

# The Digital Cybersecurity Expert: How Far Have We Come?

Dawei Wang<sup>†</sup>, Geng Zhou<sup>†</sup>, Xianglong Li<sup>†</sup>, Yu Bai<sup>†</sup>, Li Chen<sup>\*†</sup>, Ting Qin<sup>†</sup>, Jian Sun<sup>†</sup>, and Dan Li<sup>‡</sup>

<sup>†</sup>Zhongguancun Laboratory, <sup>‡</sup>Tsinghua University

{wangdw, zhougeng, lixl, baiyu, chenli, qingting, sunjian}@zgclab.edu.cn, toldan@tsinghua.edu.cn

**Abstract**—The increasing deployment of large language models (LLMs) in the cybersecurity domain underscores the need for effective model selection and evaluation. However, traditional evaluation methods often overlook specific cybersecurity knowledge gaps that contribute to performance limitations. To address this, we develop CSEBenchmark, a fine-grained cybersecurity evaluation framework based on 345 knowledge points expected of cybersecurity experts. Drawing from cognitive science, these points are categorized into factual, conceptual, and procedural types, enabling the design of 11,050 tailored multiple-choice questions. We evaluate 12 popular LLMs on CSEBenchmark and find that even the best-performing model achieves only 85.42% overall accuracy, with particular knowledge gaps in the use of specialized tools and uncommon commands. Different LLMs have unique knowledge gaps. Even large models from the same family may perform poorly on knowledge points where smaller models excel. By identifying and addressing specific knowledge gaps in each LLM, we achieve up to an 84% improvement in correcting previously incorrect predictions across three existing benchmarks for two cybersecurity tasks. Furthermore, our assessment of each LLM’s knowledge alignment with specific cybersecurity roles reveals that different models align better with different roles, such as GPT-4o for the Google Senior Intelligence Analyst and Deepseek-V3 for the Amazon Privacy Engineer. These findings underscore the importance of aligning LLM selection with the specific knowledge requirements of different cybersecurity roles for optimal performance.

## 1. Introduction

The rapid advancement of large language models (LLMs) has the potential to revolutionize the cybersecurity field, with the concept of a “digital cybersecurity expert” gaining traction. As these models become increasingly sophisticated, there is growing interest in their ability to assist or even replace human experts in various cybersecurity tasks. The cybersecurity industry has already begun exploring this possibility, with Microsoft introducing Copilot for Security to proactively detect, investigate, and respond to threats [1], and Google launching Gemini in Security to support threat intelligence analysis and streamline security operations [2]. These developments raise a critical question: **How far have we come in achieving a digital cyberse-**

**curity expert?** Answering this question is crucial for understanding the current capabilities and limitations of LLMs in the cybersecurity domain, which in turn has significant implications for the future of the field. As organizations increasingly rely on these models to support or even replace human experts, it is essential to have a clear understanding of their strengths and weaknesses to ensure the effective and responsible deployment of LLMs in cybersecurity roles.

Recent studies have attempted to evaluate LLMs’ capabilities in cybersecurity, which primarily focus on two main areas: their performance on specific security tasks [3]–[18] and their understanding of cybersecurity knowledge [3], [19]–[23]. These studies have identified several limitations of LLMs in cybersecurity applications, while offering valuable insights to the community. However, despite these contributions, these works are insufficient to comprehensively assess the knowledge of LLMs in cybersecurity due to the following limitations:

**L1- Lack of a comprehensive knowledge framework for cybersecurity experts:** Existing evaluation methods fail to address the fundamental question: what constitutes a cybersecurity expert? These methods often focus narrowly on specific skills or tasks, without establishing a comprehensive framework for the knowledge a cybersecurity expert should possess. As a result, the evaluation questions lack depth and fail to systematically cover necessary areas. Some knowledge domains are overemphasized, while equally important ones are arbitrarily neglected, leading to incomplete and unbalanced assessments.

**L2- Inability to identify specific knowledge gaps of LLMs:** Current knowledge-based assessments are coarse-grained, making it difficult to assess LLMs’ understanding of specific knowledge points and identify their true knowledge gaps. While some studies [20], [23] have categorized subdomains within cybersecurity, evaluations within these subdomains lack sufficient detail, limiting their usefulness for model improvement. In task-based assessments, although LLMs’ poor performance on certain tasks is apparent, the lack of clear definitions of the required knowledge makes it difficult to identify the causes of failure. This highlights the need for fine-grained evaluation datasets that can provide actionable insights for model enhancement.

**L3- Mismatch between question design and knowledge mastery requirements:** Different types of knowledge points require different levels of mastery from cybersecurity experts. For example, knowledge of *HTTP status codes* only requires memorization, while *SSL* requires an understanding

\* Corresponding author.

of its internal mechanisms, and *Wireshark* requires hands-on proficiency. Each type of knowledge point requires a tailored evaluation approach. However, existing evaluations often use a one-size-fits-all question design, leading to over-emphasis of some areas and insufficient assessment of others, making it difficult to accurately measure LLMs’ mastery across different knowledge types.

To address these limitations, we design a cognitive science-based, fine-grained knowledge assessment framework for cybersecurity experts, called CSEBenchmark. CSEBenchmark uses multiple-choice questions to evaluate LLMs. To accurately depict the knowledge and skills required of cybersecurity experts, we collect three well-known cybersecurity expert roadmaps [24]–[26], which outline the essential skills and knowledge needed, and consolidate them into a knowledge framework encompassing seven subdomains, including Fundamental IT Skills (FIS), Operating Systems (OS), Networking Knowledge (NK), Web Knowledge (WK), Security Skills and Knowledge (SSK), Cloud Skills and Knowledge (CSK), and Programming Skills and Knowledge (PSK). The entire framework consists of 345 fine-grained knowledge points, providing a comprehensive assessment of LLMs’ understanding of these knowledge domains. Given the varying levels of mastery required for different knowledge points, we categorize them based on cognitive science into three types: factual knowledge (to be memorized), conceptual knowledge (requiring understanding of underlying principles), and procedural knowledge (requiring hands-on practice). For each category, we gather targeted materials and design tailored question templates to ensure a comprehensive and accurate evaluation. We use GPT-4-Turbo to generate the questions, followed by 672 man-hours of review and 100 man-hours of corrections, resulting in 11,050 high-quality multiple-choice questions.

We apply CSEBenchmark to 12 popular LLMs, revealing GPT-4o as the overall best-performing model and Deepseek-V3 as the top open-source model. However, the overall accuracy of the models is only as high as 85.42%, indicating room for improvement. We also reveal that LLMs have notable gaps in procedural knowledge, especially in the use of specialized tools and uncommon commands. Additionally, they even struggle with some foundational factual and conceptual points. Notably, different LLMs exhibit unique knowledge gaps, and even larger models from the same family may underperform on certain knowledge points where smaller models excel. By supplementing these knowledge gaps, we successfully enhance their performance across three existing benchmarks [3], [8], [18] for two cybersecurity tasks, achieving an improvement of up to 84% in correcting previously incorrect predictions, which validates the reliability of our findings. Finally, we evaluate the job-role knowledge alignment of LLMs based on six real-world cybersecurity roles, demonstrating that LLMs are not yet fully capable of meeting real-world job requirements. Each cybersecurity role reveals unique knowledge gaps within the LLMs, emphasizing the need for role-specific improvements.

**Contributions.** Our contributions are summarized as fol-

lows:

- *New evaluation framework.* We introduce CSEBenchmark, the first cognitive science-based cybersecurity knowledge assessment framework that encompasses 345 fine-grained knowledge points across seven key subdomains critical to cybersecurity experts. This framework offers a comprehensive evaluation of LLMs’ understanding of cybersecurity. The benchmark includes 11,050 high-quality multiple-choice questions, with 772 man-hours spent on review and correction, and \$234.5 allocated for question generation. We release our framework <sup>1</sup> to provide the community with the tools to assess emerging LLMs and conveniently track their progress in mastering cybersecurity expertise.

- *New findings.* We evaluate 12 popular LLMs using CSEBenchmark, incurring a total of 1.08 GPU-weeks and costing \$2140.01. The results indicate that current LLMs still fall short of fulfilling the role of a cybersecurity expert, particularly in handling specialized tools and uncommon commands. By addressing these knowledge gaps, we achieve an improvement of up to 84% in correcting previously incorrect predictions across three existing cybersecurity evaluation datasets, validating the effectiveness of our findings. Lastly, we assess the job-role knowledge alignment of LLMs across six real-world cybersecurity job roles, revealing that LLMs struggle to fully meet these roles’ requirements. Different LLMs show varying degrees of suitability, suggesting that model selection should be tailored to specific task demands.

## 2. Background and Related Work

### 2.1. Large Language Model

Large language models (LLMs) have seen rapid development, leading to significant advancements in natural language processing and understanding. These models, such as OpenAI’s GPT series and Meta’s Llama, are capable of handling a variety of tasks including translation, summarization, content generation, and question answering. Techniques such as Zero-shot learning enable LLMs to approach new tasks without specific training examples [27], while Few-shot learning allows them to adapt quickly with minimal examples [28]. Additionally, Chain-of-Thought (CoT) reasoning enhances complex problem solving by guiding models to break down multi-step tasks logically, yielding clearer and more accurate responses [29]. These capabilities make LLMs highly versatile, finding use in applications such as customer service chatbots, virtual assistants, content recommendation systems, and creative writing. Their flexibility and adaptability have made them useful in business, education, healthcare, and more. Their ability to process and generate human-like text has made them increasingly popular across various fields, sparking interest in their potential to assist or even replace human experts.

1. <https://github.com/NASP-THU/CSEBenchmark>

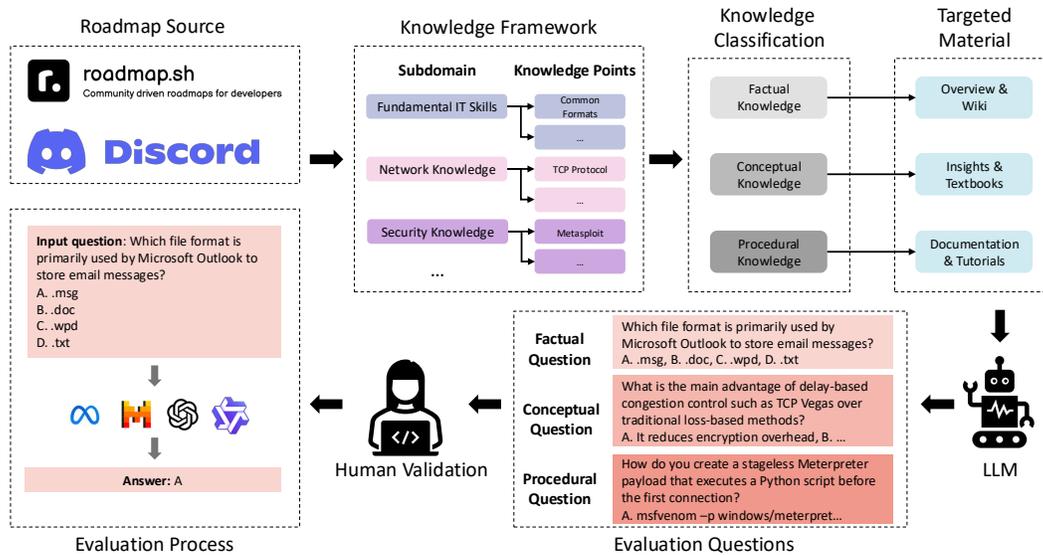


Figure 1. Overview of the construction process of CSEBenchmark.

In cybersecurity, LLMs have started to demonstrate their value in helping with complex tasks that were traditionally performed by experts. For example, LLMs have been applied to support threat intelligence analysis by gathering, processing, and summarizing threat data from multiple sources, helping analysts identify potential risks more efficiently [11], [30]–[34]. In incident response, LLMs help by providing real-time recommendations, generating response playbooks, and analyzing incident logs to determine the root cause of security breaches [35]–[38]. For vulnerability assessment, they help by scanning codebases for known vulnerabilities [39]–[42], suggesting patches [9], [43]–[45], predicting potential weaknesses based on historical data [46], [47], and performing reverse engineering to identify hidden or complex vulnerabilities [48]–[51]. Additionally, LLMs are used to automate routine security operations such as reading documentation [52]–[54], understanding code [55]–[57], and assisting in vulnerability management [58], [59], which significantly reduces the workload for security teams. Despite these advancements, the question remains: how far have we progressed towards developing LLMs that can fully assume expert roles in cybersecurity?

## 2.2. Evaluation of LLMs in Cybersecurity

Evaluating LLMs involves assessing their capabilities to meet specific standards and effectively perform targeted tasks. These evaluations are generally divided into task-based and knowledge-based assessments. Task-based assessments, on the one hand, evaluate the model’s ability to perform cybersecurity-related tasks, such as analyzing threat intelligence, managing vulnerabilities or generating secure code. These assessments typically involve end-to-end tasks framed within real-world security scenarios. For example, in threat intelligence analysis, LLMs are primarily required to analyze real-world threat intelligence reports, assessing

their capabilities in named entity recognition, intelligence classification, summarization, and attribution [3]–[7]. Similarly, evaluations in vulnerability management typically provide carefully selected code snippets, requiring LLMs to comprehend code, debug, generate unit tests, identify vulnerabilities, and apply patches to assess their capabilities in each of these areas [8]–[14], [18]. In secure code generation, LLMs are tasked with generating diverse code, and their capability in secure coding is assessed by evaluating the security of the code they produce [15]–[17]. Although these task-based evaluations intuitively demonstrate model performance across various tasks, they have limitations in identifying the underlying reasons for results due to the lack of quantification of the knowledge needed for each task, making it challenging to conduct a targeted analysis or identify specific weaknesses in the models.

Knowledge-based assessments, on the other hand, gauge a model’s understanding of specialized cybersecurity domains, often through multiple-choice questions (MCQs) generated from relevant materials. For example, SecQA [19] generates approximately 200 questions from the book “*Computer Systems Security: Planning for Success*” to assess security principles knowledge. CyberMetric [20] and SecEval [23] use 10,000 and 2,126 questions, respectively, drawn from textbooks, documentation, and industry guides to assess expertise across areas such as penetration testing, cryptography, and network security. CTIBench [3] generates 2,500 questions from CTI frameworks, regulations, and public resources to evaluate knowledge in cyber threat intelligence. CyberPal.AI [21] builds on CTIBench, SecEval, and other publicly available questions, such as CISSP Assessment Questions and SecMMLU, to evaluate a broader range of LLM knowledge. Likewise, SECURE [22] tests knowledge in cybersecurity advisory through 2,036 questions based on MITRE ATT&CK and CWE. Despite these

efforts, existing studies only assess LLMs based on fragmented knowledge and lack a comprehensive model of the knowledge and skills needed by a cybersecurity expert. Consequently, these assessments do not address the questions posed in this paper. To bridge these gaps, this paper introduces a comprehensive assessment framework involving 345 knowledge points across 7 subdomains, with 11,050 high-quality questions specifically designed to evaluate LLMs' cybersecurity capabilities.

### 3. CSEBenchmark

This paper introduces a cognitive science-based cybersecurity expert knowledge framework, which forms the foundation of CESBenchmark, the first evaluation dataset designed to assess the capabilities of LLMs in progressing toward a digital cybersecurity expert. The construction process is shown in Figure 1, which is divided into four steps: developing the knowledge framework (Section 3.1), classifying the knowledge points (Section 3.2), collecting targeted materials and generating questions based on the classified knowledge points (Section 3.3), and validating and correcting the generated questions (Section 3.4).

#### 3.1. Knowledge Framework

To evaluate whether LLMs can function as digital cybersecurity experts, we need to assess whether they possess the knowledge that a human cybersecurity expert should have, which is often documented in roadmaps. A roadmap is a structured guide that outlines the essential skills and knowledge required for a particular role. In this study, we select the well-known community-driven roadmap website, *roadmap.sh*, as our source. This project has gained 295k stars on GitHub and provides a detailed overview of the skills and knowledge needed for various roles in the IT industry. We use the *Cybersecurity Expert Roadmap* [24] and the *Ethical Hacking Roadmap* [25] as the basis for the CSEBenchmark knowledge framework. Additionally, we supplement our framework with the roadmap titled “*From Power Button to PWN: A Roadmap to Computer Security*,” [26] collected from *Hacking & Coding Discord* communities.

Based on these three roadmaps, we develop a cybersecurity expert knowledge framework, as illustrated in Listing 1. This framework consists of seven subdomains, each representing a key area of expertise for cybersecurity professionals: Fundamental IT Skills (FIS), Operating Systems (OS), Networking Knowledge (NK), Web Knowledge (WK), Security Skills and Knowledge (SSK), Cloud Skills and Knowledge (CSK), and Programming Skills and Knowledge (PSK). Each subdomain is organized into a hierarchical tree structure, with knowledge points arranged by level, culminating in 345 leaf nodes that represent the most specific knowledge points. This structure enables a fine-grained assessment of cybersecurity experts, offering a comprehensive depiction of the core knowledge required in the field.

#### 3.2. Knowledge Classification

As discussed previously, different types of knowledge require varying levels of mastery. Cybersecurity, as an interdisciplinary field, spans both theoretical and practical domains. It encompasses knowledge points that include factual content to be memorized, concepts that require deep understanding, and skills that require hands-on practice. This framework aligns well with the cognitive science knowledge

```
{
  "Cyber Security": {
    "Fundamental IT Skills": {
      "Common computer formats": {
        "label": "factual"
      }, ...
    },
    "Operating Systems": {
      "Windows": {
        "User management in Windows": {
          "label": "conceptual"
        }, ...
      }, ...
    },
    "Networking Knowledge": {
      "Understand Common Protocols": {
        "TCP": {
          "label": "conceptual"
        }, ...
      }, ...
    },
    "Web Knowledge": {
      "SQL": {
        "label": "procedural",
      }, ...
    },
    "Security Skills and Knowledge": {
      "Footprinting and Reconnaissance": {
        "Google Dorks": {
          "label": "procedural"
        }, ...
      }, ...
    },
    "Cloud Skills and Knowledge": {
      "IaaS": {
        "label": "conceptual"
      }, ...
    },
    "Programming Skills and Knowledge": {
      "Python": {
        "label": "procedural"
      }, ...
    }
  }
}
```

Listing 1. Example of the knowledge framework.

classification theory [60], which serves as the basis for categorizing cybersecurity knowledge in this study into factual, conceptual, and procedural types. These categories correspond to specific information, theoretical understanding, and practical skills, respectively. This classification enables a more nuanced evaluation of knowledge mastery, allowing an accurate and tailored assessment of each knowledge point.

To classify the 345 knowledge points in the CSEbenchmark knowledge framework, we invite two cybersecurity practitioners to label each point based on their understanding of the required level of mastery. When disagreements arise, a more experienced cybersecurity expert is consulted for a

final decision. This process results in 121 factual knowledge points, 136 conceptual knowledge points, and 88 procedural knowledge points, with examples shown in Listing 1. These labels reflect practitioners’ views on the necessary level of understanding for each knowledge point, making the CSEBenchmark more aligned with real-world practices.

### 3.3. Question Generation

After completing the knowledge classification, it is essential to generate targeted questions suited to each type of knowledge. First, we need to collect targeted material: for factual knowledge, brief descriptions from the roadmap or relevant wiki entries serve as primary sources for question generation, as factual knowledge mainly requires recall, and these sources provide direct, relevant content. For conceptual knowledge, we select insights from reputable websites or content sourced from textbooks, as these materials often include the author’s understanding of the knowledge points, which help assess the test subject’s deeper understanding of the concepts. For procedural knowledge, official documentation or tutorials are referenced, since they outline practical steps, meeting the needs for evaluating proficiency in hands-on tasks. Following these criteria, we manually collected the most relevant English material entry for each knowledge point to support effective question generation. We use the *pymupdf4llm* [61] library to convert PDFs to markdown format and manually preprocess the material to remove irrelevant text, such as image references, while restoring the original chapter structure information for use in subsequent steps.

After collecting targeted materials, we utilize an LLM to automatically generate questions from them, producing one correct answer and three distractors for each question. Specifically, we use the GPT-4-turbo model for question generation, given its strong performance in text processing. To help the model accurately grasp the characteristics of different knowledge types, we first define the question for each knowledge category in the prompt (see Table 1). These definitions clarify the focus of the questions across knowledge types, ensuring that the model accurately reflects the unique attributes of each type. To further guide the model, we provide eight human-generated sample questions for each knowledge type, helping it recognize the distinct characteristics of each category and avoid misclassification. When generating questions, we explicitly specify the relevant knowledge type and emphasize the exclusion of unrelated categories. The model then selects the correct answer from the provided material and generates three distractors, ensuring the questions meet our expectations.

Due to the limited input window of the LLM, it cannot process all of the materials at once. Additionally, overly lengthy material may lead the model to overlook important details, necessitating the division of the material. The conventional approach involves setting a token threshold and splitting the material into smaller segments [20]. However, this method may disrupt the structure of the material, resulting in a loss of contextual information. To avoid this

TABLE 1. DEFINITIONS FOR QUESTION GENERATION ACROSS DIFFERENT KNOWLEDGE TYPES.

Type	Definition
Factual	Multiple-choice questions focusing on factual knowledge emphasize memory and recall.
Conceptual	Multiple-choice questions focusing on conceptual knowledge emphasize understanding and applying abstract concepts
Procedural	Multiple-choice questions focusing on procedural knowledge emphasize the mastery of specific operational steps and procedural skills, particularly in the context of solving targeted problems within defined scenarios.

issue, we divide the material according to its chapter structure, ensuring that each section retains complete contextual integrity after segmentation.

We observe that materials of the same length may differ in information density. For materials with a higher information density, a greater number of questions should be generated, while for those with lower information density, fewer questions are appropriate. An inappropriate number of questions could lead to repetition or inadequate coverage of the material. Therefore, we aim to quantify the information density of the material and adaptively determine the number of questions to generate. Specifically, we define information density as the number of topics, reframing the task of quantifying information density as a topic extraction problem—a task easily handled by the LLM. In the prompt, we instruct the LLM to first identify all topics and then generate five questions per topic, achieving an adaptive match between the number of questions and the information density of the material.

We generate a total of 11,743 questions for 345 knowledge points. To eliminate the impact of duplicate questions, we apply Semantic Textual Similarity for deduplication. We use SentenceTransformers [62] to convert questions into vectors and apply a similarity threshold of 0.85, validated experimentally for accuracy, to identify and remove duplicates. When duplicates are detected, only the earlier occurrence is retained. Following the question generation process, we obtain a final set of 11,468 unique questions, incurring a total cost of \$234.5.

### 3.4. Dataset Validation and Correction

Due to the well-known issue of hallucination [63], questions generated by the LLM are not always reliable. To address this, we conduct manual validation and correction of the 11,468 deduplicated questions. Specifically, we engage human annotators with cybersecurity expertise to answer each question without access to the original material, avoiding the potential influence of any inaccuracies in the source content. When discrepancies arise between the expert responses and LLM-produced answers, a senior cybersecurity expert conducts a secondary review to ensure accuracy. The entire validation process takes a total of 672 man-hours.

During validation, we find that 1,726 questions exhibit the following issues: (1) 384 questions contain incorrect

answers; (2) 298 questions have multiple correct options; (3) 261 questions lack context in the question stem, resulting in incomplete or hard-to-understand questions; (4) 7 questions display a mismatch in question type; (5) 397 questions show weak relevance to the knowledge point; (6) 216 questions have low-quality distractors that are overly simple or obvious; (7) 14 questions are duplicates of other questions, despite having passed initial similarity checks; and (8) 149 questions lack a correct option. We attempt to manually correct these problematic questions. For issue (1), we replace the incorrect answer directly. For issues (2) and (6), we use the LLM to generate three similar but incorrect options based on the correct answer. For issues (3) and (8), we replace the correct answer or add the missing context based on annotators’ feedback. For issues (4), (5), and (7), we remove these questions as they do not contribute to an accurate assessment. In total, we successfully corrected 1,308 problematic questions, enhancing the CSEBenchmark dataset.

TABLE 2. DISTRIBUTION OF KNOWLEDGE POINTS AND QUESTIONS ACROSS SUBDOMAINS IN THE CSEBENCHMARK DATASET.

Subdomain	Type	#Knowledge	#Tokens	#Questions
FIS	Factual	21	19.8K	124
	Conceptual	2	3.3K	12
	Procedural	2	18.7K	25
OS	Factual	5	8.4K	25
	Conceptual	18	0.3M	433
	Procedural	16	0.4M	650
NK	Factual	30	14.9K	168
	Conceptual	31	0.6M	757
	Procedural	12	93.2K	140
WK	Factual	0	0	0
	Conceptual	0	0	0
	Procedural	6	1.8M	2202
SSK	Factual	50	22.2K	268
	Conceptual	79	0.9M	1040
	Procedural	46	2.0M	2451
CSK	Factual	15	15.7K	75
	Conceptual	6	91.3K	144
	Procedural	0	0	0
PSK	Factual	0	0	0
	Conceptual	0	0	0
	Procedural	6	2.0M	2536
<b>Count</b>		345	8.4M	11,050

The finalized CSEBenchmark dataset comprises 11,050 high-quality multiple-choice questions, covering seven subdomains. The distribution of question types and quantities is shown in Table 2. Notably, the distribution of knowledge points and questions exhibits a skew, primarily driven by two factors: inherent variations in knowledge point distribution across subdomains, which stem from the roadmap design, and the uneven distribution of questions, which correlates with the token count in each corpus, as larger corpus naturally encompass a greater number of topics.

## 4. Experimental Investigation

### 4.1. Experiment Settings

**LLM selection and configuration.** In this study, we select 12 state-of-the-art LLMs for evaluation, as shown in Table 3. These models have demonstrated strong performance in text processing and are widely applied across various tasks. The selected models include both popular open-source models and several commercial closed-source models, with parameter scales ranging from 3B to 671B, reflecting the cybersecurity knowledge capabilities of models at different scales. Specially, we introduce a mixture-of-experts (MoE) model, Mixtral 8×7B, which consists of 8 experts, each with 7B parameters, totaling approximately 45B parameters. We also introduce an inference model, Deepseek-R1, which is trained on Deepseek-V3 and, unlike other models, autonomously generates its own chain of thought, systematically deducing intermediate steps to ensure accurate reasoning and logical coherence. For OpenAI and Deepseek models, we access them via their respective APIs [64], [65], while for other open-source models, we use the OpenAI-Compatible Server from *vLLM* [66] to ensure code consistency. To assess the knowledge levels of these models more precisely, we set the temperature parameter to 0.2, which is commonly used in precision tasks [64], to minimize the influence of random output on evaluation results.

TABLE 3. SELECTED LLMs IN THIS STUDY.

Model Name	#Params	Cutoff Date	Type
GPT-3.5-Turbo-0125	175B	2021-09	Closed
GPT-4-Turbo-2024-04-09	Unk.	2023-12	Closed
GPT-4o-2024-08-06	Unk.	2023-10	Closed
Llama-3.2-3B-Instruct	3B	2023-12	Open
Llama-3.1-8B-Instruct	8B	2023-12	Open
Llama-3.1-70B-Instruct	70B	2023-12	Open
Mixtral-8x7B-Instruct-v0.1	45B	2023-12	Open
Qwen-2.5-3B-Instruct	3B	2023-02	Open
Qwen-2.5-7B-Instruct	7B	2023-02	Open
Qwen-2.5-72B-Instruct	72B	2023-02	Open
Deepseek-V3-241226	671B	Unk.	Open
Deepseek-R1-250120	671B	Unk.	Open

**Platform.** The experiments are conducted on a platform with an Intel(R) Xeon(R) Platinum 8468 processor, 2.0 TB RAM, 172 cores and 8 NVIDIA H100 GPUs with 80 GB HBM3 each. The entire experiment requires a total of 1.08 GPU-weeks.

**Experiment Setup.** Recognizing that different prompts can influence how the models activate their embedded knowledge, we employ three interaction methods—Zero-shot, Few-shot, and CoT—in our experiments to minimize the impact of these prompting techniques on the models’ output<sup>2</sup>. For each question, we use the highest score from the three prompting methods as the final result, representing the actual knowledge ceiling that the model can achieve. In the Zero-shot method, we provide questions directly

2. For Deepseek-R1, since it inherently incorporates the CoT method, we only use the CoT approach.

without any examples, asking the model to produce results independently. For the Few-shot method, we build on the Zero-shot approach by providing 5 example question-answer pairs that are not included in the dataset; this 5-shot strategy is widely used in related research [6], [19]. Finally, in the CoT method, we use the common prompt, “*Let’s think step by step,*” to guide the model’s reasoning process. Full prompts are provided in the Appendix A.

**Measurement Method.** To reduce the impact of LLM randomness on the evaluation results, we have the model perform five independent inferences for each question, considering the response correct only if all of the inferences yield the correct answer. Additionally, to avoid any preference the model may have for specific options, we systematically rotate the correct answer across the four choices (A, B, C, D) and evaluate each arrangement independently. We consider the model to have truly mastered a knowledge point only if it answers correctly in all four arrangements, indicating that its success is due to understanding rather than guessing.

Given that LLM outputs are in the loose format of natural language text, we need to extract the exact options selected by the models. A common approach is to evaluate the probability of the first token in the model output [67], [68]; however, recent research indicates that this method lacks robustness [69], [70]. Therefore, we follow their recommendations to extract the model’s selected answers from the original responses. Specifically, we use the xFinder-llama38it model for option extraction, a state-of-the-art model for identifying multiple-choice answers, which has demonstrated 95.47% accuracy on generalization sets [71]. We randomly sample 4782 original responses for manual verification, finding an actual accuracy of 92.47% for this extraction process, which supports the validity of the results presented in this study.

**Evaluation Metrics.** We use the accuracy for all questions associated with each knowledge point as our evaluation metric, categorizing accuracy into four ranges: 100% indicates that the LLMs have fully mastered the knowledge point, meeting the level expected of cybersecurity experts; [90%, 100%) suggests that LLMs are approaching expert-level understanding; [80%, 90%) indicates partial mastery with room for improvement; and below 80% reflects poor performance, indicating areas that require focused attention.

**Research Question.** In the following subsections, we evaluate the performance of the selected 12 state-of-the-art LLMs in the CSEBenchmark, with a primary focus on the following research questions:

**RQ1.** Do the selected LLMs possess the knowledge expected of cybersecurity experts?

**RQ2.** What knowledge gaps remain in the selected LLMs when positioned as cybersecurity experts?

**RQ3.** Can the results of CSEBenchmark help improve LLM performance in cybersecurity tasks?

**RQ4.** How well do the selected LLMs align with real-world cybersecurity job roles?

## 4.2. LLM Cybersecurity Expertise Assessment (RQ1)

Table 4 presents the accuracy performance of the 12 selected LLMs on CSEBenchmark. Overall, GPT-4o ranks first with an accuracy of 85.42%, followed closely by Deepseek-V3 at 84.92% and Qwen-2.5-72B at 84.40%, with less than a 1.2% difference among the top three models. GPT-4-Turbo follows in fourth place at 83.86%<sup>3</sup>. Deepseek-R1 and Llama-3.1-70B achieve 80.62% and 80.00%, respectively. The remaining models show a larger performance gap of over 5% compared to the top six, with the rankings as follows: Qwen-2.5-7B (74.90%), Mixtral-8×7B (73.58%), GPT-3.5-Turbo (68.44%), Llama-3.1-8B (69.30%), Qwen-2.5-3B (68.07%), and Llama-3.2-3B (52.95%). Notably, GPT-4o not only performs well in terms of accuracy but also operates at just 30% of the cost of GPT-4-Turbo, making it a preferred choice among closed-source LLMs for cybersecurity expert scenarios. Among open-source LLMs, Deepseek-V3 performs the best, coming close to the top-performing GPT-4o. Due to its open-source nature, Deepseek-V3 also offers greater scalability and practicality. Notably, although Qwen-2.5-72B’s accuracy is slightly lower than Deepseek-V3 (0.6%), its substantially smaller model size (72B vs. 671B) makes it a more cost-effective and practical choice for real-world applications. We observe that the Qwen-2.5 series consistently outperforms the Llama-3.1 and Llama-3.2 series of similar parameter scales. Additionally, the Mixtral-8×7B MoE model lags behind the single 7B model, Qwen-2.5-7B. Although the MoE structure is theoretically designed to enhance performance through specialized expert modules, it does not show a significant advantage in this evaluation, suggesting that the multi-expert mechanism has limited effectiveness for knowledge tasks in this context. We also observe that, despite Deepseek-R1’s strong reasoning capabilities, it does not exhibit an advantage in the safety knowledge evaluation. Its overall accuracy is even lower than that of its training base, Deepseek-V3. This suggests that in knowledge tasks, strong reasoning ability may not necessarily compensate for precise knowledge recall and retrieval. Over-reliance on reasoning could instead lead to information distortion or misjudgment.

### Finding 1

GPT-4o is the best-performing LLM overall, while Deepseek-V3 leads among open-source options. However, even these top LLMs cover only 85.42% of the knowledge required by cybersecurity experts.

In all subdomains, GPT-4o performs best in three—OS (82.67%), WK (86.15%), SSK (80.26%), CSK (97.26%), and PSK (89.04%)—while Deepseek-V3 leads in CSK (97.72%) and PSK (89.87%), and Qwen-2.5-72B leads in FIS (96.27%) and NK (92.58%). Although current LLMs

3. Note that since GPT-4-Turbo is also used for generating the questions, its results may involve cyclical use, as discussed in Section 5.1.

TABLE 4. ACCURACY OF THE TESTED LLMs ACROSS SEVEN SUBDOMAINS AND THREE KNOWLEDGE CATEGORIES (ACRONYMS USED).

Type	Label	GPT-3.5T	GPT-4T	GPT-4o	L3.1-8B	L3.1-70B	L3.2-3B	M-8x7B	Q2.5-3B	Q2.5-7B	Q2.5-72B	DS-V3	DS-R1
Subdomain	FIS	87.58	92.55	95.65	88.20	91.30	80.75	86.34	87.58	91.30	<b>96.27</b>	93.79	91.93
	OS	61.91	80.60	<b>82.67</b>	64.08	74.37	48.83	69.95	65.25	69.58	80.60	81.32	79.87
	NK	81.03	91.46	92.39	83.19	88.64	70.61	84.32	79.72	87.23	<b>92.58</b>	91.92	89.86
	WK	67.94	84.74	<b>86.15</b>	67.71	80.79	49.41	72.48	64.80	72.93	84.11	84.65	79.16
	SSK	62.92	78.21	<b>80.26</b>	65.28	74.57	51.74	68.79	65.79	70.44	79.76	79.70	74.79
	CSK	87.67	95.89	97.26	92.24	95.43	83.56	92.24	88.13	93.61	96.35	<b>97.72</b>	96.80
	PSK	71.77	88.13	89.04	69.91	84.15	47.79	76.30	67.67	77.68	87.97	<b>89.87</b>	85.29
Category	Fact.	86.82	93.64	<b>94.85</b>	86.06	92.42	80.00	88.33	87.58	90.45	94.24	94.24	91.67
	Conc.	86.25	93.88	<b>94.84</b>	88.60	93.34	78.54	89.52	86.34	91.32	94.59	94.26	92.58
	Proc.	61.62	80.07	<b>81.83</b>	62.17	75.00	43.09	67.62	61.02	68.72	80.55	81.37	76.14
Overall		68.44	83.86	<b>85.42</b>	69.30	80.00	52.95	73.58	68.07	74.90	84.40	84.92	80.62

do not fully meet the knowledge requirements of security experts, their highest accuracies exceed 90% in the FIS, NK, and CSK subdomains, indicating that their knowledge in these areas is approaching cybersecurity experts. Figure 2 presents a box plot of LLM accuracy across each subdomain. In the FIS and CSK subdomains, all LLMs achieve accuracies above 80%, with a median of 91%, indicating that the tested LLMs are generally approaching the knowledge level of cybersecurity experts in these areas. In the NK subdomain, LLM performance varies widely, with the lowest accuracy at 71% and a median of 88%. Although the top-performing LLMs exceed 90% accuracy in this subdomain, most LLMs still have substantial room for improvement in knowledge coverage. In the OS, WK, SSK, and PSK subdomains, accuracy differences among LLMs increase significantly, with the lowest accuracy falling below 51% and a median slightly above 72%, indicating lower knowledge levels in these subdomains.

**Finding 2**

LLMs have not yet fully met the knowledge requirements of cybersecurity experts in any subdomain. However, their knowledge in the FIS, NK, and CSK subdomains is close to the expert level, while significant improvement is needed in the OS, WK, SSK, and PSK subdomains.

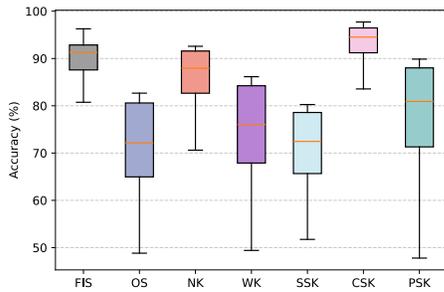


Figure 2. Accuracy distribution of LLMs across subdomains.

We also evaluate the accuracy of the tested LLMs across three knowledge categories, with results presented in Ta-

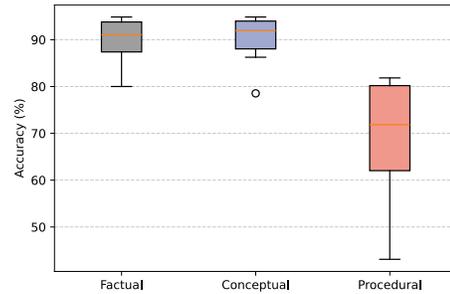


Figure 3. Accuracy distribution of LLMs across knowledge categories.

ble 4. GPT-4o achieves the highest accuracy across all three categories, at 94.85%, 94.84%, and 81.83%, respectively, with the ranking of the other models remaining largely consistent with their overall performance. The accuracy distribution across these categories is illustrated in the box plot in Figure 3. In the Factual and Conceptual categories, the accuracy of LLMs is relatively concentrated, with almost all models achieving close to 80% accuracy and a median close to 92%, indicating that LLMs are adept at mastering these types of knowledge. This may be because factual and conceptual knowledge often appears in direct statements or explanatory forms within the training corpus, allowing models to extract and retain information from context more effectively. In contrast, the accuracy drops significantly for procedural knowledge, with the lowest accuracy at only 43.09% and a median of 71.86%. This discrepancy likely arises because LLM pretraining is not tailored to reinforce real-world cybersecurity operations or procedural tasks, making it challenging for models to develop a deep understanding and flexible application of complex operations from the corpus alone. Given that cybersecurity heavily relies on practical skills, this limitation presents a significant obstacle for LLMs to become cybersecurity experts.

**Finding 3**

LLMs demonstrate a good grasp of factual and conceptual knowledge, but perform poorly in procedural knowledge.

We conduct a fine-grained evaluation of LLM performance across 345 knowledge points, with the results displayed as a heatmap in Figure 4. In the heatmap, each row represents the accuracy of different LLMs on the same knowledge point, while each column shows the performance of the same LLM across various knowledge points. Among the 345 knowledge points, certain LLMs achieve 100% accuracy on 241 points, indicating that LLMs meet the knowledge standards of security experts for these points. Additionally, on 35 knowledge points, certain LLMs reach an accuracy above 90%, suggesting that LLMs are approaching expert-level knowledge in these areas. Of these 276 knowledge points, 230 are factual or conceptual knowledge, accounting for 83.33%, further confirming the strong performance of LLMs in these knowledge types. The remaining 46 knowledge points are procedural, focusing on essential operations and troubleshooting for operating systems and network tools. These include troubleshooting strategies, error interpretation, software installation on Linux, MacOS, and Windows, basic commands (e.g., *ping*, *netstat*), log analysis, file manipulation (e.g., *cat*, *grep*), and scripting languages (e.g., *Python*, *JavaScript*). Although these procedural knowledge points involve a degree of practical skill, their high frequency in real-world tasks means their fixed syntax and relatively simple logic are well-represented in pretraining data, enabling LLMs to achieve high accuracy on these points.

#### Finding 4

LLMs achieve the expected level of cybersecurity expertise on 241 knowledge points and approach expert-level performance on an additional 35 points, covering 80.0% of all points. These are primarily factual and conceptual knowledge, along with some high-frequency procedural knowledge.

### 4.3. LLM Knowledge Gap Assessment (RQ2)

As mentioned above, LLMs meet or approach the knowledge requirements of cybersecurity experts on 276 knowledge points, but notable knowledge gaps remain on the other 69 points. Benefiting from the fine-grained design of knowledge points in CSEBenchmark, we are able to analyze these specific knowledge gaps in each LLM in greater detail than existing studies that rely solely on overall score evaluations. Among these 69 knowledge points, 40 have accuracies between 80% and 90%, indicating that LLMs have a partial grasp of these points but still have room for improvement. Of these, 11 are factual knowledge points, covering topics like basic coding, operating system version differences, threat intelligence, authentication methods, and security models. Another 11 are conceptual knowledge points, addressing core security concepts and network protocols, such as *MacOS permissions management*, *DNS*, *VPNs*, and *DDoS attacks*. The remaining 18 points are procedural knowledge, primarily involving system operations, common

commands, and tool applications, such as installation and configuration in Linux and Windows, network scanning tools (e.g., *nmap*), log analysis (e.g., *event logs*, *packet captures*), introductory reverse engineering, and scripting and programming languages (e.g., *Bash*, *PowerShell*).

There are 29 knowledge points where the highest accuracy achieved by any LLM remains below 80%, indicating substantial room for improvement in these areas. Of these, 4 are factual knowledge (*P2P*, *Local Auth*, *VirusTotal*, and *Sandboxing*) and 1 is conceptual knowledge (*Brute Force vs Password Spray*). We observe that, although these points appear straightforward, LLMs still struggle with them. For instance, one question on *Local Auth* is: “What additional security measure is recommended to enhance the security of a system using local authentication? A. Use of SSL B. Centralized user management C. Cloud-based authentication D. Reduction of password strength.” The correct answer is A. However, when the position of the correct answer is shuffled with other options, LLMs often select the wrong answer, indicating that the model’s understanding of this knowledge point is not solid. The remaining 24 points are procedural knowledge, involving the use of cybersecurity and forensic tools, including common Windows commands, SQL, Kali Linux, network analysis tools (e.g., *netflow*, *Wireshark*), forensic tools (e.g., *FTK Imager*, *Autopsy*, *memdump*, *winhex*), exploitation frameworks (e.g., *Exploit Pack*, *Metasploit*), social engineering tools (e.g., *Social-Engineer Toolkit*), wireless security tools (e.g., *Aircrack-ng*), penetration testing tools (e.g., *Burp Suite*, *John the Ripper*, *Nikto*, *OpenVAS*), system information gathering tools (e.g., *enum4linux*), and malicious command libraries (e.g., *GT-FOBINS*, *LOLBAS*, *WADCOMS*). Compared to more commonly encountered tools mentioned above (e.g., *cat* and *grep*), these points are more specialized and have unique application contexts, resulting in lower representation in pretraining corpora and making it challenging for LLMs to effectively learn and master them.

#### Finding 5

Overall, LLMs show notable gaps in nuanced procedural knowledge involving specialized tools and uncommon commands, even struggling with certain straightforward factual and conceptual points.

We further analyze the knowledge gaps in each LLM, with the accuracy distribution across all knowledge points shown in Figure 5.

**GPT-4o:** As the best-performing LLM overall, GPT-4o achieves 100% accuracy on 200 knowledge points and exceeds 90% accuracy on an additional 42 points, covering 70.14% of all knowledge points. However, its accuracy falls below 80% on 47 points, primarily in areas such as foundational concepts (e.g., *Peer-to-Peer (P2P)*, *Private vs Public Keys*), security tool usage (e.g., *VirusTotal*, *Wireshark*, *Metasploit*), attack and defense techniques (e.g., *Brute Force vs Password Spray*), system configuration tasks (e.g., *Com-*

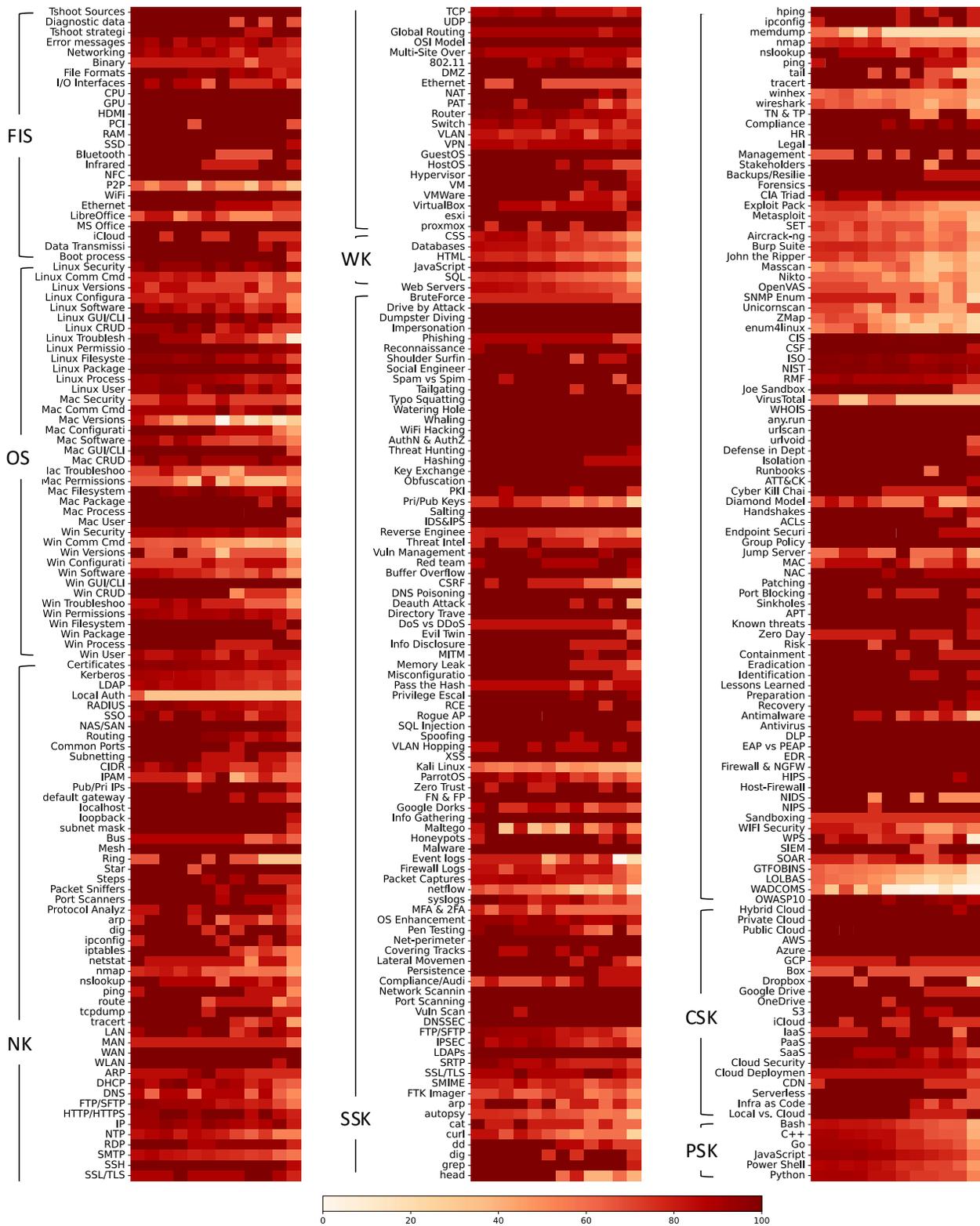


Figure 4. Heatmap of accuracy across 345 knowledge points for 12 models. The y-axis labels denote individual knowledge points, with subdomain names in parentheses for grouped items. Each section contains 12 columns representing models from left to right: GPT-4o, Deepseek-V3, Qwen-2.5-72B, GPT-4-Turbo, Deepseek-R1, Llama-3.1-70B, Qwen-2.5-7B, Mixtral-8x7B, GPT-3.5-Turbo, Llama-3.1-8B, Qwen-2.5-3B, Llama-3.2-3B.



Figure 5. Proportion of knowledge points across four accuracy ranges for each LLM.

mon Commands in Windows), and security management (e.g., SOAR).

**Deepseek-V3:** Deepseek-V3 is the best-performing open-source LLM, achieving 100% accuracy on 209 knowledge points and exceeding 90% accuracy on an additional 31 points, covering 69.57% of all knowledge points. However, the model struggles with 49 knowledge points, particularly system and network fundamentals (e.g., P2P, DNS), authentication (e.g., MFA, Jump Server), security tools (e.g., VirusTotal, Metasploit), penetration testing (e.g., Burp Suite, OpenVAS), system configuration (e.g., Windows commands, MacOS troubleshooting), and data analysis (e.g., SQL, Net-Flow). It also fails to differentiate system versions and privilege escalation techniques (e.g., GTFOBins, LOLBAS), highlighting gaps in practical cybersecurity knowledge.

**Qwen-2.5-72B:** Qwen-2.5-72B also demonstrates strong performance, achieving 100% accuracy on 200 knowledge points and exceeding 90% accuracy on 39 more, covering 69.28% of all knowledge points. However, it struggles with 50 knowledge points, primarily in the following areas: foundational system and network concepts (e.g., Peer-to-Peer (P2P), iCloud), security compliance and management (e.g., Roles of Compliance and Auditors), security tool usage (e.g., VirusTotal, Metasploit), attack and defense techniques (e.g., Brute Force vs Password Spray), system configuration tasks (e.g., Common Commands in Linux), and basic programming and data query tools (e.g., SQL, Google Dorks).

**GPT-4-Turbo:** Ranking third overall, GPT-4-Turbo covers the most knowledge points with 100% accuracy, achieving perfect scores on 207 points, and over 90% accuracy on an additional 28 points, totaling 68.12% of all knowledge points. However, the model's accuracy falls below 80% on 44 points, mainly in the following areas: foundational system and access management concepts (e.g., Peer-to-Peer (P2P), Local Auth), roles in security compliance and management, cryptography and authentication mechanisms (e.g., WPA vs WPA2 vs WPA3 vs WEP, Brute Force vs Password Spray), security tool usage (e.g., VirusTotal, Metasploit), basic programming and data query tools (e.g., SQL, Google Dorks), and system configuration tasks (e.g., Common Commands in Windows).

**Deepseek-R1:** While Deepseek-R1 excels in reasoning, its

performance in security knowledge assessment is less remarkable. It achieves 100% accuracy on 183 knowledge points and exceeds 90% on 31 more, covering 62.03% of the total. However, it falls short on 68 knowledge points, particularly in authentication and access control (e.g., MFA & 2FA, Jump Server), network security (e.g., NIDS, VLAN, DNS), security tools (e.g., VirusTotal, Metasploit, Wireshark), penetration testing (e.g., Aircrack-ng, OpenVAS, Masscan), system administration (e.g., Linux installation, Windows commands), and forensic analysis (e.g., FTK Imager, WinHex).

**Llama-3.1-70B:** Llama-3.1-70B shows a noticeable gap from the top five models, achieving 100% accuracy on only 186 knowledge points, with an additional 33 points exceeding 80% accuracy, covering 63.48% of all knowledge points. The model performs poorly on 63 knowledge points, primarily in the following areas: operating system versions and configuration management (e.g., Different Versions and Differences in Linux, Local Auth), network interfaces and standards (e.g., Ethernet, VLAN), cloud storage and virtualization tools (e.g., iCloud, VirtualBox), security tool usage (e.g., VirusTotal, Metasploit), encryption and authentication mechanisms (e.g., Private vs Public Keys, Brute Force vs Password Spray), and basic programming operations (e.g., SQL, Bash).

**Qwen-2.5-7B:** As the best-performing small model, Qwen-2.5-7B achieves 100% accuracy on 177 knowledge points, with an additional 17 points exceeding 90%, covering 56.23% of all knowledge points. However, the model's accuracy falls below 80% on 82 points, particularly in areas such as network and communication protocols (e.g., Bluetooth, Peer-to-Peer (P2P), Ethernet), operating systems and file management (e.g., Linux version differences, common Windows commands, MacOS troubleshooting), and authentication and security (e.g., MFA&2FA, user permissions management, types of password attacks). Additionally, the model shows weaker recognition and understanding in information gathering and vulnerability scanning tools (e.g., nmap, Masscan, Unicornscan), data forensics and analysis tools (e.g., Wireshark, FTK Imager, Event Logs), and foundational web and database knowledge (e.g., HTML, SQL, Web Servers).

**Other five LLMs:** The remaining 5 LLMs have fewer knowledge points with 100% accuracy or above 90%, with coverage below 50%, indicating that these LLMs fall short of the expected knowledge level for cybersecurity experts on more than half of the points. Among them, Mixtral-7x8B has 27.8% of knowledge points with accuracy below 80%, while GPT-3.5-Turbo, Llama-3.1-8B, and Qwen-2.5-3B each have around 30% of points below this threshold. Llama-3.2-3B performs the worst, with 51% of knowledge points below 80% accuracy. These results suggest that these models are currently insufficient for performing at a cybersecurity expert level.

Interestingly, we observe that different-sized LLMs within the same series also exhibit variations in their knowledge gaps. This suggests that the knowledge gaps of smaller LLMs are not merely a subset of those found in larger

models. In fact, larger models may have gaps in areas where smaller models perform well. For example, Llama-3.1-70B underperforms when using `tcpdump`, while Llama-3.1-8B achieves 100% accuracy on this knowledge point. This highlights the importance of not relying solely on model size when selecting an LLM, but instead considering the specific tasks and knowledge gaps to make a more informed choice.

#### Finding 6

Different LLMs exhibit distinct knowledge gaps as cybersecurity experts. Even smaller models in the same series can sometimes outperform larger ones in specific knowledge points.

### 4.4. Enhancing LLMs Through CSEBenchmark (RQ3)

After identifying the knowledge gaps of each LLM using CSEBenchmark, we attempt to improve their performance based on these gaps. To this end, we focus on two fundamental security tasks—vulnerability detection and threat intelligence analysis—and select three state-of-the-art open-source, task-based evaluation datasets—VuldetectBench [18], SecLLMHolmes [8], and CTI-RCM [3]. VuldetectBench and SecLLMHolmes focus on vulnerability detection, with the former containing 1,000 real-world vulnerability snippets and the latter featuring 15 pairs of CVE code samples before and after patches, tested across four prompting strategies for a total of 120 cases. CTI-RCM includes 1,000 CVE descriptions from 2024, evaluating LLMs’ threat intelligence analysis capabilities by assessing their accuracy in mapping vulnerabilities to their corresponding CWE classifications. To highlight the effectiveness of the improvements made using CSEBenchmark, we choose three models from the relatively lower-performing Llama series—Llama-3.1-8B, Llama-3.1-70B, and Llama-3.2-3B—as subjects for enhancement. Additionally, to assess whether high-performing LLMs can likewise benefit from these enhancements, we include GPT-4o in our experiments.

First, we perform an initial evaluation of the original LLMs on the three assessment datasets and record instances where each model makes incorrect predictions. Next, we extract the knowledge gaps (i.e., knowledge points with an accuracy below 90%) of each model from CSEBenchmark and provide this gap information to the LLMs for a reevaluation of the error instances. The proportion of previously incorrect predictions corrected in the reevaluation reflects the performance improvement of the LLMs after addressing their knowledge gaps. We employ a Retrieval-Augmented Generation (RAG) approach to inject the models with knowledge points related to their knowledge gaps. Specifically, for ease of implementation, we construct a vector database for each LLM using Milvus [72] and use corresponding question-answer pairs from CSEBenchmark to address the model’s knowledge gaps. For embedding, we

utilize the BGE-M3 model [73]. Before issuing the request to the LLMs, we use each dataset’s task instruction to query the vector database, retrieve the top-5 most relevant entries, and incorporate them into the original prompt, with the instruction, “Please use the following retrieved context to answer the question,” effectively addressing the models’ knowledge gaps.

TABLE 5. PERFORMANCE IMPROVEMENT OF LLMs AFTER KNOWLEDGE GAP SUPPLEMENTATION, WITH THE NUMBERS ON EITHER SIDE OF THE ARROW REPRESENTING THE COUNT OF ERROR INSTANCES BEFORE AND AFTER ENHANCEMENT. THE PERCENTAGES REPRESENT THE PROPORTION OF PREVIOUSLY INCORRECT INSTANCES THAT BECAME CORRECT AFTER ENHANCEMENT.

Model	Benchmark		
	VuldetectBench	SecLLMHolmes	CTI-RCM
L3.2-3B	495→108 (78%)	65→45 (31%)	758→701 (8%)
L3.1-8B	373→59 (84%)	59→44 (25%)	434→370 (15%)
L3.1-70B	439→311 (29%)	66→50 (24%)	350→315 (10%)
GPT-4o	405→343 (15%)	73→55 (25%)	248→226 (9%)

The results in Table 5 show that after addressing the knowledge gaps, all LLMs show improvements across the three datasets, confirming that the knowledge gaps identified by CSEBenchmark enhance LLM performance, with the highest improvement reaching 84%. For example, C++ is a knowledge gap for both Llama-3.2-3B and Llama-3.1-8B. The question-answer pairs on pointer operations within the knowledge points, such as “*What should you do to a pointer after deleting the memory it points to, to avoid dangling pointer issues? Set the pointer to nullptr*” and “*To ensure a reference cannot change the bound object, which declaration is appropriate? const int &cri = i*”, help the models better understand the concept of pointer safety, which in turn enable them to correctly identify potential vulnerabilities related to improper pointer operations and memory deallocation in code. Similarly, in CTI-RCM, RAG improves XSS vulnerability classification by providing definitions, enhancing model performance. Furthermore, we find that the retrieved semantically relevant question-answer pairs from the model’s entire knowledge gap may not always precisely match the required knowledge but still contribute to overall performance improvement. For instance, a Go-related null pointer dereferencing question-answer pair helps the model identify a C++ null pointer dereferencing vulnerability in VulDetectBench. Note that RAG technique used in this study is straightforward, and optimizing its design in the future could further enhance LLM performance.

#### Finding 7

The knowledge gaps identified by CSEBenchmark can be used to improve model performance.

### 4.5. LLM Job Role Assessment (RQ4)

Although we assess the selected LLMs on 345 knowledge points, real-world cybersecurity roles typically do not require proficiency at all of these points (though more

coverage is generally beneficial). To evaluate how well these LLMs’ knowledge aligns with the specific requirements of real-world cybersecurity positions, we gather job requirements from companies such as Amazon, Google, and Microsoft. Based on role descriptions, we manually map these requirements to our knowledge points. For example, the Amazon Security Engineer role specifies a requirement for “experience with a focus in areas such as systems, network, and/or application security.” Drawing on our own expertise, we map this requirement to relevant CSEBenchmark knowledge points in system security (e.g., *Linux security concepts*), network security (e.g. *DoS vs DDoS*), and application security (e.g., *Web Based Attacks and OWASP10*) to assess each LLM’s alignment with the core skills needed for this role. In total, we identify six distinct roles for the analysis: Google’s Senior Intelligence Analyst and Red Team Security Consultant, Amazon’s Privacy Engineer, ISC Security Engineer, and Security Engineer, and Microsoft’s Red Team Security Engineer. The mapped knowledge points for each job role is provided in Appendix B.

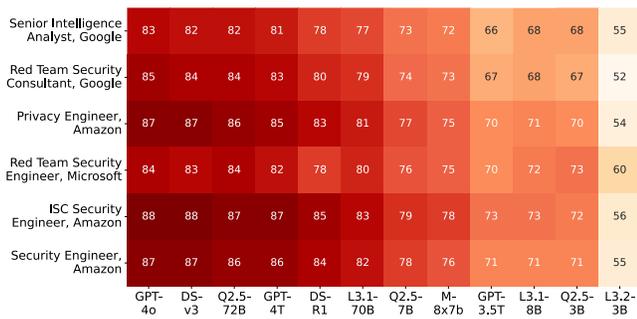


Figure 6. Heatmap of selected LLMs’ match scores across six real-world cybersecurity job roles.

We calculate the overall accuracy of mapped knowledge points as the job-role match score, with the results shown in Figure 6. The ranking is closely aligned with the performance of each model on the CSEBenchmark. GPT-4o achieves the highest knowledge match scores for the Google Senior Intelligence Analyst and the Google Red Team Security Consultant. For the Amazon Privacy Engineer, ISC Security Engineer, and Security Engineer, Deepseek-V3 and GPT-4o share the top position. Similarly, for the Microsoft Red Team Security Engineer role, Qwen-2.5-72B and GPT-4o are tied for first place. Notably, knowledge match scores for even the highest ranked LLMs are below 90%, indicating that these models still do not fully meet the real-world cybersecurity job requirements.

#### Finding 8

LLMs show limited knowledge alignment with cybersecurity job roles in the real world, with the highest match below 90%. Different LLMs exhibit unique strengths aligned with specific roles.

In addition, we group the required competencies for each role into core categories based on job descriptions and create radar charts to visually highlight current gaps for LLMs in each position, as shown in Figure 7. For the Google Senior Intelligence Analyst role, gaps appear in *Cybersecurity Analysis and Incident Response* and *Security Tools*, with top match scores of 77 and 70, respectively. For the Google Red Team Consultant role, the main gap is in *Offensive Security and Red Teaming*, with a maximum score of 79. The roles of the Amazon Privacy Engineer and Security Engineer show gaps in *Incident Response and Security Specializations* and *Security Operations and Incident Response*, with top scores of 76 and 73, respectively. For the Amazon ISC Security Engineer role, LLMs perform more consistently, with scores above 80 across all areas. The Microsoft Red Team Security Engineer role highlights gaps in *Cybersecurity Tools and Technologies* and *Forensics and Reverse Engineering*, with highest scores of 76 and 75.

#### Finding 9

Different cybersecurity roles reveal unique competency gaps for LLMs.

## 5. Discussion

### 5.1. Potential Cyclical Use and Model Bias

We observe that GPT-4-Turbo generates and answers its own questions, which can be seen as cyclical use. However, this does not impact the results in our paper. For other models, no cyclical use occurs, ensuring the validity of their results. Note that GPT-4-Turbo and GPT-4o are distinct models with different training data and methodologies. For GPT-4-Turbo, we believe there is no “unfair cyclical use,” as our carefully designed prompts ensure it solely relies on its summarization capabilities rather than its internal knowledge. To verify this, human experts manually examine 500 randomly selected questions to identify their corresponding source passages within the corpus. The process involves first identifying potential passages by searching for distinctive keywords in each question, followed by a thorough analysis to determine whether the passages contained all key concepts relevant to the question. A passage is considered the source if it fully encompasses these key concepts. In all cases, a corresponding passage is found, confirming that GPT-4-Turbo generates questions exclusively based on the provided material. Since the corpus is not available when answering the questions, no unfair cyclical use occurs, ensuring the credibility of the results.

Additionally, considering the possibility that GPT-4-Turbo might introduce its own preferences when generating questions, potentially leading to bias, we conduct an evaluation to assess topic selection fairness. We randomly select three distinct knowledge points (*Kerberos*, *Packet Sniffer*, and *Nikto*) and asked GPT-4-Turbo, GPT-4o, Llama-3.1-70B, and Qwen-2.5-72B to extract topics from the corpus.

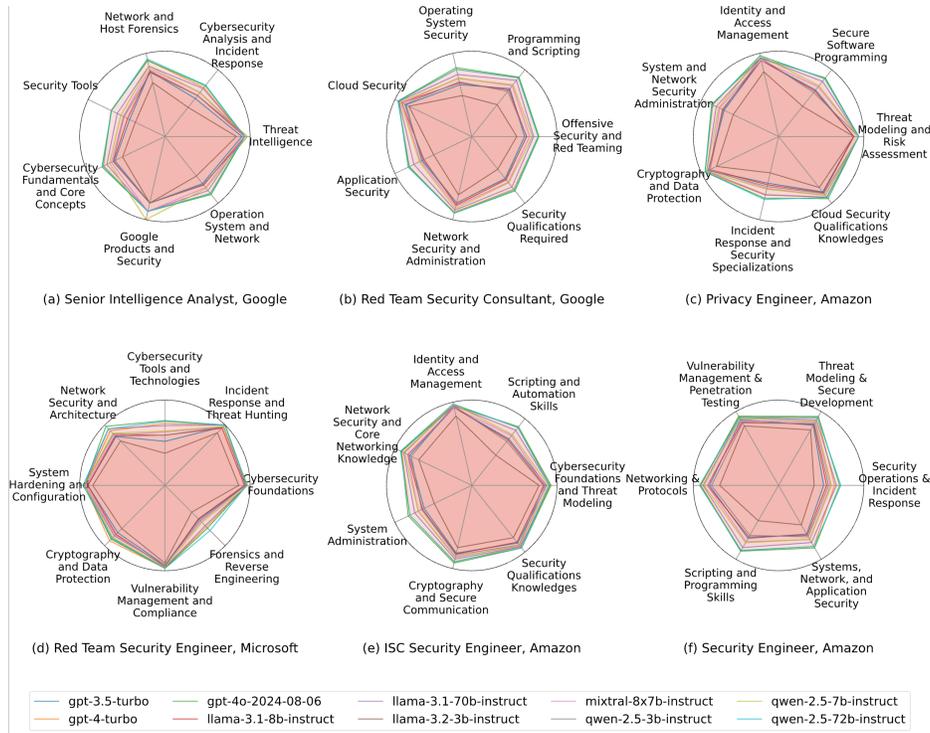


Figure 7. Radar chart showing the alignment of the selected LLMs with the requirements of six real-world cybersecurity job roles.

These topics directly influence the question distribution, as five questions will be generated for each topic. Therefore, any skew in the topic distribution can reflect potential model bias. The topic distribution in semantic space (via BGE-M3) shows consistent results across the four models, with no bias observed (see Appendix C).

## 5.2. Limitation and Future Work

Despite undergoing 772 hours of manual review and correction, the CSEBenchmark still presents certain limitations. First, our knowledge framework, based on three public cybersecurity roadmaps, covers 345 knowledge points of cybersecurity experts. However, some specialized areas, such as hardware security, may be underrepresented. To improve the framework, we plan to expand the knowledge points through interviews with cybersecurity professionals, ensuring that it addresses emerging needs. Second, each knowledge point question in the CSEBenchmark is generated based on a single, most relevant, and official source (e.g., textbooks, reputable websites, or blogs), providing a degree of reliability. However, a single source may sometimes fail to comprehensively cover the full scope of a knowledge point. We plan to address this by supplementing each knowledge point with additional relevant materials. Third, in our evaluation, we employ three commonly used prompting methods—Zero-shot, Few-shot, and CoT—to probe the upper knowledge limits of LLMs, using the highest score as the final result to reveal critical knowledge gaps. However, in practical applications, more advanced prompting techniques

may further improve LLM performance, and we aim to incorporate such advanced techniques for a more thorough assessment of LLM capabilities. Lastly, CSEBenchmark relies on *xFinder* as the back-end technology to extract answers from free text. Compared to regex-based methods, *xFinder* provides substantial accuracy improvements; however, sampling indicates that an error rate of 8% persists. To ensure fair and objective evaluation outcomes, it is necessary to further enhance *xFinder*'s accuracy in future work.

**Impact of Time on Evaluation Results.** Due to varying knowledge cutoff dates, some newer source materials may only appear in the training data of models with later cutoffs. However, our objective is to highlight existing knowledge gaps in LLMs—gaps that may stem from limited training or incomplete data. These gaps are objectively present, regardless of the cause, making discussions on knowledge cut-off dates secondary. Our focus remains on objectively identifying and analyzing these gaps to accurately assess the practical limits of LLM capabilities in cybersecurity. Furthermore, with the rapid evolution of LLMs, the conclusions in this study may become outdated over time. Continued evaluation is essential to answer the question, “*how far have we come in achieving a digital cybersecurity expert?*” and to ensure that our findings reflect the latest advances and changes in LLM capabilities.

## 6. Conclusion

To assess the knowledge gaps in LLMs in fulfilling the role of a digital security expert, this study develops

a cybersecurity knowledge model based on cognitive science, encompassing 345 fine-grained knowledge points, and constructs a benchmark dataset, CSEBenchmark, containing 11,050 questions. Evaluation across 12 popular LLMs reveals that their overall accuracy is currently limited to 85.42%, with notable gaps in specialized procedural knowledge, such as the use of professional tools and uncommon commands. Additionally, different LLMs have unique knowledge gaps, and even larger models within the same family may underperform on certain knowledge points where smaller models perform better. By addressing these knowledge gaps, we achieve up to an 84% improvement in correcting previously incorrect predictions across three benchmarks for two cybersecurity tasks, thereby validating the effectiveness of our findings.

## Acknowledgments

We are grateful to our shepherd and the anonymous reviewers for their valuable guidance and insightful comments. This research is supported by Zhongguancun Laboratory and the Beijing Outstanding Young Scientist Program (No. JWZQ20240101008).

## References

- [1] M. Security, "Microsoft copilot for security," 2024. [Online]. Available: <https://www.microsoft.com/en-us/security/business/ai-machine-learning/microsoft-copilot-security>
- [2] G. Cloud, "Supercharge security with ai," 2024. [Online]. Available: <https://cloud.google.com/security/ai>
- [3] M. T. Alam, D. Bhushl, L. Nguyen, and N. Rastogi, "Ctibench: A benchmark for evaluating llms in cyber threat intelligence," *arXiv preprint arXiv:2406.07599*, 2024.
- [4] S. Srikanth, M. Hasanuzzaman, and F. T. Meem, "Evaluating the usability of llms in threat intelligence enrichment," *arXiv preprint arXiv:2409.15072*, 2024.
- [5] S. Shafee, A. Bessani, and P. M. Ferreira, "Evaluation of llm-based chatbots for osint-based cyber threat awareness," *Expert Systems with Applications*, p. 125509, 2024.
- [6] Z. Liu, J. Shi, and J. Buford, "Cyberbench: A multi-task benchmark for evaluating large language models in cybersecurity," 02 2024.
- [7] H. Ji, J. Yang, L. Chai, C. Wei, L. Yang, Y. Duan, Y. Wang, T. Sun, H. Guo, T. Li et al., "Sevenllm: Benchmarking, eliciting, and enhancing abilities of large language models in cyber threat intelligence," *arXiv preprint arXiv:2405.03446*, 2024.
- [8] S. Ullah, M. Han, S. Pujar, H. Pearce, A. Coskun, and G. Stringhini, "Llms cannot reliably identify and reason about security vulnerabilities (yet?): A comprehensive evaluation, framework, and benchmarks," in *IEEE Symposium on Security and Privacy*, 2024.
- [9] K. Alrashedy and A. Aljasser, "Can llms patch security issues?" *arXiv preprint arXiv:2312.00024*, 2023.
- [10] A. Zibaeirad and M. Vieira, "Vulnllmeval: A framework for evaluating large language models in software vulnerability detection and patching," *arXiv preprint arXiv:2409.10756*, 2024.
- [11] A. K. Zhang, N. Perry, R. Dulepet, E. Jones, J. W. Lin, J. Ji, C. Menders, G. Hussein, S. Liu, D. Jasper et al., "Cybench: A framework for evaluating cybersecurity capabilities and risk of language models," *arXiv preprint arXiv:2408.08926*, 2024.
- [12] R. Tian, Y. Ye, Y. Qin, X. Cong, Y. Lin, Z. Liu, and M. Sun, "Debugbench: Evaluating debugging capability of large language models," *arXiv preprint arXiv:2401.04621*, 2024.
- [13] C. Fang, N. Miao, S. Srivastav, J. Liu, R. Zhang, R. Fang, R. Tsang, N. Nazari, H. Wang, H. Homayoun et al., "Large language models for code analysis: Do {LLMs} really do their job?" in *33rd USENIX Security Symposium (USENIX Security 24)*, 2024, pp. 829–846.
- [14] L. Yang, C. Yang, S. Gao, W. Wang, B. Wang, Q. Zhu, X. Chu, J. Zhou, G. Liang, Q. Wang et al., "On the evaluation of large language models in unit test generation," in *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering*, 2024, pp. 1607–1619.
- [15] M. Bhatt, S. Chennabasappa, C. Nikolaidis, S. Wan, I. Evtimov, D. Gabi, D. Song, F. Ahmad, C. Aschermann, L. Fontana et al., "Purple llama cyberseval: A secure coding benchmark for language models," *arXiv preprint arXiv:2312.04724*, 2023.
- [16] J. Gong, N. Duan, Z. Tao, Z. Gong, Y. Yuan, and M. Huang, "How well do large language models serve as end-to-end secure code producers?" *arXiv preprint arXiv:2408.10495*, 2024.
- [17] Y. Yang, Y. Nie, Z. Wang, Y. Tang, W. Guo, B. Li, and D. Song, "Seccodeplt: A unified platform for evaluating the security of code genai," *arXiv preprint arXiv:2410.11096*, 2024.
- [18] Y. Liu, L. Gao, M. Yang, Y. Xie, P. Chen, X. Zhang, and W. Chen, "Vuldetechbench: Evaluating the deep capability of vulnerability detection with large language models," *arXiv preprint arXiv:2406.07595*, 2024.
- [19] Z. Liu, "Secqqa: A concise question-answering dataset for evaluating large language models in computer security," *arXiv preprint arXiv:2312.15838*, 2023.
- [20] N. Tihanyi, M. A. Ferrag, R. Jain, T. Bisztray, and M. Debbah, "Cybermetric: A benchmark dataset based on retrieval-augmented generation for evaluating llms in cybersecurity knowledge," in *2024 IEEE International Conference on Cyber Security and Resilience (CSR)*, 2024, pp. 296–302.
- [21] M. Levi, Y. Alluouche, D. Ohayon, and A. Puzanov, "Cyberpal.ai: Empowering llms with expert-driven cybersecurity instructions," *arXiv preprint arXiv:2408.09304*, 2024.
- [22] D. Bhusal, M. T. Alam, L. Nguyen, A. Mahara, Z. Lightcap, R. Frazier, R. Fieblinger, G. L. Torales, and N. Rastogi, "Secure: Benchmarking generative large language models for cybersecurity advisory," *arXiv preprint arXiv:2405.20441*, 2024.
- [23] G. Li, Y. Li, W. Guannan, H. Yang, and Y. Yu, "Seceval: A comprehensive benchmark for evaluating cybersecurity knowledge of foundation models," <https://github.com/XuanwuAI/SecEval>, 2023.
- [24] K. Ahmed, "Cyber security roadmap: Learn to become a cyber security expert," 2024. [Online]. Available: <https://roadmap.sh/cyber-security>
- [25] A. N, "Ethical hacking - roadmap.sh," 2024. [Online]. Available: <https://roadmap.sh/tr/ethical-hacking-yyvh9>
- [26] loredous, "From power button to pwn: A roadmap to computer security," 2024. [Online]. Available: <https://faq.hackncode.dev/from-power-button-to-pwn-a-roadmap-to-computer-security/>
- [27] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le, "Finetuned language models are zero-shot learners," *arXiv preprint arXiv:2109.01652*, 2021.
- [28] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell et al., "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.
- [29] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou et al., "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.

- [30] V. Clairoux-Trepanier, I.-M. Beauchamp, E. Ruellan, M. Paquet-Clouston, S.-O. Paquette, and E. Clay, "The use of large language models (llm) for cyber threat intelligence (cti) in cybercrime forums," *arXiv preprint arXiv:2408.03354*, 2024.
- [31] Z. L. Kucsván, M. Caselli, A. Peter, and A. Continella, "Inferring recovery steps from cyber threat intelligence reports," in *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer, 2024, pp. 330–349.
- [32] L. Huang and X. Xiao, "Ctikg: Llm-powered knowledge graph construction from cyber threat intelligence," in *First Conference on Language Modeling*, 2024.
- [33] S. Mitra, S. Neupane, T. Chakraborty, S. Mittal, A. Piplai, M. Gaur, and S. Rahimi, "Localintel: Generating organizational threat intelligence from global and local cyber knowledge," *arXiv preprint arXiv:2401.10036*, 2024.
- [34] R. Fieblinger, M. T. Alam, and N. Rastogi, "Actionable cyber threat intelligence using knowledge graphs and large language models," in *2024 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, 2024, pp. 100–111.
- [35] S. Hays and D. J. White, "Employing llms for incident response planning and review," *arXiv preprint arXiv:2403.01271*, 2024.
- [36] T. Ahmed, S. Ghosh, C. Bansal, T. Zimmermann, X. Zhang, and S. Rajmohan, "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, 2023, pp. 1737–1749.
- [37] Y. Chen, H. Xie, M. Ma, Y. Kang, X. Gao, L. Shi, Y. Cao, X. Gao, H. Fan, M. Wen et al., "Automatic root cause analysis via large language models for cloud incidents," in *Proceedings of the Nineteenth European Conference on Computer Systems*, 2024, pp. 674–688.
- [38] X. Zhang, S. Ghosh, C. Bansal, R. Wang, M. Ma, Y. Kang, and S. Rajmohan, "Automated root causing of cloud incidents using in-context learning with gpt-4," in *Companion Proceedings of the 32nd ACM International Conference on the Foundations of Software Engineering*, 2024, pp. 266–277.
- [39] G. Lu, X. Ju, X. Chen, W. Pei, and Z. Cai, "Grace: Empowering llm-based software vulnerability detection with graph structure and in-context learning," *Journal of Systems and Software*, vol. 212, p. 112031, 2024.
- [40] R. Ghosh, O. Farri, H.-M. von Stockhausen, M. Schmitt, and G. M. Vasile, "Cve-llm: Automatic vulnerability evaluation in medical device industry using large language models," *arXiv preprint arXiv:2407.14640*, 2024.
- [41] X. Du, G. Zheng, K. Wang, J. Feng, W. Deng, M. Liu, B. Chen, X. Peng, T. Ma, and Y. Lou, "Vul-rag: Enhancing llm-based vulnerability detection via knowledge-level rag," *arXiv preprint arXiv:2406.11147*, 2024.
- [42] Y. Yang, X. Zhou, R. Mao, J. Xu, L. Yang, Y. Zhang, H. Shen, and H. Zhang, "Dlap: A deep learning augmented large language model prompting framework for software vulnerability detection," *Journal of Systems and Software*, p. 112234, 2024.
- [43] J. Yu, Y. Chen, D. Tang, X. Liu, X. Wang, C. Wu, and H. Tang, "Llm-enhanced software patch localization," *arXiv preprint arXiv:2409.06816*, 2024.
- [44] T. Ahmed and P. Devanbu, "Better patching using llm prompting, via self-consistency," in *2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2023, pp. 1742–1746.
- [45] D. Hidvégi, K. Etemadi, S. Bobadilla, and M. Monperrus, "Cigar: Cost-efficient program repair with llms," *arXiv preprint arXiv:2402.06598*, 2024.
- [46] Y. Deng, C. S. Xia, C. Yang, S. D. Zhang, S. Yang, and L. Zhang, "Large language models are edge-case fuzzers: Testing deep learning libraries via fuzzgpt," *arXiv preprint arXiv:2304.02014*, 2023.
- [47] D. Wang, G. Zhou, L. Chen, D. Li, and Y. Miao, "Prophetfuzz: Fully automated prediction and fuzzing of high-risk option combinations with only documentation via large language model," in *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security*. Salt Lake City, UT, USA: Association for Computing Machinery, 2024.
- [48] P. Hu, R. Liang, and K. Chen, "Degpt: Optimizing decompiler output with llm," in *Proceedings 2024 Network and Distributed System Security Symposium (2024)*. <https://api.semanticscholar.org/CorpusID>, vol. 267622140, 2024.
- [49] H. Tan, Q. Luo, J. Li, and Y. Zhang, "Llm4decompile: Decompiling binary code with large language models," *arXiv preprint arXiv:2403.05286*, 2024.
- [50] X. She, Y. Zhao, and H. Wang, "Wadec: Decompiling webassembly using large language model," in *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering*, 2024, pp. 481–492.
- [51] H. Rong, Y. Duan, H. Zhang, X. Wang, H. Chen, S. Duan, and S. Wang, "Disassembling obfuscated executables with llm," *arXiv preprint arXiv:2407.08924*, 2024.
- [52] X. Ma, L. Luo, and Q. Zeng, "From one thousand pages of specification to unveiling hidden bugs: Large language model assisted fuzzing of matter {IoT} devices," in *33rd USENIX Security Symposium (USENIX Security 24)*, 2024, pp. 4783–4800.
- [53] R. Meng, M. Mirchev, M. Böhme, and A. Roychoudhury, "Large language model guided protocol fuzzing," in *Proceedings of the 31st Annual Network and Distributed System Security Symposium (NDSS)*, 2024.
- [54] J. Wang, L. Yu, and X. Luo, "Llmif: Augmented large language model for fuzzing iot devices," in *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 2024, pp. 196–196.
- [55] D. Liu, Z. Lu, S. Ji, K. Lu, J. Chen, Z. Liu, D. Liu, R. Cai, and Q. He, "Detecting kernel memory bugs through inconsistent memory management intention inferences," in *33rd USENIX Security Symposium (USENIX Security 24)*, 2024, pp. 4069–4086.
- [56] D. Nam, A. Macvean, V. Hellendoorn, B. Vasilescu, and B. Myers, "Using an llm to help with code understanding," in *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, 2024, pp. 1–13.
- [57] Y. Zhang, "Detecting code comment inconsistencies using llm and program analysis," in *Companion Proceedings of the 32nd ACM International Conference on the Foundations of Software Engineering*, 2024, pp. 683–685.
- [58] J. Luo, H. Shi, Y. Zhang, R. Wang, Y. Shen, Y. Chen, X. Shi, R. Liu, C. Hu, and Y. Jiang, "Cvecenter: Industry practice of automated vulnerability management for linux distribution community," in *Companion Proceedings of the 32nd ACM International Conference on the Foundations of Software Engineering*, 2024, pp. 329–339.
- [59] P. Liu, J. Liu, L. Fu, K. Lu, Y. Xia, X. Zhang, W. Chen, H. Weng, S. Ji, and W. Wang, "Exploring {ChatGPT}'s capabilities on vulnerability management," in *33rd USENIX Security Symposium (USENIX Security 24)*, 2024, pp. 811–828.
- [60] B. S. Bloom, M. D. Engelhart, E. J. Furst, W. H. Hill, and D. R. Krathwohl, "Taxonomy of educational objectives: Cognitive and affective domains," *New York: David McKay*, pp. 20–24, 1956.
- [61] PyMuPDF4LLM, "Pymupdf4llm - pymupdf 1.24.13 documentation," 2024. [Online]. Available: <https://pymupdf.readthedocs.io/en/latest/pymupdf4llm/>
- [62] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. [Online]. Available: <https://arxiv.org/abs/1908.10084>

[63] Y. Zhang, Y. Li, L. Cui, D. Cai, L. Liu, T. Fu, X. Huang, E. Zhao, Y. Zhang, Y. Chen et al., "Siren's song in the ai ocean: a survey on hallucination in large language models," *arXiv preprint arXiv:2309.01219*, 2023.

[64] OpenAI, "Api reference - openai api," 2024. [Online]. Available: <https://platform.openai.com/docs/api-reference/chat/create>

[65] Deepseek, "Your first api call — deepseek api docs," 2025. [Online]. Available: <https://api-docs.deepseek.com>

[66] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. E. Gonzalez, H. Zhang, and I. Stoica, "Efficient memory management for large language model serving with pagedattention," in *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

[67] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt, "Measuring massive multitask language understanding," *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[68] D. Hendrycks, C. Burns, S. Basart, A. Critch, J. Li, D. Song, and J. Steinhardt, "Aligning ai with shared human values," *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[69] X. Wang, B. Ma, C. Hu, L. Weber-Genzel, P. Röttger, F. Kreuter, D. Hovy, and B. Plank, "' my answer is c': First-token probabilities do not match text answers in instruction-tuned language models," *arXiv preprint arXiv:2402.14499*, 2024.

[70] X. Wang, C. Hu, B. Ma, P. Röttger, and B. Plank, "Look at the text: Instruction-tuned language models are more robust multiple choice selectors than you think," *arXiv preprint arXiv:2404.08382*, 2024.

[71] Q. Yu, Z. Zheng, S. Song, Z. Li, F. Xiong, B. Tang, and D. Chen, "xfinder: Robust and pinpoint answer extraction for large language models," *arXiv preprint arXiv:2405.11874*, 2024.

[72] J. Wang, X. Yi, R. Guo, H. Jin, P. Xu, S. Li, X. Wang, X. Guo, C. Li, X. Xu et al., "Milvus: A purpose-built vector data management system," in *Proceedings of the 2021 International Conference on Management of Data*, 2021, pp. 2614–2627.

[73] J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, and Z. Liu, "Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation," 2024.

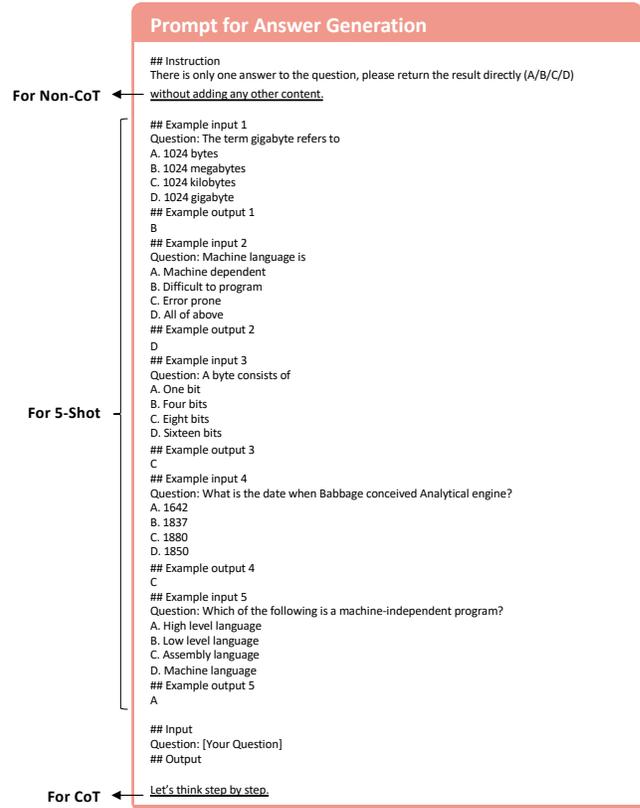


Figure 8. Prompt for answer generation.

## Appendix A. Prompts for Answer Generation

Figure 8 presents the prompt used for generating answers.

## Appendix B. Knowledge Points for Each Job Role

Due to space limitations, the mapped knowledge points for each job role are provided in our repository: <https://github.com/NASP-THU/CSEBenchmark>

## Appendix C. Topic Distribution in Semantic Space

Figure 9 presents the topic distribution in semantic space across LLMs.

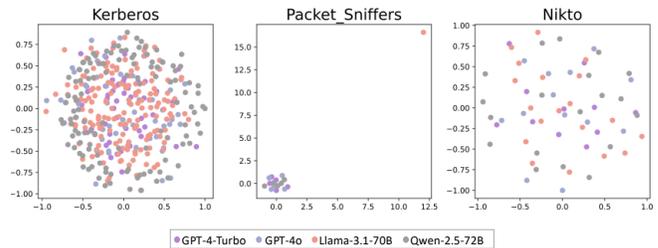


Figure 9. Topic distribution in semantic space across GPT-4-Turbo, GPT-4o, Llama-3.1-70B, and Qwen-2.5-72B.

## Appendix D. Meta-Review

The following meta-review was prepared by the program committee for the 2025 IEEE Symposium on Security and Privacy (S&P) as part of the review process as detailed in the call for papers.

### D.1. Summary

This paper proposes a new benchmark, CSEBenchmark, to evaluate the cybersecurity knowledge of Large Language Models. CSEBenchmark contains 11,050 questions, covering three types of knowledge: factual knowledge (to be memorized), conceptual knowledge (requiring understanding of underlying principles), and procedural knowledge (requiring hands-on practice). To construct the benchmark, it took 672 man-hours of reviewing the LLM-generated questions and 100 man-hours of corrections.

### D.2. Scientific Contributions

- Independent Confirmation of Important Results with Limited Prior Research
- Provides a New Data Set For Public Use
- Provides a Valuable Step Forward in an Established Field

### D.3. Reasons for Acceptance

- 1) This paper has independently confirmed important results with limited prior research. The paper demonstrates that having cybersecurity knowledge can significantly boost the performance of vulnerability detection and threat intelligence analysis tasks, via retrieval-augmented generation (RAG)
- 2) This paper provides a new data set for public use. CSEBenchmark enables a fine-grained and detailed evaluation of LLMs on cybersecurity knowledge.
- 3) This paper provides a valuable step forward in an established field. The paper provides comprehensive evaluation of cybersecurity expertise in popular LLM models and identifying their knowledge gaps in this area.

### D.4. Noteworthy Concerns

- 1) The dataset could be biased since only GPT-4-Turbo is used to generate the dataset. It might be more reasonable to use different LLMs to generate questions, combined with manual verification.
- 2) It is unclear whether the proposed evaluation framework approximates expert level knowledge of human security analysts.

## Appendix E. Response to the Meta-Review

**Response to concern 1.** Thank you for pointing out this issue. We discuss the impact of cyclical use in Section 5.1. In future work, we will explore using other advanced LLMs such as DeepSeek and Claude in GPT-4-Turbo's current role in question generation, enabling a more comprehensive evaluation through cross-model question generation and answering.

**Response to concern 2.** We acknowledge the importance of validating whether the proposed evaluation framework approximates expert-level knowledge of human security analysts. However, given the time and cost constraints of traditional expert surveys, we are currently unable to conduct such an experiment. Nonetheless, we believe our study provides a valuable benchmark for assessing LLM performance in cybersecurity tasks, and future work could incorporate expert evaluations to further refine the framework.