's control that must exist to exploit the vulnerability. Possible values:
   - *Low (L):* Exploiting the vulnerability does not require any special conditions.
   - *High (H):* Exploitation requires specific conditions or configurations.
3. **Privileges Required (PR):** Privileges Required assesses the level of access or privileges an attacker must have to successfully exploit the vulnerability. Possible values:

- *None (N):* The attacker does not need any privileges; they can exploit the vulnerability as an unauthenticated user. This typically results in a higher score since the barrier to exploitation is minimal.
- *Low (L):* The attacker needs basic user privileges. This implies that some level of access is required, but not necessarily elevated privileges.
- *High (H):* Exploitation requires administrative or high-level privileges. This makes exploitation significantly harder.

4. **User Interaction (UI):** User Interaction measures the degree to which a user must participate in the attack. Possible values:
   - *None (N):* The attacker can exploit the vulnerability without any interaction from the user. This indicates a higher risk as the attack can be automated and spread more easily.
   - *Required (R):* Successful exploitation requires user interaction, such as clicking a link or opening a file. This adds a layer of difficulty for the attacker since it relies on social engineering or user behavior.
5. **Scope (S):** Scope evaluates whether a vulnerability in one component impacts resources beyond its security scope. This metric assesses the potential broader implications of an exploit. Possible values:
   - *Unchanged (U):* The vulnerability only affects resources within the same security scope. This typically means the impact is contained within a single component or system.
   - *Changed (C):* Exploitation of the vulnerability impacts resources beyond the vulnerable component's security scope.
6. **Confidentiality Impact (C):** Confidentiality Impact measures the extent to which information disclosure can occur due to a successful exploit. Possible values:
   - *None (N):* There is no impact on confidentiality; no sensitive information is disclosed.
   - *Low (L):* Access to some information is gained, but the attacker cannot control what is disclosed, or the information is not highly sensitive.
   - *High (H):* Complete disclosure of all data on the affected system. This indicates a severe impact, potentially exposing highly sensitive information.
7. **Integrity Impact (I):** Integrity Impact assesses the extent to which data can be altered or tampered with by an attacker. Possible values:
   - *None (N):* There is no impact on the integrity of the data or system.
   - *Low (L):* Data can be modified, but the extent is limited or the attacker does not have control over the modifications.
   - *High (H):* The attacker can make extensive or significant modifications to the data, potentially compromising the integrity of the entire system or dataset.
8. **Availability Impact (A):** Availability Impact measures the potential disruption to the availability of the affected component due to a successful exploit.
   - *None (N):* There is no impact on availability; the system remains fully operational.
   - *Low (L):* Performance is degraded, or there are occasional disruptions, but the system remains available.
   - *High (H):* The system is completely unavailable or severely degraded, indicating a significant impact on availability and potentially causing a denial of service.

C Evaluation Prompts

C.1 CTI-MCQ: Cyber Threat Intelligence Multiple Choice Questions

Prompt used for LLM Evaluation We use the following prompt to generate answers from LLMs for the MCQs

> You are a cybersecurity expert specializing in cyber threat intelligence. You are given a multiple-choice question (MCQ) from a Cyber Threat Intelligence (CTI) knowledge benchmark dataset. Your task is to choose the best option among the four provided. Return your answer as a single uppercase letter: A, B, C, or D.
>
> **Question:**
> $\{question\}$
>
> **Options:**
> A) $\{option_a\}$
> B) $\{option_b\}$
> C) $\{option_c\}$
> D) $\{option_d\}$
>
> **Important:** The last line of your answer should contain only the single letter corresponding to the best option, with no additional text.

C.2 CTI-RCM: Cyber Threat Intelligence Root Cause Mapping

Prompt used for LLM Evaluation We use the following prompt to identify the vulnerability root cause mapping:

> You are a cybersecurity expert specializing in cyber threat intelligence. Analyze the following CVE description and map it to the appropriate CWE. Provide a brief justification for your choice. Ensure the last line of your response contains only the CWE ID.
>
> CVE Description:

C.3 CTI-VSP: Cyber Threat Intelligence Vulnerability Severity Prediction

Prompt used for LLM Evaluation We use the following prompt to predict CVSS v3.1 string using LLM:

> Analyze the following CVE description and calculate the CVSS v3.1 Base Score. Determine the values for each base metric: AV, AC, PR, UI, S, C, I, and A. Summarize each metric's value and provide the final CVSS v3.1 vector string.
>
> Valid options for each metric are as follows:
> - Attack Vector (AV): Network (N), Adjacent (A), Local (L), Physical (P)
> - Attack Complexity (AC): Low (L), High (H)
> - Privileges Required (PR): None (N), Low (L), High (H)
> - User Interaction (UI): None (N), Required (R)
> - Scope (S): Unchanged (U), Changed (C)
> - Confidentiality (C): None (N), Low (L), High (H)
> - Integrity (I): None (N), Low (L), High (H)
> - Availability (A): None (N), Low (L), High (H)
>
> Summarize each metric's value and provide the final CVSS v3.1 vector string. Ensure the final line of your response contains only the CVSS v3 Vector String in the following format:
>
> Example format: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
>
> CVE Description:

C.4 CTI-TAA: Cyber Threat Intelligence Threat Actor Attribution

Prompt used for LLM Evaluation We use the following prompt for the threat actor attribution task:

Table 2: A sample of questions incorrectly answered by all the LLMs under evaluation

Question Type	Question	Option A	Option B	Option C	Option D	Correct Answer
Mitigation	Under the MITRE ATT&CK framework for Enterprise, which mitigation can help prevent adversaries from creating or interacting with system services using a lower permission level?	Behavior Prevention on Endpoint	Malware Detection	Privileged Account Management	Restrict File & Directory Permissions	C
Tool	In the context of MITRE ATT&CK for Enterprise, which of the following tools can use PowerShell to discover email accounts as per T1087.003 Account Discovery: Email Account?	TrickBot	MailSniper	Magic Hound	Lizar	C
Method	What is a common method used by adversaries for outbound traffic in Technique ID T1102.002?	Using DNS tunneling	Sending emails to command servers	Making HTTP requests to compromised blogs	Using FTP to upload data	C
Malware	Which of the following malware examples triggers on a magic packet in TCP or UDP packets?	BUSHWALK	Ryuk	SYNful Knock	Penquin	D
Detection	How does Clop avoid sandbox detection?	Using GetTickCount function	Disabling system clock Scheduled Task/Job	Calling NtDelayExecution	Using the sleep command	D

> You are a cybersecurity expert specializing in cyber threat intelligence. You are given a threat report that describes a cyber incident. Any direct mentions of the threat actor group, specific campaign names, or malware names responsible have been replaced with [PLACEHOLDER]. Your task is to analyze the report and attribute the incident to a known threat actor based on the techniques, tactics, procedures (TTPs), and any other relevant information described. Please provide the name of the threat actor you believe is responsible and briefly explain your reasoning.
> Threat Report:

D CTI-TAA Evaluation

To evaluate the threat actor attribution task, we crafted the CTI-TAA dataset. The ground truth for the *correct answer* was established by extracting information directly from original documents, ensuring an exact match with named entities. Additionally, we collected aliases associated with the threat actors using Malpedia [6]. To ensure comprehensive coverage of possible aliases, we referenced the individual Malpedia pages of the threat actors and explored alias pages, capturing secondary aliases derived from primary ones that were not initially included.

We consider all identified aliases as equivalent to the ground truth. Furthermore, we included all related threat actors for the original threat actors, sourced from related or associated groups' information on the MITRE website. This forms the initial plausible set of actors. Given that an alias of one threat actor may sometimes be an alias of another, creating a chain due to incomplete alias lists on individual pages, and considering that related actors may have their aliases, we employed a graph-searching algorithm to determine the accuracy of model predictions.

We used a breadth-first search (BFS) algorithm to verify correctness, starting from the LLM response and traversing only nodes connected by an alias. The response is marked as correct if a match with the ground truth is found before the search is exhausted. We considered nodes connected via both aliases and related group nodes to identify related groups. The response is considered related if a match with the ground truth is found through this expanded search. If no matches are found, the response is deemed incorrect.

The algorithm is shown in Algorithm 1. We make the algorithm included in our released code.

E CTI-MCQ Samples

We show a sample of incorrectly answered MCQs by LLMs from CTI-MCQ dataset in Table 2.

F Compute Cost

We show the approximate number of tokens used in Prompts and in LLM responses in Table 3.

Algorithm 1 Evaluate Model's Prediction for the CIT-TAA task

```
Require: LLM response, Ground truth entities, Alias and related actor mappings
Ensure: Correctness or plausibility of the response
Initialize correct = False, related = False
Check for Correctness
Initialize BFS queue with LLM response node
while queue is not empty and correct is False do
    Dequeue a node
    if node matches ground truth then
        correct = True
    else
        Enqueue all alias-connected nodes
    end if
end while
Check for Plausibility
if correct is False then
    Initialize BFS queue with LLM response node
    while queue is not empty and related is False do
        Dequeue a node
        if node matches ground truth then
            related = True
        else
            Enqueue all alias and related group-connected nodes
        end if
    end while
end if
Determine Result
if correct then
    Response is Correct
else if related then
    Response is Related
else
    Response is Incorrect
end if
```

Model	CTI-MCQ	CTI-RCM	CTI-VSP	CTI-TAA
Prompt	419	143	454	149
ChatGPT-4	105	269	485	32
ChatGPT-3.5	10	135	170	21
Gemini-1.5	5	100	344	32
LLAMA3-70B	1	140	442	25
LLAMA3-8B	5	136	365	21

Table 3: Approximate number of tokens for Prompt and LLM response for each task (in thousands of token units)