Cybersecurity with LLMs and RAGs: Challenges and Innovations

Marco Simoni[1,2] and Andrea Saracino[3]

[1] Istituto di Informatica e Telematica, Consiglio Nazionale Delle Ricerche, Pisa, Italy
[2] Università di Roma La Sapienza, Roma, Italy
[3] TeCIP, Scuola Universitaria Superiore Sant'Anna, Pisa, Italy
marco.simoni@iit.cnr.it, andrea.saracino@santannapisa.it

Abstract. Despite the significant advances that Large Language Models (LLMs) offer in processing vast amounts of data and providing actionable insights quickly, their application in the technical field of cybersecurity poses significant challenges. These include the tendency to produce hallucinatory and unreliable results when these models are tested on questions where factuality is important. Furthermore, while Retrieval Augmented Generation (RAG) systems are useful in enriching model answers with relevant information, they struggle with issues related to retrieval speed, choice of embeddings and thresholds and handling multi-hop queries. This paper describes these challenges and discusses strategies to overcome them in order to improve the adaptability and reliability of these models in responding to rapidly evolving cybersecurity threats.

Keywords: Large Language Models · Retrieval Augmented Generation · Threat Intelligence · Malware Analysis · Vulnerability Detection

1 Introduction

Generative AI improves cybersecurity by increasing efficiency in detecting and responding to threats, which is critical in an industry facing a skills shortage. AI-driven frameworks quickly analyze large data sets, provide actionable insights and accelerate threat identification to prevent security breaches [5]. The integration of Large Language Models (LLMs) and Retrieval Augmented Generation (RAG) systems based on Transformer architectures [43], has significantly improved natural language processing and expanded its use in areas such as code analysis and automated defense.

However, the application of LLMs and RAG systems in the field of cybersecurity poses significant challenges. Large language models often exhibit reliability problems in technical domains such as cybersecurity due to *hallucinations*, i.e. they produce inaccurate content when dealing with complex, underrepresented topics [18] [20]. The ability of LLMs to provide accurate answers correlates with their exposure to certain topics during training, highlighting a limitation in dealing with diverse and specialized knowledge [26].

Retrieval Augmented generation can overcome or limit hallucinations by providing the model with more relevant and richer information to draw on, reducing the dependence on having seen relevant information directly during training [18]. On the other hand, RAG systems face significant challenges such as scalability, slow retrieval speed, improper choice of embeddings and similarity thresholds, and difficulty in answering multi-hop question answering, which critically impact their operational efficiency and effectiveness [39] [47].

Contribution In this paper, we explore how Large Language Models (LLMs) and Retrieval Augmented Generation (RAG) systems can be used for cybersecurity threat analysis. We focus on the challenges these models face in improving cybersecurity expertise, vulnerability detection and malware analysis. Key topics include hallucinations, choosing the right RAG hyperparameters, processing complex multi-hop queries, and establishing appropriate evaluation criteria. We provide case studies and empirical evidence that demonstrate the practical benefits and limitations of current LLM and RAG models in cybersecurity. For example, one of the cases of cybersecurity expertise where factuality is required involves testing LLMs for CVE (Common Vulnerabilities and Exposures) questions, where we show how and why these models fail. For each challenge or limitation identified, the paper suggests possible solutions to improve the effectiveness and reliability of these models. The aim is to guide future research and development and ensure that LLMs in cybersecurity are robust, efficient and ethical.

Organisation. The rest of the paper is structured as follows. Section 2 gives an overview of related research on the application of LLMs for cybersecurity. Section 3 describes the challenges and possible solutions for LLMs in three different areas: *Cybersecurity Expertise*, *Malware Analysis*, and *Vulnerability Detection*. Section 4 describes the challenges and possible solutions that RAG systems face in the field of cybersecurity. Section 5 addresses the complexity associated with evaluating the effectiveness and reliability of LLMs and RAG systems in cybersecurity. The conclusions and future work are presented in Section 6.

2 Related Work

This section overviews recent developments in the application of Large Language Models in improving *Cybersecurity Expertise and Threat Intelligence*, *Malware Analysis* and *Vulnerability Detection*.

Cybersecurity Expertise and Threat Intelligence Research on chatbots in cybersecurity emphasizes their role in various aspects. Yoo et al. (2024) and Abu-Amara et al. (2024) discuss the impact of GDPR and gamified chatbots in education. Pieterse (2024) explores the use of ChatGPT in CTF cybersecurity challenges and highlights its limitations in providing direct solutions [50] [1] [32]. Chamberlain and Casey examine ChatGPT in penetration tests and CTF exercises and

find that it has the potential to create dynamic scenarios [4]. Happe et al. [13] demonstrates GPT-3.5's extension to penetration testing. Voros et al. [44] show how knowledge distillation from LLMs can effectively categorize URLs and improve scanning processes. Mitra et al. [28] present an automated system that uses LLMs to create organization-specific threat intelligence to improve SoC operations. Boffa et al. [3] make use of language models (LMs) to automate the analysis of Unix shell logs and improve the identification of attacker tactics. Juttner et al. [17] use ChatGPT to simplify IDS alerts for non-experts and improve home network security. Sewak et al. [36] integrate LLMs with Enterprise Knowledge Graphs to form Threat Intelligence Graphs, achieving up to 99% recall in detecting malicious scripts. Yu et al. [51] use GPT-3 to generate semantic honeywords that enhance security against breaches.

While these studies highlight the potential of LLMs and chatbots in cybersecurity, limitations persist. Reliance on existing data can lead to outdated responses if models are not updated. For instance, Pieterse [32] and Chamberlain and Casey [4] note ChatGPT's struggle with real-time problem-solving while Mitra et al. [28] and Sewak et al. [36] emphasize the need for robust frameworks for managing intelligence.

Malware Analysis Over the past two years, Large Language Models (LLMs) have been increasingly integrated into malware analysis frameworks for detection and deobfuscation. Patsakis et al. [31] explore LLMs' utility in deobfuscating Emotet malware. Devadiga et al. [7] introduce a model that merges Generative Adversarial Networks (GANs) and LLM embeddings to produce synthetic malware that successfully evades detection, significantly improving evasion rates. Li et al. [23] utilize the LLaMA-7b model for high-accuracy ransomware detection by analyzing grayscale bitmap images of Portable Executable files. Zahan et al. [52] employ LLMs to surpass traditional static analysis tools in detecting npm ecosystem malware, featuring a novel multi-stage decision-maker workflow that yields high precision and F1 scores. Lastly, Simoni and Saracino [35] leverage the BERT transformer for high-accuracy classification and categorization of Android malware, improving the understanding of complex API interactions.

These studies reveal significant limitations, including dependency on outdated data, limited context length affecting API sequence analysis, and the need for regular updates to maintain effectiveness against evolving threats, increasing operational overhead.

Vulnerability Detection This paragraph reviews various studies focused on the use of Large Language Models (LLMs) for vulnerability detection in software systems. Akuthota et al. [2] harness GPT-3.5 Turbo to perform in-depth vulnerability assessments of code snippets. Achieving an accuracy of 0.77, this study emphasizes the need for continuous methodological enhancements to boost the efficacy of security evaluations. Ullah et al. [42] provide a critical evaluation of various LLMs, highlighting their current limitations in consistently identifying and reasoning about security vulnerabilities. The study uses a bespoke evaluation framework named SecLLMHolmes, which conducts a granular analysis

across multiple dimensions to uncover the nondeterministic and often inaccurate reasoning of LLMs under varied coding scenarios. Sun et al. [38] set out to decouple the inherent vulnerability reasoning capabilities of LLMs from their other functional capabilities. By isolating these aspects, the study evaluates how LLMs perform when their reasoning is supported by structured prompts and external data sources. Wang et al. [45] introduce an innovative vulnerability detection model that combines LLMs with Conformer mechanisms. This hybrid approach aims to capitalize on both the global understanding capabilities of LLMs and the local feature detection strengths of Conformers. Lu et al. [25] integrate graph structural information and in-context learning into GRACE, an LLM framework to enhance software vulnerability detection.

These studies highlight the promise of LLMs in vulnerability detection but also reveal significant limitations. Key issues include the need for continuous methodological improvements, inconsistencies in reasoning, and the challenge of integrating additional data and frameworks to improve performance.

3 LLM Challenges and Possible Solutions

Large language models (LLMs) based on the Transformer architecture [43] have significantly advanced natural language processing (NLP). These models, trained on extensive text datasets, can generate coherent and contextually relevant text, translate, summarize, and answer questions. Models like GPT [34] revolutionized language understanding and production using unsupervised methods. GPT-3 extended these capabilities, enabling various NLP tasks without specialized training.

This section addresses the challenges LLMs face in cybersecurity, discussing *Cybersecurity Expertise*, *Malware Analysis*, and *Vulnerability Detection*, along with potential solutions.

3.1 Cybersecurity Expertise and Threat Intelligence

Challenges. Large language models often struggle with technical questions where accuracy is critical. This problem is commonly referred to as *Hallucination*, meaning that the models produce answers that are not true or reliable.

The problem of truthfulness in LLMs can be understood in terms of statistical uncertainty [30][16], which comes in two types: **epistemic** and **aleatoric** [48]. *Epistemic uncertainty* arises from a lack of knowledge about the correct answer. This can happen if the model does not have enough training data or if its capacity is limited. *Aleatory uncertainty* happens due to the inherent randomness of the prediction task. For example, some questions may have multiple correct answers.

In the field of cybersecurity, where threats evolve rapidly and precise responses are required, hallucinations in LLMs are particularly concerning. High epistemic certainty, i.e. a strong understanding of the correct responses, is essential for accurate predictions in this domain. There is a strong correlation between the number of times an LLM was exposed to documents on specific

fact-based questions during pretraining and its ability to answer these questions correctly [18]. Despite their size, even very large models have difficulty with questions that find little support in the training data. This shows that it is a major challenge to train LLMs to deal effectively with diverse and rare information. An example of hallucinations can be seen in Example 1.1, where models such as GPT-4[4], GPT-3.5[5] and MIXTRAL 7x8 [15] fail to find the correct answer to the question: *What is CVE-2019-15949*. Example 1.1 demonstrates that none of these models could accurately describe CVE-2019-15949, each giving different incorrect descriptions. The ground truth is that CVE-2019-15949 refers to a security vulnerability that affects Nagios XI versions before 5.6.6, allowing remote command execution as root. In contrast, GPT-4 incorrectly describes it as a vulnerability affecting phpMyAdmin. MIXTRAL 7x8 incorrectly relates it to a Remote Administration Tool called Imminent Monitor. GPT-3.5 incorrectly states it is a vulnerability in the Apache HTTP Server. Table 1 presents a performance comparison between GPT-4 and MIXTRAL 7x8 in identifying correct responses to a set of 50 questions related to CVEs. The metric *Accuracy* quantifies the percentage of correctly identified CVEs. Additionally, the metrics *Similarity* and *Correctness* are evaluated using the RAGAS framework [8], as detailed in Section 5.

Example 1.1: Hallucination on CVE Questions.

Question: What is CVE-2019-15949?
Ground Truth: CVE-2019-15949 refers to a security vulnerability that affects Nagios XI versions before 5.6.6, which allows remote command execution as root.
GPT-4 Response: CVE-2019-15949 refers to a security vulnerability that affects certain versions of phpMyAdmin, a very popular free and open-source administration tool for MySQL and MariaDB databases.
MIXTRAL 7x8 Response: CVE-2019-15949 is a vulnerability related to the Remote Administration Tool (RAT) called Imminent Monitor.
GPT-3.5 Response: CVE-2019-15949 refers to a vulnerability in the Apache HTTP Server, specifically in the mod_proxy module.

Table 1: Performance comparison of models on 50 CVE Queries.

Model	**Similarity (μ, σ)**	**Correctness (μ, σ)**	**Accuracy**
GPT-4	0.826, 0.0310	0.503, 0.0508	0.50
MIXTRAL 7x8	0.845, 0.039	0.53, 0.089	0.51

[4] https://chatgpt.com/
[5] https://chatgpt.com/?model=text-davinci-002-render-sha

Possible Solution. One possible solution to this type of problem is to enrich the context that forms the model prompt. Enriching the context in model prompts, especially through Retrieval Augmented Generation architectures, can significantly improve the quality of model responses by providing the model with more relevant and richer information on which to base its output (see Section 4). RAG can mitigate some of the limitations faced by LLMs by reducing their dependency on having directly seen relevant information during training [18]. When models can access relevant documents dynamically, they perform better on questions, especially those with sparse data support.
Example 1.2 shows how GPT-4 [6] has been able to provide the accurate answer if allowed to enrich its context, searching on the web before the generation of the answer.

Example 1.2: Context Enriching

Question: What is CVE-2019-15949?
GPT-4 Response: CVE-2019-15949 is a critical security vulnerability in Nagios XI, versions prior to 5.6.6. This vulnerability allows for remote command execution as root, which can be exploited under certain conditions.

However, to ensure the effectiveness of RAG systems, measures outlined in Section 4 should be implemented.
Table 2 compares the token generation probabilities between two language models, GEMMA 2b [27] and Mistral 7b [14], for a query related to *EternalBlue* vulnerability [24]: *What is the CVE related to EternalBlue Vulnerability?*. Mistral 7b, a larger model, correctly identifies the CVE, whereas GEMMA 2b makes errors and shows greater uncertainty. This highlights the importance of developing specific vocabularies to make LLMs proficient in cybersecurity. As seen, the CVE numbers are tokenized individually, increasing the probability of errors in both cases.

3.2 Malware Analysis

Challenges. When dealing with malware analysis, a major technical challenge for LLMs is the limitation of context length, which is particularly problematic for frameworks that rely on API sequence analysis to detect malware [35]. LLMs have a fixed input size, which limits the number of tokens they can process at once. This complicates the analysis of long API call sequences, which are necessary for the detection of malicious behavior.
This limitation can cause data to be truncated or split, losing important information and disrupting the temporal and logical relationships within API sequences, resulting in inaccurate detection. In addition, maintaining LLM effectiveness against evolving threats requires regular updates, which increases operational overhead.

[6] https://chatgpt.com/g/g-76OfIUJeh-gooogle-search/

GEMMA 2b	Probability	Mistral 7b	Probability
The	92.01%	The	75.63%
CVE	97.99%	E	51.44%
related	98.04%	ternal	99.96%
to	100.00%	Blue	99.98%
Eternal	99.89%	vulner	98.92%
Blue	99.68%	ability	100.00%
vulnerability	99.99%	is	92.90%
is	99.89%	related	67.36%
CVE	80.88%	to	100.00%
-	99.73%	C	83.02%
2	99.95%	VE	100.00%
0	99.84%	-	100.00%
1	62.65%	2	100.00%
7	52.36%	0	100.00%
-	99.98%	1	100.00%
0	74.16%	7	100.00%
1	39.62%	-	100.00%
5	28.39%	0	100.00%
2	20.35%	1	100.00%
.	95.10%	4	100.00%
EOS	99.89%	4	98.56%

Table 2: Token Generations and Probabilities for GEMMA 2b and mistral 7b Models

Possible Solutions. To circumvent the limitations of LLM context length, techniques such as *segmentation* [41] and *windowing* [10] split long API sequences into overlapping segments, preserving temporal and logical relationships for accurate malware detection. Models such as *Mamba* [11] and *Hyena* [33] further improve the handling of long sequences. Mamba uses selective state space models (SSMs) for input-dependent parameterization and effectively manages sequences up to one million in length, outperforming traditional transformers. Hyena combines long convolutions and data-driven gating for efficient, sub-quadratic alternatives to standard attention mechanisms.

3.3 Vulnerability Detection

Challenges The integration of LLMs with techniques such as fuzzing has improved test case generation and vulnerability detection capabilities in complex software systems [49]. Despite advances in vulnerability detection, there are several challenges that hinder the practical use of LLMs in cybersecurity. LLMs suffer from inconsistency and non-determinism as they produce different results under similar conditions, which is problematic for tasks that require high reliability [42]. They often misinterpret the source code, leading to inaccurate security assessments, and their sensitivity to minor code changes indicates a

lack of robustness [42]. Furthermore, LLMs rely on the integration of additional structural and contextual data to improve performance, as shown by the GRACE model [25]. Their significant computational cost may preclude resource-constrained organizations, and handling large datasets raises legal and privacy concerns, making compliance with data protection regulations difficult. The dynamic nature of cybersecurity threats necessitates frequent updates to LLMs, making them difficult to maintain and scale. These challenges underscore the need to adapt LLM technologies to meet the stringent requirements of security applications.

Possible Solutions Integrating LLMs with planning or rule-based systems [6] can leverage both data-driven insights and established security protocols, thereby enhancing the models' reasoning capabilities in complex environments.
Additionally, incorporating adversarial training can make LLMs more resilient to minor code modifications, enhancing their ability to generalize and reducing sensitivity to superficial changes. Developing models capable of continuous learning will allow them to adapt to new and emerging threats, maintaining their effectiveness over time.

4 RAG Limitations and Possible Solutions

Retrieval-Augmented Generation (RAG) extends traditional language models by integrating *Non-Parametric Knowledge Bases* [22] through *Retrievers* [19] [54] to improve accuracy and relevance for domain-specific queries [21] [12]. These *retrievers* retrieve contextually relevant information from their own knowledge base, inserting them into a *context* that the LLM uses to generate answers. The information in the knowledge base is represented by *Embeddings* [29]. These embeddings encode terms as vectors, so that similar terms have vectors that are close to each other. This proximity in vector space allows the embeddings to facilitate semantic matching between user queries and the information in the knowledge base.

Challenges Retrieval-Augmented Generation (RAG) systems face several key challenges that impact their effectiveness and operational efficiency:

1. **Scalability**: RAG systems' knowledge bases can grow large, consuming significant memory and limiting deployment in resource-constrained environments.
2. **Retriever Speed**: Fast retrieval of relevant documents is crucial in cybersecurity. Slow retriever response times can delay critical security measures and response to threats.
3. **Choice of Embeddings and Similarity Threshold**: Selecting the appropriate embeddings and setting the right similarity threshold for matching queries to documents in the knowledge base are critical. Incorrect choices can lead to poor retrieval accuracy, impacting the quality of the generated responses.

4. **Multi-Hop Question Answering**: Cybersecurity often involves complex, multi-hop queries that require drawing conclusions from interconnected entities. RAG systems often struggle with accurately answering multi-hop questions, where the answer requires synthesizing information from multiple entities [39].

Potential solutions for improving RAG systems in cybersecurity To improve RAG systems in cybersecurity, consider these targeted solutions:
Specialized Retrievers: Use different retrievers for different cybersecurity topics to better adapt to specific data characteristics.
Improve the retrieval speed: Restructure the knowledge base data by generating questions for each text section. Use these questions to match incoming queries and speed up the query by narrowing down the search space.
Different embeddings for different retrievers: Use different embeddings to minimize bias and improve the quality and impartiality of search results.
Setting similarity thresholds: Test retrievers with different questions and set similarity thresholds based on the median or third quartile of the similarity distributions of the most relevant documents. This approach provides a balance between precision and recognition. See Algorithm 1 for the procedure.
Processing Multi-Hop Queries: Implement entity recognition during preprocessing to create new questions for each identified entity, capturing comprehensive information for more accurate answers. As shown in Figure 1, the process for handling multi-hop requests begins with a complex user query: *What is the difference between Carberp and Rovnix?*. The system breaks down this query into simpler single-hop queries through the following steps:

1. **Entity Extraction:** Extracts key entities from the query: *Carberp* and *Rovnix*.
2. **Generation of New Queries:** Formulates two single-hop questions: *What is Carberp?* and *What is Rovnix?*
3. **Query Processing via RAG:** Processes each new query and the original query with the RAG system, retrieves relevant information, and summarizes it into a common context for the LLM to generate a response.

This approach enables the RAG system to efficiently handle multi-hop queries by simplifying them into single-hop queries.

5 Challenges in Evaluating RAG Systems and LLMs in Cybersecurity

The evaluation of Retrieval Augmented Generation systems and Large Language Models in the field of cybersecurity poses a particular challenge due to their dual role in information retrieval and content generation. The complexity of cybersecurity tasks combined with the lack of standardized benchmarks that cover a wide range of real-world deployment scenarios significantly complicates this evaluation process [8] [9] [37].

Algorithm 1 Setting Similarity Thresholds for Document Retrieval

```
procedure SetSimilarityThresholds(queries, documents, N)
    distribution ← empty list
    for each query in queries do
        scores ← computeSimilarities(query, documents)
        topScores ← getTopN(scores, N)
        distribution.append(topScores)
    end for
    return determineThreshold(distribution)
end procedure
function computeSimilarities(query, documents)
    return list of similarities(query, documents)
end function
function getTopN(scores, N)
    return sort and select top N fromscores
end function
function determineThreshold(distribution)
    if documents are text-heavy then
        return median(distribution)
    else
        return thirdQuartile(distribution)
    end if
end function
```

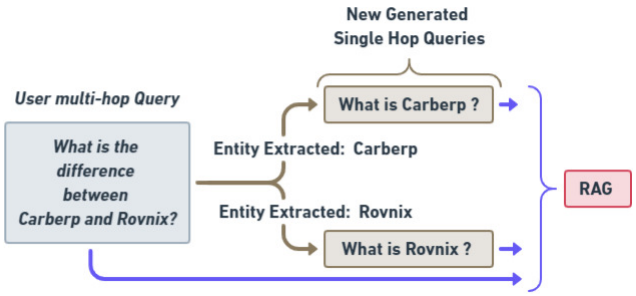

Fig. 1: Handling Multi-Hop Queries.

Challenges **Cybersecurity Dataset Limitations:** The development of LLM in cybersecurity is hindered by the lack of specialized datasets, as sensitive information limits the availability of data. This limitation hinders the learning of complex cybersecurity concepts. One exception is the dataset *Cybermetric* [40] with 10,000 questions on cybersecurity.

Accuracy of information retrieval: Ensuring the accuracy of information retrieved by RAG systems is critical in the cybersecurity field, as relevance and precision influence decision-making. Traditional metrics such as precision, recall and F1 score cannot fully capture the nuances required.

Effectiveness of LLMs in utilizing retrieved information: Evaluating how effectively LLMs integrate retrieved data into responses is critical. Metrics must assess the coherence and contextual appropriateness of the content generated in response to cybersecurity threats.
Quality of the generated content: The quality of the content of RAG systems and LLMs must be actionable, accurate and specific to cybersecurity. Comprehensive assessment frameworks are needed to evaluate the utility, accuracy and actionability of the generated responses.
Adaptation to real-world performance: Traditional evaluation methods often fail in dynamic cybersecurity environments. Evaluation frameworks should incorporate adaptive testing mechanisms to simulate real-world scenarios and measure LLM performance as threats evolve [46].

Possible Evaluation Solutions The RAGAS framework [8] offers customized metrics to evaluate RAG systems and LLMs in cybersecurity: *Answer Relevance* (cosine similarity between question embeddings and generated answers), *Answer Similarity* (semantic congruence with correct answers), and *Answer Correctness* (combining factual accuracy and semantic similarity). An innovative validation method involves using GPT-4 as a judge to evaluate model performance, leveraging its high agreement rate with human judgment (80%) [53]. GPT-4 assesses the quality of responses by selecting the most appropriate among competing models' answers to various cybersecurity queries.

6 Conclusions

Expertise is essential in the ever-evolving field of cybersecurity. The paper showed how Large Language Models suffer when answering cybersecurity questions that require factual knowledge, such as Common Vulnerabilities and Enumeration questions. It also became clear that these models are not able to handle long sequences, which is especially important when analysing malware. Retrieval augmented generation systems are helpful in reducing LLM's hallucinations as they provide more relevant and richer information on which to base the output of the models. However, this research has shown that these systems have problems in terms of scalability and handling multi-hop questions. Furthermore, the structures of the retriever and the choice of embeddings and similarity threshold need to be fine-tuned in order to develop fast and reliable RAG systems. Finally, the challenges in evaluating LLMs in the cybersecurity domain were pointed out, due to the lack of benchmark datasets and the difficulty in evaluating the dual role of RAG systems. Future work should focus on integrating LLMs and RAGs into a unique agent that preserves the efficiency of LLMs and the accuracy of RAGs when factuality is required.

References

1. Abu-Amara, F., Hosani, R., Tamimi, H., et al.: Spreading cybersecurity awareness via gamification: zero-day game. International Journal of Information Technology

(2024), https://link.springer.com/article/10.1007/s41870-024-01810-4
2. Akuthota, V., Kasula, R., Sumona, S.T., Mohiuddin, M., Reza, M.T., Rahman, M.M.: Vulnerability detection and monitoring using llm. In: 2023 IEEE 9th International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE). pp. 309–314. IEEE (2023)
3. Boffa, M., Drago, I., Mellia, M., Vassio, L., Giordano, D., Valentim, R., Houidi, Z.B.: Logprécis: Unleashing language models for automated malicious log analysis: Précis: A concise summary of essential points, statements, or facts. Computers & Security **141**, 103805 (2024)
4. Chamberlain, D., Casey, E.: Capture the flag with chatgpt: Security testing with ai chatbots. In: International Conference on Cyber Warfare and Security. vol. 19, pp. 43–54 (2024)
5. CrowdStrikes: Llms in cybersecurity: the bright side (2024), https://www.crowdstrike.com/cybersecurity-101/artificial-intelligence/large-language-model-llm/
6. Deng, Y., Zhang, W., Lam, W., Ng, S.K., Chua, T.S.: Plug-and-play policy planner for large language model powered dialogue agents. In: The Twelfth International Conference on Learning Representations (2024), https://openreview.net/forum?id=MCNqgUFTHI
7. Devadiga, D., Koo, H., Singh, A., Jin, G., Han, A., Chaudhari, K., Potdar, B., Shringi, A., Kumar, S.: Gleam: Gan and llm for evasive adversarial malware. In: 2023 14th International Conference on Information and Communication Technology Convergence (ICTC). pp. 53–58. IEEE (2023)
8. Es, S., James, J., Espinosa Anke, L., Schockaert, S.: RAGAs: Automated evaluation of retrieval augmented generation. In: Aletras, N., De Clercq, O. (eds.) Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations. pp. 150–158. Association for Computational Linguistics, St. Julians, Malta (Mar 2024), https://aclanthology.org/2024.eacl-demo.16
9. Gennari, J., Lau, S.h., Perl, S., Parish, J., Sastry, G.: Considerations for evaluating large language models for cybersecurity tasks (2024)
10. Gowda, V.P., Murugavelu, M., Thangamuthu, S.K.: Continuous kannada speech segmentation and speech recognition based on threshold using mfcc and vq. International Journal of Electrical and Computer Engineering **9**(6), 4684 (2019)
11. Gu, A., Dao, T.: Mamba: Linear-time sequence modeling with selective state spaces. arXiv preprint arXiv:2312.00752 (2023)
12. Guu, K., Lee, K., Tung, Z., Pasupat, P., Chang, M.: Retrieval augmented language model pre-training. In: III, H.D., Singh, A. (eds.) Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 119, pp. 3929–3938. PMLR (13–18 Jul 2020), https://proceedings.mlr.press/v119/guu20a.html
13. Happe, A., Cito, J.: Getting pwn'd by ai: Penetration testing with large language models. In: Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering. pp. 2082–2086 (2023)
14. Jiang, A.Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D.S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L.R., Lachaux, M.A., Stock, P., Scao, T.L., Lavril, T., Wang, T., Lacroix, T., Sayed, W.E.: Mistral 7b (2023), https://arxiv.org/abs/2310.06825

15. Jiang, A.Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D.S., Casas, D.d.l., Hanna, E.B., Bressand, F., et al.: Mixtral of experts. arXiv preprint arXiv:2401.04088 (2024)
16. Johnson, D.D., Tarlow, D., Duvenaud, D., Maddison, C.J.: Experts don't cheat: Learning what you don't know by predicting pairs. CoRR **abs/2402.08733** (2024). https://doi.org/10.48550/ARXIV.2402.08733, https://doi.org/10.48550/arXiv.2402.08733
17. Jüttner, V., Grimmer, M., Buchmann, E.: Chatids: Explainable cybersecurity using generative ai. arXiv preprint arXiv:2306.14504 (2023)
18. Kandpal, N., Deng, H., Roberts, A., Wallace, E., Raffel, C.: Large language models struggle to learn long-tail knowledge. In: International Conference on Machine Learning. pp. 15696–15707. PMLR (2023)
19. Karpukhin, V., et al.: Dense passage retrieval for open-domain question answering. arXiv preprint arXiv:2004.04906 (2020)
20. Kocoń, J., Cichecki, I., Kaszyca, O., Kochanek, M., Szydło, D., Baran, J., Bielaniewicz, J., Gruza, M., Janz, A., Kanclerz, K., et al.: Chatgpt: Jack of all trades, master of none. Information Fusion **99**, 101861 (2023)
21. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.t., Rocktäschel, T., et al.: Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in Neural Information Processing Systems **33**, 9459–9474 (2020)
22. Li, K., Zhou, H., Tu, Z., Feng, B.: Cskb: A cyber security knowledge base based on knowledge graph. In: Security and Privacy in Digital Economy: First International Conference, SPDE 2020, Quzhou, China, October 30–November 1, 2020, Proceedings 1. pp. 100–113. Springer (2020)
23. Li, X., Zhu, T., Zhang, W.: Efficient ransomware detection via portable executable file image analysis by llama-7b (2023)
24. Liu, Z.: Working mechanism of eternalblue and its application in ransomworm (2021), https://arxiv.org/abs/2112.14773
25. Lu, G., Ju, X., Chen, X., Pei, W., Cai, Z.: Grace: Empowering llm-based software vulnerability detection with graph structure and in-context learning. Journal of Systems and Software p. 112031 (2024)
26. Mallen, A., Asai, A., Zhong, V., Das, R., Khashabi, D., Hajishirzi, H.: When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In: Rogers, A., Boyd-Graber, J., Okazaki, N. (eds.) Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 9802–9822. Association for Computational Linguistics, Toronto, Canada (Jul 2023). https://doi.org/10.18653/v1/2023.acl-long.546, https://aclanthology.org/2023.acl-long.546
27. Mesnard, T., Hardin, C., Dadashi, R., Bhupatiraju, S., Pathak, S., Sifre, L., Rivière, M., Kale, M.S., Love, J., Tafti, P., Hussenot, L., Chowdhery, A., Roberts, A., Barua, A., Botev, A., Castro-Ros, A., Slone, A., Héliou, A., Tacchetti, A., Bulanova, A., Paterson, A., Tsai, B., Shahriari, B., Lan, C.L., Choquette-Choo, C.A., Crepy, C., Cer, D., Ippolito, D., Reid, D., Buchatskaya, E., Ni, E., Noland, E., Yan, G., Tucker, G., Muraru, G., Rozhdestvenskiy, G., Michalewski, H., Tenney, I., Grishchenko, I., Austin, J., Keeling, J., Labanowski, J., Lespiau, J., Stanway, J., Brennan, J., Chen, J., Ferret, J., Chiu, J., et al.: Gemma: Open models based on gemini research and technology. CoRR **abs/2403.08295** (2024). https://doi.org/10.48550/ARXIV.2403.08295, https://doi.org/10.48550/arXiv.2403.08295

28. Mitra, S., Neupane, S., Chakraborty, T., Mittal, S., Piplai, A., Gaur, M., Rahimi, S.: Localintel: Generating organizational threat intelligence from global and local cyber knowledge. arXiv preprint arXiv:2401.10036 (2024)
29. Muennighoff, N., Tazi, N., Magne, L., Reimers, N.: Mteb: Massive text embedding benchmark. arXiv preprint arXiv:2210.07316 (2022)
30. Osband, I., Wen, Z., Asghari, S.M., Dwaracherla, V., Ibrahimi, M., Lu, X., Roy, B.V.: Epistemic neural networks. In: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023 (2023), http://papers.nips.cc/paper_files/paper/2023/hash/07fbde96bee50f4e09303fd4f877c2f3-Abstract-Conference.html
31. Patsakis, C., Casino, F., Lykousas, N.: Assessing llms in malicious code deobfuscation of real-world malware campaigns. arXiv preprint arXiv:2404.19715 (2024)
32. Pieterse, H.: Friend or foe–the impact of chatgpt on capture the flag competitions. In: International Conference on Cyber Warfare and Security. vol. 19, pp. 268–276 (2024)
33. Poli, M., Massaroli, S., Nguyen, E., Fu, D.Y., Dao, T., Baccus, S., Bengio, Y., Ermon, S., Ré, C.: Hyena hierarchy: Towards larger convolutional language models. In: International Conference on Machine Learning. pp. 28043–28078. PMLR (2023)
34. Radford, A., et al.: Improving language understanding by generative pre-training. OpenAI Blog (2018)
35. Saracino, A., Simoni, M.: Graph-based android malware detection and categorization through bert transformer. In: Proceedings of the 18th International Conference on Availability, Reliability and Security. ARES '23, Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3600160.3605057, https://doi.org/10.1145/3600160.3605057
36. Sewak, M., Emani, V., Naresh, A.: Crush: Cybersecurity research using universal llms and semantic hypernetworks (2023)
37. Sultana, M., Taylor, A., Li, L., Majumdar, S.: Towards evaluation and understanding of large language models for cyber operation automation. In: 2023 IEEE Conference on Communications and Network Security (CNS). pp. 1–6. IEEE (2023)
38. Sun, Y., Wu, D., Xue, Y., Liu, H., Ma, W., Zhang, L., Shi, M., Liu, Y.: Llm4vuln: A unified evaluation framework for decoupling and enhancing llms' vulnerability reasoning. arXiv preprint arXiv:2401.16185 (2024)
39. Tang, Y., Yang, Y.: Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop queries (2024)
40. Tihanyi, N., Ferrag, M.A., Jain, R., Debbah, M.: Cybermetric: A benchmark dataset for evaluating large language models knowledge in cybersecurity. arXiv preprint arXiv:2402.07688 (2024)
41. Ulfattah, R.A., Endah, S.N., Kusumaningrum, R., Adhy, S.: Continuous speech segmentation using local adaptive thresholding technique in the blocking block area method. TELKOMNIKA (Telecommunication Computing Electronics and Control) **18**(1), 407–418 (2020)
42. Ullah, S., Han, M., Pujar, S., Pearce, H., Coskun, A., Stringhini, G.: Llms cannot reliably identify and reason about security vulnerabilities (yet?): A comprehensive evaluation, framework, and benchmarks. In: IEEE Symposium on Security and Privacy (2024)
43. Vaswani, A., et al.: Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems (2017)

44. Vörös, T., Bergeron, S.P., Berlin, K.: Web content filtering through knowledge distillation of large language models. In: 2023 IEEE International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT). pp. 357–361. IEEE (2023)
45. Wang, J., Huang, Z., Liu, H., Yang, N., Xiao, Y.: Defecthunter: A novel llm-driven boosted-conformer-based code vulnerability detection mechanism. arXiv e-prints pp. arXiv–2309 (2023)
46. Wang, S., Song, Y., Drozdov, A., Garimella, A., Manjunatha, V., Iyyer, M.: kNN-LM does not improve open-ended text generation. In: Bouamor, H., Pino, J., Bali, K. (eds.) Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. pp. 15023–15037. Association for Computational Linguistics, Singapore (Dec 2023). https://doi.org/10.18653/v1/2023.emnlp-main.929, https://aclanthology.org/2023.emnlp-main.929
47. Xiong, W., Li, X.L., Iyer, S., Du, J., Lewis, P., Wang, W.Y., Mehdad, Y., Yih, W.t., Riedel, S., Kiela, D., et al.: Answering complex open-domain questions with multi-hop dense retrieval. arXiv preprint arXiv:2009.12756 (2020)
48. Yadkori, Y.A., Kuzborskij, I., György, A., Szepesvári, C.: To believe or not to believe your llm (2024), https://arxiv.org/abs/2406.02543
49. Yao, Y., Duan, J., Xu, K., Cai, Y., Sun, Z., Zhang, Y.: A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. High-Confidence Computing p. 100211 (2024)
50. Yoo, C., Wolff, J., Lehr, W.: Lessons from gdpr for ai policymaking. Virginia Journal of Law & Technology (2024), https://scholarship.law.upenn.edu/cgi/viewcontent.cgi?article=1376&context=faculty_articles
51. Yu, F., Martin, M.V.: Honey, i chunked the passwords: Generating semantic honeywords resistant to targeted attacks using pre-trained language models. In: International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment. pp. 89–108. Springer (2023)
52. Zahan, N., Burckhardt, P., Lysenko, M., Aboukhadijeh, F., Williams, L.: Shifting the lens: Detecting malware in npm ecosystem with large language models. arXiv preprint arXiv:2403.12196 (2024)
53. Zheng, L., Chiang, W.L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., et al.: Judging llm-as-a-judge with mt-bench and chatbot arena. Advances in Neural Information Processing Systems **36** (2024)
54. Zhou, G., He, T., Zhao, J., Hu, P.: Learning continuous word embedding with metadata for question retrieval in community question answering. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 250–259 (2015)